

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.

Ich versichere hiermit, dass ich meine Bachelorarbeit (bzw. Projektarbeit oder Studienarbeit bzw. Hausarbeit) mit dem Thema:

Entwicklung einer mobilen Lern-Applikation

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Abstract

Zusammenfassung

Schlüsselwörter

Abstract

Keywords

Inhaltsverzeichnis

Abkürzungsverzeichnis	VI
Abbildungsverzeichnis	VII
Tabellenverzeichnis	VII
1 Einleitung	1
2 Planungsphase	2
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
3 Definitionsphase	7
3.1 Theoretische Grundlagen	7
3.1.1 Lernen ist nicht gleich Verstehen	7
3.1.2 Synapsen	7
3.1.3 Neuronen	7
3.1.4 Neuronales Netz	7
3.1.5 Optimaler Lernplan anhand des Spacing Effect	8
3.1.6 Tagging innerhalb der Lern-App	10
3.2 Pflichtenheft	11
3.3 Use Case Diagramm	11
3.4 Use Case Beschreibung	12
3.5 Datenbankstruktur	17
3.5.1 Datenbankmodell	17
3.5.2 Data Dictionary	18
3.6 HMI	22

3.6.1	Seitenhierarchie	22
3.6.2	UI-Mockups	23
3.7	Flow-Chart-Diagramm	32
4	Entwurf	33
4.1	Geschäftsfälle anhand des BPMN Workflows	33
4.2	Auswahl und Begründung des Datenbankkonzepts	33
4.3	Auswahl der Klassenbibliotheken/Frameworks	34
4.4	Design Patterns für relevante Problemstellungen	34
4.5	Software-Komponenten	34
4.5.1	Gesamtkomposition	34
4.5.2	Frontend	35
4.5.3	Backend	36
4.6	Deployment Diagramm	36
4.7	Klassen Diagramm	36
4.8	Aktivitätsdiagramm	36
4.9	Sequenzdiagramme	36
4.10	Prototyp (optional)	36
5	Ergebnisse	37
5.1	Der finale Zeitplan	37
	Referenzen	38
A	Anhang	A

Abkürzungsverzeichnis

DHBW Duale Hochschule Baden-Württemberg	I
--	----------

Abbildungsverzeichnis

3.1.1 Ebbinghaus Vergessenskurve	8
3.1.2 Ebbinghaus verzögerte Vergessenskurve durch Wiederholungen	9
3.3.1 Use Case Diagramm	11
3.5.1 LearnAhead Datenbankstruktur	17
3.6.1 Die Seitenhierarchie in LearnAhead	22
3.6.2 Login und Registrierung	23
3.6.3 Home-Screen and Profilansicht	24
3.6.4 Erstellen einer Lernkategorie and Ansicht Lernkategorien	25
3.6.5 Ansicht Zusammenfassungen and Erstellen einer Zusammenfassung	26
3.6.6 Ansicht Fragen und Tests and Ansicht Fragen	27
3.6.7 Erstellen einer Frage and Ansichts Tests	28
3.6.8 Test	29
3.6.9 Test	30
3.6.10 Erstellen eines Lernziels	31
3.7.1 Flow-Chart-Diagramm	32
4.5.1 Gesamtkomposition	34
4.5.2 Frontend UI Navigation	35
4.5.3 Frontend UI Kommunikation mit ViewModel	35
4.5.4 Backend Komposition	36

Tabellenverzeichnis

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

Listingsverzeichnis

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 im Play Store 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
- DSGVO

2.1.4 Ausblick

Bei erfolgreichen Entwicklungsergebnissen soll die Lösung in Betrieb genommen und der Öffentlichkeit, per Play Store, 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 Jira¹ 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.

¹<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 im erweiterten Scrum erfolgt in Jira, hier ist der Backlog angelegt, in welchem die Sprint-Planung erfolgt. Meetings werden auf Discord durchgeführt. Die Dokumentation wird mit \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 gemergt 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) gemergt.

2.6 Evaluierung der Plattformen und Programmiersprachen für die App-Entwicklung

Die Lern-App gqqLearnAhead 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 [6]. 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 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. [3]

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. [5]

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. [2, S. 27]

„**Verstehen**“ hingegen bedeutet, dass ein Mensch ein Konzept erstellt und dieses somit auf unbekannte Fragen und neue Situationen anwenden kann. [2, 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. [2, 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. [2, S. 114–115]

3.1.2 Synapsen

3.1.3 Neuronen

3.1.4 Neuronales Netz

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. [2, 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. [7] 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.

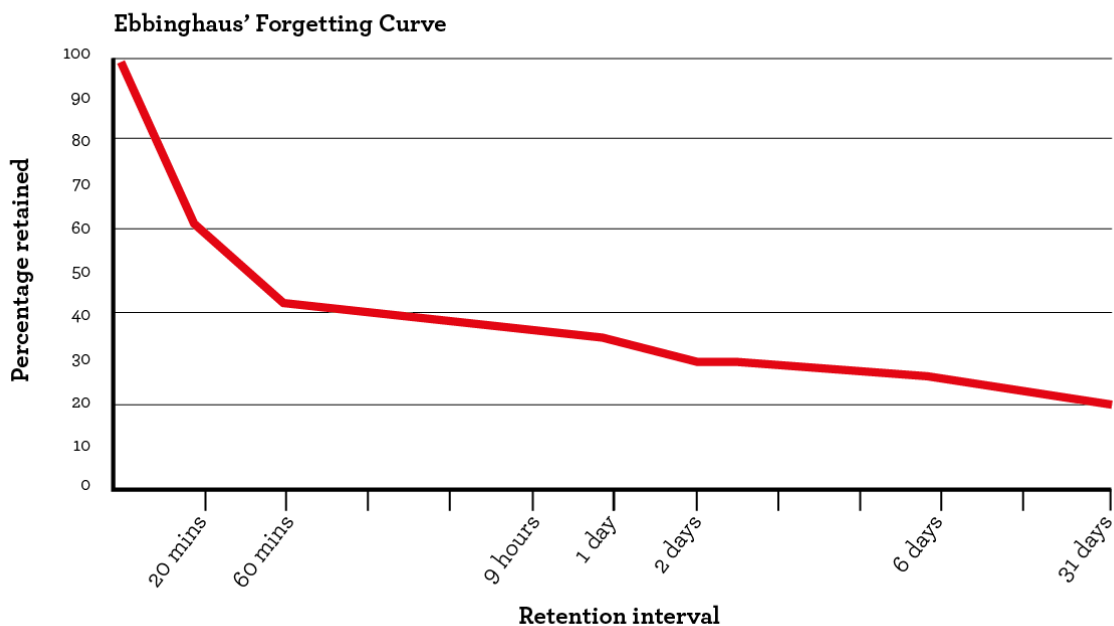


Abb. 3.1.1: Ebbinghaus Vergessenskurve

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. [7] 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.

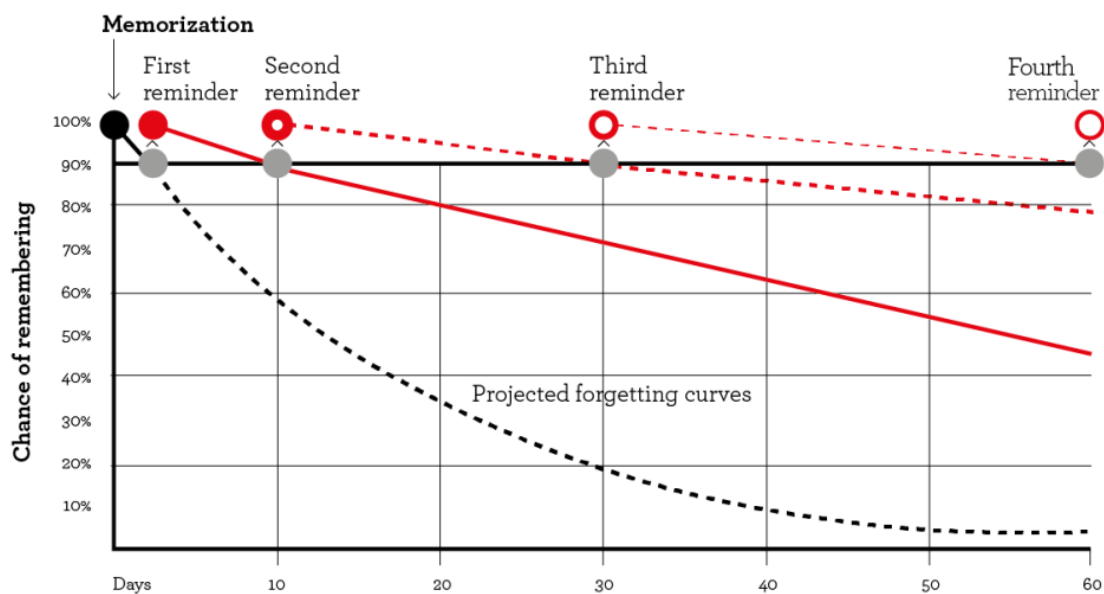


Abb. 3.1.2: Ebbinghaus verzögerte Vergessenskurve durch Wiederholungen

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. [2, 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. [2, 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.

3.2 Pflichtenheft

Das Pflichtenheft ist analog zum Arbeitspaketplan in diesem Dokument nicht genauer beschrieben, sondern in Form eines Backlog auf Jira² aufgeführt. Dort kann entweder der Reiter „Roadmap“ oder der Reiter „Board“ aufgerufen werden.

3.3 Use Case Diagramm

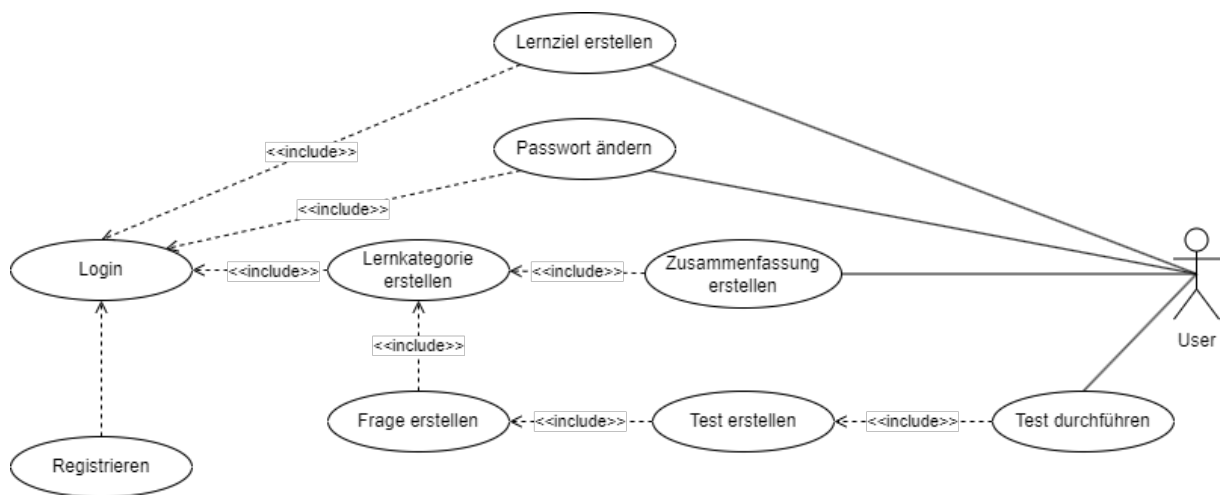


Abb. 3.3.1: Use Case Diagramm

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

3.4 Use Case Beschreibung

Name:	Login
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"> • Rückmeldung, dass falsche Zugangsdaten verwendet wurden • Möglichkeit, das Kennwort über die E-Mail zurückzusetzen <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> • keine Zustandsänderung
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.

Name:	Registrieren
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"> • Rückmeldung, dass E-Mail bereits verwendet wurde <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> • Keine Zustandsänderung • Account wird nicht angelegt
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.

Name:	Lernziel erstellen
Ziel:	Anlegen eines neuen Lernziels
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> • User muss eingeloggt sein • Lernziel mit gleichem Namen ist noch nicht erstellt
Nachbedingung:	Lernziel ist erstellt
Fehlerfälle:	<p>Lernziel mit gleichem Namen existiert bereits</p> <ul style="list-style-type: none"> • Rückmeldung, dass dieses Lernziel bereits existiert <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> • keine Zustandsänderung • Lernziel wird nicht angelegt
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:	

Name:	Lernkategorie erstellen
Ziel:	Anlegen einer neuen Lernkategorie
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> • User muss eingeloggt sein • Lernkategorie mit gleichem Namen ist noch nicht erstellt
Nachbedingung:	Lernkategorie ist erstellt
Fehlerfälle:	<p>Lernkategorie mit gleichem Namen existiert bereits</p> <ul style="list-style-type: none"> • Rückmeldung, dass diese Lernkategorie bereits existiert <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> • keine Zustandsänderung • Lernkategorie wird nicht angelegt
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:	

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

Name:	Frage erstellen
Ziel:	Anlegen einer neuen Frage
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> • User muss eingeloggt sein • Frage mit gleichem Index ist noch nicht in der gleichen Lernkategorie erstellt
Nachbedingung:	Frage für Lernkategorie ist erstellt
Fehlerfälle:	<p>Frage mit gleichem Namen existiert bereits in dieser Lernkategorie</p> <ul style="list-style-type: none"> • Rückmeldung, dass Frage mit gleicher Bezeichnung bereits existiert <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> • keine Zustandsänderung • Frage wird nicht angelegt
Akteure:	User
Auslösendes Event:	Benutzer öffnet eine Lernkategorie, klickt auf „Tests / Fragen“ und danach auf „Frage erstellen“
Beschreibung/ Erweiterungen:	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.
Alternativen:	

Name:	Test erstellen
Ziel:	Anlegen eines neuen Tests
Kategorie:	Primär
Vorbedingung:	<ul style="list-style-type: none"> • User muss eingeloggt sein • Test mit gleichem Namen ist noch nicht in der gleichen Lernkategorie erstellt
Nachbedingung:	Test für Lernkategorie ist erstellt
Fehlerfälle:	<p>Test mit gleichem Namen existiert bereits in dieser Lernkategorie</p> <ul style="list-style-type: none"> • Rückmeldung, dass Test mit gleicher Bezeichnung bereits existiert <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> • keine Zustandsänderung • Test wird nicht angelegt
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 Datenbankstruktur

3.5.1 Datenbankmodell

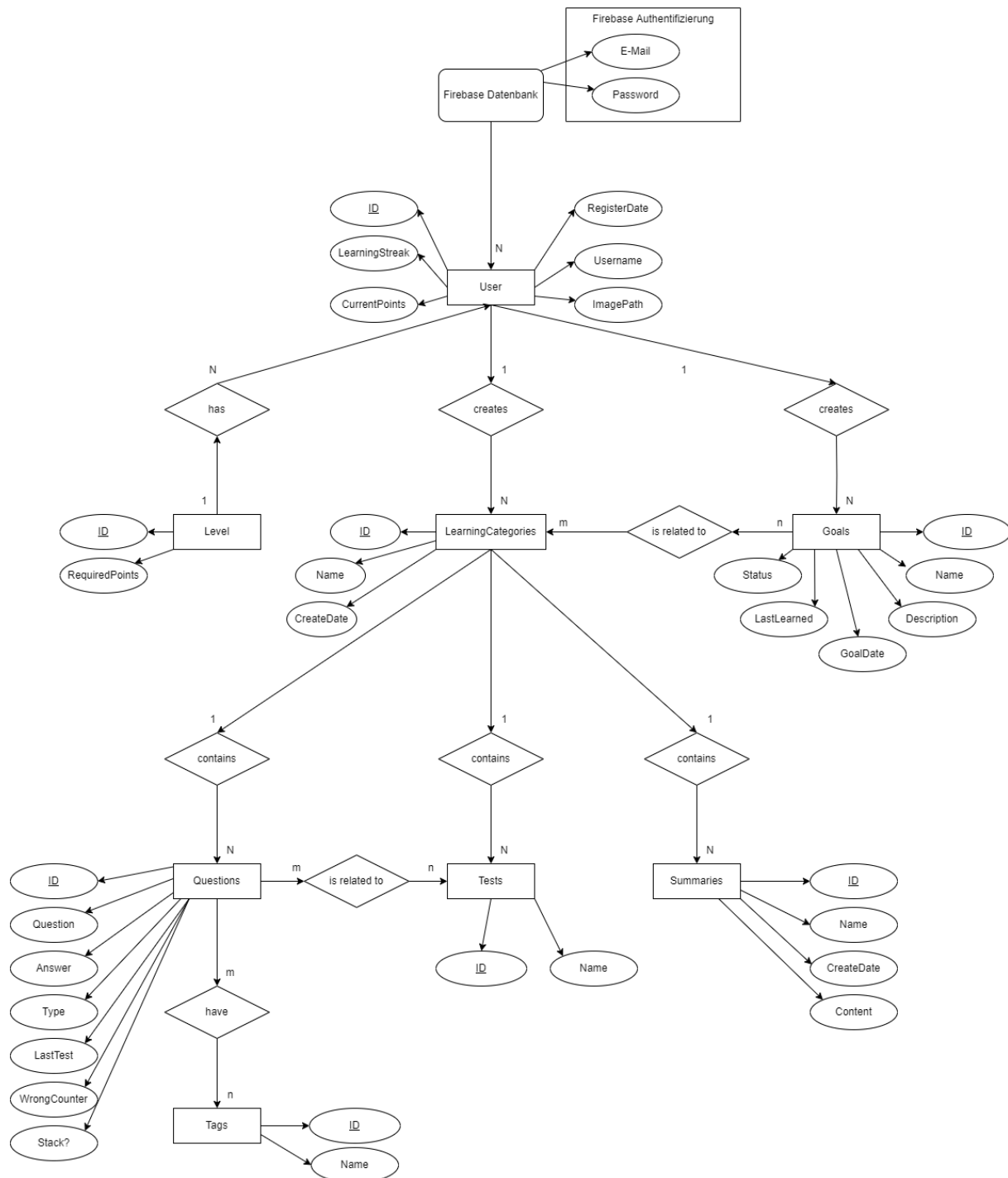


Abb. 3.5.1: LearnAhead Datenbankstruktur

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
Password	string	-	Nein	-	-	Das Passwort 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.

Level

Attribut	Datentyp	Länge	Null	Default	Schlüssel	Beschreibung
ID	string	-	Nein	-	P	
RequiredPoints	number	-	Nein	-	-	Dies gibt die benötigten Punkte für das spezielle Level an
Level	number	-	Nein	-	-	Dies gibt das spezielle Level anhand der Punkte an

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
LastLearned	timestamp	-	Ja	Null	-	Dies gibt, wann das Lernziel das letzte mal gelernt wurde.

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.

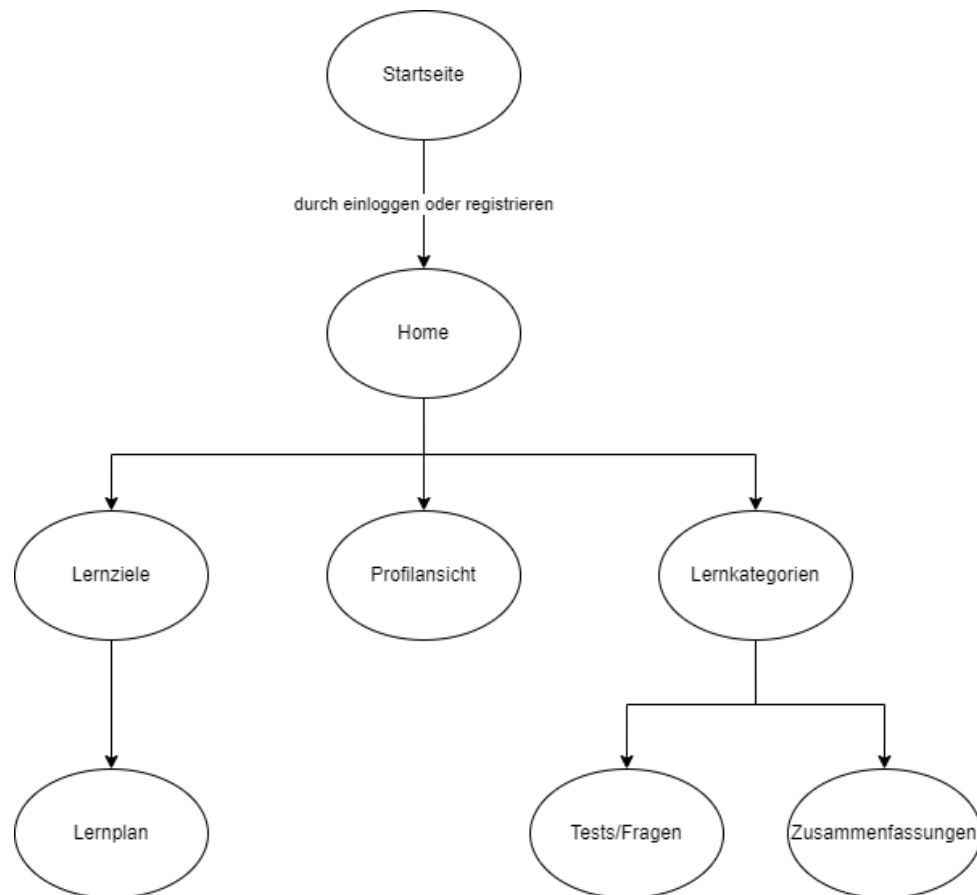


Abb. 3.6.1: Die Seitenhierarchie in LearnAhead

3.6.2 UI-Mockups

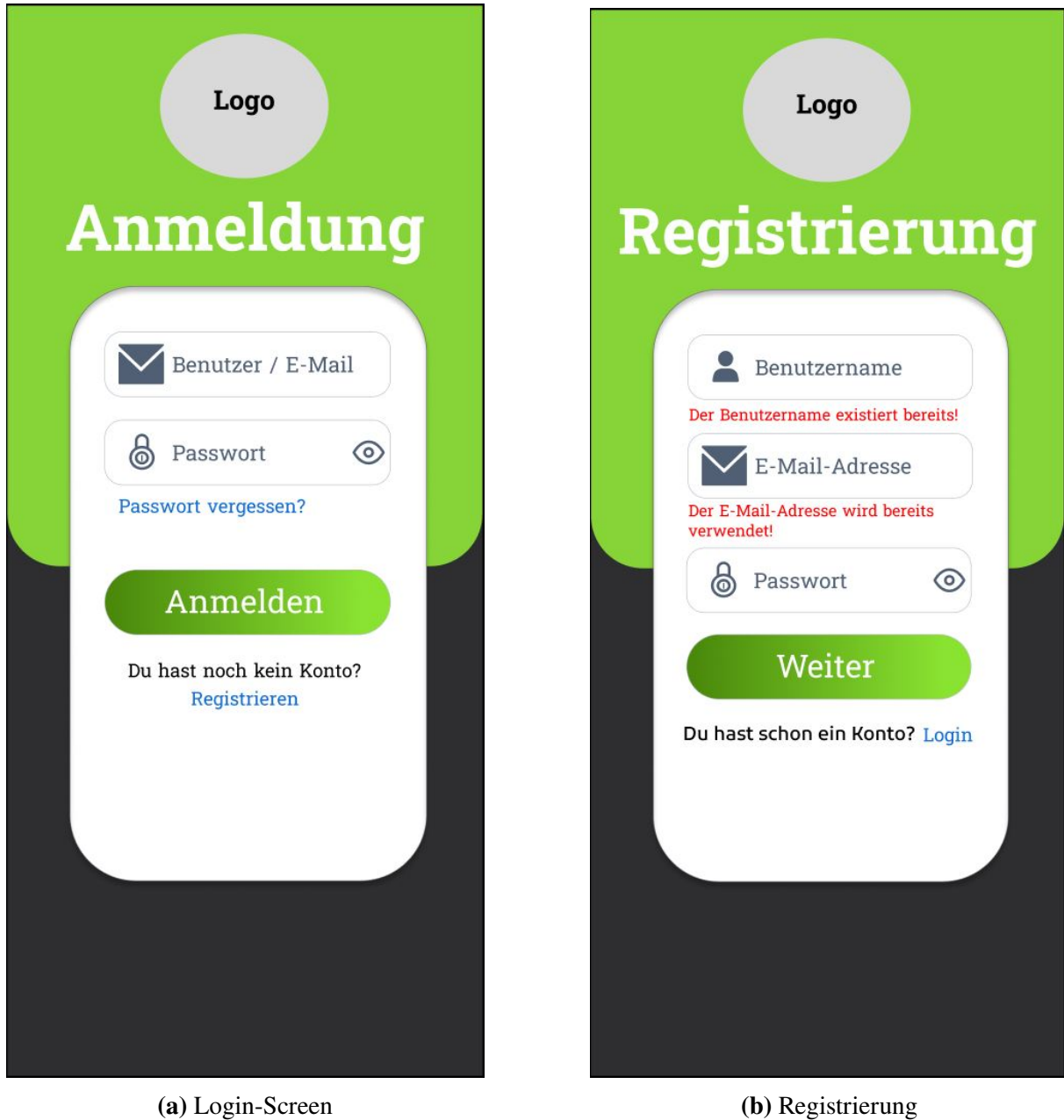
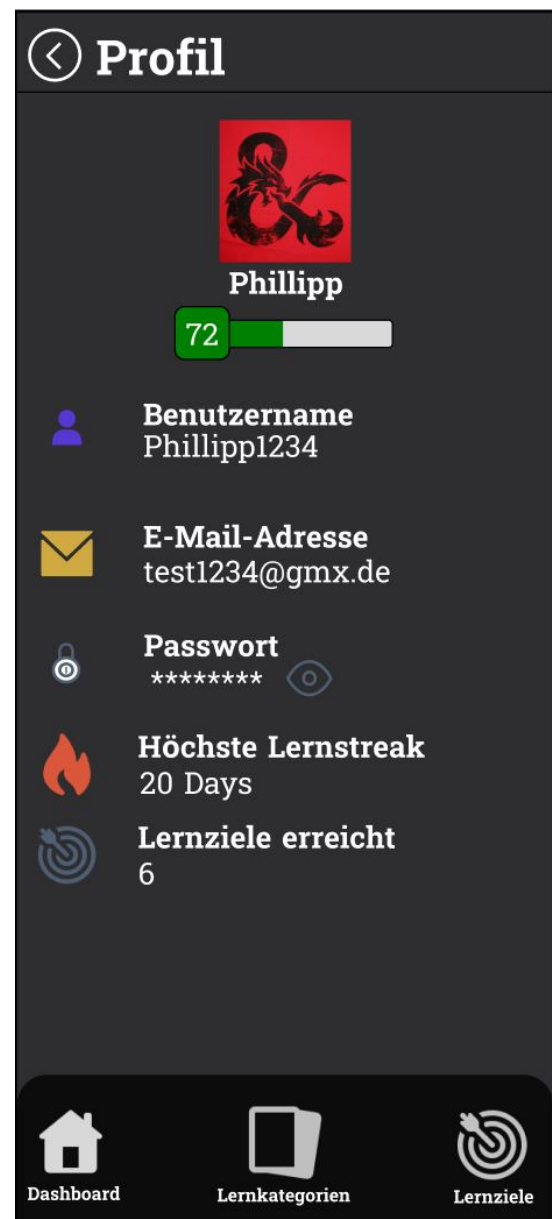


Abb. 3.6.2: Login und Registrierung

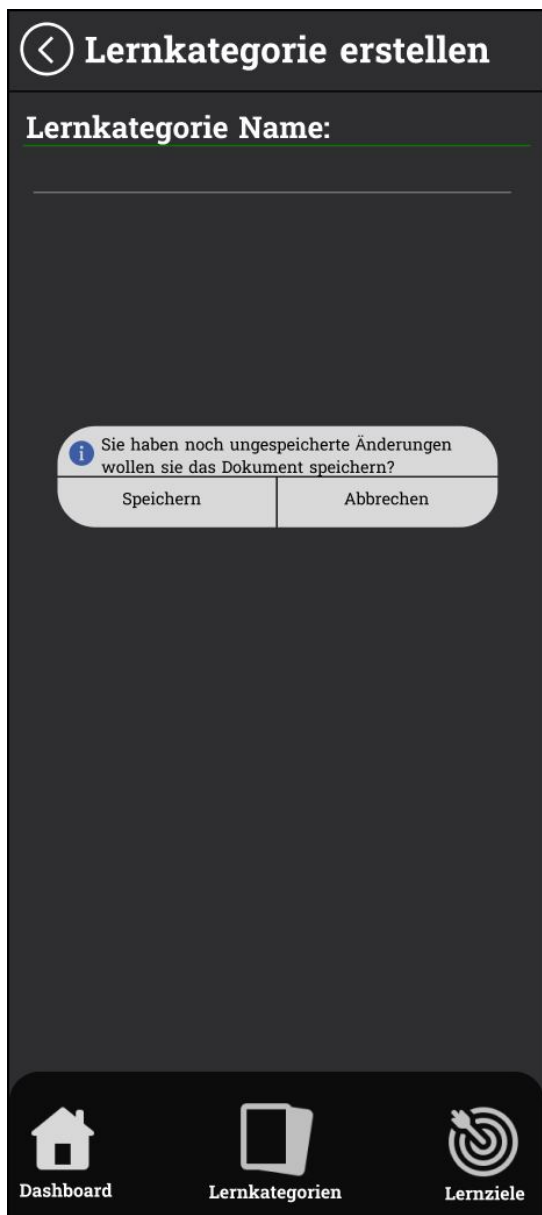


(a) Home-Screen



(b) Profilansicht

Abb. 3.6.3: Home-Screen and Profilansicht

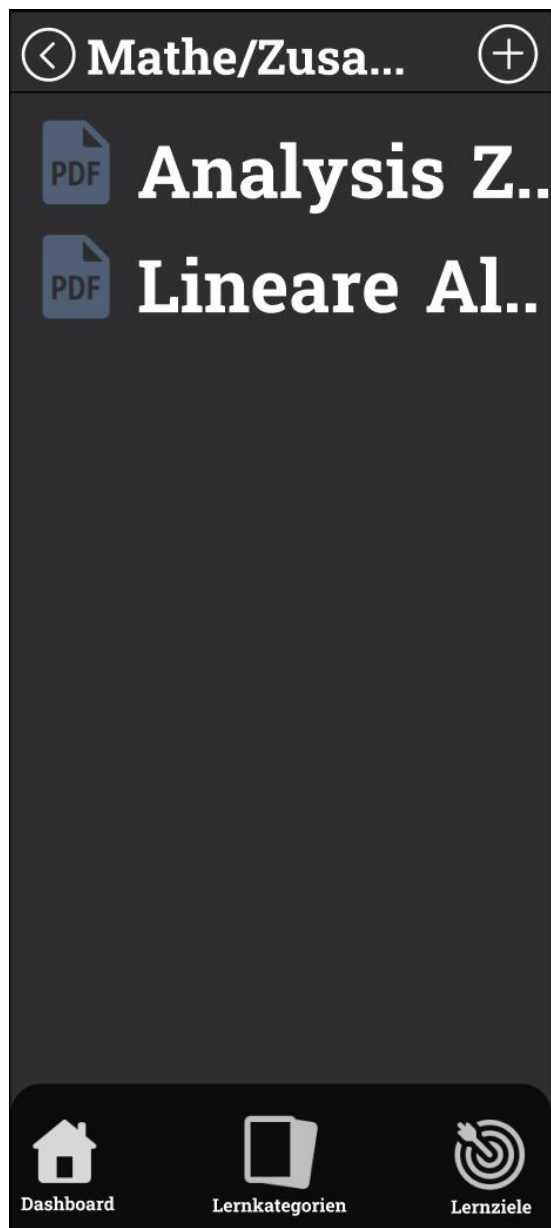


(a) Erstellen einer Lernkategorie

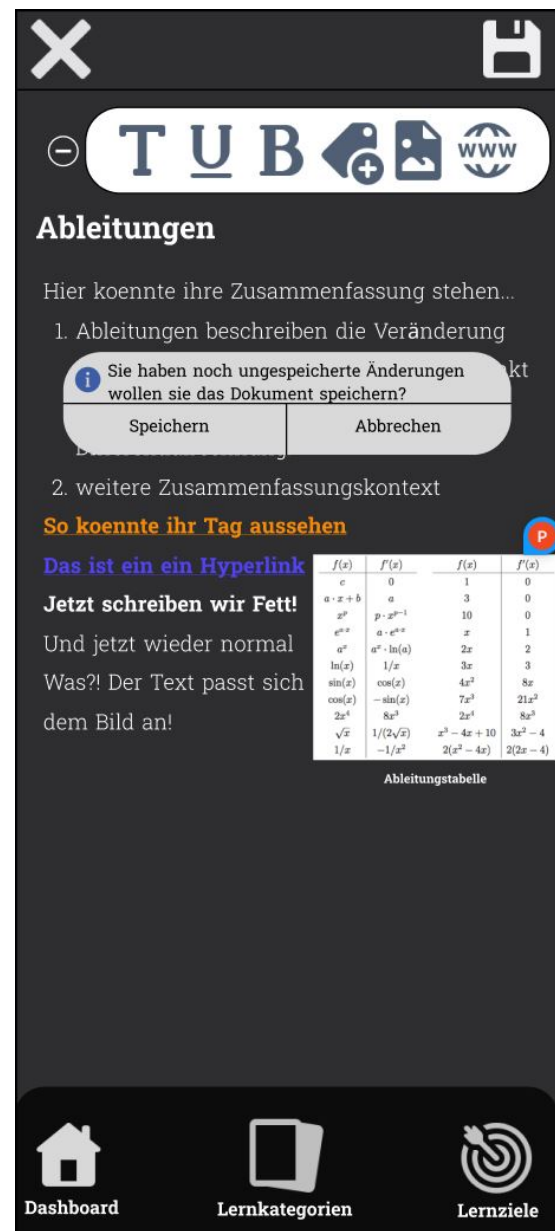


(b) Ansicht Lernkategorien

Abb. 3.6.4: Erstellen einer Lernkategorie and Ansicht Lernkategorien

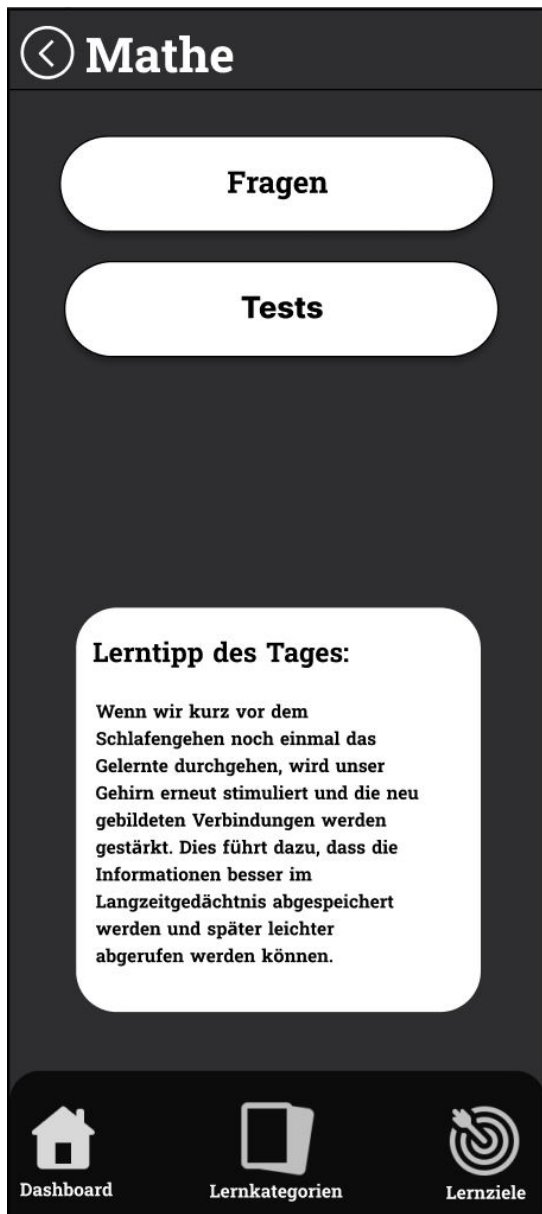


(a) Ansicht Zusammenfassungen

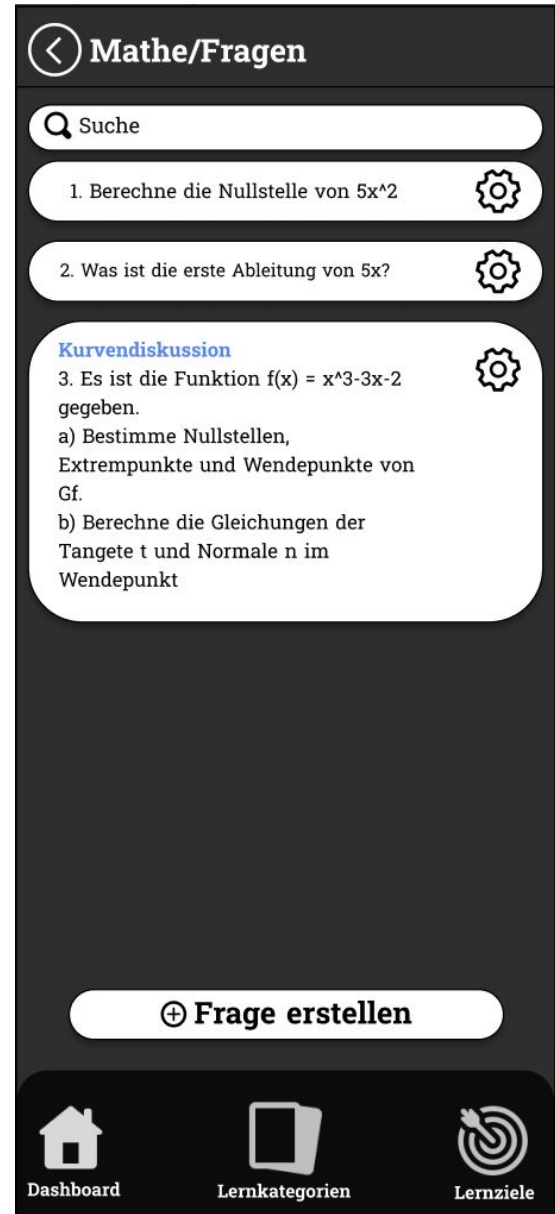


(b) Erstellen einer Zusammenfassung

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

← **Frage erstellen**

Wähle eine Art von Test aus...

- ✓ Karteikarte - Umdrehen
- ✓ Tags hinzufügen

Frage:

[Kurvendiskussion](#)

3. Es ist die Funktion $f(x) = x^3 - 3x - 2$ gegeben.

a) Bestimme Nullstellen, Extrempunkte und Wendepunkte von Gf.

b) Berechne die Gleichungen der Tangente t und Normale n im Wendepunkt

Antwort / Lösung:

Bestimme zuerst die Nullstellen von $f(x) = x^3 - 3x - 2$ indem du die Funktion gleich 0 setzt:

$$0 = x^3 - 3x - 20 = x$$

Die erste Nullstelle muss erraten werden. Durch ausprobieren ermittelt

✓ Frage hinzufügen

Dashboard Lernkategorien Lernziele

(a) Erstellen einer Frage

← **Mathe/Tests**

Anmerkung: Klicke auf einen Test um ihn zu editieren. Wenn du ihn starten möchtest, gehe bitte zurück und klicke auf "Test Starten"

Analysis zuletzt aktualisiert vor 2 Tagen

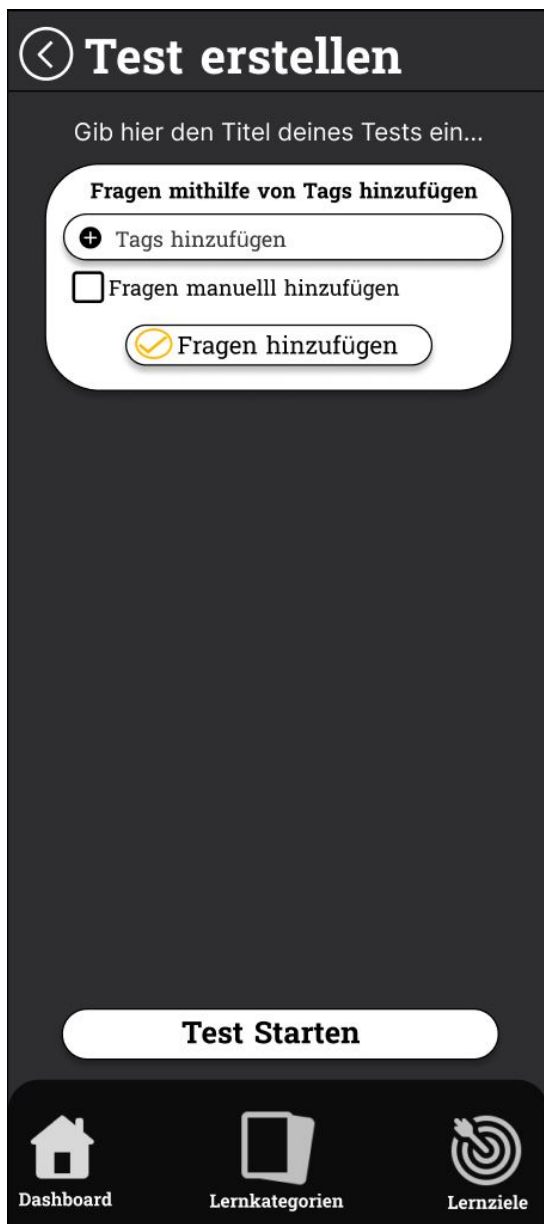
Lineare Algebra zuletzt aktualisiert vor 5 Tagen

⊕ **Test erstellen**

Dashboard Lernkategorien Lernziele

(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).

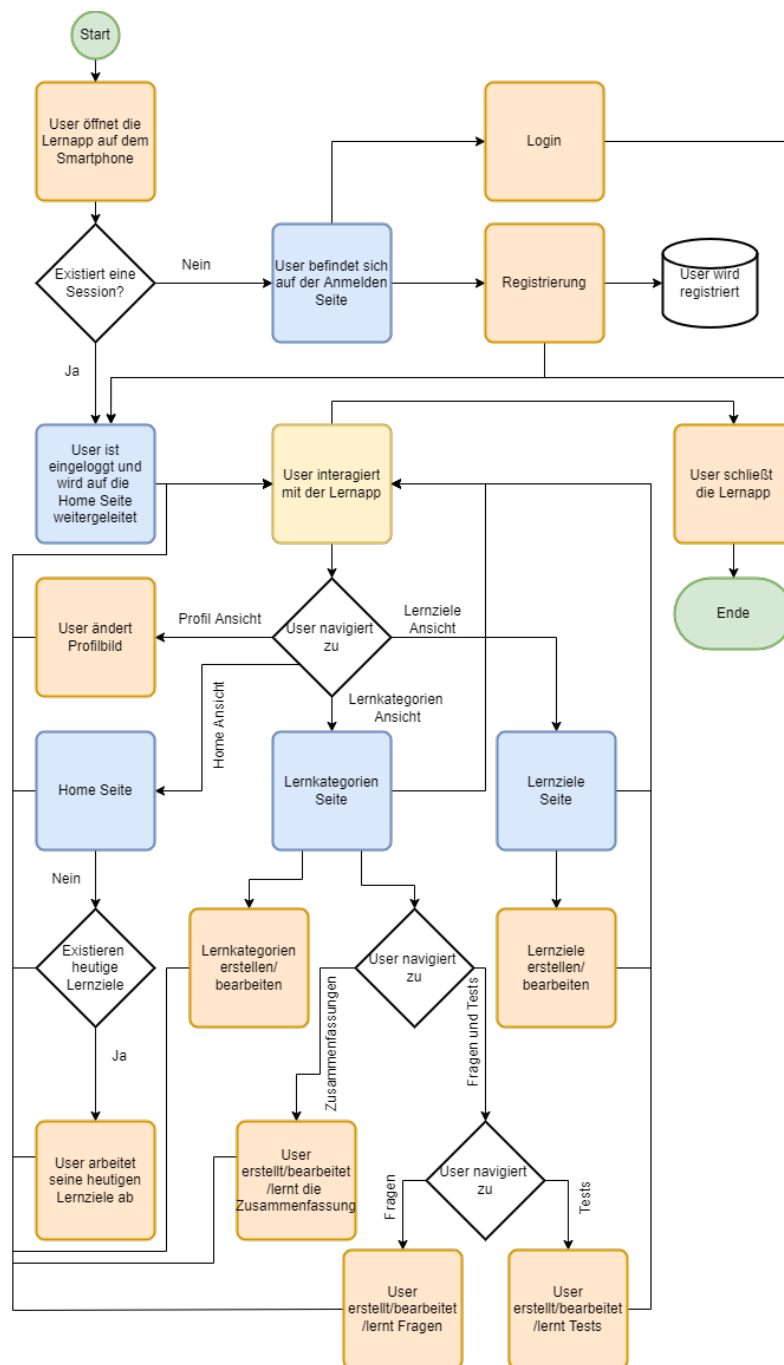


Abb. 3.7.1: Flow-Chart-Diagramm

4 Entwurf

4.1 Geschäftsfälle anhand des BPMN Workflows

4.2 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. [8] 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 4.3). 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. [4]

4.3 Auswahl der Klassenbibliotheken/Frameworks

4.4 Design Patterns für relevante Problemstellungen

4.5 Software-Komponenten

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

4.5.1 Gesamtkomposition

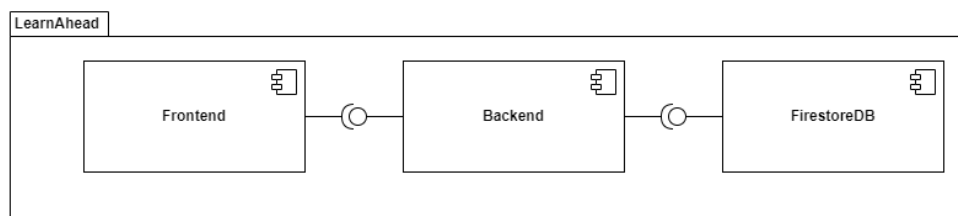


Abb. 4.5.1: Gesamtkomposition

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 4.6 und Klassen Diagramm 4.7.

4.5.2 Frontend

Es gibt zwei Diagramme für das Frontend. Das Diagramm 4.5.2 zeigt die Navigation zwischen den einzelnen Pages.

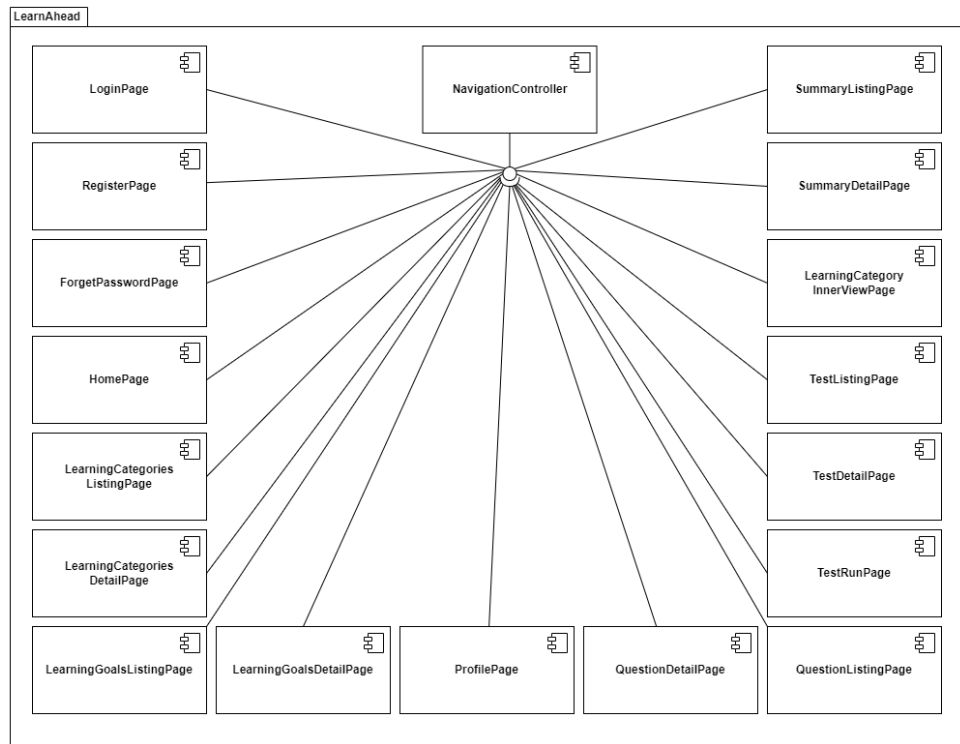


Abb. 4.5.2: Frontend UI Navigation

Das Diagramm 4.5.3 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 4.4 zu finden.

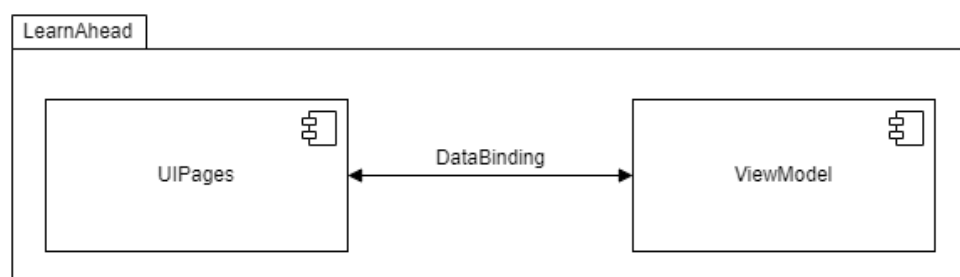


Abb. 4.5.3: Frontend UI Kommunikation mit ViewModel

4.5.3 Backend

Das Diagramm 4.5.4 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.

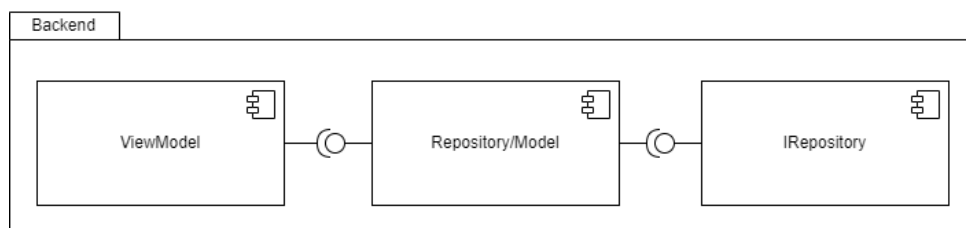


Abb. 4.5.4: Backend Komposition

4.6 Deployment Diagramm

4.7 Klassen Diagramm

4.8 Aktivitätsdiagramm

4.9 Sequenzdiagramme

4.10 Prototyp (optional)

5 Ergebnisse

5.1 Der finale Zeitplan

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 - 08.06.2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Fragen und Tests.
Erstellung von Zusammenfassungen	12.05.2023 - 02.06.2023	Implementierung der Funktion zum Erstellen sowie Verwalten von Zusammenfassungen von Lernkategorien.
Optimale Pausenberechnung realisieren	08.06.2023 - 14.06.2023	Erstellung eines Algorithmus, welcher den Nutzer die optimale Pause vorschlägt sowie erinnert.
Optimale Lernplan generieren	08.06.2023 - 14.06.2023	Erstellung eines optimalen Lernplans auf Basis der Lernziele.
Durchführen von Tests	20.06.2023 - 30.06.2023	Durchführung von umfassenden Tests, um die Qualität, Funktionalität und Stabilität der App sicherzustellen.
Bugs beheben	01.07.2023 - 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.

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] Henning Beck. *Das neue Lernen heißt Verstehen*. ger. 3. Auflage. Berlin: Ullstein, 2021. ISBN: 978-3-550-20049-6 (siehe S. 7–10).
- [3] 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).
- [4] *Firebase Pricing*. en. URL: <https://firebase.google.com/pricing> (besucht am 07.07.2023) (siehe S. 33).
- [5] 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).
- [6] *Registrierung - Support - Apple Developer*. URL: <https://developer.apple.com/de/support/enrollment/> (besucht am 10.06.2023) (siehe S. 6).
- [7] 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. 8, 9).
- [8] *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. 33).

A Anhang