

# **Entwicklung einer mobilen Lern-Applikation**

## **Studienarbeit**

über die Theoriephase des dritten Studienjahrs

an der Fakultät für Technik  
im Studiengang Informationstechnik

an der Duale Hochschule Baden-Württemberg (DHBW) Ravensburg  
Campus Friedrichshafen

von  
Phillipp Patzelt, Nico Bayer

Frau Schmidt Mail nachfragen

Bearbeitungszeitraum:	Nochmal überprüfen
Matrikelnummer, Kurs:	8138164. 3056312, TIT20
Dualer Partner:	
Betreuer des Dualen Partners:	Claudia Zinser

# Selbstständigkeitserklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017.

Hiermit versichern wir, dass wir unsere Studienarbeit mit dem Thema:

## **Entwicklung einer mobilen Lern-Applikation**

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

# **Abstract**

**Zusammenfassung**

**Schlüsselwörter**

**Abstract**

**Keywords**

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>VII</b>
<b>Abbildungsverzeichnis</b>	<b>VIII</b>
<b>Tabellenverzeichnis</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Planungsphase</b>	<b>2</b>
2.1 Lastenheft . . . . .	2
2.1.1 Allgemeine Beschreibung . . . . .	2
2.1.2 Anforderungen . . . . .	2
2.1.3 Fachliches Umfeld . . . . .	2
2.1.4 Ausblick . . . . .	2
2.1.5 Erweiterungsmöglichkeiten (optional) . . . . .	2
2.2 Arbeitspaketplan . . . . .	3
2.3 Vorrausichtlicher Zeitplan . . . . .	4
2.4 Qualitätsmanagement Maßnahmen . . . . .	5
2.5 Konfigurationsmanagement Maßnahmen . . . . .	5
2.6 Evaluierung der Plattformen und Programmiersprachen für die App-Entwicklung . . . . .	6
<b>3 Definitionsphase</b>	<b>7</b>
3.1 Theoretische Grundlagen . . . . .	7
3.1.1 Lernen ist nicht gleich Verstehen . . . . .	7
3.1.2 Synapsen und deren Rolle im Lernprozess . . . . .	7
3.1.3 Neuronen: Die Bausteine des Gehirns . . . . .	8
3.1.4 Neuronale Netze: Die Grundlage des Lernens und des Gedächtnisses . . . . .	9
3.1.5 Optimaler Lernplan anhand des Spacing Effect . . . . .	10
3.1.6 Tagging innerhalb der Lern-App . . . . .	12
3.1.7 Lernkategorien & Lernziele . . . . .	13
3.1.8 Gamification und ihr positiver Effekt auf das Lernen . . . . .	13
3.2 Pflichtenheft . . . . .	14
3.3 Use Case Diagramm . . . . .	15
3.4 Use Case Beschreibung . . . . .	16
3.5 Datenbankstruktur . . . . .	21
3.5.1 Datenbankmodell . . . . .	21

---

3.5.2	Data Dictionary . . . . .	22
3.6	HMI . . . . .	26
3.6.1	Seitenhierarchie . . . . .	26
3.6.2	UI-Mockups . . . . .	27
3.7	Flow-Chart-Diagramm . . . . .	36
<b>4</b>	<b>Entwurf</b>	<b>37</b>
4.1	Auswahl und Begründung des Datenbankkonzepts . . . . .	37
4.2	Auswahl von wichtigen Bibliotheken/Frameworks . . . . .	37
4.2.1	DaggerHilt Dependency Injection Framework . . . . .	38
4.2.2	Markwon: Eine Markdown-Bibliothek für Android . . . . .	38
4.2.3	Glide: Ein Framework zum Laden von Bildern in Android . . . . .	39
4.2.4	Android Navigation: Ein Framework zur Navigation in Android-Anwendungen	39
4.2.5	Firebase Android SDK: Eine Bibliothek zur Integration von Firebase in Android-Anwendungen . . . . .	40
4.3	Wahl des Architektur-Patterns: Model-View-ViewModel (Model View View- Model (MVVM)) . . . . .	40
4.4	Software-Komponenten . . . . .	42
4.4.1	Gesamtkomposition . . . . .	42
4.4.2	Frontend . . . . .	43
4.4.3	Backend . . . . .	44
4.5	Deployment Diagramm . . . . .	45
4.6	Klassen Diagramm . . . . .	46
4.7	Sequenzdiagramme . . . . .	48
4.8	Prototyp von LearnAhead . . . . .	53
<b>5</b>	<b>Implementierung</b>	<b>54</b>
5.1	Installationsanleitung . . . . .	54
5.1.1	Anleitung für User . . . . .	54
5.1.2	Anleitung für Developer . . . . .	55
5.1.3	Verwendung des Android-Emulators . . . . .	55
5.1.4	Übertragen der App zu Entwicklungszwecken ohne Deployment . . . .	56
5.2	Code Conventions . . . . .	56
5.3	IT-Sicherheit in LearnAhead . . . . .	57
5.3.1	Firebase Authentifizierung . . . . .	57
5.3.2	Firebase Datenspeicherung . . . . .	57

<b>6</b>	<b>Ergebnisse</b>	<b>58</b>
6.1	Nutzen der Lern-App . . . . .	58
6.2	Feedback von Präsentation der Studienarbeit . . . . .	58
6.3	Der finale Zeitplan . . . . .	59
<b>7</b>	<b>User-Tests</b>	<b>60</b>
7.1	Offene Tests, allgemeine Ziele . . . . .	61
7.2	Freie Tests . . . . .	63
<b>8</b>	<b>Fazit</b>	<b>64</b>
8.1	Ablauf des Projekts . . . . .	64
8.2	Ausblick und die Zukunft . . . . .	65
	<b>Referenzen</b>	<b>69</b>

---

## Abkürzungsverzeichnis

<b>DHBW</b> Duale Hochschule Baden-Württemberg . . . . .	I
<b>MVVM</b> Model View ViewModel . . . . .	V
<b>DSGVO</b> Datenschutz Grundverordnung . . . . .	2
<b>App</b> Applikation . . . . .	6
<b>JVM</b> Java Virtual Machine . . . . .	6
<b>LTP</b> Long-Term Potentiation . . . . .	8
<b>LTD</b> Long-Term Depression . . . . .	8
<b>DI</b> Dependency Injection . . . . .	38
<b>UI</b> User-Interface . . . . .	39
<b>SDK</b> Software Development Kit . . . . .	40
<b>APIs</b> Application Programming Interfaces . . . . .	40
<b>UML</b> Unified Modeling Language . . . . .	45
<b>APK</b> Android Package . . . . .	45
<b>PWA</b> Progressive Web App . . . . .	66

## Abbildungsverzeichnis

3.1.1 Ebbinghaus Vergessenskurve . . . . .	10
3.1.2 Ebbinghaus verzögerte Vergessenskurve durch Wiederholungen . . . . .	11
3.3.1 Use Case Diagramm . . . . .	15
3.5.1 LearnAhead Datenbankstruktur . . . . .	21
3.6.1 Die Seitenhierarchie in LearnAhead . . . . .	26
3.6.2 Login und Registrierung . . . . .	27
3.6.3 Home-Screen and Profilansicht . . . . .	28
3.6.4 Erstellen einer Lernkategorie and Ansicht Lernkategorien . . . . .	29
3.6.5 Ansicht Zusammenfassungen and Erstellen einer Zusammenfassung . . . . .	30
3.6.6 Ansicht Fragen und Tests and Ansicht Fragen . . . . .	31
3.6.7 Erstellen einer Frage and Ansichts Tests . . . . .	32
3.6.8 Test . . . . .	33
3.6.9 Test . . . . .	34
3.6.10 Erstellen eines Lernziels . . . . .	35
3.7.1 Flow-Chart-Diagramm . . . . .	36
4.3.1 MVVM Diagramm . . . . .	41
4.4.1 Gesamtkomposition . . . . .	42
4.4.2 Frontend UI Navigation . . . . .	43
4.4.3 Frontend UI Kommunikation mit ViewModel . . . . .	43
4.4.4 Backend Komposition . . . . .	44
4.5.1 Das Deployment Diagramm von LearnAhead . . . . .	45
4.6.1 Allgemeines Klassendiagramm der LearnAhead App . . . . .	47
4.7.1 Sequenzdiagramm - User erstellt eine Frage . . . . .	48
4.7.2 Sequenzdiagramm - User erstellt einen Test . . . . .	49
4.7.3 Sequenzdiagramm - User erstellt eine Zusammenfassung . . . . .	49
4.7.4 Sequenzdiagramm - User erhält eine Belohnung für täglichen Login . . . . .	50
4.7.5 Sequenzdiagramm - User erstellt Lernziel . . . . .	50
4.7.6 Sequenzdiagramm - User ändert sein Profilbild . . . . .	51
4.7.7 Sequenzdiagramm - User führt Test aus . . . . .	52

## Tabellenverzeichnis

2.3.1 Vorräusichtlicher Entwurf eines Zeitplan für die Entwicklung der Lern-App. . .	4
--	---



# 1 Einleitung

„Es ist keine Schande nichts zu wissen, wohl aber nichts lernen zu wollen.“

Diese Worte verwendete schon der griechische Philosoph Platon. Wir hören nie auf zu lernen, das ist ein Fakt.

Der Prozess des Lernens lässt sich nicht aktiv unterbinden. Lernen ist der Impuls, Informationen zu verarbeiten, einzuordnen und zu verstehen. Das Behalten des Erlernten passiert ebenfalls automatisch, wenn auch bei vielen Dingen nicht dauerhaft. Der Mensch ist so angelegt, dass er seine Umwelt begreifen will. Somit ist Lernen ein intrinsisch motivierter Prozess, der zwar angeregt, aber nicht verordnet werden kann. Etwa 100 Milliarden Nervenzellen besitzt unser Gehirn, die sogenannten Neuronen. An deren Ende liegen Synapsen, die elektrische Signale der Neuronen in Form von chemischen Botenstoffen an die nächsten Neuronen abgeben. Diese Kettenreaktionen tragen die Informationen und Signale durch unser neuronales Netz im Gehirn an die Stellen, wo es benötigt wird, egal ob wir unsere Muskeln bewegen wollen oder unsere Sinne einsetzen, und lösen dort Reaktionen aus.

Obwohl der Mensch also nicht nur in der Theorie, sondern auch in der Praxis für das Lernen konstruiert ist, macht das Lernen vielen Menschen nicht Spaß und sie behaupten: „Sie können nicht lernen“.

Faktisch gesehen ist das jedoch nicht richtig! Jahrhunderte, ja sogar Jahrtausende der Evolution haben uns dazu ausgebildet nie aufzuhören zu lernen. Die heutige Welt bietet jedoch viele Tücken, welche das Lernen einschränken können.

Das Lernen ist ein komplexer, jedoch kein komplizierter Prozess, es muss nicht der komplette Bio-chemische Ablauf verstanden werden, jedoch sollte man einige wichtige Grundlagen beachten.

## **2 Planungsphase**

### **2.1 Lastenheft**

#### **2.1.1 Allgemeine Beschreibung**

Eine mobile LernApp, mit der Möglichkeit Lerninhalte zusammenzufassen und von diesen in Abhängigkeit Tests zu erstellen. Von den eingetragenen Lerninhalten können dann Tests erstellt werden, welche aus verschiedenen Fragen bestehen. Es besteht die Möglichkeit die Inhalt nach Kategorien zu gruppieren und Lernzielen zu zuweisen. Mit diesen Lernzielen wird dem User dann ein empfohlener Lernplan erstellt.

#### **2.1.2 Anforderungen**

- Lerninhalte zusammenfassen
- Tests erstellen
- Lernkategorien
- Auf Android verfügbar
- Lernziele und dazugehörigen automatisch erstellten Lernplan
- User-Profil mit Benutzername und Kennwort

#### **2.1.3 Fachliches Umfeld**

- Plattformabhängig mit Android Studio
- Mobile Lösung
- Datenbank
- IT-Security
- Datenschutz Grundverordnung (DSGVO)

#### **2.1.4 Ausblick**

Bei erfolgreichen Entwicklungsergebnissen soll die Lösung in Betrieb genommen und der Öffentlichkeit, auf Android, zur Verfügung gestellt werden.

#### **2.1.5 Erweiterungsmöglichkeiten (optional)**

- Importieren und Exportieren von Lerninhalten auf WhatsApp oder Ähnlichem

## 2.2 Arbeitspaketplan

Der Arbeitspaketplan bezeichnet die Aufzählung jedes Arbeitspakets auf Basis des Lastenhefts.

Ein Arbeitspaket wird durch folgendes definiert:

- Ein definiertes Ergebnis (was soll in diesem Arbeitspaket erreicht werden)
- Der zeitliche Aufwand des Arbeitspakets
- Die Vorbedingungen, die beim Bearbeiten zu beachten sind
- Die Dauer

Um die Arbeitspakete grafisch aufbereitet darstellen zu können, werden diese in die Anwendung Jira<sup>1</sup> ausgelagert. Hier wird der Arbeitspaketplan als Unterteilung der einzelnen Epics in User Stories dargestellt. Erledigte Epics und User Stories sind dabei unter dem Reiter „Fertig“ einsehbar. In den einzelnen User Stories wird ein definiertes Ergebnis aus Sicht des Nutzers beschrieben. Der zeitliche Aufwand der einzelnen Arbeitspakete ergibt sich aus der Summe der Dauer der zugewiesenen Tasks.

In Kombination mit Scrum werden dabei vor Beginn des jeweiligen Sprints die einzelnen Tasks geschätzt und auf einen festen Arbeitsaufwand festgelegt.

Im Laufe des Sprints werden dann die zugewiesenen Stunden von den zugewiesenen Bearbeitern abgebaut und im jeweiligen Task aktualisiert. Ist der Task beendet, so wird er mit „Done“ markiert und besitzt somit keinen Arbeitsaufwand mehr.

Vorbedingungen, sowie die festgelegte Dauer für die Bearbeitung eines Arbeitspaketes werden durch die Sprints definiert. Durch Aufteilen der Tasks in verschiedene, nacheinander ablaufende Sprints, können Vorbedingungen durch Einteilung in einen vorangehenden Sprint gesetzt werden. Darüber hinaus können den einzelnen Tasks Prioritäten zwischen eins und vier zugeordnet werden, was eine Hierarchie innerhalb eines Sprints ermöglicht. Zusätzlich limitiert die Dauer des Sprints die Bearbeitungszeit für den jeweiligen Task.

---

<sup>1</sup><https://studienarbeitlernapp.atlassian.net/jira/software/projects/LER/boards/1>

## 2.3 Vorrausichtlicher Zeitplan

Zu Beginn der Studienarbeit wurde ein vorrausichtlicher Zeitplan erstellt, der die einzelnen Meilensteine und deren Fertigstellungstermine beinhaltet. Dieser Zeitplan ist in Tabelle 2.3.1 unten zu sehen.

Meilenstein	Abgeschlossen bis	Beschreibung
Literaturrecherche	KW 5 2023	Das Durchführen einer umfangreichen Literaturrecherche auf Basis von wissenschaftlichen Dokumenten.
Use-Case-Erstellung	KW 45 2022	Identifizierung und Dokumentation der Hauptfunktionalitäten und Anwendungsfälle der Lern-App.
UI-Konzept	KW 52 2022	Entwicklung eines visuellen Konzepts für die Benutzeroberfläche (UI) der Lern-App.
Datenbank-Konzept	KW 4 2023	Design und Auswahl des Datenbanksystems, die für die App benötigt wird.
Architektur-Konzept	KW 6 2023	Realisierung einer Code-Architektur und Auswahl verschiedener Komponenten sowie Frameworks, die in der App verwendet werden.
Architektur-Prototyp	KW 8 2023	Erstellung eines ersten Prototypen der die vorgeschlagene Architektur implementiert.
Login / Registrierung	KW 11 2023	Implementierung der Funktionen für Anmeldung, Registrierung und Passwortwiederherstellung.
Lernkategorien & Lernziele	KW 13 2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Lernkategorien und -zielen.
Erstellung von Fragen und Tests	KW 18 2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Fragen und Tests.
Erstellung von Zusammenfassungen	KW 24 2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Zusammenfassungen von Lernkategorien.
Optimale Pausenberechnung realisieren	KW 26 2023	Erstellung eines Algorithmus, welcher den Nutzer die optimale Pause vorschlägt sowie erinnert.
Optimale Lernplan generieren	KW 27 2023	Erstellung eines optimalen Lernplans auf Basis der Lernziele.
Durchführen von Tests	KW 28 2023	Durchführung von umfassenden Tests, um die Qualität, Funktionalität und Stabilität der App sicherzustellen.
Bugs beheben	KW 29 2023	Behebung von Fehlern und Problemen in der App.
Dokumentation	KW 29 2023	Erstellung einer wissenschaftlichen Arbeit, die das Vorgehen, Funktionen, die Implementierung sowie die Verwendung der App begründet.

**Tab. 2.3.1:** Vorrausichtlicher Entwurf eines Zeitplan für die Entwicklung der Lern-App.

## 2.4 Qualitätsmanagement Maßnahmen

Die Qualität der Lern-App wird durch „Unit-Tests“ sowie von externen Probanden durch festgelegte Tests ermittelt. Die Unit-Tests werden von den Entwicklern der App durchgeführt, um die Qualität der einzelnen Komponenten im Backend zu gewährleisten. Nach Abschluss der Implementierungsphase wird eine geschlossene Beta durchgeführt, bei der die Tester informiert werden, welche Funktionen getestet werden sollen. Folgende Funktionalitäten sollen überprüft werden:

- Anmeldung
- Registrierung
- Passwortwiederherstellung
- Erstellen von Lernkategorien
- Erstellen von Lernzielen
- Erstellen von Fragen
- Erstellen von Tests
- Erstellen von Zusammenfassungen
- Generierung eines optimalen Lernplans
- Berechnung der optimalen Pausen

Zu jeder Funktionalität soll der Tester eine kurze textuelle Beschreibung abgeben, welche positiven sowie negativen Aspekte er festgestellt hat. Anhand der Rückmeldungen kann die App weiter verbessert und Fehler behoben werden. Zusätzlich sollen die Tests auf unterschiedlichen Android-Geräten durchgeführt werden, um die Kompatibilität der App zu gewährleisten.

## 2.5 Konfigurationsmanagement Maßnahmen

Die agile Planung erfolgt im Kanban in Jira, hier ist der Backlog angelegt, in welchem die einzelnen Aufgaben liegen. Meetings werden auf Discord durchgeführt. Die Dokumentation wird mit  $\text{\LaTeX}$  geschrieben.

Die Versions- und Releaseverwaltung erfolgt in einem GitHub Repository unter dem Git-Branching-Modell Gitflow. Gitflow sieht zwei Branches vor um den Verlauf des Projekts aufzuzeichnen. Der main-Branch enthält den offiziellen Release-Verlauf und der develop-Branch dient als Integrations-Branch für Features. Der develop-Branch enthält den kompletten Versionsverlauf des Projekts, während der main-Branch eine verkürzte Version enthält.

Jedes neue Feature sollte sich auf seinem eigenen Branch befinden, der zu Sicherungs-/Zusammenarbeitszwecken zum zentralen Repository gepusht werden kann. Ein neuer feature-Branch wird aus dem develop-branch gemerget. Wenn ein Feature fertig ist, wird es zurück in den develop-Branch gemerget. Features sollten niemals direkt mit dem main-Branch interagieren.

Wenn der develop-Branch genügend Features für ein Release enthält, wird vom develop-Branch ein release-Branch geforkt. Damit beginnt der neue Release-Zyklus. In diesem Branch sollten ab diesem Punkt keine neuen Features mehr hinzugefügt werden, sondern nur Bugfixes und ähnliche Release-orientierte Änderungen. Ist er zur Auslieferung bereit, wird der release-Branch in den main-Branch gemerget und mit einer Versionsnummer getaggt. Zusätzlich wird der release-Branch in den develop-Branch gemerged.

Maintenance- bzw. hotfix-Branches eignen sich für schnelle Patches von Produktions-Releases. Sie werden aus dem main-Branch geforkt. Sobald der Bugfix abgeschlossen ist, wird er sowohl in den main- als auch in den develop-Branch (oder den aktuellen release-Branch) gemerget.

## **2.6 Evaluierung der Plattformen und Programmiersprachen für die App-Entwicklung**

Die Lern-App „LearnAhead“ wird als Android-App umgesetzt, weil die Entwickler selbst nur Android-Geräte zur Verfügung haben. Zusätzlich kommt noch hinzu, dass eine Apple-Entwickler-Lizenz ca. 99\$ pro Jahr kostet [37]. Die Kosten für die Veröffentlichung einer Android-App betragen hierbei nur einmalig 25\$ [1]. Daraus ergibt sich, dass die Entwicklung einer Android-App deutlich günstiger ist.

Für die Entwicklung der Applikation (App) wird die Programmiersprache Kotlin verwendet. Hierbei handelt es sich um eine moderne Programmiersprache, die von JetBrains entwickelt wurde. Sie ist eine statisch typisierte Open-Source-Programmiersprache, die auf der Java Virtual Machine (JVM) ausgeführt wird. Ebenfalls bietet sie viele Vorteile gegenüber Javam darunter als Beispiel die Kompaktheit, da sie weniger Boilerplate-Code benötigt. Die Stärken liegen hierbei auf der einfachen und kompakten Syntax von Kotlin. Diese reduziert auf der einen Seite den Programmieraufwand, um die gleiche Funktionalität wie in Java zu erreichen, was den Code leichter lesbar und verständlicher macht. Zusätzlich existiert das größte Problem von Java, die NullPointerException, in Kotlin nicht. [6]

## 3 Definitionsphase

### 3.1 Theoretische Grundlagen

#### 3.1.1 Lernen ist nicht gleich Verstehen

Im Alltag werden die Begriffe „Lernen“ und „Verstehen“ oft synonym verwendet. In der Wissenschaft werden diese Begriffe jedoch unterschiedlich definiert. „**Lernen**“ ist ein Prozess, bei welchem Wissen, Emotionen, Fertigkeiten, aber auch Verhalten, Einstellungen und Werte durch Erfahrungen verändert werden. Hierbei werden neue Informationen erfasst und im Gehirn gespeichert, wobei das Verständnis noch nicht voll ausgereift ist. [32]

Beim Erlernen neuer Informationen treffen diese zuerst auf den Hippocampus des Gehirns und lösen dort ein Aktivitätsmuster der Nervenzellen aus. Dabei wird durch dieses Erregungsmuster entschieden, ob das neue Wissen an das Großhirn weitergeleitet wird. Nachdem die wichtigsten Informationen aufgenommen wurden, müssen die Teile in der Großhirnrinde trainiert werden, damit das Netzwerk die Möglichkeit hat, die Kontaktstellen anzupassen. Das bedeutet je öfter eine Information erneut erlernt wird, desto tiefer verankert sich diese im Langzeitgedächtnis. Im Gegenzug kann das erlangte Wissen auch verblassen, wenn die Nervenzellen nicht länger stimuliert werden. [3, S. 27]

„**Verstehen**“ hingegen bedeutet, dass ein Mensch ein Konzept erstellt und dieses somit auf unbekannte Fragen und neue Situationen anwenden kann. [3, S. 86] Das heißt, dass beim Verstehensprozess sich neue Infos nicht gemerkt werden sollen, sondern dass die Informationen neu verarbeitet werden und somit ein tieferes Verständnis entsteht. [3, S. 113] Bei dem Verstehensprozess kann es einen „Aha-Moment“ geben, wodurch das Verständnis plötzlich eintritt und das Denkmodell sich neu formt. Somit ist das Aufbauen eines Denkmodells für das Verstehen entscheidend. [3, S. 114–115]

#### 3.1.2 Synapsen und deren Rolle im Lernprozess

Synapsen sind essentiell für die Informationsverarbeitung im Gehirn und spielen eine zentrale Rolle im Lernprozess [29]. Sie sind die Verbindungen zwischen den Neuronen und ermöglichen die Informationsübertragung und -verarbeitung. Eine Synapse besteht aus einem präsynaptischen Endknopf, der Neurotransmitter in den synaptischen Spalt freisetzt, einem synaptischen Spalt, der den Raum zwischen den Neuronen darstellt, und einer postsynaptischen Dichte, die Neurotransmitter aufnimmt und den postsynaptischen Impuls auslöst [2].

Das Lernen auf zellulärer Ebene involviert die Änderung der Stärke dieser synaptischen Verbindungen, ein Prozess, der als synaptische Plastizität bezeichnet wird. Die synaptische Plastizität ermöglicht es dem Gehirn, auf Erfahrungen zu reagieren und zu lernen, und wird oft mit dem Sprichwort „Neurons that fire together, wire together“ umschrieben [28].

Es gibt zwei Haupttypen der synaptischen Plastizität: Long-Term Potentiation (LTP) und Long-Term Depression (LTD). LTP erhöht die Stärke synaptischer Verbindungen und ist eng mit dem Prozess des Lernens und der Gedächtnisbildung verbunden. Auf der anderen Seite verringert LTD die Stärke der Verbindungen und wird mit dem Löschen unnötiger oder fehlerhafter Informationen assoziiert [7].

Ein interessanter Aspekt der synaptischen Plastizität ist die Tatsache, dass sie Aktivitätsabhängig ist. Das bedeutet, dass Veränderungen in der Stärke synaptischer Verbindungen nicht zufällig auftreten, sondern auf das Vorhandensein spezifischer Muster neuronaler Aktivität reagieren [29]. Dieses Phänomen erklärt, warum bestimmte Lernerfahrungen stärkere und dauerhaftere Spuren in unserem Gedächtnis hinterlassen als andere.

Die Fähigkeit von Synapsen, ihre Stärke und Struktur zu ändern, bildet die Grundlage für die Anpassungsfähigkeit und Flexibilität unseres Gehirns und ermöglicht es uns, ein Leben lang zu lernen und uns an neue Erfahrungen anzupassen [2].

### **3.1.3 Neuronen: Die Bausteine des Gehirns**

Neuronen sind die grundlegenden funktionellen Einheiten des Gehirns und des Nervensystems. Sie sind spezialisiert auf das Senden, Empfangen und Verarbeiten von Informationen und bilden ein komplexes Netzwerk, das als Grundlage für alle kognitiven Funktionen, einschließlich Lernen und Gedächtnis, dient [2].

Ein typisches Neuron besteht aus drei Hauptkomponenten: dem Zellkörper (auch Soma genannt), den Dendriten und dem Axon. Der Zellkörper enthält den Kern der Zelle und andere lebenswichtige Organellen. Die Dendriten sind verzweigte Fortsätze, die Signale von anderen Neuronen empfangen, und das Axon ist ein langer Fortsatz, der Signale an andere Neuronen weiterleitet [29].

Die Übertragung von Informationen zwischen Neuronen findet in spezialisierten Verbindungen namens Synapsen statt. Wenn ein Signal das Ende des Axons erreicht, wird ein chemischer Neurotransmitter freigesetzt, der die synaptische Verbindung überquert und ein Signal in der nachfolgenden Neuronenzelle auslöst [2].



Laut Manfred Spitzer ist das Neuron nicht nur ein einfacher Signalüberträger, sondern spielt auch eine aktive Rolle in Lernprozessen [38]. Neuronen können ihre Aktivität und Vernetzung in Abhängigkeit von der erhaltenen Information modifizieren. Das bedeutet, dass der Lernprozess nicht nur die Stärkung bestehender neuronalen Verbindungen beinhaltet, sondern auch die Bildung neuer Verbindungen und den Abbau unnötiger Verbindungen. Dieser dynamische und anpassungsfähige Charakter des neuronalen Netzwerks bildet die Grundlage für unsere Fähigkeit, ein Leben lang zu lernen und uns an ständig wechselnde Umstände anzupassen [38].

### **3.1.4 Neuronale Netze: Die Grundlage des Lernens und des Gedächtnisses**

Ein neuronales Netz ist ein komplexes System von miteinander verbundenen Neuronen, das Informationen verarbeiten und speichern kann. In seiner einfachsten Form besteht ein neuronales Netz aus einer Eingangsschicht (den Dendriten), einer oder mehreren versteckten Schichten (den Neuronen) und einer Ausgangsschicht (den Axonen) [2].

Neuronale Netze sind hochdynamisch und anpassungsfähig. Sie ändern ihre Struktur und Funktion in Abhängigkeit von den Informationen, die sie erhalten, ein Prozess, der als synaptische Plastizität bezeichnet wird [29]. Dies ermöglicht es uns, zu lernen und uns an wechselnde Umgebungen und Erfahrungen anzupassen.

Manfred Spitzer beschreibt, dass diese Flexibilität der neuronalen Netze es uns ermöglicht, komplexe kognitive Fähigkeiten wie Gedächtnis, Lernen, Aufmerksamkeit und Emotionen zu entwickeln [38]. Im Gegensatz zu einem Computer, der auf festen Algorithmen basiert, kann unser Gehirn seine Struktur und Funktion an die Anforderungen der Umwelt anpassen. Dies ermöglicht es uns, ein Leben lang zu lernen und uns an neue Situationen anzupassen.

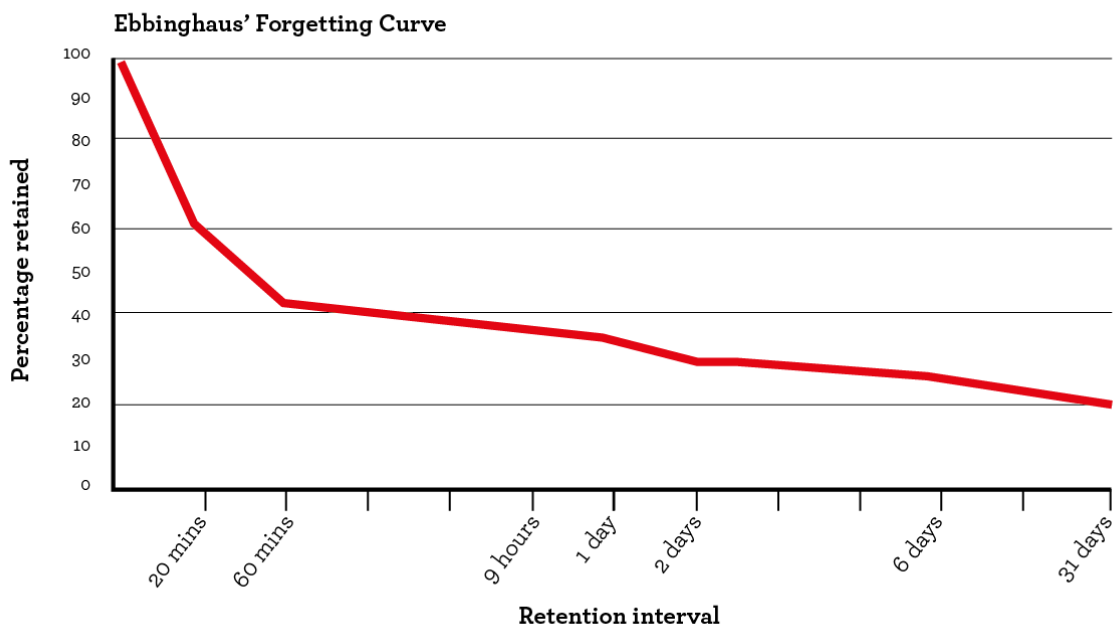
Es ist wichtig zu beachten, dass, obwohl die Metapher des neuronalen Netzwerks hilft, die Funktion des Gehirns zu erklären, sie die Komplexität und Feinheit des tatsächlichen biologischen Systems nicht vollständig einfängt. Die Beziehungen zwischen Neuronen sind nicht einfach binär, sondern beinhalten eine Vielzahl von Neurotransmittern und Mechanismen, die eine fein abgestimmte Kontrolle und Modulation der neuronalen Aktivität ermöglichen [7].

Trotz dieser Einschränkungen bleibt das Modell des neuronalen Netzwerks ein mächtiges Werkzeug, um zu verstehen, wie unser Gehirn Informationen verarbeitet und speichert, und bildet die Grundlage für viele Ansätze in der kognitiven Psychologie und der künstlichen Intelligenz [28].

### 3.1.5 Optimaler Lernplan anhand des Spacing Effect

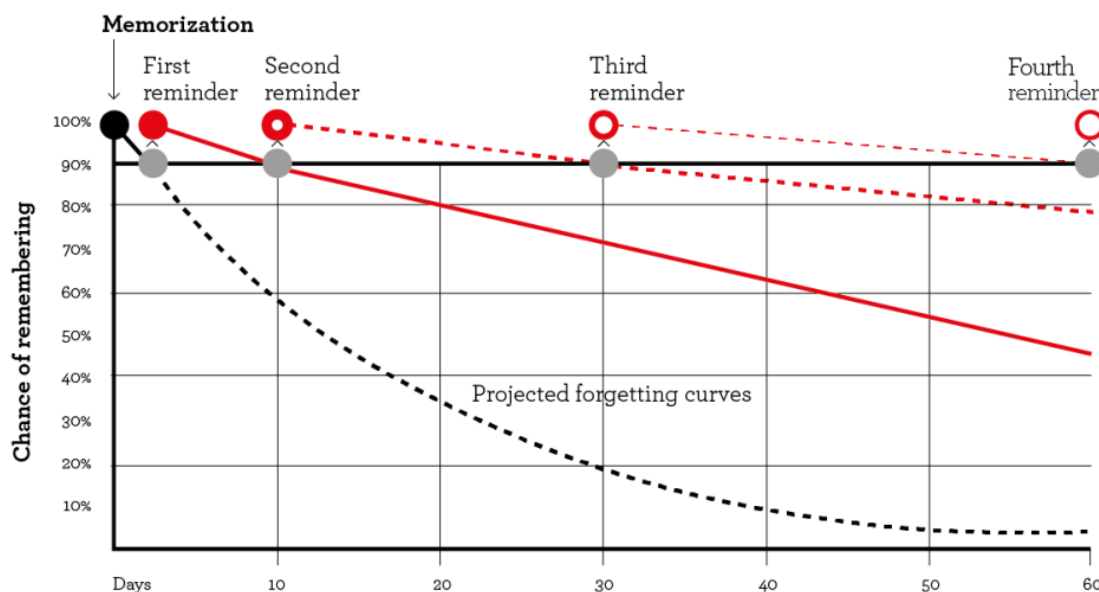
„Das neue Lernen heißt verstehen“ von Ulrich Beck vergleicht dieser das langfristige Lernen mit dem gießen einer Blume. Hierbei ist es wichtig, dass die Blumen in regelmäßigen Abständen gegossen werden, anstatt sie einmal mit einer großen Menge Wasser zu gießen. Bei den Menschen ist dies ähnlich. Viele lernen den ganzen Tag und versuchen so viel wie möglich in kurzer Zeit zu lernen, was jedoch nicht effektiv ist, da das Gehirn nicht in der Lage ist, so viel auf einmal zu verarbeiten und dieses Wissen nach einer Prüfung wieder vergessen wird. [3, S. 159]

Dadurch ist es wichtig Pausen zwischen dem Lernen einzulegen, damit das Gehirn die Informationen verarbeiten kann. Dieser Effekt wird auch „Spacing Effect“ genannt und wurde von dem deutschen Psychologen und Pionier der Gedächtnisforschung Hermann Ebbinghaus entdeckt. [39] Die größten Entdeckungen waren im Bereich des Vergessens und des Lernens. Im folgenden Diagramm 3.1.1 ist die Kurve des Vergessens von Ebbinghaus dargestellt. Hierbei ist zu erkennen, dass nach einer Stunde bereits 50% des Wissens vergessen wurde. Nach einem Tag sind nur noch 20% des gelernten Wissens vorhanden.



**Abb. 3.1.1:** Ebbinghaus Vergessenskurve

Durch das regelmäßige Wiederholen des gelernten Wissens kann die Vergessenskurve verzögert werden. Im folgenden Diagramm 3.1.2 wird verdeutlicht, dass durch das Wiederholen des Wissens die Kurve des Vergessens verzögert wird. Hierbei besteht die Möglichkeit, dass nach der dritten Wiederholung das Wissen nach 60 Tagen noch zu 90% vorhanden ist. [39] Der Abstand der Wiederholungen bezieht sich hierbei darauf, dass erst wieder der Stoff gelernt wird, wenn dieser nur noch eine 90 prozentige Chance hat, vorhanden zu sein. Hierdurch wird klargestellt, dass das Wiederholen des Stoffes wichtig ist, um das Wissen langfristig beizubehalten.



**Abb. 3.1.2:** Ebbinghaus verzögerte Vergessenskurve durch Wiederholungen

Die Daumenregel für die optimale Pausenlänge beträgt 10 - 20% der Zeit, bis diese Information angewendet werden muss. Wenn beispielsweise eine Prüfung in 30 Tagen stattfindet, sollte der Stoff alle 3 - 6 Tage wiederholt werden. [3, S. 160] Dieser Fall kann beschrieben werden:

$$\text{Pausenlänge} = (30 \text{ Tage}) \times (0.1 - 0.2)$$

Die allgemeine Formel könnte man beschreiben als:

$$\text{Pausenlänge} = (\text{Zeit bis zur Anwendung}) \times (0.1 - 0.2)$$

Aus dieser Formel kann abgeleitet werden, dass zwischen fünf und zehn Wiederholungen notwendig sind, um das Wissen langfristig zu behalten. [3, S. 160] Mit diesen Informationen kann ein optimaler Lernplan erstellt werden, der die Wiederholungen automatisch berechnet und den Nutzer daran erinnert, wann er wieder lernen muss.

### 3.1.6 Tagging innerhalb der Lern-App

Um den Benutzer:in das Lernen so einfach wie möglich zu gestalten wurde entschieden, dass dieser die Möglichkeit hat, bestimmte Lerninhalte zu taggen. Das Verwenden von Tags wird in den folgenden Funktionen der App ermöglicht:

- Fragen erstellen
- Test erstellen
- Zusammenfassungen erstellen
- Lernhilfe bei beantworten fehlerhafter Fragen

Wenn ein Nutzer eine **Frage erstellen** möchte, dann muss die neu erstellte Frage mindestens einen Tag besitzen. Es ist möglich, dass eine Frage mehrere Tags enthalten darf.

Wenn ein Nutzer einen **Test erstellen** möchte, dann hat dieser die Option, eine Frage manuell oder mehrere Fragen anhand von Tags hinzufügen zu können. Hierbei werden alle Fragen, die den gewählten Tag beinhalten, dem Test zugeordnet. Dies vereinfacht das Erstellen eines Tests, da der Nutzer nicht jede Frage einzeln hinzufügen muss.

In der Lern-App kann der Nutzer mithilfe eines Editors eine **Zusammenfassung erstellen**. Dieser soll grundsätzlich die gleichen Funktionen wie andere gängige Texteditoren besitzen. Das heißt im genaueren Sinne, dass der Nutzer die Option hat, seinen Text zu formatieren, Bilder sowie Links einzufügen und den Text zu speichern. Zusätzlich soll es möglich sein, dass ein Text mit einem Tag versehen werden kann.

Wenn ein Anwender eine Frage hintereinander mehrfach falsch beantwortet, dann soll anhand des Tags eine **Lernhilfe** angeboten werden. Hierbei wird der Nutzer gefragt, ob dieser sich den Teil der Zusammenfassung nochmal anschauen möchte. Dabei ist es erforderlich, dass in der Zusammenfassung mindestens ein Tag dieser Frage vorhanden ist. Wenn der Nutzer sich die Zusammenfassung anschauen möchte, dann soll der Tag als Sprungmarke dienen und den Nutzer direkt zu dem zugehörigen Teil der Zusammenfassung weiterleiten.

### 3.1.7 Lernkategorien & Lernziele

Um dem Benutzer die einfache Kategorisierung und Planung seiner Lerninhalte zu ermöglichen gibt es die Möglichkeit Lernkategorien und Lernziele zu erstellen.

Im Reiter „Lernkategorien“ können Lernkategorien erstellt werden. Eine Lernkategorie besteht aus

- Zusammenfassungen
  - Eine Zusammenfassung muss vom Benutzer erstellt werden und bietet mithilfe von „Markdown“ (Siehe Markdown Guide) die Möglichkeit Informationen strukturiert darzustellen.
- Tests/Fragen
  - Eine Frage muss vom Benutzer erstellt werden und besitzt auch eine zugehörige Antwort. Jede Frage besitzt einen Fragentyp, der die Form der Frage definiert und wie diese den Benutzer abfragt. In der bisherigen Version der App werden nur Fragen vom Karteikarten-Fragentyp unterstützt. Eine Ansammlung von Fragen kann zu einem Test zusammengefasst und somit gruppiert abgefragt werden.

### 3.1.8 Gamification und ihr positiver Effekt auf das Lernen

Gamification bezeichnet die Anwendung von Spielelementen in nicht-spielerischen Kontexten, um Motivation und Engagement zu steigern [8]. Im Bildungsbereich hat die Gamification-Strategie zunehmend an Popularität gewonnen, da sie das Lernen attraktiver und interaktiver gestaltet [26].

Durch die Einbeziehung von Spielelementen wie Punkten, Leveln, Belohnungen, Herausforderungen und Leaderboards wird der Lernprozess in einen Wettbewerbs- und Unterhaltungskontext eingebettet [30]. Dies kann dazu beitragen, die Motivation der Lernenden zu erhöhen, was wiederum zu einer höheren Teilnahme und verbesserten Lernergebnissen führen kann [27].

Ein weiterer Vorteil der Gamification ist, dass sie durch Feedback und Belohnungen eine unmittelbare Rückmeldung zum Lernprozess ermöglicht. Dies kann dazu beitragen, die Selbstwirksamkeit der Lernenden zu erhöhen und ein Gefühl der Leistung und des Fortschritts zu fördern [31].

Trotz der potenziellen Vorteile ist es wichtig, die Gamification sorgfältig zu gestalten und durchzuführen. Eine schlecht durchdachte Gamification kann zu einer Überbetonung von Wettbewerb und Leistung führen, was wiederum zu Stress und Angst bei den Lernenden führen kann [35]. Daher ist es wichtig, ein ausgewogenes Gleichgewicht zwischen Wettbewerb und Zusammenarbeit, Belohnung und Herausforderung zu schaffen, um ein gesundes und effektives Lernumfeld zu gewährleisten [8].

Bei LearnAhead wird die Gamification mit einem Punktesystem umgesetzt. Beim Lernen bekommt der Benutzer direktes Feedback und erhöht somit seine eigene Punktzahl. Für ein Level werden 100 Punkte benötigt, das Level hat kein Limit. Punkte werden auf dem Profil angezeigt und mit folgenden Ereignissen verdient.

- Abschliessen eines Tests - 20 Punkte
- Einloggen an einem neuen Tag - zwischen 10 und 100 Punkte, je nach Anzahl der sequentiell aufeinanderfolgenden Einlogg-Tage
- Erreichen eines Lernziels - 20 Punkte
- Richtiges beantworten einer Frage -  $5 \cdot (\text{Anzahl wie oft man Frage falsch beantwortet hat})$  Punkte.

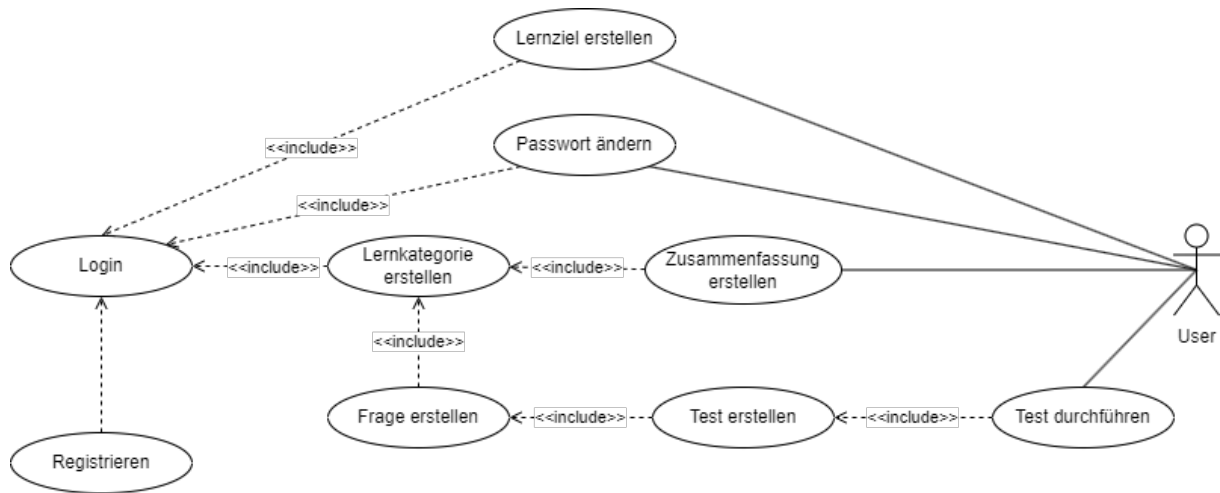
## 3.2 Pflichtenheft

Das Pflichtenheft ist analog zum Arbeitspaketplan in diesem Dokument nicht genauer beschrieben, sondern in Form eines Backlog auf Jira<sup>2</sup> aufgeführt. Dort kann entweder der Reiter „Roadmap“ oder der Reiter „Board“ aufgerufen werden.

---

<sup>2</sup><https://studienarbeitlernapp.atlassian.net/jira/software/projects/LER/boards/1>

### 3.3 Use Case Diagramm



**Abb. 3.3.1:** Use Case Diagramm

### 3.4 Use Case Beschreibung

<b>Name:</b>	<b>Login</b>
Ziel:	Anmelden mit bestehenden Logindaten in der App
Kategorie:	Primär
Vorbedingung:	Der Benutzer muss bereits einen Account erstellt haben
Nachbedingung:	Der Benutzer ist eingeloggt
Fehlerfälle:	<p>Falsche Logindaten</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass falsche Zugangsdaten verwendet wurden</li> <li>• Möglichkeit, das Kennwort über die E-Mail zurückzusetzen</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet die App und klickt auf einloggen
Beschreibung/ Erweiterungen:	Der Benutzer meldet sich an, um die verschiedenen Dienste der App zu verwenden
Alternativen:	Benutzer kann sich registrieren, sofern die E-Mail nicht schon registriert ist.

<b>Name:</b>	<b>Registrieren</b>
Ziel:	Anlegen eines neuen Accounts
Kategorie:	Primär
Vorbedingung:	E-Mail ist noch nicht mit einem anderen Account registriert
Nachbedingung:	Der Benutzer ist eingeloggt
Fehlerfälle:	<p>E-Mail bereits verwendet</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass E-Mail bereits verwendet wurde</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• Keine Zustandsänderung</li> <li>• Account wird nicht angelegt</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet die App und klickt auf Registrieren.
Beschreibung/ Erweiterungen:	Ein Benutzer möchte einen neuen Account erstellen.
Alternativen:	Login mit bestehendem Account.



<b>Name:</b>	<b>Lernziel erstellen</b>
Ziel:	Anlegen eines neuen Lernziels
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> <li>• User muss eingeloggt sein</li> <li>• Lernziel mit gleichem Namen ist noch nicht erstellt</li> </ul>
Nachbedingung:	Lernziel ist erstellt
Fehlerfälle:	<p>Lernziel mit gleichem Namen existiert bereits</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass dieses Lernziel bereits existiert</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> <li>• Lernziel wird nicht angelegt</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet Lernziel Reiter und klickt auf „Lernziel erstellen“
Beschreibung/ Erweiterungen:	Durch das Ausfüllen der Eingabefelder und anschließendes bestätigen, kann der User ein Lernziel erstellen. Aus sämtlichen Lernzielen wird daraufhin ein Lernplan erstellt.
Alternativen:	

<b>Name:</b>	<b>Lernkategorie erstellen</b>
Ziel:	Anlegen einer neuen Lernkategorie
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> <li>• User muss eingeloggt sein</li> <li>• Lernkategorie mit gleichem Namen ist noch nicht erstellt</li> </ul>
Nachbedingung:	Lernkategorie ist erstellt
Fehlerfälle:	<p>Lernkategorie mit gleichem Namen existiert bereits</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass diese Lernkategorie bereits existiert</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> <li>• Lernkategorie wird nicht angelegt</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet Lernkategorie Reiter und klickt auf „Lernkategorie erstellen“
Beschreibung/ Erweiterungen:	Durch das Ausfüllen der Eingabefelder und anschließendes bestätigen, kann der User eine Lernkategorie erstellen. In einer Lernkategorie können Zusammenfassungen, Fragen und Tests erstellt werden.
Alternativen:	

<b>Name:</b>	<b>Zusammenfassung erstellen</b>
<b>Ziel:</b>	Anlegen einer neuen Zusammenfassung
<b>Kategorie:</b>	Primär
<b>Vorbedingung:</b>	<ul style="list-style-type: none"> <li>• User muss eingeloggt sein</li> <li>• Zusammenfassung mit gleichem Namen ist noch nicht in der gleichen Lernkategorie erstellt</li> </ul>
<b>Nachbedingung:</b>	Zusammenfassung für Lernkategorie ist erstellt
<b>Fehlerfälle:</b>	<p>Zusammenfassung mit gleichem Namen existiert bereits in dieser Lernkategorie</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass diese Zusammenfassung bereits existiert</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> <li>• Zusammenfassung wird nicht angelegt</li> </ul>
<b>Akteure:</b>	User
<b>Auslösendes Event:</b>	Benutzer öffnet eine Lernkategorie, klickt auf „Zusammenfassungen“ und danach auf „Zusammenfassung erstellen“
<b>Beschreibung/ Erweiterungen:</b>	Durch das Ausfüllen der Eingabefelder und anschließendes bestätigen, kann der User eine Zusammenfassung erstellen. In einer Zusammenfassung können Inhalte eingefügt und formatiert werden.
<b>Alternativen:</b>	

<b>Name:</b>	<b>Frage erstellen</b>
<b>Ziel:</b>	Anlegen einer neuen Frage
<b>Kategorie:</b>	Primär
<b>Vorbedingung:</b>	<ul style="list-style-type: none"> <li>• User muss eingeloggt sein</li> <li>• Frage mit gleichem Index ist noch nicht in der gleichen Lernkategorie erstellt</li> </ul>
<b>Nachbedingung:</b>	Frage für Lernkategorie ist erstellt
<b>Fehlerfälle:</b>	<p>Frage mit gleichem Namen existiert bereits in dieser Lernkategorie</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass Frage mit gleicher Bezeichnung bereits existiert</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> <li>• Frage wird nicht angelegt</li> </ul>
<b>Akteure:</b>	User
<b>Auslösendes Event:</b>	Benutzer öffnet eine Lernkategorie, klickt auf „Tests / Fragen“ und danach auf „Frage erstellen“
<b>Beschreibung/ Erweiterungen:</b>	Durch das Ausfüllen der Eingabefelder und anschließendes bestätigen, kann der User eine Frage erstellen. Eine Frage kann unterschiedliche Vorlagen (beispielsweise Karteikarte) besitzen, jedoch gibt es immer eine Meldung (Text der Frage) und eine gültige Antwort.
<b>Alternativen:</b>	

<b>Name:</b>	<b>Test erstellen</b>
Ziel:	Anlegen eines neuen Tests
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> <li>• User muss eingeloggt sein</li> <li>• Test mit gleichem Namen ist noch nicht in der gleichen Lernkategorie erstellt</li> </ul>
Nachbedingung:	Test für Lernkategorie ist erstellt
Fehlerfälle:	<p>Test mit gleichem Namen existiert bereits in dieser Lernkategorie</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass Test mit gleicher Bezeichnung bereits existiert</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> <li>• Test wird nicht angelegt</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet eine Lernkategorie, klickt auf „Tests / Fragen“ und danach auf „Test erstellen“
Beschreibung/ Erweiterungen:	Durch das Ausfüllen der Eingabefelder und anschließendes bestätigen, kann der User einen Test erstellen. Ein Test kann mehrere Fragen besitzen. Wenn ein Test gestartet wird, bekommt der User nach abschließen des Tests eine Meldung, wie viele und welche Fragen er richtig beantwortet hat.
Alternativen:	

### 3.5.1 Datenbankmodell



### 3.5.2 Data Dictionary

#### User

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Username	string	-	Nein	-	-	Der Benutzername des Nutzers
E-Mail	string	-	Nein	-	-	Die E-Mail-Adresse des Nutzers
ProfileImageURL	string	-	Ja	Null	-	Der Link des Profilbilds welches im Firebase Storage gespeichert ist
RegisterDate	timestamp	-	Nein	-	-	Das Datum an dem sich der Nutzer registriert hat.
LearningStreak	number	-	Ja	0	-	Dies gibt die Anzahl an, wie viele Tage der Nutzer aufeinander gelernt hat.
AchievedGoals	number	-	Ja	0	-	Dies gibt die Anzahl an, wie viele Lernziel der Nutzer erreicht hat.
CurrentPoints	number	-	Ja	0	-	Die aktuelle Level Punkte des Nutzers
LearningCategories	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle LearningCategories, wo alle Lernkategorien drin sind, die der Nutzer erstellt hat.
Goals	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle Goals, wo alle Lernziele drin sind, die der Nutzer erstellt hat.

**Goal**

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Name	string	-	Nein	-	-	Der Name des Lernziels
Description	string	-	Ja	Null	-	Dies soll als Beschreibung des Lernziels gelten. Hier können z.B. die relevanten Themen aufgelistet werden.
Status	string	-	Ja	ToDo	-	Dies gibt den aktuellen Status des Lernziels an. Es gibt die folgende Status: - ToDo - In Progress - Done
StartDate	timestamp	-	Ja	Server Timestamp	-	Dies gibt an, wann das Lernziel startet
EndDate	timestamp	-	Nein	-	-	Dies gibt an, bis wann das Lernziel abgeschlossen sein soll

**LearningCategory**

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Name	string	-	Nein	-	-	Der Name der Lernkategorie
CreateDate	timestamp	-	Ja	Server Timestamp	-	Dies gibt an, wann die Lernkategorie erstellt wurde
Goals	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle Goal, wo alle Lernziele drin sind, die der Nutzer erstellt hat.
Questions	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle Question, wo alle Fragen drin sind, die der Nutzer erstellt hat.
Summaries	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle Summary, wo alle Zusammenfassungen drin sind, die der Nutzer erstellt hat.
Tests	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle Test, wo alle Tests drin sind, die der Nutzer erstellt hat.

**Summary**

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Name	string	-	Nein	-	-	Dies ist der Name einer Zusammenfassung
CreateDate	timestamp	-	Nein	-	-	Dies ist das Erstelldatum einer Zusammenfassung
Content	string	-	Ja	Null	-	Dies ist der Inhalt einer Zusammenfassung im Raw-Format



**Test**

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Name	string	-	Nein	-	-	Dies ist der Name eines Tests
Questions	map	-	Ja	Null	-	Dies ist eine gemappte Liste zu der Tabelle Question, wo alle Fragen drin sind, die der Nutzer erstellt hat.

**Question**

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Question	string	-	Nein	-	-	Die Frage der Frage
Answer	string	-	Nein	-	-	Die Antwort auf die Frage
Type	number	-	Nein	-	-	Dies soll den Typ einer Frage angeben: z.B. Type = 0 = Karteikarten Type = 1 = Multiple Choice
LastTest	bool	-	Ja	Null	-	Dies gibt an ob die Frage beim letzten Mal richtig beantwortet wurde
WrongCounter	number	-	Ja	0	-	Dies gibt an wie oft die Frage hintereinander falsch beantwortet wurde. Wird diese dann richtig beantwortet setzt sich der Counter auf 0 zurück
Tags	map	-	Nein		-	Dies ist eine gemappte Liste zu der Tabelle Tags, wo alle Tags drin sind, die der Nutzer erstellt hat.

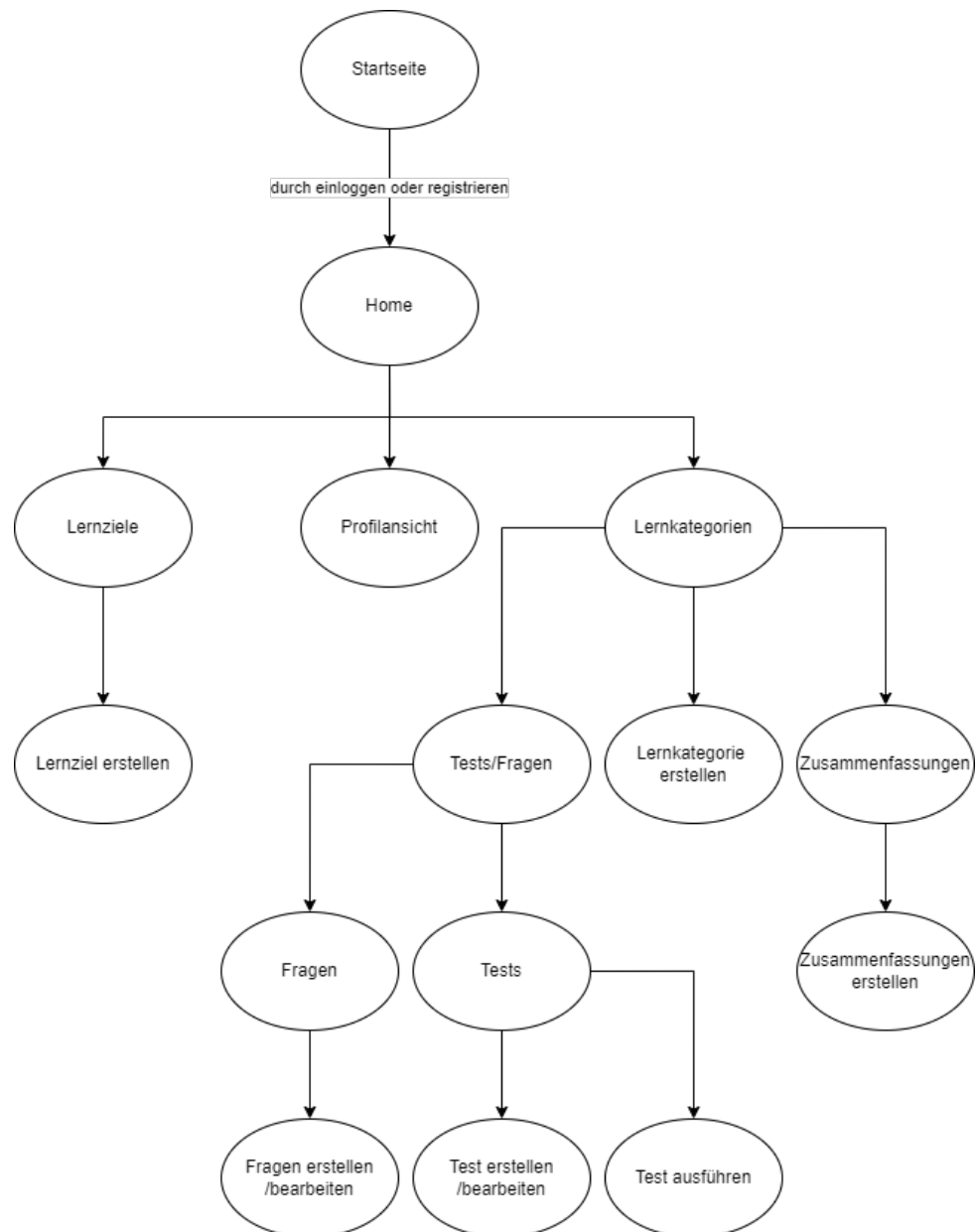
**Tag**

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
Name	string	-	Nein	-	-	Der Name des Tags

## 3.6 HMI

### 3.6.1 Seitenhierarchie

Innerhalb der Seitenhierarchie wird dargestellt, wie man in der App navigieren kann.



**Abb. 3.6.1:** Die Seitenhierarchie in LearnAhead

## 3.6.2 UI-Mockups

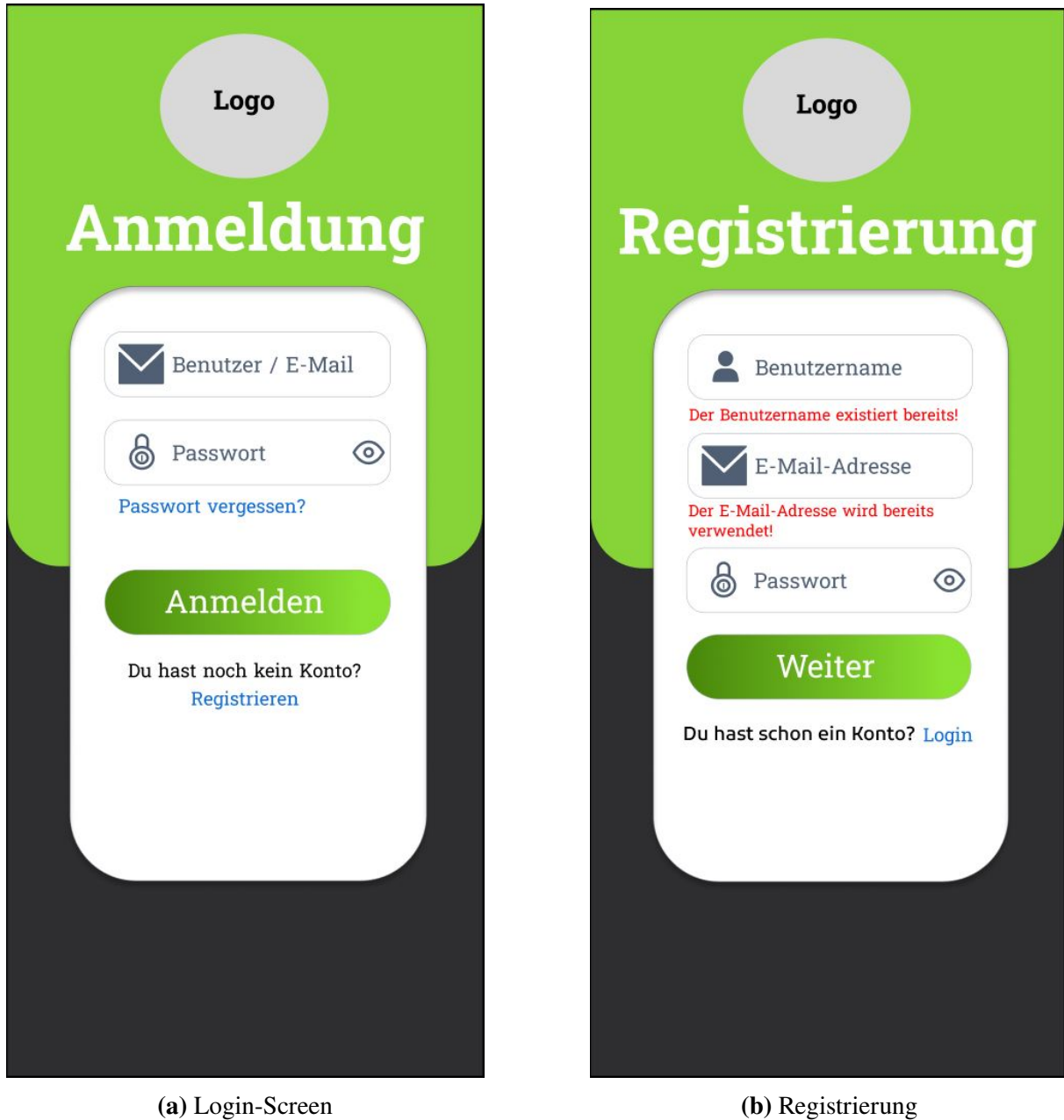
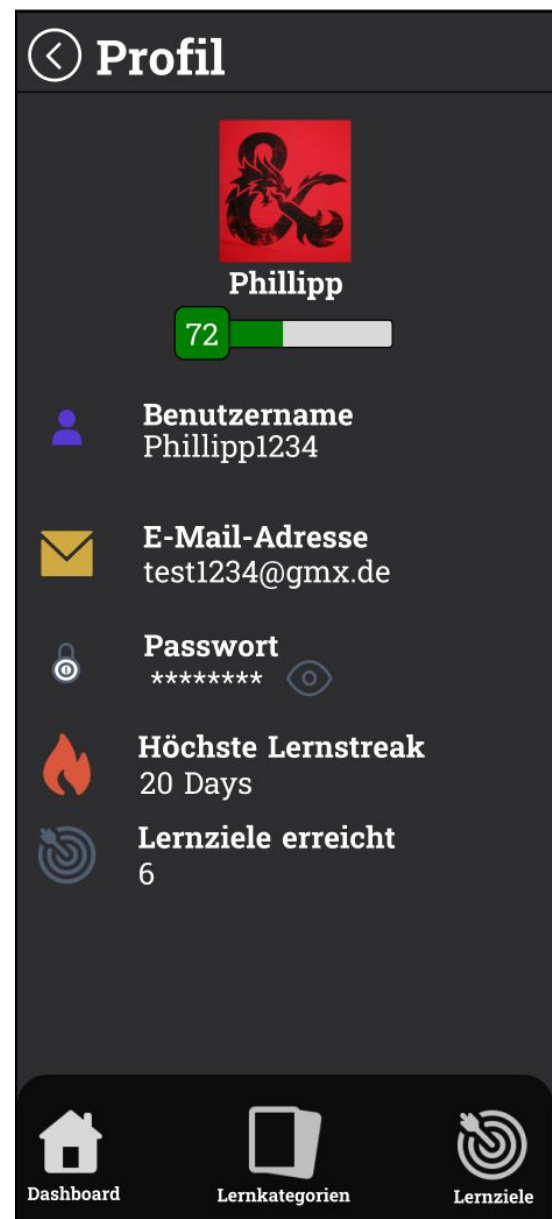


Abb. 3.6.2: Login und Registrierung

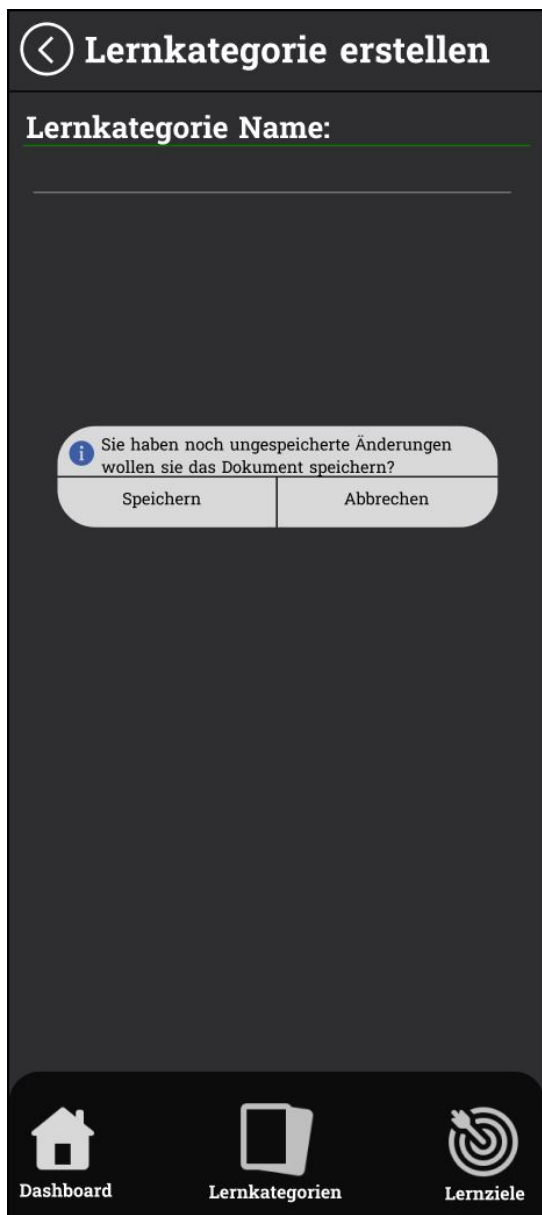


(a) Home-Screen



(b) Profilansicht

Abb. 3.6.3: Home-Screen and Profilansicht

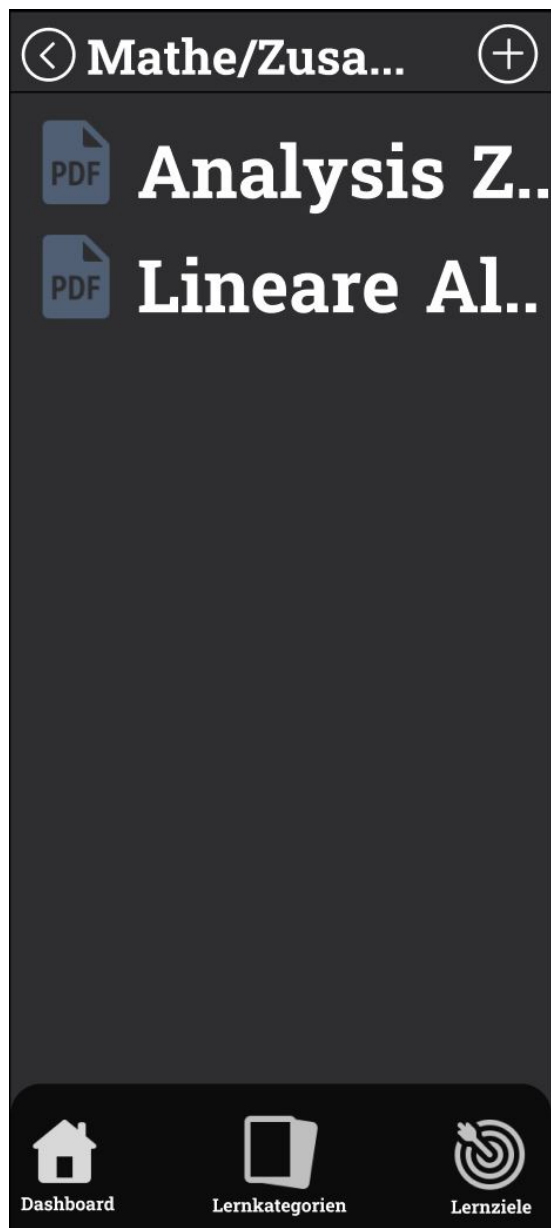


(a) Erstellen einer Lernkategorie

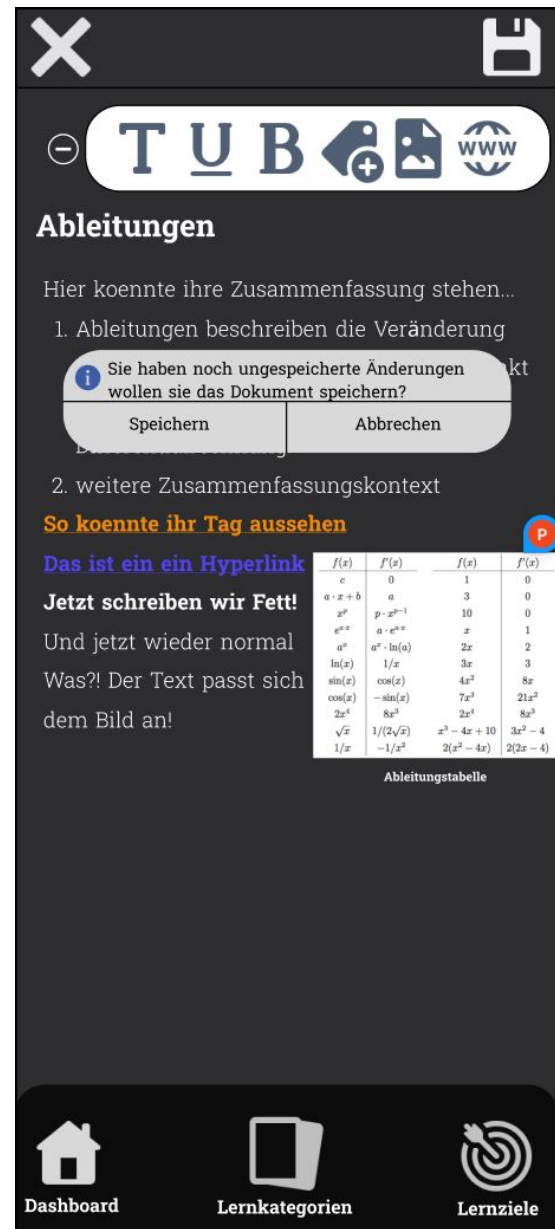


(b) Ansicht Lernkategorien

Abb. 3.6.4: Erstellen einer Lernkategorie and Ansicht Lernkategorien

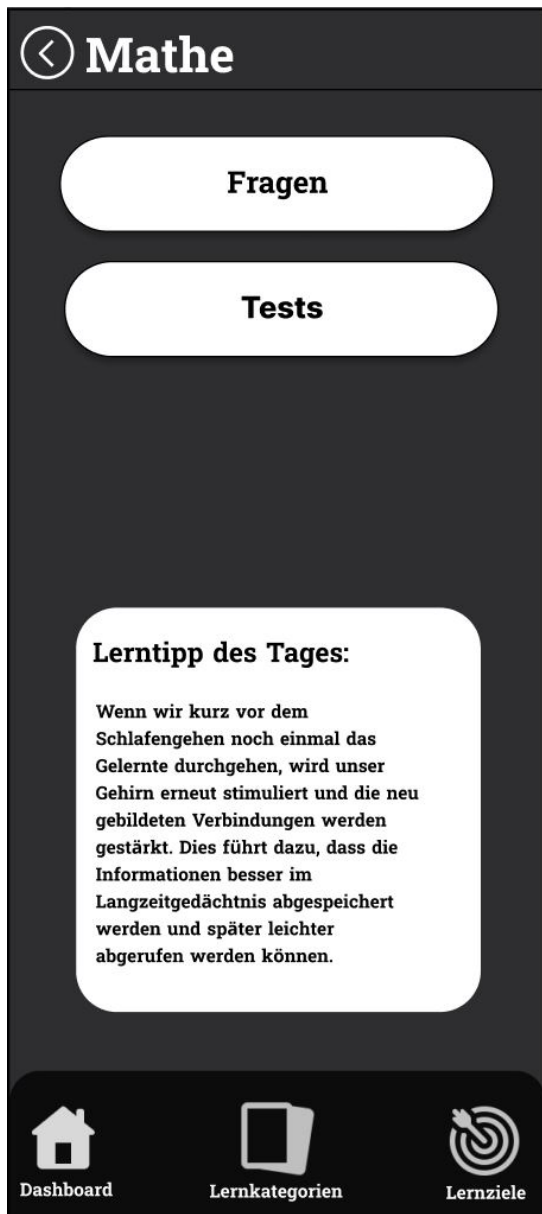


(a) Ansicht Zusammenfassungen

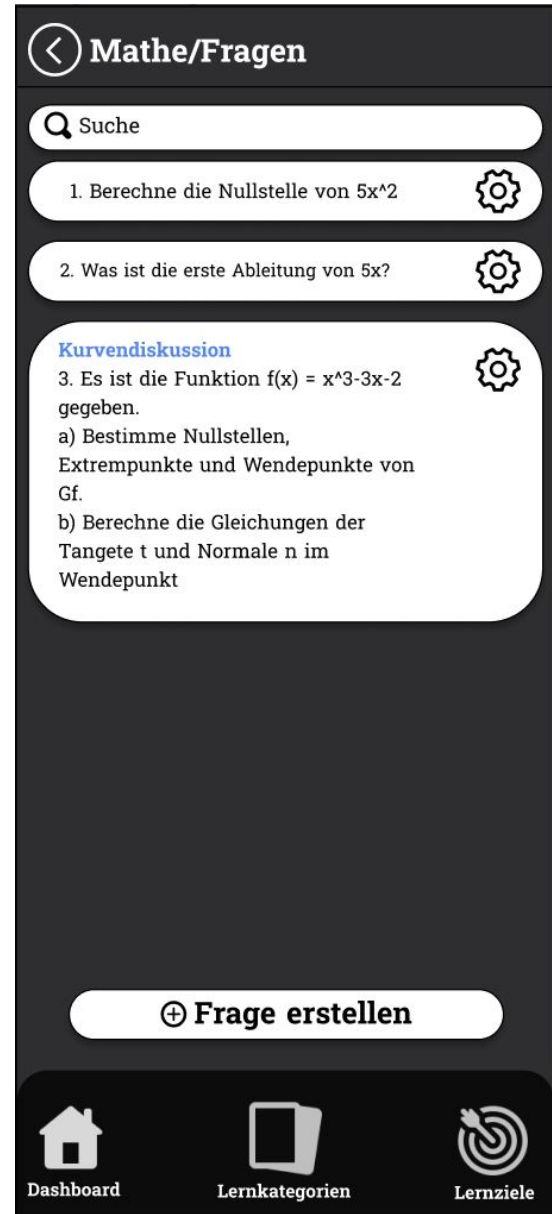


(b) Erstellen einer Zusammenfassung

Abb. 3.6.5: Ansicht Zusammenfassungen and Erstellen einer Zusammenfassung

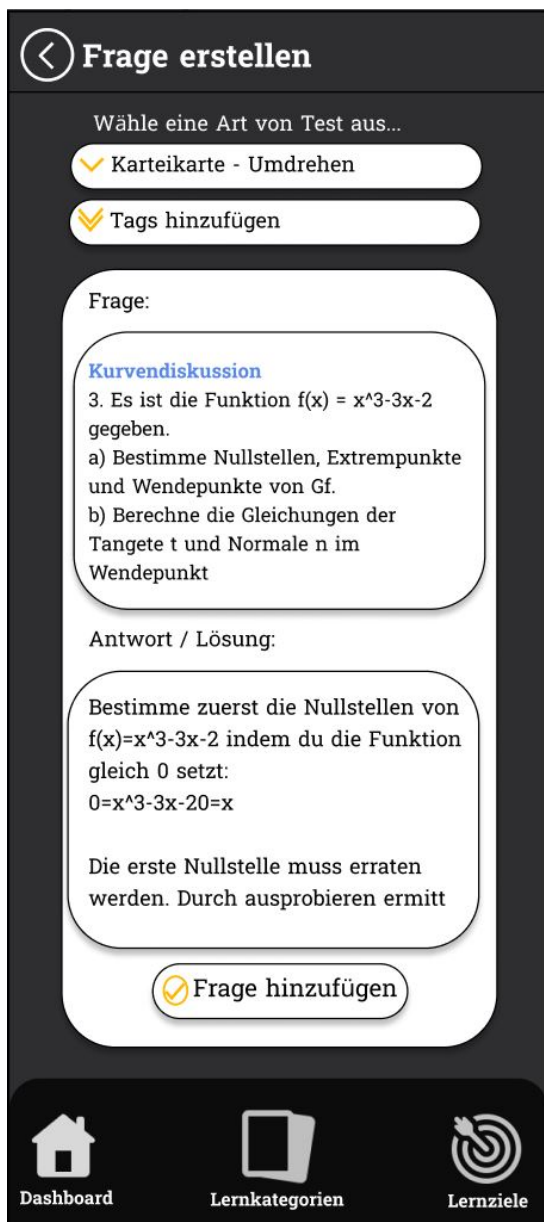


(a) Ansicht Fragen und Tests

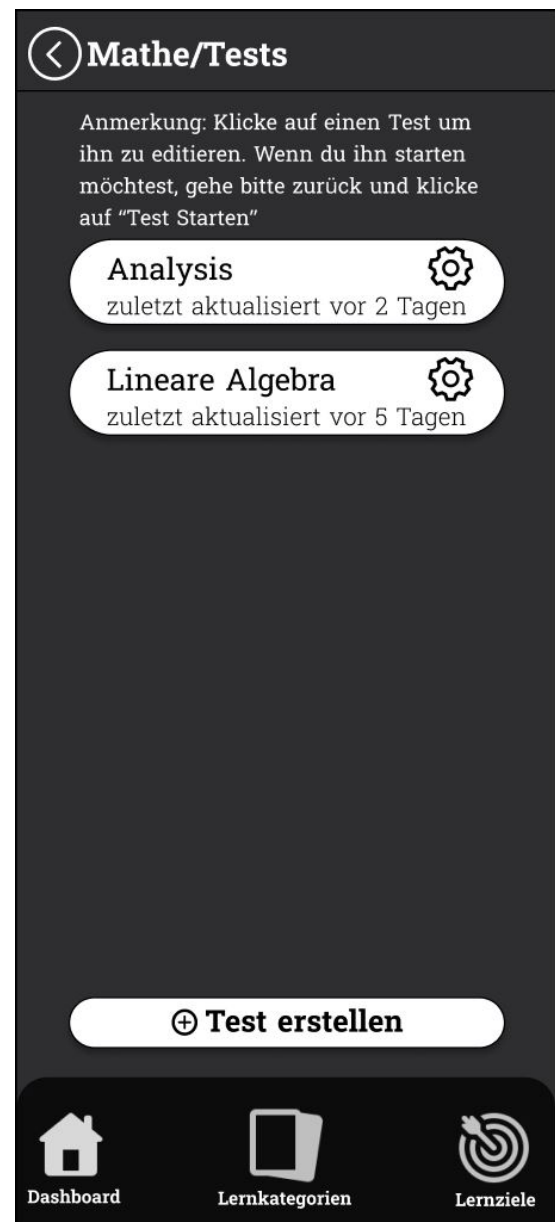


(b) Ansicht Fragen

Abb. 3.6.6: Ansicht Fragen und Tests and Ansicht Fragen



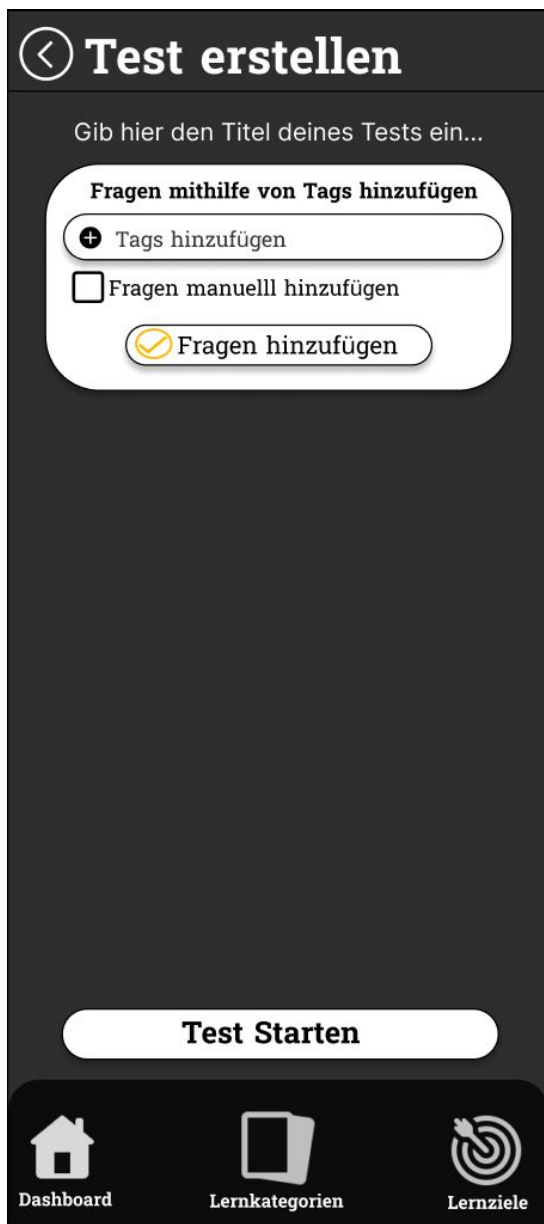
(a) Erstellen einer Frage



(b) Ansichts Tests

Abb. 3.6.7: Erstellen einer Frage and Ansichts Tests





(a) Erstellen eines Tests



(b) Ansicht einer Frage innerhalb eines Tests

Abb. 3.6.8: Test




(a) Ansicht einer Antwort innerhalb eines Tests



(b) Ansicht Lernziele

Abb. 3.6.9: Test


 **Lernziel erstellen**


**Lernzielname:**


**Start Datum:**

**Ziel Datum:**

**Beschreibung:**

 Dashboard

 Lernkategorien

 Lernziele

**Abb. 3.6.10:** Erstellen eines Lernziels

### 3.7 Flow-Chart-Diagramm

Die Interaktion des Users mit der App wird in dem Flow-Chart-Diagramm grob dargestellt. Hierbei ist noch zu erwähnen, dass die Farbe Grün für den Anfang und das Ende steht. Die Farbe Blau steht für die wichtigsten Seiten, die der User besuchen kann. Die Farbe Orange steht für die verschiedenen Aktionen, die der User ausführen kann. Die Farbe Gelb steht für die hauptsächliche Interaktion mit der App (z.B. das Wechseln der Seiten).

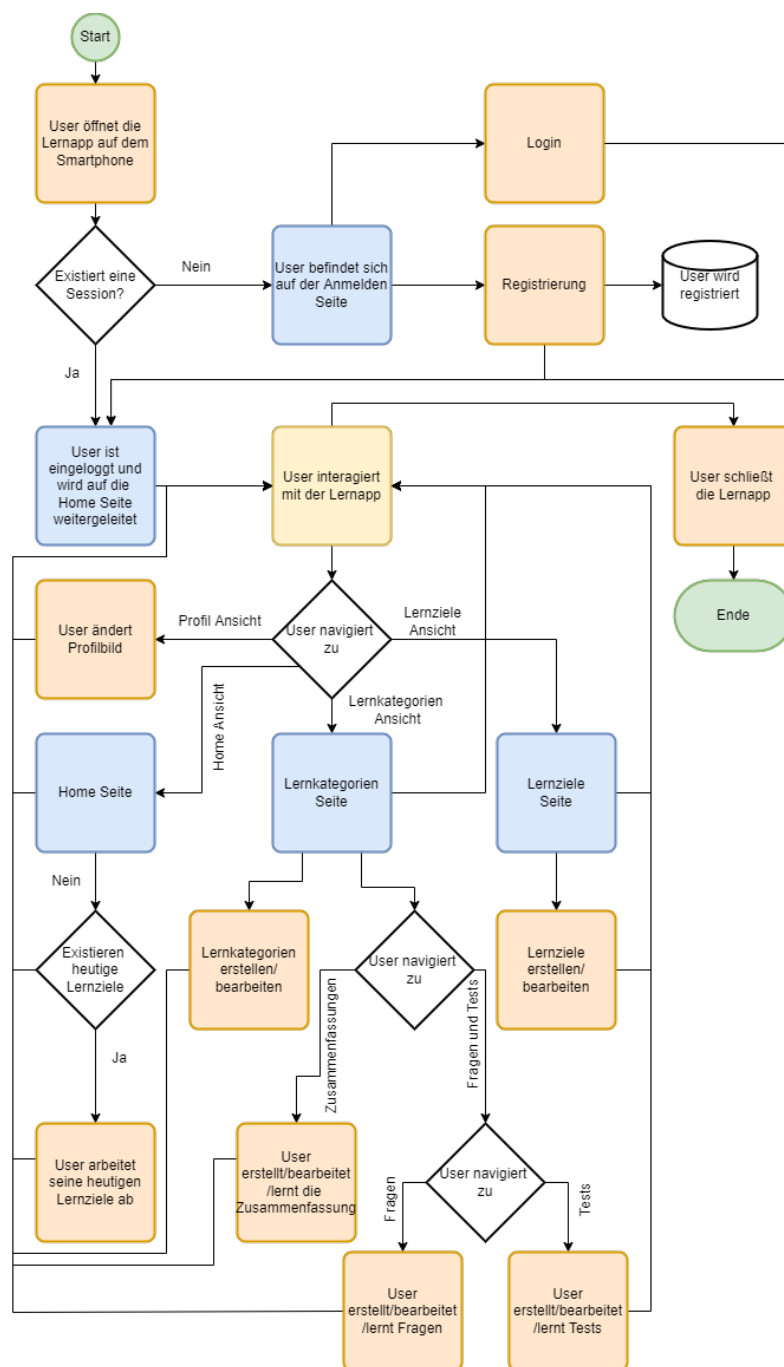


Abb. 3.7.1: Flow-Chart-Diagramm

## 4 Entwurf

### 4.1 Auswahl und Begründung des Datenbankkonzepts

Die Lern-App „LearnAhead“ benötigt eine Datenbank, um die Lerninhalte des Benutzers speichern zu können. Die Datenbank soll hierbei die folgenden Anforderungen erfüllen:

- Es soll möglich sein, Bilder einfach zu speichern und abzurufen.
- Die Datenbank soll gut skalierbar sein, um auch bei vielen Benutzern eine gute Performance zu gewährleisten.
- Es müssen gute Frameworks zur Anbindung für die Sprache Kotlin existieren.
- Die Datenbank soll in der Cloud gehostet werden, um die Wartungskosten zu minimieren.
- Die Datenbank soll möglichst kostenlos sein.

Hierbei gab es die Entscheidung zwischen einer relationalen und einer NoSQL Datenbank. Da diese für die Speicherung von Bildern zuständig ist, ist eine NoSQL Datenbank die bessere Wahl. In diesem Zusammenhang kam der Anbieter „Firebase“ in Frage. Dieser bietet zwei unterschiedliche Datenbanken an: „Cloud Firestore“ und „Realtime Database“. Realtime Database wird genutzt, wenn die Datenbank in Echtzeit synchronisiert werden soll. Da dies bei der Lern-App nicht notwendig ist, wurde sich für Cloud Firestore entschieden. Cloud Firestore baut auf den Erfolgen von Realtime Database auf und bietet zusätzlich eine bessere Skalierbarkeit und ermöglicht zudem schnellere Abfragen. [40] Für die Speicherung von Inhalten gibt es den Firebase Storage, welcher eine einfache Möglichkeit bietet, Inhalte wie z.B. Bilder oder Videos zu speichern.

Firebase bietet eine gute Anbindung für Kotlin (siehe 4.2). Zusätzlich ist Firebase kostenlos, solange die Datenbank nicht zu groß wird. Hierbei kann eine monatliche Anzahl von 50.000 Nutzern, 20.000 Schreibvorgängen und 50.000 Lesevorgängen kostenlos genutzt werden. Der Speicherplatz für den Firebase Storage beträgt 1 GB, welcher für die Lern-App ausreichend ist. [14]

### 4.2 Auswahl von wichtigen Bibliotheken/Frameworks

In diesem Abschnitt werden die wichtigsten in LearnAhead verwendeten Bibliotheken und Frameworks erläutert. Es wird kurz darauf eingegangen was genau die/das Bibliothek/Framework tut und wieso sie über andere gewählt wurde.

### 4.2.1 DaggerHilt Dependency Injection Framework

DaggerHilt ist ein Framework für Dependency Injection (DI), das speziell für Android entwickelt wurde [19]. Es basiert auf Dagger, einem weit verbreiteten und mächtigen DI-Framework für Java [20].

Dependency Injection ist ein Programmierparadigma, das dabei hilft, Objekte effizient zu erzeugen und zu verwalten. Mit Hilfe von DaggerHilt können Abhängigkeiten zentral verwaltet und bei Bedarf ausgetauscht werden. Das ermöglicht bessere Testbarkeit und modulareren Code, da jede Klasse sich nur auf ihre Aufgaben konzentrieren muss und ihre Abhängigkeiten von außen erhält [19].

Die Wahl von DaggerHilt für diese Arbeit beruht auf seiner tiefen Integration in das Android-Ökosystem. DaggerHilt ist in der Lage, Abhängigkeiten über die Grenzen von Aktivitäten, Fragmenten und Services hinweg zu verwalten und ist dabei eng mit dem Lebenszyklus von Android-Anwendungen verbunden [20].

Darüber hinaus unterstützt DaggerHilt das Konzept der Compile-Time Injection. Das bedeutet, dass Abhängigkeiten zur Kompilierzeit aufgelöst werden, was die Fehlererkennung verbessert und Laufzeitfehler minimiert [19].

### 4.2.2 Markwon: Eine Markdown-Bibliothek für Android

Markwon ist eine leistungsfähige Bibliothek, die speziell entwickelt wurde, um die Anzeige und Formatierung von Markdown-Inhalten auf Android zu erleichtern [36]. Markdown ist eine leichtgewichtige und benutzerfreundliche Markup-Sprache, die oft für die Formatierung von Text in Softwareentwicklungsprojekten, Dokumentationen und im Web verwendet wird [25].

Bei LearnAhead wird der Input mit einem in Markwon enthaltenen Editor eingegeben und daraufhin in einem Android freundlichen Format in einem TextView angezeigt.

Die Markwon-Bibliothek bietet eine voll funktionsfähige Markdown-Rendering-Engine für Android, die in einer TextView-Komponente arbeitet. Sie unterstützt alle gängigen Markdown-Syntaxelemente und bietet zusätzlich Funktionen wie Syntaxhervorhebung für Codeblöcke, Inline-Latex-Rendering und vieles mehr [36].

Für diese Arbeit wurde Markwon aufgrund seiner hohen Flexibilität und seiner vielfältigen Funktionen gewählt. Durch die Nutzung der Markwon-Bibliothek können komplexe, formatierte Textinhalte in der Android-Anwendung auf einfache und effiziente Weise angezeigt werden [36].

#### **4.2.3 Glide: Ein Framework zum Laden von Bildern in Android**

Glide ist ein effizientes Open-Source-Framework zur Handhabung und Darstellung von Bildern und Videos in Android-Anwendungen [4]. Es wurde von Bumptech entwickelt und ist bekannt für seine Leistung, insbesondere bei der Scroll-Leistung in Listen oder Rastern von Bildern.

In LearnAhead wird Glide beispielsweise dafür genutzt, um die Profilbilder der Nutzer zu laden. Eine der Hauptfunktionen von Glide ist das asynchrone Laden und Speichern von Bildern aus verschiedenen Quellen, beispielsweise aus dem Internet, ohne den Haupt-User-Interface (UI)-Thread zu blockieren. Dies führt zu einer glatteren Benutzererfahrung [4]. Zudem bietet Glide auch Funktionen wie das Zuschneiden und Transformieren von Bildern, das Cachen von Bildern für eine schnellere Wiederverwendung und die automatische Verwaltung des Lebenszyklus [5].

Die Wahl von Glide für diese Arbeit basiert auf seinen umfangreichen Funktionen, seiner Leistung und seiner Integration in das Android-Ökosystem. Mit Glide können komplexe Bildverarbeitungsanforderungen erfüllt werden, während gleichzeitig die Performance und die Benutzererfahrung der App optimiert werden [4].

#### **4.2.4 Android Navigation: Ein Framework zur Navigation in Android-Anwendungen**

Android Navigation ist eine Bibliothek, die von Google entwickelt wurde und das Navigieren zwischen den verschiedenen Bildschirmen einer Android-Anwendung erleichtert [24]. Es ist Teil der Android Jetpack-Bibliotheken, einer Sammlung von Bibliotheken, Tools und Leitlinien, die robuste und wartungsfreundliche Android-Apps unterstützen [18].

In LearnAhead wird die Bibliothek für die Navigation zwischen den verschiedenen Fragments (Teil-Bildschirmen) der App genutzt. Mit dem Android Navigation Framework können Entwickler visuell oder programmgesteuert einen Navigationspfad zwischen den verschiedenen Komponenten einer App erstellen, z. B. zwischen verschiedenen Aktivitäten oder Fragmenten. Die Bibliothek sorgt dafür, dass die Benutzeroberfläche in einem konsistenten und vorhersehbaren Zustand bleibt und beinhaltet auch Unterstützung für erweiterte Navigationsszenarien wie tiefe Links und Navigationsleisten [24].

Die Wahl dieses Frameworks für diese Arbeit beruht auf seiner Integration in das Android-Ökosystem und seinen umfangreichen Funktionen zur Handhabung von App-Navigationsszenarien. Die Verwendung des Android Navigation Frameworks erleichtert die Erstellung von strukturierten und benutzerfreundlichen Navigationsflüssen in der App [24].

#### **4.2.5 Firebase Android SDK: Eine Bibliothek zur Integration von Firebase in Android-Anwendungen**

Firebase Android Software Development Kit (SDK) ist eine Bibliothek, die eine Anbindung an Firebase und dessen verschiedenen Dienste ermöglicht [21]. Firebase, entwickelt von Google, ist eine Plattform, die verschiedene Backend-Services wie Datenbanken, Authentifizierung, Cloud-Funktionen, Hosting und mehr bietet [22].

In LearnAhead wird die Bibliothek genutzt, um die Verbindung zur Firebase-Datenbank herzustellen. Die Firebase Android SDK bietet Funktionen zur nahtlosen Integration dieser Dienste in Android-Anwendungen. Es bietet eine Reihe von Application Programming Interfaces (APIs) und Konfigurationstools, die es Entwicklern ermöglichen, auf leistungsfähige Weise auf Firebase-Dienste zuzugreifen und diese in ihren Apps zu nutzen [21].

Die Entscheidung, Firebase Android SDK für diese Arbeit zu nutzen, basiert auf seiner umfangreichen Integration in das Firebase-Ökosystem und seiner Fähigkeit, robuste und skalierbare Backend-Services bereitzustellen. Die Verwendung von Firebase Android SDK erleichtert den Zugriff auf und die Verwaltung von Firebase-Diensten innerhalb der Android-Anwendung [21].

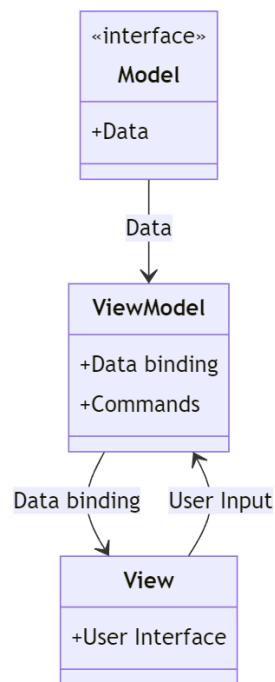
### **4.3 Wahl des Architektur-Patterns: Model-View-ViewModel (MVVM)**

Bei der Entwicklung von Software ist die Wahl des richtigen Architektur-Patterns von entscheidender Bedeutung. Für diese Arbeit wurde das Model-View-ViewModel (MVVM) Pattern gewählt, welches sich in den letzten Jahren zu einem der Standard-Architektur-Patterns für Android-Entwicklung etabliert hat [23].

MVVM teilt die Anwendungslogik in drei miteinander verbundene Komponenten auf: das Model, die View und das ViewModel [33]. Das Model repräsentiert die Daten und die Geschäftslogik der Anwendung. Die View ist verantwortlich für die Darstellung der Daten und die Interaktion mit dem Benutzer. Das ViewModel fungiert als Bindeglied zwischen Model und View und stellt die Daten für die View zur Verfügung und reagiert auf Benutzeraktionen [23].



MVVM hat mehrere Vorteile, darunter eine klare Trennung von Anliegen, verbesserte Testbarkeit und eine bessere Unterstützung für reaktive Programmierung [23]. Diese klare Trennung der Anliegen ermöglicht es, dass verschiedene Teile der Anwendung unabhängig voneinander entwickelt und getestet werden können, was zu einer besseren Code-Qualität und Wartbarkeit führt [33].



**Abb. 4.3.1:** MVVM Diagramm

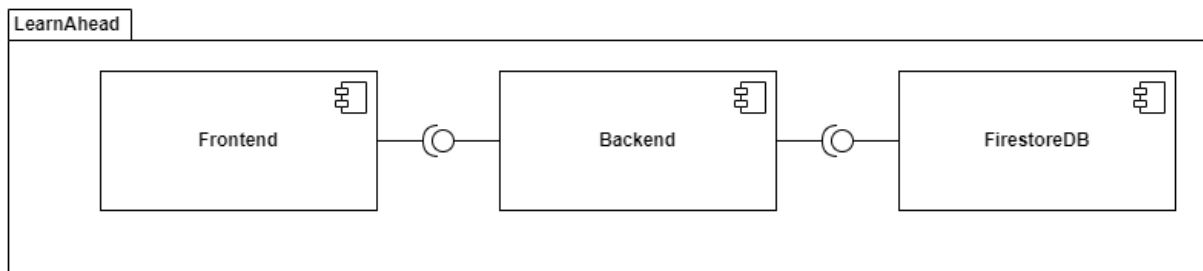
Das in Abbildung 4.3.1 dargestellte Diagramm zeigt das Model-View-ViewModel (MVVM) Designmuster. Dieses Muster wird häufig in der Softwareentwicklung verwendet, insbesondere in Anwendungen mit grafischer Benutzeroberfläche. Es besteht aus drei Hauptkomponenten:

- **Model:** Dies repräsentiert die Daten und die Geschäftslogik der Anwendung. Es ist unabhängig von der Benutzeroberfläche und enthält keine Informationen darüber, wie die Daten dargestellt werden.
- **View:** Dies ist die Benutzeroberfläche der Anwendung. Sie zeigt die Daten an, die vom Model bereitgestellt werden, und sendet Benutzereingaben an das ViewModel.
- **ViewModel:** Dies ist das Bindeglied zwischen Model und View. Es holt Daten vom Model, führt eventuell notwendige Transformationen durch und stellt sie der View zur Verfügung. Außerdem verarbeitet es Benutzereingaben von der View und führt entsprechende Aktionen im Model aus.

## 4.4 Software-Komponenten

Das System wird zunächst als Gesamtkomposition dargestellt. Anschließend werden die Subsysteme einzeln dargestellt.

### 4.4.1 Gesamtkomposition



**Abb. 4.4.1:** Gesamtkomposition

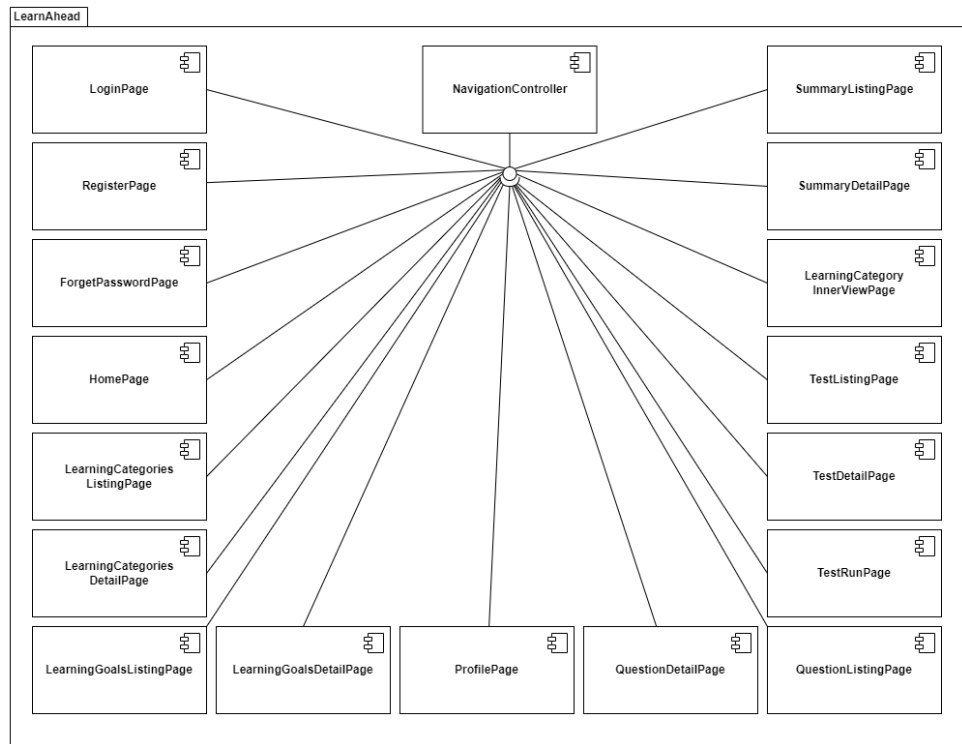
Die Gesamtkomposition besteht aus den folgenden Subsystemen:

- **Frontend:** Das Frontend ist für die Darstellung der Benutzeroberfläche zuständig. Es kommuniziert mit dem Backend, um Daten anzuzeigen und Eingaben an dieses weiterzuleiten.
- **Backend:** Das Backend ist für die Verarbeitung und Validierung der Daten zuständig. Es kommuniziert mit der FireStore Datenbank, um Daten abzurufen und zu speichern.
- **FireStoreDB** Die FireStoreDB Komponente ist die Datenbank-Komponente.

Näheres hierzu im Deployment Diagramm 4.5 und Klassen Diagramm 4.6.

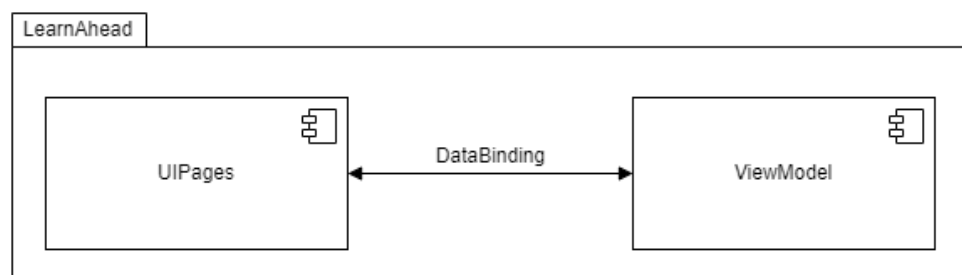
### 4.4.2 Frontend

Es gibt zwei Diagramme für das Frontend. Das Diagramm 4.4.2 zeigt die Navigation zwischen den einzelnen Pages.



**Abb. 4.4.2:** Frontend UI Navigation

Das Diagramm 4.4.3 zeigt die Kommunikation zwischen den einzelnen Pages und dem ViewModel. Hierbei ist zu beachten, dass dieses Diagramm die Kommunikation zwischen den einzelnen Pages und dem ViewModel im Allgemeinen zeigt. Das bedeutet, dass jede Page ein zugehöriges ViewModel hat. Die Kommunikation zwischen den einzelnen Pages und dem ViewModel ist prinzipiell immer gleich. Weitere Details sind in Kapitel 4.3 zu finden.



**Abb. 4.4.3:** Frontend UI Kommunikation mit ViewModel

### 4.4.3 Backend

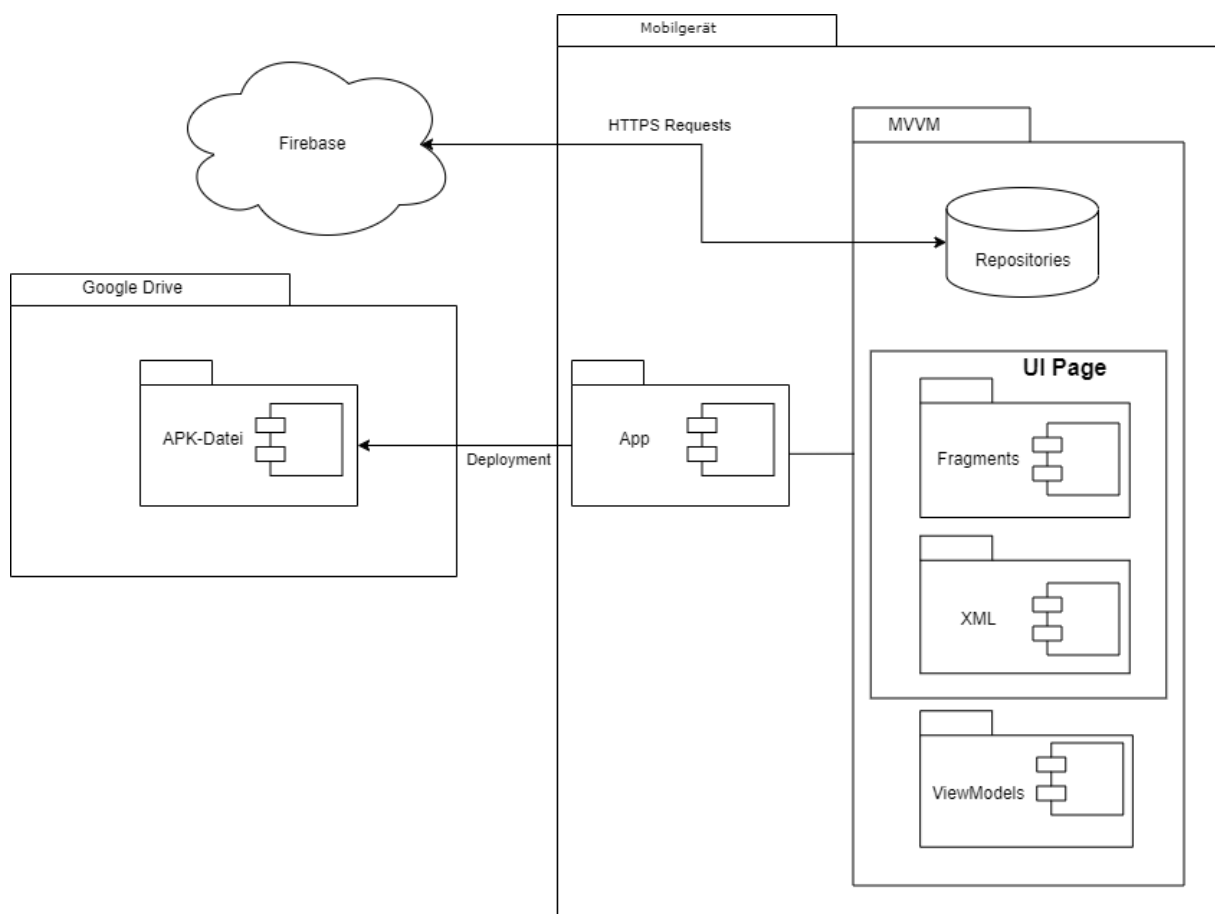
Das Diagramm 4.4.4 zeigt die Komposition des Backends. Hierbei ist zu beachten, dass dieses Diagramm die Komposition des Backends im Allgemeinen zeigt. Jedes Objekt, welches in der Datenbank gespeichert wird, hat ein zugehöriges Repository/Model. Die Kommunikation zwischen den verschiedenen Repositories und Modellen verläuft grundsätzlich immer auf die gleiche Weise. Jedes Repository baut auf einem Interface auf, welches die grundlegenden Funktionen für die Kommunikation mit der Datenbank definiert. Die Kommunikation mit der Datenbank erfolgt über das Repository.

**Abb. 4.4.4:** Backend Komposition

## 4.5 Deployment Diagramm

Ein Deployment-Diagramm ist eine Art von Diagramm, das in der Unified Modeling Language (UML) verwendet wird. Es bietet eine grafische Darstellung der physischen Verteilung von Artefakten auf Knoten. Ein Artefakt repräsentiert eine konkrete Entität, die in einem System vorhanden ist. Ein Knoten ist in diesem Kontext eine physische Einheit, auf der die Software ausgeführt wird, wie zum Beispiel ein Server oder ein PC.

Im Folgenden wird das Deployment-Diagramm der Android Studio MVVM App "LearnAhead" gezeigt. Es illustriert, wie die Komponenten der App auf einem mobilen Gerät verteilt sind und wie sie mit externen Diensten wie Firebase und dem Google Drive mit der installierbaren Android Package (APK)-Datei interagieren.



**Abb. 4.5.1:** Das Deployment Diagramm von LearnAhead

## 4.6 Klassen Diagramm

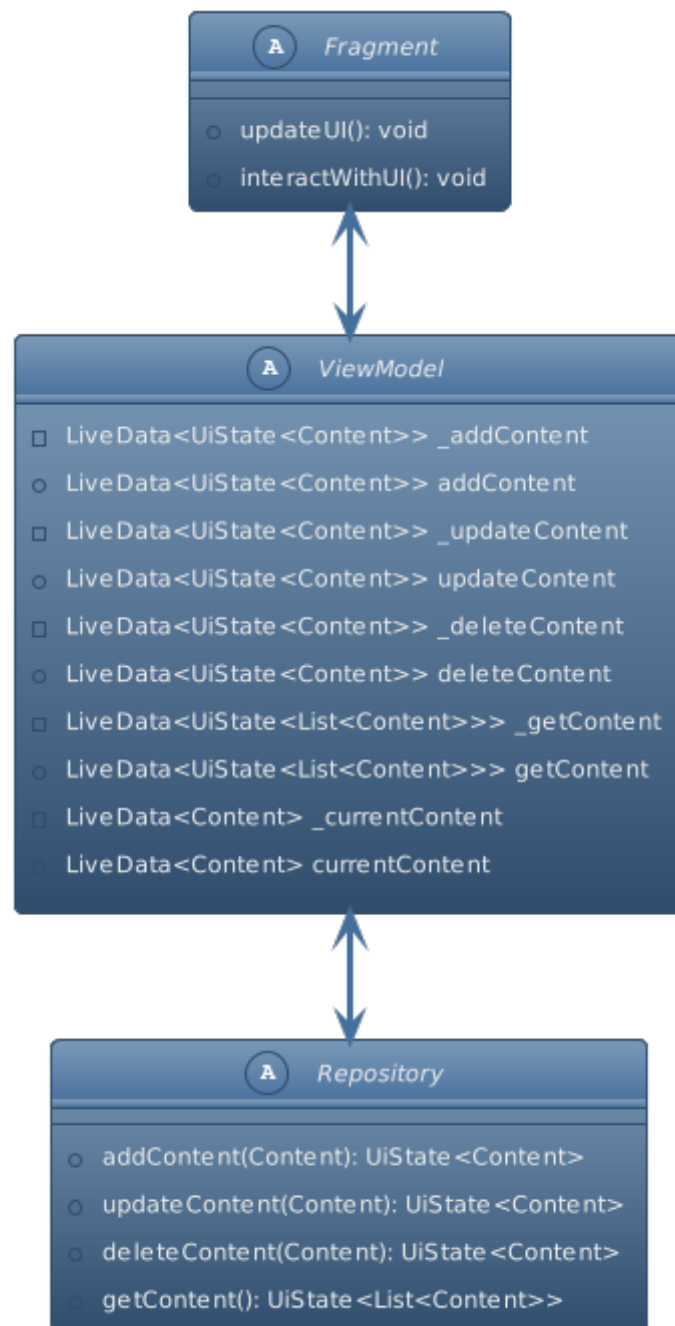
Das Klassendiagramm in Abbildung 4.6.1 zeigt die Struktur der LearnAhead App, die nach dem Model-View-ViewModel (MVVM) Muster entwickelt wurde. Der Übersicht zu gunsten ist hier in diesem Klassendiagramm nur eine Blaupause ausgestellt, da für jedes Feature ein neues ViewModel, ein neues Fragment und ein neues Model benötigt wurde.

Die Hauptklassen sind *ViewModel*, *Repository* und *Fragment*.

Die *ViewModel*-Klasse enthält LiveData-Objekte für die Methoden *addContent*, *updateContent*, *deleteContent* und *getContent*, sowie für das aktuelle Content-Objekt (*currentContent*). Die LiveData-Objekte sind öffentlich zugänglich, während die zugrunde liegenden Daten private sind.

Die *Repository*-Klasse stellt die Methoden *addContent*, *updateContent*, *deleteContent* und *getContent* zur Verfügung, die jeweils ein UiState-Objekt zurückgeben, das entweder ein Content-Objekt oder einen Fehler enthält.

Die *Fragment*-Klasse hat Methoden zum Aktualisieren der Benutzeroberfläche (*updateUI*) und zur Interaktion mit der Benutzeroberfläche (*interactWithUI*).

**Abb. 4.6.1:** Allgemeines Klassendiagramm der LearnAhead App

## 4.7 Sequenzdiagramme

Im nachfolgenden Kapitel werden einige Funktionen von LearnAhead mit Hilfe von Sequenzdiagrammen visualisiert um ihre direkte Art und Weise besser zu verstehen.

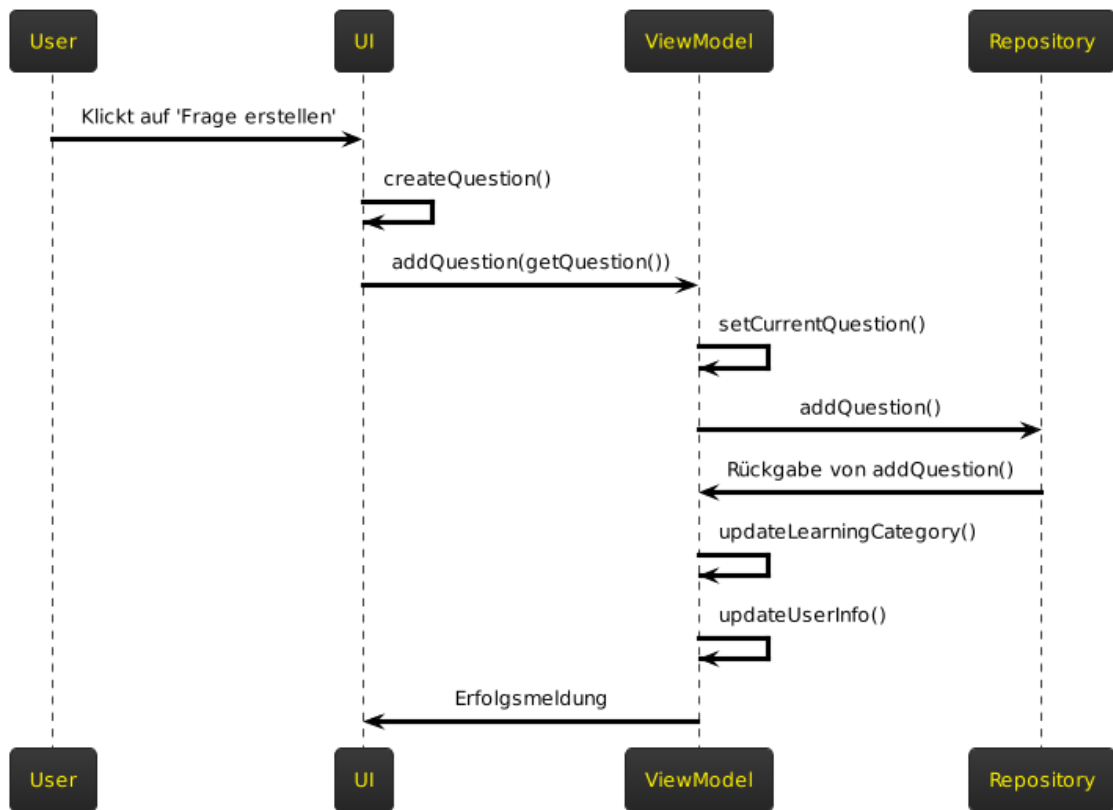


Abb. 4.7.1: Sequenzdiagramm - User erstellt eine Frage



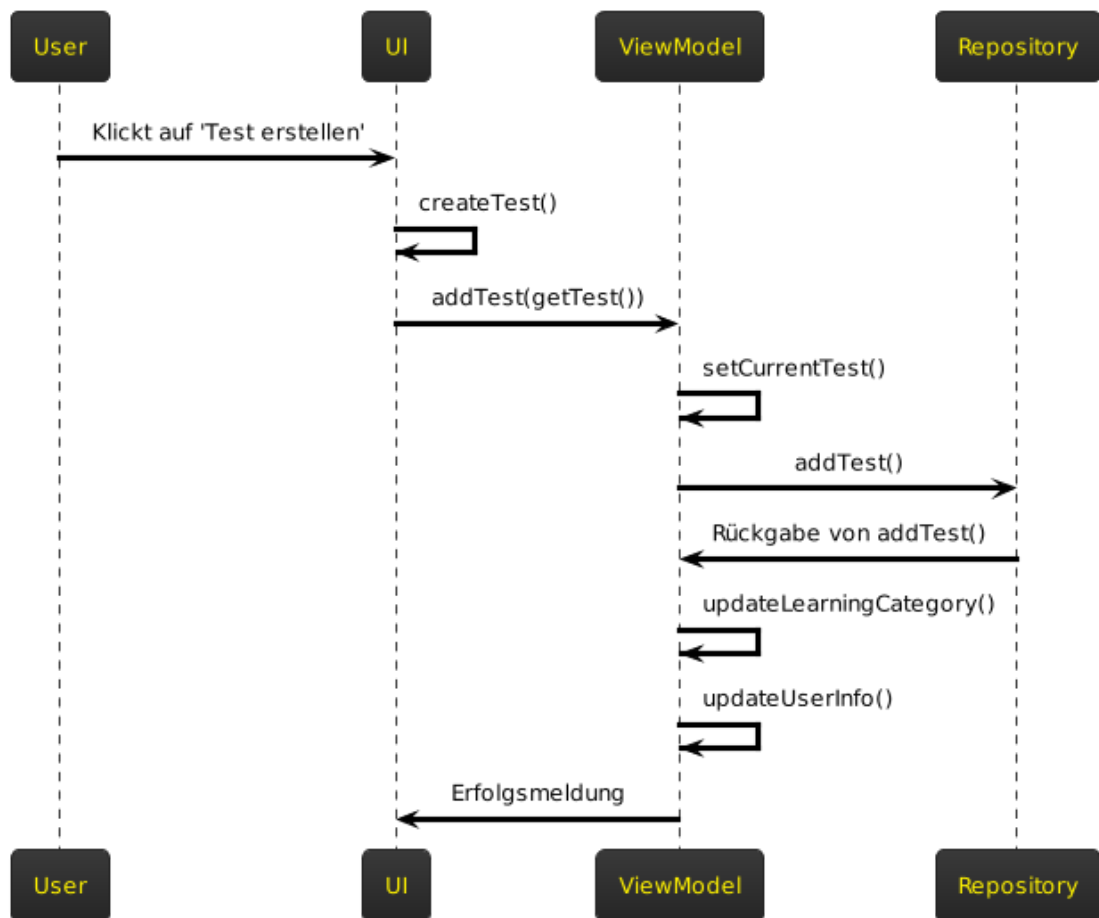


Abb. 4.7.2: Sequenzdiagramm - User erstellt einen Test

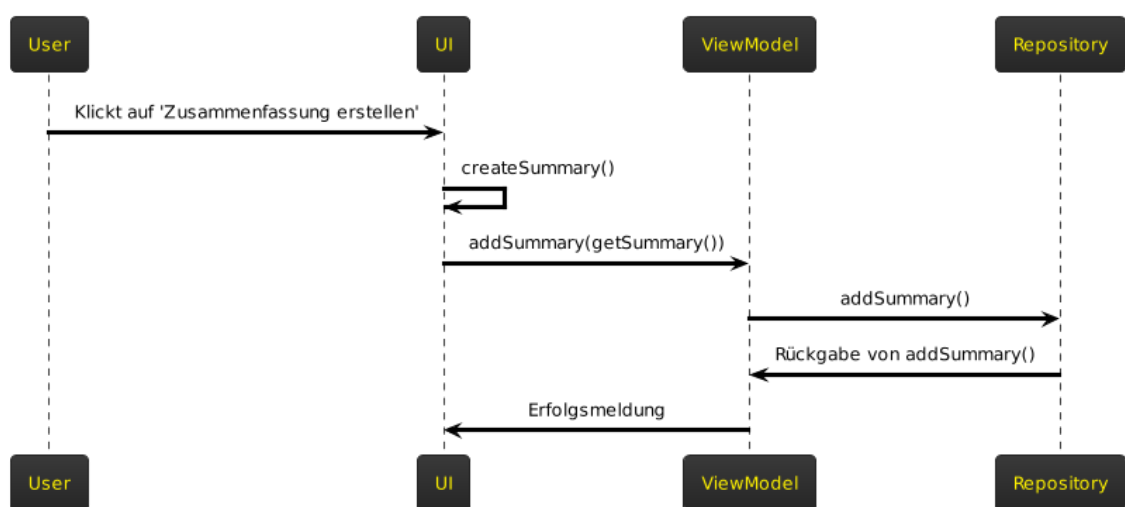


Abb. 4.7.3: Sequenzdiagramm - User erstellt eine Zusammenfassung

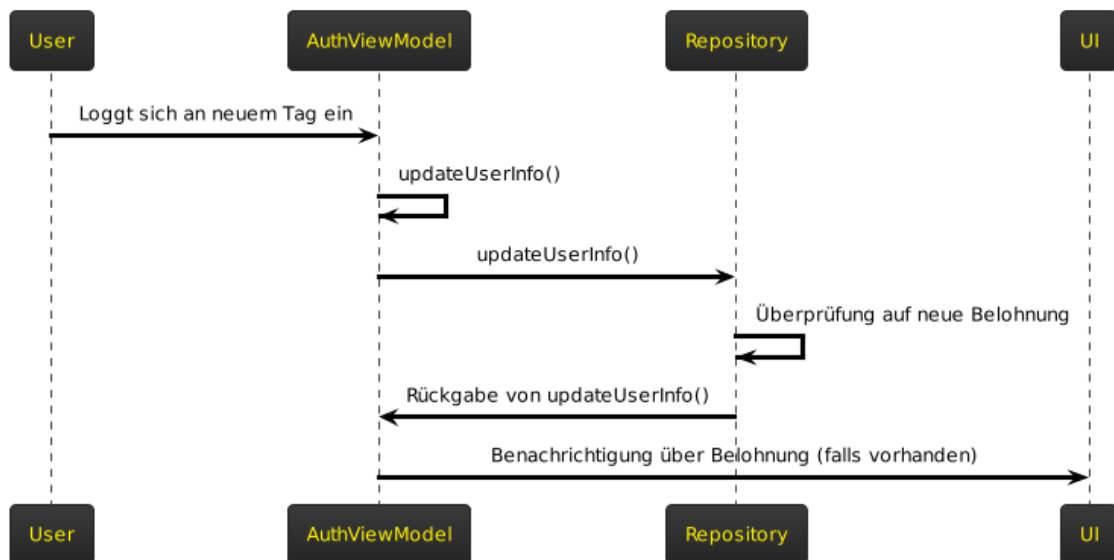


Abb. 4.7.4: Sequenzdiagramm - User erhält eine Belohnung für täglichen Login

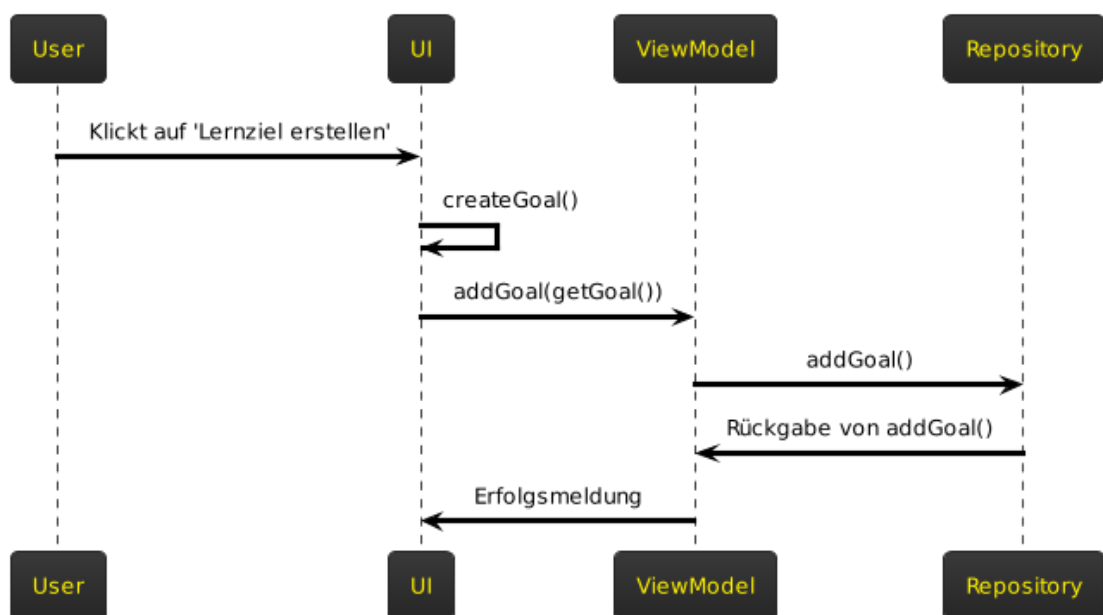
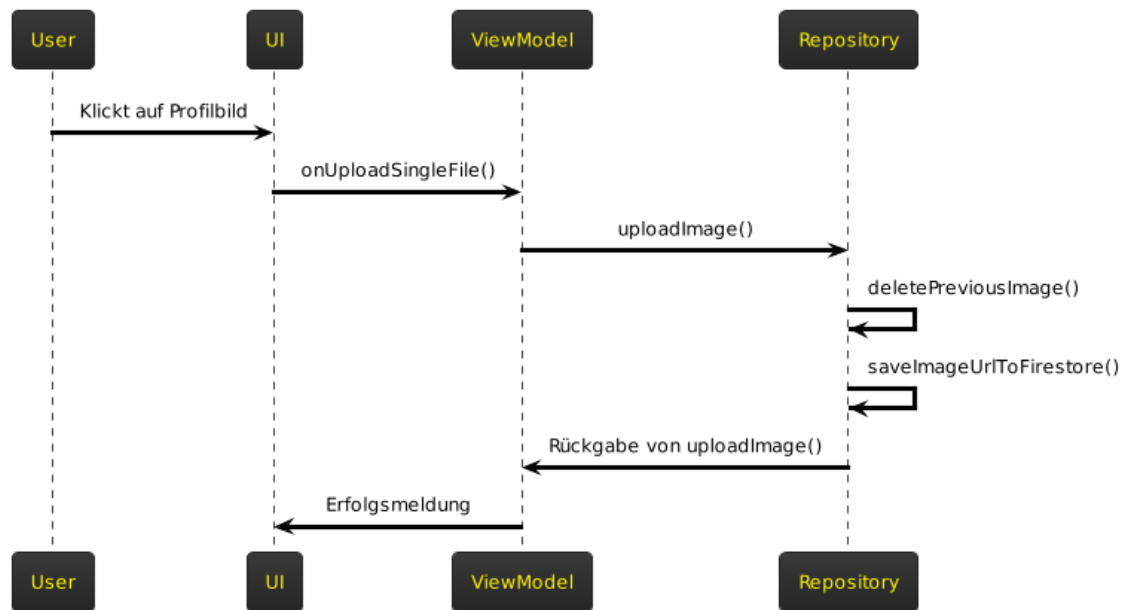


Abb. 4.7.5: Sequenzdiagramm - User erstellt Lernziel

**Abb. 4.7.6:** Sequenzdiagramm - User ändert sein Profilbild

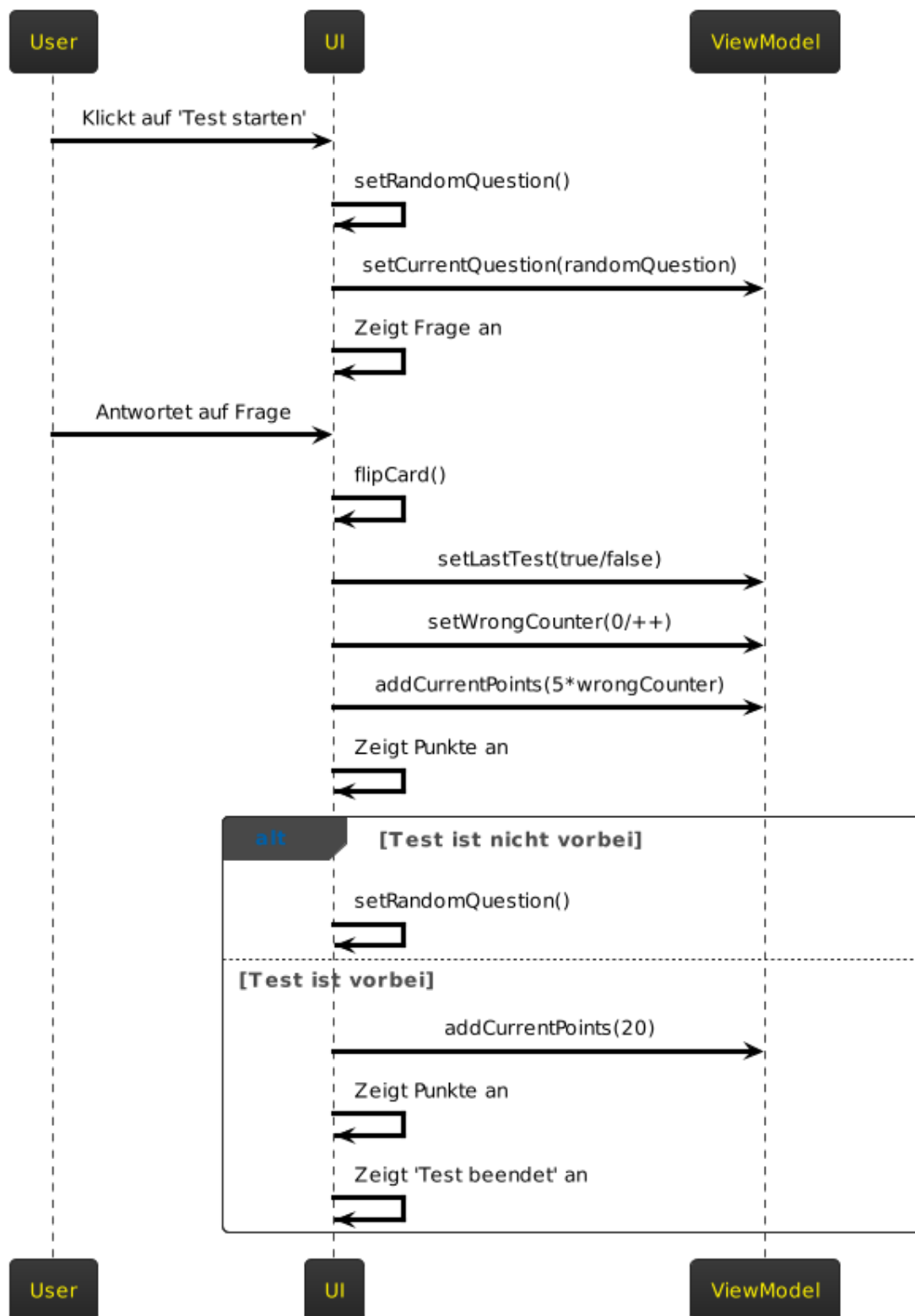


Abb. 4.7.7: Sequenzdiagramm - User führt Test aus

## 4.8 Prototyp von LearnAhead

Da LearnAhead einer strukturierte Planung unterlag, war auch ein Prototyp vonnöten, um zu überprüfen, ob das Grundkonzept mit MVVM funktioniert und sich unsere getroffenen Technologieentscheidungen bewährten.

In dem Prototyp wurde eine simple Applikation, welche Login, Register und einen Home Screen mit MVVM in Android Studio realisiert, umgesetzt.

## 5 Implementierung

In diesem Kapitel wird die komplette Implementierung von LearnAhead erläutert. Es wird explizit darauf eingegangen wie die App aus Sicht eines Users sowie eines Entwicklers zu installieren ist.

### 5.1 Installationsanleitung

In diesem Kapitel wird erläutert, wie die LearnAhead App sowohl von Endbenutzern als auch von Entwicklern installiert werden kann.

#### 5.1.1 Anleitung für User

Um die LearnAhead App auf einem Android-Gerät zu installieren, befolgen Sie bitte die folgenden Schritte:

1. Öffnen Sie den bereitgestellten Google Drive-Link in einem Webbrowser auf Ihrem Android-Gerät.
2. Sie sollten eine Vorschau der Datei „app-debug.apk“ sehen. Tippen Sie auf die Schaltfläche "Herunterladen" oder das Download-Symbol, um die Datei herunterzuladen.
3. Sie erhalten möglicherweise eine Warnung, dass diese Art von Datei Ihrem Gerät schaden kann. Da Sie die Quelle der Datei kennen, können Sie auf „OK“ oder „Herunterladen“ tippen, um fortzufahren.
4. Nachdem die Datei heruntergeladen wurde, öffnen Sie Ihre Benachrichtigungen und tippen Sie auf die heruntergeladene .apk-Datei.
5. Ihr Gerät fragt Sie möglicherweise, ob Sie die Installation von Apps aus unbekannten Quellen zulassen möchten. Sie müssen diese Option zulassen, um fortzufahren.
6. Sie sehen dann einen Bildschirm, der die Berechtigungen anzeigt, die die LearnAhead-App benötigt. Tippen Sie auf „Installieren“, um fortzufahren.
7. Nach Abschluss der Installation können Sie die App öffnen und verwenden, indem Sie auf die Schaltfläche „Öffnen“ tippen oder das App-Symbol auf Ihrem Gerät suchen und auswählen. Bitte beachte, dass bei der Verwendung von LearnAhead der Dark-Mode ausgeschaltet werden sollte, da dies sonst Einschränkungen in der UI zur Folge hat.

### 5.1.2 Anleitung für Developer

Um die LearnAhead App für Entwicklungs- und Testzwecke zu installieren, befolgen Sie bitte die folgenden Schritte:

1. Laden Sie und installieren Sie Android Studio auf Ihrem Computer, falls Sie dies noch nicht getan haben. Sie können Android Studio von der offiziellen Website herunterladen.
2. Beantragen Sie die Berechtigung, auf das Github-Repository der LearnAhead App zuzugreifen, indem Sie eine E-Mail an bayernico@web.de senden.
3. Nach Erhalt der Berechtigung, öffnen Sie Android Studio und wählen Sie „Get from Version Control“ aus dem „File“ Menü.
4. Geben Sie die URL des Github-Repositorys (<https://github.com/NicoB-Code/LearnAhead>) in das Feld „URL“ ein und klicken Sie auf „Clone“.
5. Nachdem das Repository erfolgreich geklont wurde, können Sie die App in Android Studio öffnen, bearbeiten und testen.

### 5.1.3 Verwendung des Android-Emulators

Der Android-Emulator ist ein Tool, das von Android Studio bereitgestellt wird und es Entwicklern ermöglicht, ihre Apps auf einem virtuellen Android-Gerät zu testen. Bitte beachten Sie, dass ihr virtuelles Gerät mindestens Nougat (Android 7) benötigt. Um den Android-Emulator zu verwenden, befolgen Sie bitte die folgenden Schritte:

1. Starten Sie Android Studio und öffnen Sie Ihr Projekt.
2. Wählen Sie „AVD Manager“ aus dem „Tools“ Menü.
3. Klicken Sie auf „Create Virtual Device“, um ein neues virtuelles Gerät zu erstellen.
4. Wählen Sie einen Gerätetyp aus der Liste aus und klicken Sie auf „Next“.
5. Wählen Sie ein System-Image aus, das Sie auf dem virtuellen Gerät installieren möchten, und klicken Sie auf „Next“. Wenn Sie noch kein System-Image heruntergeladen haben, können Sie dies tun, indem Sie auf „Download“ neben dem gewünschten Image klicken.
6. Überprüfen Sie die Konfiguration des virtuellen Geräts und klicken Sie auf „Finish“, um das virtuelle Gerät zu erstellen.
7. Um das virtuelle Gerät zu starten, klicken Sie auf das grüne Dreieckssymbol neben dem Gerät in der Liste der verfügbaren virtuellen Geräte im AVD Manager.
8. Sobald das virtuelle Gerät gestartet ist, können Sie Ihre App darauf ausführen, indem Sie „Run“ aus dem „Run“ Menü wählen und das laufende virtuelle Gerät aus der Liste der verfügbaren Geräte auswählen.

### 5.1.4 Übertragen der App zu Entwicklungszwecken ohne Deployment

Um die App auf einem tatsächlichen Android Gerät nutzen zu können, ohne sie jedes mal in den neu deployen zu müssen, können gewisse Entwicklertools genutzt werden, welche im nachfolgenden erklärt werden.

#### Übertragung per USB

1. Verbinden Sie Ihr Android-Gerät über ein USB-Kabel mit Ihrem Computer.
2. Aktivieren Sie auf Ihrem Android-Gerät die Option „USB-Debugging“ in den Entwicklungseinstellungen.
3. In Android Studio, wählen Sie „Run“ aus dem „Run“ Menü und wählen Sie Ihr angeschlossenes Gerät aus der Liste der verfügbaren Geräte.
4. Android Studio installiert die App auf Ihrem Gerät und startet sie.

#### Übertragung per Wi-Fi-Pairing

1. Stellen Sie sicher, dass Ihr Android-Gerät und Ihr Computer mit demselben Wi-Fi-Netzwerk verbunden sind.
2. Aktivieren Sie auf Ihrem Android-Gerät die Option „Wireless Debugging“ in den Entwicklungseinstellungen (hierfür müssen Sie den Entwicklermodus aktiviert haben - siehe <https://blog.deinhandy.de/android-entwickleroptionen-aktivieren-so-funktioniert>)
3. In Android Studio, wählen Sie „Pair Device with QR Code“ aus dem „Run“ Menü und scannen Sie den angezeigten QR-Code mit Ihrem Android-Gerät.
4. Nachdem das Gerät erfolgreich gepaart wurde, können Sie „Run“ aus dem „Run“ Menü wählen und Ihr gepaartes Gerät aus der Liste der verfügbaren Geräte auswählen.
5. Android Studio installiert die App auf Ihrem Gerät und startet sie.

## 5.2 Code Conventions

Die Lernapp „LearnAhead“ wurde in der Programmiersprache Kotlin entwickelt. Damit eine einheitliche Codebasis gewährleistet werden kann, wurden die offiziellen Code Conventions von Kotlin verwendet. Diese sind unter folgendem Link zu finden: Offizielle Kotlin Code Conventions.



## 5.3 IT-Sicherheit in LearnAhead

Ein wichtiger Aspekt bei der Entwicklung der LearnAhead App war die Gewährleistung der IT-Sicherheit. Dies wurde durch die Verwendung von Firebase für die Authentifizierung und Datenspeicherung erreicht [13].

### 5.3.1 Firebase Authentifizierung

Firebase Authentifizierung bietet eine sichere und zuverlässige Authentifizierungslösung. Es unterstützt eine Vielzahl von Authentifizierungsmethoden, einschließlich E-Mail und Passwort, und es integriert sich nahtlos mit anderen Firebase-Diensten [10].

Firebase Authentifizierung bietet auch eine Reihe von Sicherheitsfunktionen, um die Konten der Benutzer zu schützen. Dazu gehören unter anderem:

- **Passwortschutz:** Firebase speichert Passwörter sicher und ermöglicht es den Benutzern, ihre Passwörter zurückzusetzen, wenn sie diese vergessen haben [10].
- **Kontosicherheit:** Firebase bietet Schutzmaßnahmen gegen Brute-Force- und DDoS-Angriffe [16].
- **Datenschutz:** Firebase hält sich an die Datenschutzbestimmungen und -standards, einschließlich der DSGVO [15].

### 5.3.2 Firebase Datenspeicherung

Die Benutzerdaten der LearnAhead App werden sicher in Firebase gespeichert. Firebase bietet eine Reihe von Funktionen, die die Sicherheit der gespeicherten Daten gewährleisten: [17]

- **Datenverschlüsselung:** Firebase speichert Daten in einer verschlüsselten Form, sowohl während der Übertragung als auch im Ruhezustand [12].
- **Zugriffskontrolle:** Firebase ermöglicht es, detaillierte Sicherheitsregeln festzulegen, um den Zugriff auf die gespeicherten Daten zu kontrollieren [9].
- **Compliance:** Firebase hält sich an wichtige Compliance-Standards und -Vorschriften, um die Sicherheit und den Schutz der gespeicherten Daten zu gewährleisten [11].

Durch die Verwendung von Firebase konnte die LearnAhead App eine robuste und sichere Lösung für die Authentifizierung und Datenspeicherung implementieren [13].

## **6 Ergebnisse**

Im folgenden Kapitel werden die Ergebnisse unserer Lern-App präsentiert und aufgelistet.

### **6.1 Nutzen der Lern-App**

Der Nutzen unserer geplanten und dahingehend entwickelten Lern-App LearnAhead liegt darin die Studierenden der DHBW bei ihrem Lernprozess zu unterstützen.

Es können nicht nur Lerninhalte in eine Lernkategorie kategorisiert, sondern auch, mit Hilfe der Lernziele, ihr Zeitlicher Rahmen erfasst werden.

Die Lerninhalte können mit den Zusammenfassungen einer Lernkategorie mit dem Markdown-Editor erfolgreich erfasst und kondensiert werden.

Zu diesen Lerninhalten kann man dann Tests und Fragen erstellen, welche sich auf diese beziehen.

### **6.2 Feedback von Präsentation der Studienarbeit**

Im Rahmen der Studienarbeiten wurde uns am 25.05.2023 die Möglichkeit geboten unsere Studienarbeiten in den Räumen N001 bis N004 im Rahmen eines Plakat Forums vorzustellen.

Hierbei wurde unter anderem die Idee für das weiterreichen der Lerninhalte gegeben, da diese momentan noch von Hand eingegeben werden müssen.

LearnAhead stieß auf großes Interesse und es wurde vieles hinterfragt, was uns noch umso mehr Ideen einbrachte, von denen einige in den Ausblick wanderten, da die Möglichkeiten bei der Entwicklung einer Lern-App praktisch unendlich sind. Mehr dazu siehe Kapitel 8.2.

## 6.3 Der finale Zeitplan

Im Folgenden findet sich der finale Zeitplan, so wie er zum Schluss umgesetzt wurde.

Meilenstein	Zeitplan	Beschreibung
Literaturrecherche	14.10.2022 - 31.01.2023	Das Durchführen einer umfangreichen Literaturrecherche auf Basis von wissenschaftlichen Dokumenten.
Use-Case-Erstellung	14.10.2022 - 11.11.2022	Identifizierung und Dokumentation der Hauptfunktionalitäten und Anwendungsfälle der Lern-App.
UI-Konzept	11.11.2022 - 02.02.2023	Entwicklung eines visuellen Konzepts für die Benutzeroberfläche (UI) der Lern-App.
Datenbank-Konzept	20.01.2023 - 16.02.2023	Design und Auswahl des Datenbanksystems, die für die App benötigt wird.
Architektur-Konzept	03.02.2023 - 16.02.2023	Realisierung einer Code-Architektur und Auswahl verschiedener Komponenten sowie Frameworks, die in der App verwendet werden.
Architektur-Prototyp	10.02.2023 - 16.02.2023	Erstellung eines ersten Prototypen der die vorgeschlagene Architektur implementiert.
Login / Registrierung	17.02.2023 - 30.03.2023	Implementierung der Funktionen für Anmeldung, Registrierung und Passwortwiederherstellung.
Lernkategorien & Lernziele	31.03.2023 - 11.05.2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Lernkategorien und -zielen.
Erstellung von Fragen und Tests	12.05.2023 - 11.07.2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Fragen und Tests.
Erstellung von Zusammenfassungen	12.05.2023 - 11.07.2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Zusammenfassungen von Lernkategorien.
Optimale Pausenberechnung realisieren	12.07.2023 - 14.07.2023	Erstellung eines Algorithmus, welcher den Nutzer die optimale Pause vorschlägt sowie erinnert.
Optimale Lernplan generieren	12.07.2023 - 14.07.2023	Erstellung eines optimalen Lernplans auf Basis der Lernziele.
Durchführen von User-Tests	09.07.2023 - 16.07.2023	Durchführung von umfassenden Tests, um die Qualität, Funktionalität und Stabilität der App sicherzustellen.
Bugs beheben	14.07.2023 - 16.07.2023	Behebung von Fehlern und Problemen in der App.
Dokumentation	14.10.2022 - 16.07.2023	Erstellung einer wissenschaftlichen Arbeit, die das Vorgehen, Funktionen, die Implementierung sowie die Verwendung der App begründet.

Anhand des finalen Zeitplans lässt sich erkennen, dass die Implementierung von den Zusammenfassungen, Fragen und Tests mehr Zeit in Anspruch genommen hat, als ursprünglich geplant. Daraus folgt, dass das „Springen“ mithilfe von Tags nicht mehr implementiert werden konnte.

## 7 User-Tests

Um Fehlverhalten und mögliche Verständnisprobleme beim Benutzen der App aufzudecken, wurden User Tests durchgeführt. Hierbei wird explizit darauf geachtet, dass nicht nur frei getestet wird, sondern die Tester anhand definierter Aufgaben und Ziele alle Funktionalitäten durchlaufen.

Die User Tests werden deshalb in:

1. Offene Tests, allgemeine Ziele
2. Freie Tests

unterteilt.

Im Folgenden findet sich der ausführliche Testbogen, welcher an die Test-User verschickt wurde.

*Danke, dass du dich entschieden hast uns zu helfen und mit uns unsere App zu testen. Höchstwahrscheinlich wurdest du von einem von uns beiden angesprochen und gebeten zu helfen. Deswegen kann es sein, dass du nicht beide von uns kennst, wir sind: Phillipp Patzelt und Nico Bayer. Alle samt Studenten im 6. Semester an der DHBW Ravensburg.*

*Zusammen haben wir im letzten dreiviertel Jahr eine Lern-App erdacht, geplant und erstellt im Rahmen wissenschaftlichen Arbeit. Nun sind wir im Endspurt und stehen kurz vor dem offiziellen Release von LearnAhead, einer mobilen Lern-Applikation.*

*Doch nun kurz vor der Veröffentlichung unserer App müssen wir ein letztes mal testen ob auch alles klappt. Hier kommst du ins spiel. Bei diesem letzten Test benötigen wir Hilfe. Es kann sein, dass uns durch das montaelange Arbeiten an unserer App, grobe Fehler gar nicht mehr auffallen oder Teile unserer App unpraktisch oder nicht intuitiv zu bedienen sind.*

*Deswegen wurdest du ausgewählt vor dem regulären Betrieb unsere App zu testen und währenddessen diesen Bericht auszufüllen. Bitte gib uns diesen Bogen ausgefüllt bis zum 30.06.2023 zurück, damit wir pünktlich vor der Abgabe und dem Release von LearnAhead am 17.07.2023 die letzten Fehler korrigieren können.*

*Bitte scheue dich nicht uns ein ehrliches Feedback zu geben und wir wünschen im Namen von uns beiden viel Spaß und Erfolg bei diesen Tests!*

## 7.1 Offene Tests, allgemeine Ziele

Das Ziel der offenen Tests ist es, durch eine bewusst grob gewählte Aufgabenstellung, die App auf Praktikabilität zu testen, zu sehen ob der Nutzer die App etwas stören könnte und wie intuitiv er sich in der App bewegt.

### Aufgabe:

Bitte rufe unsere App auf deinem Mobilgerät auf, falls du sie noch nicht installiert hast, tue dies, indem du die „app-debug.apk“ vom bereitgestellten Google-Drive Link herunterziehst und installierst.

Du wirst nun auf der Login-Seite landen. Deine erste Aufgabe ist nun auf eigene Faust Folgendes zu tun: **Eine Zusammenfassung zu erstellen.**

Versuche von selbst die dafür notwendigen Schritte zu erschließen und auszuführen.

Bitte notiere dafür im Folgenden dein Mobilgerät, sowie alles was dir auffällt. Konntest du die Aufgabe leicht ausführen? Gabs es irgendwelche Fehler oder kamst du an irgendeinem Punkt an, an dem du nicht mehr weiterwusstest?

Notiere dazu bitte nur konkret die Antwort auf diese Fragen. Gib bitte erst im nächsten Abschnitt dann dein Feedback oder die Dinge, welche dir in deinen Augen als unpraktisch auffallen.

Wenn du deine Aufgabe geschafft hast, verbleibe bitte im Edit-Fenster der Zusammenfassung.

Genutztes Mobilgerät: \_\_\_\_\_

### Antwort auf die obige Frage:

---

---

---

---

---

---

---

---

---

---

### Feedback zum ersten Eindruck, Wünsche und Dinge die dir unpraktisch vorkommen:

---

---

---

---

---

---

---

---

Sehr schön, nun befindest du dich in einem Kernteil unserer App, den Zusammenfassungen. Deine nächste Aufgabe ist es: **Einen kleinen Markdown Text verfassen und ihn dir anzeigen lassen.**

Falls du mit Markdown noch nicht vertraut bist, kein Problem! Im Internet findest du viele Anleitungen wie Markdown funktioniert, hier ein Beispiel: <https://www.markdownguide.org/> Folge hierbei der gleichen Prozedur wie vorhin, suche dir selbst einen Weg durch die App. Fallen dir Fehler auf? Wusstest du wo du suchen sollst um die Aufgabe zu erledigen? Im Folgenden darfst du alles was dir auffällt UND dein Feedback und Kritik auf einmal aufschreiben.

---

---

---

---

---

---

---

---

Gut! Nun widmest du dich einem neuen Teil der App und somit auch einer neuen Aufgabe: **Ändere dein Profilbild.**

Wie zuvor, bitte notiere Fehler und Dinge die dir als unpraktisch auffallen.

---

---

---

---

---

---

---

---

Wir sind fast beim letzten Teil der offenen Tests angelangt, nun wird deine Aufgabe sein: **Ein Lernziel zu erstellen und zu editieren.**

Notiere auch hier bitte alle Fehler und alles unpraktische was dir auffällt.

---

---

---

---

---

---

---

---

Beim letzten Teil der offenen Tests geht es darum, **deine erstellten Elemente zu löschen und dich auszuloggen**.

Wie bereits im vorherigen Ablauf notiere alle Fehler und unpraktischen Elemente.

---

---

---

---

---

---

## 7.2 Freie Tests

In den freien Tests geht es darum, dass du selbst gezielt versuchst etwas kaputt zu machen oder Fehler zu produzieren. Teste bitte auf vielen unterschiedlichen Geräten, ob alles funktioniert und du noch irgendetwas entdeckst, das du bisher noch nicht erwähnt hast.

Falls du tatsächlich noch etwas findest, beschreibe uns bitte genau, wie es dazu kam, auf welcher Seite du warst, welches Gerät du genutzt hast und was du geklickt hast.

Des Weiteren erwähne hier auch alle noch nicht genannten Verbesserungsvorschläge an unsere App und wir werden sehen, was wir tun können, um diese so schnell wie möglich zu implementieren. Danke und viel Erfolg!

---

---

---

---

## 8 Fazit

LernAhead ist ein Projekt, welches von uns erfolgreich abgeschlossen wurde, jedoch noch einen Weg vor sich hat, der sich erst herauskristallisieren muss.

Im folgenden Abschnitt wird der Ablauf des Projekts kritisch betrachtet sowie mögliche zukünftige Features erwägt und daraus ein Resümee gezogen.

### 8.1 Ablauf des Projekts

Im Rahmen der Studienarbeit sollen die Studierenden sich „in ein komplexes, aber eng umgrenztes Gebiet vertiefend einarbeiten und den allgemeinen Stand des Wissens erwerben.“[34]

Am Beginn von LearnAhead wurden uns von der zuständigen Betreuerin Frau Claudia Zinser zwei wissenschaftliche Bücher ausgehändigt, aus welchen wir Informationen über neuronalen Lernens und andere dem Lernen zugeschriebene Gebiete erfassten.

Diese Masse an Wissen musste erst einmal neben dem eigentlichen Hochschulbetrieb verarbeitet werden. Dies zog sich bis in die Mitte der 5. Praxisphase, da nicht nur die beiden ausgehändigten Bücher gelesen wurden, sondern auch andere Dokumente Teil unserer Literaturrecherche waren.

Bei der kompletten Durchführung der Studienarbeit war es so, dass diese immer einen kleinen Zeitslot benötigte, egal wann, da es sich hierbei ja um ein kontinuierliches Projekt handelte.

Es war interessant und äußerst lehrreich ein Projekt zu erleben, welches von Grundauf, von Anfang bis Ende geplant und implementiert wurde.

Wir sind mit dem Ablauf der Studienarbeit sehr zufrieden und haben es eingesehen, dass wir an manchen Stellen bei der Planung überambitioniert waren, jedoch freuen wir uns, das Projekt an kommende Generationen von Studenten weiterzugeben und das Projekt wachsen zu sehen.

LearnAhead war kein Projekt, was mit bereits vorhandenem Wissen aus dem Boden gestampft wurde, es wurden uns an vielen Stellen über den Prozess des Lernens, nach nun mehr als fast 15 Jahren Lernen in allen möglichen Schulen, die Augen geöffnet.

Abseits der Hochschule haben wir nun ein umfassendes Interesse an dem neurologischen Prozess des Lernens, da sich hierfür auch sehr viele Erkenntnisse über maschinelles Lernen ziehen lassen.



## 8.2 Ausblick und die Zukunft

Mit der Veröffentlichung von LearnAhead geht ein kleines Kapitel in unserer Geschichte zu Ende, aber ein anderes Tor wird noch viel weiter aufgestoßen, welches in die Zukunft von LearnAhead führt.

Viele Features könnten noch in LearnAhead implementiert werden, im nachfolgenden Abschnitt werden einige davon aufgelistet.

1. Lerninhalte, sowie dessen Tests und Fragen müssen momentan noch manuell eingegeben werden. Dies hat zwar den Vorteil, dass hierdurch die Verbindungen im Gehirn des Lernenden gefestigt werden.

Für die Zukunft gäbe es noch die Möglichkeit die App in so einer Form zu erweitern, dass Lerninhalte und Tests oder ganze Lernkategorien importiert, beziehungsweise exportiert werden können, so dass ein neuer Jahrgang den vergangenen Jahrgang nach der Lernkategorie für die anstehende Klausur fragen könnte.

2. Ein anderes Team aus Studierenden haben im Rahmen ihrer Studienarbeit eine Lern-App entwickelt, welche ihren Fokus auf kollaboratives Lernen legt. In Absprache mit der zuständigen Betreuerin Claudia Zinser könnten hierbei beide Lern-Apps über eine API oder Ähnliches miteinander verbunden werden.
3. Bei den Zusammenfassungen ist es aktuell noch mühsam die Inhalte selber von Grund auf einzugeben. Um dem entgegenzuwirken könnte ein Template-System aufgesetzt werden, welches es den Nutzern ermöglicht Vorlagen für Zusammenfassungen zu erstellen.
4. Bisher wird dem User das Lernen nur durch einige zufällig ausgewählte Lerntipps beigebracht. Unsere Vision am Anfang von LearnAhead war ein Bereich der App, in welchem man das Lernen lernen kann und allgemeine Falschannahmen zum lernen aufgeklärt werden.
5. Der User wird dafür belohnt, wenn er eine Frage nach mehrmaligem falsch beantworten richtig beantwortet. Es könnte zur Unterstützung des Lernprozesses nun ein Algorithmus implementiert werden, welcher den User nach mehrmaligem falsch beantworten einer Frage auf die zugehörige Zusammenfassung hinleitet.

Hierfür müssten Zusammenfassungsinhalte mit Fragen verknüpft werden. Dies könnte beispielsweise über die Tag-Funktion geschehen, indem man einen Tag die Zugehörigkeit zu einer Zusammenfassung zuspricht.

6. Der Markdown-Editor unterstützt bisher nur Basic-Markdown Syntax. Markwon stellt die

Möglichkeit zur Verfügung Plugins einzubauen, welche diese Syntax erweitern. Es könnte eine Funktion zum Einfügen von mathematischen Operationen implementiert werden, wie es MathJax (<https://www.mathjax.org/>) in LaTeX implementiert.

7. Momentan wird der Lernfortschritt einer Lernkategorie dem Benutzer nicht vollständig ersichtlich. In der weiteren Entwicklung des Projektes könnte deswegen ein Lernfortschritts-Balken implementiert werden, welcher dem User seinen momentan Lernfortschritt einer Lernkategorie anzeigt.
8. Tags sind momentan statisch anlegbar. Dies bedeutet, dass wenn ein Tag erstellt wird und dabei ein Fehler unterläuft, muss dieser gelöscht und korrekt neu erstellt werden. Um dem entgegenzuwirken könnte man ein Tag-Bearbeitungssystem entwickeln.
9. LearnAhead ist momentan durch seine Plattform-Abhängigkeit nur auf Android Mobilgeräte beschränkt. Um dies zu verbessern könnte LearnAhead zu einer Progressive Web App (PWA) umfunktioniert werden, um Plattformunabhängigkeit zu ermöglichen.
10. Derzeit muss der User sich die Daten für seine Lernziele quantitativ merken und selbst verwalten. Dieses Problem könnte mit einem Kalender-System gelöst werden, welches dem User eine bessere zeitliche Übersicht über seine Lernziele gibt.
11. Wir haben uns entschieden, unsere App nicht im Google Play Store zu veröffentlichen, da dies im Rahmen unseres Studiums nicht erforderlich ist. Eine der Herausforderungen bei der Veröffentlichung einer App im Play Store ist die Notwendigkeit einer umfassenden Datenschutzerklärung, sowie das dreimalige nicht akzeptieren von gültigen amtlichen Ausweisdokumenten seitens Google. Diese zu erstellen und sicherzustellen, dass unsere App vollständig konform ist, würde zusätzliche Zeit und Ressourcen erfordern. Da unser Hauptziel darin besteht, die technischen sowie wissenschaftlichen Fähigkeiten und Kenntnisse zu erwerben, die für die Entwicklung einer funktionsfähigen App erforderlich sind, haben wir uns darauf konzentriert, eine App zu erstellen, die unseren Anforderungen entspricht, anstatt sie für eine breitere Öffentlichkeit zugänglich zu machen. Um die App für die breite Öffentlichkeit zugänglich zu machen, könnte man diesen Schritt noch in der Zukunft von LearnAhead gehen.

## Referenzen

- [1] appsoluts. *Der Unterschied zwischen iOS Entwicklung & Android Entwicklung !* de-DE. Nov. 2016. URL: <https://appsoluts.de/android-entwicklung-vs-ios-entwicklung/> (besucht am 10.06.2023) (siehe S. 6).
- [2] M.F. Bear, B.W. Connors und M.A. Paradiso. *Neuroscience: Exploring the Brain*. Wolters Kluwer Health, 2015 (siehe S. 7–9).
- [3] Henning Beck. *Das neue Lernen heißt Verstehen*. ger. 3. Auflage. Berlin: Ullstein, 2021. ISBN: 978-3-550-20049-6 (siehe S. 7, 10–12).
- [4] Bumptech. *Glide: An image loading and caching library for Android focused on smooth scrolling*. 2022. URL: <https://bumptech.github.io/glide/> (besucht am 12.07.2023) (siehe S. 39).
- [5] Bumptech. *Glide Repository on GitHub*. 2022. URL: <https://github.com/bumptech/glide> (besucht am 12.07.2023) (siehe S. 39).
- [6] Patrick Böllhoff. *Kotlin vs Java: strengths, weaknesses and when to use which*. en-US. Section: Mobile app development. Jan. 2022. URL: <https://kruschecompany.com/kotlin-vs-java/> (besucht am 10.06.2023) (siehe S. 6).
- [7] S.F. Cooke und T.V.P. Bliss. „Synaptic plasticity and learning“. In: *Current opinion in neurobiology* 16.2 (2006), S. 146–152 (siehe S. 8, 9).
- [8] S. Deterding u. a. „From game design elements to gamefulness: Defining gamification“. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. ACM. 2011, S. 9–15 (siehe S. 13, 14).
- [9] *Firebase Access Control*. 2023. URL: <https://firebase.google.com/docs/storage/security/user-security> (besucht am 15.07.2023) (siehe S. 57).
- [10] *Firebase Authentication*. 2023. URL: <https://firebase.google.com/products/auth> (besucht am 15.07.2023) (siehe S. 57).
- [11] *Firebase Compliance*. 2023. URL: <https://firebase.google.com/support/privacy#firebase-is-certified-under-major-privacy-and-security-standards> (besucht am 15.07.2023) (siehe S. 57).
- [12] *Firebase Encryption*. 2023. URL: <https://firebase.google.com/docs/storage/security/start> (besucht am 15.07.2023) (siehe S. 57).
- [13] *Firebase Overview*. 2023. URL: <https://firebase.google.com/docs> (besucht am 15.07.2023) (siehe S. 57).

- 
- [14] *Firebase Pricing*. en. URL: <https://firebase.google.com/pricing> (besucht am 07.07.2023) (siehe S. 37).
  - [15] *Firebase Privacy*. 2023. URL: <https://firebase.google.com/support/privacy> (besucht am 15.07.2023) (siehe S. 57).
  - [16] *Firebase Security*. 2023. URL: <https://firebase.google.com/docs/auth/admin/security> (besucht am 15.07.2023) (siehe S. 57).
  - [17] *Firebase Storage*. 2023. URL: <https://firebase.google.com/products/storage> (besucht am 15.07.2023) (siehe S. 57).
  - [18] Google. *Android Jetpack*. 2022. URL: <https://developer.android.com/jetpack> (besucht am 12.07.2023) (siehe S. 39).
  - [19] Google. *Dagger Hilt*. 2021. URL: <https://dagger.dev/hilt/> (besucht am 12.07.2023) (siehe S. 38).
  - [20] Google. *Dependency injection with Hilt*. 2021. URL: <https://developer.android.com/training/dependency-injection/hilt-android> (besucht am 12.07.2023) (siehe S. 38).
  - [21] Google. *Firebase Android SDK on GitHub*. 2022. URL: <https://github.com/firebase/firebase-android-sdk> (besucht am 12.07.2023) (siehe S. 40).
  - [22] Google. *Firebase: Mobile App Development Platform*. 2022. URL: <https://firebase.google.com/> (besucht am 12.07.2023) (siehe S. 40).
  - [23] Google. *Guide to app architecture*. 2023. URL: <https://developer.android.com/topic/architecture> (besucht am 12.07.2023) (siehe S. 40, 41).
  - [24] Google. *Navigation Component | Android Developers*. 2022. URL: <https://developer.android.com/guide/navigation> (besucht am 12.07.2023) (siehe S. 39, 40).
  - [25] John Gruber. *Markdown*. 2004. URL: <https://daringfireball.net/projects/markdown/> (besucht am 12.07.2023) (siehe S. 38).
  - [26] J. Hamari, J. Koivisto und H. Sarsa. „Does gamification work? A literature review of empirical studies on gamification“. In: *Proceedings of the 47th Hawaii International Conference on System Sciences, Hawaii, USA* (2014) (siehe S. 13).
  - [27] M.D. Hanus und J. Fox. „Learning in color: How color and affect influence learning outcomes“. In: *IEEE Transactions on Affective Computing* 7.3 (2015), S. 243–252 (siehe S. 13).
  - [28] D.O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, 1949 (siehe S. 8, 9).

- [29] E.R. Kandel u. a. *Principles of Neural Science*. McGraw-Hill Medical, 2012 (siehe S. 7–9).
- [30] K.M. Kapp. *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. Pfeiffer, 2012 (siehe S. 13).
- [31] R.N. Landers. „Developing a Theory of Gamified Learning: Linking Serious Games and Gamification of Learning“. In: *Simulation & Gaming* 45.6 (2014), S. 752–768 (siehe S. 13).
- [32] Prof Dr Günter W. Maier. *Definition: Lernen*. de. Text. URL: <https://wirtschaftslexikon.gabler.de/definition/lernen-41169> (besucht am 03.07.2023) (siehe S. 7).
- [33] Microsoft. *The MVVM Pattern*. 2009. URL: <https://docs.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern> (besucht am 12.07.2023) (siehe S. 40, 41).
- [34] *Modulhandbuch Informatik DHBW Ravensburg*. 2023. URL: <https://www.ravensburg.dhbw.de/fileadmin/user/public/SP/FN/Informatik/Informatik.pdf> (besucht am 15.07.2023) (siehe S. 64).
- [35] S. Nicholson. *A RECIPE for Meaningful Gamification*. Springer, 2015, S. 1–20 (siehe S. 14).
- [36] Noties. *Markwon: a Markdown library for Android*. 2022. URL: <https://noties.io/Markwon/> (besucht am 12.07.2023) (siehe S. 38, 39).
- [37] *Registrierung - Support - Apple Developer*. URL: <https://developer.apple.com/de/support/enrollment/> (besucht am 10.06.2023) (siehe S. 6).
- [38] M. Spitzer. *Lernen: Gehirnforschung und die Schule des Lebens*. Spektrum Akademischer Verlag, 2002 (siehe S. 9).
- [39] Farnam Street. *The Spacing Effect: How to Improve Learning and Maximize Retention*. en-US. Dez. 2018. URL: <https://fs.blog/spacing-effect/> (besucht am 05.07.2023) (siehe S. 10, 11).
- [40] *Wählen Sie eine Datenbank aus: Cloud Firestore oder Echtzeitdatenbank | Firebase Realtime Database*. de-x-mtfrom-en. URL: <https://firebase.google.com/docs/database/rtdb-vs-firestore?hl=de> (besucht am 07.07.2023) (siehe S. 37).