

Ostbayerische Technische Hochschule Regensburg

Fakultät für Informatik

# Bachelorarbeit

im Studiengang Informatik

zur Erlangung des akademischen Grades  
Bachelor of Science

**Thema:** Analyse, Konzeption und Implementierung eines Tools für das automatisierte Testen, Einreichen und Benoten von Programmieraufgaben

**Autor:** Andreas Huber <andreas.huber@st.oth-regensburg.de>  
Matr.-Nr. 3180161

**Version vom:** 3. Februar 2022

**Betreuer:** Prof. Dr. Markus Heckner

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation</b>	<b>1</b>
1.1	Herausforderung digitales Lernen . . . . .	1
1.2	Kurs Digital Skills als Zusatzstudium . . . . .	1
1.3	Struktur der Arbeit . . . . .	1
<b>2</b>	<b>Anforderungsanalyse</b>	<b>1</b>
2.1	Funktionale Anforderungen . . . . .	1
2.2	Nichtfunktionale Anforderungen . . . . .	1
<b>3</b>	<b>Softwarearchitektur</b>	<b>2</b>
3.1	Vergleich vorhandener Systeme . . . . .	2
3.1.1	CS50 der Harvard University . . . . .	2
3.1.2	Code FREAK der Fachhochschule Kiel . . . . .	5
3.1.3	GitHub Classroom . . . . .	7
3.2	Finale Architektur Online-Learning Platform . . . . .	9
3.2.1	Tutors als Aufgabensammlung . . . . .	9
3.2.2	GitHub Classroom als Abgabe- und Bewertungssystem . . . . .	9
3.2.3	Replit als Online-Entwicklungsumgebung . . . . .	9
<b>4</b>	<b>Konfiguration und Implementierung</b>	<b>10</b>
4.1	Tutors als Aufgabensammlung . . . . .	10
4.2	GitHub Classroom . . . . .	10
4.2.1	Konfiguration . . . . .	10
4.2.2	Erste Aufgaben . . . . .	10

4.2.3	Tests und Benotung . . . . .	10
4.3	Erstellung eines Replit-Starter-Templates . . . . .	11
4.3.1	Allgemein . . . . .	11
4.3.2	OTH-Console . . . . .	11
4.3.3	SSH-Keys . . . . .	11
4.3.4	Wrapper-Tools (get, check, submit) . . . . .	11
<b>5</b>	<b>Studie: Test an fachfremden Studierenden</b>	<b>11</b>
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>11</b>
	<b>Literaturverzeichnis</b>	<b>iv</b>

# **1 Einleitung und Motivation**

## **1.1 Herausforderung digitales Lernen**

## **1.2 Kurs Digital Skills als Zusatzstudium**

## **1.3 Struktur der Arbeit**

# **2 Anforderungsanalyse**

## **2.1 Funktionale Anforderungen**

## **2.2 Nichtfunktionale Anforderungen**

## 3 Softwarearchitektur

Dieses Kapitel befasst sich mit der für das Projekt benötigten Toolchain. Eine Toolchain ist eine Sammlung verschiedener Anwendungen, die gemeinsam eine Lösung bzw. ein Produkt erzeugen. Durch den Vergleich mit verschiedenen bestehenden digitalen Lernplattformen ist es möglich, optimale Software-Werkzeuge für die Hochschule Regensburg zu finden und schließlich einzusetzen.

### 3.1 Vergleich vorhandener Systeme

#### 3.1.1 CS50 der Harvard University

##### 3.1.1.1 Allgemeines

CS50 ist die ursprüngliche Bezeichnung eines Lernkurses über Informatik, welcher von der Harvard University ins Leben gerufen wurde und weiterhin betreut wird. Der Kurs wurde aufgrund seines Erfolgs digitalisiert und wird nun als CS50x auf der Lernplattform edX angeboten. Folgende Recherchen und Aussagen sind jeweils immer auf die Online-Version CS50x bezogen.

Der Kurs CS50 lehrt Schüler<sup>1</sup> die Grundlagen der Informatik. Dabei werden diverse Programmierübungen abgefragt. Aufgrund der hohen Anzahl an Teilnehmern besitzt der Kurs ein automatisiertes Abgabe- und Benotungssystem.

Das System hinter CS50 wird mittlerweile vielfältig eingesetzt und wurde zu einem universalen Online-Lernsystem erweitert. Jeder kann sich durch eine Authentifizierung über die Plattform GitHub im Abgabesystem von CS50 einloggen und eigene Kurse erstellen. [1]

---

<sup>1</sup>Aus Gründen der besseren Lesbarkeit wird in der gesamten Arbeit auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter. Ist eine spezifische Geschlechtergruppe gemeint, wird das entsprechende Adjektiv vorangestellt (z.B. „männliche Studenten“)

### 3.1.1.2 Ablauf für Studierende

Dem Teilnehmer wird jede Woche ein neues Kapitel präsentiert. Er kann sich dabei sowohl durch ein Vorlesungsvideo, als auch durch geschriebene Materialien über das Thema der Woche informieren. Mit Beginn der Woche bekommt der Teilnehmer neben den Materialien auch Programmieraufgaben, welche er mit dem vorher genannten System bearbeiten kann. [2]

Die Programmieraufgaben können wahlweise über die, auf AWS Cloud9 basierenden, Online-Entwicklungsumgebung „CS50-IDE“ von Harvard oder in jeder anderen beliebigen Entwicklungsumgebung der Wahl bearbeitet werden. Dies wird durch die Architektur des Systems ermöglicht. Jede Funktionalität der Automatisierung geschieht durch Kommandozeilen-Tools. Dieses System hat den Vorteil, dass es unabhängig von der eingesetzten IDE funktioniert, es wird lediglich ein Terminal mit den jeweiligen Tools benötigt. [3]

Vor der Abgabe der endgültigen Lösung mit dem sogenannten Werkzeug „submit50“, ist es möglich den Code mit einem weiteren Werkzeug namens „check50“ überprüfen zu lassen. Außerdem gibt es viele weitere Werkzeuge, ein Beispiel hierfür ist „style50“, welches die Qualität und den Style des Programmcodes überprüft und bewertet. [4]

### 3.1.1.3 Architektur

Die Harvard University hält den Aufbau von CS50 weitestgehend transparent. Viele der eingesetzten Werkzeuge sind öffentlich als Open-Source-Projekte unter der GitHub-Organisation „CS50“ zu finden [5]. Darunter befinden sich unter anderem folgende Projekte:

- submit50: Abgabe von Code
- check50: Funktionalitätstests des Codes
- render50: Erzeugung von .PDF-Dateien aus Code
- ide50: Online-Entwicklungsumgebung
- style50: Überprüfung der Code-Qualität
- compare50: Plagiatserkennung von abgegebenen Projekten
- server50: Webserver

In Abbildung 1 ist der Aufbau der Lernplattform CS50 vereinfacht grafisch dargestellt. Die Rechtecke und Menschen sind die beteiligten Komponenten. Die Pfeile zwischen den Komponenten beschreiben die verbindende Relation. Von der „CS50-Programm“-Komponente ausgehende gestrichelte Pfeile zeigen, je nach ausgeführtem Befehl, mögliche Relationen.

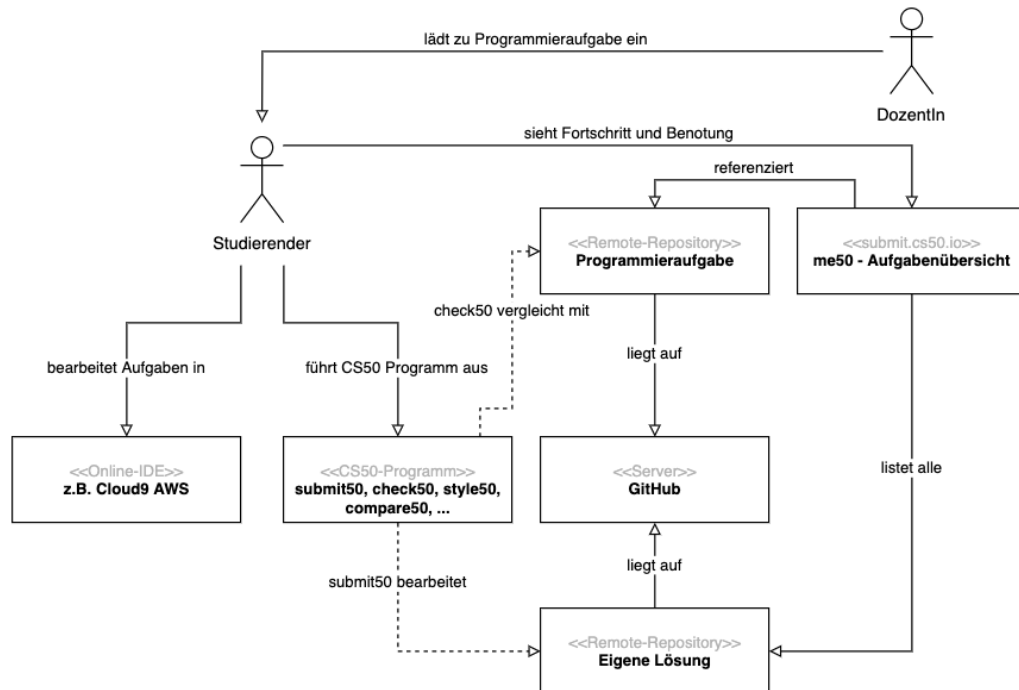


Abbildung 1: CS50 Architektur  
Quelle: Eigene Darstellung

#### 3.1.1.4 Probleme beim Einsatz für die OTH

Die Toolchain von CS50 wäre adäquat für den Einsatz an der OTH-Regensburg. Eines der Projekte ist aktuell noch nicht Open-Source: die Website zur Erstellung von neuen Kursen, Abgaben und Mitgliederverwaltung. Dieses Projekt ist essenziell für die Verwendung der Werkzeuge an der Regensburger Hochschule. Nach Rücksprache mit Herrn Carter Zenke der Harvard University ist ein Neuaufbau dieser Website mit einhergehender Veröffentlichung als Open-Source-Projekt gerade in Planung. Einen genauen öffentlichen Zeitplan hierfür gibt es aktuell nicht. Infolgedessen ist ein Einsatz des CS50-Systems an der OTH zum heutigen Datum nicht möglich.

### **3.1.2 Code FREAK der Fachhochschule Kiel**

#### **3.1.2.1 Allgemeines**

Code FREAK ist eine All-in-one-Lösung für Online-Programmieraufgaben mit automatisiertem Feedback. Die von der Hochschule Kiel entwickelte Open-Source-Software soll den Einstieg in eine digitale Lernumgebung einfach machen. Die Zielgruppe von Code FREAK bezieht sich hierbei explizit auf Universitäten und Einrichtungen einer höheren Bildung. [6]

Des Weiteren wirbt Code FREAK mit einer LMS-Integration. LMS ist die englische Abkürzung für Learning Management System (dt. Lernplattform). Viele Schulen und Universitäten verwenden bekannte LMS-Systeme, wie Moodle, um Kurse, Fächer und Studierende zu verwalten [7]. Dies hat den Vorteil, dass Studierende kein extra Nutzerkonto für Code FREAK anlegen müssen. Sie können sich direkt mit ihrem gewohnten Zugang anmelden.

#### **3.1.2.2 Ablauf für Studierende**

Studierende können nach Anmeldung am System, an für sie sichtbaren Kursen (Assignments) teilnehmen. Dies geschieht meist durch einen vom Dozenten generierten Einladungslink. Jedes Assignment kann mehrere Aufgaben (Tasks) enthalten. Die einzelnen Aufgaben können dann entweder über die integrierte Online-Entwicklungsumgebung, durch Hochladen der Lösung, oder durch die Angabe eines Git-Repository-Links bearbeitet werden.

Mit dem Klick auf den Knopf „Start Evaluation“ wird hochgeladene Lösung der Aufgabe überprüft. Die Ergebnisse der einzelnen Test-Schritte können dann in einem weiteren Tab eingesehen werden. Wenn alle Tests der Aufgabe bestanden wurden, wird der Task mit einem grünen Haken versehen. So sieht der Schüler in der Assignment-Ansicht auf einen Blick, welche Aufgaben er bereits gelöst hat.

#### **3.1.2.3 Architektur**

Code FREAK wird als fertiger Docker-Container ausgeliefert. Docker ist eine Software um mithilfe von Containervirtualisierung einzelne Anwendungen voneinander zu isolieren. Durch die Containervirtualisierung werden unter anderem viele Abhängigkeits- und Einrichtungsprobleme beseitigt. Der Container enthält alle Bibliotheken und Abhängigkeiten, die die Software für die Laufzeit benötigt [8]. Do-



cker entstand auf Basis von Linux-Containern, welche ein oder mehrere Prozesse vom restlichen System isolieren können [9].

Durch diese Handhabung kann Code FREAK mit nur einem Befehl in der Kommandozeile installiert und gestartet werden. Die Software benötigt grundsätzlich nur einen Container. Benutzt ein Studierender jedoch die integrierte Online-Entwicklungsumgebung (Online-IDE), muss für jede Instanz ein zusätzlicher Container mit der Laufzeitumgebung der IDE gestartet werden. Jeder weitere Container benötigt weiteren Arbeitsspeicher.

#### **3.1.2.4 Probleme beim Einsatz für die OTH**

Beim Einsatz von Code FREAK an der Hochschule Regensburg gibt es einige Probleme. Das erste Problem bezieht sich auf den vorher erwähnten Arbeitsspeicher. Die Praxis zeigt, dass eine Instanz der Online-IDE schon nach wenigen Dateien 3 Gigabytes an Arbeitsspeicher verwendet. Um eine reibungslose und parallele Nutzung für alle Studierende des Kurses gewährleisten zu können, werden dementsprechend sehr hohe Serverkosten fällig. [10]

Eine weitere Hürde ist die Stabilität der Software. Code FREAK befindet sich, Stand heute, mitten in der Entwicklung, weshalb einige Funktionen und Features noch nicht ordnungsgemäß funktionieren. Darunter die vorher geworbene LMS-Integration [11]. Zum heutigen Zeitpunkt ist LDAP die einzige Möglichkeit sich mit dem System zu authentifizieren. Ein eigenes Anmeldeverfahren gibt es nicht.

LDAP steht für „Lightweight Directory Access Protocol“ und ist ein Netzwerkprotokoll zur Durchführung von Abfragen und Änderungen in einem verteilten Verzeichnisdienst. LDAP ist der De-facto Industriestandard für Authentifizierung und Autorisierung. [12]

Der Kurs Digital Skills soll den Teilnehmern einen Überblick über den Arbeitssalltag eines Informatikers geben. Dazu gehört unter anderem die Kommandozeile. Eine Anforderung der Lernplattform ist deshalb, dass man (optional) neue Aufgaben herunterladen und Lösungsversuche über Befehle in der Kommandozeile testen und abgeben kann. In Code FREAK kann man Aufgaben lediglich über die Oberfläche testen und abgeben.

### **3.1.3 GitHub Classroom**

#### **3.1.3.1 Allgemeines**

GitHub Classroom ist ein weiterer Kandidat für den Einsatz an der Hochschule Regensburg. Die Plattform ermöglicht die automatisierte Erstellung von Repositories auf GitHub. Außerdem hilft Classroom dabei, Aufgabenvorlagen und dazugehörige Abgaben einfach zu verwalten und automatisch zu benoten. Dabei enthalten die von GitHub Classroom erstellten Aufgaben-Repositories bereits vorkonfigurierte Zugriffskontrollen. [13]

#### **3.1.3.2 Ablauf für Studierende**

Studierende bekommen pro Programmieraufgabe einen Einladungslink. Nach Annahme der Einladung wird pro Studierenden automatisch ein Repository für die jeweilige Aufgabe angelegt. Dieses Repository enthält die Vorlage, welche zum Bearbeiten der Aufgabe benötigt wird.

Sobald der Studierende eine Lösung zur Korrektur abgeben möchte, kann er per Push die Änderungen in das Remote-Repository hochladen. Je nach Konfiguration der Aufgabe starten daraufhin serverseitig ein oder mehrere Tests. Wenn alle Tests bestanden sind, hat der Studierende die Aufgabe erfolgreich abgeschlossen.

#### **3.1.3.3 Architektur**

GitHub Classroom ist ein Bildungsservice der Firma GitHub Inc. und ist Stand heute kostenfrei [14]. Die Software basiert auf der Automatisierung von Repositories. Jeder bei GitHub registrierte Nutzer, kann einen sogenannten „Classroom“ erstellen und darin Aufgaben auf Basis vorhandener öffentlichen GitHub-Repositories erstellen.

#### **3.1.3.4 Probleme beim Einsatz für die OTH**

Es gibt keine Möglichkeit das System von GitHub Classroom auf einen lokalen Git-Server zu replizieren. Aufgrund dessen schafft man sich durch die Verwendung von Classroom eine externe Abhängigkeit an GitHub. Dies kann unter Umständen zu erheblichen Problemen führen, wenn der Dienst beispielsweise nicht erreichbar oder aufgelöst wird. Ersteres ist durch die Größe und Infrastruktur des Unternehmens nicht (häufig) zu erwarten.

Eine weitere Hürde ist der Bedarf an weiterer Software. GitHub Classroom alleine ist nicht ausreichend, um als vollständige Lernplattform für den Kurs Digital Skills geeignet zu sein. Hier bietet es sich an, eine eigene statische Website mit Anleitungen und Erklärungen zu bauen, welche dann jeweils auf Classroom Einladungslinks verweist. Als Online-Entwicklungsumgebung kann der Dienst Replit verwendet werden. In Replit ist es möglich, ein vorhandenes Git-Repository als Template für eine neue Umgebung zu verwenden. Dieses Template könnte Wrapper-Programme für den git-Workflow enthalten. Der Vorteil daran: Studierende bekommen erste Erfahrungen mit der Kommandozeile, müssen jedoch keine komplexen Git-Kommandos absetzen.

## **3.2 Finale Architektur Online-Learning Platform**

### **3.2.1 Tutors als Aufgabensammlung**

### **3.2.2 GitHub Classroom als Abgabe- und Bewertungssystem**

GitHub Classroom eignet sich, für den Einsatz an der Hochschule, vor allem durch seine Flexibilität und Einfachheit. Durch diese Flexibilität ist es möglich, GitHub Classroom nur als eine austauschbare Komponente des Systems zu sehen. Classroom übernimmt im System die Rolle des Aufgabenservers. Hier werden alle Aufgabenvorlagen, sowie alle Versuche der Studierenden gespeichert und bewertet.

### **3.2.3 Replit als Online-Entwicklungsumgebung**

Der Programmierkurs soll für jeden Teilnehmer ohne komplizierte Installationen durchführbar sein. Aus diesem Grund wurde die Entscheidung gefällt, eine Online Entwicklungsumgebung einzusetzen.

Der Vorteil an Replit ist, dass man dem Studierenden durch ein Vorlagenrepository alle nötigen Konfigurationen und Programme im Vorhinein bereitstellen kann. Sobald ein sogenanntes „Repl“ (Bezeichnung für ein Projekt in Replit) mit der erwähnten Vorlage erstellt wurde, kann der Studierende ohne Installation geräteübergreifend online programmieren. Der Teilnehmer muss daraufhin lediglich auf den „Run“-Knopf drücken, welcher die später näher erläuterte „OTH-Console“ startet und schließlich alle benötigten Abhängigkeiten bereitstellt.

## 4 Konfiguration und Implementierung

### 4.1 Tutors als Aufgabensammlung

### 4.2 GitHub Classroom

#### 4.2.1 Konfiguration

Für die Einrichtung wurde eine GitHub Organisation erstellt. GitHub Organisationen können von jedem Nutzer GitHub-Nutzer erstellt werden und benötigen lediglich einen Namen und eine Kontakt-E-Mail-Adresse. Die Organisation dieses Kurses beherbergt alle Aufgabenvorlagen, die später näher erläuterte Vorlage für Replit, sowie alle Aufgabenrepositories der Studierenden.

#### 4.2.2 Erste Aufgaben

Nach der Erstellung der OTH-Organisation mussten die Aufgabenvorlagen erstellt werden. Aufgabenvorlagen sind in GitHub Classroom normale Repositories, welche in GitHub als Vorlage markiert wurden. Sie beinhalten meist zusätzlich Tests, um den Code darin zu prüfen.

Sobald die Organisation mit Aufgaben gefüllt war, konnte der „Classroom“ für den Kurs angelegt werden. Anfangs ist ein Classroom, wie die Organisation auch, leer. Über die Oberfläche können neue Assignments erstellt werden. Assignments sind Aufgaben, welche dem Studierenden zur Verfügung stehen. Für jedes vorher angelegte Aufgabenrepository wurde ein Assignment erstellt. Bei der Erstellung gibt man verschiedene Konfigurationsparameter an. Dazu gehört die Auswahl, ob eine Aufgabe von Einzelpersonen, oder einer Gruppe bearbeitet werden kann, oder ob die jeweiligen Versuche für alle Studierenden oder nur für die Lehrer sichtbar sind. Ferner gibt es die Möglichkeit eine Deadline, sowie den Starter Code anzugeben. Für den Starter Code wurde in diesem Fall jeweils das Aufgabenrepository ausgewählt.

#### 4.2.3 Tests und Benotung

Im nächsten Schritt legt man die Benotung und das Feedback fest. Das Autograding (die Benotung) geschieht über Kommandos in der Konsole. In unserem Fall beinhaltet jedes Aufgabenrepository einen Ordner mit pytest-Tests. Pytest ist eine Code-

Test-Bibliothek für Python. Die Assignments wurden so konfiguriert, dass GitHub nach jedem Push zum Repository des Studierenden die pytest-Tests gestartet werden. Wenn alle Tests erfolgreich sind, erhält der Studierende eine vorkonfigurierte Punktzahl. Durch Classroom ist es außerdem möglich, jedem Test eine individuelle Punktzahl zuzuweisen. So kann man dem Schüler neben der erfolgreichen Ausführung beispielsweise noch Bonuspunkte für das Fangen von nicht geplanten Eingaben vergeben.

## **4.3 Erstellung eines Replit-Starter-Templates**

### **4.3.1 Allgemein**

### **4.3.2 OTH-Console**

### **4.3.3 SSH-Keys**

### **4.3.4 Wrapper-Tools (get, check, submit)**

## **5 Studie: Test an fachfremden Studierenden**

## **6 Zusammenfassung und Ausblick**

# Abbildungsverzeichnis

1	CS50 Architektur . . . . .	4
---	----------------------------	---

# Literaturverzeichnis

- [1] H. University. Cs50: Introduction to computer science. [Online]. Verfügbar: <https://pll.harvard.edu/course/cs50-introduction-computer-science>
- [2] ——. Cs50's introduction to computer science. [Online]. Verfügbar: <https://www.edx.org/course/introduction-computer-science-harvardx-cs50x>
- [3] ——. Online - cs50 docs. [Online]. Verfügbar: <https://cs50.readthedocs.io/ide/online/>
- [4] ——. submit50 - cs50 docs. [Online]. Verfügbar: <https://cs50.readthedocs.io/submit50>
- [5] ——. Cs50 github. [Online]. Verfügbar: <https://github.com/cs50>
- [6] K. U. of Applied Sciences. (2019-2020) Code freak | code feedback, review & evaluation kit. [Online]. Verfügbar: <https://codefreak.org/#about>
- [7] moodle.de. Lernerfolg mit moodle. [Online]. Verfügbar: <https://moodle.de>
- [8] A. Mouat, *Using Docker: Developing and Deploying Software with Containers*. O'Reilly Media, Inc., 2015.
- [9] R. Hat. Was ist ein linux-container? [Online]. Verfügbar: <https://www.redhat.com/de/topics/containers/whats-a-linux-container>
- [10] K. U. of Applied Sciences. Installation guide :: Code freak docs. [Online]. Verfügbar: [https://docs.codefreak.org/codefreak/for-admins/installation.html#\\_dedicated\\_docker\\_host](https://docs.codefreak.org/codefreak/for-admins/installation.html#_dedicated_docker_host)
- [11] ——. Code freak documentation :: Code freak docs. [Online]. Verfügbar: <https://docs.codefreak.org/codefreak/index.html>
- [12] LDAP. Ldap.com - lightweight directory access protocol. [Online]. Verfügbar: <https://ldap.com>
- [13] G. Inc. Github classroom. [Online]. Verfügbar: <https://classroom.github.com>
- [14] A. Jones. (2020, 03) Set up your digital classroom with github classroom. [Online]. Verfügbar: <https://classroom.github.com>