

PROFIL



Corneliu Mavrodi

Persönliche Daten

78089 Unterkirnach

cornelium@tutanota.com

github.com/lenrocme

lenrocme.github.io/portfolio/

play.google.com/myApps

Soft Skills

Analytisches Denken,
Qualitätsbewusstsein,
Selbstmotivation, Detail orientiert,
Problemlösungsfähigkeiten.

Ausbildung

Fachinformatiker Anwendungsentwickler

Kompetenzschwerpunkte

Entwicklung

Android auf Kotlin & Jetpack Compose & Room;
Implementierung Rest-API service with .NET C#, Entity Framework, Fluent Validation und JSON Web Token;
Frontend: VanillaJS, CSS, HTML;
Desktop-Anwendung VBA & Microsoft Access & MSSQL.

Testen

Testen Rest-API mit Postman und Cypress

Sonstige

Scriptsprachen Python und VBS

Weitere technologische Berührungspunkte

Präsentation erstellen

Erstellung komplexer Powerpoint-Präsentationen

Projekte (Auszug)

Arbeitsplatz & Praktikum bei „k und k information service GmbH“, Fellbach

Testen eines Bereichs der Web Anwendung

Dauer:	2 Monate
Rolle:	Tester (Vollzeitangestellter)
Aufgaben:	Überprüfung der Anforderungen auf logische Widersprüche, Überprüfung der entwickelten Funktionen (entlang der qualitativen Maßstäbe)
Technologien:	MSSQL, SSMS, Git, Azure DevOps, Github, JSON Web Token

Desktop Anwendung (ERP)

Dauer:	4 Monate
Rolle:	Entwickler (Vollzeitangestellter)
Aufgaben:	Dokumentation erstellen. Entwicklung gewünschter Funktionalität & Wartung. Durchführung von Code- Reviews, Lieferung von Aktualisierungen
Technologien:	VBA, VBS, Microsoft Access, PowerShell, MSSQL, Citrix, Git, Azure DevOps, ODBC SQL

Story-time: Nachdem ich das Problem mit der Dekompilierung gelöst hatte, wurde die Dekompilierungszeit von durchschnittlich 7 Minuten deutlich reduziert. Wie hat das funktioniert? Um Zugriff auf die IDE zu erhalten, muss der Entwickler zuerst die Anwendung (Microsoft Access) mit allen erforderlichen Dateien kompilieren. Nach Änderungen im Code ist es erforderlich, die geänderte Datei im Repository hochzuladen. Dies hat durchschnittlich etwa 7 Minuten gedauert, da das vorhandene Skript die gesamte Anwendung (alle Dateien) dekompiert hat. Ich habe das Dekompilierungsskript geändert, sodass es nur die erforderlichen (geänderte) Dateien anstatt aller dekompiert. Dies hat die Arbeitszeit der Entwickler erheblich reduziert. Als Ergebnis wurde Dekompilierungszeit 20 Sekunden statt 7 Minuten.

Rest-API Web Service

Dauer:	3 Monate
Rolle:	Entwickler (Praktikant)
Aufgaben:	Implementierung Rest-API Web Service, dokumentieren und protokollieren umgesetzter Anforderungen
Architektur:	MVC
Technologien:	Microsoft Azurite Blob Storage Emulator, Microsoft SQL Server Express, Entity Framework, Fluent Validation, JSON Web Token, Git, Azure DevOps

Story-time: War Zustand REST-API legacy code, ohne „clean Code“, ohne MVC-Pattern, spaghetti code und sogar ohne funktionierendes Autorisierungssystem: Meine Aufgabe war es, es in eine neue Version von .NET zu überarbeiten, nämlich Version 6, damit es cross-platform funktioniert. Ich habe jedes bestehende Endpoint von Grund auf neu geschrieben, clean Code mit Dokumentation verfasst und das Autorisierungssystem über JWT implementiert. Ein Teil dieser Arbeit wurde auch für mein IHK-Projekt verwendet.

Entwicklung Web Anwendung

Dauer:	2-3 Monate
Rolle:	Entwickler, Designer (Praktikant)
Aufgaben:	Implementierung von Softwarelösungen in Web
Technologien:	Microsoft Azure Blob Storage, MSSQL, Entity Framework, Git, Azure DevOps

Entwicklung einer Android Anwendung

Dauer:	1 Monate
Rolle:	Entwickler (Praktikant)
Aufgaben:	Übertragen der Funktionalität einer iOS-Anwendung auf die Android-Plattform, Entwicklung Android Anwendung
Architektur:	MVC
Technologien:	Android Views, Room Datenbank, Recycle View, Git, Azure DevOps

Eigene publizierte Projekte

Android Anwendung Einkaufen Liste

Dauer:	ca 2 Monate
Beschreibung:	Erstellung der editierbare Einkaufslisten. Mehr lesen...
Programmiersprache:	Kotlin
Architektur:	MVC
Technologien:	Android Views, Shared Preferences, Recycle View, Android Fragments, Git
Downloads:	5
Link:	play.google.com/malferma.quickshoppinglist

Android Anwendung IP Subnetting

Dauer:	ca. 2 Monate
Beschreibung:	IP Subnetting/Calculator. Zufallsgenerator von IP und Netzmaske, durch Benutzereingabe (Antwort über Subnetting) eine Punkteherausforderung mit anderen Benutzern erhalten. Mehr lesen...
Programmiersprache:	Kotlin
Architektur:	MVVM
Technologien:	Jetpack Compose, Room Database, Firebase, ViewModels, Git
Downloads:	3
Link:	play.google.com/malferma.ipsubnetting

Android Anwendung Lohnrechnung

Dauer:	ca. 3 Woche + updates
Beschreibung:	Lohnsteuerrechner 2022/2023. Steuern von Brutto nach Netto zu berechnen. Mehr lesen...
Programmiersprache:	Kotlin
Architektur:	MVVM
Technologien:	Jetpack Compose, Room Database, ViewModels, Git
Downloads:	1
Link:	play.google.com/malferma.brunetto

Android Anwendung Rechner Lebenserwartung

Dauer: ca. 2 Woche
Beschreibung: Eine Informationstafel, durch die Anzeige des Unterschieds zwischen Ihrem aktuellen Alter und der durchschnittlichen Lebenserwartung erinnert diese Tafel an die Kostbarkeit der Zeit und ermutigt Sie, aktiv zu werden und das Beste aus jedem Tag zu machen. [Mehr lesen...](#)
Programmiersprache: Kotlin
Architektur: MVVM
Technologien: Jetpack Compose, Room Database, ViewModels, Git
Name: Memento mori
Downloads: 1
Link: play.google.com/malferma.mementomori

Pet-Projekte

Sudoku Game & Resolver

Datum: 07/2021
Dauer: ca. 2 Woche
Beschreibung: Sudoku game und Sudoku Löser. [Mehr lesen...](#)
Programmiersprache: C#
Technologien: .Net, Windows Forms
Link: <https://youtu.be/LoRkP-z2GDI>
Source: <https://github.com/lenrocme/SudokuSolver-GameApp>

Implementierung von ERP Desktop Anwendung

Datum: 05/2021
Dauer: ca. 2 Monate
Beschreibung: Eine Desktop Anwendung für ein vorstellbar Schrothandler, die Möglichkeit mehrere Konten mit verschiedenen Rolle zu erstellen anbietet & Kunden Daten zu speichern & Quittung PDF zu erstellen, daten der Quittung direkt in Anwendung bearbeiten. [Mehr lesen...](#)
Programmiersprache: Java
Architektur: MVC(Versuch)
Technologien: JavaFx, MariaDb, JDBC, FXML, CSS, JavaFx Scene Builder, iText
Link: <https://youtu.be/aptd1oTyj0s>

Desktop Anwendung on python

Datum: 02/2021
Dauer: ca. 1 Monat
Beschreibung: Kleine Desktop Anwendung für Subnetting IPv4 als Rechner auch als Generator der IP und Netzmaske in Format einer Frage, für Bildungszwecke. [Mehr lesen...](#)
Programmiersprache: Python
Architektur: Keine. Imperative programming.
Technologien: Kivy, KivyMD, PyInstaller, Inno Setup
Link: <https://youtu.be/TuCl4Yz7uml>