

12. Juni 2024

Bobby Bilali, Fokko Vos, Robin Ruf, Mahir Gönen

ipso Bildung AG

Magnetify

Aktoren und Sensoren

Inhaltsverzeichnis

[1 Versionierung 2](#_Toc169108628)

[2 Ausgangslage 3](#_Toc169108629)

[3 Ziele 3](#_Toc169108630)

[4 Mockups 4](#_Toc169108631)

[5 Testkonzept 5](#_Toc169108632)

[5.1 Test-Cases 5](#_Toc169108633)

[6 Testprotokoll 6](#_Toc169108634)

[6.1 Usability-Testprotokoll 6](#_Toc169108635)

[6.2 Unit-Tests Protokoll 7](#_Toc169108636)

[7 MAUI-App 8](#_Toc169108637)

[7.1 Struktur 8](#_Toc169108638)

[7.2 Klassen 9](#_Toc169108639)

[7.2.1 Common 9](#_Toc169108640)

[7.2.2 Services 9](#_Toc169108641)

[7.2.3 ViewModels 9](#_Toc169108642)

[7.2.4 Diagramm 10](#_Toc169108643)

[7.3 Bibliotheken 11](#_Toc169108644)

[8 Verweise 12](#_Toc169108645)

[8.1 Bildverzeichnis 12](#_Toc169108646)

[8.2 Tabellenverzeichnis 12](#_Toc169108647)

# 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

# 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 Mikrotesla (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.

# 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.

# Mockups

|  |  |  |
| --- | --- | --- |
| 1 - Mockup: HomePage | Ein Bild, das Text, Screenshot, Handy, Kommunikationsgerät enthält.  Automatisch generierte Beschreibung  2 - Mockup: RecentPage | 3 - Mockup: AboutPage |
| **HomePage** | **RecentPage** | **AboutPage** |
| 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 Mikrotesla (µT) gross angezeigt und unten werden die letzten gemessenen Werten aufgelistet.  Unten sind drei Buttons für die Navigation in der App. | Die Seite listet eine Historie der zuletzt gemessenen Werte des Magnetfelds in Mikrotesla, sowohl in einfacher als auch in detaillierter Form. | Zeigt die aktuelle Version und die Entwickler an. Die Namen sind verlinkt auf jeden einzelnen GitHub-Profil.  Ausserdem ist ein Copyright markiert. |

# 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.

## Test-Cases

|  |  |  |
| --- | --- | --- |
| Usability-Testcases | | |
| 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

# Testprotokoll

## 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 |  | *Ist der Buildvorgang komplett durch?* |
| 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 |  | *Ist der obere Button (Logo) aktiv?* |
| U08 | Farbbalken ändert sich entsprechend den Magnetfeldstärken | Ja |  | *Ändert sich der Wert unter «Magnetic Field»?* |

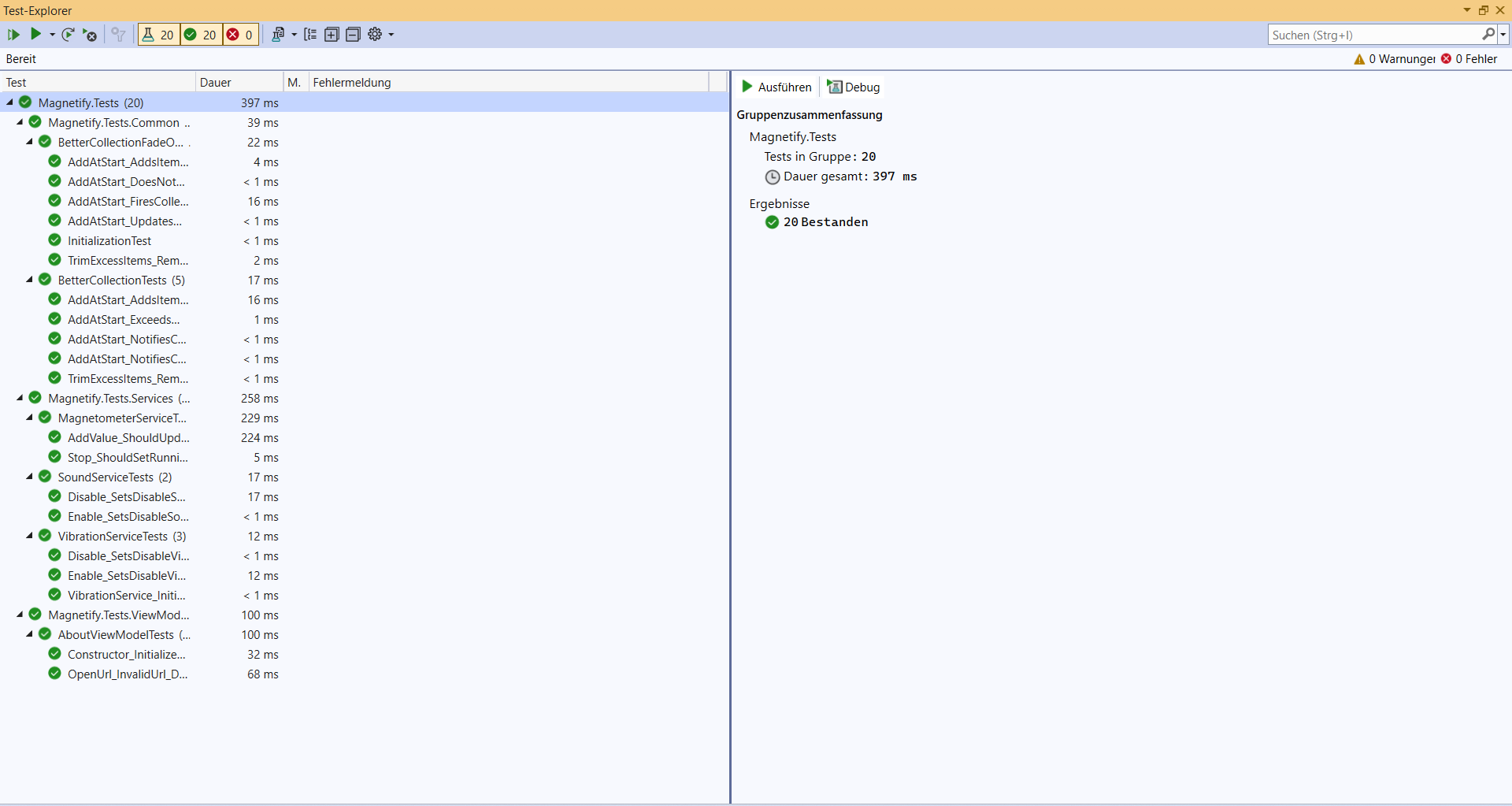
3 - Usability-Testprotokoll

## Unit-Tests Protokoll

**Tester:** Fokko Vos & Mahir Gönen

**Datum/Uhrzeit:** 12.06.2024 um 12:00 UTC

**Gerät/Version:** Windows (Visual Studio)

****

4 - Unit-Tests Protokoll

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

# MAUI-App

## 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.

## Klassen

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

### 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

### 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

### 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

### Diagramm

Ein Bild, das Text, Screenshot, Display, Software enthält.

Automatisch generierte Beschreibung

5 - Klassendiagramm

## Bibliotheken

|  |  |  |
| --- | --- | --- |
| **Paketname** | **Version** | **Beschreibung** |
| Microsoft.Extensions.Logging.Debug | 8.0.0 | Dieses Paket stellt eine Implementierung des Debug-Ausgabe-Loggeranbieters für Microsoft.Extensions.Logging bereit. Es ermöglicht das Protokollieren von Nachrichten an einen Debugger-Monitor, indem Nachrichten mit System.Diagnostics.Debug.WriteLine() 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 Xamarin.Forms 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 INotifyPropertyChanged implementieren, eine Benachrichtigung über Eigenschaftenänderungen hinzu. |

8 - Bibliotheken

Grobe Beschreibungen von GPT

# Verweise

## Bildverzeichnis

[1 - Mockup: HomePage 4](#_Toc169108648)

[2 - Mockup: RecentPage 4](#_Toc169108649)

[3 - Mockup: AboutPage 4](#_Toc169108650)

[4 - Unit-Tests Protokoll 7](#_Toc169108651)

[5 - Klassendiagramm 10](#_Toc169108652)

## Tabellenverzeichnis

[1 - Versionierung 2](#_Toc169108653)

[2 - Usability Test-Cases 5](#_Toc169108654)

[3 - Usability-Testprotokoll 6](#_Toc169108655)

[4 - Projekt Struktur 8](#_Toc169108656)

[5 - Übersicht: Common 9](#_Toc169108657)

[6 - Übersicht: Services 9](#_Toc169108658)

[7 - Übersicht: ViewModels 9](#_Toc169108659)

[8 - Bibliotheken 11](#_Toc169108660)