



Projektdokumentation

Widerstandsrechner zum Enkodieren und Dekodieren der Werte auf einem Widerstand

Auszubildener:

Ausbildungsbetrieb:

Berufsförderungswerk Schömburg gGmbH
Bühlhof 6
75328 Schömburg

Inhaltsverzeichnis

Projektinitialisierung	1
1.1 Analyse	1
1.2 Recherche	3
1.3 Zusammenfassung der Analyse	4
2. Projektplanung	5
2.1 Meilensteine festlegen	5
2.2 Zeitplanerstellung	5
3. Projektdurchführung	7
3.1 Grafiken Vorbereitung in GIMP	7
3.2 Oberflächengestaltung	7
3.3 Hilfsklasse	8
3.4 Decoding Klasse	9
3.5 Encoding Klasse	11
4. Qualitätssicherung	12
4.1 Testanforderungen bestimmen	12
4.2 Dokumentation der Testergebnisse	12
4.3 Optimierung	13
5. Projektabschluss	14
5.1 Übergabe des Projektes	14
5.2 Projektdokumentation erstellen	14
6. Abkürzungsverzeichnis	15
7. Abbildungsverzeichnis	16

1. Projektinitialisierung

Während eines Meetings mit den Kollegen aus der Elektronikabteilung kamen wir zu dem Ergebnis, dass eine App hilfreich wäre, die sowohl die Widerstandsbänder basierend auf den vom Benutzer eingegebenen Daten darstellt als auch die Farbbänder ausliest und den Gesamtwiderstandswert ermittelt.

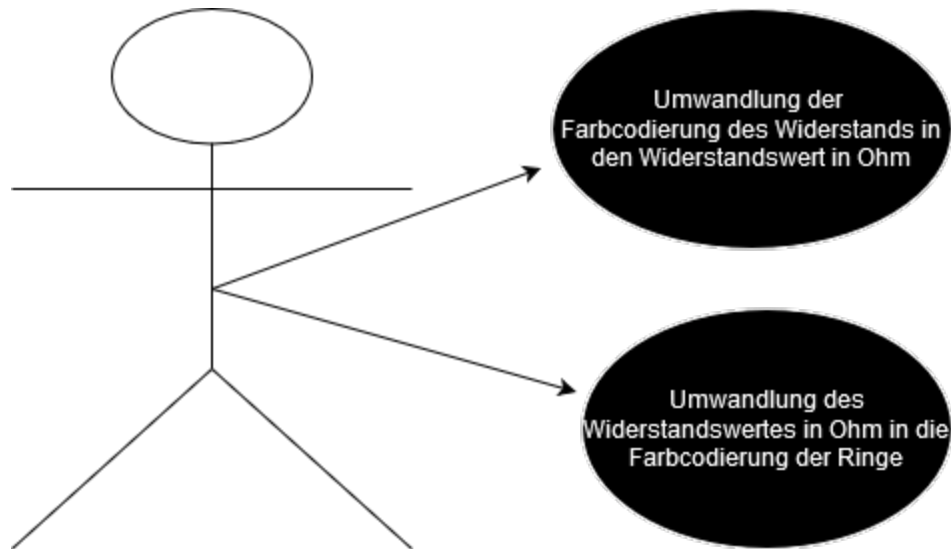


Abbildung 1 Use Case Diagramm

Dieses Projekt hat zum Ziel, eine benutzerfreundliche Android-App zu entwickeln, die die Kodierung und Dekodierung von Widerstandswerten erleichtert. Die App richtet sich an Techniker und Ingenieure in der Elektronikbranche, die regelmäßig mit Widerständen arbeiten.

1.1 Analyse

Mein Kollege aus der Elektroabteilung war sehr genau bei der Definition der Anforderungen für die App:

1. Visualisierung der Widerstandskodierung:

- Die Anwendung muss eine Kodierung des Widerstands visuell darstellen. Dies bedeutet, dass die App in der Lage sein muss, Widerstandswerte in farbige Ringe zu übersetzen, wie es bei traditionellen Widerständen der Fall ist.
-

2. Dekodierung des Widerstandswertes:

- Die Anwendung muss den Wert des Widerstands aus der Kodierung dekodieren können. Das bedeutet, dass die App die Farbcodierung eines Widerstands aufnehmen und den entsprechenden numerischen Wert berechnen und anzeigen muss.

3. **Einfache und schnelle Dateneingabe:**

- Die Eingabe der Daten muss einfach und relativ schnell sein. Die Benutzeroberfläche sollte intuitiv und benutzerfreundlich gestaltet sein, damit Nutzer ohne großen Aufwand die notwendigen Daten eingeben können.

4. **Kompatibilität mit dem Android-System:**

- Die App muss für das Android-System geeignet sein. Dies umfasst die Kompatibilität mit verschiedenen Android-Versionen und -Geräten, um eine breite Nutzerbasis abzudecken.

5. **Unterstützung für fünf Ringe:**

- Die Anwendung muss ausschließlich für elektrische Widerstände mit fünf Farbringen ausgelegt sein.

Das Layout und das Farbschema spielen für ihn keine Rolle, solange die ersten vier Punkte umgesetzt werden. Das bedeutet, dass die Funktionalität der App oberste Priorität hat, während das Design und die ästhetischen Aspekte zweitrangig sind.

Aufgrund der oben genannten Bedürfnisse und Anforderungen habe ich entschieden:

1. **Realisierungszeit:**

- Die App ist in 40 Stunden realisierbar. Dies beinhaltet die Entwicklung, das Testen und die Implementierung der grundlegenden Funktionalitäten.

2. **Entwicklungsumgebung:**

- Die App soll in Android Studio gebaut werden. Android Studio bietet eine umfassende Entwicklungsumgebung, die speziell für die Android-Entwicklung optimiert ist und zahlreiche Tools und Bibliotheken bereitstellt.

3. **Programmiersprache:**

- Die Programmierung soll in Java erfolgen. Java ist eine weit verbreitete Programmiersprache, die von Android nativ unterstützt wird und eine robuste und stabile Grundlage für die Entwicklung bietet.

4. **Architektur der App:** Die App folgt dem Model-View-Controller (MVC) Architekturpattern, um die Trennung von Logik und UI (User Interface) zu gewährleisten.

5. **Grafische Ressourcen:**

- Die App benötigt grafische Ressourcen. Dies umfasst unter anderem Icons, Grafiken für die Widerstandskodierungen und andere visuelle Elemente, die für eine ansprechende Benutzererfahrung erforderlich sind.

6. Klassendefinitionen:

- In der App sollen vier Klassen definiert werden:
 - **MainActivity:** Dies ist die Hauptklasse der App und dient als Einstiegspunkt. Sie verwaltet die Benutzeroberfläche und die Interaktionen des Nutzers.
 - **Hilfskla**
 - **sse:** Diese Klasse wird zusätzliche Ressourcen und Methoden enthalten, die von verschiedenen Teilen der App genutzt werden können, um den Code modular und wiederverwendbar zu gestalten.
 - **Decoding:** Diese Klasse enthält die Methoden zur Umwandlung von Widerstandswerten in visuelle Codes. Sie nimmt numerische Eingaben und gibt die entsprechenden Farbkodierungen aus.
 - **Encoding:** Diese Klasse enthält die Methoden zur Ermittlung des Widerstandswerts aus der visuellen Repräsentation. Sie nimmt Farbkodierungen als Eingabe und gibt den berechneten Widerstandswert aus.

Zusammenfassend lässt sich sagen, dass die geplante App eine benutzerfreundliche und funktionale Lösung bieten soll, um die Kodierung und Dekodierung von Widerstandswerten zu visualisieren und zu berechnen. Durch die Verwendung bewährter Entwicklungswerkzeuge und -techniken wird sichergestellt, dass die App zuverlässig und effizient arbeitet.

1.2 Recherche

Ich habe die folgenden Apps für das Betriebssystem aus dem Play Store getestet:

Elektronic: Electrocalc

- Benötigt eine Datenschutzeinwilligung und enthält Werbung, die die Nutzung der App stört.
- Die App bietet nur die Möglichkeit, den Wert des Widerstands aus farbigen Ringen zu dekodieren.

WiderstandsRechner

- Bietet eine attraktive Oberfläche mit modernen und bequemen Eingabemöglichkeiten.
- Die App ermöglicht auf einer Oberfläche sowohl das Dekodieren als auch das Enkodieren von Widerstandswerten.
- Die App erlaubt die Bearbeitung von Widerständen mit 3, 4, 5 und 6 Ringen.

Electrodoc

- Benötigt eine Datenschutzeinwilligung und enthält Werbung.
- Die App bietet nur die Möglichkeit, den Wert des Widerstands aus farbigen Ringen zu dekodieren.

Widerstandsberechnung

- Die App ermöglicht auf zwei Oberflächen sowohl das Dekodieren als auch das Enkodieren von Widerstandswerten.
- Sie erlaubt die Bearbeitung von Widerständen mit 3, 4, 5 und 6 Ringen.

Resistor Color Code Calculator

- Benötigt eine Datenschutzeinwilligung und enthält Werbung.
- Die App bietet nur die Möglichkeit, den Wert des Widerstands aus farbigen Ringen zu dekodieren.
- Die App erlaubt die Bearbeitung von Widerständen mit 3, 4, 5 und 6 Ringen.

Widerstandsfarbcodes

- Bietet eine Oberfläche mit bequemen Eingabemöglichkeiten.
- Die App ermöglicht auf einer Oberfläche sowohl das Dekodieren als auch das Enkodieren von Widerstandswerten.
- Die App erlaubt die Bearbeitung von Widerständen mit 4, 5 und 6 Ringen.

Widerstand Farbcodes-Rechner

- Bietet eine Oberfläche mit bequemen Eingabemöglichkeiten.
- Die App ermöglicht auf einer Oberfläche sowohl das Dekodieren als auch das Enkodieren von Widerstandswerten.
- Die App erlaubt die Bearbeitung von Widerständen mit 4 Ringen.

1.3 Zusammenfassung der Analyse

Ich habe folgende Erfahrungen gesammelt:

Meine App kann Werbung enthalten, die jedoch den Nutzer nicht stören darf. Aufgrund mangelnder Kenntnisse im GIMP-Umfeld kann ich nicht dieselben Eingabemöglichkeiten wie im WiderstandsRechner bieten.

Meine App soll attraktiv sein, indem sie Funktionalität bietet: Sie ermöglicht auf einer Oberfläche sowohl das Dekodieren als auch das Enkodieren von Widerstandswerten.

Die Analyse hat mich 1 Stunde gekostet.

2. Projektplanung

2.1 Meilensteine festlegen

Meine Meilensteine zeigt das untenstehende Diagramm.

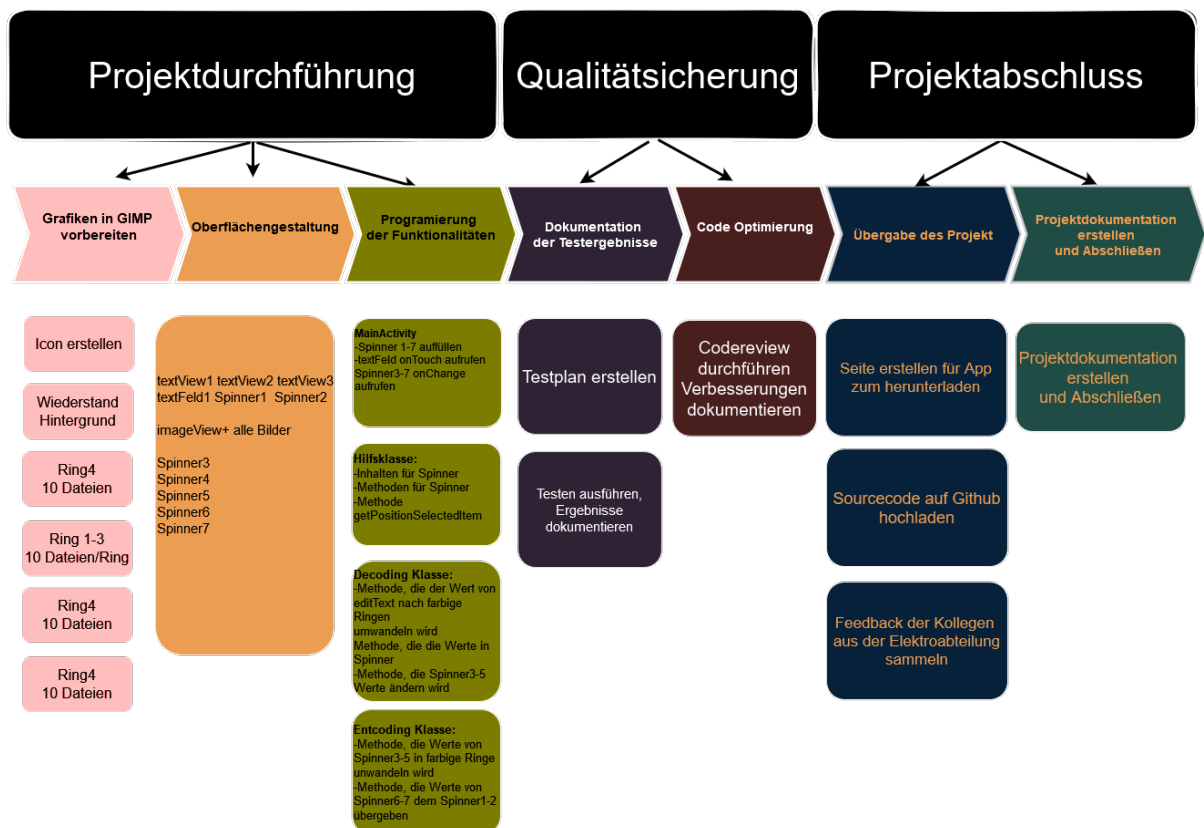


Abbildung 2 Meilensteine

Die Phase hat mich 1 Stunde gekostet.

2.2 Zeitplanerstellung

Ich habe 40 Stunden geteilt:

Projektinitialisierung	
Analyse	0,5 Stunde
Recherche	1,0 Stunde
Projektplanung	
Meilensteine festlegen	1,0 Stunde
Zeitplan erstellen	2,0 Stunden
Projektdurchführung	
Grafiken in GIMP vorbereiten	4,0 Stunden
Oberflächengestaltung	3,5 Stunden

Programmierung der Funktionalitäten: <ul style="list-style-type: none"> ➤ Decoding Klasse <ul style="list-style-type: none"> ○ Aufnehmen des Wertes ○ Ausgeben des Wertes ➤ Encoding Klasse <ul style="list-style-type: none"> ○ Aufnehmen des Wertes ○ Ausgeben des Wertes ➤ Zusätzliche Klasse 	4,0 Stunden 4,0 Stunden 4,0 Stunden 4,0 Stunden 1,0 Stunde
Qualitätssicherung	
Testanforderungen bestimmen	2,0 Stunden
Optimierung	4,0 Stunden
Dokumentation der Testergebnisse	1,0 Stunden
Projektabschluss	
Webseite mit App erstellen	0,5 Stunde
Quellencode auf Githube.com veröffentlichen	0,5 Stunde
Feedback von dem Kolleg von Electroabteilung sammeln	1,0 Stunde
Projektdokumentation erstellen	6,0 Stunden

Tabelle 1 Zeitplanung

Ich habe mich für das V-Model als Projektmethode entschieden.

Die Projektplanung hat mich 2 Stunde gekostet.

3. Projektdurchführung

3.1 Grafiken Vorbereitung in GIMP

Ich habe als erstes eine visuelle Repräsentation des Widerstands vorbereitet und unter `widerstand.png` gespeichert. Auf der Ebene habe ich 5 Ringe in verschiedenen Farben erstellt. Jeder Ring hat von 8 bis 10 Farben. Jeden farbigen Ring habe ich als *.png exportiert, gemäß dem vorher geplanten Muster. Nach kurzer Recherche im Internet habe ich zwei Dateien heruntergeladen und bearbeitet, sodass ich mein Icon als `android_ico192x192.png` speichern konnte. Die Datei hat eine Auflösung von 192x192 Pixeln. Alle oben genannten

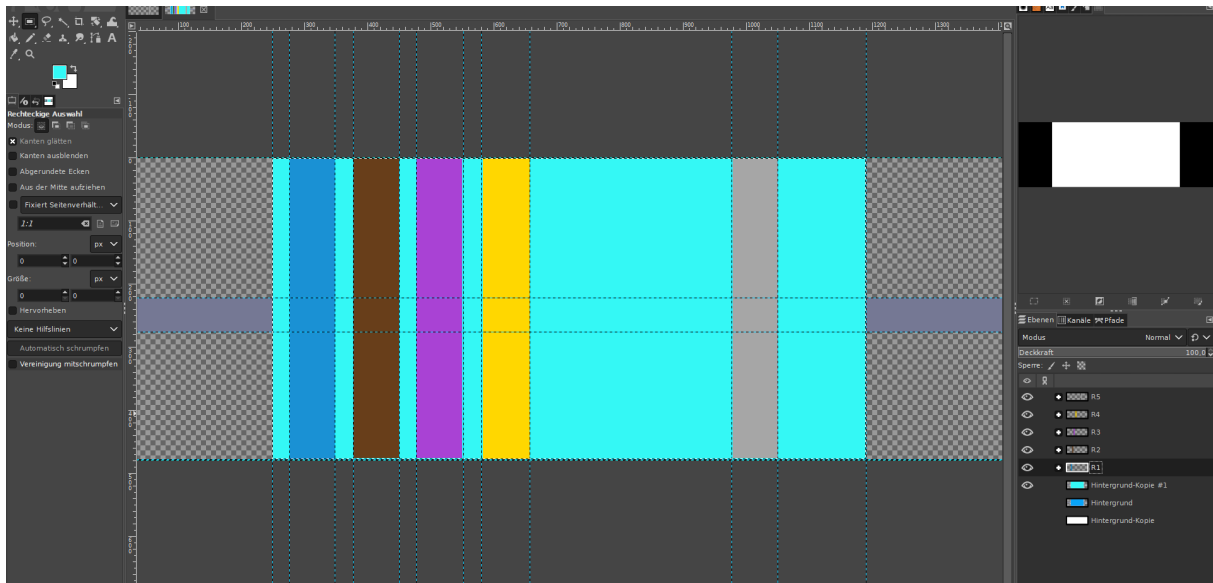


Abbildung 3 Das GIMP Screenshot

Die Grafiken Vorbereitung hat mich 2 Stunden gekostet.

3.2 Oberflächengestaltung

Ich habe das Projekt WDRS für die minimale API (Application Programming Interface) 24, das heißt Android 7 - Nougat, erstellt.

In der Datei `manifest.xml` habe ich die Bildschirmausrichtung auf „portrait“ festgelegt. Zuerst habe ich alle User-Interface-Elemente im Hauptlayout platziert, jedoch musste ich mein ursprüngliches Layout-Projekt ändern, weil Android Studio mir nicht erlaubte, `editTextNumber`, `spinner1` und `spinner` in eine Zeile zu platzieren. Das Element `textView3` musste ich unter `editTextNumber` und das Element `spinner2` unter `textView3` platzieren.

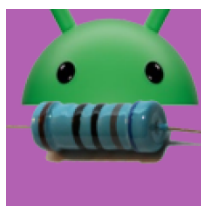


Abbildung 4 Die Ikone der App

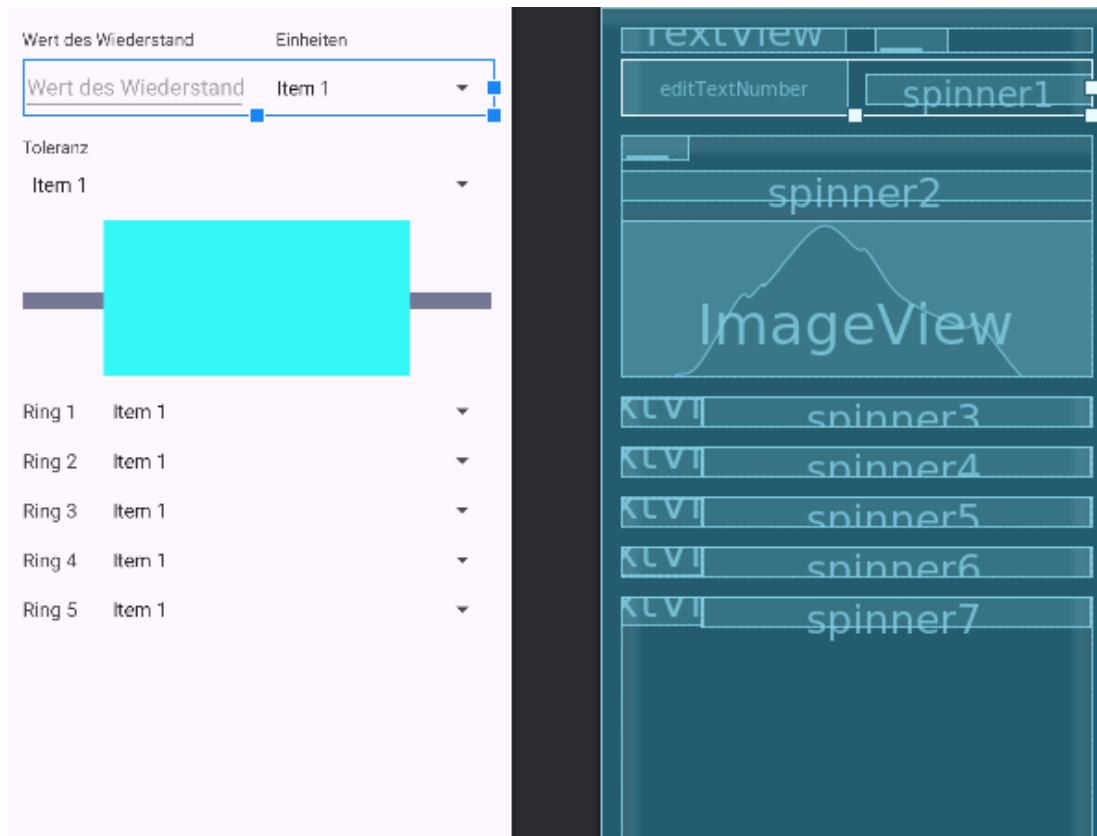


Abbildung 5 Das Layout

Ich habe alle grafischen Elemente in den Ordner `/res/drawable` importiert und diese mit den `ImageViews` verknüpft. Das `ImageView` "widerstand" habe ich auf "visible" eingestellt, alle anderen auf "gone". In der Datei `strings.xml` habe ich acht Strings erstellt, die ich mit dem `textViews` verknüpft habe. Zusätzlich habe ich auch die Ikone mit der App verknüpft.

Die Oberflächengestaltung hat mich 3 Stunden gekostet.

3.3 Hilfsklasse

Diese Klasse, Hilfsklasse, spielt eine unterstützende Rolle in der Anwendung. Ihre Hauptaufgaben umfassen:

1. **Deklaration und Initialisierung:** Die Klasse wird deklariert und initialisiert, um verschiedene Hilfsfunktionen bereitzustellen, die im gesamten Projekt verwendet werden.
2. **Methodenaufrufe:** Die Klasse enthält Methoden wie `setValueOfASpinnerTrueChangeValueOtherSpinner` und `changeColorOfRing`, die von der Hauptaktivität aufgerufen werden, um die Werte der Spinner zu ändern und die Farben der Ringe anzupassen.

3. **Farbänderungen:** Die Methode `changeColorOfRing` wird verwendet, um die Farben der Ringe basierend auf den Spinner-Werten zu ändern, was eine visuelle Rückmeldung für den Benutzer darstellt.
4. **Wertberechnungen:** Die Klasse bietet eine Methode zur Berechnung von Werten, die in verschiedenen Spinnern angezeigt werden sollen, und stellt diese Werte der Hauptaktivität zur Verfügung.
5. **Zustandsverwaltung:** Die Klasse hilft bei der Verwaltung des Zustands der Benutzeroberfläche, indem sie globale Variablen wie `itsInput` verwendet, um bestimmte Funktionen wie Listener vorübergehend zu deaktivieren.

Durch diese Aufgaben trägt die Klasse wesentlich zur Funktionalität und Benutzerfreundlichkeit der Anwendung bei.

Zuerst habe ich eine Hilfsklasse erstellt. In dieser Klasse habe ich die Arrays: `einheiten[]`, `einheitenII[]`, `toleranz[]`, `toleranzII[]`, `ring1[]` und `ring2und3[]` deklariert und initialisiert. Dann habe ich die Methode `setAdapter` erstellt. Diese Methode habe ich in der `MainActivity` für jeden Spinner separat aufgerufen, wodurch die Spinner mit den Arrays gefüllt wurden.

Die nächste Methode, die ich in der Hilfsklasse definiert habe, war `setValueOfSpinnerTrueChangeValueOtherSpinner()`. Diese Methode nimmt die Variable `selectedPosition` vom Eingabe-Spinner und ändert den Wert des Ausgabe-Spinners durch die Änderung der `selectedPosition`.

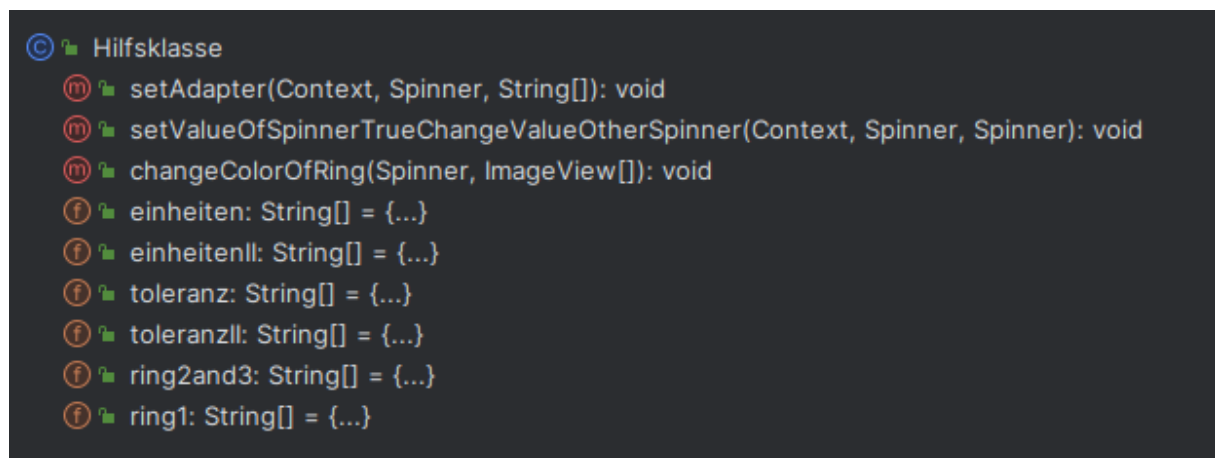


Abbildung 6 Hilfsklasse Struktur

Die Hilfsklasse Erstellung hat mich 1 Stunde gekostet.

3.4 Decoding Klasse

Die Rolle der Decoding-Klasse ist:

- Den Wert aus dem EditText-Feld (number) entnehmen.
- Den Wert in drei Zahlen aufteilen: number1, number2 und number3.
- Aufgrund der Werte von number1, number2 und number3 die Ringe 1, 2 und 3 anzeigen.
- Aufgrund der Daten von Spinner 1 und Spinner 2 die Ringe 4 und 5 anzeigen.
- Die richtigen Werte auf Spinner 3, 4, 5, 6 und 7 einstellen.

Zuerst habe ich ein Hilfsklassen-Objekt definiert. Dann habe ich die Methode `changingValueSpinner6TrueSpinner1()` erstellt. Innerhalb dieser Methode habe ich `setOnItemSelectedListener()` platziert, die über das Hilfsklassen-Objekt die Methode `setValueOfSpinnerTrueChangeValueOtherSpinner` aufruft. Später habe ich eine ähnliche Methode für Spinner 2 und Spinner 7 vorbereitet.

In der MainActivity habe ich ein Decoding-Objekt deklariert und damit beide Methoden separat aufgerufen. Als nächsten Schritt musste ich den Wert aus dem EditText-Feld entnehmen und so bearbeiten, dass ich ihn auf Spinner 3, 4 und 5 verteilen konnte. Zum Beispiel: Der zu decodierende Wert ist 357 Ohm, daher wird Spinner 3 auf 3 gesetzt, Spinner 4 auf 5 und Spinner 5 auf 7. Jetzt muss man das richtige Item entsprechend dem Wert anzeigen.

Ich habe zuerst die Methode `whatIsTheNumber` erstellt. Diese erledigt alle Aufgaben. Danach habe ich in der MainActivity eine globale boolean-Variable `itIsInput` deklariert und mit `true` initialisiert. Diese Variable wird als Flag in weiteren Schritten verwendet. Der nächste Schritt war die Methode `changingValuesOf1to3()`. Die Methode enthält einen `TextChangedListener` und ruft `whatIsTheNumber` auf, wenn `itIsInput` wahr ist. Am Ende habe ich die Methode `changingValuesOf1to3()` in der Klasse MainActivity aufgerufen.

Die letzte Aufgabe, die ich erledigen musste, war das Anzeigen der farbigen Ringe auf einem Widerstandshintergrund. Dafür habe ich in der Klasse Hilfsklasse die öffentliche Methode `changeColorOfRing()` erstellt. Diese Methode ändert die Ringfarbe basierend auf den Werten in den Spinnern. Danach habe ich die Methode in den Methoden `changingValueSpinner6TrueSpinner1()`, `changingValueSpinner7TrueSpinner2()` und fünfmal in der Methode `changingValuesOnSpinners1to3()` aufgerufen.

Die Decoding Klasse Vorbereitung hat mich 8 Stunden gekostet.

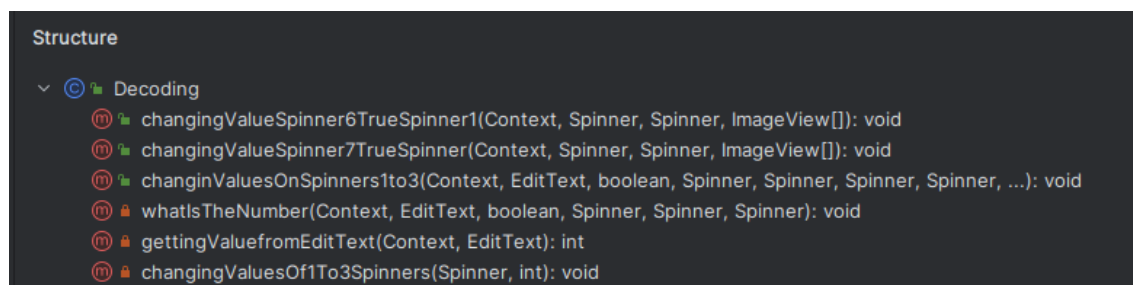


Abbildung 7 Decoding Klasse Struktur

3.5 Encoding Klasse

Die Klasse Encoding ist für die Handhabung von Benutzerinteraktionen und die Aktualisierung von UI-Komponenten in einer Android-Anwendung zuständig. Die wichtigsten Aufgaben und Funktionen dieser Klasse sind:

1. Verwalten der Spinner-Interaktionen:

- Die Methoden `changingValueSpinner1TrueSpinner6` und `ChangingValueSpinner2TrueSpinner7` setzen Listener auf Spinner, die bei einer Änderung der ausgewählten Elemente ausgeführt werden. Diese Methoden rufen Funktionen aus der Hilfsklasse `Hilfsklasse` auf, um die Werte anderer Spinner zu ändern und die Farben von Bildansichten (`ImageViews`) anzupassen.

2. Berechnung und Anzeige eines Wertes in einem EditText-Feld:

- Die Methode `setValueInEditText` setzt Listener auf drei Spinner (`spinner3`, `spinner4`, `spinner5`). Wenn sich die Auswahl in einem dieser Spinner ändert, wird ein numerischer Wert basierend auf den ausgewählten Positionen der Spinner berechnet und in ein EditText-Feld geschrieben. Dies erfolgt nur, wenn ein bestimmtes Flag (`itsInput`) in der `MainActivity` auf `true` gesetzt ist.

3. Hilfsfunktion zur Berechnung des Wertes:

- Die Methode `getValueForEditText` berechnet einen numerischen Wert basierend auf den Positionen der übergebenen Spinner (`spinner3`, `spinner4`, `spinner5`). Der Wert wird aus den einzelnen Positionen der Spinner berechnet und zurückgegeben.

Die Klasse verwendet eine Instanz der Hilfsklasse (`hilfsklasse`), um spezifische Aufgaben auszuführen, wie das Ändern von Spinner-Werten und das Anpassen der Farben von Bildansichten.

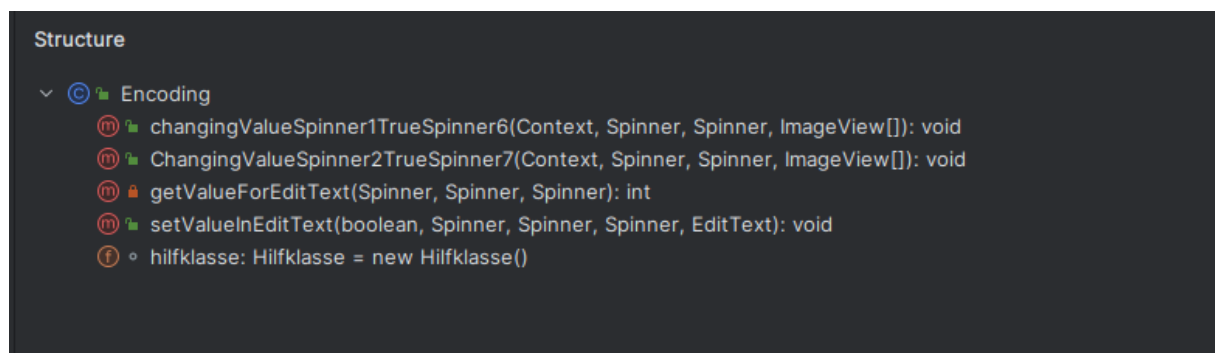


Abbildung 8 Encoding Klasse Struktur

Die Encoding Klasse Vorbereitung hat mich 6 Stunden gekostet.

4. Qualitätssicherung

4.1 Testanforderungen bestimmen

Dieser Testplan definiert die Testanforderungen und Testfälle für die App, um sicherzustellen, dass alle Funktionen korrekt und fehlerfrei arbeiten.

- **Funktionalitätstest:** Überprüfen der Hauptfunktionen der App.
- **Benutzeroberfläche:** Sicherstellen, dass die UI korrekt und benutzerfreundlich ist.
- **Leistungstest:** Sicherstellen, dass die App performant läuft.
- **Kompatibilitätstest:** Überprüfen der Funktionsfähigkeit auf verschiedenen Geräten und Android-Versionen.

Das Testplan Vorbereitung hat mich 1 Stunde gekostet.

4.2 Dokumentation der Testergebnisse

Funktionalität:

- | | |
|---|----|
| 1. Visualisierung der Widerstandskodierung: | Ja |
| 2. Dekodierung des Widerstandswertes: | Ja |

Benutzeroberfläche:

Einfache und schnelle Dateneingabe:	Teilweise
-------------------------------------	-----------

Es ist problematisch, im EditText eine dreistellige Zahl einzugeben, da der Cursor immer wieder an die erste Position springt. Daher wird empfohlen, die Daten über Spinner 1, 2, 3 einzugeben.

Leistungstest:

Die App läuft reibungslos auf allen Geräten mit API 31 und höher.

Kompatibilitätstest:

Die App wurde auf zwei realen Geräten und zwei VM (virtuellen Maschinen) getestet. Auf Geräten mit mindestens API 31 funktionierte die App problemlos. Auf VM mit API 24 kam es jedoch zu einem Crash.

Die Phase hat mich 2 Stunden gekostet.

4.3 Optimierung

Ich habe versucht, die aufgetretenen Probleme mit editText Eingabe zu lösen und den Code insgesamt zu optimieren. Leider konnte ich nur das Problem identifizieren, jedoch nicht innerhalb der vorgegebenen Zeit lösen. Um die Eingabe von Zahlen für EditText komfortabler zu gestalten, habe ich Zahlen zu den Farben in der Liste für die Spinner von 3 bis 5 hinzugefügt. Aufgrund von Zeitmangel konnte ich die Code-Optimierung nicht durchführen.

Die Probe das Problem zu lösen hat mich 7 Stunden gekostet.

5. Projektabschluss

5.1 Übergabe des Projektes

Ich habe eine Webseite erstellt ([hier ist die Seite](#)). Auf dieser Seite habe ich die ausführbare Datei (app-debug.apk), den Sourcecode-Link und die Dokumentation hochgeladen.

Ich habe meinen Kollegen aus der Elektroabteilung informiert, dass ich Feedback benötige. Hier ist sein Feedback:

Die Anwendung erfüllt ihre Funktion, sie ist intuitiv zu bedienen und sieht gut aus. Jedoch sinkt ihre Attraktivität aufgrund des Problems bei der Eingabe des Widerstandswertes.

Das erhaltene Feedback betonte die Wichtigkeit einer reibungslosen Eingabe des Widerstandswertes. Zukünftige Iterationen der App werden sich auf die Behebung dieses Problems konzentrieren und die Benutzererfahrung weiter optimieren.

Die Phase hat mich 2 Stunden gekostet.

5.2 Projektdokumentation erstellen

Als letzten Schritt habe ich die oben genannte Dokumentation erstellt.

Die Phase hat mich 6 Stunden gekostet.

6. Benutzeranleitung

Installation:

- Laden Sie die Datei app-debug.apk von der [Projektwebseite](#) herunter.
- Installieren Sie die APK-Datei auf Ihrem Android-Gerät. Das Gerät muss im Entwicklermodus sein.

Verwendung:

- Starten Sie die App.
 - Geben Sie den Wert in das Textfeld ein.
 - Lesen Sie das Ergebnis aus dem Bild oder dem Spinner unter dem Bild ab.
- Oder:
 - Geben Sie die Werte in die Spinner ein.
 - Lesen Sie das Ergebnis aus dem Textfeld ab.

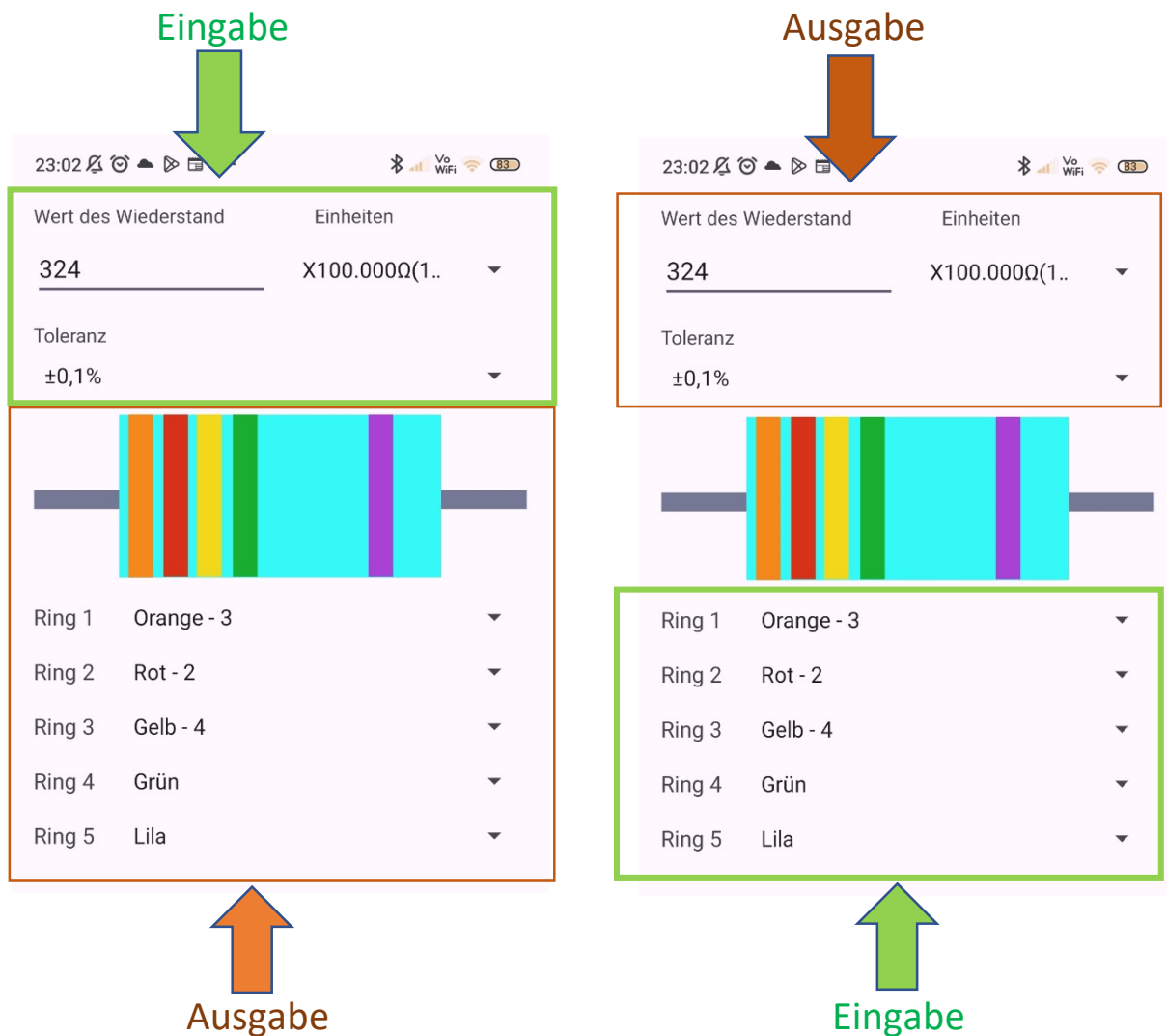


Abbildung 9 Benutzeranleitung

7. Abkürzungserklärung

API (Application Programming Interface): Ein Satz von Routinen, Protokollen und Tools für den Aufbau von Softwareanwendungen. Eine API definiert, wie Softwarekomponenten miteinander interagieren.

MVC steht für **(Model-View-Controller)** und ist ein Architekturpattern, das häufig in der Softwareentwicklung verwendet wird, um die Struktur von Anwendungen zu organisieren. Es teilt eine Anwendung in drei Hauptkomponenten, um die Trennung von Geschäftslogik, Benutzeroberfläche und Eingabe zu ermöglichen.

UI (User Interface): Benutzeroberfläche. Die Schnittstelle, über die ein Benutzer mit einem Computer, einer Softwareanwendung oder einem Gerät interagiert.

VM (Virtuelle Maschine): Eine Software, die die Funktionen eines physischen Computers imitiert und es ermöglicht, mehrere Betriebssysteme auf einem einzigen physischen Rechner zu betreiben.

8. Abbildung und Tabelle Verzeichnis

Abbildung 1 Use Case Diagramm	1
Abbildung 2 Meilensteine	5
Abbildung 3 Das GIMP Screenshot.....	7
Abbildung 4 Die Ikone der App.....	7
Abbildung 5 Das Layout.....	8
Abbildung 6 Hilfsklasse Struktur	9
Abbildung 7 Decoding Klasse Struktur	10
Abbildung 8 Encoding Klasse Struktur.....	11
Abbildung 9 Benutzeranleitung.....	15

9. Tabelle Verzeichnis

Tabelle 1 Zeitplanung	6
-----------------------------	---