

Abschlussprüfung Winter 2026 Fachinformatiker für Anwendungsentwicklung Dokumentation zur betrieblichen Projektarbeit

Entwicklung eine „Interaktive UML Lern-App mit Quiz, Notizen, Lernpfaden (Levels) und Kursverwaltung“

25. September 2025

Marburg, 2025

Eingereicht von:

Josiane Kanouo Maneyo

Umschülerin zur Fachinformatikerin für Anwendungsentwicklung



IAD GmbH Marburg

Neue Kasseler Straße 62e

35039 Marburg

Betreuer: Marcus Brauer

Abgabedatum:

Inhaltsverzeichnis

Abbildungsverzeichnis III

Tabellenverzeichnis IV

Listings V

Abkürzungsverzeichnis VI

Inhalt

[1 Einleitung 1](#_Toc210643342)

[1.1 Projektumfeld 1](#_Toc210643343)

[1.2 Projektziel 1](#_Toc210643344)

[1.3 Projektbegründung 2](#_Toc210643345)

[1.4 Projektschnittstellen 2](#_Toc210643346)

[1.5 Projektabgrenzung 2](#_Toc210643347)

[2 Projektplanung 2](#_Toc210643348)

[2.1 Projektphasen 2](#_Toc210643349)

[2.2 Abweichungen vom Antrag 2](#_Toc210643350)

[2.3 Ressourcenplanung 2](#_Toc210643351)

[2.4 Vorgehensmodell 3](#_Toc210643352)

# 1 Einleitung

## Projektumfeld

Das Projekt wird im Rahmen der Umschulung zur Fachinformatikerin für Anwendungsentwicklung an der IAD Bildungszentrum Marburg GmbH¹ durchgeführt. Die IAD, Teil der IAD Group mit über 50 Jahren Erfahrung, ist einer der größten privaten Bildungsanbieter in Deutschland für IT und Management. Sie bietet praxisorientierte Ausbildungen und Umschulungen an, darunter den dualen Fachinformatiker für Anwendungsentwicklung mit Fokus auf Softwareentwicklung (z. B. Kotlin, Mobile Apps) und Datenhaltung. Der Marburg-Standort (Wilhelm-Röpke-Straße 11) dient als zentraler Hub mit modernen IT-Labs und Kooperationen mit der Agentur für Arbeit. Jährlich werden ca. 8.000 Teilnehmer bundesweit qualifiziert, mit hohen Erfolgsquoten: 90 % der Absolventen finden direkt einen Job, 95 % Zufriedenheit und 98 % Bestehensrate bei Prüfungen. Die Projektarbeit erfolgt als Einzelprojekt. Die Verantwortung umfasst Analyse, Entwurf, Implementierung, Test und Dokumentation. Fachliche Betreuung erfolgt durch Marcus Brauer, Dozent an der IAD, der die technische Umsetzung unterstützt und die Einhaltung der IHK-Vorgaben überwacht. Zielgruppe der Anwendung sind Lernende und Lehrkräfte, die eine digitale Plattform zur Kursauswahl, Bearbeitung von Quizfragen, Verwaltung von Fehlerlisten und Erstellung persönlicher Notizen nutzen können. Die Anwendung adressiert den Übergang von statischen Papierübungen zu interaktiven Tools, die ortsunabhängiges Training mit direktem Feedback und zentraler Auswertung ermöglichen. Neben der bestehenden Lernumgebung wird mit der App eine moderne, mobile Erweiterung geschaffen, die neuen Lernenden den Einstieg durch ein interaktives Tutorial erleichtert und Rückmeldungen direkt in die Weiterentwicklung einfließen lässt.

Projektziel

Ziel des Projekts ist die Entwicklung einer interaktiven Android-Anwendung², die Lernenden und Lehrkräften des IAD Bildungszentrums ein Werkzeug zum Üben, Wiederholen und Verwalten von Lerninhalten bietet. Die Anwendung ersetzt traditionelle, statische Methoden und schafft eine strukturierte, motivierende Lernumgebung für UML-Diagramme.

Die App umfasst Funktionen wie Kursauswahl, Quiz mit direktem Feedback, Fehlerlisten, persönliche Notizen, ein Levelsystem und einen Prüfungsmodus mit Timer. Administratoren erhalten einen Bereich zur Verwaltung von Kursen und Fragen. Technisch wird die Anwendung mit Kotlin³ in Android Studio umgesetzt und nutzt Firebase⁴ für Authentifizierung, Datenspeicherung in Firestore und Medienverwaltung in Storage. Sicherheitsregeln und Offline-Funktionen gewährleisten Stabilität auch ohne Internetverbindung, wodurch Lernende von transparenter Fortschrittskontrolle profitieren und Lehrkräfte zentrale Inhaltsverwaltung erhalten. Neben den Kernfunktionen wird die App um zwei wesentliche Erweiterungen ergänzt: ein Onboarding-Tutorial, das neuen Nutzenden in wenigen Schritten die wichtigsten Funktionen erklärt, und eine In-App-Feedback-Funktion, mit der Rückmeldungen direkt an die Entwicklerin gesendet werden können. Dadurch wird der Einstieg erleichtert, die Benutzerfreundlichkeit erhöht und die kontinuierliche Verbesserung der Anwendung unterstützt.

Projektbegründung

Der Unterricht an der IAD basiert derzeit überwiegend auf Papierübungen und Word-Dokumenten für UML-Diagramme. Diese Methode ist unflexibel, bietet kein direktes Feedback und erlaubt keine zentrale Verwaltung oder Auswertung. Lehrkräfte haben dadurch keinen einfachen Überblick über den Lernstand, und Ergebnisse lassen sich nicht systematisch dokumentieren oder vergleichen. Die geplante Anwendung digitalisiert diese Prozesse, erhöht Transparenz und steigert die Motivation. Lernende können gezielt Fehler wiederholen, ihren Fortschritt verfolgen und individuelle Notizen anlegen.

Eine Eigenentwicklung wurde gewählt, da bestehende Lernplattformen wie Moodle oder Quizlet keine Kombination aus Fehlerwiederholung, Notizen und Levelsystem bieten. Die Lösung ist somit speziell auf die Bedürfnisse der Umschulungsteilnehmenden zugeschnitten und unterstützt die Digitalisierung des Unterrichts, wodurch der Lernprozess effizienter, praxisnäher und motivierender gestaltet wird als mit herkömmlichen Papierübungen. Durch das Onboarding-Tutorial wird der Einstieg in die Anwendung auch für technisch weniger erfahrene Lernende deutlich einfacher. Die integrierte Feedback-Funktion ermöglicht es, Verbesserungsvorschläge und Fehler direkt zu melden, was eine nachhaltige Weiterentwicklung und Qualitätssicherung unterstützten.

Projektschnittstellen

Die Anwendung kommuniziert ausschließlich mit Firebase-Diensten (Authentication, Cloud Firestore, Storage, Security Rules) über sichere HTTPS-Verbindungen. Offline-Nutzung wird durch lokale Datenspeicherung ermöglicht. Beteiligte Personen sind Marcus Brauer als Auftraggeber, Josiane Kanouo Maneyo als Entwicklerin und Testpersonen aus der IAD. Externe Systeme oder APIs werden nicht integriert. Zusätzlich wird die Feedback-Sammlung in Firebase zur Speicherung eingehender Rückmeldungen genutzt.

1.5 Projektabgrenzung

Das Projekt beschränkt sich auf die Entwicklung der Android-Anwendung mit den im Antrag festgelegten Kernfunktionen. Ausgeschlossen sind eine Webversion, KI-Funktionen, komplexe Statistiken oder die Veröffentlichung im Google Play Store. Das Projekt endet mit einem funktionsfähigen Prototyp, der die geforderten Anforderungen vollständig erfüllt. Es wird keine UML-Diagramm-Erstellung integriert; der Fokus liegt auf quizbasiertem Lernen und Verwaltung.

2 Projektplanung

2.1 Projektphasen

Das Projekt umfasst einen Zeitrahmen von 80 Stunden und gliedert sich in acht aufeinanderfolgende Phasen: Analyse, Konzept, Entwurf, Implementierung, Test und Dokumentation. Die Analysephase diente der Anforderungsermittlung, gefolgt von Entwurf und Programmierung sowie Tests zur Sicherstellung der Funktionalität. Die Soll-Phase und Implementierung umfassen zusätzlich die Entwicklung eines Onboarding-Tutorials sowie einer In-App-Feedback-Funktion, um die Benutzerfreundlichkeit zu erhöhen. Eine detaillierte Übersicht der Stundenverteilung ist im Anhang A1 enthalten.

2.2 Abweichungen vom Antrag

Gegenüber dem ursprünglichen Antrag wurde der Funktionsumfang leicht erweitert.  
Ergänzt wurden ein Onboarding-Tutorial und eine In-App-Feedback-Funktion, um das Nutzererlebnis zu verbessern. Der Zeitrahmen von 80 Stunden blieb unverändert; der Mehraufwand wurde durch optimierte Arbeitsschritte ausgeglichen.

2.3 Ressourcenplanung

Zur Umsetzung stehen ein Windows-Laptop, ein Android-Smartphone sowie die Software Android Studio, Kotlin und Firebase zur Verfügung. Versionierung und Datensicherung erfolgen über GitHub. Zusätzliche Tools wie Draw.io und Microsoft Word unterstützen die Erstellung von Diagrammen und der Dokumentation.

2.4 Vorgehensmodell

Es wurde ein phasenorientiertes Vorgehensmodell mit agilen Elementen angewandt. Die Struktur orientiert sich am Wasserfallmodell⁵, ergänzt durch kurze Feedbackzyklen während der Implementierung. Dies ermöglichte eine flexible Reaktion auf Designänderungen und Testergebnisse bei gleichzeitiger Einhaltung eines klaren Ablaufs. Während der Implementierung wurden nach jeder abgeschlossenen Funktion (Login, Quiz, Feedback, Tutorial) kurze Test- und Feedback-Zyklen durchgeführt, um Stabilität und Nutzerfreundlichkeit sicherzustellen.

Fußnoten

¹ IAD Bildungszentrum Marburg GmbH: [www.iad.de](http://www.iad.de).

² Android-Anwendung: Basierend auf Kotlin für mobile Entwicklung, siehe developer.android.com.

³ Kotlin: Moderne Programmiersprache für Android, siehe kotlinlang.org.

⁴ Firebase: Cloud-Plattform von Google für Authentifizierung und Daten, siehe firebase.google.com.

⁵ Wasserfallmodell: Traditionelles Modell der Softwareentwicklung.   
  
  
3 Analysephase

3.1 Ist-Analyse

Der Unterricht an der IAD Bildungszentrum Marburg GmbH basiert derzeit auf klassischen Lehrmethoden, bei denen die Lernenden UML-Diagramme in Papier- oder PDF-Form üben. Eine unmittelbare Rückmeldung zu den Ergebnissen erfolgt nicht; Korrekturen werden von Lehrkräften manuell durchgeführt und können nicht zentral dokumentiert werden. Auch der individuelle Lernfortschritt ist nicht transparent einsehbar. Vorhandene Tools wie draw.io oder PlantUML² ermöglichen zwar das Zeichnen von Diagrammen, unterstützen jedoch keinen interaktiven Lernprozess mit Feedback-, Notiz- oder Fortschrittsfunktionen. Das Fehlen einer digitalen, benutzerfreundlichen Lösung führt zu hohem Zeitaufwand und geringerer Motivation bei den Lernenden. Eine moderne, mobile App-Lösung kann diese Probleme beseitigen, indem sie Lernfortschritte automatisch erfasst, Fehler dokumentiert und Lerninhalte flexibel zur Verfügung stellt.

3.2 Soll-Analyse

Ziel dieses Projekts ist die Entwicklung einer interaktiven Android-Anwendung³, die den Lernprozess für UML-Diagramme digitalisiert und strukturiert. Die App soll Lernenden ermöglichen, eigenständig zu üben, sofort Feedback zu erhalten und ihre Fehler nachzuvollziehen. Lehrkräfte erhalten gleichzeitig ein einfaches Werkzeug zur Pflege von Inhalten und Fragen. Im Rahmen der Projektweiterentwicklung wurde der Funktionsumfang gezielt erweitert. Neben den Kernfunktionen wie Login, Kurswahl, Quiz, Fehlerliste, Notizen und Levelsystem enthält die App nun ein kurzes Onboarding-Tutorial, das neue Nutzende in wenigen Schritten durch die Hauptbereiche führt, sowie eine In-App-Feedback-Funktion, über die Anregungen oder Fehlermeldungen direkt an die Entwicklerin übermittelt werden können. Beide Erweiterungen erhöhen die Benutzerfreundlichkeit und unterstützen eine kontinuierliche Verbesserung der Anwendung. Die vollständige Funktionsübersicht befindet sich im Anhang A2.

3.3 Wirtschaftlichkeitsbetrachtung

Das Projekt wird im Rahmen der Umschulung zur Fachinformatikerin für Anwendungsentwicklung durchgeführt und über Leistungen nach dem Sozialgesetzbuch III (Arbeitslosengeld I) finanziert⁴. Somit entstehen für den Bildungsträger keine zusätzlichen Kosten. Der Hauptnutzen liegt in der Effizienzsteigerung des Lernprozesses. Die App reduziert Papierverbrauch, spart Korrekturzeit und erhöht die Motivation der Lernenden durch direktes Feedback und sichtbare Fortschritte. Gleichzeitig ermöglicht die Feedback-Funktion eine laufende Optimierung der Inhalte und trägt somit zur Qualitätssicherung bei. Der geplante Zeitaufwand für das gesamte Projekt beträgt 80 Stunden, verteilt auf Analyse, Planung, Implementierung, Test und Dokumentation.

3.4 Qualitätsanforderungen

Die UML-Lern-App legt besonderen Wert auf eine hohe Benutzerfreundlichkeit und Stabilität. Die Oberfläche ist klar strukturiert und ermöglicht eine intuitive Navigation. Eine zuverlässige Offline-Funktion gewährleistet, dass bereits geladene Inhalte auch ohne Internetverbindung verfügbar sind. Hinsichtlich der Datensicherheit werden die Zugriffsrechte in Firestore so geregelt, dass jeder Nutzer nur seine eigenen Informationen sehen kann. Firebase Security Rules sichern alle Operationen ab. Der Quellcode folgt dem MVVM-Muster, um eine klare Trennung zwischen Logik und Oberfläche sowie eine einfache Erweiterbarkeit zu gewährleisten. Performance und Zuverlässigkeit werden durch lokales Caching, regelmäßige Tests und optimierte Abfragen sichergestellt. Neue Nutzende werden durch das Onboarding-Tutorial schnell eingeführt, während die Feedback-Funktion eine aktive Kommunikation zwischen Entwicklerin und Anwendern ermöglicht.

3.5 Anwendungsfälle

Das System unterstützt zwei Akteure: Lernende und Administratoren. Lernende können nach dem Login Kurse auswählen, Fragen beantworten und Ergebnisse direkt einsehen. Fehler werden gespeichert und können wiederholt geübt werden. Administratoren verwenden die App, um Kurse und Fragen zu pflegen und Inhalte laufend zu aktualisieren. Das Onboarding-Tutorial erleichtert den Einstieg für neue Nutzende, während die Feedback-Funktion Rückmeldungen an die Entwicklerin ermöglicht. Eine grafische Darstellung der Anwendungsfälle befindet sich im Anhang A3.

3.6 Verweis auf das Lastenheft

Alle funktionalen und nicht-funktionalen Anforderungen sind im Anhang A4 – Lastenheft detailliert beschrieben. Das Lastenheft definiert Muss- und Kann-Kriterien sowie Abgrenzungen des Projekts und bildet die Grundlage für die Abnahme.

¹ IAD Bildungszentrum Marburg GmbH – https://www.iad-bildung.de  
² draw.io / PlantUML – Werkzeuge zur Erstellung von UML-Diagrammen  
³ JetBrains s.r.o.: Kotlin Documentation, 2025  
⁴ Förderung nach SGB III (Arbeitsagentur Marburg), 2025

4 Entwurfsphase

4.1 Systemarchitektur

Die Architektur der UML-Lern-App basiert auf dem Model-View-ViewModel-Muster (MVVM). Dieses Architekturkonzept trennt die Datenhaltung, die Benutzeroberfläche und die Geschäftslogik klar voneinander. Dadurch ist die Anwendung leicht wartbar und flexibel erweiterbar.

Das Modell (Model) speichert und verwaltet alle Anwendungsdaten, etwa Informationen zu Benutzern, Kursen und Fragen, und stellt die Verbindung zur Firebase-Datenbank her. Das ViewModel bildet die Mittelschicht zwischen Daten und Benutzeroberfläche: Es ruft Daten ab, bereitet sie für die Anzeige auf und überwacht Änderungen, damit die Oberfläche sich automatisch aktualisiert. Die Ansicht (View) umfasst die grafischen Layouts in XML, über die Lernende mit der App interagieren. Diese klare Trennung erleichtert die Entwicklung, das Testen und spätere Anpassungen einzelner Komponenten. Das gewählte Muster entspricht modernen Architekturprinzipien mobiler Anwendungen.

4.2Datenmodellierung  
Für die Datenspeicherung wird Firebase Firestore genutzt – eine dokumentenorientierte NoSQL-Datenbank, die sich besonders für mobile, cloudbasierte Anwendungen eignet. Die Datenstruktur wurde so gestaltet, dass sie einfach, logisch und erweiterbar bleibt.

Es existieren sechs zentrale Sammlungen: Benutzer, Kurse, Fragen, Fehler, Notizen und Feedback. Jede Sammlung hat eine eindeutige Aufgabe. In der Sammlung Benutzer werden persönliche Angaben gespeichert: Name, E-Mail, Rolle, Punkte, aktuelles Level und das Registrierungsdatum. Die Sammlung Kurse beschreibt verfügbare Lerneinheiten mit Titel, Beschreibung, Levelanforderung und Aktivitätsstatus. Fragen sind einem Kurs zugeordnet und bestehen aus dem Fragetext, mehreren Antwortmöglichkeiten, der richtigen Lösung und der erreichbaren Punktzahl. Fehler dokumentieren falsch beantwortete Aufgaben und dienen der gezielten Wiederholung. Notizen ermöglichen es Lernenden, eigene Kommentare oder Erklärungen zu speichern. Die Sammlung Feedback nimmt Rückmeldungen der Nutzerinnen und Nutzer zur Verbesserung der App auf.  
  
Zur Abbildung der Rollen und Rechte wurde im Benutzerobjekt ein Feld „Rolle“ integriert. Lernende können Inhalte lesen, Notizen erstellen und Feedback senden, während Administratorinnen und Administratoren zusätzliche Schreibrechte besitzen, um Kurse und Fragen zu verwalten. Diese Trennung wird über die Firestore-Sicherheitsregeln technisch kontrolliert.  
Das zugehörige ER-Diagramm im Anhang A6 zeigt die Beziehungen zwischen den einzelnen Sammlungen.  
  
4.3 Benutzeroberfläche

Die grafische Gestaltung orientiert sich am Material Design von Google und folgt dem IAD-Farbschema mit einem kräftigen Rot (#C62828) als Akzentfarbe und Weiß als Hintergrund. Das Erscheinungsbild ist bewusst schlicht und klar gehalten, mit großzügigen Abständen, klarer Typografie und einem hohen Kontrast. Als Schriftart kommt Roboto zum Einsatz.

Die wichtigsten Ansichten der App sind: die Login- und Registrierungsseite, die Kursübersicht, die Quiz-Ansicht, die Ergebnisanzeige, die Fehlerliste, die Notizverwaltung, die Feedback-Seite sowie die Einstellungsseite mit den Bereichen Allgemein, Konto, Datenschutz und Über die App. Nach der ersten Anmeldung wird zusätzlich ein kurzes Onboarding-Tutorial angezeigt, das die wichtigsten Funktionen vorstellt und danach deaktiviert wird.

Für Administratorinnen und Administratoren existiert eine eigene Verwaltungsoberfläche, über die neue Kurse oder Fragen angelegt, geändert und gelöscht werden können. Alle Ansichten wurden zunächst als Mockups entworfen (siehe Anhang A5) und anschließend als XML-Layouts umgesetzt. Symbole stammen aus den „Outlined Material Icons“, und sämtliche Oberflächen sind responsiv, um auf unterschiedlichen Gerätegrößen gleichmäßig zu wirken.

## 4.4 Technischer Aufbau

Die App besteht aus mehreren Aktivitäten und Fragmenten, die zusammen die einzelnen Module bilden. Beim Start prüft eine Splash-Activity, ob der Benutzer bereits angemeldet ist. Wenn nicht, öffnet sich die Login-Activity; andernfalls erfolgt die Weiterleitung in die Hauptansicht. Die Main-Activity enthält eine Navigationsleiste, über die man zwischen Kursen, Fehlerliste, Notizen und Profil wechseln kann.  
  
Die Quiz-Funktion wurde besonders sorgfältig entworfen. Vor Beginn kann die gewünschte Anzahl an Fragen ausgewählt werden (z. B. 5, 10 oder alle). Die Fragen und Antwortoptionen erscheinen in zufälliger Reihenfolge, damit kein Lerneffekt durch Positionserkennung entsteht. Bei Multiple-Choice-Aufgaben werden Teilpunkte vergeben: richtige Häkchen erhöhen, falsche reduzieren die Punktzahl, jedoch nie unter null. Alle Aufgaben sind gleich gewichtet; jede trägt dieselbe Punktzahl. Zusätzlich stehen zwei Modi zur Verfügung: ein Übungsmodus ohne Zeitlimit und ein Prüfungsmodus mit Timer, bei dem die Bearbeitung automatisch beendet wird, sobald die Zeit abläuft.  
  
Nach Abschluss des Quiz wird das Ergebnis angezeigt. Dabei berechnet die App die erreichten Punkte und aktualisiert das Lernlevel. Falsch beantwortete Fragen werden automatisch in der Fehlerliste gespeichert, um sie später erneut üben zu können. Die App funktioniert auch offline, da Firestore Daten zwischenspeichert und Änderungen automatisch synchronisiert, sobald eine Internetverbindung besteht.  
  
Für die Verwaltung ist ein eigener Administrationsbereich vorgesehen, der nur für Konten mit Administratorrolle sichtbar ist. Änderungen an Kursen oder Fragen werden direkt in der Datenbank gespeichert und stehen ohne App-Update sofort zur Verfügung.  
4.5 Zusammenfassung

Die Entwurfsphase bildet das technische und gestalterische Fundament der UML-Lern-App. Durch den Einsatz des MVVM-Musters bleibt die Struktur übersichtlich, während Firebase Firestore eine sichere und flexible Datenhaltung ermöglicht. Das Design nach den Material-Design-Richtlinien sorgt für ein modernes Erscheinungsbild und eine intuitive Benutzerführung. Die vollständige Umsetzung aller Kern- und Zusatzfunktionen – von Login, Kursverwaltung und Quiz bis hin zu Notizen, Fehlerlisten, Feedback, Einstellungen und Onboarding – macht die Anwendung zu einer umfassenden, digitalen Lernplattform für UML-Diagramme. Damit sind alle Grundlagen geschaffen, um in der nächsten Phase die Implementierung erfolgreich umzusetzen.

Fußnoten:

1. Microsoft (2025): MVVM-Guidelines – Architectural Patterns.

2. Google (2025): Material Design Principles.