

Bericht – GestPlayer

Erstellt von:

Tristan Kley	3005277	Tris- tan.Kley@stud.uni- duisburg-essen.de
Sven Vaupel	2268341	Sven.Vau- pel@stud.uni-duis- burg-essen.de
Julian Föckler	3013082	julian.foeck- ler@gmail.com

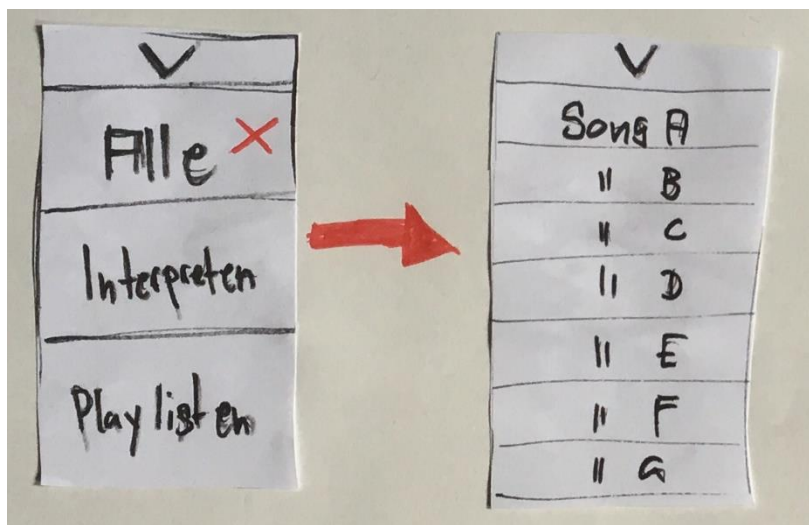
Inhaltsverzeichnis

1 Konzeption.....	4
1.1 UI-Skizzen / Storyboards.....	4
1.2 Interaktionskonzept	7
2 Architektur	10
2.1 UML Komponentendiagramm.....	10
2.2 Architekturübersicht.....	10
2.3 Entscheidungsfindung für Architekturkomponenten	11
3 Implementierung	12
3.1 UML Klassendiagramm	12
3.2 Screenshots der App als Storyboard.....	12
4 Testen.....	17
4.1 Testplan	17
4.2 Dokumentation des Testablaufs / Testergebnisse.....	17

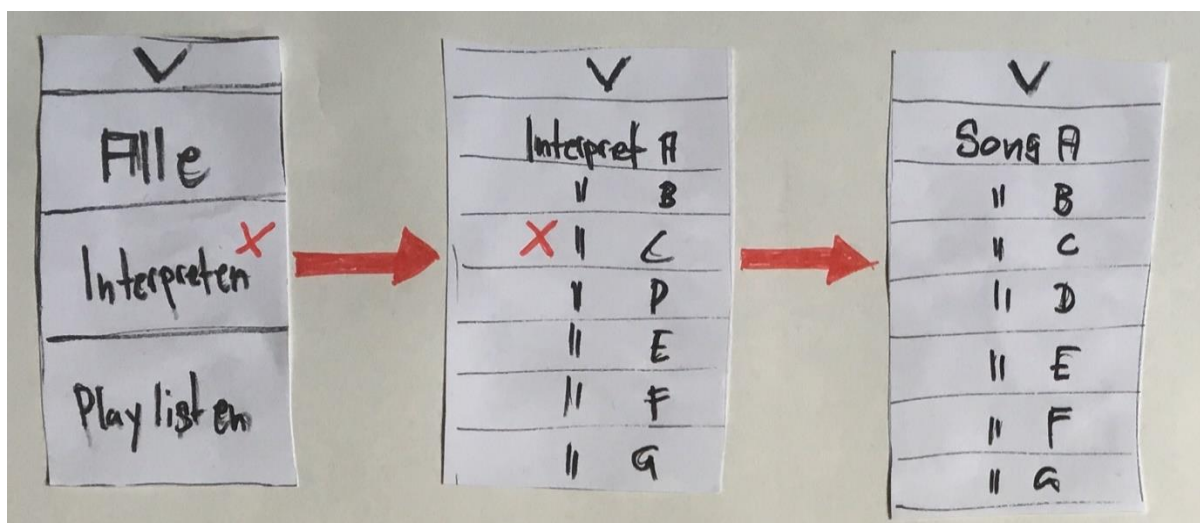
1 Konzeption

1.1 UI-Skizzen / Storyboards

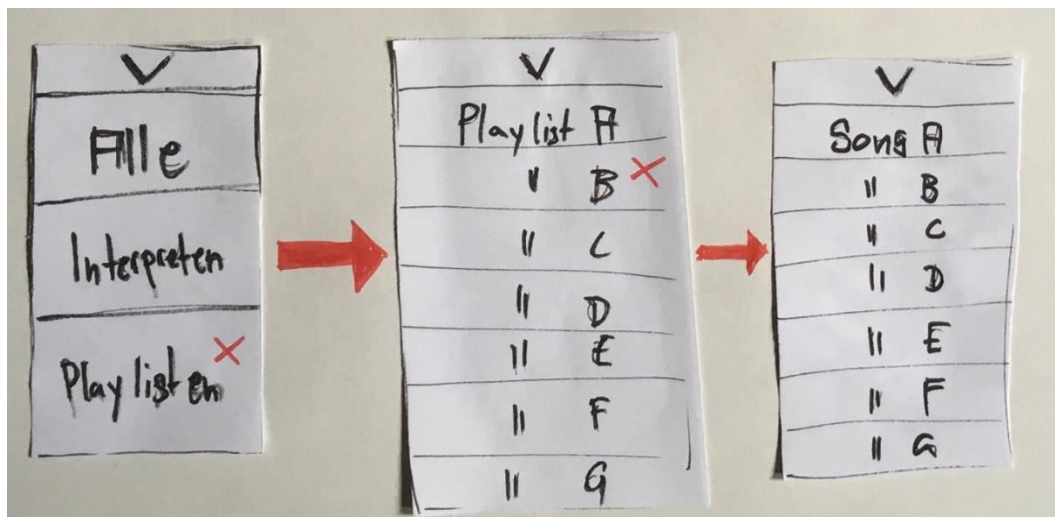
Im Folgenden zeigen die sequenziellen Storyboards die Umsetzung je einer vordefinierten Userstory. Die UI-Elemente eines Storyboards machen durch Annotationen deutlich, wodurch eine Transition von einem UI-Element zu dem nächsten ausgelöst wird.



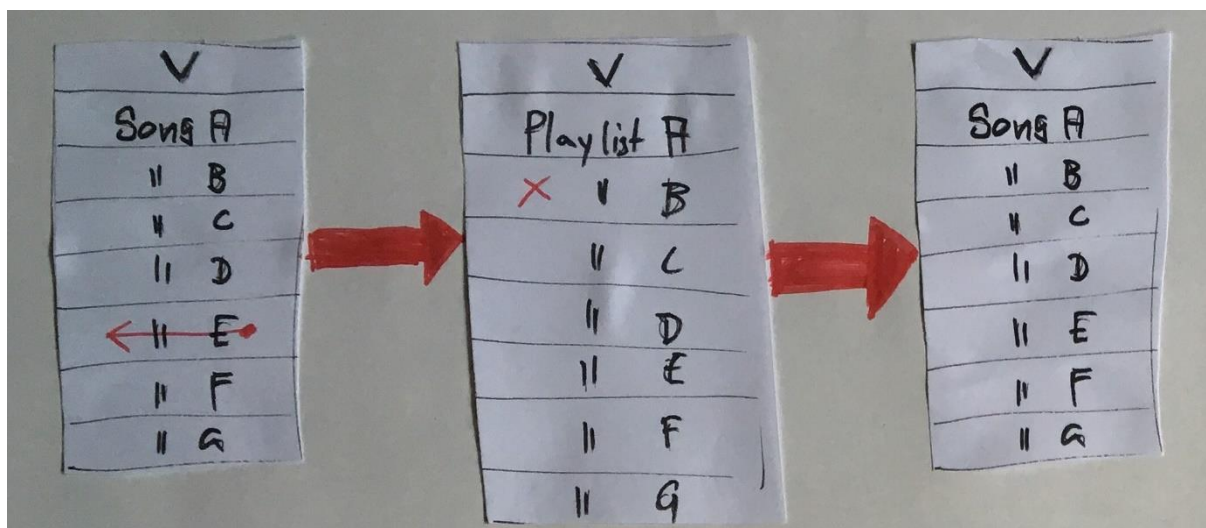
01: Nutzer wählt alle Songs



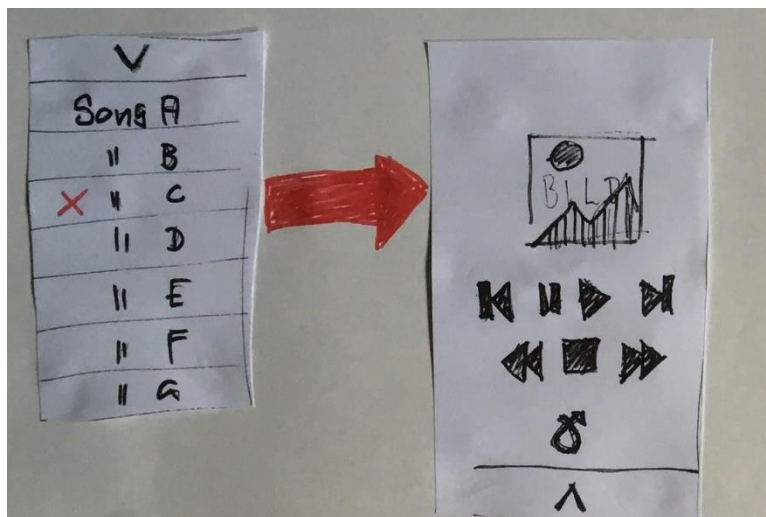
02: Nutzer wählt alle Interpreten



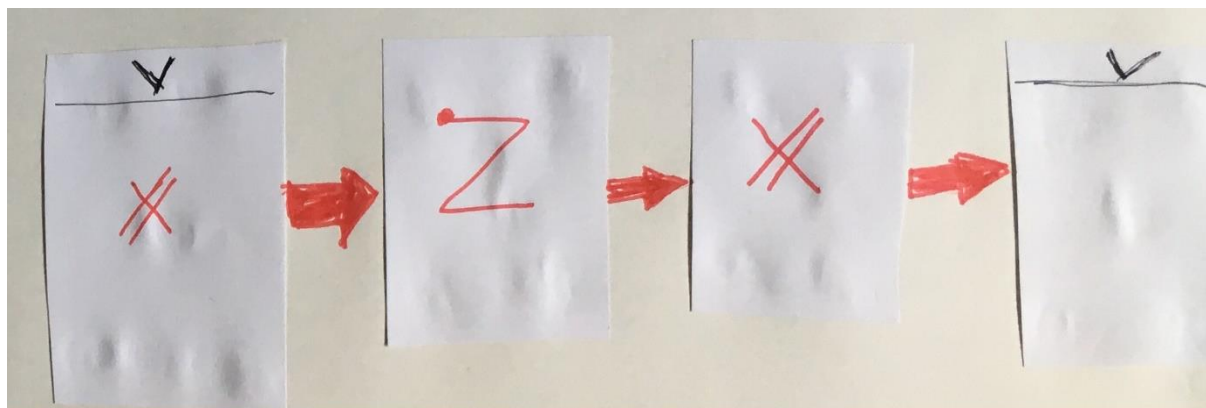
03: Nutzer wählt alle Playlists



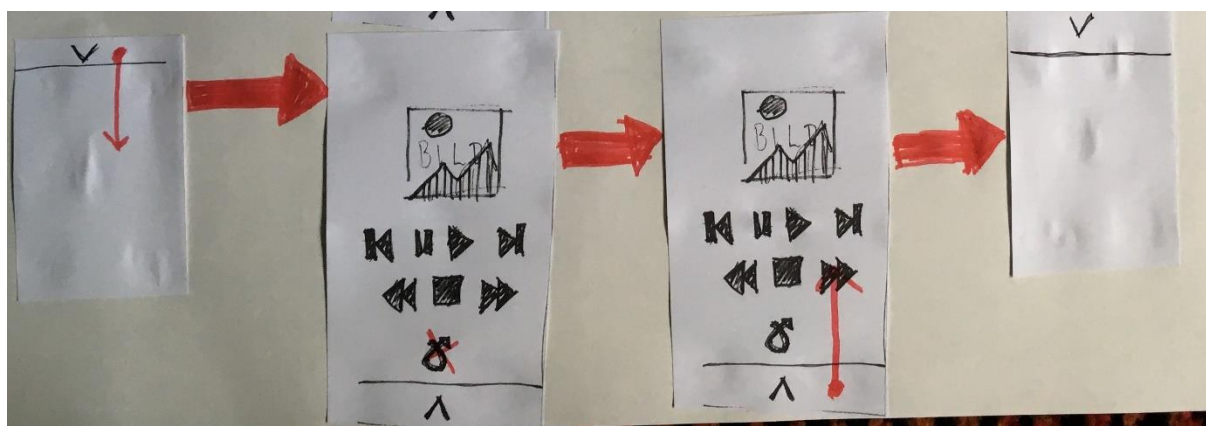
04: Nutzer fügt Song einer Playlist hinzu



05: Nutzer startet die Wiedergabe von Songs



06: Nutzung des Gesteninterpreters



07: Nutzung des Wiedergabescreens

1.2 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 man von hier aus, 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) Buttons 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 befindlicher 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 Longpress an eine beliebige Stelle aus darauf. Es öffnet sich eine weiße Seite, auf der folgende Gestern ausgeführt werden können, um die Wiedergabe eines Song / einer Playlist zu beeinflussen.

- Ein X, aktiviert die zufällige Wiedergabe der Playlist
- Ein Z, aktiviert die Shuffle Funktionen
- Ein Kreis, im/gegen den Uhrzeigersinn spult das Lied 10 Sekunden vor/zurück
- Ein Singletap wechselt zwischen Wiedergabe und Pausierung des Songs
- Ein Fling up/down erhöht/verringert die Lautstärke
- Ein Swip links/rechts spring zum vorherigen/nächsten Song
- Ein Longtap beendet die Gestensteuerung und bringt den Benutzer zum zuletzt benutzten Screen zurück

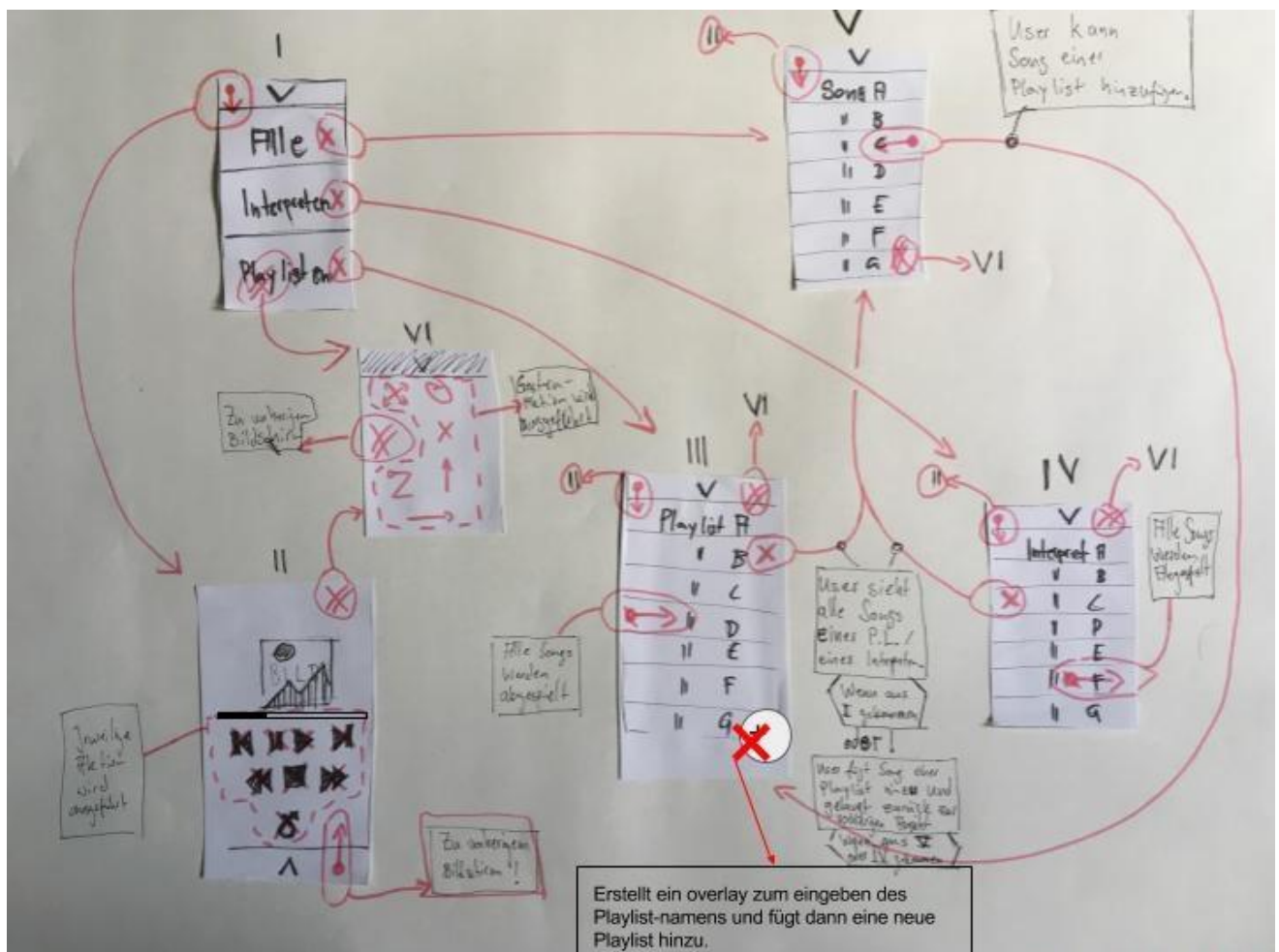
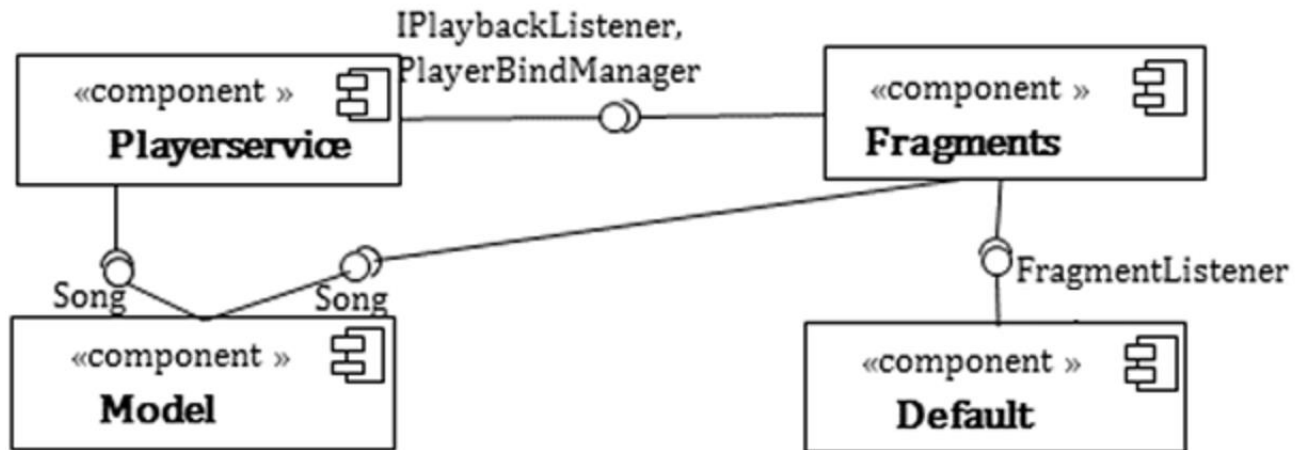


Schaubild I: vollständiges Interaktionskonzept

2 Architektur

2.1 UML Komponentendiagramm



2.2 Architekturübersicht

Das Komponentendiagramm stellt 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` an. Hier liegt die Activity, welche die einzelnen `Fragments` austauscht. `FragmentListener` erlaubt es den einzelnen `Fragments` auf die Activity zuzugreifen.

2.3 Entscheidungsfindung für Architekturkomponenten

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.

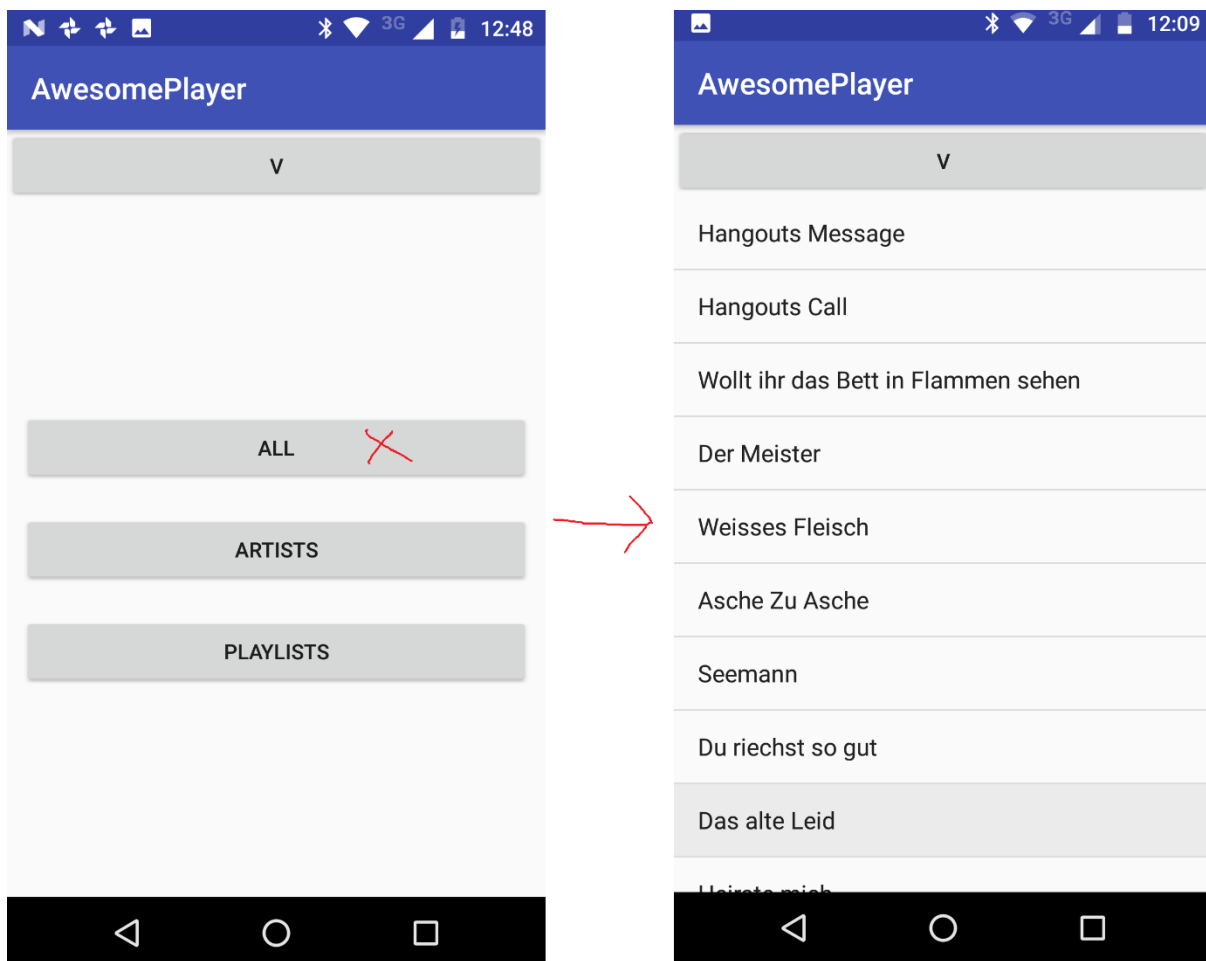
3 Implementierung

3.1 UML Klassendiagramm

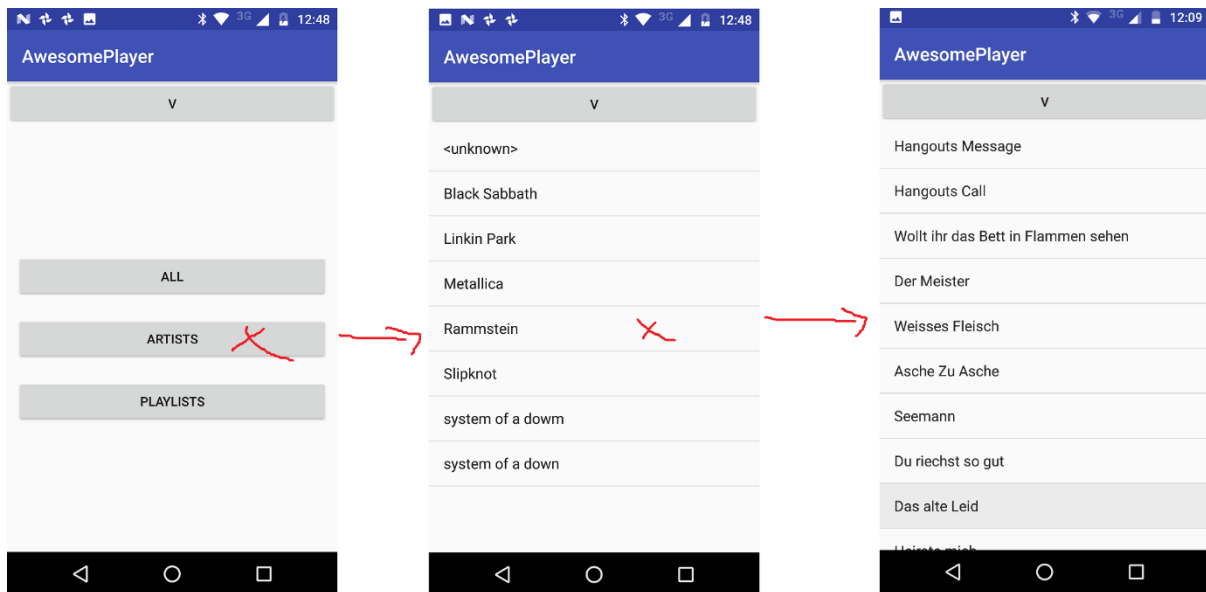
Das UML Klassendiagramm befindet sich aufgrund von mangelnder Lesbarkeit in dem eingereichten ZIP-Order namens „MSE Class Diagramm.svg“.

3.2 Screenshots der App als Storyboard

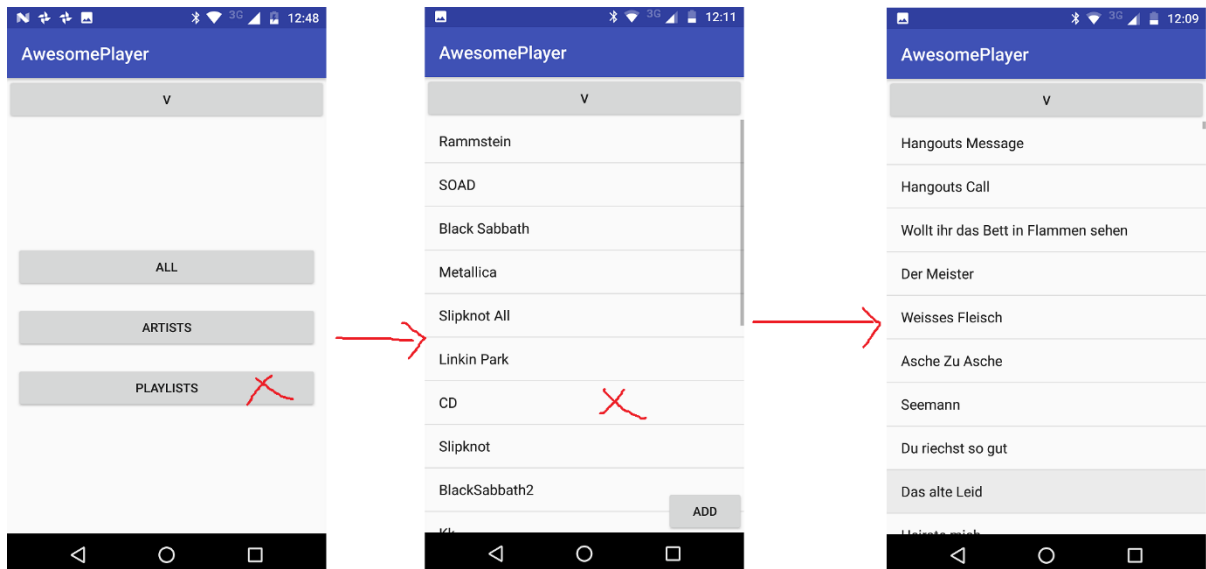
Im Folgenden befinden sich nun konkrete Screenshots der Applikation, wie sie schon im Papierprototypen geplant wurden.



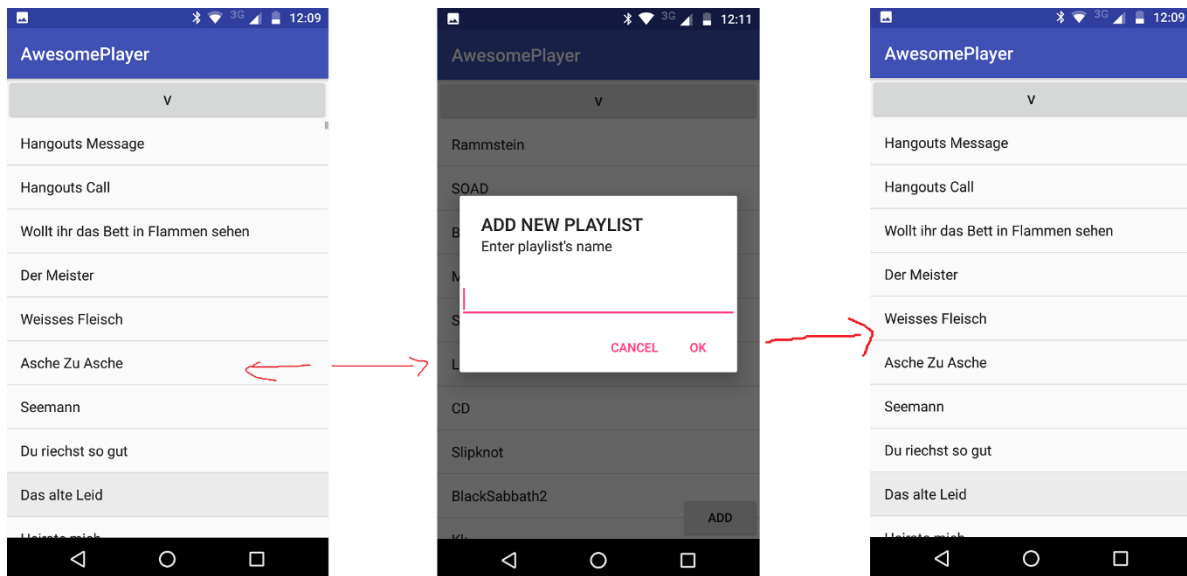
01: Nutzer wählt alle Songs



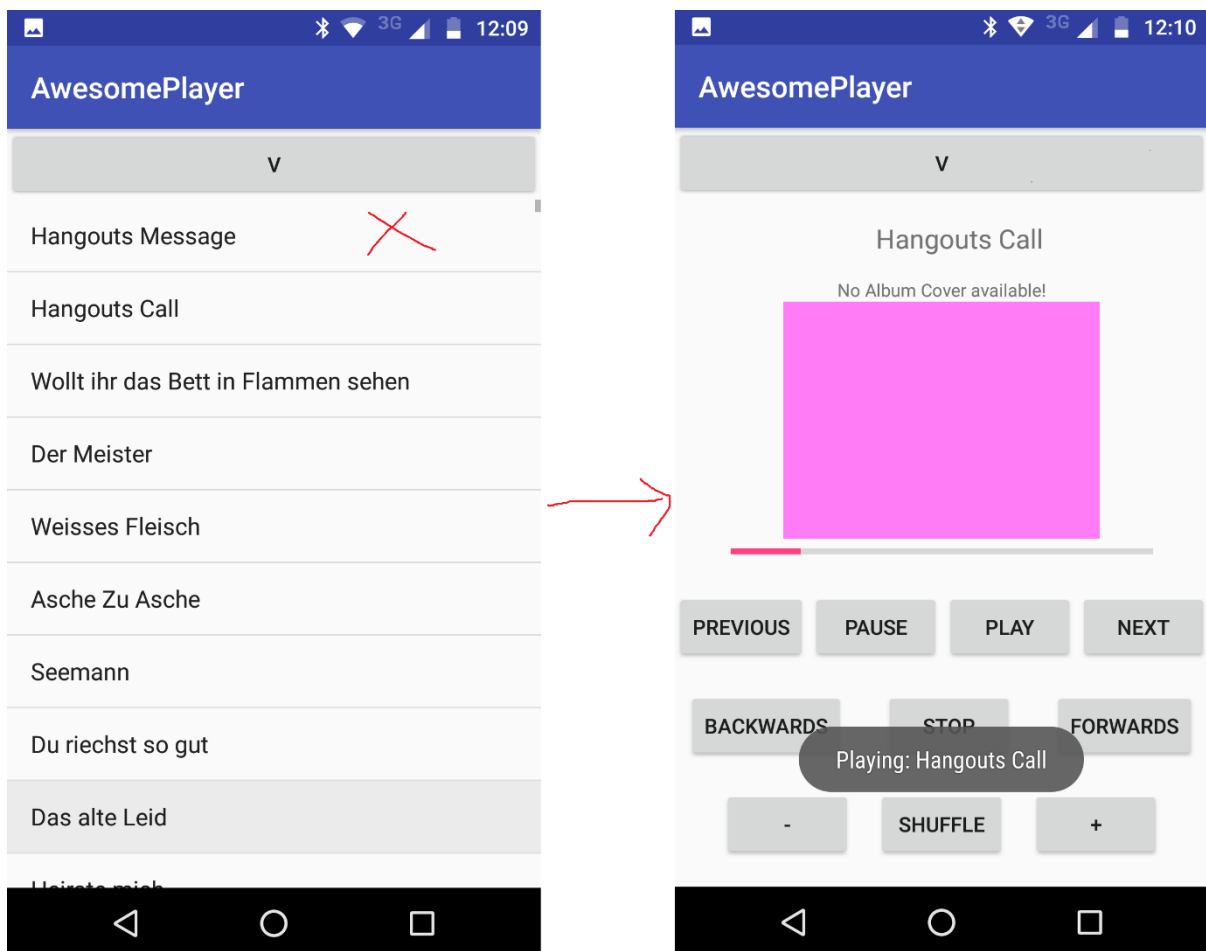
02: Nutzer wählt alle Interpreten



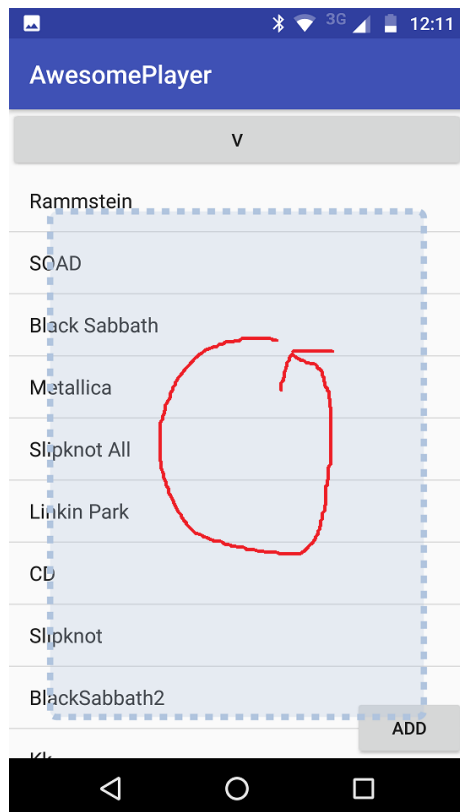
03: Nutzer wählt alle Playlists



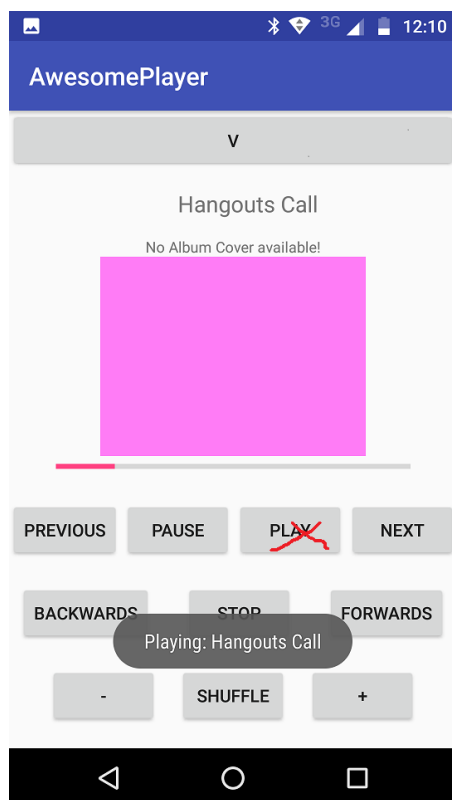
04: Nutzer fügt Song einer Playlist hinzu



05: Nutzer startet die Wiedergabe von Songs



06: Nutzung des Gesteninterpreters



07: Nutzung des Wiedergabescreens

4 Testen

4.1 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. Mit diesen Tests testen wir die 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

4.2 Dokumentation des Testablaufs / Testergebnisse

Testziel	Alle MP3's auf dem Gerät werden angezeigt
Schnittstelle/Klasse	MedialistFragment
Vorbedingung	MP3's auf dem Gerät verfügbar
Nachbedingung	MP3's wurden erkannt
Bestehenskriterium	Alle MP3's auf dem gerät werden erkannt und korrekt angezeigt.

Testurteil	Bestanden: Alle MP3's auf dem Gerät wurden erkannt und korrekt angezeigt.
------------	---

1.

Testziel	Playlists erstellen und Songs hinzufügen
Schnittstelle/Klasse	MedialistFragment, PlaylistFragment
Vorbedingung	Song/Playlistobjekte wurden erstellt
Nachbedingung	Songobjekte können Playlistobjekten hinzugefügt werden
Bestehenskriterium	Playlists kann erstellt werden und Songs hinzugefügt werden.
Testurteil	Bestanden: Playlists konnten erstellt werden und Songs hinzugefügt werden.

2.

Testziel	Funktionen während der Songwiedergabe
Schnittstelle/Klasse	PlayerFragment, GestureCanvasFragment
Vorbedingung	Songwiedergabe wurde gestartet

Nachbedingung	Songs reagieren auf Gesten/Knopfdruck
Bestehenskriterium	Sämtliche Funktionen während der Songwiedergabe funktionieren
Testurteil	Bestanden: Sämtliche Funktionen während der Songwiedergabe funktionierten

3.

Testziel	Songs nach Interpreten sortieren
Schnittstelle/Klasse	AlbumListFragment, MediaListFragment
Vorbedingung	Songs besitzen sind Interpreten zugeordnet
Nachbedingung	Songobjekte existieren
Bestehenskriterium	Songobjekte können Interpreten zugeordnet werden
Testurteil	Bestanden: Songs konnten nach Interpreten sortiert werden.

4.

Testziel	Wiedergabe aller Songs eines Interpreten / einer Playlists
----------	--

Schnittstelle/Klasse	AlbumListFragment, PlaylistFragment
Vorbedingung	Interpretenzuordnung/ Playlistobjekte existieren
Nachbedingung	Wiedergabefunktionen auf Interpretenzuordnung/ Playlistobjekten funktioniert
Bestehenskriterium	Es können alle Songs eines Interpreten bzw. einer Playlist wiedergegeben werden.
Testurteil	Bestanden: Es konnten alle Songs eines Interpreten bzw. einer Playlist wiedergegeben werden.

5.

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