

AberGymMobile

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von:

Antonio Peric
Antonio Kuvac

Betreuer:

Prof. Mag. Ing. Hans Christian Hammer

Projektpartner:

DI. Christian Aberger

Leonding, April 2023

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2023

Peric Antonio & Kuvac Antonio

Abstract

This diploma thesis aims to modernize the training plan process at the LionFit gym. Currently, training plans are created on a web application and printed as a PDF, and training data is recorded on the printed PDF. The project involves developing a native app using Android Studio and Flutter to provide a digital solution for managing and tracking training plans. The app will read data from the training plan database and update it with training data after each session, streamlining the entire process and making it more efficient and convenient for the gym's customers.

Zusammenfassung

Diese Diplomarbeit zielt darauf ab, den Trainingsplanprozess im Fitnessstudio LionFit zu modernisieren. Derzeit werden Trainingspläne in einer Web-Applikation erstellt und als PDF ausgedruckt, während die Trainingsdaten auf dem ausgedruckten PDF erfasst werden. Das Projekt beinhaltet die Entwicklung einer nativen App mit Android Studio und Flutter, um eine digitale Lösung für die Verwaltung und Überwachung von Trainingsplänen zu bieten. Die App wird Daten aus der Trainingsplandatenbank lesen und sie nach jeder Sitzung mit Trainingsdaten aktualisieren, um den gesamten Prozess zu vereinfachen und für die Kunden des Fitnessstudios effizienter und bequemer zu gestalten.

Danksagung

An dieser Stelle möchten wir uns bei den Personen bedanken, die uns während dieser Diplomarbeit unterstützt und motiviert haben. Gleich zu Beginn möchten wir uns gebührend bei Herrn Prof. Ing. Mag. Hans Christian Hammer bedanken, welcher uns immer mit seiner konstruktiven Kritik zur Seite stand.

Wir bedanken uns auch bei unseren Kollegen der 5CHIF, die uns stets mit ihrer Hilfsbereitschaft, und mit ihrem Wissen weitergeholfen haben.

Außerdem möchten wir uns bei unserem Projektpartner, DI. Christian Aberger bedanken, für die Bereitstellung von Ressourcen, und für die Beratung bei Fragen über unsere Diplomarbeit, und wie wir diese umsetzen sollen.

Abschließend möchten wir uns noch bei unseren Eltern bedanken, die uns während unserer Schulkarriere in der Höheren Technischen Bundeslehranstalt Leonding begleitet haben, und immer ein offenes Ohr für unsere Belange hatten.

Inhaltsverzeichnis

1	Pflichtenheft	1
1.1	Über das Team	1
1.2	Betreuer und Partner	1
1.3	Ausgangssituation und Zielstellung	1
1.4	Zielgruppe	2
1.5	Funktionale Anforderungen	3
1.6	App	3
2	Umfeldanalyse	4
3	Planung	5
3.1	Projektpartnermeetings	5
3.2	Use-Case Diagramm	5
3.3	Entity Relationship Diagram - ERD	7
4	Technologien	10
4.1	Flutter	10
4.2	Visual Studio Code	12
4.3	Docker	13
4.4	Android Studio	14
4.5	AdobeXD	17
4.6	IntelliJ	18
5	Umsetzung	20
6	Zusammenfassung	22
	Literaturverzeichnis	VII
	Abbildungsverzeichnis	VIII

Tabellenverzeichnis	IX
Quellcodeverzeichnis	X
Anhang	XI

1 Pflichtenheft

1.1 Über das Team

Tabelle 1: *Tabelle 1.1: Informationen über den Bertreuer und Partner.*

Betreuer	Prof. Mag. Ing. Hans Christian Hammer
Partner	DI. Christian Aberger

1.2 Bertreuer und Partner

Tabelle 2: *Tabelle 1.2: Informationen über das Projekt und das Team.*

Projektname	AGM - Abergymmobile
Teamleiter	Antonio Kuvac
Teammitglieder	Antonio Kuvac, Antonio Peric
Erstellt am	12.7.2023

1.3 Ausgangssituation und Zielstellung

1.3.1 Ausgangssituation

Aktuell ist der Prozess der Trainingsplangenerierung und -verwaltung im Fitnessstudio LionFit ineffizient und umständlich für Kunden. Daher möchte das Fitnessstudio eine digitale Lösung implementieren, um den Prozess zu vereinfachen und die Effizienz zu steigern. Derzeit werden die Trainingspläne in einer Web-Applikation erstellt und als PDF ausgedruckt, während die Trainingsdaten auf dem ausgedruckten PDF erfasst werden. Dieser Prozess ist unpraktisch, da die Kunden bei jeder Trainingseinheit ein ausgedrucktes PDF mitnehmen und die Daten manuell eingeben müssen.

1.3.2 Zieldefinition

- Entwicklung einer nativen App für Android Devices zur Abarbeitung von Trainingsplänen in einem Fitnessstudio.
- Die Daten des Trainingsplans werden aus der Datenbank der Trainingsplanverwaltung ausgelesen.
- Am Ende der Trainingssession wird die Trainingsdatenbank mit den Trainingsdaten ergänzt.

1.3.3 Nicht Ziele

- Entwicklung einer zu komplex gestalteten App die zum Verwenden vom Zettel Trainingsplan anregt.
- Entwicklung einer sehr fehlerhaften App.
- Entwicklung einer App dessen Design die Benutzer nicht anspricht.

1.4 Zielgruppe

Diese Zielgruppe umfasst sowohl regelmäßige Fitnessstudio-Besucher als auch Sportler, die unabhängig von einem Fitnessstudio trainieren und eine digitale Lösung für die Verwaltung ihrer Trainingspläne suchen. Besonders praktisch für diese Zielgruppe ist, dass sie jederzeit Zugang zu ihren Trainingsplänen und -daten auf ihrem Smartphone haben. Dies bietet mehr Flexibilität und Übersicht bei der Gestaltung und Überwachung des Trainings. Außerdem müssen die Kunden nicht mehr auf ausgedruckte Trainingspläne zurückgreifen und können stattdessen auf eine sichere und zuverlässige digitale Lösung setzen.

1.5 Funktionale Anforderungen

Die Benutzeroberfläche der native App soll eine einfache und ansprechende Gestaltung aufweisen, um auch nicht computeraffinen Personen eine leichte Handhabung zu ermöglichen. Zusätzlich soll durch die Modernisierung des Trainingsplanprozesses die Kommunikation zwischen Verwaltung und Koordinatoren im Fitnessstudio LionFit vereinfacht und automatisiert werden.

1.5.1 An die App

- Durcharbeitung der Trainingspläne als To-Do-Liste.
- Überprüfung und Anpassung des Trainingsplans nach Abschluss jeder Übung.
- Verfügbarkeit des überarbeiteten Trainingsplans zu jeder Zeit für eine erneute Durcharbeitung.
- Aufbewahrung des alten Trainingsplans in der Historie für zukünftige Referenzen.

1.6 App

1.6.1 Allgemeine Beschreibung

Mit unserer mobilen Anwendung können Sie Ihre Trainingspläne bequem über Ihr Smartphone durcharbeiten. Die Anwendung präsentiert Ihnen Ihre Trainingspläne in Form einer To-Do-Liste, damit Sie jede Übung sorgfältig durchführen können. Nach Abschluss jeder Übung können Sie Ihren Trainingsplan überprüfen und anpassen, um sicherzustellen, dass Sie optimal vorankommen. Der überarbeitete Trainingsplan steht Ihnen jederzeit zur Verfügung, damit Sie ihn erneut durcharbeiten können. Darüber hinaus wird Ihr alter Trainingsplan in der Historie gespeichert, damit Sie ihn später als Referenz nutzen können.

2 Umfeldanalyse

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Citing [1] properly.

Was ist eine GUID? Eine GUID kollidiert nicht gerne.

Kabellose Technologien sind in abgelegenen Gebieten wichtig [2].

3 Planung

Die erste und auch längste Phase dieses Projekts war die Planung. In der Planungsphase wurden verschiedenste Kreativitätstechniken angewandt und Dokumente, die für die Planung unerlässlich sind, erstellt.

3.1 Projektpartnermeetings

Der Grundstein für diese Arbeit waren die abgehaltenen Meetings mit dem Projektpartner dieses Projekts. Bei den ersten Meetings wurde definiert, was die App können muss und wie die App auszusehen hat um Missverständnisse im voraus zu behandeln und um sich ein Bild darüber zu machen, welche Anforderungen das Projekt hat und welche Herausforderungen zu bewältigen sind. Mit diesen Meetings wurde der Projektpartner auch auf aktuellen Stand gehalten und auftretende Probleme und Anmerkungen, die bei der Arbeit am Projekt entstanden sind, wurden besprochen und abgehandelt.

3.2 Use-Case Diagramm

Ein Use Case Diagramm oder auch Anwendungsfalldiagramm ist ein Diagramm von Anwendungsfällen und ihren Beziehungen zu ihrer Umgebung und zu anderen Anwendungsfällen. Damit beschreibt es, welche Funktionen und Dienste ein System für einen Anwender bereitstellt. Mit einem Use Case Diagramm werden weder die Abläufe des Systems beschrieben, noch die Reihenfolge der Funktionen oder Dienste dargestellt. Ein Anwendungsfalldiagramm visualisiert lediglich die Zusammenhänge zwischen einer Menge von Use Cases und den involvierten Akteuren. Damit eignet es sich sehr gut zur Anforderungsanalyse, also zur Ermittlung oder Verfeinerung von Anforderungen.

3.2.1 Elemente im Use-Case Diagramm

Folgende Elemente sind in einem Use-Case Diagramm üblicherweise enthalten:

- **System:** Das System ist an sich kein logisches Modellelement, sondern grenzt den Kontext ab. Das System wird durch eine Box im Diagramm dargestellt, in dem das System seine Funktionen und Dienste zur Verfügung stellt.
- **Akteur:** Der Akteur befindet sich außerhalb des Systems. Er wird als Strichmännchen gezeichnet und kann eine konkrete Person aber auch ein abstraktes Element wie zum Beispiel ein Sensor sein. Wird ein Akteur definiert, muss dieser auch immer mit mindestens einem Use Case in Verbindung stehen.
- **Anwendungsfall:** Ein Anwendungsfall wird meistens als Ellipse visualisiert. Ein Diagramm kann mehrere Anwendungsfälle besitzen. Ist ein Use Case nur durch andere Anwendungsfälle ausführbar, wird er als abstrakt bezeichnet.
- **Beziehungen:** Zwischen Akteuren und Anwendungsfällen oder zwischen Anwendungsfällen selbst bestehen Beziehungen. Sie werden meistens als Linien oder Pfeile zwischen den jeweiligen Objekten eingezeichnet.
- **Notizen:** Mit Notizen lassen sich Informationen hinzufügen, um das Verständnis zu erhöhen. Sie werden mit einem Rechteck dargestellt, dessen obere rechte Ecke eingeknickt ist. Eine gestichelte Linie verbindet die Notiz mit dem zu erklärenden Element.

3.2.2 Verwendung des Use-Case Diagramms in der Diplomarbeit

Das Use Case Diagramm wurde verwendet um Klarheit über die Funktionen und den Akteuren zu schaffen und für Vergewisserung zu sorgen, dass keine wichtigen Verbindungen vergessen wurden. Am Ende der Planungsphase angelangt sieht das Use-Case Diagramm wie folgt aus:

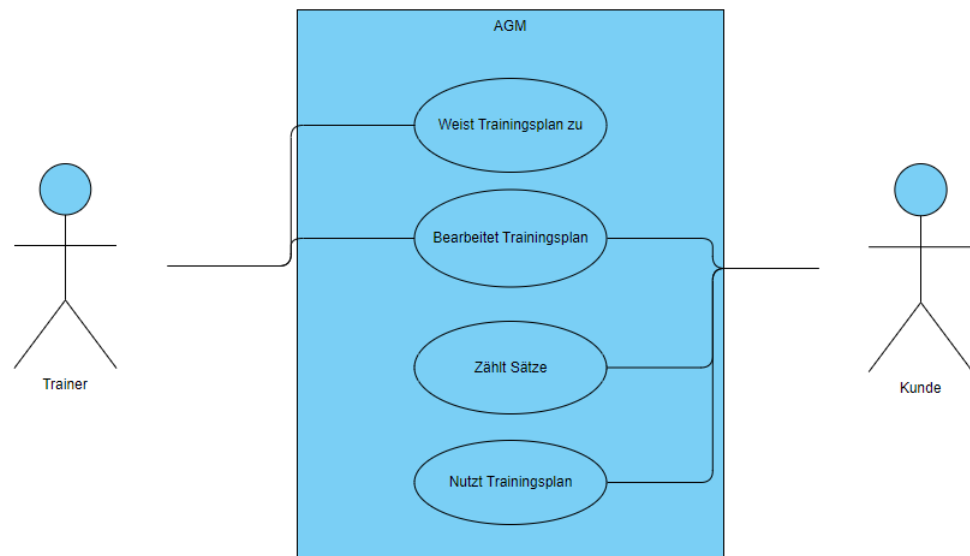


Abbildung 1: Use-Case Diagramm

3.3 Entity Relationship Diagram - ERD

Ein ERD ist ein Diagramm der Beziehungen (Relationship) zwischen Entitäten (Entity) in einem Datenbankschema darstellt. Es wird verwendet, um das Design einer Datenbank zu modellieren und darzustellen und wie Daten in der Datenbank organisiert und miteinander verbunden sind.

3.3.1 Komponenten eines ERDs

Üblicherweise bestehen ERD-Diagramme aus drei Hauptkomponenten, und zwar aus Entitäten, Attributen und Beziehungen.

- **Entitäten:** Eine Entität repräsentiert eine Klasse von Objekten, die bestimmte Eigenschaften oder Attribute gemeinsam haben. Ein Beispiel für eine Entität wäre zum Beispiel ein Kunde der als Attribute einen Namen und eine Wohnadresse hat. Entitäten werden meistens als Rechtecke dargestellt
- **Attribute:** Attribute sind die Eigenschaften, die die Entitäten beschreiben, wie beispielsweise der Name eines Kunden oder die Menge eines Produkts. Attribute werden meistens durch Ellipsen dargestellt können aber auch in der Entität selbst drinnen stehen.
- **Beziehungen:** Beziehungen beschreiben die Art und Weise, wie Entitäten miteinander in Verbindung stehen. Man unterscheidet zwischen drei Arten von Beziehungen. Die erste ist One-to-One. Diese Beziehung tritt auf, wenn jeder Datensatz in der ersten Tabelle nur einen entsprechenden Datensatz in der zweiten Tabelle hat und umgekehrt. Ein Beispiel dazu wäre eine Person und sein Reisepass, denn jede Person hat nur einen Reisepass und ein Reisepass gehört nur zu einer Person. Die zweite Art ist One-to-Many. Diese Beziehung tritt auf, wenn ein Datensatz in der ersten Tabelle mehrere Datensätze in der zweiten Tabelle hat. Ein Beispiel dafür ist eine Mutter die mehrere Kinder hat, aber jedes Kind hat nur eine leibliche Mutter. Die Letzte Art von Beziehung ist Many-to-Many. Diese Beziehung tritt auf, wenn ein Datensatz in der ersten Tabelle mehrere Datensätze in der zweiten Tabelle hat, aber auch umgekehrt. Ein Beispiel für diese Beziehung ist ein Student der mehrere Kurse hat und ein Kurs der mehrere Studenten hat.

3.3.2 Verwendung des Entity Relationship Diagrams in der Diplomarbeit

Um eine funktionierende Datenbank für das Backend zu gewährleisten wurde in Zusammenarbeit mit den Mitgliedern vom Projekt AberGym mithilfe eines ERDs eine Datenbankstruktur erstellt die sowohl für die Anforderungen von AberGym als auch für die Anforderungen von AberGymMobile angepasst.

4 Technologien

4.1 Flutter

Flutter ist ein Open-Source-Framework für die Entwicklung von mobilen Anwendungen, Webanwendungen und Desktop-Anwendungen. Es wurde von Google entwickelt und erstmals im Jahr 2017 veröffentlicht. Flutter verwendet die Programmiersprache Dart und eine eigene Widget-Bibliothek, um eine reaktionsfähige und ansprechende Benutzeroberfläche zu erstellen.

4.1.1 Vorteile

Flutter bietet mehrere Vorteile:

- **Plattformübergreifende Entwicklung:** Mit Flutter können Entwickler plattformübergreifende Anwendungen erstellen, die auf verschiedenen Betriebssystemen wie iOS, Android, Web und Desktop laufen. Dies reduziert die Entwicklungskosten und spart Zeit und Ressourcen.
- **Schnelle Entwicklung:** Flutter bietet die Funktion "Hot Reload", die es Entwickler*innen ermöglicht, Änderungen in Echtzeit zu sehen, ohne die Anwendung neu starten zu müssen. Dadurch wird die Entwicklung von Flutter-Anwendungen schneller und effizienter.
- **Reaktionsfähigkeit:** Flutter-Anwendungen sind schnell und reaktionsfähig, da sie auf der leistungsstarken Grafik-Engine Skia basieren. Dies ermöglicht es Entwickler*innen, reibungslose Benutzererfahrungen mit flüssigen Animationen und Grafiken zu schaffen.
- **Einfache UI-Erstellung:** Mit der eigenen Widget-Bibliothek von Flutter können Entwickler*innen schnell und einfach ansprechende Benutzeroberflächen erstellen. Die Bibliothek enthält viele vorgefertigte Widgets, die einfach angepasst werden können.

- **Native Performance:** Flutter-Anwendungen werden in nativem Code ausgeführt, was zu einer höheren Leistung und Geschwindigkeit führt als bei Hybrid- oder webbasierten Anwendungen.

4.1.2 Dart

Dart ist die von Flutter verwendete Programmiersprache. Sie ist eine objektorientierte Programmiersprache, die von Google entwickelt wurde. Sie wurde erstmals im Jahr 2011 vorgestellt und ist eine relativ neue Programmiersprache im Vergleich zu anderen Sprachen wie Java, Python und C++. Die Sprache wurde entwickelt, um die Herausforderungen bei der Entwicklung von Webanwendungen zu bewältigen und ist auch für die Entwicklung von plattformübergreifenden mobilen Anwendungen geeignet.

Sie ist eine statisch typisierte Sprache, was bedeutet, dass Variablen und Funktionen vor der Laufzeit überprüft werden. Dadurch können Entwickler*innen Fehler frühzeitig erkennen und vermeiden. Die Sprache unterstützt auch dynamische Typisierung, was die Entwicklung von Anwendungen erleichtert, die auf sich ändernden Datenstrukturen basieren.

Dart ist eine schnelle Sprache, die eine effiziente Ausführung ermöglicht und eine hohe Leistung bietet. Es ist auch eine einfach zu erlernende Sprache.

Dart bietet auch eine Vielzahl von Funktionen, die die Entwicklung von Anwendungen erleichtern. Dazu gehören Funktionen wie asynchrone Programmierung, um die Leistung bei der Verarbeitung von Netzwerk- und E/A-Operationen zu verbessern, sowie die Möglichkeit, Funktionen als Parameter zu übergeben, um die Flexibilität und Wiederverwendbarkeit von Code zu erhöhen.

4.2 Visual Studio Code

Visual Studio Code ist ein kostenloses, plattformübergreifendes Code-Editor-Tool von Microsoft. Es ist ein beliebtes Tool für Entwickler*innen, da es eine Vielzahl von Funktionen und Erweiterungen bietet, um die Produktivität und Effizienz zu verbessern.

Visual Studio Code bietet eine intuitive Benutzeroberfläche, die es Entwickler*innen erleichtert, schnell und einfach zu navigieren und Code zu schreiben. Es bietet auch integrierte Debugging-Tools, die Entwickler*innen helfen, Fehler zu finden und zu beheben, sowie integrierte Versionskontrolltools für Git, um Änderungen am Code effektiv zu verwalten.

4.2.1 Extensions

Eine der wesentlichsten Funktionen in VS Code sind Extensions (Erweiterungen). VS Code-Extensions können verschiedene Funktionen bieten, wie z.B. Syntax-Hervorhebung, Autovervollständigung, Debugging-Tools, Git-Integration und vieles mehr. Es gibt eine breite Palette von Erweiterungen, die für verschiedene Programmiersprachen und Frameworks verfügbar sind, um die Entwicklungserfahrung zu verbessern und die Produktivität zu steigern.

Die Installation von VS Code-Extensions ist einfach und unkompliziert. Sie können über den Visual Studio Code-Marktplatz oder direkt aus der Editor-Benutzeroberfläche installiert werden. Nach der Installation stehen die neuen Funktionen und Tools sofort zur Verfügung.

4.3 Docker

Docker ist eine Open-Source-Plattform, die es Entwicklern ermöglicht, Anwendungen in isolierten Containern zu erstellen, bereitzustellen und auszuführen. Docker-Container sind leichtgewichtig und portabel und bieten eine effektive Möglichkeit, Anwendungen in verschiedenen Umgebungen und Infrastrukturen auszuführen.

4.3.1 Funktionsweise

Docker verwendet Container, um Anwendungen zu isolieren und eine konsistente Umgebung für ihre Ausführung zu schaffen. Container sind ähnlich wie virtuelle Maschinen, jedoch leichter und schneller zu erstellen, da sie den Kernel des Host-Betriebssystems nutzen. Jeder Container enthält alles, was eine Anwendung zum Ausführen benötigt, einschließlich des Codes, der Abhängigkeiten und der Konfiguration.

Die Funktionsweise von Docker basiert auf einem Schichtmodell, das aus drei Komponenten besteht:

1. **Docker Engine:** Dies ist das Kernstück von Docker und besteht aus dem Docker-Daemon und der Docker-CLI (Command Line Interface). Der Docker-Daemon ist ein Hintergrundprozess, der die Verwaltung und Ausführung von Containern übernimmt, während die Docker-CLI als Schnittstelle für den Benutzer dient, um mit dem Docker-Daemon zu interagieren.
2. **Images:** Ein Docker-Image ist eine Vorlage oder Blaupause für die Erstellung von Containern. Es enthält den Code, die Abhängigkeiten und Konfigurationen einer Anwendung sowie alle anderen erforderlichen Komponenten, die zur Ausführung der Anwendung benötigt werden. Docker-Images werden über Docker Files erstellt, die eine Liste von Anweisungen enthalten, um das Image zu konfigurieren und zu erstellen.
3. **Container:** Ein Docker-Container ist eine Instanz eines Docker-Images, die ausgeführt wird. Ein Container kann gestartet, gestoppt und gelöscht werden. Jeder Container ist isoliert und hat seine eigene Dateisystemumgebung, Netzwerkschnittstellen und Ressourcenlimits.

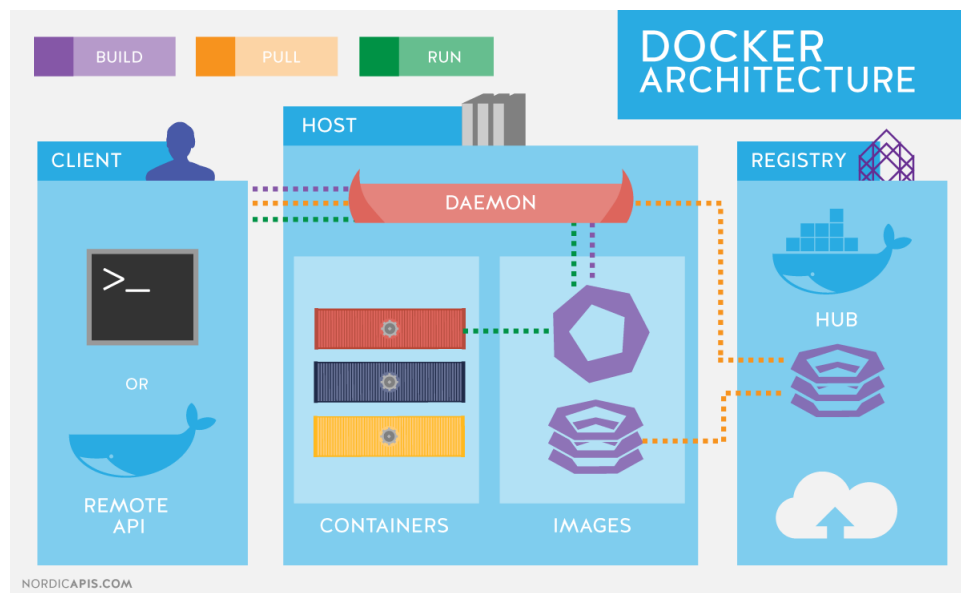


Abbildung 2: Docker Architektur

4.3.2 Vorteile

- **Portabilität:** Docker-Container sind plattformunabhängig und können auf verschiedenen Betriebssystemen und Infrastrukturen ausgeführt werden.
- **Flexibilität:** Mit Docker können Anwendungen schnell erstellt, abgeändert und bereitgestellt werden, ohne die zugrunde liegende Infrastruktur ändern zu müssen.
- **Skalierbarkeit:** Docker ermöglicht eine einfache horizontale Skalierung von Anwendungen, indem es das Erstellen und Bereitstellen von Containern automatisiert.
- **Sicherheit:** Docker bietet Sicherheitsfunktionen wie Isolation und eingeschränkte Ressourcenkontrolle, um eine sicherere Ausführung von Anwendungen zu gewährleisten.
- **Effizienz:** Docker-Container sind leicht und benötigen weniger Ressourcen als virtuelle Maschinen, was zu einer höheren Effizienz und Leistung führt.

4.4 Android Studio

Android Studio ist eine integrierte Entwicklungsumgebung (IDE), die speziell für die Entwicklung von Android-Apps entwickelt wurde. Es wurde von Google entwickelt und ist kostenlos für Entwickler verfügbar, um Android-Apps zu erstellen und zu bearbeiten.

4.4.1 Funktionen

Android Studio verfügt über eine Vielzahl von Funktionen, die dabei helfen, schneller und effizienter Android-Apps zu entwickeln. Zu den wichtigsten Funktionen gehören:

- **Intelligentes Code-Editing:** Android Studio bietet intelligentes Code-Editing mit automatischen Vorschlägen, Fehlererkennung und Refactoring-Funktionen.
- **Emulator:** Entwickler können den Android-Emulator nutzen, um ihre Apps auf verschiedenen Android-Geräten zu testen, ohne physische Geräte besitzen zu müssen.
- **Layout-Editor:** Android Studio verfügt über einen Layout-Editor, mit dem Entwickler die Benutzeroberfläche ihrer Apps visuell gestalten können.
- **Gradle Build-System:** Android Studio verwendet das Gradle Build-System, das Entwicklern ermöglicht, komplexe Abhängigkeiten und Builds zu verwalten.
- **Integration mit anderen Tools:** Android Studio ist nahtlos in andere Google-Tools wie Firebase und Google Cloud Platform integriert.

4.4.2 Emulator

Der Android Studio Emulator ist ein wichtiges Tool für Android-Entwickler, das es ihnen ermöglicht, ihre Apps auf verschiedenen Android-Geräten zu testen, ohne physische Geräte besitzen zu müssen. Der Emulator wird mit Android Studio mitgeliefert und kann einfach über die IDE gestartet werden.

Einer der größten Vorteile des Emulators ist, dass Entwickler ihre Apps auf verschiedenen Android-Versionen und Gerätekonfigurationen testen können, um sicherzustellen, dass ihre Apps auf allen unterstützten Geräten reibungslos funktionieren. Der Emulator kann eine Vielzahl von Android-Versionen und -Gerätekonfigurationen emulieren, einschließlich verschiedener Bildschirmauflösungen und -größen, Prozessortypen und Speicherkapazitäten.

Ein weiterer Vorteil des Emulators ist, dass er den Entwicklungsprozess beschleunigen kann, indem er den Build- und Bereitstellungsprozess verkürzt. Anstatt jedes Mal eine neue Version der App auf einem physischen Gerät zu testen, können Entwickler die App einfach im Emulator starten und testen, was Zeit spart und die Entwicklungszeit verkürzt.

Die Einrichtung des Emulators in Android Studio ist einfach und erfordert nur wenige Schritte. Entwickler müssen zunächst sicherstellen, dass sie die neueste Version von Android Studio heruntergeladen und installiert haben. Sobald sie Android Studio geöffnet haben, können sie den Emulator über das AVD Manager-Tool starten, das im Menü "Werkzeuge" zu finden ist.

Es gibt jedoch auch einige Nachteile beim Verwenden des Emulators. Einer der größten Nachteile ist die Geschwindigkeit. Da der Emulator ein virtuelles Gerät ist, kann er langsamer sein als ein physisches Gerät. Dies kann dazu führen, dass Entwickler länger warten müssen, um ihre Apps im Emulator zu testen.

Ein weiterer Nachteil ist, dass der Emulator nicht alle Funktionen eines physischen Geräts emulieren kann. Beispielsweise kann der Emulator keine Anrufe oder Textnachrichten empfangen und senden, da er kein physisches Mobilfunkmodem hat. Dies bedeutet, dass Entwickler nicht alle Aspekte ihrer App im Emulator testen können und gegebenenfalls auf physische Geräte zurückgreifen müssen.

Zusammenfassend lässt sich sagen, dass der Android Studio Emulator ein wertvolles Tool für Android-Entwickler ist, um ihre Apps auf verschiedenen Geräten und Android-Versionen zu testen. Obwohl der Emulator einige Nachteile hat, überwiegen die Vorteile in den meisten Fällen und er ist ein unverzichtbares Werkzeug für die App-Entwicklung.

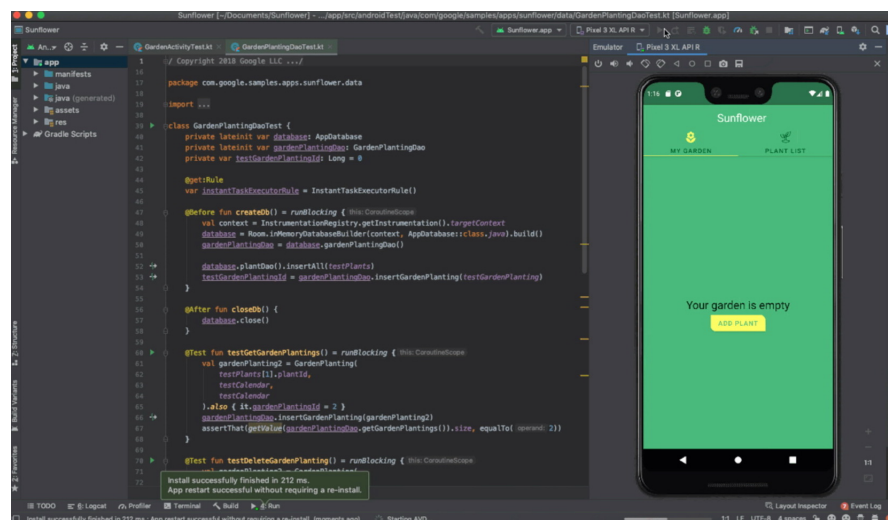


Abbildung 3: Android Studio Emulator

4.5 AdobeXD

AdobeXD ist eine Design-Software, die speziell für die Erstellung von Benutzeroberflächen und Interaktionen für Mobile Apps, Webseiten und andere digitale Plattformen entwickelt wurde.

Funktionen

Die wichtigsten Funktionen sind intuitive Layout-Tools, um Entwürfe schnell zu erstellen und zu bearbeiten, Vektor-Tools, um hochwertige Grafiken zu erstellen, Prototyping-Funktionen, um interaktive Prototypen zu erstellen und zu testen und die Möglichkeit, Designs in Echtzeit zu teilen und Feedback von Stakeholdern zu erhalten.

Plattformübergreifendes Design

AdobeXD unterstützt plattformübergreifendes Design, was bedeutet, dass man nur eine einzige Design-Datei erstellen muss, um diese dann auf AdobeXD auf verschiedenen Geräten und Plattformen verwenden zu können. Das ermöglicht, schnell und effizient Designs für verschiedene Geräte und Plattformen zu erstellen.

Integration mit anderen Tools

AdobeXD ist nahtlos in andere Adobe-Tools wie Photoshop und Illustrator integriert. Das ermöglicht, Designs nahtlos zwischen verschiedenen Adobe-Tools zu übertragen. Es ist auch in andere Tools wie Slack und Microsoft Teams integriert, um die Zusammenarbeit zu erleichtern.

Cloud-basierte Zusammenarbeit

AdobeXD bietet eine Cloud-basierte Zusammenarbeit an, mit der es möglich ist, Designs in Echtzeit zu teilen und Feedback von Stakeholdern zu erhalten. Designer*innen können Links zu ihren Designs freigeben und Stakeholder können Kommentare und Feedback direkt in die Designs geben.

4.6 IntelliJ

IntelliJ ist eine integrierte Entwicklungsumgebung (IDE) für die Softwareentwicklung, die von JetBrains entwickelt wurde. Die IDE ist in Java geschrieben und unterstützt eine Vielzahl von Programmiersprachen wie Java, Kotlin, Groovy, Scala, PHP, Python, und mehr. IntelliJ ist in zwei Versionen erhältlich: die Community Edition, die kostenlos erhältlich ist, und die Ultimate Edition, die kostenpflichtig ist und zusätzliche Funktionen und Tools bietet.

4.6.1 Funktionen

IntelliJ bietet zahlreiche Funktionen und Tools, um Entwicklern bei der Entwicklung von Softwareprojekten zu helfen. Hier sind einige der wichtigsten Funktionen:

- **Code-Editing:** IntelliJ bietet intelligentes Code-Editing mit Code-Vervollständigung, Syntax-Hervorhebung, Refactoring und Code-Analyse-Funktionen.
- **Debugger:** Der integrierte Debugger ermöglicht es Entwicklern, Code zu debuggen und Fehler schnell zu finden.
- **Build-Tools:** IntelliJ unterstützt eine Vielzahl von Build-Tools wie Gradle, Maven und Ant.
- **Integration mit anderen Tools:** IntelliJ ist nahtlos in andere Tools und Frameworks wie Git, JUnit und Spring integriert.
- **Version-Control-System:** IntelliJ unterstützt verschiedene Version-Control-Systeme wie Git, Subversion und Mercurial.
- **Code-Qualität:** IntelliJ bietet Code-Qualität-Tools, um Entwicklern dabei zu helfen, fehlerfreien Code zu schreiben und Code-Qualitätsstandards einzuhalten.
- **Frameworks:** IntelliJ unterstützt eine Vielzahl von Frameworks wie Spring, Hibernate, Struts und mehr.

4.6.2 Vorteile

IntelliJ bietet eine Vielzahl von Vorteilen für Entwickler, darunter:

- **Effizienz:** IntelliJ verbessert die Effizienz der Entwickler durch seine intelligenten Funktionen und Tools.
- **Einfache Integration:** IntelliJ lässt sich einfach in andere Tools und Frameworks integrieren.
- **Bessere Code-Qualität:** Die Code-Qualität-Tools von IntelliJ helfen Entwicklern, fehlerfreien Code zu schreiben und Code-Qualitätsstandards einzuhalten.
- **Unterstützung verschiedener Programmiersprachen:** IntelliJ unterstützt eine Vielzahl von Programmiersprachen und Frameworks.
- **Gute Dokumentation:** Die Dokumentation von IntelliJ ist umfassend und hilft Entwicklern, die IDE schnell zu erlernen.

5 Umsetzung

Siehe tolle Daten in Tab. 3.

Siehe und staune in Abb. 4. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

	Regular Customers	Random Customers
Age	20-40	>60
Education	university	high school

Tabelle 3: Ein paar tabellarische Daten



Abbildung 4: Don Knuth – CS Allfather

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus. Dann betrachte den Code in Listing 1.

Listing 1: Some code

```

1  # Program to find the sum of all numbers stored in a list (the not-Pythonic-way)
2
3  # List of numbers
4  numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
5
6  # variable to store the sum
7  sum = 0
8
9  # iterate over the list
10 for val in numbers:
11     sum = sum+val
12
13 print("The sum is", sum)

```

6 Zusammenfassung

Aufzählungen:

- Itemize Level 1
 - Itemize Level 2
 - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
 - a. Enumerate Level 2
 - i. Enumerate Level 3 (vermeiden)

Desc Level 1

Desc Level 2 (vermeiden)

Desc Level 3 (vermeiden)

Literaturverzeichnis

- [1] P. Rechenberg, G. Pomberger *et al.*, *Informatik Handbuch*, 4. Aufl. München – Wien: Hanser Verlag, 2006.
- [2] Association for Progressive Communications, „Wireless technology is irreplaceable for providing access in remote and scarcely populated regions,” 2006, letzter Zugriff am 23.05.2021. Online verfügbar: <http://www.apc.org/en/news/strategic/world/wireless-technology-irreplaceable-providing-access>

Abbildungsverzeichnis

1	Use-Case Diagramm	7
2	Docker Architektur	14
3	Android Studio Emulator	16
4	Don Knuth – CS Allfather	21

Tabellenverzeichnis

1	<i>Tabelle 1.1: Informationen über den Bertreuer und Partner.</i>	1
2	<i>Tabelle 1.2: Informationen über das Projekt und das Team.</i>	1
3	Ein paar tabellarische Daten	20

Quellcodeverzeichnis

1	Some code	21
---	---------------------	----

Anhang