

Masterarbeit

„Entwicklung einer mobilen Tutor Anwendung „Teach Me“ für Android
Betriebssystem– Konzeption, prototypische Umsetzung und
Anwendungsvergleich.“

Vorgelegt von: Aleksandr Soloninov

Matrikelnummer:

Email: inf3032@hs-worns.de

Fachsemester: 4

Abgabedatum: 07.10.2019

Zusammenfassung

Abstract

Danksagung

Abkürzungen

Abbildungen

Zeitplan

Dauer: 6 Monate (07.04.2019 – 07.10.2019)

Bis 08.04.: Literaturrecherche

Bis 15.04.: Konzeption und Entwicklungssprachen Auswahl

Bis 31.05.: Implementierung der App

Bis 24.06.: Erster App Test und Vergleich an der Universität Heidelberg durchzuführen und Feedback sammeln

Bis 01.07.: Ergebnisse von erstem Test zählen und Rohfassung Hauptteil

Bis 07.07.: App Evolution

Bis 01.08.: Zweiter App Test und Vergleich an der Universität Mainz durchzuführen und Feedback sammeln

Bis 07.08.: Ergebnisse von zweitem Test zählen und App Evolution

Bis 23.08.: Dritter App Test und Vergleich an der Hochschule Worms durchzuführen und Feedback sammeln

Bis 10.09.: Auswertungsvergleich und Hauptteil Verfassung

Bis 15.09.: Vervollständigung Hauptteil und Schlussverfassung

Bis 18.09.: Überarbeitung und Korrektur

Bis 21.09.: Layout, Titelblatt und restliche Korrektur

Bis 26.09.: Druck

Bis 07.10.: Abgabe

Inhaltsverzeichnis

Abkürzungen

Abbildungen

Zeitplan

1. Einleitung

1.1 Problemstellung

1.2 Motivation

1.3 Forschungskonzept

1.4 Zielsetzung und Erkenntnisinteresse

1.5 Forschungskonzept

2. Grundlagen zur Entwicklung einer Android Applications

2.1 Integrierte Entwicklungsumgebung

2.1.1 Android Studio

2.1.2 Basic for Android

2.1.3 Visual Studio

2.1.4 AIDE

2.2 Mobile Applikationen

2.2.1 Native App

2.2.2 WebApp

2.2.3 Hybride App

2.3 Android Betriebssystem

2.3.1 Anwendungskomponenten

2.3.2 Pattern

2.4 Entwicklung Sprachen für Android Betriebssystem

2.4.1 Java

2.4.2 Kotlin

2.4.4 Scriptsprachen

2.5 Versionierung

3. Rolle von Tutorien im Studienprozess

3.1 Sinn und Zweck von Tutoren an Universitäten

3.2 Arten von Tutorien

3.2.1 Orientierungstutorien

3.2.2 Fachtutorien

3.3 Aufgaben der Tutoren und Tutorinnen

4. Konzeption

4.1 Analyse bestehender Softwareprodukten

4.2 Funktionale Anforderungen

4.3 Nicht-funktionale Anforderungen

4.4 Mockup und Prototype

5. Implementierung

5.1 App Architecture

5.1.1 Komponenten Zusammenhang

5.1.2 Struktur von Quizfragen

5.2 App Engine

5.2.1 TextViewEngine

5.2.2 VideoViewEngine

5.2.3 WebEngine

5.2.4 SpeechEngine

5.3 App Activities

5.3.1 Base Activity

5.3.2 Main Activity

5.3.3 Search Activity

5.3.4 Question Activity

5.3.5 Question Select Activity

5.3.6 Question Interview Activity

5.3.7 Score Card Activity

5.3.8 Favorite Activity

5.3.8 Settings Activity

6 App Anwendungsvergleich

6.1 Auswertungsmethode und Vergleich Dokumentation

6.2 App Anforderung vergleich

6.3 Forschungsversuch an Ruprecht-Karls-Universität Heidelberg

6.3.1 Forschungsversuch am 3.06.2019

6.3.2 Forschungsversuch am 7.06.2019

6.3.3 Forschungsversuch am 17.06.2019

6.3.4 Forschungsversuch am 20.06.2019

6.3.5 Feedback und App Evolution

6.4 Forschungsversuch an Johannes-Gutenberg-Universität Mainz

6.4.1 Forschungsversuch am

6.4.2 Forschungsversuch am

6.4.3 Forschungsversuch am

6.4.4 Forschungsversuch am

6.4.5 Feedback und App Evolution

6.5 Forschungsversuch an der Hochschule Worms

6.5.1 Forschungsversuch am

6.5.2 Forschungsversuch am

6.5.5 Feedback und App Evolution

6.6 Gesamtforschungsauswertung und Kapitelfazit

7 Ausblick und Fazit

8 Literaturverzeichnis

Anhang I

Anhang II

Anhang II

Anhang IV

Anhang V

Anhang V

1. Einleitung

1.1 Problemstellung

Der Mensch will sich immer weiterentwickeln, dazu gehört auch sein Wissensstand.

Vor ca. 30 Jahren konnte man sich nicht vorstellen, dass man sich so viel Wissen aneignen kann, ohne dabei die Haustür zu verlassen. Man musste immer in Bibliotheken, Seminaren und Tutorien gehen.

In der Vorlesung wird nur ein Basisstoff vorgetragen und Studenten können nicht immer den Dozenten Fragen stellen, deswegen gibt es seit ca. 10 Jahren an vielen Universitäten extra „Tutorien-Kurse“ für Studenten. Meist wird dort nichts neues gelernt und die Tutorien können auch nicht immer alle Fragen beantworten oder viele Studenten sind schon berufstätig und haben deshalb gar keine Zeit einen Tutor zu besuchen. Dazu kommt noch in einigen Fällen die Sprachbarriere der Studierenden. Internet und Google Suchmaschine bieten vielartige Möglichkeiten, um brauchbare und verwendbare Information zu finden, nur benötigt man dafür viel Zeit. Mobile Endgeräte entwickeln sich und bieten weitere Möglichkeiten mit einer App Zeit und Sprachbarrieren zu reduzieren.

1.2 Motivation

Mit der Entwicklung der Technologie müssen traditionelle Lehrmethoden überarbeitet werden. Dieser Trend verursacht eine gemischte Reaktion. Einerseits gibt es Bedenken, dass neue Technologien Lehrer in der Regel ersetzen werden. Andererseits wird die Technologie nur bestehende Probleme beim Lernen lösen.

In den letzten Jahren hat sich gezeigt, dass moderne Geräte als Unterhaltungsgeräte bezeichnet werden. Ihnen stehen vermeintlich ernstere gewohnheitsmäßige Lehrmethoden gegenüber. Tatsächlich sind digitale Geräte längst zur alltäglichen Realität geworden. Darüber hinaus sind sie für die jüngere Generation bekannter und verständlicher als die Lehrmaterialien, die ältere Menschen gewohnt sind. Durch den Einsatz von Tablets und Smartphones sowie Lernspielen wird der Lernprozess sogar visueller.

1.3 Forschungskonzept

Die folgenden Fragen sollen beantwortet werden:

- Inwiefern lässt sich eine mobile Tutor App als ein Tutor verwenden?
- Kann eine mobile Tutor App einen realen Tutor oder Tutorin ersetzen?
- Ist Programmiersprache Java immer noch gut für Android Entwicklung anzuwenden?

Um den Fragen nachzuzeichnen, wird die App „Teach Me“ an der Hochschule Worms, Universität Heidelberg und Universität Mainz getestet. Hierbei handelt es sich um eine Live App zu testen und einen Fragebogen mit Feedback auszuführen. Der Live Test wird mit Hilfe von folgenden vier Android Geräten durchgeführt:

- Samsung Galaxy Note 8
- Samsung Galaxy S7
- Lenovo Tab 10
- Huawei P

Im Rahmen der Masterarbeit werden die folgenden Methodik-Schritte durchgeführt:

1. Eine Analyse von Fragebögen und Feedback wird durchgeführt, um zu untersuchen, wie sich die App durch das Lernen helfen kann.
2. Wie wurde die App mit Benutzer kommuniziert und wie wurde auf die „Feedback“ von dem Benutzer aufgegriffen?

1.4 Zielsetzung und Erkenntnisinteresse

Das Ziel dieser Masterarbeit ist es, eine Konzeption und eine Implementierung einer nativen Android App „Teach Me“ für die Informatik Studierenden an vielen Universitäten, die als ein Tutor dienen soll, zu entwickeln. In der App kann man verschiedene Kurse auswählen und die dazugehörige Information lesen. Mit einem Quiz und Q&A in der App können Studierende ihr Lernwissen überprüfen.

Dazu kommen folgenden Aufgaben der Arbeit:

- Eine Analyse des Android Betriebssystems durchzuführen.
- Eine Analyse von bereits bestehenden Entwicklungssprachen für Android Betriebssysteme durchzuführen.
- Eine Konzeption und Implementierung einer nativen Android App „Teach Me“.

- Ein Anwendungsvergleich von „Teach Me“ und einem „Real-Tutor“ durch Informatik Studierende an die Ruprecht-Karls-Universität Heidelberg, Johannes-Gutenberg-Universität Mainz und Hochschule Worms durchzuführen.

1.5 Forschungsstand

Durch die allgegenwärtige Verbreitung mobiler Geräte interagieren Menschen unterschiedlich mit Inhalten und der Welt. Durch die steigende Produktivität von Smartphones, Smartwatches und Tablets ermöglicht das mobile Lernen den Studenten den Zugriff auf Lernstoff von überall her, häufig von mehreren Geräten.

Google hat eine Reihe von Strategien für mobile Anwendungen implementiert, die das Wachstum der Branche vorantreiben. Die Reihenfolge, in der Suchergebnisse angezeigt werden, hängt davon ab, ob die Website für mobile Geräte angepasst ist. Google gab außerdem seine Absicht bekannt, seinen Suchindex in den Hauptindex für mobile Geräte und PCs zu unterteilen. [1]

Eine Studie von McGraw-Hill Education und Hannover Research unter mehr als 2.600 Studenten zeigte, dass etwa zwei Drittel der Befragten ihr Smartphone für Bildungszwecke nutzen. [2]

Ein Experiment an der Middlesex University, bei dem mobiles Lernen im Anatomieunterricht bei Erststudierenden eingesetzt wird. Die Studenten der Versuchsgruppe verwendeten eine spezielle Anwendung, die die Struktur der Muskeln und des Skeletts in 3D auf iPad-Geräten anzeigt. Ein Quiz und Spielelemente wurden in die App integriert, um die Aufmerksamkeit zu erhalten. Die Studenten fanden das neue Format „lustig“ und sagten, dass sie es mehr mögen als traditionelle Vorträge. Studenten, die das iPad verwenden, erhielten im Durchschnitt auch höhere Noten. [3]

2. Grundlagen zur Entwicklung einer Android Applications

Android bietet Entwicklern viele Möglichkeiten: es ist eine Universelle, offene Plattform, die von Millionen von Nutzern auf der ganzen Welt verwendet wird. Es gibt viele Tools für Android-Entwickler, die Ihnen helfen, schnell zu beginnen. Aber noch besser ist, dass jedes Jahr die Tools immer größer werden und Ihre Effizienz ständig steigt.

2.1 Integrierte Entwicklungsumgebung

Die Integrierte Entwicklungsumgebung ist eine integrierte Software-Entwicklungsumgebung.
Diese Software ermöglicht die Arbeit des Programmierers bequemer und produktiver zu machen.

IDE-Anforderungen:

- Sprachsyntax-Hervorhebung und Zeilennummerierung.
- Funktion zum Beenden des Codeschreibens und anzeigen von Parametern.
- Debuggen der Anwendung.
- Die Möglichkeit der Integration mit dem System der Kontrolle der Versionen des Codes.

IDE ist der Ort, an dem der Entwickler, die meiste Zeit verbringt, so dass die Wahl richtig gemacht werden muss.

2.1.1 Android Studio

Android Studio ist die offizielle IDE für Android von Google erstellt. Deshalb ist Android Studio die Nummer eins für Entwickler, die Apps erstellen möchten, nach Googles Materialdesign und Zugriff auf erweiterte Plattformfunktionen.

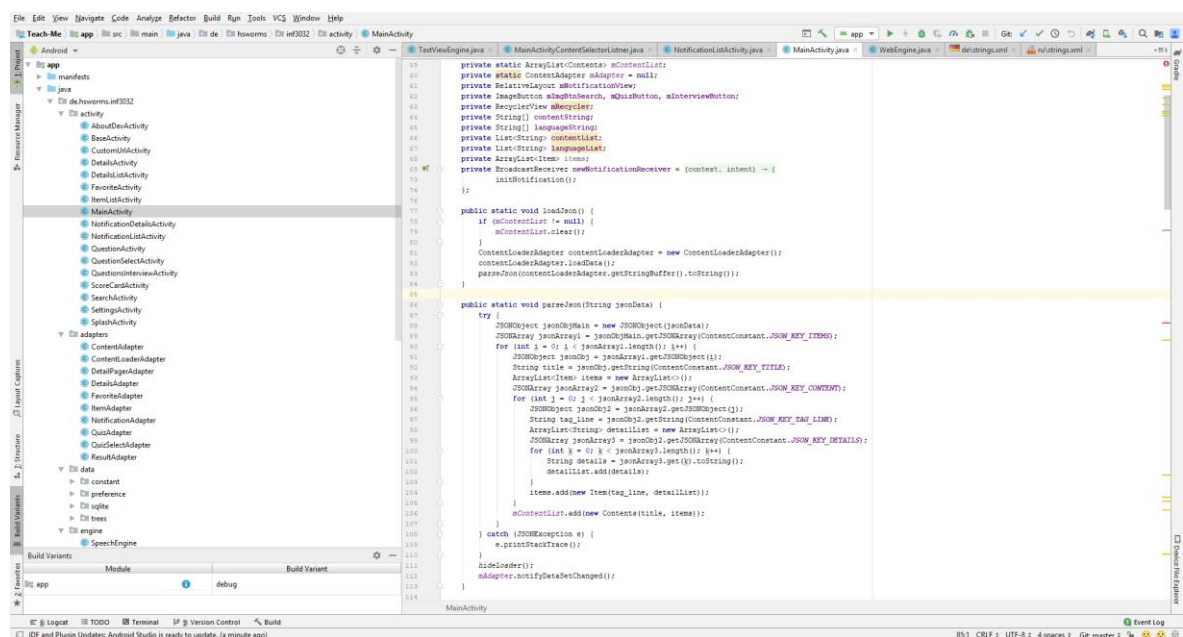


Abbildung 5: Android Studio 3.4.1 mit Geöffneter Projekt „Teach Me“

Android Studio fungiert als Editor für die von Ihnen gewählte Programmiersprache. Es unterstützt Java, C++ sowie Kotlin. Android Studio fungiert auch als Compiler, der APK-Dateien und Dateisysteme erstellen kann, um Ihr Projekt zu organisieren. Darüber hinaus enthält es einen XML-Editor und einen erweiterten Layout-Editor. Android Studio bietet eine ganze Reihe von zusätzlichen Tools. Glücklicherweise können die meisten jetzt ein einzelnes Paket herunterladen. Im Wesentlichen kommt dieses Paket mit dem Android SDK, aber Java JDK muss immer noch separat heruntergeladen und installiert werden.

AVD Manager

Das AVD Manager-Tool kommt mit Android Studio. Die Abkürzung AVD steht für Android Virtual Device. Es ist ein Emulator, um Android-Anwendungen auf Computer auszuführen. Es ist ein sehr nützliches Tool, mit dem Programmierer Ihre Anwendungen testen können, ohne sie auf physischen Geräten installieren zu müssen. Noch wichtiger ist, dass mit AVD Manager viele Emulatoren mit unterschiedlichen Bildschirmgrößen, Spezifikationen und Android-Versionen erstellt werden können. Sie können sehen, wie Ihre Kreation auf jedem Gerät aussehen wird, und damit Unterstützung unter den beliebtesten Gadgets bieten. Die Leistung des Werkzeugs wird ständig verbessert.

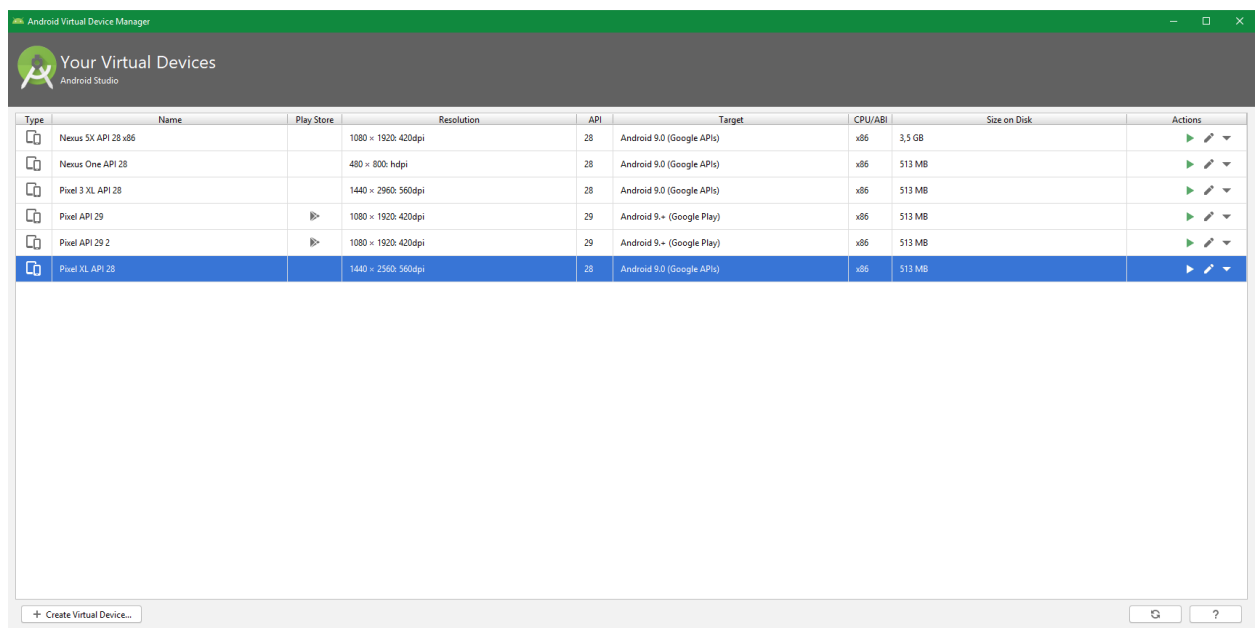


Abbildung \$: ADV Manager

Android Device Monitor

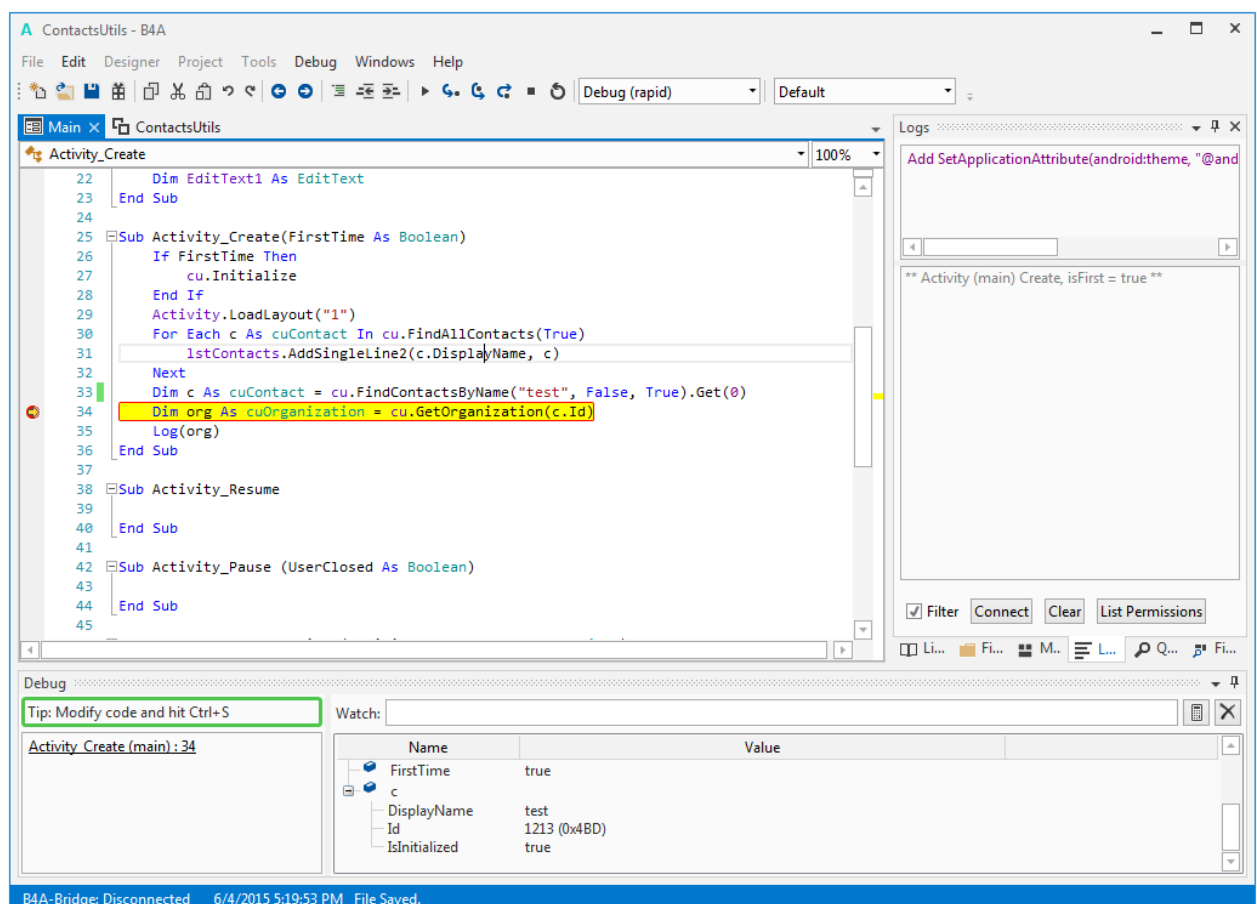
Ein weiteres integriertes Tool ist Android Device Monitor, mit dem Sie Ihr physisches oder virtuelles Gerät überwachen können, während es läuft. Über ADM können Sie Informationen darüber erhalten, wie viele Prozesse im Stream ausgeführt werden, Netzwerkstatistiken, LogCat usw. Dieses Tool eignet sich hervorragend zum Testen der Anwendungsleistung.

Android Debug Bridge

Android Debug Bridge ist ein Befehlszeilentool, mit dem Sie Dateien auf und von Ihrem Gerät kopieren, Apps installieren und deinstallieren, auf allen Android-basierten Geräten sichern und wiederherstellen können, einschließlich virtueller Geräte.

2.1.2 Basic for Android

Basic for Android ist ein wenig bekanntes Tool für die Entwicklung von Android-Anwendungen von Anywhere Software, spezialisiert auf das Konzept der schnellen Anwendungsentwicklung. B4A ist eine IDE und Interpreter, mit dem Entwickler die Anwendungen mit der Programmiersprache Basic erstellen können. Basic ist eine prozedurale Programmiersprache, die fast wie normales Englisch gelesen wird. Obwohl B4A ein Projekt ist, das nur von einer Person entwickelt wurde, ist es ihm gelungen, viele nützliche erweiterte Funktionen wie drahtloses Debuggen über Bluetooth, einen visuellen Editor zum Hinzufügen und Organisieren von Ansichten usw. zu verbinden. Basic for Android ist nicht kostenlos.

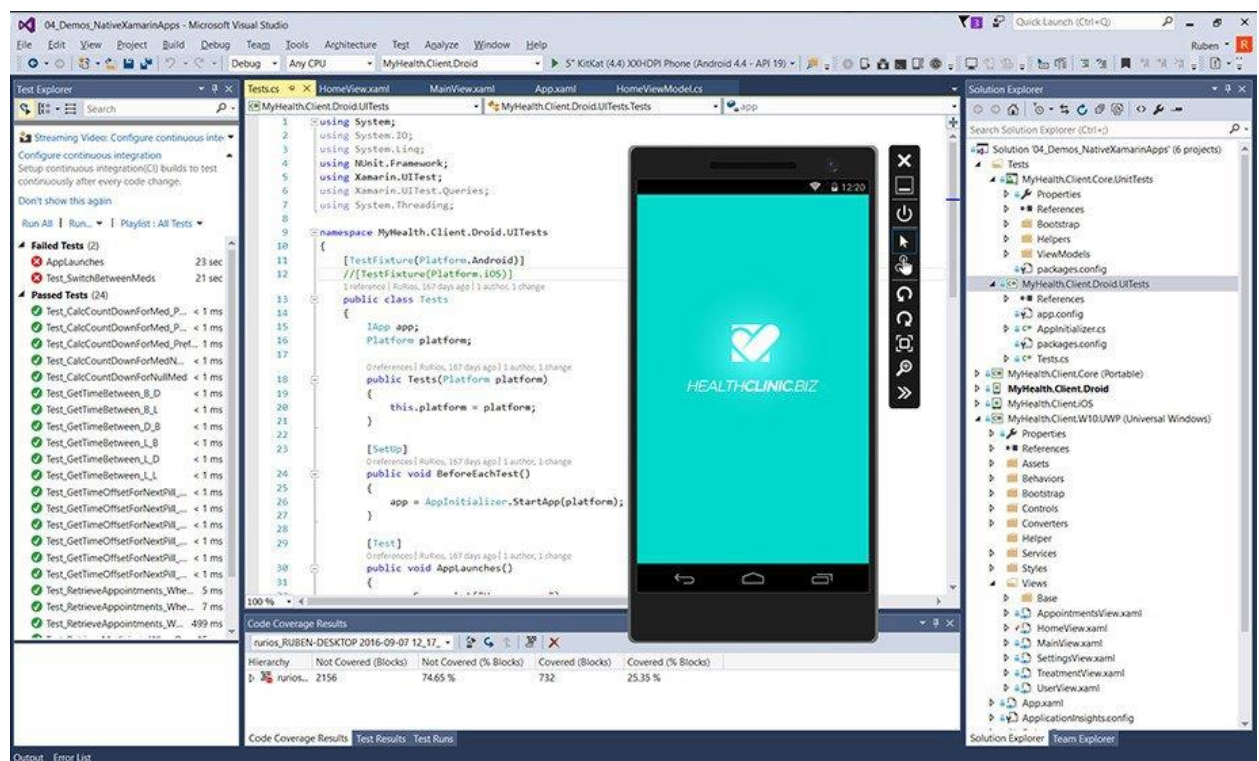


Mit B4A können Sie fast alles tun, was Sie mit Java tun, aber viel schneller und mit weniger Vorlagen. Jede Leistungsminderung im Vergleich zu anderen IDEs ist minimal. Sie sollten jedoch immer noch die offizielle Methode zum erstellen von Anwendungen erlernen,

insbesondere wenn Sie bestimmte Java-Bibliotheken verwenden möchten, die für Basic for Android eingewickelt werden können.

2.1.3 Visual Studio

Visual Studio ist Microsofts IDE, die eine Reihe von Entwicklungssprachen unterstützt, einschließlich C#. VB.net, JavaScript und mehr. Mit dem In Visual Studio enthaltenen Xamarin-Framework können Entwickler plattformübergreifende Anwendungen mit C# erstellen und dann auf mehreren Geräten testen, die mit der Cloud verbunden sind. Es ist eine gute und Kostenlose Wahl, wenn Sie Vorhaben, die App sowohl für Android als auch für IOS zu veröffentlichen, aber nicht den Wunsch haben, Ihren Code zweimal zu schreiben. Es ist auch eine gute Wahl für diejenigen, die bereits mit C# und/oder Visual Studio vertraut sind. Der Nachteil ist, dass Xamarin bei der Verwendung von Java-Bibliotheken unbequem ist und wie bei jeder anderen Android Studio-Alternative die Unterstützung von Google und die erweiterten integrierten Funktionen verloren gehen.



2.1.4 AIDE

AIDE steht für Android IDE und ist insofern einzigartig, als es auf Android selbst funktioniert. Das bedeutet, dass Sie Apps mit Ihrem Telefon oder Tablet erstellen und Sie dann auf demselben Gerät testen können. IDE funktioniert sehr gut mit Samsung DeX.



AIDE fehlt einige Funktionen aus Android Studio und es hat keinen wirklichen Vorteil gegenüber funktionelleren IDEs für die Entwicklung unter Android. Es ist nur für das Lernen von Java und Android SDK geeignet, da es die Möglichkeit bietet, ein programmierlehrbuch zu Lesen und gleichzeitig den Code von dort in Echtzeit zu überprüfen. Dies ist ein echter Glücksfall für diejenigen, die gerade erst begonnen haben, Ihre eigenen mobilen Projekte zu entwickeln. AIDE ist nicht kostenlos und unabhängig davon, ob die öffentliche Meinung gut ist oder nicht — es lohnt sich, es selbst zu kaufen und zu versuchen.

2.2 Mobile Applikationen

Der Entwickler kann aus drei Optionen für Mobile Anwendungen wählen:

- Native App
- WebApp
- Hybride App

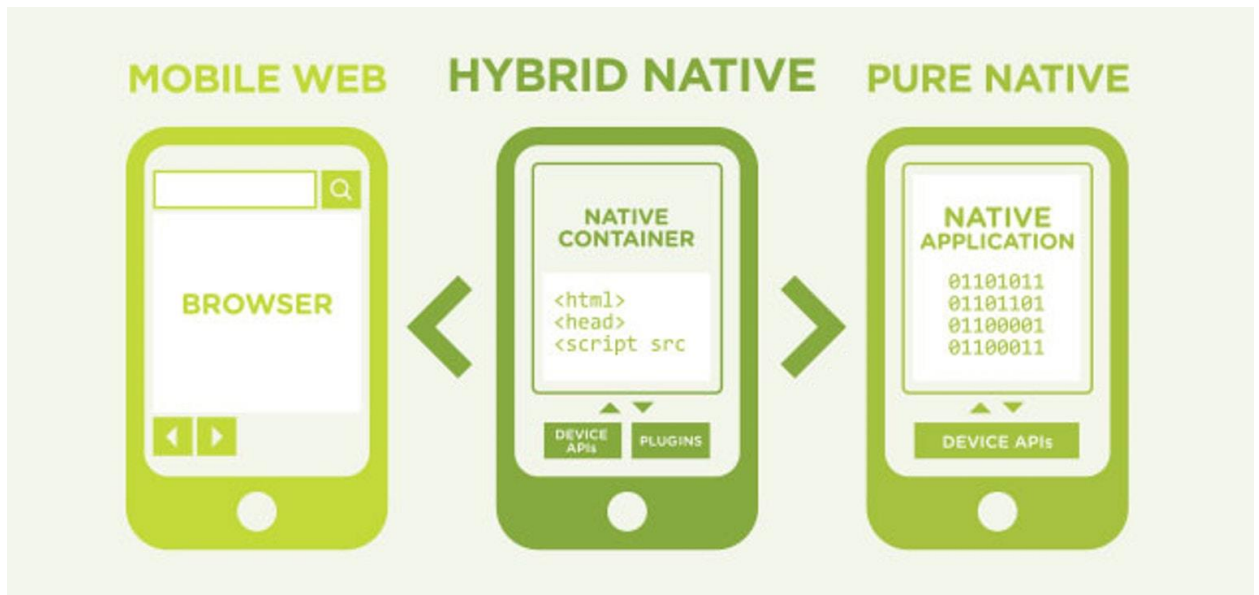


Abbildung \$: Native- Web und Hybride App

Die Auswahl eines geeigneten mobilen App-Modells ist ein sehr wichtiger Schritt in der Entwicklung, der von mehreren Faktoren beeinflusst wird, wie Z. B.:

- der technischen Bewertung der Entwickler
- die Notwendigkeit, auf Informationen auf dem Gerät zuzugreifen;
- die Auswirkungen der Internetgeschwindigkeit auf die App;
- eine oder viele Plattform-App

Als Nächstes wird im Kapitel jede Art von Anwendung analysiert, die vor- und Nachteile untersucht. Es wird bestimmt, was ist die beste Wahl in der einen oder anderen Situation ist, wenn eine Mobile App erstellt wird.

2.2.1 Native App

Unter Native App wird eine Mobile Anwendung gemeint, die für eine bestimmte Plattform erstellt und direkt auf dem Gerät des Benutzers installiert wird. Solche Apps lädt der Benutzer über den App Store einer Plattform wie dem Google Play Store für Android und dem Apple App Store für iOS herunter.

Mit nativen Anwendungen können Unternehmen die Anwendung nach individuellen Anforderungen herstellen, so dass der Benutzer Sie zusätzlich zu der Website oder einem anderen Kanal, den er bereits verwendet hat, bequem nutzen kann. Diese Integrität ist ein wesentlicher Vorteil von nativen Anwendungen.

Einige andere wichtige Vorteile von nativen Anwendungen:

- Mit der Geolocation können Unternehmen Ihre Treueprogramme oder Promotionen anpassen. Verbraucher können benachrichtigt werden, wenn Sie in der Nähe von physischen Geschäften sind, oder haben die Möglichkeit, einen regionalen Rabatt zu erhalten.
- Diese Aktivitäten oder Inaktivität des Benutzers können leicht gesammelt und analysiert werden, wodurch die Wirksamkeit der gesamten Anwendung oder Ihrer einzelnen Funktionen bewertet werden kann.
- Native Anwendungen neigen dazu, besser zu funktionieren. Web-Anwendungen werden manchmal erstellt, um native zu simulieren, aber Sie sind auf Internetgeschwindigkeit und Designfähigkeiten beschränkt.

Und mögliche Nachteile:

- Native Anwendungen sind oft teurer in der Entwicklung, insbesondere für Unternehmen, die Anwendungen auf plattformübergreifenden Betriebssystemen benötigen
- Native Apps müssen von jedem App Store genehmigt werden, und der Prozess, auf den Benutzer aufmerksam gemacht werden, kann schwierig sein.

2.2.2 Web App

Web App Apps arbeiten über einen Webbrowser auf dem Gerät des Benutzers. Diese Apps sind im wesentlichen individualisierte Websites, die so gestaltet sind, dass Sie wie native Apps Aussehen und verwendet werden, aber Sie befinden sich nicht wirklich auf dem Gerät des Benutzers. Sie können mit Cloud-Speicher verglichen werden, verglichen mit Daten, die auf der Festplatte Ihres Computers gespeichert sind. Mit einer guten, qualitativ hochwertigen Entwicklung, die Größenanpassung und scrollen beinhaltet, funktionieren Webanwendungen oft ähnlich wie native Anwendungen.

Hier sind einige der wichtigsten Vorteile von Webanwendungen:

- Webbasierte Anwendungen werden leichter unterstützt und können auf der Plattform mit jedem Betriebssystem funktionieren
- Entwickler können Apps anbieten, ohne dass Sie von App Stores genehmigt werden müssen

- Schnellere Entwicklung von Schleifen mit CSS, HTML und JavaScript

Und ein paar Nachteile:

- Webanwendungen haben keinen Zugriff auf das Gerät des Benutzers.
- Benutzer müssen Sie über das Netzwerk verwenden, was die Sicherheitskontrolle erheblich reduziert
- Die Suche nach einer App kann schwierig sein.

2.2.3 Hybride App

Hybridanwendungen sind etwas zwischen nativen und Webanwendungen. Sie werden tatsächlich so erstellt, dass Sie wie native Anwendungen Aussehen und verwendet werden. Sie werden auch auf dem Telefon des Benutzers installiert und können in App Stores gefunden werden. Der Unterschied besteht darin, dass Sie unbedingt innerhalb einer nativen Anwendung gehostet und erstellt werden müssen, um über WebView zu arbeiten, und auf diese Weise können Sie auf die Informationen auf dem Gerät des Benutzers für größere Funktionen zugreifen.

Zusätzliche Vorteile von Hybrid-Anwendungen:

- Hybrid-Anwendungen bieten die beste Funktionalität und Personalisierung für den Benutzer
- Entwickler sind nicht auf eine einzige Plattform beschränkt, sondern können eine hybride Anwendung erstellen, die mit mehreren Plattformen funktioniert
- Hybride sind eine gute Option für Entwickler, die visuell gesättigte Anwendungen wie Spiele erstellen

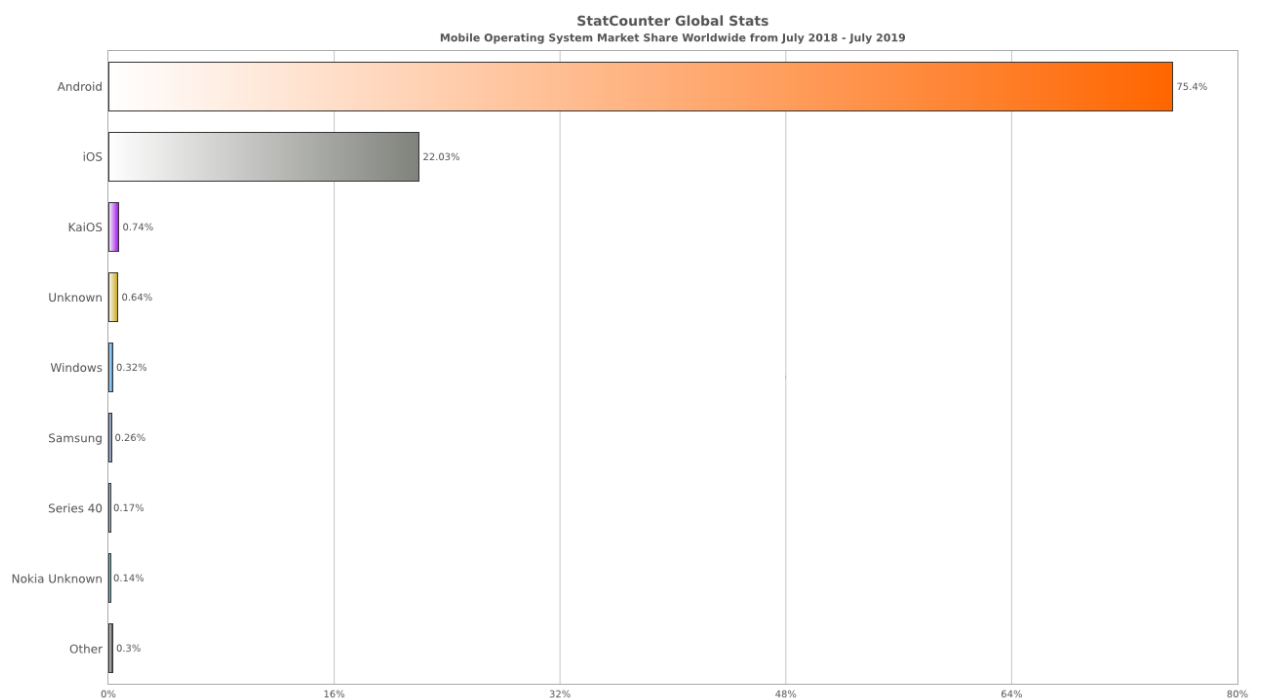
In jedem Fall gibt es einige Nachteile, die bei der Auswahl einer Hybrid-App in Betracht gezogen werden sollten:

- Zu komplexe Anwendungen sind am besten nativ
- Die Entwicklung erfordert zusätzliche Zeit und Mühe, damit eine solche Anwendung vom Benutzer als nativ aussieht und sich anfühlt
- App Stores können Hybridanwendungen ablehnen, die nicht reibungslos genug funktionieren

2.3 Android Betriebssystem

Android ist Betriebssystem für Smartphones, Internet-Tablets, E-books, digitale Player, Armbanduhren, Spielekonsolen, Netbooks, Smartbooks, Google-Brillen, Fernseher und andere Geräte. Basierend auf dem Linux-Kernel und Googles eigener Java Virtual machine-Implementierung. Ursprünglich entwickelt von Android, Inc., die dann von Google gekauft wurde. Anschließend hat Google die Gründung der Open Handset Alliance initiiert, die sich jetzt mit der Unterstützung und Weiterentwicklung der Plattform beschäftigt. Mit Android könnten Java-Anwendungen erstellt, die das Gerät über die von Google entwickelten Bibliotheken steuern. Mit dem Android Native Development Kit können Bibliotheken und Anwendungskomponenten portieren, die in C und anderen Sprachen geschrieben sind.

Anfang 2019 gelang es Forschungen aus verschiedenen Quellen herauszufinden, dass der durchschnittliche Anteil von Android-Smartphones auf dem Smartphone-Markt 75,4% der weltweit verkauften Smartphones überstieg. Insgesamt wurden im Jahr 2019 mehr als 344 Millionen Geräte verkauft.



2.3.1 Anwendungskomponenten

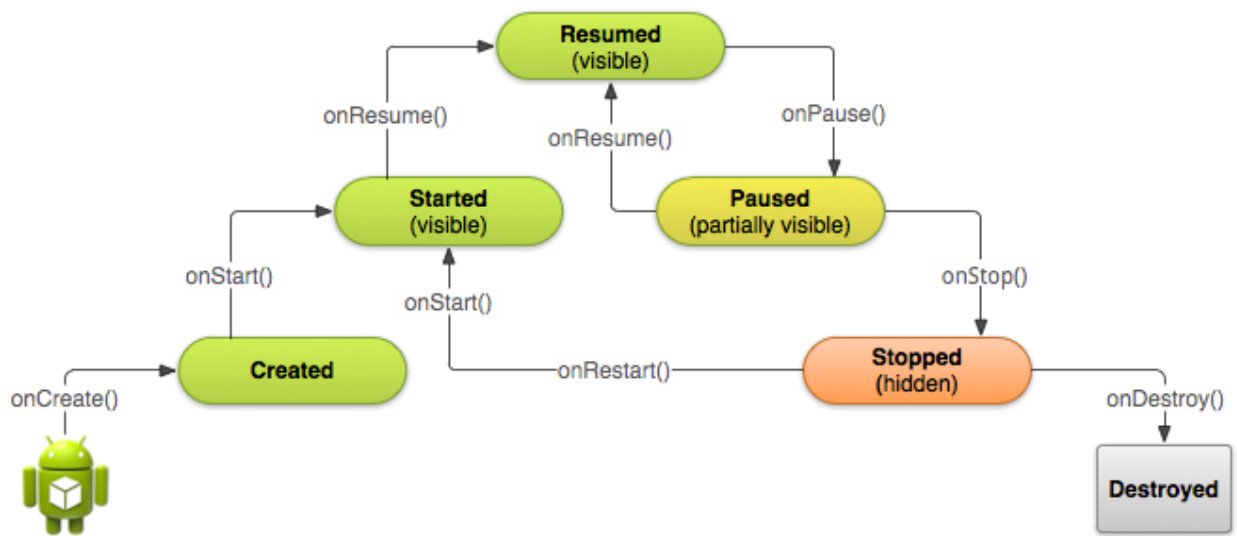
Android-Apps bestehen aus den folgenden teilen:

- **Activity** ist ein Ansichtsschema für Android-Apps. Zum Beispiel der Bildschirm, den der Benutzer sieht. Die Android-App kann mehrere Aktivierungen haben und kann während der Ausführung der App zwischen Ihnen wechseln.
- **Views** Benutzeroberfläche aktiviert, die von Klassen-Widget erstellt wird.
- **Services** führt Hintergrundaufgaben aus, ohne eine Benutzeroberfläche bereitzustellen. Sie können den Benutzer über das Android-Benachrichtigungssystem Benachrichtigen.
- **Content Provider** stellt Daten für Anwendungen bereit. Die Anwendung kann Daten mit anderen Anwendungen teilen. Android enthält eine SQLite-Datenbank, die ein content-Provider sein kann
- **Intents** sind asynchrone Nachrichten, die es einer Anwendung ermöglichen, Funktionen von anderen Diensten abzufragen oder zu aktivieren. Die App kann direkte Intents zu einem Dienst oder Activity machen oder von Android nach registrierten Intent Diensten und Apps Fragen. Zum Beispiel kann die Anwendung über Intent einen Kontakt aus der Kontaktanwendung des Geräts anfordern. Die Anwendung registriert sich selbst im Internet über Intent Filter.
- **Broadcast Receiver** akzeptiert Systemnachrichten und implizite Intents, kann verwendet werden, um auf eine Änderung des Systemstatus zu reagieren. Die Anwendung kann sich als Empfänger bestimmter Ereignisse registrieren und kann gestartet werden, wenn ein solches Ereignis eintritt.

Der Lebenszyklus einer Anwendung in Android wird vom System streng überwacht und hängt von den Bedürfnissen des Benutzers, den verfügbaren Ressourcen ab. Die Entscheidung, die Anwendung zu starten, trifft das System. Das System unterliegt bestimmten angegebenen und logischen Regeln, mit denen Sie bestimmen können, ob eine Anwendung heruntergeladen, angehalten oder beendet werden kann. Wenn der Benutzer derzeit mit einem bestimmten Fenster arbeitet, hat das System der entsprechenden Anwendung Priorität. Umgekehrt, wenn das Fenster unsichtbar ist und das System entscheidet, dass die Anwendung beendet werden muss, um zusätzliche Ressourcen zu mobilisieren, wird die Anwendung mit niedrigerer Priorität beendet. In Android sind die Ressourcen begrenzter, so dass Android die Funktionsweise von Apps strenger überwacht.

Grundlegende Methoden des Anwendungslebenszyklus:

- `protected void onCreate()`
- `protected void onStart()`
- `protected void onRestart()`
- `protected void onResume()`
- `protected void onPause()`
- `protected void onStop()`
- `protected void onDestroy();`



#ToDo

2.3.2 Pattern

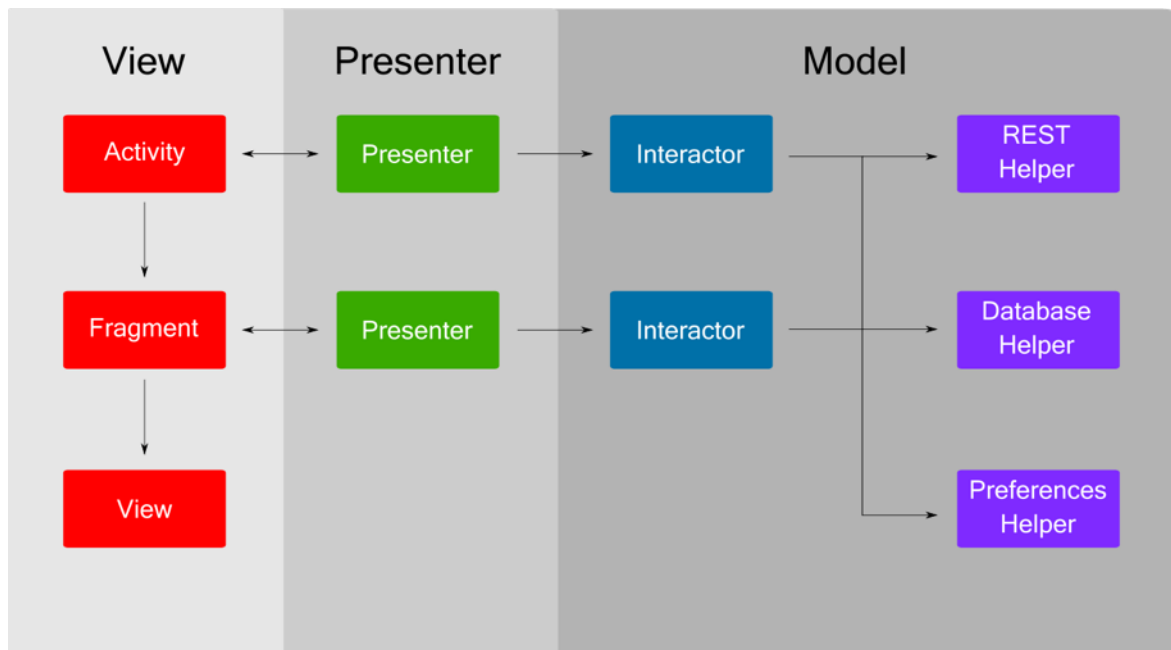
Bei der Entwicklung komplexer Anwendungen können Probleme auftreten, die wahrscheinlich vorher aufgetreten sind und bereits eine große Anzahl von Lösungen haben. Solche Lösungen werden als Muster bezeichnet. In der Regel sprechen Sie über Designmuster und Architekturmuster. Sie vereinfachen die Entwicklung von Anwendungen, daher ist es ratsam, sie zu verwenden, wenn möglich.

Приложение, Teach Me, было разработано, используя паттерн архитектуры Model View Presenter.

Model View Presenter

Model View Presenter ist ein Entwicklungsmuster für das Android-Betriebssystem, das vorschlägt, die Anwendung in die folgenden drei Teile aufzuteilen:

- Model ist ein Wrapper für die empfangenen Daten. Dabei gibt es einen besonderen Unterschied, wo die Daten nicht sein sollten-die Daten der Netzwerkanforderungen oder die Daten der Interaktion des Benutzers mit der Benutzeroberfläche. Ein guter Ort, um Caches zu implementieren. Es ist eine gute Praxis, für jede Antwort des Servers ein einzigartiges Modell zu erstellen, um Schnittpunkte und nachfolgende Probleme bei api-änderungen zu reduzieren. [8-10]
- Presenter verbindet zwischen der Verarbeitung von Daten, die von Model und dem Aufruf von Methoden von View abgerufen werden, und implementiert so die Reaktion der ui-Komponenten auf die Daten. Presenter-Methoden werden von activity/fragment-lebenszyklusmethoden aufgerufen und sind oft symmetrisch. [8-10]
- View-zeigt die empfangenen Daten aus Presenter an. In der richtigen Implementierung hat das View-Objekt keine Ahnung von den empfangenen Daten von außen, es sollte nur anzeigen was Presenter von ihm verlangt. View kann jede Aktivität oder Fragment in der App unter Android OS sein. [8-10]



Model View Presenter sollte Schnittstellen für mehr Flexibilität bei der Codeänderung darstellen.

MVP hat eine Reihe von Vorteilen gegenüber dem Standard App Entwicklung Schema, eines davon ist eine gute Testabdeckung, die sowohl für die Implementierung der App als auch für die Flexibilität wichtig ist. Die Anwendung weiß nicht, was passiert, bevor das Element auf dem Bildschirm reflektiert wird, die gesamte Logik ist von der Ansicht geschlossen. Zwei Punkte sind für die Schaffung einer guten Architektur berücksichtigt. Da MVP die Logik und die Anzeige von Elementen teilen, und wenn die Anwendung eine Client - Server-Struktur teilt und Anforderungen an den Server. Mit MVP ist es auch möglich, einen ziemlich einfachen und verständlichen Code zu erreichen, dies ist der Dritte Punkt einer guten Anwendungsarchitektur.

2.4 Entwicklung Sprachen für Android Betriebssystem

2.4.1 Java

Die offizielle Programmiersprache, die von der Android Studio-Entwicklungsumgebung unterstützt wird. Laut einer jährlichen Umfrage der Stackoverflow ist Java 2018 in die fünf beliebtesten Programmiersprachen eingestiegen.

Java bezieht sich auf die meisten offiziellen Google-Dokumentation, und es ist nicht schwer, bezahlte und Kostenlose Bibliotheken und Handbücher zu finden-es gibt viele von Ihnen.

Leider verhindert die Komplexität von Java, dass jeder daran programmiert. Als objektorientierte Programmiersprache hat es eine Reihe von Funktionen in Form von Klassenkonstruktoren, Ausnahmen, die zu fallenden Anwendungen, während der Arbeit und anderen Momenten führen, die bei der Entwicklung immer berücksichtigt werden müssen. Der Code in Java ist jedoch leicht zu Lesen und zu strukturieren, insbesondere wenn die akzeptierten Standards für die Gestaltung eingehalten werden.

Bei der Entwicklung in Java unter Android werden nicht nur Java-Klassen verwendet, die Code enthalten, sondern auch XML-Manifestdateien, die dem System grundlegende Informationen über das Programm liefern, und Gradle, Maven oder Ant Build-Systeme, die Befehle in Groovy, POM und XML-Sprachen schreiben. Standardmäßig werden in Projekten Gradle verwendet, und in den Anfangsphasen der Entwicklung in Java müssen die in Groovy geschriebenen Dateien praktisch nicht bearbeitet werden. Für das Layout des UI-Teils wird normalerweise auch die XML-Sprache verwendet.

2.4.2 Kotlin

Die Sprache Kotlin wurde offiziell im Mai 2017 auf Google I/O eingeführt und von Google als die zweite offizielle Programmiersprache unter Android nach Java positioniert, nur ein wenig einfacher zu verstehen. Java-Kenntnisse werden hier benötigt, um die Arbeitsprinzipien von Kotlin, die Allgemeine Struktur der Sprache und Ihre Besonderheiten zu verstehen. Viele Entwickler betrachten Kotlin als Wrapper über Java und empfehlen, es erst zu lernen, nachdem Sie sich mit Ihrem Java-Wissen vertraut gemacht haben.

Kotlin ist mit Java kompatibel und verursacht keine Leistungseinbußen und größere Dateigrößen. Der Unterschied zu Java ist, dass es weniger dienstlichen, sogenannten boilerplate-Code benötigt, so dass es stromlinienförmiger und leichter zu lesen ist. Seine Schöpfer haben es geschafft, NullPointerException zu vermeiden und die Kompilierung wird wegen der kleinen Dinge wie dem vergessenen"; "-Zeichen nicht mehr unterbrochen.

2.4.4 Scriptsprachen

Lua

Lua ist eine alte Skriptsprache, die ursprünglich als add-on für Programme in komplexeren Sprachen erstellt wurde. In dieser Sprache gibt es einige Merkmale, die Lua von einer Reihe von ähnlichen unterscheiden. Zum Beispiel der Beginn von Arrays mit 1 statt 0, oder das Fehlen von nativen Klassen.

Daher kann Lua für bestimmte Aufgaben als primäre Programmiersprache verwendet werden. Das beste Beispiel dafür ist das Corona SDK. Mit Corona können Sie leistungsstarke, funktionsreiche Anwendungen erstellen, die auf Windows, Mac, Android, iOS und sogar Apple TV und Android TV bereitgestellt werden können. Corona bietet auch Monetarisierungsmöglichkeiten, und es ist ein anständiger Markt, in dem Sie nützliche Plugins finden können.

HTML 5 + CSS + JavaScript

Diese drei Sprachen, die einst für die Entwicklung von Front-End-Anwendungen in einer Webumgebung entwickelt wurden, haben sich seitdem zu etwas größerem entwickelt. HTML 5, CSS und JavaScript-Tools reichen nun aus, um eine Vielzahl von Anwendungen für Mobile Geräte und für klassische PCs zu erstellen. Im Wesentlichen erstellt der Programmierer eine Webanwendung, die die ganze Macht und Magie von Offline-Plattformen nutzen kann.

Programmierer können auf diese Weise Android-Anwendungen erstellen, indem Sie die Funktionen von Adobe Cordova verwenden. Es ist ein Open-Source-Framework, das auch iOS-Betriebssysteme unterstützt, Windows 10 Mobile, Blackberry, Firefox, und viele andere. Was auch immer Cordova nützlich ist, es erfordert eine ernsthafte Arbeit, um eine anständige Anwendung darin zu erstellen. Daher bevorzugen viele Programmierer das Ionic Framework-Projekt.

Es gibt eine andere Möglichkeit: die Verwendung der React Native Bibliothek. Es kann auf Android, iOS bereitgestellt werden. Facebook, Instagram und andere große Unternehmen nutzen diese Bibliothek, so dass Sie sich auf Ihre Zuverlässigkeit verlassen können.

2.5 Git Versionierung

Git ist eine Art Version Control System. Version Control System ist ein Programm, um mit ständig wechselnden Informationen zu arbeiten. VCS kann viele Versionen derselben Datei speichern und zu einem früheren Zustand zurückkehren.

Diplomarbeit nutzt Hosting-Service von Github.com und das Projekt Teach Me ist unter:

<https://github.com/AuroraSyN/Teach-Me> verfügbar

Github.com es ist ein Web-Service von Projekten mit dem Git-Versionskontrollsystem sowie als soziales Netzwerk für Entwickler. Benutzer können eine unbegrenzte Anzahl von Repositories erstellen, von denen jedes ein Wiki, ein Issue-tracking-System, eine Möglichkeit zur Durchführung von Code Review und vieles mehr bietet.

3. Rolle von Tutorien in Studienprozess

3.1 Sinn und Zweck von Tutoren an Universitäten

Ein Tutorium ist eine unterstützende Veranstaltung. Hier können Studenten gemeinsam mit einem Tutor den Stoff einer Lehrveranstaltung wiederholen und vertiefen. Ein Tutorium ist in den meisten Fällen freiwillig, kann aber von einem Dozenten vorgeschrieben werden. Wird eine solche Veranstaltung angeboten, sollte man die Chance ergreifen und diese besuchen. Oft dienen Tutorien auch dazu, gemeinsam Hausaufgaben zu erarbeiten, während der Tutor den Anwesenden hilft und ihnen mit Rat und Tat zur Seite steht.

Ein Tutor ist meist selbst Student in einem höheren Fachsemester. Er unterstützt Dozenten als studentische Hilfskraft und erleichtert anderen Studenten mit seinem Wissen den Start ins Studium.

3.2 Arten von Tutorien

Tutoren können in einer Vielzahl von Tutorien tätig sein, die jeweils unterschiedliche Ziele verfolgen. Dabei kann es sich um persönliche Betreuung oder Gruppenbetreuung handeln. So findet sich in Großbritannien z.B. das Personal Tutoring System', bei dem jedem Studierenden ein Hochschullehrer zugewiesen wird, der ihm als Ansprechperson dienen soll. In Anbetracht der hohen Studierendenzahlen wird im universitären Umfeld in Deutschland eher auf die Betreuung von Gruppen gesetzt. Dabei sind vor allem studentische Tutoren im Einsatz. Im Fachbereich 05 der Universität Kassel sind vor allem zwei Arten von Tutorien zu finden: Orientierungstutorien und Fachtutorien.

3.2.1 Orientierungstutorien

Orientierungstutorien erstrecken sich meist über drei Tage vor Beginn des ersten Semesters und sollen dazu dienen, Studienanfängern eine soziale Orientierung an der Hochschule, im Studienfach und am Hochschulort zu geben, um somit den Studienanfang zu erleichtern. Diese können sich an die Studienanfänger generell, oder aber auch nur an bestimmte Zielgruppen richten. Hierbei steht unter anderem der Gedanke im Mittelpunkt, die Hochschule nicht nur als Lernort sondern auch als Lebensraum zu verstehen. Der Beginn des Studiums bringt viele neue Anforderungen mit sich. Neben den veränderten fachlichen Anforderungen stehen die Erwartungen an selbstverantwortliche Lernkompetenzen und zielorientiertes Arbeiten. Darüber hinaus ist für viele mit dem Eintritt in eine Hochschule zudem ein Wohnortswechsel verbunden. Einerseits muss der Studierende nun lernen, für sich und seine akademischen Leistungen selbst Verantwortung zu tragen, indem er sich z.B. selbst seinen Stundenplan organisiert und für die

Erstellung und Einreichung der Leistungsnachweise verantwortlich ist. Andererseits muss er auch lernen, außerhalb der Universität zu Recht zu kommen und sich an das selbstständige Leben in einer eigenen Wohnung in einer fremden Stadt zu gewöhnen. Der Fachbereich hat bis jetzt gute Erfahrungen mit Orientierungstutorien gemacht. Dort wurden Studienanfängern in größeren Gruppen neben der fachlichen Einweisung auch Informationen über die Stadt und das Leben als Student gegeben, um ihnen so die Umstellung zu erleichtern.

3.2.2 Fachtutorien

Fachtutorien werden in der Regel als Begleitung zu Kernveranstaltungen im ersten Studienjahr angeboten und dienen dazu, das fachliche Lernen im Rahmen einer kleineren Gruppe zu unterstützen. Bei Fachtutorien kann es sich um separate Tutorien handeln oder um integrierte Veranstaltungen, bei denen sich Lehrende, Tutor und Studierende zur selben Zeit im selben Raum befinden. Fachtutorien ermöglichen den Studenten zudem erforderliche Grundfähigkeiten zu erwerben, aufzubauen und zu trainieren: "Viele Lernangebote gehen von einer funktierten Lernselbstständigkeit der Studierenden aus, ohne ihnen beim Entwickeln zu helfen oder Lernziele auszuformulieren". Was genau in den Tutorien passieren soll, wird vom jeweiligen Dozenten bestimmt. Elementar ist dabei die Zusammenarbeit vom Dozenten und Tutor, um das Tutorium möglichst gut an die Veranstaltung anzupassen und dem Studierenden den größtmöglichen Lernerfolg zu garantieren. Durch Vorgaben des Dozenten können die Studierenden sich sicher sein, dass die Inhalte des Tutoriums mit den Anforderungen des Dozenten übereinstimmen. Durch Absprachen wird für die Teilnehmer zudem eher der Bezug zur Veranstaltung und somit auch der Sinn des Tutoriums sichtbar. Rückmeldungen der Tutoren können dann hilfreich sein, um dem Dozenten ein Bild über den Wissens- und Kompetenzstand seiner Studierenden zu verschaffen, und eventuell besonders problematische Themen noch einmal aufzugreifen.

3.3 Aufgaben der Tutoren und Tutorinnen

Neben der eigenen Lehrtätigkeit in einem Tutorium helfen Tutoren unter anderem bei der Vorbereitung von Veranstaltungen, unterstützen Professoren und betreuen ausländische Studenten.

#ToDo

4. Konzeption

4.1 Analyse bestehender Softwareprodukten

4.2 Funktionale Anforderungen

4.3 Nicht-funktionale Anforderungen

4.4 Mockup und Prototype


```

47 public WebEngine(WebView webView, Activity activity) {
48     this.webView = webView;
49     this.mActivity = activity;
50     this.mContext = mActivity.getApplicationContext();
51     mVideoViewer = VideoViewEngine.getInstance();
52 }
53
54 public void initWebView() {
55     webView.getSettings().setJavaScriptEnabled(true);
56     webView.getSettings().setLoadWithOverviewMode(true);
57     webView.getSettings().setAppCacheMaxSize(AppConstant.SITE_CACHE_SIZE);
58     webView.getSettings().setAppCachePath(mContext.getCacheDir().getAbsolutePath());
59     webView.getSettings().setAllowFileAccess(true);
60     webView.getSettings().setAppCacheEnabled(true);
61     webView.getSettings().setCacheMode(WebSettings.LOAD_DEFAULT);
62     webView.getSettings().setLoadWithOverviewMode(true);
63     webView.getSettings().setDomStorageEnabled(true);
64     webView.getSettings().setDefaultTextEncodingName("utf-8");
65     webView.getSettings().setPluginState(WebSettings.PluginState.ON);
66     if (!isNetworkAvailable(mContext)) {
67         webView.getSettings().setCacheMode(WebSettings.LOAD_CACHE_ELSE_NETWORK);
68     }
69     if (AppPreference.getInstance(mContext).getTextSize().equals("Small")) {
70         webView.getSettings().setTextSize(WebSettings.TextSize.SMALLER);
71     } else if (AppPreference.getInstance(mContext).getTextSize().equals("Normal")) {
72         webView.getSettings().setTextSize(WebSettings.TextSize.NORMAL);
73     } else if (AppPreference.getInstance(mContext).getTextSize().equals("Large")) {
74         webView.getSettings().setTextSize(WebSettings.TextSize.LARGER);
75     }
76 }

```

Abbildung \$: WebView.java aus Projekt „Teach Me“

5.2.2 VideoView

```

22 public static VideoViewEngine getInstance() {
23     if (videoViewer == null) {
24         videoViewer = new VideoViewEngine();
25     }
26     return videoViewer;
27 }
28 public void show(final Activity activity) {
29     dismiss();
30     AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(activity, R.style.DialogTheme);
31     LayoutInflater inflater = LayoutInflater.from(activity);
32     View promptsView = inflater.inflate(R.layout.layout_video_view, root: null);
33     alertDialogBuilder.setView(promptsView);
34     alertDialogBuilder.setCancelable(true);
35     videoLayout = promptsView.findViewById(R.id.videoView);
36     progressBar = promptsView.findViewById(R.id.progressBar);
37     dialog = alertDialogBuilder.create();
38     dialog.show();
39     WindowManager.LayoutParams lp = new WindowManager.LayoutParams();
40     Window window = dialog.getWindow();
41     lp.copyFrom(window.getAttributes());
42     lp.width = WindowManager.LayoutParams.MATCH_PARENT;
43     lp.height = WindowManager.LayoutParams.MATCH_PARENT;
44     window.setAttributes(lp);
45     dialog.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
46     View v = window.getDecorView();
47     v.setBackgroundResource(android.R.color.black);
48
49     if (Build.VERSION.SDK_INT < 19) {
50         v.setSystemUiVisibility(View.GONE);
51     } else if (Build.VERSION.SDK_INT >= 19) {
52         int uiOptions = View.SYSTEM_UI_FLAG_HIDE_NAVIGATION | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY;
53         v.setSystemUiVisibility(uiOptions);
54     }
55 }

```

Abbildung \$: VideoView.java aus Projekt „Teach Me“

5.2.3 TextView

```
34 private void init() {
35     Typeface regularFont = Typeface.createFromAsset(getContext().getAssets(), path: "fonts/times.ttf");
36     Typeface boldFont = Typeface.createFromAsset(getContext().getAssets(), path: "fonts/timesbd.ttf");
37
38     Typeface currentTypeFace = this.getTypeface();
39     if (currentTypeFace != null && currentTypeFace.getStyle() == Typeface.BOLD) {
40         this.setTypeface(boldFont);
41     } else {
42         this.setTypeface(regularFont);
43     }
44 }
```

Abbildung \$: TextView.java aus Projekt „Teach Me“

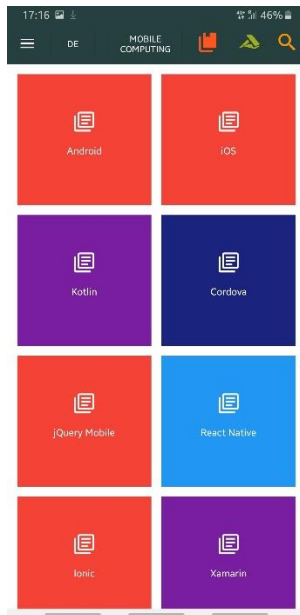
5.2.4 Speech

```
24 public Speech(Activity activity) {
25     mActivity = activity;
26     if (AppPreference.getLanguage().equals("English")) {
27         mLocale = new Locale(ContentConstant.TTS_LOCALE_EN);
28     } else if (AppPreference.getLanguage().equals("German")) {
29         mLocale = new Locale(ContentConstant.TTS_LOCALE_DE);
30     } else if (AppPreference.getLanguage().equals("Russian")) {
31         mLocale = new Locale(ContentConstant.TTS_LOCALE_RU);
32     } else {
33         mLocale = new Locale(ContentConstant.TTS_LOCALE_EN);
34     }
35 }
36
37 public void releaseEngine() {
38     if (mTextToSpeech != null) {
39         if (mTextToSpeech.isSpeaking()) {
40             mTextToSpeech.stop();
41         }
42         mTextToSpeech.shutdown();
43         mTextToSpeech = null;
44     }
45 }
46
47 public void startEngine(final String text) {
48     releaseEngine();
49     if (mTextToSpeech == null) {
50         mTextToSpeech = new TextToSpeech(mActivity.getApplicationContext(), onInit(status) -> {
51
52             boolean isStarted = isLangPacAvailable(status);
53             if (isStarted) {
54                 Toast.makeText(mActivity, mActivity.getApplicationContext().getResources().getString(R.string.tts_wait_msg), Toast.LENGTH_LONG).show();
55                 speak(text);
56             } else {
57                 invokePacInstaller();
58             }
59         });
60     }
61 }
62
63 }
64 }
```

Abbildung \$: SpeechView.java aus Projekt „Teach Me“

5.3 Activities

5.3.2 Main Activity



5.3.3 Search Activity

5.3.4 Question Activity

5.3.5 Question Select Activity

5.3.6 Question Interview Activity

5.3.7 Score Card Activity

5.3.8 Favorite Activity

5.3.8 Settings Activity

6. Anwendungsvergleich

6.1 Auswertungs- und Vergleich Dokumentation

6.2 App Anforderungen vergleich

6.2 Forschungsversuch an Ruprecht-Karls-Universität Heidelberg

6.2.1 Forschungsversuch am 3.06.2019

6.2.2 Forschungsversuch am 7.06.2019

6.2.3 Forschungsversuch am 17.06.2019

6.2.4 Forschungsversuch am 20.06.2019

6.2.5 App Evolution

6.3 Forschungsversuch an Johannes-Gutenberg-Universität Mainz

6.3.1 Forschungsversuch am 10.07.2019

6.3.2 Forschungsversuch am 12.07.2019

6.3.3 Forschungsversuch am 23.07.2019

6.3.4 Forschungsversuch am 25.07.2019

6.3.5 App Evolution

6.4 Forschungsversuch an der Hochschule Worms

6.4.1 Forschungsversuch am 08.08.2019

6.4.2 Forschungsversuch am 12.08.2019

6.4.5 App Evolution

6.5 Gesamtauswertung und Kapitelfazit

7. Ausblick und Fazit

8. Literaturverzeichnis

Anhang I

Quellcode *BaseActivity.java*

```
package de.hsworms.inf3032.activity;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.LinearLayout;

import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.fragment.app.FragmentManager;

import com.google.android.material.navigation.NavigationView;

import de.hsworms.inf3032.R;
import de.hsworms.inf3032.data.constant.AppConstant;
import de.hsworms.inf3032.utility.ActivityUtilities;
import de.hsworms.inf3032.utility.AppUtilities;
import de.hsworms.inf3032.utility.DialogUtilities;

public class BaseActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener, DialogUtilities.OnCompleteListener {

    private static LinearLayout mLoadingView, mNoDataView;
    private Activity mActivity;
    private Context mContext;
    private Toolbar mToolbar;
    private DrawerLayout mDrawerLayout;
    private NavigationView mNavigationView;

    public static void hideLoader() {
        if (mLoadingView != null) {
            mLoadingView.setVisibility(View.GONE);
        }
        if (mNoDataView != null) {
            mNoDataView.setVisibility(View.GONE);
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```



```

    mActivity = BaseActivity.this;
    mContext = mActivity.getApplicationContext();

}

public NavigationView getNavigationView() {
    return mNavigationView;
}

public void initDrawer() {
    mDrawerLayout = findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle
        (this, mDrawerLayout, mToolbar, R.string.openDrawer, R.string.closeDrawer) {
        public void onDrawerClosed(View view) {
            super.onDrawerClosed(view);
        }

        public void onDrawerOpened(View drawerView) {
            super.onDrawerOpened(drawerView);
        }
    };
    mDrawerLayout.setDrawerListener(toggle);
    toggle.syncState();

    mNavigationView = findViewById(R.id.navigation View);
    mNavigationView.setItemIconTintList(null);
    getNavigationView().setNavigationItemSelectedListener(this);
}

public void initToolbar(boolean isTitleEnabled) {
    mToolbar = findViewById(R.id.toolbar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(isTitleEnabled);
}

public void setToolbarTitle(String title) {
    if (getSupportActionBar() != null) {
        getSupportActionBar().setTitle(title);
    }
}

public void enableUpButton() {
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
    }
}

public void initLoader() {
    mLoadingView = findViewById(R.id.loadingView);
    mNoDataView = findViewById(R.id.noDataView);
}

```

```

    }

    public void showLoader() {
        if (mLoadingView != null) {
            mLoadingView.setVisibility(View.VISIBLE);
        }

        if (mNoDataView != null) {
            mNoDataView.setVisibility(View.GONE);
        }
    }

    public void showEmptyView() {
        if (mLoadingView != null) {
            mLoadingView.setVisibility(View.GONE);
        }
        if (mNoDataView != null) {
            mNoDataView.setVisibility(View.VISIBLE);
        }
    }

    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.action_questions) {
            ActivityUtilities.getInstance().invokeNewActivity(mActivity,
            QuestionSelectActivity.class, true);
        } else if (id == R.id.action_interview_questions) {
            ActivityUtilities.getInstance().invokeNewActivity(mActivity,
            QuestionsInterviewActivity.class, false);
        } else if (id == R.id.action_fav) {
            ActivityUtilities.getInstance().invokeNewActivity(mActivity, FavoriteActivity.class,
            false);
        } else if (id == R.id.action_settings) {
            ActivityUtilities.getInstance().invokeNewActivity(mActivity, SettingsActivity.class,
            false);
        } else if (id == R.id.action_about_dev) {
            ActivityUtilities.getInstance().invokeNewActivity(mActivity, AboutDevActivity.class,
            false);
        } else if (id == R.id.action_share) {
            AppUtilities.shareApp(mActivity);
        } else if (id == R.id.notification) {
            ActivityUtilities.getInstance().invokeNewActivity(mActivity,
            NotificationListActivity.class, false);
        } else if (id == R.id.action_rate_app) {
            AppUtilities.rateThisApp(mActivity);
        } else if (id == R.id.privacy_policy) {
            ActivityUtilities.getInstance().invokeCustomUrlActivity(mActivity,
            CustomUrlActivity.class, getResources().getString(R.string.privacy),
            getResources().getString(R.string.privacy_url), false);
        } else if (id == R.id.action_exit) {
            FragmentManager manager = getSupportFragmentManager();
            DialogUtilities dialog = DialogUtilities.newInstance(getString(R.string.exit),
            getString(R.string.close_prompt), getString(R.string.yes), getString(R.string.no),

```

```

AppConstant.BUNDLE_KEY_EXIT_OPTION);
    dialog.show(manager, AppConstant.BUNDLE_KEY_DIALOG_FRAGMENT);
}

    if (mDrawerLayout != null &&
mDrawerLayout.isDrawerOpen(GravityCompat.START)) {
        mDrawerLayout.closeDrawer(GravityCompat.START);
    } return true;
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}

@Override
public void onComplete(Boolean isOkPressed, String viewIdText) {
    if (isOkPressed) {
        if (viewIdText.equals(AppConstant.BUNDLE_KEY_EXIT_OPTION)) {
            mActivity.finishAffinity();
        }
    }
}
}
}

```

Anhang II

Quellcode *WebView.java*

```

package de.hsworms.inf3032.engine;

import android.app.Activity;
import android.app.DownloadManager;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.ConnectivityManager;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.view.View;
import android.webkit.DownloadListener;
import android.webkit.ValueCallback;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

import androidx.fragment.app.Fragment;

import java.io.File;

import de.hsworms.inf3032.R;
import de.hsworms.inf3032.data.constant.AppConstant;

```

```

import de.hsworms.inf3032.data.preference.AppPreference;
import de.hsworms.inf3032.listeners.WebListener;
import de.hsworms.inf3032.utility.FilePickerUtilities;
import de.hsworms.inf3032.utility.PermissionUtilities;

public class WebView {

    public static final int KEY_FILE_PICKER = 554;
    private static final String GOOGLE_DOCS_VIEWER =
"https://docs.google.com/viewerng/viewer?url=";
    private android.webkit.WebView webView;
    private Activity mActivity;
    private Context mContext;
    private Fragment mFragment;
    private ValueCallback<Uri> mUploadMessage;
    private ValueCallback<Uri[]> mFilePathCallback;

    private WebListener mWebListener;
    private String mDownloadUrl;
    private VideoView mVideoViewer;
    private WebChromeClient.CustomViewCallback mVideoViewCallback;

    public WebView(android.webkit.WebView webView, Activity activity) {
        this.webView = webView;
        this.mActivity = activity;
        this.mContext = mActivity.getApplicationContext();
        mVideoViewer = VideoView.getInstance();
    }

    public void initWebView() {
        webView.getSettings().setJavaScriptEnabled(true);
        webView.getSettings().setLoadWithOverviewMode(true);

webView.getSettings().setAppCacheMaxSize(AppConstant.SITE_CACHE_SIZE);

webView.getSettings().setAppCachePath(mContext.getCacheDir().getAbsolutePath(
));
        webView.getSettings().setAllowFileAccess(true);
        webView.getSettings().setAppCacheEnabled(true);
        webView.getSettings().setCacheMode(WebSettings.LOAD_DEFAULT);
        webView.getSettings().setLoadWithOverviewMode(true);
        webView.getSettings().setDomStorageEnabled(true);
        webView.getSettings().setDefaultTextEncodingName("utf-8");
        webView.getSettings().setPluginState(WebSettings.PluginState.ON);
        if (!isNetworkAvailable(mContext)) {

webView.getSettings().setCacheMode(WebSettings.LOAD_CACHE_ELSE_NETWORK);
        }
        if
(AppPreference.getInstance(mContext).getTextSize().equals(mContext.getResources().getString(R.string.small_text))) {
            webView.getSettings().setTextSize(WebSettings.TextSize.SMALLER);
        } else if
(AppPreference.getInstance(mContext).getTextSize().equals(mContext.getResources().getString(R.string.default_text))) {
            webView.getSettings().setTextSize(WebSettings.TextSize.NORMAL);
        } else if
(AppPreference.getInstance(mContext).getTextSize().equals(mContext.getResources().getString(R.string.large_text))) {
            webView.getSettings().setTextSize(WebSettings.TextSize.LARGER);
        }
    }

    public void initListeners(final WebListener webListener) {

```

```

        this.mWebListener = webListener;

        webView.setWebChromeClient(new WebChromeClient() {
            @Override
            public void onProgressChanged(android.webkit.WebView view, int
newProgress) {
                super.onProgressChanged(view, newProgress);
                webListener.onProgress(newProgress);
            }

            @Override
            public boolean onShowFileChooser(android.webkit.WebView view,
ValueCallback<Uri[]> filePath, FileChooserParams fileChooserParams) {

                if (mFilePathCallback != null) {
                    mFilePathCallback.onReceiveValue(null);
                }
                mFilePathCallback = filePath;
                invokeImagePickerActivity();
                return true;
            }

            @Override
            public void onReceivedTitle(android.webkit.WebView view, String
title) {
                super.onReceivedTitle(view, title);
                webListener.onPageTitle(webView.getTitle());
            }

            @Override
            public void onShowCustomView(View view, CustomViewCallback
callback) {
                super.onShowCustomView(view, callback);
                mVideoViewCallback = callback;
                mVideoViewer.show(mActivity);
                mVideoViewer.setVideoLayout(view);
            }

            @Override
            public void onHideCustomView() {
                super.onHideCustomView();
                mVideoViewer.dismiss();
                mVideoViewCallback.onCustomViewHidden();
            }

        });

        webView.setWebViewClient(new WebViewClient() {

            @Override
            public boolean shouldOverrideUrlLoading(android.webkit.WebView
webView, String webUrl) {

                loadPage(webUrl);
                return true;
            }

            @Override
            public void onPageStarted(android.webkit.WebView view, String
url, Bitmap favicon) {
                webListener.onStart();
            }
        });

```

```

    }

    @Override
    public void onPageFinished(android.webkit.WebView view, String
url) {
        webListener.onLoaded();
    }

    });

    webView.setDownloadListener(new DownloadListener() {
        @Override
        public void onDownloadStart(String url, String userAgent,
String contentDisposition, String
mimetype,
                                long contentLength) {
            mDownloadUrl = url;
            downloadFile();
        }
    });

}

public void loadPage(String webUrl) {
    if (isNetworkAvailable(mContext)) {

        if (webUrl.startsWith("tel:") || webUrl.startsWith("sms:") ||
webUrl.startsWith("smsto:")
            || webUrl.startsWith("mms:") ||
webUrl.startsWith("mmsto:")
            || webUrl.startsWith("mailto:") /*||
webUrl.contains("youtube.com") */
            || webUrl.contains("geo:")) {
            invokeNativeApp(webUrl);
        } else if (webUrl.contains("?target=blank")) {
            invokeNativeApp(webUrl.replace("?target=blank", ""));
        } else if (webUrl.endsWith(".doc") || webUrl.endsWith(".docx") ||
webUrl.endsWith(".xls")
            || webUrl.endsWith(".xlsx") || webUrl.endsWith(".pptx")
            || webUrl.endsWith(".pdf")) {
            webView.loadUrl(GOOGLE_DOCS_VIEWER + webUrl);
            webView.getSettings().setBuiltInZoomControls(true);
        } else {
            webView.loadUrl(webUrl);
        }

    } else {
        mWebListener.onNetworkError();
    }
}

public void loadHtml(String htmlString) {
    if (htmlString.startsWith("tel:") || htmlString.startsWith("sms:") ||
htmlString.startsWith("smsto:")
        || htmlString.startsWith("mms:") ||
htmlString.startsWith("mmsto:")
        || htmlString.startsWith("mailto:") /*||
htmlString.contains("youtube.com") */
        || htmlString.contains("geo:")) {
        invokeNativeApp(htmlString);
    } else if (htmlString.contains("?target=blank")) {
        invokeNativeApp(htmlString.replace("?target=blank", ""));
    } else if (htmlString.endsWith(".doc") ||

```

```

htmlString.endsWith(".docx") || htmlString.endsWith(".xls")
    || htmlString.endsWith(".xlsx") ||
htmlString.endsWith(".pptx") || htmlString.endsWith(".pdf")) {
    webView.loadUrl(GOOGLE_DOCS_VIEWER + htmlString);
    webView.getSettings().setBuiltInZoomControls(true);
} else {
    webView.loadDataWithBaseURL(null, htmlString, "text/html;
charset=utf-8", "UTF-8", null);
}
}

private boolean isNetworkAvailable(Context context) {
    ConnectivityManager connectivityManager = ((ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE));
    return connectivityManager.getActiveNetworkInfo() != null &&
connectivityManager.getActiveNetworkInfo().isConnected();
}

private void invokeNativeApp(String url) {
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
    mActivity.startActivity(intent);
}

public void invokeImagePickerActivity() {
    if (PermissionUtilities.isPermissionGranted(mActivity,
PermissionUtilities.SD_WRITE_PERMISSIONS,
PermissionUtilities.REQUEST_WRITE_STORAGE_UPLOAD)) {
        Intent chooseImageIntent =
FilePickerUtilities.getPickFileIntent(mActivity);
        if (mFragment == null) {
            mActivity.startActivityForResult(chooseImageIntent,
KEY_FILE_PICKER);
        } else {
            mFragment.startActivityForResult(chooseImageIntent,
KEY_FILE_PICKER);
        }
    }
}

public void uploadFile(Intent data, String filePath) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        Uri[] results = null;
        if (filePath != null) {
            results = new Uri[]{Uri.fromFile(new File(filePath))};
        }

        if (results == null) {
            String dataString = data.getDataString();
            if (dataString != null) {
                results = new Uri[]{Uri.parse(dataString)};
            }
        }

        if (mFilePathCallback != null) {
            mFilePathCallback.onReceiveValue(results);
            mFilePathCallback = null;
        }
    } else if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.KITKAT) {
        Uri result = data == null ? Uri.fromFile(new File(filePath)) :
data.getData();
        if (mUploadMessage != null) {
            mUploadMessage.onReceiveValue(result);
        }
    }
}

```

```

        mUploadMessage = null;
    }

}

public void cancelUpload() {
    if (mFilePathCallback != null) {
        mFilePathCallback.onReceiveValue(null);
    }
    mFilePathCallback = null;
}

public void downloadFile() {
    if (PermissionUtilities.isPermissionGranted(mActivity,
PermissionUtilities.SD_WRITE_PERMISSIONS,
PermissionUtilities.REQUEST_WRITE_STORAGE_DOWNLOAD)) {
        DownloadManager.Request request = new DownloadManager.Request(
            Uri.parse(mDownloadUrl));

        request.allowScanningByMediaScanner();

request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_
NOTIFY_COMPLETED); //Notify client once download is completed!

request.setDestinationInExternalPublicDir(Environment.DIRECTORY_DOWNLOADS,
"Downloading file...");
        DownloadManager dm = (DownloadManager)
mContext.getSystemService(Context.DOWNLOAD_SERVICE);
        dm.enqueue(request);
    }
}
}

```

Anhang III

Anhang IV

Anhang V

Anhang VI