Bericht – GestPlayer

Erstellt von:

|  |  |  |
| --- | --- | --- |
| Tristan Kley | 3005277 | Tristan.Kley@stud.uni-duisburg-essen.de |
| Sven Vaupel | 2268341 | Sven.Vaupel@stud.uni-duisburg-essen.de |
| Julian Föckler | 3013082 | julian.foeckler@gmail.com |

Inhaltsverzeichnis

[Konzeption 4](#_Toc488416247)

[UI-Skizzen / Storyboards 4](#_Toc488416248)

[Interaktionskonzept 7](#_Toc488416249)

[Architektur 9](#_Toc488416250)

[UML Komponentendiagramm 9](#_Toc488416251)

[Architekturübersicht (natürlichspracheliche Beschreibung des Komponentendiagramms) 9](#_Toc488416252)

[Entscheidungsfindung für Architekturkomponenten (Komponentendiagramm mit Quellcode vernküfen) 9](#_Toc488416253)

[Implementierung 9](#_Toc488416254)

[UML Klassendiagramm 9](#_Toc488416255)

[Testen 9](#_Toc488416256)

[Testplan 9](#_Toc488416257)

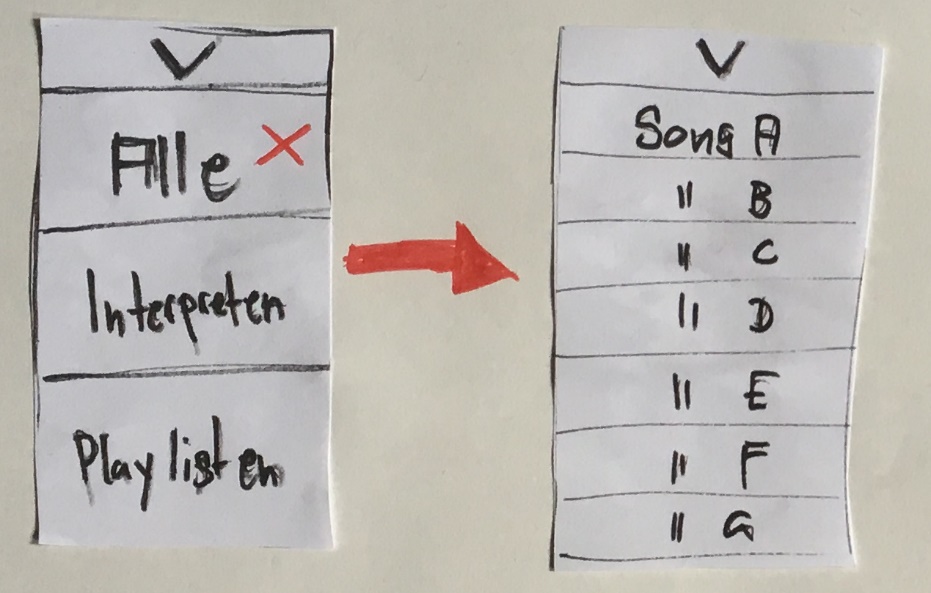
[Dokumentation des Testablaufs / Testergebnisse 9](#_Toc488416258)

# Konzeption

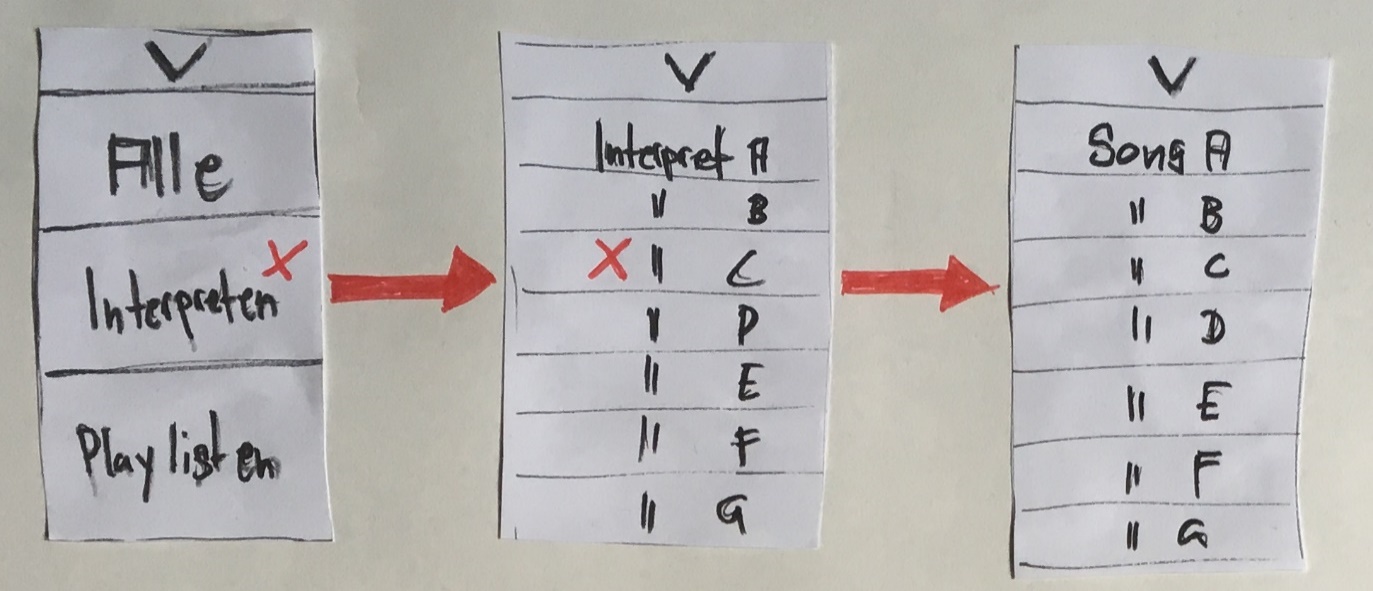
## UI-Skizzen / Storyboards

Im Folgenden zeigen die sequenziellen Storyboards die Umsetzung einer vordefinierten Userstory. Die UI-Elemente eines Storyboards machen durch Annotationen deutlich, wodurch eine Transition von einem UI-Element zu dem nächsten ausgelöst wird.

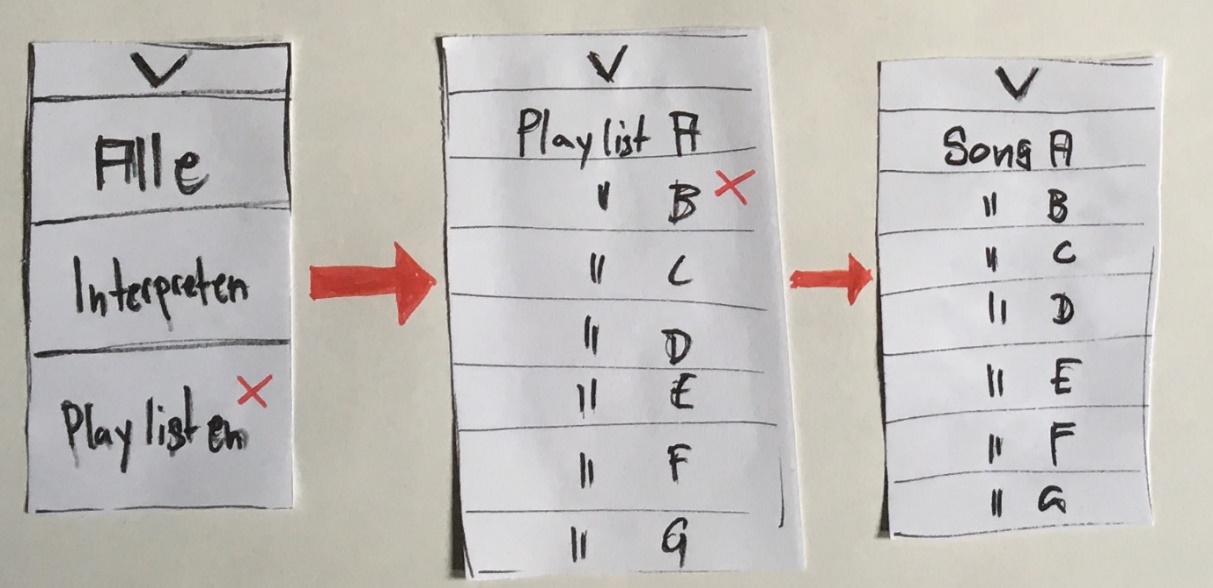
01: Nutzer wählt alle Songs



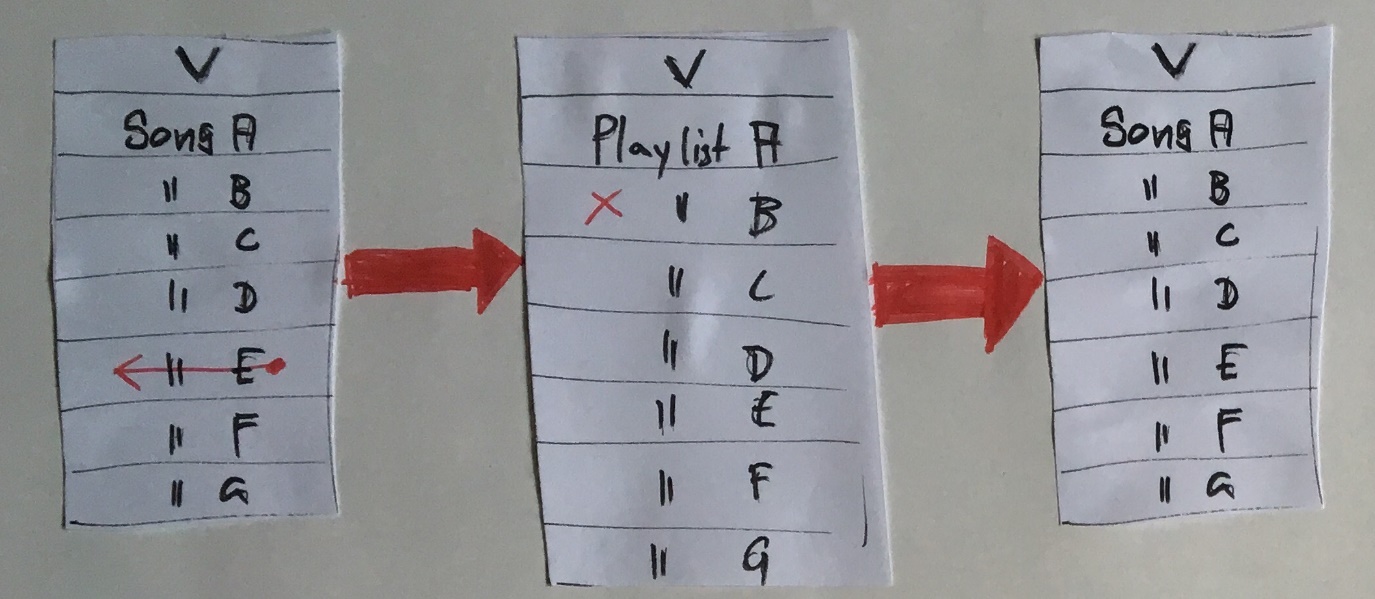
02: Nutzer wählt alle Interpreten



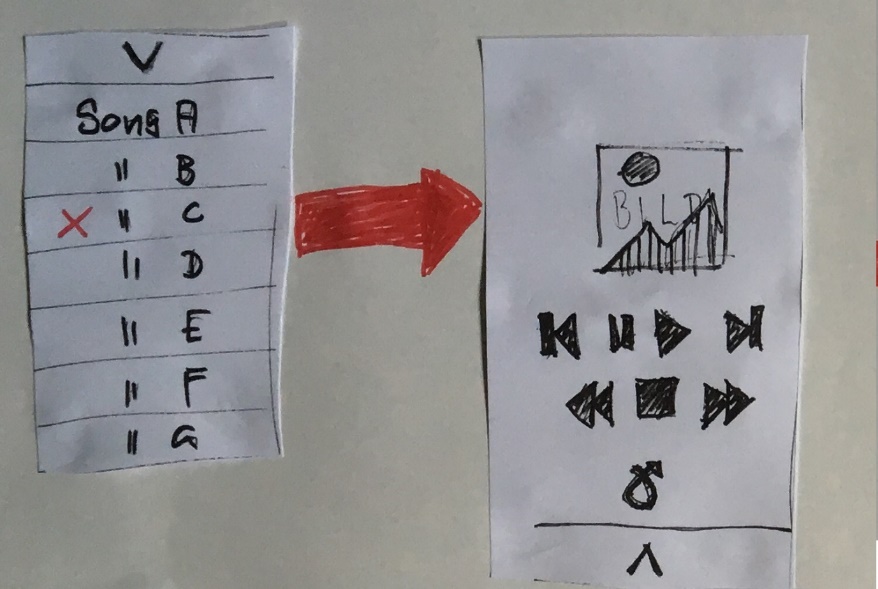
03: Nutzer wählt alle Playlists

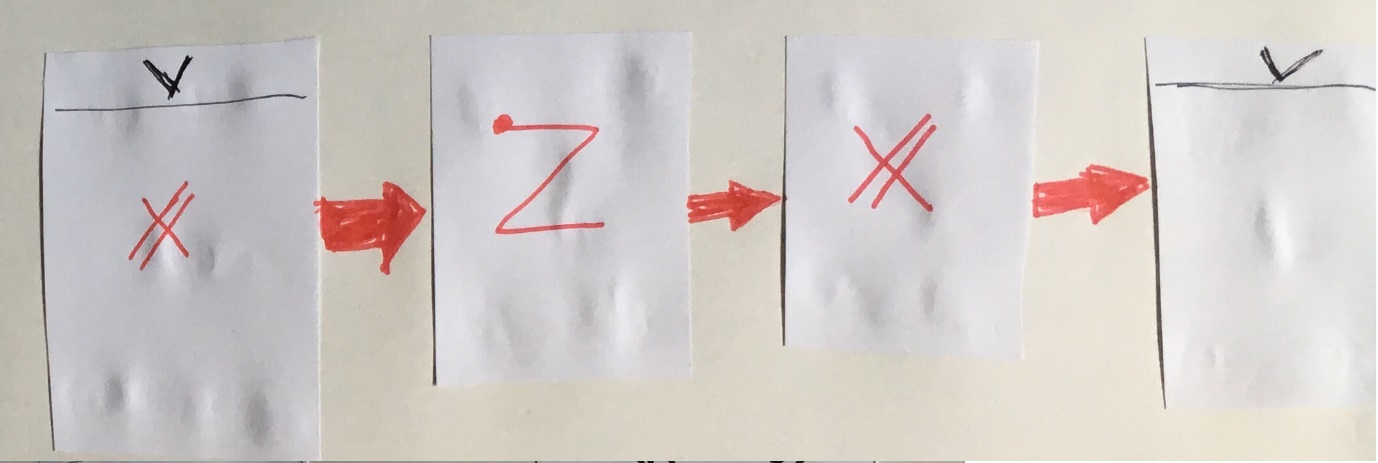


04: Nutzer fügt Song einer Playlist hinzu

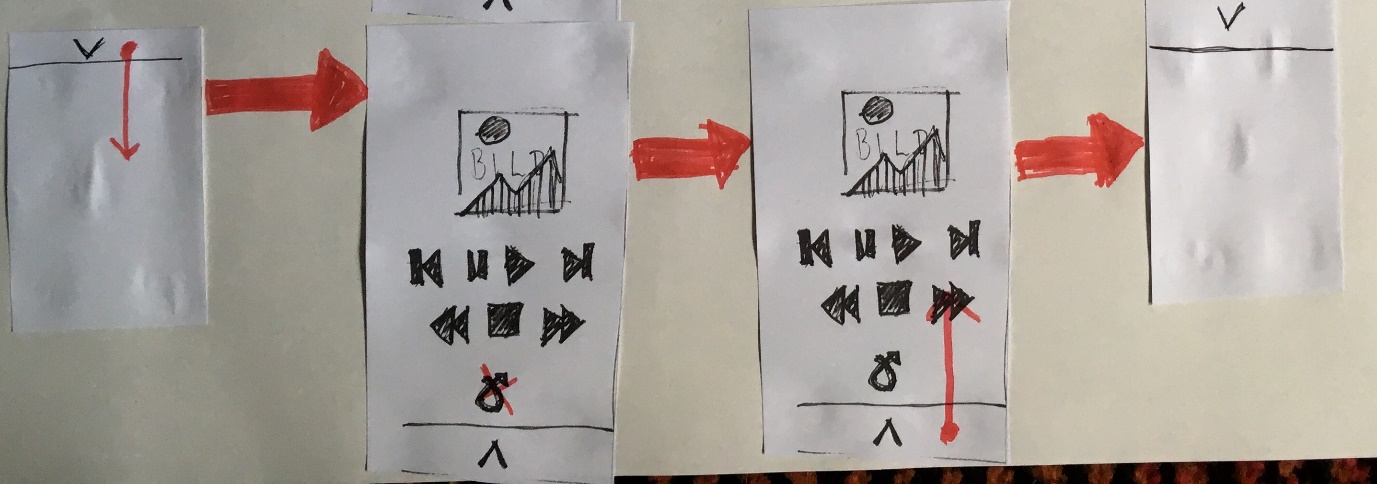


05: Nutzer startet die Wiedergabe von Songs



06: Nutzung des Gesteninterpreters

07: Nutzung des Wiedergabescreens



## Interaktionskonzept

Die Anwendung implementiert einen Gestengesteuerten mobilen MP3 Player. Bei der Entwicklung wurde eine möglichst intuitive Bedienung der Anwendung angestrebt. Sämtliche Übergänge zwischen den einzelnen Screens können ebenfalls dem angehangenen Schaubild I entnommen werden. Die Bedienung wurde wie folgt umgesetzt.

**Screen I**: Dieser Screen ist der Startbildschirm der Anwendung, von wo aus man Zugriff auf sämtliche Songs, Interpreten und Playlists, welche sich auf dem Gerät befinden, hat. Außerdem erreicht von hier, wie von jedem anderen Screen auch, per Top-Down-Swipe, Screen II.

**Screen II:** Dieser Screen besitzt die Hauptschaltflächen zur Bedienung von Songs und Playlists. Auf diesem Screen sieht man den aktuellen Songtitel, das zugehörige Albumcover, wenn vorhanden, und Knöpfe um zum vorherigen Song zu gelangen, den Song zu überspringen, den Song zu pausieren, die Playlist zu stoppen, die Songlautstärke anzupassen, im Song nach vorne und hinten zu Spulen und einen Knopf, welcher sämtliche Songs in der Playlist zufällig wiedergeben lässt. Wenn der Benutzer auf diesem Screen von unten nach oben swipet, minimiert sich die Ansicht, und der Benutzer gelangt zurück zum vorherigen Screen.

**Screen III:** Dieser Screen zeigt dem Benutzer sämtliche Playlists an, welche sich auf seinem Gerät befinden. Auf diesen Screen gelangt man, indem man auf Screen I den Playlistbutton drückt, oder indem man in der Songauswahl einen Song einer Playlist hinzufügen möchte. Des Weiteren wird dem Benutzer auf diesem Screen per Klick auf einen „+“ Button die Möglichkeit geboten, eine neue Playlist anzulegen, oder per Klick auf eine Playlist, die komplette Playlist abzuspielen. In diesem Fall gelangt der Benutzer wieder auf Screen II.

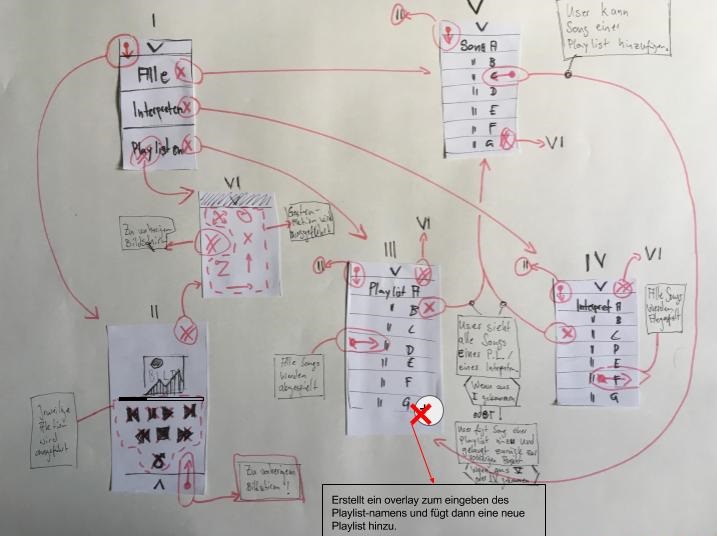
**Screen IV**: Dieser Screen zeigt eine Auflistung sämtlicher sich auf dem Gerät Interpreten. Auf diesen Screen gelangt der Benutzer, indem er von Screen I aus, auf „Interpreten“ klickt. Wählt der Benutzer einen Interpreten auf diesem Screen aus, werden alle Songs des dazugehörigen Interpreten abgespielt und man gelangt zu Screen II.

**Screen V**: Dieser Screen zeigt dem Benutzer sämtliche sich auf dem Gerät befindlichen Songs an. Auf diesen Screen gelangt der Benutzer, indem er auf Screen I auf „Alle“ klickt. Der Benutzer kann auf diesem Screen einen Swipe auf einem Song ausführen, um auf Screen III zu gelangen, wo er diesen Song wiederum einer Playlist hinzufügen kann. Klickt der Benutzer einen Song auf diesem Screen an, wird dieser Song abgespielt und man gelangt zu Screen II.

**Screen VI**: Dieser Screen setzt die Gestensteuerung der Anwendung um. Man gelangt von allen Screens per Doppelklick an eine beliebige Stelle aus darauf. Es öffnet sich eine weiße Seite, auf der folgende Gestern ausgeführt werden können, um die Wiedergabe einen Song / einer Playlist zu beeinflussen.

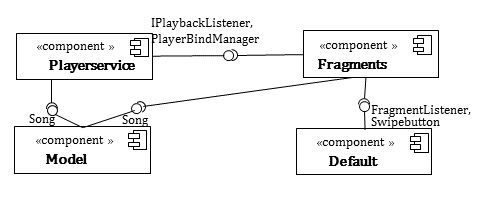
* Ein (stilisiertes) X, setzt aktiviert die zufällige Wiedergabe der Playlist
* Ein (stilisierter) Kreis in
* Usw

Mit einem Doppelklick an eine beliebe Stelle dieses Screens gelangt der Benutzer zu dem vorherigen Screen zurück.

Schaubild I: vollständiges Interaktionskonzept

# Architektur

## UML Komponentendiagramm



## Architekturübersicht (natürlichsprachliche Beschreibung des Komponentendiagramms)

Das Komponentendiagramm stell die vier Komponenten der Applikation und ihre Interaktionen untereinander dar.

**Playerservice**: Diese Komponente bietet zu einem das Interface IPlaybackListener und zum anderen die Klasse PlayerBindManger an. IPlaybackListener erlaubt es der implementierenden Klasse über Änderungen in Songwiedergabe benachrichtigt zu werden. PlayerBindManager erlaubt es der nachfragenden Klasse mit dem Playerservice zu kommunizieren.

**Model**: Diese Komponente bietet die Klasse Song an. Instanzen dieser Klasse repräsentieren ein abzuspielendes Lied.

**Fragments**: In dieser Komponente werden die einzelnen Fragments instanziiert.

**Default**: Diese Komponente bietet die Klasse FragmentListener und Swipebutton an. Hier liegt die Activity, welche die einzelnen Fragments austauscht. Fragmentlistener erlaubt es den Fragments auf die Activity zuzugreifen. Swipebutton …

## Entscheidungsfindung für Architekturkomponenten (Komponentendiagramm mit Quellcode verknüpfen?)

Die Entscheidungsfindung für die Architekturkomponenten war vielmehr ein emergenter Prozess, anstelle eines von Beginn an bestehenden Architekturplanes, welcher sich im Laufe der Implementierung ergab. Einzig die Tatsache, dass wir nur eine einzige Activity benutzen, welche die Fragments wechselt, war geplant. Die restlichen Komponenten ergaben sich aus den entstehenden Problemen bei der Implementierung und den daraus resultierenden Lösungsansätzen.

# Implementierung

## UML Klassendiagramm (Tool: SimpleUML für Android Studios)

# Testen

## Testplan

Der Testplan der Applikation weißt einige Besonderheiten auf. Bei der Softwareentwicklung ist es generell von großem Wert, die ursprünglichen Requirements im Entwicklungsprozess aktuell zu halten und etwaige Veränderungen zu dokumentieren. Da jedoch zu Beginn sämtliche Requirements vorgegeben waren, und es keine Änderungen an den Requirements im Projektverlauf gab, können sich die Tests ausschließlich auf die eigentliche Software konzentrieren. Um die Software auf Korrektheit zu überprüfen, bedienen wir uns der Methodik der Modultests. Mit diesen Tests testen wir isolierte Kernfunktionen der Applikation auf ihre Korrektheit. Diese Kernfunktionen leiten wir direkt aus den vorgegebenen Userstories ab.

Folgende Tests sind durchzuführen:

1. Alle MP3’s auf dem Gerät sollen in der App angezeigt werden
2. Playlists erstellen und Songs hinzufügen
3. Sämtliche Funktionen während der Wiedergabe eines Songs
4. Songs nach Interpreten sortieren
5. Wiedergabe aller Songs eines Interpreten / einer Playlist

## Dokumentation des Testablaufs / Testergebnisse

1.

|  |  |
| --- | --- |
| Testziel | Alle MP3’s auf dem Gerät angezeigt |
| Schnittstelle/Klasse |  |
| Vorbedingung |  |
| Nachbedingung |  |
| Bestehenskriterium | Alle MP3’s auf dem gerät wurden erkannt und korrekt angezeigt. |
| Testurteil | Bestanden: Alle MP3’s auf dem gerät wurden erkannt und korrekt angezeigt. |

2.

|  |  |
| --- | --- |
| Testziel | Playlists erstellen und Songs hinzufügen |
| Schnittstelle/Klasse |  |
| Vorbedingung |  |
| Nachbedingung |  |
| Bestehenskriterium | Playlists konnten erstellt werden uns Songs hinzugefügt werden. |
| Testurteil | Bestanden: Playlists konnten erstellt werden uns Songs hinzugefügt werden. |

3.

|  |  |
| --- | --- |
| Testziel | Funktionen während der Songwiedergabe |
| Schnittstelle/Klasse |  |
| Vorbedingung |  |
| Nachbedingung |  |
| Bestehenskriterium | Sämtliche Funktionen während der Songwiedergabe funktionierten |
| Testurteil | Bestanden: Sämtliche Funktionen während der Songwiedergabe funktionierten |

4.

|  |  |
| --- | --- |
| Testziel | Songs nach Interpreten sortieren |
| Schnittstelle/Klasse |  |
| Vorbedingung |  |
| Nachbedingung |  |
| Bestehenskriterium | Songs konnten nach Interpreten sortiert werden. |
| Testurteil | Bestanden: Songs konnten nach Interpreten sortiert werden. |

5.

|  |  |
| --- | --- |
| Testziel | Wiedergabe aller Songs eines Interpreten / einer Playlists |
| Schnittstelle/Klasse |  |
| Vorbedingung |  |
| Nachbedingung |  |
| Bestehenskriterium | Es konnten alle Songs eines Interpreten bzw. einer Playlist wiedergegeben werden. |
| Testurteil | Bestanden: Es konnten alle Songs eines Interpreten bzw. einer Playlist wiedergegeben werden. |

Abschließend lässt sich festhalten, dass alle Test bestanden wurden und somit die Applikation den gewünschten Anforderungen entspricht.