

**Technischen Hochschule Nürnberg  
Georg Simon Ohm**

**Informatik**

**Bachelorarbeit**

**Entwicklung und Realisierung einer Applikation  
für das Erstellen von Sendungen für ein Webradio**

Autor: Christian Freund

Matrikelnummer: 2137526

Erstkorrektur: Prof. Dr. U. Wienkop

Zweitkorrektur: Dipl.-Inf. (FH) A. Schneider

Bearbeitungszeit: 14.03.2013 – 14.08.2013

## **I. Erklärung**

Ich, Christian Freund, erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

## **II. Vorwort**

Das Thema der Bachelorarbeit habe ich vor allem deshalb gewählt, da es mich persönlich sehr interessiert hat, wie ein solches Programm funktioniert.

Privat bin ich schon öfters mit solchen oder ähnlichen Programmen in Kontakt gekommen, doch die dahintersteckende Technik sowie die dafür nötigen Funktionen waren für mich immer ein Rätsel. Das Thema sollte für mich so gewählt sein, das ich viel Neues lernen konnte, vor allem in Bereichen, die vom Studium nur angeschnitten oder gar nicht abgedeckt wurden.

Danken möchte ich im besonderen Herrn Prof. Dr. Uwe Wienkop sowie Herrn Dipl.-Inf. (FH) Alexander Schneider, für das Thema dieser Bachelorarbeit und die Betreuung über den ganzen Zeitraum.

Des weiteren möchte ich noch dem IT-Projektteam danken, die sich um die Erstellung der Serverseite gekümmert haben und immer schnell und zuverlässig auf Änderungswünsche eingegangen sind.

### III. Inhaltsverzeichnis

I. Erklärung.....	2
II. Vorwort.....	2
IV. Ausarbeitung.....	5
1. Problemstellung.....	5
1.1. Das alte System.....	5
1.2. Das neue System.....	6
1.3. Unterstützung mehrerer Sendungsformate.....	6
1.4. All-in-One-Lösung.....	6
1.5. Benutzerfreundlichkeit.....	7
2. Zielsetzung.....	8
2.1. Auswahl der Plattform: iPad.....	8
2.1.1. Vorteile des iPad.....	8
2.1.2. Alternative Systeme.....	9
2.2. Verwendete Werkzeuge.....	12
2.2.1. Xcode.....	12
2.2.1.1. AVFoundation.....	12
2.2.1.2. CoreMedia.....	13
2.2.2. iPad Mini.....	13
2.3. Sendungsformate.....	14
2.3.1. Livesendung.....	14
2.3.2. Vorgefertigte Sendung.....	15
2.3.3. Alternative Sendungsformate.....	16
2.4. Aufgabenstellung.....	18
3. Umsetzung.....	19
3.1. Design.....	19
3.2. Login.....	20
3.2.1. Login über ein externes Tool.....	21
3.2.2. Speicherung der Logindaten.....	21
3.3. Sendungsauswahl.....	23
3.3.1. Rest-Abfrage der Sendungen.....	24
3.3.2. Sendungswechsel.....	25

3.4. Livesendung.....	25
3.4.1. Statusupdates.....	26
3.4.2. Abfrage von Liedern.....	27
3.4.3. Sprachübertragung zum Server.....	27
3.5. Vorgefertigte Sendung .....	28
3.5.1. Nutzung lokaler Lieder.....	29
3.5.2. Verschiedene Mischeffekte.....	29
3.5.3. Aufzeichnungen von Sprachaufnahmen.....	30
3.5.4. Verarbeitung der Wiedergabeliste.....	31
4. Herausforderungen.....	32
4.1. Sprachübertragung bei Livesendungen.....	32
4.2 Vorgefertigte Sendungen als MPEG-4.....	33
4.3 Audioexporter.....	34
4.4 XML-Parser.....	34
5. Schluss.....	36
5.1 Ausblick.....	36
5.2. Fazit.....	37
IV. Glossar.....	38
V. Stichwortverzeichnis.....	39
VI. Quellen.....	40
VII. Anhang.....	41

## **IV. Ausarbeitung**

### **1. Problemstellung**

Die Verbesserung des Ohm-Webradios, vor allem auf Anwenderseite, stand immer im Vordergrund. So sollte eine Schnittstelle für den Benutzer entwickelt werden, mit der er einfach und schnell eine ganze Sendung erstellen kann, ohne viel Aufwand oder einer physischen Präsenz im Radiostudio.

Die Bachelorarbeit beschäftigte sich vor allem mit der Entwicklung und Umsetzung einer Applikation für die Moderatoren, um einfach und schnell Sendungen produzieren zu können.

Der dazugehörige Server wurde und wird noch von einer IT-Projektgruppe umgesetzt.

#### **1.1. Das alte System**

Das alte System für das Radio bestand aus einer Komplettlösung, die keine Schnittstelle nach außen geboten hat. Dadurch waren die Benutzer gezwungen im Studio zu erscheinen, damit sie ihre Radiosendung produzieren können. Gerade in der heutigen Zeit war das natürlich untragbar und viel zu aufwendig. Dadurch wurde das Radio mit der Zeit immer unbeliebter und es lief fast nur noch zufällig generierte Musik aus der Datenbank.

Außerdem gab es kein richtiges Sendungsmanagement, jeder der eine Sendung machen wollte, konnte den Zugang anfordern und nach einer grundlegenden Einweisung einfach beginnen. Dadurch konnten Hörer meist nicht rechtzeitig informiert werden, dass es eine Sendung geben würde.

## **1.2. Das neue System**

Das neue System auf Serverseite wurde so konzipiert, dass es eine Schnittstelle nach außen bietet. Dadurch können verschiedene Anwendungen sich von außen verbinden, da es sich um eine Modulbauweise handelt, können für verschiedene Anwendungen auch verschiedene Module entwickelt werden, um Anwendungsbesonderheiten abzudecken.

Außerdem bietet das neue System ein deutlich komplexeres Sendungsmanagement, dadurch sind Anwendungen aber dazu gezwungen sich an diesem Sendungsmanagement zu orientieren.

## **1.3. Unterstützung mehrerer Sendungsformate**

Durch den Benutzer sind viele verschiedene Sendungsformate gewünscht, für jeden Moderator und jede Situation ein genau passendes Format. Natürlich konnten nur die wichtigsten Sendungsformate für die engere Auswahl ausgewählt werden. Dabei sollte auch darauf geachtet werden, dass das neue System nicht nur für erfahrende Moderatoren sondern auch für Anfänger und einmalig sendende Benutzer geeignet sein sollte.

## **1.4. All-in-One-Lösung**

Jede Anwendung sollte den ganzen Umfang an Funktionalität bieten, so dass der Benutzer nicht dauernd zwischen verschiedenen Systemen wechseln muss. Dadurch sind alle wichtigen Informationen an einem Platz konzentriert und übersichtlich dargestellt. Die Applikation kann dadurch auch von sich aus reagieren, falls der Benutzer eine seiner Sendungen vergisst.

## **1.5. Benutzerfreundlichkeit**

Wie schon erwähnt, sollte die Anwendung nicht nur von Moderatoren mit viel Erfahrung im Bereich Webradio bedienbar sein, sondern auch für Anwender, die das zum ersten Mal machen oder nur ein einziges Mal für eine Sondersendung. Dadurch musste gewährleistet werden, dass die Anwendung sehr einfach zu bedienen ist. Sie sollte auch ohne viel Übung schnell und einfach zu guten Resultaten führen. Das stellte natürliche hohe Anforderungen an die Umsetzung und das Layout jeder Anwendung, gerade bei mobilen Endgeräten, bei denen die Bedienung eingeschränkter ist als einem Laptop oder Desktop-PC mit Tastatur und Maus.

## **2. Zielsetzung**

Während der Bearbeitung des Projektes wurden des öfteren die Anforderungen erweitert oder verändert. Die Gründe dafür waren neue Ideen oder die Verbesserung der Benutzerfreundlichkeit. Die verwendete Hardware spielte bei der Umsetzung der Bachelorarbeit auch eine große Rolle.

### **2.1. Auswahl der Plattform: iPad**

Für die Umsetzung der Anwendung standen mehrere Systeme zur Auswahl. Das iPad von Apple wurde schließlich ausgewählt, da es zahlreiche Vorteile gegenüber den anderen Systemen geboten hat. Auch die Verbreitung dieses Systems spielte bei der Auswahl eine Rolle, da viele der derzeitigen Moderatoren benutzen ein solches Tablet privat.

#### **2.1.1. Vorteile des iPad**

Das iPad wurde nicht ohne gute Gründe ausgewählt, es bietet gegenüber den Alternativen deutliche Vorteile, dazu gehören zum Beispiel die Hardwareunterstützung für Audioverarbeitung. Durch diese Hardwareunterstützung kann eine Sendung sehr viel schneller erzeugt werden, als durch eine reine Softwareunterstützung möglich wäre. Allerdings ist diese Hardwareunterstützung nur dann verfügbar, wenn es sich um einige bestimmte Dateiformate handelt.

Ein weiterer Vorteil ist die standardisierte Hardware des iPad, so kann man zutreffende Aussagen über die Verarbeitungsdauer einer Sendung treffen, natürlich sind kleinere bis mittlere Abweichungen möglich.

Da die Hardware bei allen iPads identisch ist, ist auch die Bildschirmgröße identisch,



dadurch ändert sich die Benutzeroberfläche nicht, wenn man die Anwendung auf unterschiedlicher Hardware ausführt.

Zusätzlich spielte auch Verbreitung des iPads eine große Rolle. Das iPad ist eins der am weitesten verbreiteten Tablets. Somit erreicht man mit der Anwendung ein breiteres Spektrum an möglichen Benutzern.

Neben den Hardwarevorteilen gibt es auch noch einige Softwarevorteile, die einen starken Einfluss auf die Auswahl hatten. So bietet die iOS SDK bereits mitgelieferte Bibliotheken für die Audioverarbeitung, die gleichzeitig noch intern mit den Hardwareressourcen verknüpft sind.

Des weiteren sind fast alle Objekte der Benutzeroberfläche anpassbar, skalierbar und vollständig animiert, wodurch eine sehr einfach zu bedienende Benutzeroberfläche garantiert ist. Da die meisten anderen Applikationen auch diese standardisierten Objekte benutzen, ist es nicht notwendig sich an eine neuartige Bedienung zu gewöhnen.

Ein weiterer wichtiger Vorteil für die Entscheidung war, dass es sich bei diesem System um ein mobiles Endgeräte handelt, dadurch erhöhen sich die Einsatzmöglichkeiten enorm, da man so nicht an einen festen Standort gebunden ist. So kann man auch von unterwegs, eine passende Internetverbindung vorausgesetzt, eine Sendung produzieren.

Viele der alternativen Systeme bieten nur einen kleinen Teil oder keinen dieser Vorteile.

### **2.1.2. Alternative Systeme**

Neben dem iPad gab es noch viele Konkurrenten, dabei haben viele von diesen auch Vorteile, aber auch Nachteile. Zu den wichtigsten Konkurrenten gehörten Windows 7 und Android 4.0 und höher.

#### **Windows 7**

Zu den Vorteilen gehörten vor allem die teilweise sehr viel stärkere Hardware. Dadurch kann die Hardwareunterstützung des iPad's meist sogar übertroffen werden.

Ein weiterer Vorteil ist die sehr starke Verbreitung, dadurch könnte man sehr viele potenzielle Moderatoren erreichen.

Der große Nachteil dieses Systems ist allerdings die Ortsbindung, die von Desktop-PCs ausgeht. Auch ein Laptop kann diesen Makel nicht ganz beheben, denn diese sind meist weniger leistungsfähig und benötigen unter Umständen zusätzliche Peripherie, wie ein Mikrofon.

Ein weiterer Nachteil ist die nicht standardisierte Hardware, dadurch kann die Leistung und die Benutzeroberfläche je nach eingesetzter Hardware sehr unterschiedlich sein.

Ein weiterer Grund für die Entscheidung gegen dieses System war das Alter.

Windows 7 ist schon relativ alt und es gibt bereits ein neues Betriebssystem von Microsoft, nämlich Windows 8. Deshalb wird die Unterstützung dieses Systems durch den Hersteller in einiger Zeit ausgesetzt oder beendet.

## **Windows 8**

Hier gibt es so in etwa die gleichen Vorteile wie bei Windows 7, allerdings gibt es auch einige markante Unterschiede. Windows 8 eignet sich auch für Touchscreen-Laptops und Tablets. Allerdings ist es im Vergleich zu Windows 7 nur gering vertreten. Dadurch hat man nicht diese Breitenwirkung. Zusätzlich wurde als Kritikpunkt dieses System gebracht, dass sich die Hardware noch mehr unterscheidet. Da hier nicht nur die verbaute Hardware sondern auch die Eingabegeräte unterscheiden können. Somit wäre es sehr aufwendig gewesen eine gute Benutzeroberfläche zu gestalten, die aber gleichzeitig nicht zu überladen oder schwer bedienbar auf unterschiedlichen Systemen ist.

## **Mac OS**

Dieses System bietet eine sehr ähnliche Vielfalt, wie das eingesetzte System.

Allerdings ist dieses System nicht ganz so weit verbreitet wie das iPad.

Da es sich hierbei nicht um ein mobiles Endgerät handelt, ist die Breite der Einsatzmöglichkeiten reduziert.

Im Vergleich zur iOS SDK bietet Mac OS aber mehr Funktionalität auch eine Hardwareunterstützung ist vorhanden.

## **Linux**

Die Hauptgründe gegen dieses System war die geringe Verbreitung und damit die relativ geringe Anzahl an Benutzern, die dieses System nutzen könnten.

## **Java**

Dieses System bietet zwar einige Vorteile, aber leider auch einige Nachteile.

Die Anwendung wäre auf vielen verschiedenen Systemen ausführbar, da Java seine eigene virtuelle Maschine mitbringt, in der das Programm ausgeführt wird.

Allerdings liegt in dieser virtuellen Maschine auch der Haken, sie ist relativ langsam, verbraucht selbst viel Leistung und ist fehleranfällig.

Außerdem ist die Hardware des Systems nicht standardisierte, dadurch kann es zu starken Schwankungen von Leistungen und starken Veränderungen in der Darstellung kommen.

Java ist zwar weit verbreitet, aber gerade auf mobilen Endgeräten ist der Einsatz nicht möglich, mit der Ausnahme Android.

Daher wurde sich gegen dieses System entschieden, da man nicht die Vielfalt der Möglichkeiten der Anwendung ausschöpfen kann.

## **Android**

Hierbei handelt sich es um eine Abwandlung von Java, die für mobile Endgeräte konzipiert wurde. Sie bietet einen ähnlichen Funktionsumfang wie Java. Allerdings variieren auch hier die eingesetzte Hardware sehr stark, gerade die Größe des Bildschirms schwankt hier sehr stark, da die neuste Androidversion nicht nur für Tablets sondern auch für neue Smartphones zur Verfügung steht.

Durch diesen breiten Einsatz ist es fast unmöglich eine Benutzeroberfläche zu erstellen, die auf jeden Endgerät bedienbar bleibt. Auch die Leistung würde sehr schwanken, da kaum beziehungsweise keine Hardwareunterstützung garantiert werden kann.

## **2.2 Verwendete Werkzeuge**

Während der Bearbeitung der Arbeit musste ich mehrere Werkzeuge benutzen, dies war zum einen ein iPad Mini, das für die Kompatibilitätstests herangezogen wurde und zum anderen um das Programm Xcode. Bei Xcode handelt es sich um eine Entwicklungsumgebung, für die verwendete Sprache Objective-C, für Applikationen von Apple.

Zusätzlich bietet Xcode einen Simulator für das iPad.

### **2.2.1 Xcode**

Das Programm bietet eine vollständige Entwicklungsumgebung für Applikationen, sowohl für die Programmierung als auch für die Entwicklung der Benutzeroberfläche. Zur Entwicklung der Benutzeroberfläche wurde das Storyboard verwendet, dieses stellt alle Anzeigefenster und ihre Beziehungen dar. Dadurch konnte ich die Benutzeroberfläche besser zwischen verschiedenen Seiten abgleichen, damit eine gewisse Form erhalten bleibt.

Viele grundsätzliche und oft benutzte Funktionen sind automatisch schon eingebunden, einige zusätzliche Funktionen müssen über Frameworks hinzugefügt werden. Die wichtigsten Frameworks, die benutzt wurden, sind: AVFoundation und CoreMedia.

#### **2.2.1.1. AVFoundation**

Bei diesem Framework handelt es sich um eine Sammlung von Funktionen, die sich mit dem Verarbeiten und Bearbeiten von Audio- und Videodateien.

Aus dieser Sammlung wurden vor allem Funktionen benutzt, die sich mit der Audioverarbeitung befassen. Dazugehören Funktionen wie der Exporter und Tracks.

Diese Funktionen werden immer dann aufgerufen, wenn eine ganze Sendung produziert wird.

Auch wurde über diese Bibliothek, im Zusammenspiel mit CoreMedia unter Verwendung von Streams, das Übermitteln der Sprachübertragung bei Livesendungen ermöglicht.

Dabei wird die Sprachübertragung in Pakete aufgeteilt und anschließend über deinen geöffneten Stream an den Server geschickt.

Dieses Framework wurde gewählt, weil es standardmäßig zum iOS SDK gehört, dadurch kann man davon ausgehen, dass es regelmäßig gewartet und verbessert wird. Wichtig wird das sobald eine neue Version des SDK veröffentlicht wird. Ein externes Framework müsste dann mühsam von Hand neu eingepflegt werden.

Im Zusammenspiel mit dem Framework CoreMedia können die Funktionen auch auf die Hardwareunterstützung zurückgreifen, um eine deutlich bessere Performance zu ermöglichen.

#### **2.2.1.2. CoreMedia**

Dieses Framework stellt die grundsätzlichen Funktionen für die Arbeiten mit Audiodateien zur Verfügung und bietet dafür auch die Hardwareunterstützung an.

Über dieses Framework werden zum Beispiel die Länge eines Musikstücks ausgelesen. Außerdem bietet es die Hardwareunterstützungsschnittstelle für ein anderes Framework an.

#### **2.2.2. iPad Mini**

Das iPad Mini bietet die gleichen Funktionalitäten wie ein normales iPad, es ist nur kleiner und nicht ganz so leistungsfähig. Es ist also zum Testen der Kompatibilität gut geeignet, für die Leistung leider nicht. Eine Aussage über die Leistung ist aber immer nur relativ, da neben der vom Hersteller angegebenen Leistung auch die sonstigen

Applikationen auf dem Tablet eine große Rolle spielen. Gerade wenn leistungsintensive Applikationen oder viele kleine Applikationen im Hintergrund laufen, kann es die Leistung dieser Applikation mindern.

## **2.3. Sendungsformate**

Jeder Moderator hat seine eigene Art eine Sendung zu gestalten, das spiegelt sich in der Anzahl der denkbaren Sendungsformate wieder. Es gibt allgemeinere Formate und sehr spezielle. Da es nicht möglich ist, für jeden das passende Format zur Verfügung zu stellen, wurde der Umfang für die Bachelorarbeit auf die beiden wichtigsten Formate beschränkt. Die beiden unterstützten Formate sind die Livesendung und die vorgefertigte Sendung. Sie gehören zu den am häufigsten benutzten und anpassungsfähigsten Formaten und können auch von Anfängern schnell und einfach verstanden werden.

### **2.3.1. Livesendung**

Bei der Livesendung handelt es sich um das Facetten-reichste Format, es kann in vielen Situationen genutzt werden. Dabei wird nicht die ganze Sendung vom Tablet aus gesendet, sondern nur die Sprachaufnahmen, alles andere wird über den REST-Service geregelt. Das Vermindert die verbrauchte Internetbandbreite deutlich. Dieses Sendungsformat hat viele Vorteile gegenüber anderen, da die Einsatzmöglichkeiten sehr viel größer sind. Da nur sehr wenig Internetbandbreite nötig ist, ist es möglich über mobiles Internet eine Sendung zu produzieren. Das könnte zum Beispiel nützlich sein, wenn man auf einer Messe oder einer anderen Veranstaltung ist. So könnten Interviews direkt von der Messe aufgezeichnet und gesendet werden. Allerdings gibt es hier einige äußere Faktoren zu beachten, da bei solchen Events sehr viele Leute versuchen werden über mobiles Internet etwas zu machen, kann es zu

Verbindungsabbrüchen oder ganzen Verbindungsausfällen kommen. Hier würde zwar der Server reagieren und die Sendung mit zufälligen Material weiter gestalten, aber die Hörer könnten enttäuscht sein. Die Livesendung bietet im Vergleich zu anderen Sendungsformaten auch immer die Möglichkeit auf Wünsche der Hörer einzugehen, doch muss man aufpassen, den der Server verfügt zwar über eine große Auswahl an Liedern, doch sind nicht immer alle Wünsche dabei. Ein Hochladen eigener Musik ist nicht möglich, das wäre aus mehreren Gründen problematisch. Zu diesen Gründen zählen vor allem rechtliche Bestimmungen zu maximalen Anzahl an GEMA-pflichtigen Liedern, zum anderen aber auch an technischen Hindernissen. Ein Lied mit vier bis 5 Megabyte über mobiles Internet hochzuladen würde unter Umständen sehr lange dauern, außerdem könnte es dadurch zu Problemen mit der Synchronisierung zwischen Anwendung und Server kommen.

### **2.3.2. Vorgefertigte Sendung**

Die vorgefertigte Sendung ist das zweite umgesetzte Format, dabei handelt es sich wie der Name schon sagt um eine komplette Sendung, die als ein Stück von der Anwendung zum Server geschickt wird. Dieses Format sollte nicht über mobiles Internet versucht werden, da eine starke Internetverbindung nötig ist und sehr viel Datentransfer entsteht. Hier kann mit rund 100 Megabyte pro Stunde Sendung gerechnet werden.

Eine weitere Besonderheit dieses Formates ist es, dass nur Musik auf dem iPad benutzt werden kann. Dadurch ist der Sendungsersteller unabhängig von der derzeitigen Servermusik und kann seine Sendung individuell an seinen Stil anpassen. Das ist natürlich auch der Nachteil für alle Moderatoren, die nicht soviel eigene Musik haben.

Bei diesem Sendungsformat ist es auch möglich, Sprachaufnahmen zwischen den Liedern einfließen zu lassen. Dabei wird das interne Mikrofon genutzt und die Dateien werden zwischengespeichert und nach Beenden der Applikation entfernt um den Speicherplatz wieder freizugeben, wobei die Sprachaufnahmen relativ klein sind.

Wohingegen das zusammengeschnittene Stück sehr groß ist, deshalb sollte immer genügend Speicherplatz auf dem Tablet übrig sein.

Das Herunterladen von Musikstücken des Servers ist aus rechtlichen Gründen nicht möglich. Beim Hochladen der Sendung muss neben der Sendung auch eine Liste mit den darin enthaltenen Liedern sein, um eventuelle rechtliche Streitigkeiten besser klären zu können.

Ein weiterer Vorteil dieses Sendungsformats ist es, das der Moderator die Sendung schon lange vor der Ausstrahlung fertig machen kann. So kann er sich selbst entscheiden, wann und wie er eine Sendung in einigen Tagen produziert.

Das ist gerade dann nützlich, wenn man einen Sendetermin bekommt, an dem man nicht aktiv teilnehmen kann, aufgrund von anderen Verpflichtungen

### **2.3.3. Alternative Sendungsformate**

Neben den beiden umgesetzten Sendungsformaten gibt es noch zahlreiche andere. Jede Variante bietet aber andere Vorteile beziehungsweise Nachteile.

Ein Sendeformat für Veranstaltungen aller Art wurde wohl am häufigsten gewünscht. Es funktioniert ähnlich wie das umgesetzte Livesendungsformat. Dabei hat der Moderator allerdings keinen Einfluss auf die abgespielten Musikstücke. Er kann also nur Kommentare einfließen lassen. Zusätzlich hätte er noch die Möglichkeit Sprachaufnahmen anzufertigen und diese hochzuladen. Der Server würde diese Sprachaufnahmen dann von selbst nach dem aktuellen Lied abspielen. Diese Besonderheit lohnt sich gerade auf Messen oder Veranstaltungen, wenn man direkt von einem zum nächsten Interviewpartner übergeht, aber die Hörer nicht mit langatmigen Interviewfolgen ohne Pause für Musik verärgern möchte. Allerdings verursacht dies auch sehr viel mehr Aufwand, da die hoch geladenen Sprachaufnahmen als Lieder in das Serversystem eingefügt werden müssen.

Außerdem hat eine solche Übertragung immer ein großes Problem, bricht sie ab, muss sie von vorne begonnen werden. Bei einer Liveübertragung, muss nur der Satz oder das Wort, das abgerissen ist, wiederholt werden.



Ein weiteres Sendungsformat wäre die vom Tablet gesendete Livesendung. Sie ist eine Mischung aus vorgefertigter Sendung und Livesendung. Dabei wird zwar die Musik vom Tablet benutzt aber sie wird direkt auf den Server weitergeleitet anstatt zu einem einzigen Stück zusammengeschnittenen zu werden. Dadurch würden die beiden umgesetzten Sendungsformate zu einem einzigen verschmelzen. Doch nicht nur die Vorteile würden sich stärker auswirken, auch die Nachteile. Es würde sehr viel Internetbandbreite benötigt, und das dauerhaft. Der Moderator müsste immer dafür sorgen, dass er das nächste Lied rechtzeitig auswählt, damit die Sendung nicht verstummt. Der Server würde nicht erkennen, wann die Anwendung nicht mehr reagiert, also wann er einspringen muss und die Sendung mit der Servermusik weiterführen soll.

Außerdem müsste der Server eine direkte Leitung zum Radiosender öffnen, da sonst die Verzögerung zu groß werden würde, dadurch wird die Sicherheit des Serversystems empfindlich gestört.

Des Weiteren wäre noch ein gemischtes Sendungsformat in Frage gekommen. Dabei hätte der Moderator auf beide Musikquellen Zugriff, als auf die Musik auf seinem Tablet und auch auf die Musik des Servers. Diese Variante ist allerdings sehr problematisch, da der Server dauernd zwischen seiner eigenen Musik und der vom Tablet gesendeten Musik abwechseln müsste. Dadurch bräuchte man große Puffer um diesen Wechsel immer problemlos zu gewährleisten, bei kurzfristigen Änderungen kann es deshalb zu großen Datentransfer kommen und viele unnütze Daten geschickt und verworfen werden.

Dieses System wäre nur dann sinnvoll einsetzbar, wenn eine dauerhafte starke Internetverbindung bestehen würde, also für den mobilen Einsatz ungeeignet.

Deshalb wurde dieses Sendeformat auch abgelehnt und lieber zwei einzelne Formate erstellt.

## 2.4. Aufgabenstellung

Grundsätzlich sollte also eine Applikation entwickelt werden, die einfach zu bedienen ist, zwei Sendungsformate unterstützt und das iPad als Plattform benutzt.

Die beiden Sendungsformate Livesendung und vorgefertigte Sendung gehören dabei, wie schon erwähnt, zu den wichtigsten und am häufigsten gebrauchten Sendungsformaten.

### **3. Umsetzung**

Die folgenden Kapitel behandeln vor allem die tatsächliche Umsetzung, dabei werden die wichtigsten Punkte der Applikation angesprochen und erklärt.

Dabei wurde auf die einfache Bedienbarkeit und die daraus entstehenden Konsequenzen, wie zum Beispiel der lineare Applikationsaufbau, um Dateninkonsistenz zu vermeiden.

#### **3.1. Design**

Das Design der Benutzeroberfläche sollte relativ schlicht gehalten werden, so das neue Nutzer sich nicht lange einfinden müssen, wenn sie die Anwendung zum ersten Mal bedienen.

Als Hintergrundfarbe wurde ein dunkles grau gewählt um den Kontrast zu anderen Elementen der Benutzeroberfläche deutlicher hervorzuheben. Gerade die weißen beziehungsweise hellgrauen Schaltflächen und Tabellen heben sich somit gut von dem dunklen Hintergrund ab.

Für alle anderen Elemente wurde die standardisierte Einstellung benutzt.

Schaltflächen haben abgerundete Ecken und ein leichtes grau als Hintergrundfarbe. Tabellen hingegen sind ohne abgerundete Ecken und weiß, die einzelnen Zellen sind entweder ein oder zweizeilig.

Die Benutzeroberfläche ist so angepasst, das der Benutzer keinen Fehler bei der Bedienung machen kann, sollte etwas ausgewählt werden, werden die nicht möglichen Aktionen ausgeblendet und die möglichen werden eingeblendet. Dadurch kann es nur schwer zu Bedienungsfehlern kommen, die starke Auswirkungen auf das System haben oder dieses zum Absturz bringen.

Gerade bei leistungsintensiven Funktionen, wie dem Aufzeichnen einer Sprachaufnahme ist das sehr wichtig, da sonst nicht nur Rechenleistung sondern auch Speicherplatz unnötig verbraucht werden würde.

Bei Start einer Sprachaufnahme werden alle anderen Elemente ausgeblendet, bis auf die Schaltfläche zum Stoppen der Aufnahme.

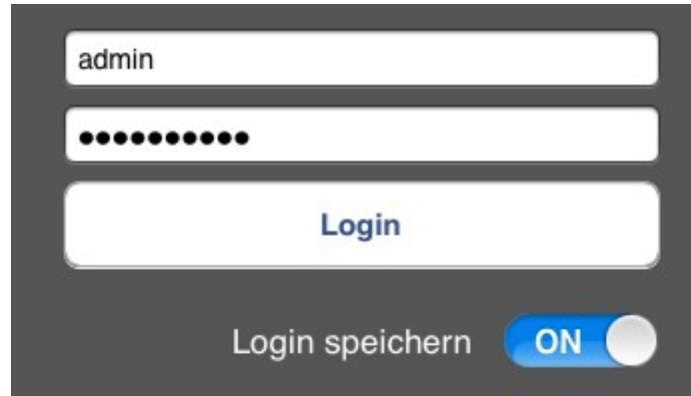
Die Bedienung soll für den Anwender intuitiv erlernbar sein, wodurch eine lange Einweisungszeit vermieden werden soll.

So wurde auf eine klare Beschriftung und ein einfaches und schlankes Design Wert gelegt, wobei auch die bereits erwähnte Bedienfehlervermeidung eine große Rolle spielte.

Eine wichtige Designentscheidung war die Gestaltung der Tabellenzellen, dabei wurde bewusst auf zu viele Informationen verzichtet um den Benutzer nicht zu überfordern. So werden nur Titel und Interpret angezeigt, sollte das nicht angegeben sein, wird der Name der Datei angegeben.

Eine Ausnahme bildet hier die Sprachaufnahme, die immer den Titel „Sprachaufnahme“ trägt und als Untertitel den Dateinamen.

### 3.2. Login



**Abbildung 1: Loginbildschirm**

Der Login ist sehr schlicht aufgebaut. Dieser Bildschirm ist als Startbildschirm gekennzeichnet, erst durch eine Authentifizierung kommt man zum nächsten Bildschirm. Außerdem bietet dieser Bildschirm noch Zugang zur allgemeinen Applikationseinstellung, wo verschiedene grundlegende Einstellungen getroffen werden können.

Die Zugangsdaten können gespeichert werden, so dass es nicht nötig ist, diese jedes Mal neu eingeben zu müssen.

### **3.2.1. Login über ein externes Tool**

Der Login ist nicht selbst geschrieben, er läuft über ein externes Tool, das vereinfacht die Authentifizierung, da nur Benutzername und Passwort abgefragt und an die vorgefertigte Funktion weitergeleitet werden müssen. Sollte die Authentifizierung erfolgreich sein, werden die Logindaten benutzt um die Liste der ausstehenden Sendungen des Nutzers abzurufen.

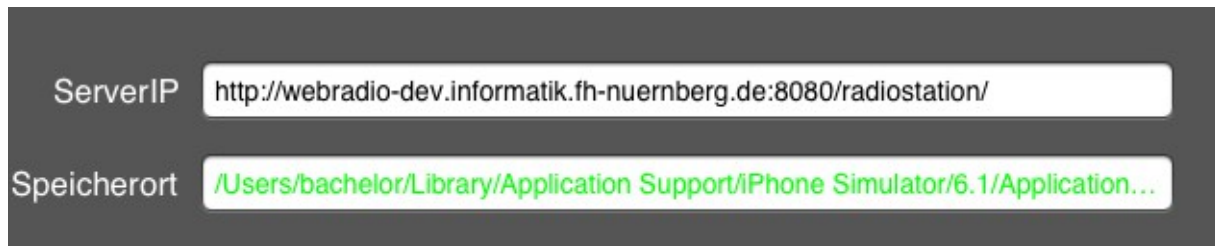
Bei einer fehlerhaften Authentifizierung werden die Felder für Benutzername und Passwort geleert beziehungsweise durch die gespeicherten Daten ersetzt.

### **3.2.2. Speicherung der Logindaten**

Die Speicherung der Logindaten kann mittels eines Schalters an- beziehungsweise ausgeschaltet werden (siehe Abbildung 1 auf Seite 20). Sollten die Logindaten gespeichert werden, werden die in den sogenannten NSUserDefaults gespeichert. Diese Art der Speicherung verhindert das Auslesen der Daten durch Dritte, da es sich hier um eine System-interne Speichermöglichkeit handelt. Allerdings können sich andere Personen dann auch einloggen, wenn sie physischen Zugriff auf das Tablet haben. Daher empfiehlt es sich, das automatische Speichern der Logindaten nur zu aktivieren, wenn man der einzige Benutzer dieses Tablets ist.

Neben den Logindaten werden noch andere Daten in diesen Speicher aufbewahrt, da sich diese Speichermethode bewährt hat.

### 3.2.3. Generelle Applikationseinstellung



**Abbildung 2: Applikationseinstellung**

Bei den generellen Applikationseinstellungen handelt es sich um einen extra Bildschirm, auf den aktuell nur zwei Einstellungen getroffen werden können.

Das ist zum einen der Speicherplatz für alle anwendungsbezogenen Dateien und zum anderen die Serveradresse des Radiosenders.

Das macht gerade deshalb Sinn, da sich die Serveradresse eventuell ändern kann.

Derzeitig befindet sich der Server in der Aufbauphase und wird unter Umständen auf eine andere Domain umgelegt, sobald die Entwicklung abgeschlossen ist.

Die Anwendung speichert seine Daten automatisch im Dokumentenordner des Tablets, wer das nicht möchte, kann es auf diesen Bildschirmabschnitt ändern.

Die getroffenen Einstellungen werden in den `NSUserDefaults` gespeichert und sind somit nur innerhalb der Anwendung durch das Programm einsehbar. Alle Funktionen, die Dateien anlegen, löschen oder verändern, greifen auf die in den `NSUserDefaults` gespeicherten Einträge zurück.

Weiterhin kann dieser Bildschirm erweitert werden, falls es in der Zukunft weitere wichtige Einstellungen geben sollte.

Über eine Schaltfläche werden die eingegebenen Daten gespeichert und man kehrt zum Loginbildschirm zurück, die andere Schaltfläche ist zum Zurücksetzen der Eingaben.

Das Zurücksetzen setzt die Werte auf die in den `NSUserDefaults` gesetzten Werte, falls keine vorhanden sind, werden die Werte auf den Standardpfad gesetzt.

### 3.3. Sendungsauswahl

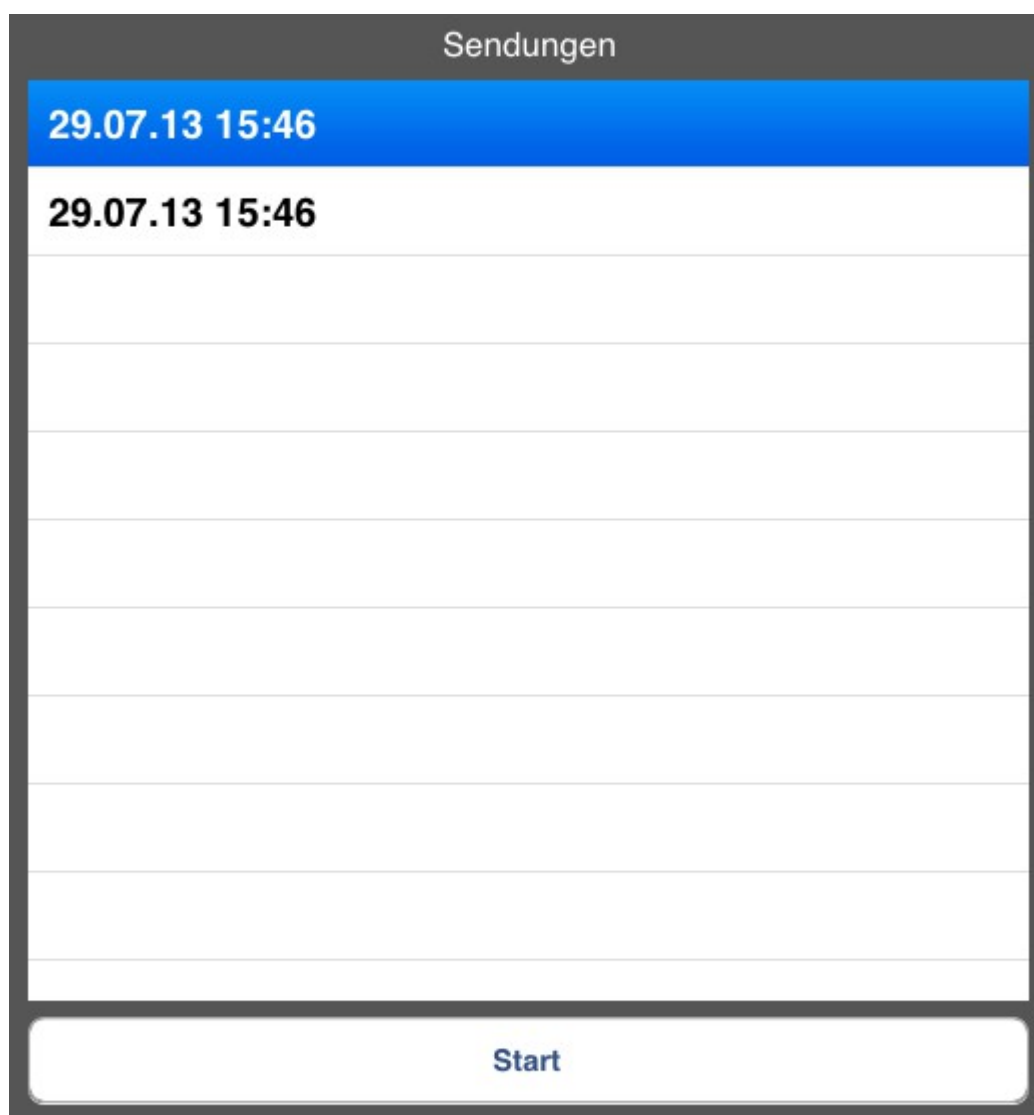


Abbildung 3: Sendungsauswahl

Dieser Bildschirm behandelt vor allem die Auswahl der möglichen Sendungen.

Der Bildschirm teilt sich in zwei Teile, die linke Seite umfasst eine Tabelle in der alle offenen Sendungen des eingeloggten Moderators angezeigt werden. Hierbei wird nur das Datum und die Uhrzeit angezeigt(siehe Abbildung 3 auf Seite 22).

Die rechte Seite enthält vor allem Informationen, sobald links eine Sendung ausgewählt wurde. Dabei werden auf verschiedene Punkte eingegangen. Das wären der Sendungstitel, der kurz und knapp den Inhalt der Sendung umreißt. Dies ist dann nötig, wenn ein Moderator mehrere Sendungen am gleichen Tag hat oder seine Sendungen immer anders gestalten möchte. Darunter befindet sich ein Feld für die Sendungsbeschreibung, diese gibt eine ausführlichere Inhaltsbeschreibung wieder, so können weitere Informationen über die Sendung eingesehen werden. Des weiteren wird die Sendungslänge angezeigt, dadurch hat der Moderator einen guten Überblick, was er an Musik braucht oder wie viel Zeit er investieren muss. Außerdem kann er bei einer vorgefertigten Sendung auch schon etwa die benötigte Zeit für die Verarbeitung und das Hochladen der Musik erahnen.

Das letzte Feld ist für das Sendungsformat reserviert, hier wird kurz angezeigt, um welches Sendeformat es sich handelt, aktuell können es nur die beiden umgesetzten Formate sein, Livesendung und vorgefertigte Sendung.

Dadurch hat der Moderator immer direkt eine komplette Übersicht über eine Sendung, kann seine Musik anpassen und eine gute Sendung produzieren.

Die ausgewählte Sendung wird in die NSUserDefaults eingetragen, um die wichtigsten Daten auch auf anderen Bildschirmen abrufen zu können.

Über eine Schaltfläche wird man dann weitergeleitet auf den eigentlichen Bildschirm zur Sendungserstellung.

Neben den angezeigten Daten gibt es noch einige System-interne Daten, die mit in die NSUserDefaults übernommen werden, darunter fällt zum Beispiel die Sendungsnummer des Servers.

### **3.3.1. Rest-Abfrage der Sendungen**



Sobald der Login authentifiziert wurde, wird an den Server eine Anfrage gesendet, dadurch werden alle offenen Sendungen abgefragt. Als Antwort erhält man eine XML-Datei mit allen wichtigen Informationen. Diese Antwort kann sehr unterschiedlich ausfallen. Da bei Moderatoren mit vielen Sendungen diese Datei sehr viel größer sein wird, als bei einem Moderator, der nur eine oder zwei Sendungen hat. Diese Datei wird zwischengespeichert und direkt im Anschluss von einer speziellen Funktion ausgelesen und in eine verwendbare Form gebracht. Sollten Informationen, wie zum Beispiel die Sendungsbeschreibung, fehlen wird das entsprechende Feld ausgeblendet.

### **3.3.2. Sendungswechsel**

Einmal entscheiden, kann man die Auswahl der Sendung nicht mehr rückgängig machen. Das hat mehrere Gründe, zum einen die verschiedenen Sendungsformate. So ist es nicht möglich, einfach zwischen den beiden Sendungsformaten zu wechseln, ohne die aktuelle Liste an Liedern zu verlieren. Da die Liedlisten aus zwei verschiedenen Quellen kommen, ist es nur schwer diese abzugleichen und gleiche Musikstücke zu erkennen. Daher müssten die ganzen Liedlisten geleert werden, wäre vor allem bei einer Livesendung fatal, wenn die Musik plötzlich stoppen würde, bis der Moderator wieder auf der richtigen Sendung ist und neue Lieder in die Liste eingefügt hat.

Sollte man sich wirklich, aus versehen für die falsche Sendung entschieden haben, bleibt nur eins, die Applikation neu starten. So ist aber auch sichergestellt, dass keine alten Daten mehr im Zwischenspeicher zu finden sind.

### **3.4. Livesendung**

Bei der Livesendung handelt es sich um das technisch einfachere Sendungsformat. Die Musik wird nicht auf dem Tablet verarbeitet, sondern nur Anweisungen gegeben, welche Lieder als nächstes an der Reihe sind.

Dadurch muss hier nur eine Liste geführt und aktualisiert werden. Um die verbrauchte Internetbandbreite zu vermindern, wird nur alle zwei Sekunden eine Meldung an den Server geschickt und nach dem aktuellen Status gefragt.

An der oberen Seite befindet sich eine Statusleiste, die das aktuelle Lied anzeigt und dessen Fortschritt. Die Farbe der Statusleiste ändert sich je weiter das Lied fortgeschritten ist. Dadurch hat der Moderator einen guten Überblick, wann er seine Kommentare einfließen lassen kann, die Hörer wären sicher nicht erfreut, wenn der Moderator mitten im Lied seinen Kommentar zum nächsten Lied abgeben würde.

Der restliche Bildschirm teilt sich in drei Teile, der linke Teil besteht nur aus der Wiedergabeliste, welche die ausgewählten Lieder enthält, wobei das aktuelle immer durch das oberste Lied repräsentiert wird. Der mittlere Teil ist vor allem für das Management der Wiedergabeliste verantwortlich, so befinden sich Schaltflächen für das Hinzufügen und Entfernen von Liedern aus der Wiedergabeliste und das Ändern der Reihenfolge dieser Liste im mittleren Teil des Bildschirms. Zusätzlich gibt es noch eine weitere Schaltfläche, welche die Sprachübertragung auslöst. Ein weiterer Druck auf diese Schaltfläche, beendet die Übertragung wieder. Die rechte Seite besteht aus einer weiteren Liste und einigen Funktionen, zum Management dieser Liste.

Bei der Liste handelt es sich um eine Sammlung aller vom Server abgefragter Lieder. Zusätzliche Schaltflächen und Eingabemöglichkeiten dienen zum Management dieser Liste, die Hauptfunktionen sind dabei das Leeren und das Füllen.

Die Bedienung ist sehr schlicht und einfach gehalten, so werden zum Beispiel Schaltflächen für bestimmte Aktionen erst nach der Auswahl eines Liedes sichtbar und verschwinden auch nach Betätigung wieder.

### **3.4.1. Statusupdates**

Die Statusupdates spielen gerade bei diesem Sendungsformat eine tragende Rolle. So wird nicht nur die Wiedergabeliste synchronisiert sondern auch der aktuelle Zustand des aktuell abgespielten Liedes übermittelt.

Das ist gerade bei einer un stetigen Internetverbindung, zum Beispiel auf einer Messe oder im Stadion, sehr wichtig, da so kleinere Verbindungsabbrüche nicht weiter tragisch sind. Bei der nächsten Synchronisierung wird die Wiedergabeliste angepasst und eventuelle Inkonsistenzen beseitigt.

Da eine solche Anfrage relativ lange braucht, bis eine Antwort zurückkommt und bis diese interpretiert ist, wird nur alle zwei Sekunden eine solche Anfrage gesendet. Ein weiterer Vorteil dieser zwei Sekunden Verzögerung ist die Schonung der Internetbandbreite. Dadurch ist diese Methode für das mobile Internet geeignet.

### **3.4.2. Abfrage von Liedern**

Das Abfragen der Lieder ist für den Benutzer sehr einfach. Er muss nur in ein Textfeld das entsprechende Wort eingeben. Dabei macht es keinen Unterschied, ob es sich um ein Genre, einen Interpreten oder den Titel des Liedes handelt. Mit dem eingegebenen Text wird eine Anfrage an den Server angefertigt und abgeschickt.

Der Server liefert daraufhin eine Datei mit allen Liedern, die zu dem eingegebenen Text passt. Dabei kann es passieren, dass sehr viele Lieder ausgewählt werden, die teilweise gar nicht gewünscht sind. Sollte das passieren, kann die Liste der möglichen Lieder geleert werden und eine genauere Anfrage geschickt werden.

Die Liste ist generell unbegrenzt, allerdings kann es passieren, dass eine sehr große Liste, das Programm merklich verlangsamt. Sollte dies eintreten, hilft eine einfache Leerung der Liste um die Performance zurückzugewinnen.

### **3.4.3. Sprachübertragung zum Server**

Um den Moderatoren nicht nur die Möglichkeit zu bieten, Musik nach ihrer Wahl zu spielen, gibt es das Feature der Sprachübertragung. Dadurch wird dem Moderator die Möglichkeit geboten Kommentare, Interviews oder ähnliche Sprachbeiträge in die Sendung einfließen zu lassen.

So wird durch einen einfachen Klick auf eine Schaltfläche die Sprachübertragung gestartet, ein weiterer Klick beendet die Übertragung. Während der Übertragung ist es nicht möglich irgendwelche Änderungen an der Wiedergabeliste durchzuführen.

Das soll vor allem verhindern, dass der Moderator vergisst die Sprachübertragung zu beenden und so Kommentare oder Aussagen weitergeleitet werden, ohne dass es Moderator bemerkt oder gewollt hat.

Ein Toggle-Modus wäre auch möglich, doch ist es bei längeren Interviews oder Sprachbeiträgen nicht wünschenswert eine Schaltfläche dauerhaft gedrückt halten zu müssen.

Sobald die Schaltfläche ausgelöst wurde, wird ein Puffer angelegt, der durch den Output der angeschlossenen Peripherie gefüllt wird. Sollte der Puffer zu einem gewissen Stand gefüllt sein, wird dieser Puffer an den Server geschickt und dort in die Musik eingemischt.

Durch den Puffer auf Applikationsseite und einen weiteren Puffer auf der Serverseite kann es zu mehreren Sekunden Verzögerung kommen, bis ein Sprachbeitrag wirklich beim Hörer ankommt.

### **3.5. Vorgefertigte Sendung**

Bei der vorgefertigten Sendung handelt es sich um die zweite Variante der umgesetzten Sendungsformate. Dabei wird das Tablet aber deutlich mehr beansprucht, als bei der Livesendung. Die Benutzeroberfläche ähnelt der Livesendung, doch gibt es einige deutliche Veränderungen.

Es gibt zum Beispiel die Statusleiste nicht, die hier in diesem Format auch nicht gebraucht wird. Dafür wurden neue Funktionalitäten im linken Teil der dreiteiligen Benutzeroberfläche eingefügt. Dabei handelt es sich um verschiedene Mischeffekte, für die Lieder in der Wiedergabeliste. Der mittlere Teil ist fast unverändert, so können immer noch Lieder zur Wiedergabeliste hinzugefügt werden oder wieder entfernt, auch die Reihenfolge der Lieder kann noch verändert werden. Doch die Schaltfläche für die Sprachübertragung wurde entfernt, da es hierfür einen gesonderten Eintrag in der Liedliste gibt, dieser steht ganz oben. Die rechte Seite bietet nur noch die Liedliste an, da weitere Funktionalitäten nicht nötig sind. Es werden automatisch alle Lieder des entsprechenden Formats ausgewählt und eingefügt. Allerdings kann es bei der

Benutzung von iTunes zu einigen Problemen kommen, wenn iTunes die Namen der Lieder ändert, das muss über iTunes deaktiviert werden.

### **3.5.1. Nutzung lokaler Lieder**

Dieses Sendeformat greift nur auf Lieder zu, die sich bereits auf dem Tablet befinden. Dadurch ist es nicht nötig, Anfragen oder ähnliches an den Server zu senden. Die Musik muss allerdings schon vor dem Start der Anwendung auf dem Tablet sein. Ein rekursiver Algorithmus durchforstet dann alle Ordner und fügt die Lieder der Liste hinzu. Das kann je nach Leistung und belegtem Speicher einige Zeit dauern.

Unterstützte Dateitypen sind aktuell nur Dateien mit dem Typ MPEG-3, da eine Mischung verschiedener Taktfrequenzen im späteren Verlauf zu Problemen führen könnte. Es wäre möglich gewesen, diese Lieder mit unterschiedlicher Taktfrequenz auf einen einheitlichen Wert zu bekommen, allerdings hätte das zu massiven Qualitätsverlusten geführt, daher wurde sich entschieden, nur Dateien des Typs MPEG-3 zu unterstützen.

### **3.5.2. Verschiedene Mischeffekte**

Bei der vorgefertigten Sendung sind drei Mischeffekte verfügbar, dabei handelt es sich um Fade-In, Fade-Out und den Crossfade. Diese drei Mischeffekte sind für alle Lieder der Wiedergabeliste verfügbar, mit Ausnahme von Crossfade für das letzte Lied. Dort kann es zwar eingestellt werden, allerdings hat es keinen Effekt.

Bei den Mischeffekten handelt es sich um die klassischen Effekte, die jeder Moderator einfach und leicht in seine Sendung einfließen lassen kann.

Fade-In bezeichnet das Einblenden der Musik, so startet die Musik mit vollkommener Stille und wird kontinuierlich lauter, bis sie das Maximum erreicht.

Dieser Effekt wirkt sich direkt am Start des jeweiligen Liedes aus, nicht in der Mitte oder Schluss

Der Fade-Out-Effekt ist das genaue Gegenteil des Fade-In-Effekts, er lässt die Musik langsam ausblenden. Deshalb ist dieser Effekt nur am Ende des jeweiligen Musikstücks verfügbar.

Eine Besonderheit bietet der Crossfade, er wirkt sich nicht nur auf ein Musikstück aus, sondern auch auf das darauffolgende. So lässt man beide Lieder überlagern, deshalb ist es auch beim letzten Lied nicht verfügbar, denn eine Überlagerung mit einem weiteren Lied ist nicht möglich. Wichtig ist hier, dass ein Fade-Out-Effekt des ersten Liedes und ein Fade-In-Effekt des zweiten Liedes zu einem Einbruch der Lautstärke führen kann, falls es mit einem Crossfade-Effekt gemischt wird.

Daher ist davon abzuraten, alle Effekte zu mischen, wobei Fade-In-Effekt und Fade-Out-Effekt zusammen gut funktionieren können.

### **3.5.3. Aufzeichnungen von Sprachaufnahmen**

Da in der vorgefertigten Sendung keine Sprachübertragung stattfindet, gibt es eine extra Aufnahmefunktion. Dazu muss eine Sprachaufnahme aus der Liedliste in die Wiedergabeliste hinzugefügt werden. Durch Auswählen der Sprachaufnahme in der Wiedergabeliste erscheint neben den Mischeffekten auch eine Aufnahmeschaltfläche. Sobald diese gedrückt wird, startet die Sprachaufnahme und alle anderen Elemente dieser Anzeige verschwinden. Dieses Vorgehen verhindert, dass eine Sprachaufnahme über längere Zeit unbeachtet weiter läuft und somit viel Speicherplatz und Leistung beansprucht. Erst durch das Betätigen der Schaltfläche für das Beenden der Sprachaufnahme werden die anderen Elemente wieder sichtbar.

Das Dateiformat für die Aufnahme ist das WAV-Format, das mit dem MPEG-3-Format vereinbar ist. Der Speicherort kann durch die allgemeinen Anwendungsanstellungen festgelegt werden, es gibt einen extra Unterordner für die Sprachaufnahmen.

Außerdem gibt es einen internen Zähler über alle ungenutzten Sprachaufnahmen der Wiedergabeliste, erst wenn dieser auf Null steht, kann die Verarbeitung der Sendung starten.

Zur Aufnahme wird im Normalfall das integrierte Mikrofon benutzt, sollte man eine höhere Qualität brauchen, kann man ein externes Mikrofon anschließen. Durch die

Hardwareunterstützung aus der CoreMedia-Bibliothek wird automatisch das externe Mikrofon als Dateneingang erkannt und aufgenommen.

#### **3.5.4. Verarbeitung der Wiedergabeliste**

Sobald die Sendungslänge, inklusive einer gewissen Toleranz, erreicht ist, wird in der Mitte des Bildschirms eine Schaltfläche sichtbar, mit dem man die Sendung endgültig produzieren und hochladen kann.

Dafür gibt es zwei Bedingungen, zum einen darf die Sendungslänge nicht mehr als zwei Minuten unterschritten oder überzogen werden, zum anderen müssen alle Sprachaufnahmen fertig aufgenommen werden. Sind diese beiden Bedingungen erfüllt, kann die Verarbeitung beginnen. Dann werden alle Lieder zu einer einzigen, sehr großen Datei zusammengeschnitten und mit den Effekten, die man einstellen konnte, versehen.

Die fertige Datei wird zusammen mit der Liste der Lieder an den Server geschickt, der sich dann darum kümmert, das es zur entsprechenden Zeit gesendet wird.

## **4. Herausforderungen**

Es gab viele kleinere und größere Herausforderungen bei der Bearbeitung der Arbeit, viele kleinere Sachen erforderten nur ein Umdenken der Herangehensweise. Es gab aber auch größere Herausforderungen, die wichtigsten waren eindeutig die Sprachübertragung bei Livesendungen, das Vorgefertigte Sendungen nur als MPEG-4-Datei verfügbar sind, das Verarbeiten der vorgefertigten Sendungen und das Auslesen der XML-Dateien.

### **4.1. Sprachübertragung bei Livesendungen**

Die Sprachübertragung zum Server stellte eine der größten Herausforderungen dar. Da es sich aber hierbei um ein essenzielles Feature der Livesendung handelt, musste es unter allen Umständen implementiert werden und immer einsatzfähig sein.

Allerdings bietet die iOS SDK keine solche Funktion an, um direkt den Ausgang des Mikrofons auf eine geöffnete Serververbindung zu legen.

Eine externe Bibliothek hat den Nachteil, dass sie zu pflegen ist. Sobald eine neue Version des Betriebssystems oder der Bibliothek vorliegt, muss erst geprüft werden, ob diese noch kompatibel ist.

Eine Möglichkeit für eine externe Bibliothek wäre Live555 gewesen, doch funktioniert das dortige Streamingformat nicht auf dem iPad.

Daher musste es über die AVFoundation- und CoreMedia-Bibliothek gehandhabt werden. Da es auch so nicht möglich ist, direkt die Audiodaten auf einen Server zu schicken, mussten erst einige Schritte unternommen werden, um die Rohdaten aufzubereiten. Die Rohdaten werden in kleinere Pakete aufgeteilt, die gerade noch so klein sind, dass sie in einem Teil über einen geöffneten Stream geschickt werden können, ohne dass zusätzliches Splitting nötig ist.

Dieses Verfahren hat leider auch einige Nachteile, so werden die Rohdaten immer gesammelt und bearbeitet, nur nicht hochgeladen. Dadurch hat man Leistungseinbußen, doch es gibt keine Verzögerung, wenn man eine Sprachübertragung



beginnt. Dadurch wird die Zeitverschiebung zwischen der Aufnahme der Sprachübertragung und dem wirklichen Aussenden dieser Übertragung wesentlich kleiner.

## **4.2 Vorgefertigte Sendungen als MPEG-4**

Da die vorgefertigte Sendung nur als MPEG-4-Datei verfügbar ist, hat mehrere Gründe. Die Herausforderung lag vor allem darin, rechtliche und leistungsbezogene Probleme zu lösen. Grundsätzlich war es geplant, dass die Audioverarbeitung eine MPEG-3-Datei erzeugt. Dabei gibt es aber ein Problem, denn dieses Format kann nicht einfach erzeugt werden, dazu müssen bestimmte Lizenzbedingungen erfüllt werden. Außerdem bietet die iOS SDK keine Möglichkeit für die Erzeugung einer MPEG-3-Datei. Das Abspielen von MPEG-3-Dateien ist allerdings kein Problem, da sich die Lizenzbedingungen nur auf das Erstellen einer solchen Datei auswirken. Auch bietet das System Möglichkeiten eine MPEG-3-Datei abzuspielen.

Deshalb musste ein alternatives Dateiformat gefunden werden, zur Auswahl stand theoretisch jedes mögliche Audioformat. Allerdings fiel die Entscheidung auf den nächsten Verwandten des MPEG-3-Formats, das MPEG-4-Format.

Dieses ist zwar kein reines Audioformat und deshalb etwas größer, doch dafür ist es bei der Verarbeitung schneller, da die eingesetzten Bibliotheken eine Hardwareunterstützung für das MPEG-4-Format. Durch diese Hardwareunterstützung kann eine deutlich bessere Performance erreicht werden.

Andere Formate wären auch möglich gewesen, da es auch für manche Formate diese Hardwareunterstützung geben würde, allerdings sind diese Formate nicht so sehr verbreitet und hätte zusätzliche Verarbeitungsschritte benötigt, damit aus den ursprünglichen MPEG-3-Dateien ein anderes Format wird.

Daher wurde das MPEG-4-Format gewählt, dieses gewährt im Durchschnitt eine schnellere Verarbeitung und Hochladen, als jedes andere Format. Außerdem ist es einfacher für den Server, eine MPEG-4-Datei anzuspielden, als eine Datei eines anderen Formats.

## 4.3 Audioexporter

Der Audioexporter ist der Programmteil, der aus der einfachen Wiedergabeliste der vorgefertigten Sendung eine Sendung macht. Hierzu werden alle Lieder der Wiedergabeliste einzeln analysiert. Zu erst wird aus der angegebenen Quelle die Datei geladen, danach wird die Länge und die Taktfrequenz ermittelt. Mit diesen Werten kann nun die Tonspur extrahiert werden, diese Tonspur wird dann bearbeitet, mithilfe der eingestellten Mischeffekte. So werden Fade-In-Effekte beziehungsweise Fade-Out-Effekte direkt auf diese Tonspur gemischt. Die bearbeitete Tonspur wird dann in eine Sammlung aufgenommen, und dort gegebenenfalls eingemischt. Dies geschieht dann, wenn der Nutzer einen Wert für den Crossfade-Effekt festgelegt hat.

Sind alle Lieder und Sprachaufnahmen in die Tonspursammlung eingefügt, startet der Audioexporter mit seiner Arbeit und erzeugt die Sendung.

Dabei dauert es je nach Anzahl der Lieder und Gesamtlänge eine unter Umständen längere Zeit bis die Sendung fertig ist. Sobald der Audioexporter fertig ist, wird eine Liste der Lieder in XML-Form erstellt und die Sendung wird dieser XML-Datei angehängt und hochgeladen.

Dadurch wird der Benutzer der Anwendung entlastet und kann mit einem einzigen Druck einer Schaltfläche seine Sendung vollständig produzieren lassen und muss sich um nichts anderes mehr kümmern.

Die Herausforderung lag hier vor allem darin, das alle Teile ineinandergreifen und die Sendung so wie gewünscht produziert wird. Gerade die Umsetzung der Mischeffekte machte immer wieder Schwierigkeiten und selbst jetzt scheint es manchmal so zu sein, als würde sie nicht immer vollständig funktionieren.

## 4.4 XML-Parser

Bei einem Parser handelt es sich um einen Programmteil, der aus einer Datei den Inhalt ausliest und diesen in eine für die Anwendung verwendbare Form zu bringen.

Normalerweise benutzt man einen Tree-Parser, dieser macht aus der XML-Datei eine Art Baum-Struktur mit einzelnen Ästen und Blättern. Dadurch kann man auf einzelne Äste beziehungsweise Blätter zugreifen und dessen Werte auslesen.

Allerdings bietet die iOS SDK nur einen Event-basierenden Parser an, dieser besteht aus drei Funktionen, die man recht einfach beschreiben kann. Die erste Funktion wird dann aufgerufen, wenn ein Tag anfängt. Ein Tag ist ein bestimmter Term, der von spitzen Klammern umschlossen wird. Die zweite Funktion kann erst dann aufgerufen werden, wenn die erste Funktion aufgerufen wurde. Diese Funktion behandelt das Verhalten, falls ein Text jeglicher Art gefunden wurde, bevor ein schließender Tag erkannt wurde. Die letzte Funktion behandelt genau diesen Fall, und zwar das ein schließender Tag gefunden wurde.

Bei dem Verarbeiten einer XML-Datei für die Anwendung, wird beim Aufrufen der ersten Funktion ein neues Objekt angelegt. Die zweite Funktion füllt dieses Objekt mit verschiedenen Werten und die letzte Funktion übergibt das Objekt an eine Sammlung, auch Array genannt. Sobald die Verarbeitung fertig ist, wird diese Sammlung an die Anwendung übergeben und dort weiterverarbeitet.

## **5. Schluss**

Die Anwendung wurde dazu konzipiert, den Moderatoren eine Möglichkeit zu bieten, von zu hause aus ihre Sendung zu veranstalten. Dieses Ziel hoffe ich natürlich gut umgesetzt zu haben. Ein großer Praxistest muss dies aber noch bestätigen, dennoch sind alle Punkte der Zielsetzung umgesetzt und sogar noch um einige weitere Funktionen erweitert. So kann der Benutzer eine einfache, schlanke Bedienoberfläche benutzen um zwei unterschiedliche Sendungen zu produzieren oder seine ausstehenden Sendungen begutachten. Die Anwendung bietet die komplette Funktionalität um nicht auf eine andere Plattform ausweichen zu müssen.

Auch sollte der Benutzer keine Probleme haben sich in die Bedienung einzufinden, da die Anwendung sehr benutzerfreundlich ist und zusätzlich noch einen hohen Grad an Fehlervermeidung aufweist.

### **5.1 Ausblick**

Die Applikation ist so angelegt, das sie jederzeit um weitere Sendungsformate erweitert werden kann. Dazu muss nur bei der entsprechenden Sendungsauswahl eine weitere Auswahlmöglichkeit hinzugefügt werden und der entsprechende neue Bildschirmabschnitt mit den dazugehörigen Programmteil geschrieben werden.

Da das neue Webradiosystem vermutlich genau solange wie das alte in Betrieb bleiben wird, ist es nach einiger Zeit notwendig das Programm anzupassen um neue Sendungsformate zu unterstützen.

Auch neuere Hardwaremodelle dürften schnellere Verarbeitung der Musikstücke und des Programms an sich ermöglichen. Dadurch wird die Applikation in einiger Zeit noch an Leistung dazu gewinnen.

Verbessert könnte die graphische Benutzeroberfläche und die Bedienung. Dazu gehört eine komplette Überarbeitung des aktuellen Designs zusammen mit der neu Anordnung der einzelnen Komponenten.

Vermutlich werden in der nächsten Zeit weitere Anwendungen zum Erstellen von Sendungen für andere Plattformen dazukommen. Dabei sollte darauf geachtet werden, dass die Funktionalitäten dieser Anwendung auch übernommen werden.

## **5.2. Fazit**

Ich fand diese Bachelorarbeit sehr lehrreich, gerade da die Grenzen des Systems schon stark ausgereizt wurden und man immer darauf achten musste, wie viel man noch machen kann ohne das System zum Absturz zu bringen.

Hierbei sollte man erwähnen, dass nicht nur die Programmierung sondern auch die rechtlichen Besonderheiten waren sehr schwierig unter einem Hut zu bekommen. So gibt es mehrere Auflagen für die Ausstrahlung von GEMA-pflichtigen Musikstücken, aber auch was das Bearbeiten von Musik angeht.

Für mich war die verwendete Programmiersprache Objective-C auch neu, deshalb hat es doch einige Zeit gedauert, bis ich mich mit den Besonderheiten dieser Sprache anfreunden konnte.

Der Umfang der Bachelorarbeit war doch größer als am Anfang von mir eingeschätzt, da immer wieder Sachen verändert werden mussten und Probleme auftraten, wo ich keine erwartet hatte. Zum Beispiel beim Auslesen von XML-Dateien oder beim Erzeugen der Sendung.

Gerade die Umstrukturierung der Anwendung kostete viel Zeit, doch war es notwendig um eine fehlerfreie Umgebung zu schaffen. Gerade der Wechsel zwischen verschiedenen Sendungen war mit ein Dorn im Auge und musste deshalb dem linearen Aufbau weichen.

## IV. Glossar

#noch einzufügen#

## V. Stichwortverzeichnis

#noch einzufügen#

## VI. Quellen

#noch einzufügen#



## VII. Anhang

#noch einzufügen#