

12. JUNI 2024



MAGNETIFY

AKTOREN UND SENSOREN

BOBBY BILALI, FOKKO VOS, ROBIN RUF, MAHIR GÖNEN
IPSO BILDUNG AG

Inhaltsverzeichnis

1	VERSIONIERUNG	2
2	AUSGANGSLAGE	3
3	ZIELE.....	3
4	MOCKUPS.....	4
5	TESTKONZEPT.....	5
5.1	TEST-CASES.....	5
6	TESTPROTOKOLL	6
6.1	USABILITY-TESTPROTOKOLL.....	6
6.2	UNIT-TESTS PROTOKOLL.....	7
7	MAUI-APP	8
7.1	STRUKTUR.....	8
7.2	KLASSEN	9
7.2.1	<i>Common.....</i>	<i>9</i>
7.2.2	<i>Services.....</i>	<i>9</i>
7.2.3	<i>ViewModels</i>	<i>9</i>
7.2.4	<i>Diagramm</i>	<i>10</i>
7.3	BIBLIOTHEKEN.....	11
8	VERWEISE.....	12
8.1	BILDVERZEICHNIS.....	12
8.2	TABELLENVERZEICHNIS	12

1 Versionierung

Version	Autor	Datum	Änderung
0.1	Fokko Vos	07.06.2024	Erstellung des Dokuments
0.2	Fokko Vos	07.06.2024	Ergänzung und Überführung
0.3	Mahir Gönen	07.06.2024	Kleinere Ergänzungen und Anpassungen
0.4	Mahir Gönen	07.06.2024	Überarbeitung
0.5	Robin Ruf	07.06.2024	Export Mockup eingefügt
0.6	Bobby Bilali	11.06.2024	Nacharbeitung Klassen
0.7	Robin Ruf	11.06.2024	Nacharbeitung Pakete
1.0	Fokko Vos	12.06.2024	Schlussversion

1 - Versionierung

2 Ausgangslage

In der heutigen technologiegetriebenen Welt ist die Integration von Sensoren in mobile Geräte allgegenwärtig, wobei jeder Sensor spezifische Funktionen und Anwendungen hat. Metallsensoren, die Magnetfelder erkennen können, sind ein grundlegendes Werkzeug in verschiedenen Bereichen, von der Sicherheit bis zur Datenerfassung. Unsere Applikation zielt darauf ab, diese Technologie zu nutzen, um eine mobile App zu entwickeln, die die Magnetfeldstärke in Mikrottesla (μT) misst. Diese Funktion hat Anwendungen in Bereichen wie der Sicherheitsüberprüfung, der Archäologie und sogar in der Freizeitsuche nach Metallen. Durch die Entwicklung dieser App als Teil unseres Kurses werden praktische Erfahrungen in der App-Entwicklung und im Umgang mit MAUI-Sensoren gesammelt.

3 Ziele

Das Hauptziel dieses Projekts ist die Entwicklung einer funktionellen Metalldetektor-App, die die Magnetfeldstärke mit Hilfe des Magnetfeldsensors eines mobilen Geräts misst. Die App soll folgende spezifische Ziele erreichen:

Funktionalität

Erkennung von Metallobjekten und Anzeige der Magnetfeldstärke in μT .

Benutzerinteraktion

Features sind die Aktivierung von Vibrationen und Geräusche bei der Detektion von Metall und die grafische Darstellung der Messwerte zur Verbesserung der Benutzererfahrung.

Technische Umsetzung

Implementierung der App unter Verwendung des MAUI-Frameworks und Sicherstellung der Kompatibilität auf Android-Geräten.

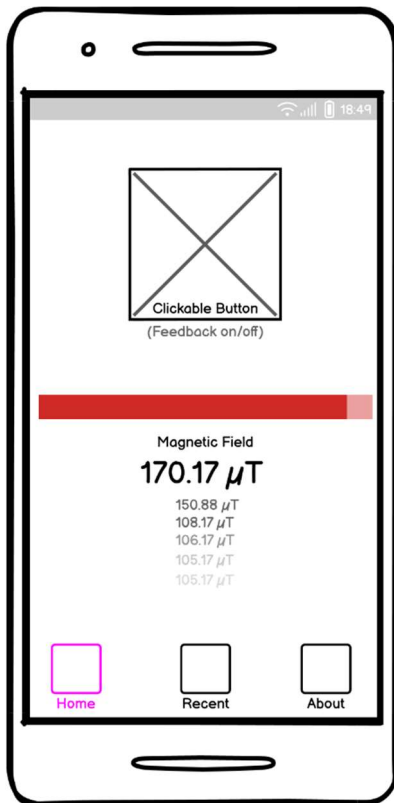
Selbstständiges Lernen

Förderung des selbstständigen Arbeitens und der praktischen Anwendung von Kenntnissen in der mobilen App-Entwicklung.

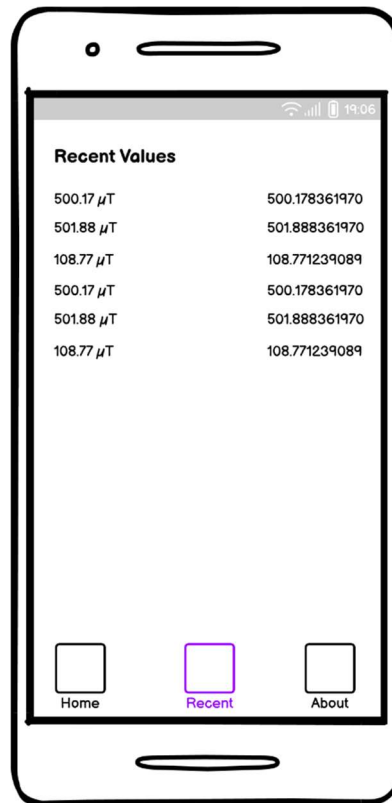
Dokumentation und Präsentation

Vollständige Dokumentation der Entwicklungsschritte und der technischen Details sowie eine abschliessende Präsentation der App.

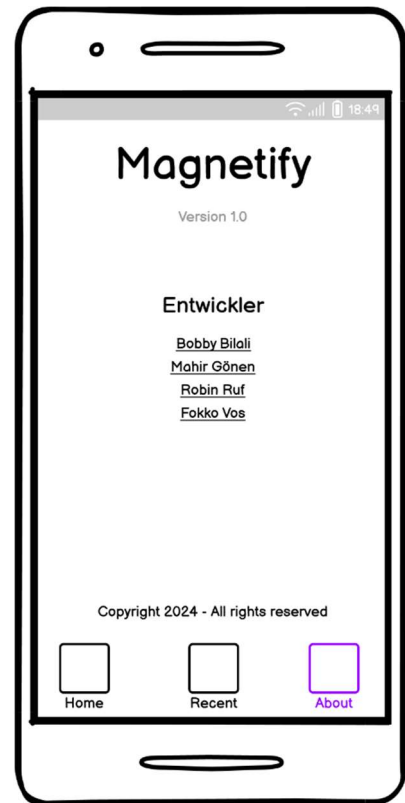
4 Mockups



1 - Mockup: HomePage



2 - Mockup: RecentPage



3 - Mockup: AboutPage

HomePage

Oben gibt es ein klickbares Logo, das als Schalter für Vibration und Ton dient.

Eine farbige visuelle Leiste repräsentiert die Intensität des Magnetfelds.

In der Mitte wird der aktuelle Magnetfeldwert in Mikrottesla (μT) gross angezeigt und unten werden die letzten gemessenen Werten aufgelistet.

Unten sind drei Buttons für die Navigation in der App.

RecentPage

Die Seite listet eine Historie der zuletzt gemessenen Werte des Magnetfelds in Mikrottesla, sowohl in einfacher als auch in detaillierter Form.

AboutPage

Zeigt die aktuelle Version und die Entwickler an. Die Namen sind verlinkt auf jeden einzelnen GitHub-Profil.

Ausserdem ist ein Copyright markiert.

5 Testkonzept

Um die Applikation Manuell zu Testen wurden einige Sinnvolle Use-Cases erstellt welche für den Test abgearbeitet werden sollen.

Neben Manuellen Usability Tests wurden für die Haupt-Funktionen der Applikation Unit-Tests geschrieben damit deren Funktion sichergestellt werden kann.

5.1 Test-Cases

<u>Usability-Testcases</u>		
Testnummer	Testbeschreibung	Erwartetes Ergebnis
U01	Installation der Applikation auf ein Android-Gerät	App lässt sich aufs Gerät installieren.
U02	Navigation durch die App	Navigation ist intuitiv und Benutzer findet sich leicht zurecht
U03	Lesbarkeit der Texte und Anzeigen	Texte und Anzeigen sind gut lesbar und verständlich
U04	Reaktionszeit der App auf Eingaben	App reagiert schnell und ohne Verzögerungen
U05	Erkennbarkeit der Icons und Buttons	Alle Icons und Buttons sind deutlich erkennbar und verständlich
U06	UI-Komponenten sind sichtbar und interaktiv	Alle Buttons und Anzeigen sind funktional und ansprechbar
U07	Magnet wird erkannt	Magnet wird erkannt. Bei eingeschalteter Vibration und Ton gibt es diese aus.
U08	Testen des Farbbalkens zur Feldstärkenanzeige	Farbbalken ändert sich entsprechend den Magnetfeldstärken

2 - Usability Test-Cases

6 Testprotokoll

6.1 Usability-Testprotokoll

Tester: Bobby Bilali

Datum/Uhrzeit: 11.06.2024 um 21:00 UTC

Gerät/Version: Samsung

Testnummer	Erwartetes Ergebnis	Erfüllt? (Ja/Nein*)	Kommentar vom Tester	*bei Nein
U01	App lässt sich aufs Gerät installieren.	Ja		<i>Ist der Buildvorgang komplett durch?</i>
U02	Navigation ist intuitiv und Benutzer findet sich leicht zurecht	Ja		-
U03	Texte und Anzeigen sind gut lesbar und verständlich	Ja		-
U04	App reagiert schnell und ohne Verzögerungen	Ja		-
U05	Alle Icons und Buttons sind deutlich erkennbar und verständlich	Ja		-
U06	Alle Buttons und Anzeigen sind funktional und ansprechbar	Ja		-
U07	Magnet wird erkannt. Bei eingeschalteter Vibration und Ton gibt es diese aus.	Ja		<i>Ist der obere Button (Logo) aktiv?</i>
U08	Farbbalken ändert sich entsprechend den Magnetfeldstärken	Ja		<i>Ändert sich der Wert unter «Magnetic Field»?</i>

3 - Usability-Testprotokoll

6.2 Unit-Tests Protokoll

Tester: Fokko Vos & Mahir Gönen

Datum/Uhrzeit: 12.06.2024 um 12:00 UTC

Gerät/Version: Windows (Visual Studio)

Test-Explorer

Bereit

Test	Dauer	M.	Fehlermeldung
✓ Magnetify.Tests (20)	397 ms		
✓ Magnetify.Tests.Common ...	39 ms		
✓ BetterCollectionFadeO...	22 ms		
✓ AddAtStart_AddsItem...	4 ms		
✓ AddAtStart_DoesNot...	< 1 ms		
✓ AddAtStart_FiresColle...	16 ms		
✓ AddAtStart_Updates...	< 1 ms		
✓ InitializationTest	< 1 ms		
✓ TrimExcessItems_Rem...	2 ms		
✓ BetterCollectionTests (5)	17 ms		
✓ AddAtStart_AddsItem...	16 ms		
✓ AddAtStart_Exceeds...	1 ms		
✓ AddAtStart_NotifiesC...	< 1 ms		
✓ AddAtStart_NotifiesC...	< 1 ms		
✓ TrimExcessItems_Rem...	< 1 ms		
✓ Magnetify.Tests.Services (...)	258 ms		
✓ MagnetometerServiceT...	229 ms		
✓ AddValue_ShouldUpd...	224 ms		
✓ Stop_ShouldSetRunni...	5 ms		
✓ SoundServiceTests (2)	17 ms		
✓ Disable_SetsDisableS...	17 ms		
✓ Enable_SetsDisableSo...	< 1 ms		
✓ VibrationServiceTests (3)	12 ms		
✓ Disable_SetsDisableVi...	< 1 ms		
✓ Enable_SetsDisableVi...	12 ms		
✓ VibrationService_Init...	< 1 ms		
✓ Magnetify.Tests.ViewMod...	100 ms		
✓ AboutViewModelTests (...)	100 ms		
✓ Constructor_Initialize...	32 ms		
✓ OpenUrl_InvalidUrl_D...	68 ms		

Gruppenzusammenfassung

Magnetify.Tests

Tests in Gruppe: 20

⌚ Dauer gesamt: 397 ms

Ergebnisse

✓ 20 Bestanden

4 - Unit-Tests Protokoll

Eine Log Datei der Tests ist im Repository unter **files** zu finden.

7 MAUI-App

7.1 Struktur

Die Maui-App wurde nach MVVM Pattern erstellt, um eine klare und strukturierte Trennung zwischen UI und Logik der App zu erhalten.

Ordner	Beschreibung
Common	Beinhaltet Dienstprogramme und Hilfsklassen, die in der gesamten Anwendung verwendet werden.
Data	Enthält Datenmodelle, die in der Anwendung verwendet werden.
Interfaces	Definiert Schnittstellen für die in der Anwendung verwendeten Dienste.
Platforms	Beinhaltet plattformspezifischen Code für Android, MacCatalyst, Tizen, Windows und iOS.
Properties	Beinhaltet Projekteinstellungen wie launchSettings.json.
Resources	Enthält statische Ressourcen wie Symbole, Schriftarten, Bilder und Stile.
Services	Enthält Service-Klassen, die Geschäftslogik und Interaktionen mit Hardware oder externen Diensten handhaben.
ViewModels	Enthält ViewModel-Klassen, die die Präsentationslogik verwalten.

4 - Projekt Struktur

Das Testprojekt spiegelt teile der Applikation welche getestet werden sollen. Die Ordnerstruktur ist also Identisch.

7.2 Klassen

Neben den Standard Klassen wie **MauiProgram**, **App** und **AppShell** verwendet Magnetify zahlreiche weitere Klassen welche zusammen die Ordentliche Funktionalität der App sicherstellen.

7.2.1 Common

Klasse	Beschreibung
BaseNotifyHandler	Eine Basisklasse, die INotifyPropertyChanged implementiert, um Änderungen von Eigenschaften in der UI zu benachrichtigen. Vereinfacht dessen Verwendung.
BetterCollection<T>	Eine Sammlung, die es ermöglicht, Elemente am Anfang hinzuzufügen und das letzte Element zu entfernen, wenn die maximale Länge überschritten wird. Benachrichtigungen über Änderungen werden nach allen Modifikationen gesendet.
BetterCollectionFadeOut	Ähnlich wie BetterCollection, jedoch mit OpacityItem-Elementen, deren Deckkraft abnimmt, je älter sie sind.

5 - Übersicht: Common

7.2.2 Services

Dienste liegen im Ordner **Services**

Klasse	Beschreibung
AlertService	Erlaubt Alerts von ViewModels aus zu Senden.
MagnetometerService	Hauptverantwortlich für die Kommunikation mit dem Magnetometer.
SoundService	Haupt Schnittstelle für das Abspielen von Audio.
VibrationService	Haupt Schnittstelle für das Vibrieren des Gerätes.

6 - Übersicht: Services

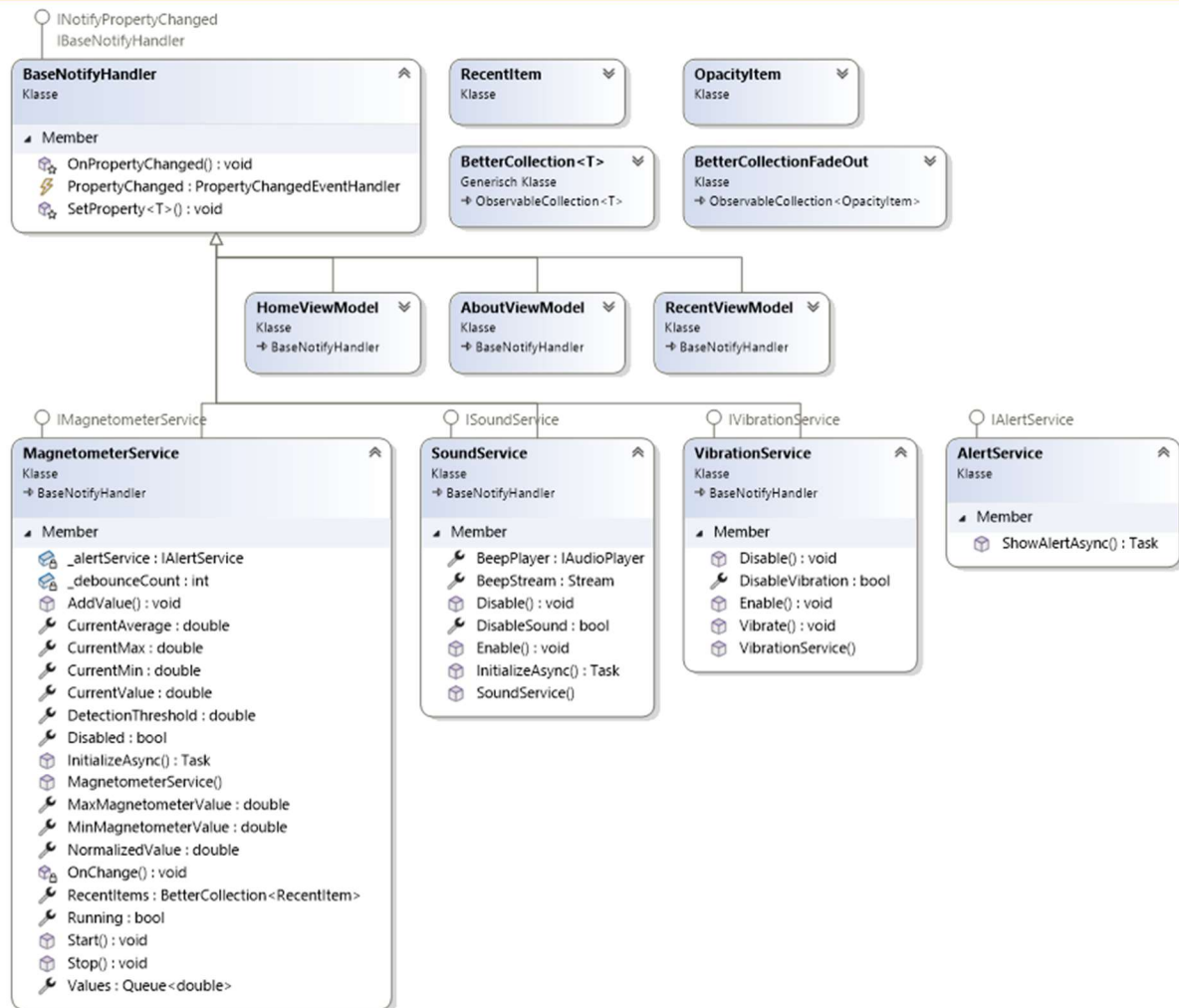
7.2.3 ViewModels

View Modelle liegen im Ordner **ViewModels**

Klasse	Beschreibung
AboutViewModel	Verantwortlich für die Logik von AboutPage
HomeViewModel	Verantwortlich für die Logik von HomePage
RecentViewModel	Verantwortlich für die Logik von RecentPage

7 - Übersicht: ViewModels

7.2.4 Diagramm



5 - Klassendiagramm

7.3 Bibliotheken

Paketname	Version	Beschreibung
Microsoft.Extensions.Logging.Debug	8.0.0	Dieses Paket stellt eine Implementierung des Debug-Ausgabe-Loggeranbieters für <code>Microsoft.Extensions.Logging</code> bereit. Es ermöglicht das Protokollieren von Nachrichten an einen Debugger-Monitor, indem Nachrichten mit <code>System.Diagnostics.Debug.WriteLine()</code> geschrieben werden.
Microsoft.Maui.Controls	8.0.7	.NET Multi-platform App UI (MAUI) ist ein plattformübergreifendes Framework zum Erstellen nativer mobiler und Desktop-Apps mit C# und XAML. Mit .NET MAUI können Sie Apps entwickeln, die auf Android, iOS, macOS, Tizen und Windows ausgeführt werden.
Microsoft.Maui.Controls.Compatibility	8.0.7	Dieses Paket enthält Kompatibilitäts-APIs und -Objekte für .NET Multi-platform App UI (MAUI) Apps. Es enthält eine Sammlung von APIs und Ansichten, die eine einfachere Migration von <code>Xamarin.Forms</code> ermöglichen.
Microsoft.NET.ILLink.Tasks	8.0.4	Dieses Paket enthält MSBuild-Aufgaben zum Ausführen des IL Linkers.
Plugin.Maui.Audio	3.0.0	Dieses Paket stellt die Fähigkeit bereit, Audio in einer .NET MAUI-Anwendung abzuspielen und aufzunehmen.
PropertyChanged.Fody	4.1.0	Dieses Paket fügt allen Klassen, die <code>INotifyPropertyChanged</code> implementieren, eine Benachrichtigung über Eigenschaftenänderungen hinzu.

8 - Bibliotheken

Grobe Beschreibungen von GPT

8 Verweise

8.1 Bildverzeichnis

1 - MOCKUP: HOMEPAGE.....	4
2 - MOCKUP: RECENTPAGE	4
3 - MOCKUP: ABOUTPAGE.....	4
4 - UNIT-TESTS PROTOKOLL	7
5 - KLASSENDIAGRAMM	10

8.2 Tabellenverzeichnis

1 - VERSIONIERUNG	2
2 - USABILITY TEST-CASES	5
3 - USABILITY-TESTPROTOKOLL	6
4 - PROJEKT STRUKTUR.....	8
5 - ÜBERSICHT: COMMON	9
6 - ÜBERSICHT: SERVICES	9
7 - ÜBERSICHT: VIEWMODELS	9
8 - BIBLIOTHEKEN.....	11