

Ostbayerische Technische Hochschule Regensburg

Fakultät für Informatik

# Bachelorarbeit

im Studiengang Informatik

zur Erlangung des akademischen Grades  
Bachelor of Science

**Thema:** Konzeption und Implementierung eines Systems zur automatisierten Überprüfung und Abgabe von Programmieraufgaben am Beispiel eines Kurses zum Thema Digital Skills

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

**Version vom:** 5. November 2021

**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	Interaktive Programmier-Lernplattform für alle Studierende . . . . .	1
1.3	Struktur der Arbeit . . . . .	1
<b>2</b>	<b>Toolchain</b>	<b>1</b>
2.1	Vergleich vorhandener Systeme . . . . .	1
2.1.1	CS50 der Harvard University . . . . .	1
2.1.2	Code FREAK der Universität Kiel . . . . .	4
2.1.3	nbgrader als Plugin für Jupyter Notebooks . . . . .	4
<b>3</b>	<b>Anforderungsanalyse</b>	<b>4</b>
3.1	Funktionale Anforderungen . . . . .	4
3.2	Nichtfunktionale Anforderungen . . . . .	4
3.3	Durchführbarkeitsanalyse . . . . .	4
<b>4</b>	<b>Softwarearchitektur</b>	<b>4</b>
4.1	Planung der Einzelkomponenten . . . . .	4
4.2	Vergleich von Frameworks und Tools zur Umsetzung . . . . .	4
4.3	Ressourcen- und Kapazitätsplanung . . . . .	4
<b>5</b>	<b>Implementierung</b>	<b>4</b>
<b>6</b>	<b>Prototypische Anwendung und Deployment der Software</b>	<b>4</b>
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>4</b>



# **1 Einleitung und Motivation**

## **1.1 Herausforderung digitales Lernen**

## **1.2 Interaktive Programmier-Lernplattform für alle Studierende**

## **1.3 Struktur der Arbeit**

# **2 Toolchain**

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.

## **2.1 Vergleich vorhandener Systeme**

### **2.1.1 CS50 der Harvard University**

#### **2.1.1.1 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 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.

#### **2.1.1.2 Ablauf**

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.

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.

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.

#### **2.1.1.3 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. 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

// TODO: Bild der Architektur mit Erklärung

#### **2.1.1.4 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 essentiell für die Verwendung der Werkzeuge an der Regensburger Hochschule. Nach Rücksprache mit Herr X 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. In Folge dessen ist ein Einsatz des CS50-Systems an der OTH zum heutigen Datum nicht möglich.

2.1.2 Code FREAK der Universität Kiel

2.1.3 nbgrader als Plugin für Jupyter Notebooks

## 3 Anforderungsanalyse

3.1 Funktionale Anforderungen

3.2 Nichtfunktionale Anforderungen

3.3 Durchführbarkeitsanalyse

## 4 Softwarearchitektur

4.1 Planung der Einzelkomponenten

4.2 Vergleich von Frameworks und Tools zur Umsetzung

4.3 Ressourcen- und Kapazitätsplanung

## 5 Implementierung

6 Prototypische Anwendung und Deployment der Software

7 Zusammenfassung und Ausblick

# Literaturverzeichnis