



# **Projekt Raspberry Pi: Study Companion**

## **Computer Science (cds-205) HS25**

2025-11-21

**Baltermia Clopath**  
FHGR  
AISE25

## **Inhaltsverzeichnis**

|      |  |   |
|------|--|---|
| 1.   | Zusammenfassung .....                          | 3 |
| 2.   | Einführung .....                               | 3 |
| 2.1. | Themenwahl / Relevanz .....                    | 3 |
| 2.2. | Eingrenzung des Themas .....                   | 3 |
| 2.3. | Fragestellung .....                            | 4 |
| 3.   | Durchführung .....                             | 4 |
| 3.1. | Raspberry Pi Installation .....                | 4 |
| 3.2. | Gewähltes Framework für den Telegram Bot ..... | 4 |
| 3.3. | AI Integration .....                           | 4 |
| 3.4. | Dockerization .....                            | 5 |
| 4.   | Resultate .....                                | 5 |
|      | Bibliografie .....                             | 6 |

## 1. Zusammenfassung

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliquam quaerat voluptatem. Ut enim aequo doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudianda sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedit, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non quo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

## 2. Einführung

### 2.1. Themenwahl / Relevanz

Während einer Präsentation eines alten Schülers vom Modul in unserer Blockwoche kam mir die Idee, etwas zu machen, was mir das ganze Studium über helfen könnte. Letztendlich habe ich mich dann für einen Telegram Bot entschieden. Einerseits, weil ich auf dieser Platform viel Zeit verbringe, anderseits, weil ich damit auch schon einige Erfahrung habe.

Der Bot - den ich *Study Companion* nenne - soll Schülern helfen, den Überblick über Lektionen, Prüfungen, Hausaufgaben und allgemeine Notizen zu behalten. Das macht er, indem er einerseits tägliche Erinnerungen schickt, aber auch auf Anfrage jegliche Informationen so schnell wie möglich bereitstellt. Das wichtigste Feature wird das Einlesen und Verarbeiten des von der FHGR bereitgestellten iCal Stundenplan sein, womit der Bot automatisch weiß, wann welche Lektion stattfindet.

Ich finde das Thema relevant, einerseits, weil ich es andern Schülern so einfach wie möglich machen möchte, den Bot selbst auf einem Raspberry laufen zu lassen. Andererseits werde ich künstliche Intelligenz verwenden, um schlanken Zusammenfassungen zu generieren. Wenn möglich, möchte ich auch einen Helligkeitssensor installieren, welcher erkennt, wann man am morgen aufsteht und somit automatisch die Tägliche Übersicht verschickt wird.

### 2.2. Eingrenzung des Themas

Der Bot soll im Moment wirklich nur das Nötigste können. Mein Ziel ist es, auch in späteren Modulen - wenn sich die Möglichkeit bietet - weitere Features hinzuzufügen. Das wichtigste:

- Einlesen von iCal
- Tägliche Zusammenfassung generiert mit AI
- Einfache Installation auf einem Raspberry Pi

## 2.3. Fragestellung

Die zentrale Fragestellung meines Projekts lautet:

Wie kann man einen Telegram-Bot erstellen, welcher Schülern mit automatischen Erinnerungen und Zusammenfassungen, durch Bereitstellung eines iCal-Kalenders und manueller Eingabe von Terminen den Alltag erleichtert und zudem leichte installation erlaubt?

Dazu habe ich noch folgende Unterfragen formuliert:

1. Welche Frameworks und Bibliotheken eignen für die Entwicklung eines Telegram-Bots
2. Wie kann der Bot den iCal-Kalender effizient einlesen und verarbeiten?
3. Kann ich dadurch eine Vielzahl an AI-Modellen unterstützen, um den Bot flexibel zu gestalten?
4. Wie kann die Installation auf einem Raspberry Pi möglichst einfach gestaltet werden?

## 3. Durchführung

### 3.1. Raspberry Pi Installation

Die Installation des Raspberry Pis werde ich kurz halten. Da der *Study Companion* den Raspberry Pi lediglich als Host verwendet, ist die Installation eines Betriebssystem mit einer Grafischen Benutzeroberfläche nicht nötig. Daher habe ich mich entschieden, Debian zu installieren.

Nach der Installation ist es noch wichtig, Docker zu installieren. Im späteren Abschnitt 3.4 wird dies verwendet.

```
1 apt install docker
```

Shell

### 3.2. Gewähltes Framework für den Telegram Bot

Durch meine mehrjährige Erfahrung mit dem .NET Framework habe ich mich auch für dieses für den Bot entschieden. Telegram stellt bereits eine eigene Bibliothek zur Verfügung [TelegramBots/Telegram.Bot](#). Ich habe mich für eine erweiterte Bibliothek - [k-paul-acct/minimal-telegram-bot](#) - entschieden, da diese viel Boilerplate Code abnimmt und die Entwicklung an die Workflows von .NET anpasst. [1] Im hintergrund verwendet diese Bibliothek aber immer noch die offizielle Telegram Bibliothek.

### 3.3. AI Integration

Der Bot muss keine komplexen Probleme mit AI Lösen, sondern lediglich Zusammenfassungen generieren. Dazu sind heutzutage schon vortrainierte Modelle verfügbar, welche dies ermöglichen. Ich habe mich für die Verwendung der OpenAI API entschieden, da diese einfach zu verwenden ist und gute Resultate liefert. Zudem stellt Microsoft die Pro Version für Studenten der FHGR mittels Github-Copilot Gratis zu Verfügung.

Um es möglichen Nutzern des Bots so einfach wie möglich zu machen, den Bot selbst zu hosten und ihre eigenen API Keys zu verwenden, verwende ich die von Microsoft entwickelte [Microsoft.Extensions.AI](#) Bibliothek. Diese abstrahiert die verschiedenen AI Anbieter und ermöglicht es, den Anbieter einfach zu wechseln, ohne den Code anpassen zu müssen. [2]

### **3.4. Dockerization**

## **4. Resultate**

## Bibliografie

- [1] Pavel Kulakov, *Minimal Telegram Bot - Intentions*. [Online]. Verfügbar unter: <https://github.com/k-paul-acct/minimal-telegram-bot?tab=readme-ov-file#intentions>
- [2] Microsoft, „Microsoft.Extensions.AI libraries“. [Online]. Verfügbar unter: <https://learn.microsoft.com/en-us/dotnet/ai/microsoft-extensions-ai>