



UNIVERSITY OF LÜBECK

INSTITUTE FOR SOFTWARE ENGINEERING AND PROGRAMMING LANGUAGES

Softwarearchitektur zur Realisierung der Lehre in der AI-LAB Infrastruktur

Software architecture for Teaching applications on the AI-Lab infrastructure

Bachelorarbeit

verfasst am

Institut für Softwaretechnik und Programmiersprachen

im Rahmen des Studiengangs

Informatik

der Universität zu Lübeck

vorgelegt von

Max Gosau

ausgegeben und betreut von

Prof. Dr. rer. nat. Martin Leucker

mit Unterstützung von

Gunnar Bergmann, Maria Ostanina

Lübeck, den 2. Dezember 2020

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Max Gosau

Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung einer Softwarearchitektur zur Verwendung der "AI-Lab Plattform" für Anwendungsfälle im Bereich der Lehre. Hierbei werden etablierte Methoden aus der Softwareentwicklung verwendet, um eine Anforderungsanalyse aufzustellen, ein Modell zu entwickeln und schließlich einen Prototypen zu implementieren. Verschiedene Software-Systeme wurden analysiert und verglichen, um festzustellen welche Systeme sich für die Integrierung in die bestehende AI-Lab-Infrastruktur eignen. Aufgrund dieser Ergebnisse wurde eine Architektur für die Verwendung der Lernmanagement-Software Moodle und des Jupyter Notebook Dienstes JupyterHub modelliert und anschließend in Form eines Prototypen implementiert.

Abstract

The goal of this thesis is to develop a Software architecture to use the "AI-Lab infrastructure" for use-cases related to teaching applications. To achieve this, established methods for software engineering were used, to perform a requirements analysis, develop a software model and finally implement a prototype. Several different softwaresystems were analyzed and compared, to determine their suitability for integration into the existing AI-Lab infrastructure. Based upon these findings, an architecture was constructed, using the learning management software Moodle and the jupyter notebook service JupyterHub and a prototype was subsequently implemented.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	1
1.2	Stand der Technik	2
1.3	Aufbau dieser Arbeit	2
2	Technische Grundlagen	3
2.1	AI-Lab Hardware	3
2.2	AI-Lab Software	3
3	Anforderungsanalyse	6
3.1	Anwendungsfall: Übungsbetrieb	6
3.2	Anwendungsfall: CTF-Competition	8
3.3	Anwendungsfall: Hackathon	9
3.4	Festlegung der Anforderungen	11
4	Analyse verwandter Software	14
4.1	Lernmanagement-Systeme	14
4.2	Jupyter Notebook Dienste	20
4.3	Software zur Umsetzung von Programmierwettbewerben	25
5	Realisierung der Software	27
5.1	Interaktion der Systemkomponenten	28
5.2	Modellierung der Nutzerinteraktionen	29
5.3	Praktische Implementierung	33
6	Evaluation	36
6.1	Test des Konzepts am Hackathon	36
6.2	Evaluation des Prototyps	36
7	Zusammenfassung und Ausblick	38
7.1	Ausblick	38
	Literatur	40

A	Anwendungsfälle	43
B	Anforderungsdokument	49

1

Einleitung

Die Thematik der künstlichen Intelligenz hat in den letzten Jahren in Forschung und Lehre immer mehr an Popularität und Relevanz gewonnen. Von Spracherkennungssoftware über autonomes Fahren bis hin zur Erkennung von Tumoren in medizinischen Bildern, künstliche Intelligenz wird bereits in vielen Aspekten des modernen Lebens eingesetzt. Um den stetig steigenden Bedarf an künstlicher Intelligenz zu decken, muss neben kompetenten Programmierern auch die für das Trainieren künstlicher Intelligenz benötigte Soft- und Hardware Infrastruktur vorhanden sein. Aus diesem Grund wurde an der Universität zu Lübeck das "KI-Lab Lübeck" Projekt ins Leben gerufen, dessen Ziel es ist, mehreren Zielgruppen, darunter in erster Linie den Forschern und Studierenden der Universität, eine Infrastruktur für die Entwicklung KI-basierter Softwaresysteme zur Verfügung zu stellen.

Neben dem Bedarf an künstlicher Intelligenz steigt auch, insbesondere durch den zum Zeitpunkt des Schreibens andauernden Ausbruch der Covid-19 Pandemie, der Bedarf an Digital Learning Lösungen. Deshalb werden Digital Learning Systeme bereits an vielen Hochschulen verwendet, um den Lehrbetrieb zu unterstützen und einen Großteil des administrativen Overheads zu übernehmen (Joseph Maria u. a., 2019). An der Universität zu Lübeck wurde im Rahmen des "KI-Lab Lübeck" Projektes die AI-Lab Infrastruktur aufgebaut. Dies führt zu der titelgebenden Frage dieser Arbeit: Wie lässt sich eine Softwarearchitektur in die bestehende AI-Lab Infrastruktur einbinden, um die AI-Lab Plattform in der Lehre zu verwenden?

1.1 Zielsetzung

Ziel dieser Arbeit ist der Entwurf einer Softwarearchitektur zur Umsetzung dreier Anwendungsfälle auf dem AI-Lab. Die Anwendungsfälle beziehen sich dabei speziell auf den Einsatz des AI-Labs in der Lehre. Genauer sollen Programmierwettbewerbe veranstaltet und Übungsaufgaben im Bereich der künstlichen Intelligenz bearbeitet werden. Insbesondere soll die Integration der Software in die bereits existierende AI-Lab Infrastruktur ermöglicht werden. Neben der Modellierung von Integrierung und Nutzerinteraktion der Software soll am Ende der Arbeit eine praktische Implementierung in Form eines

Prototypen erfolgen, welcher für einen der Anwendungsfälle ausgearbeitet und getestet werden soll. Der Schwerpunkt liegt hier in der Entwicklung eines Systems, welches die größte Anforderungs-Schnittmenge aller drei Anwendungsfälle umsetzt und die Erweiterung um spezifischere Anforderungen ermöglicht.

1.2 Stand der Technik

Da die Thematik der künstlichen Intelligenz bereits seit Jahren in Forschung und Lehre intensiv behandelt wird, existieren bereits einige Softwareprojekte, die ähnliche Anwendungsfälle aufgreifen und umsetzen. Insbesondere im Bereich der Programmierwettbewerbe gibt es beispielsweise die in 4.3 genauer diskutierte Software Kaggle und eine vom Institut für IT-Sicherheit entwickelte Software zur Durchführung der jährlichen CTF-Competition. Diese Ansätze werden später im Detail besprochen, um dort auf die entsprechenden Anwendungsfälle und Anforderungen genauer eingehen zu können.

1.3 Aufbau dieser Arbeit

Die vorliegende Arbeit ist in sieben Kapitel aufgeteilt. Nach der Einleitung werden in Kapitel 2 die Hard- und Softwaretechnischen Grundlagen des AI-Labs beschrieben. In Kapitel 3 werden die Anforderungen an die Software auf Basis der in 3.1, 3.2 und 3.3 beschriebenen Anwendungsfälle festgelegt und schließlich in 3.4 zusammengefasst. Kapitel 4 beinhaltet die Analyse diverser Softwarelösungen, mit ähnlichen Funktionen zu den in Kapitel 3 aufgestellten Anforderungen. Dabei werden in 4.1 insbesondere Lernmanagement-Systeme und in 4.2 Jupyter Notebook Dienste auf Eignung und Integrierbarkeit in das AI-Lab analysiert. In 4.3 werden Systeme beschrieben, die Programmierwettbewerbe in Bereich der KI umsetzen. In Kapitel 5 wird die Titel-gebende Softwarearchitektur modelliert und schließlich in 5.3 in Form eines Prototyps implementiert. 5.1 beschreibt die Interaktion der einzelnen System-Komponenten, in 5.2 werden die Abläufe und Interaktionen der Software aus Sicht der Nutzer modelliert. In Kapitel 6 wird die Architektur schließlich evaluiert. In 6.1 wird das Konzept anhand des "Maritimen Hackathons" in der Praxis getestet und evaluiert, und in 6.2 wird der implementierte Prototyp evaluiert. Zuletzt werden die Ergebnisse der Arbeit in Kapitel 7 zusammengefasst und mögliche Verbesserungen und Erweiterungen der Software diskutiert. In den Anhängen A und B sind ausführliche Versionen der in der Anforderungsanalyse zusammengefassten Dokumenten zu finden.

2

Technische Grundlagen

Kein Softwareprojekt existiert in einem Vakuum. So gibt es auch für die hier diskutierte Architektur bereits vorhandene Hard- und Softwaretechnische Grundlagen, auf die das System angepasst werden muss. Im Folgenden werden diese Grundlagen aufgeführt, um fundierte Entwicklungsentscheidungen über die Integrierbarkeit in das bereits bestehende System treffen zu können.

Das "KI-Lab Lübeck" ist ein Projekt mehrerer Institute der Universität zu Lübeck. Eines der Ziele des Projekts ist es, eine stabile Infrastruktur für die Entwicklung sicherer KI-Systeme zu etablieren. Diese Infrastruktur besteht aus mehreren Bereichen: Hardware (inkl. Edge Geräte), Software und einer Open Data Plattform. Das Projekt soll die nötige Hard- und Software-Infrastruktur zur Verfügung stellen, um Forschungs- und Lehrprojekte im Bereich der künstlichen Intelligenz und des maschinellen Lernens zu realisieren. Neben der Entwicklung KI-basierter Softwaresysteme, sollen auch Hardwaresysteme in Form von Robotern mit dem System programmiert und trainiert werden können.

2.1 AI-Lab Hardware

Die Hardware des AI-Labs setzt sich aus mehreren Komponenten zusammen. Für das Verarbeiten von Daten stehen zwei DGX Server, eine DGX A100 und eine DGX-2, von NVidia und vier Server von Dell zur Verfügung. Die Dell Server sind mit je vier NVidia V100 GPUs, zwei Intel Xeon Gold 6148 CPUs, 768GB Arbeitsspeicher und 3,2 TB NVME Speicher ausgestattet. Somit stehen dem AI-Lab insgesamt 40 GPUs und 256 Prozessorkerne zur Verfügung. Um die nötige Speicherkapazität für die Projekte zu gewährleisten, stehen ebenfalls mehrere Hundert Terabyte Netzwerkspeicher (NAS) zur Verfügung.

2.2 AI-Lab Software

Neben der Hardware sind in dem Projekt bereits einige Softwarelösungen implementiert. Hier wurde sich primär für Open-Source Lösungen entschieden, welche dann auf

das System angepasst wurden. Die Lösungen werden als Services von einander gekapselt, zur Verfügung gestellt, sind jedoch unter Verwendung eines "Single-Sign-On" miteinander verknüpft. Auf die unterschiedlichen Services kann mittels eines "Dashboards" zugegriffen werden. Zusätzlich zu einer Programmierinterface gibt es zwei verschiedene Datenserver. Dazu gehören ein privater Datenserver, auf dem projektspezifische Daten wie Quellcode gespeichert werden und ein öffentlicher Speicher, der für größere Datensätze wie Trainings- und Testdaten vorgesehen ist.

keycloak und OpenID Connect

Die Authentifizierung und Autorisierung auf dem AI-Lab findet per OpenID Connect, einer Erweiterung des Autorisierungsframework OAuth 2.0, statt (OpenID, 2020; OAuth, 2020). Dadurch wird "Single-Sign-On" ermöglicht, was bedeutet, dass ein Nutzer, anstatt sich bei jeder Plattform einzeln anmelden zu müssen, mit einer einzelnen Anmeldung auf alle Plattformen zugreifen kann. Über OpenID Connect können den Nutzern zusätzlich Rollen und Zugriffsberechtigungen für die jeweiligen Dienste zugewiesen werden. Die Accounts werden in Keycloak angelegt, ein Dienst, der die Authentifizierung über OpenID Connect erlaubt und eine Weboberfläche für Registrierung und Administration zur Verfügung stellt (Keycloak, 2020).

JupyterHub

Die Open-Source Software JupyterHub wird auf dem AI-Lab für das Bereitstellen von JupyterLab Servern verwendet. Diese Server bieten eine webbasierte Entwicklungsumgebung für die Implementierung und Ausführung von KI-basierten Systemen (Jupyter, 2018; Jupyter, 2016c). Zusätzlich dazu wird durch JupyterLab der Zugriff auf den Netzwerkspeicher über ein implementiertes Ordnersystem ermöglicht. JupyterHub und JupyterLab stellen somit einen zentralen Bestandteil der Infrastruktur dar, da ein Großteil der Nutzerinteraktionen über diese beiden Software Elemente stattfindet. Eine detailliertere Analyse von JupyterHub und JupyterLab ist in 4.2 zu finden.

Netzwerkspeicher

Um Projektdaten, wie beispielsweise Quellcode, zu verwalten, wird ein Netzwerkspeicher verwendet. Auf diesem läuft eine Linux-Instanz, dessen Ordnersystem zum Speichern der Projektdaten verwendet wird. Auf der Linux-Instanz läuft außerdem eine Instanz der Synology DiskStation Manager Software (Inc., 2020), die auf das gleiche Dateisystem zugreift, jedoch aufgrund von Problemen bei der Einbindung des Single-Sign-On Dienstes im Allgemeinen nicht von den Nutzern verwendet wird.

DKAN Open data

DKAN ist eine Open-Source Open Data Plattform, die auf dem AI-Lab insbesondere für das Speichern und Organisieren größerer Datensätze verwendet wird. DKAN ermöglicht

neben einer strukturierten Darstellung von Datensätzen auch das Importieren von Datensätzen aus externen Katalogen (DKAN, 2020). Die Datensätze können nach Kategorien und Schlüsselwörtern sortiert werden und bestimmten Arbeitsgruppen zugewiesen werden. Die auf der Plattform veröffentlichten Datensätze sind im Allgemeinen für alle Nutzer sichtbar, es können jedoch auch private Datensätze erstellt werden. Aus diesen Gründen ist die DKAN Open Data Plattform im Kontext des AI-Labs nur für das Speichern und Veröffentlichen von Test- und Trainingsdatensätzen vorgesehen.

3

Anforderungsanalyse

Ein wichtiger Aspekt der Software Entwicklung ist die Aufstellung der Anforderungen, die der Kunde an die Software hat. Im Kontext dieser Arbeit werden drei verschiedene Anwendungsfälle zur Verwendung des AI-Labs für Lehrzwecke betrachtet. Diese Anwendungsfälle wurden von den jeweiligen Instituten eingereicht und in Zusammenarbeit mit Mitarbeitern der Institute genauer ausgearbeitet. Die Anwendungsfälle wurden nach den IEEE Use Case Modeling Guidelines (Firesmith, 1999) erstellt und anschließend auf ein für eine wissenschaftliche Arbeit angemessenes Format angepasst. In Anhang A sind die Anwendungsfälle zusätzlich nach dem Standard verfasst zu finden. Jeder Anwendungsfall beinhaltet eine kurze Zusammenfassung, eine Beschreibung der erwarteten Teilnehmer und eine detaillierte Beschreibung des Ablaufs mit einem dazugehörigen, nach dem UML Standard erstellten Anwendungsfalldiagramm. Anschließend werden die Anforderungen der Anwendungsfälle zusammengefasst.

3.1 Anwendungsfall: Übungsbetrieb

Im Rahmen der Vorlesung "Medical Deep Learning" des Instituts für Medizinische Informatik (IMI) findet eine praktische Übung statt, welche aus zweiwöchentlichen Programmieraufgaben in Pytorch besteht. Die Aufgaben sollen den Studenten inklusive Daten und Codegerüst als Jupyter Notebooks zur Verfügung gestellt werden. Die Studenten sollen diese Aufgaben direkt im Jupyter Notebook bearbeiten und ihre Lösungen am Ende der Bearbeitungszeit einreichen können. Der prinzipielle Ablauf einer Übung ist in Abbildung 3.1 in Form eines Anwendungsfalldiagramms zu finden.

Teilnehmer

An dem Übungsbetrieb nehmen ausschließlich Studenten der Universität zu Lübeck teil. Betreut werden die Übungen von Tutoren des IMI.

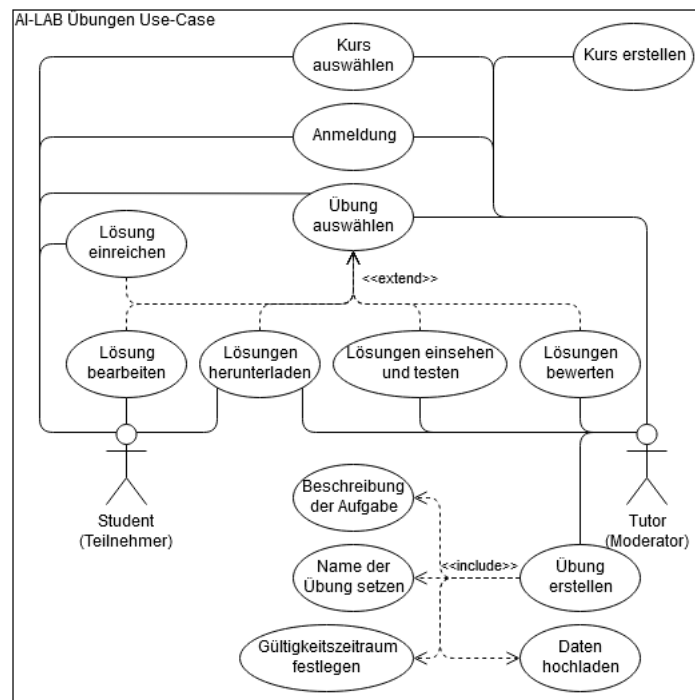


Abbildung 3.1: Modellierung des Ablaufs einer Übung anhand eines UML Anwendungsfalldiagramms.

Ablauf für Tutoren

Der Tutor meldet sich an und kann einen Kurs erstellen oder auswählen. In diesem Kurs kann vom Tutor eine neue Übung erstellt werden bzw. eine existierende Übung ausgewählt werden. Wenn eine neue Übung erstellt wird können von dem Tutor diverse Einstellungen wie Deadline, Name der Übung und Datum der Veröffentlichung festgelegt werden. Zusätzlich kann eine Beschreibung der Aufgabenstellung hinzugefügt werden. und ein initiales Codegerüst sowie ein Satz öffentlicher und gegebenenfalls versteckter Trainings- und Testdaten kann hochgeladen werden. Wurde eine Übung ausgewählt, kann der Tutor die abgegebenen Lösungen der Studenten einsehen, herunterladen, bewerten und testen. Ein zur Bewertung abgegebenes Notebook kann mit Kommentaren versehen werden und schließlich als bewertet markiert werden. Die Bewertung kann dann im Moodle Kurs der Veranstaltung eingetragen werden.

Ablauf für Studenten

Der Student meldet sich an und kann auswählen, welchen Kurs dieser bearbeiten möchte. In dem Kurs wird eine Liste mit zu bearbeitenden Übungen angezeigt, wovon der Student eine auswählen kann. Dabei ist es möglich eine bereits existierende Lösung zu bearbeiten oder eine neue Lösung anzulegen. Außerdem können mehrere Lösungen mit dem initialen Codegerüst angelegt werden, um alternative Ansätze zu testen. Nach Bearbeitung der Aufgabe kann der Student, sofern die Deadline nicht überschritten wurde, eine

der Lösungen zur Abgabe auswählen und einreichen.

3.2 Anwendungsfall: CTF-Competition

Das Institut für IT-Sicherheit (ITS) veranstaltet in regulären Intervallen eine Competition im Bereich der künstlichen Intelligenz. Dazu werden in mehreren Runden KI-Probleme von Teams bearbeitet und die erarbeiteten Lösungen anhand einer Bewertungsmetrik auf Genauigkeit getestet. Die Probleme werden als Jupyter Notebook zusammen mit einem Codegerüst von den Moderatoren zur Verfügung gestellt. Abbildung 3.2 modelliert den Ablauf einer solchen Competition anhand eines Anwendungsfalldiagramms.

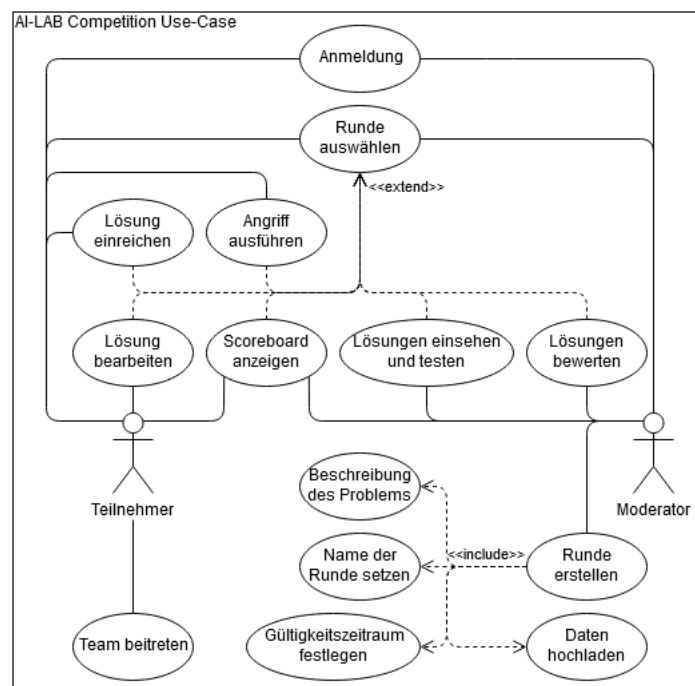


Abbildung 3.2: Modellierung des Ablaufs einer Competition anhand eines UML Anwendungsfalldiagramms.

Teilnehmer

An einer Competition nehmen vorwiegend Studenten der Universität zu Lübeck teil, welche von Mitarbeitern des ITS als Moderatoren betreut werden. Es ist ebenfalls geplant, Mitarbeiter von Uni-externen Betrieben als Teilnehmer einzubinden.

Ablauf

Vor Beginn der Competition wird von den Moderatoren ein Codegerüst, öffentliche Trainings- und Testdatensätze, versteckte Datensätze zur Überprüfung und Bewertung der

Lösungen sowie eine Bewertungsmetrik in Form eines Python Skripts bereitgestellt. Die Teilnehmer bilden Teams, in denen die Probleme bearbeitet werden und melden sich auf der Plattform an.

Eine Competition ist in mehrere Runden aufgeteilt. In einer Runde wird von den Teams ein für diese Runde spezifisches Problem bearbeitet, das heißt ein Modell wird trainiert, und ein Angriff auf ein Modell einer anderen Gruppe aus der vorherigen Runde ausgeführt. Ziel eines Angriffs ist es, zu prüfen, ob eine bestimmte Teilmenge der Trainingsdaten von dem Modell falsch klassifiziert wird. Ein erfolgreicher Angriff führt zu Punktabzug bei dem Team, dessen Modell angegriffen wurde.

In dem zur Bearbeitung der Aufgaben bereitgestellten Jupyter Notebook wird ein Modell programmiert und trainiert. Auf Anfrage des Teams wird das Modell in einer separaten, vertrauenswürdigen Instanz mithilfe des Python Skripts und dem Bewertungsdatensatz auf Genauigkeit geprüft und basierend auf diesen Ergebnissen mit einer API, beispielsweise der "root-the-box" (Mayman, 2016) Implementierung des ITS, mit Punkten bewertet.

Während der Bearbeitung soll es möglich sein, ein Scoreboard anzuzeigen, in dem Statistiken zu den einzelnen Teams zu sehen sind, beispielsweise der Punktestand des Teams und die Genauigkeit des letzten trainierten Modells. Es soll außerdem möglich sein, den Trainingsfortschritt des Modells live zu verfolgen.

3.3 Anwendungsfall: Hackathon

Ein weiterer Anwendungsfall ist der "Maritime Hackathon", welcher im Rahmen der "Digitalen Woche Kiel" für Teilnehmer aus der IT-Branche veranstaltet wird. Als Hackathon wird eine Programmier-Challenge mit Wettbewerbscharakter bezeichnet, in der mehrere Teams gegeneinander antreten, um die bestmögliche Lösung für ein Programmierproblem zu finden. In Abbildung 3.3 ist der Ablauf eines Hackathons in Form eines Anwendungsfalldiagramms zu finden.

Teilnehmer

An dem Hackathon nehmen vorwiegend Angestellte aus mittelständigen Unternehmen in der IT-Branche teil, welche von dem Organisationsteam des Hackathons betreut werden. Gegebenenfalls werden Studenten der Universität zu Lübeck eingesetzt, um Gruppen aufzufüllen.

Ablauf

Die Moderatoren legen einen Hackathon an und laden ein Codegerüst, eine Beschreibung der Aufgabe sowie Trainings- und Validierungsdatensätze hoch. Die Teilnehmer legen sich ein zeitlich begrenztes Konto an und treten der Hackathon Veranstaltung bei.

3 Anforderungsanalyse

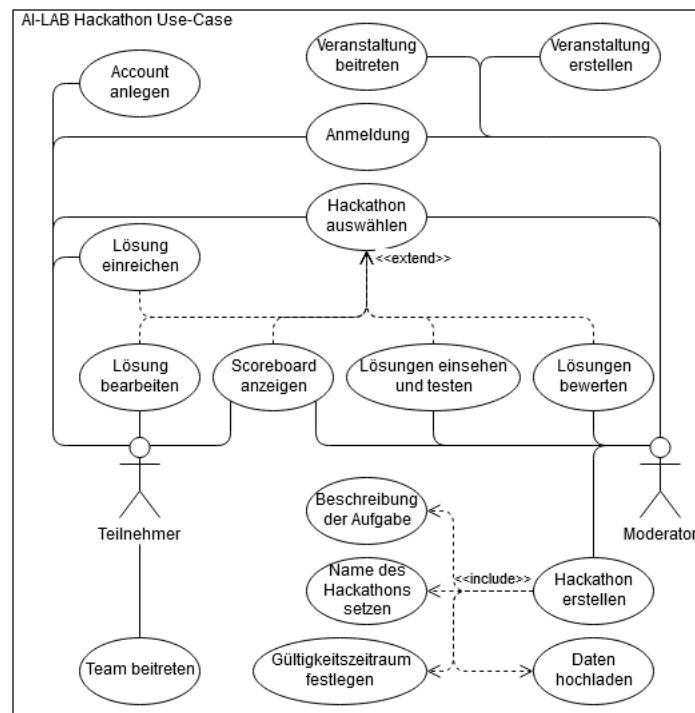


Abbildung 3.3: Modellierung des Ablaufs eines Hackathons anhand eines UML Anwendungsfallendiagramms.

In der Veranstaltung schreiben sich die Teilnehmer in Teams von ca. 2-4 Personen ein, in denen dann ein Bilderkennungsproblem, beispielsweise das Klassifizieren von Schiffen, bearbeitet wird. Dabei können alle Mitglieder eines Teams kooperativ an dem gleichen Dokument arbeiten. Die Moderatoren können die Dokumente aller Teams einsehen und bearbeiten, wohingegen die Teams nur auf ihren eigenen Workspace zugreifen können. Die Teams können je nach Erfahrung das Codegerüst bearbeiten oder eine eigene Lösung programmieren. Jeder Teilnehmer kann den programmierten Code auf den AI-Lab Servern ausführen. Nach Bearbeiten der Aufgabe speichern die Teams ihre Lösung in Form eines Textdokuments in ihrem Workspace und ein Moderator gibt einen Testdatensatz frei. Zum Schluss testen die Moderatoren die eingereichten Lösungen mit einem Evaluationsskript und stellen basierend auf den Ergebnissen einen Gewinner fest.

Für das Bearbeiten der Aufgabe sollen neben einer Python Umgebung die Pakete "numpy", "sklearn", "torch", "torchvision", "torchsummary", "tensorflow" und "tensorboard" zur Verfügung stehen. Die Teilnehmer können außerdem über die Konsole zusätzliche Python Pakete installieren. Jedem Team sollen die gleichen Rechenressourcen auf der Hardware zur Verfügung stehen. Teilnehmer sollen die Möglichkeit haben, auch nach Ende des Hackathons für etwa eine Woche die Ressourcen des AI-Labs nutzen zu können. Nach Ablauf des Gültigkeitszeitraums des Accounts wird dieser automatisch deaktiviert.

Es soll außerdem ein Videokonferenztool eingebunden werden, welches Sprach- und Videochat zwischen den Teilnehmern ermöglicht und eine Funktion zur Bildschirmübertragung zur Verfügung stellt. Neben einem „Hauptraum“ für alle Teilnehmer soll es für jede Gruppe einen Unterraum geben.

3.4 Festlegung der Anforderungen

Aus den oben spezifizierten Anwendungsfällen kann nun ein Anforderungsdokument erstellt werden. Hierbei wurde sich nach den IEEE Standards für die Erstellung von „System Requirement Specifications“ gerichtet (ISO/IEC/IEEE, 2018). Im Anhang B der Arbeit ist das vollständige Anforderungsdokument zu finden, wobei hier einige der wichtigsten Anforderungen noch einmal diskutiert werden.

Ausgehend von den Prinzipien des „Requirements Engineering“ können alle Anforderungen einer Software in funktionale und nicht-funktionale Gruppen aufgeteilt werden (Kotonya, Sommerville, Wiley u. a., 1999). Funktionale Anforderungen beschreiben das Verhalten des Systems, weshalb diese häufig, zusätzlich zu einer Beschreibung der Funktionalität, in Voraussetzungen, Eingaben, Verarbeitungsschritte und Ausgaben aufgeteilt werden. Die funktionalen Anforderungen der Anforderungsanalyse lassen sich grob in Anwendungsfall-spezifische und allgemeine Anforderungen, welche für alle Anwendungsfälle notwendig sind, aufteilen. Interessant sind hier vor allem die allgemeinen Anforderungen, da diese die Grundlage für das zu entwickelnde System festlegen und somit den größten Einfluss auf Entwicklungsentscheidungen haben. Nicht-funktionale Anforderungen beschreiben hingegen Eigenschaften des Systems, welche sich nicht auf das Verhalten, jedoch beispielsweise auf die Qualität oder technischen Voraussetzungen beziehen.

Funktionale Anforderungen

Um mehrere „Instanzen“ der unterschiedlichen Anwendungsfälle gleichzeitig betreiben zu können, bietet sich die Verwendung eines Kurs-Prinzips an. Das bedeutet, dass für jede Lehrveranstaltung (Instanz) ein „Bereich“ angelegt wird, in dem die Dateien und Arbeitsbereiche der Veranstaltung gespeichert werden. Ebenfalls sollen im System und den Kursbereichen bestimmte Nutzer unterschiedliche Rechte haben. Um diese Rechte den jeweiligen Nutzern zuzuordnen, werden den ihnen Rollen zugeteilt. Dabei lässt sich zwischen systemweiten und veranstaltungsinternen Rollen unterscheiden. Die systemweiten Rollen sind:

- *Administrator*: Administratoren können Nutzeraccounts anlegen und löschen, Veranstaltungen erstellen und Nutzern Rollen zuweisen.
- *Veranstaltungsverwalter*: Veranstaltungsverwalter können, wie Administratoren, Veranstaltungen erstellen und in diesen, Nutzern veranstaltungsinterne Rollen zuweisen. Sie haben keine Accountverwaltungsrechte.
- *Nutzer*: Nutzer, denen keine höhere Rolle zugewiesen ist, können Veranstaltungen beitreten.

Die veranstaltungsinternen Rollen sind:

- *Moderator*: Moderatoren können Inhalte einer Veranstaltung verwalten. Sie können notwendiges Material hochladen und auf die Arbeitsbereiche aller Teilnehmer zugreifen. Außerdem können sie Arbeitsgruppen anlegen und Teilnehmern eine Arbeitsgruppe zuweisen.
- *Teilnehmer*: Teilnehmer können auf ihren eigenen Arbeitsbereich und freigegebenes Material zugreifen. Sie können sich außerdem in eine Arbeitsgruppe eintragen und dann auf einen gemeinsamen Gruppen-Arbeitsbereich zugreifen.

In Tabelle 3.4 ist eine Liste der hier diskutierten, allgemeinen funktionalen Anforderungen zu finden.

Tabelle 3.4: Liste der allgemeinen funktionalen Anforderungen an die Software

A1.1	Erstellen und Löschen einer Veranstaltung
A1.2	Suchen und Beitreten einer bestimmten Veranstaltung
A1.3	Aufteilung der Nutzer und Teilnehmer in Rollen mit unterschiedlichen Rechten
A1.4	Festlegen eines Bearbeitungszeitraums für Aufgaben der Veranstaltung durch Moderatoren
A1.5	Erstellen und Löschen von Gruppen durch Moderatoren
A1.6	Selbstständiges Einschreiben der Teilnehmer in Gruppen
A1.7	Bereitstellung der zur Bearbeitung einer Veranstaltung benötigten Daten und Dokumenten in dafür vorgesehenen Workspaces der Teams
A1.8	Zugriff auf öffentliche Trainings- und Testdatensätze
A1.9	Kollaborative Bearbeitung der gleichen Dateien aller Teilnehmer eines Teams und Moderatoren
A1.10	Erstellen und Löschen von Dateien innerhalb des Workspaces durch Teilnehmer
A1.11	Ausführen und trainieren der bearbeiteten Aufgaben
A1.12	Lese- und Schreibrechte der Moderatoren für Workspaces der Teams
A1.13	Ausführen von Evaluationsskripten durch Moderatoren auf den Lösungen der Teams
A1.14	Hoch- und Runter-laden aller Dateien im Workspace
A1.15	Einreichen von Quellcode als Lösung einer Aufgabe
A1.16	Bewerten der eingereichten oder evaluierten Lösungen durch Moderatoren
A1.17	Darstellung einer Rangliste für die Programmierwettbewerbe

Nicht-funktionale Anforderungen

Neben den grundlegenden funktionalen Anforderungen gibt es auch einige nicht-funktionale Anforderungen, die von allen Anwendungsfällen vorausgesetzt werden. So sollen alle Aufgaben mithilfe von Jupyter Notebooks bearbeitet werden (mehr dazu in 4.2).

Außerdem sollen diverse Python Pakete, wie "tensorflow" und "pytorch" zusammen mit einem Python Interpreter zur Verfügung stehen. Es soll Gruppen und Moderatoren außerdem möglich sein, kollaborativ an dem gleichen Dokument zu arbeiten und die Änderungen, die andere Nutzer an einem Dokument vornehmen, in Echtzeit zu verfolgen. Nutzer sollen die Möglichkeit haben, sowohl mit der DGX, als auch mit den Dell Rechnern zu arbeiten, wobei in der CTF-Competition und dem Hackathon die Rechenressourcen jeweils fair unter den Gruppen bzw. Teilnehmern aufgeteilt werden sollen.

Die Anmeldung der Nutzer soll über den gleichen "Single-Sign-On" Dienst laufen, wie auch die anderen Services des AI-Labs. Aufgrund des zum Zeitpunkt des Schreibens andauernden Ausbruchs der Covid-19 Pandemie, findet ein großer Teil der Lehrveranstaltungen an der Universität über das Internet statt. Dadurch ist die Einbindung eines Videokonferenztools, welches ursprünglich mit geringerer Priorität klassifiziert wurde, an Signifikanz gestiegen. In Tabelle 3.5 ist eine Liste mit den allgemeinen nicht-funktionalen Anforderungen an das System zu finden.

Tabelle 3.5: Liste der allgemeinen nicht-funktionalen Anforderungen an die Software

	Sicherheit
A2.1	Anbindung der Software an den Single-Sign-On Dienst des AI-Labs
A2.2	Teilnehmer können nur auf den Workspace der eigenen Gruppe zugreifen
	Technik
A2.3	Die Bearbeitung der Aufgaben erfolgt in Jupyter Notebooks
A2.4	Die Rechenressourcen der AI-Lab Hardware werden fair unter den Teilnehmern aufgeteilt werden
A2.5	Python Pakete wie "numpy" und "tensorflow" sind vorinstalliert bzw. können installiert werden
A2.6	Integrierung einer Sprach- und Videochat Plattform
	Kompatibilität
A2.7	Auf das System kann von allen herkömmlichen Browsern und Betriebssystemen zugegriffen werden
	Ergonomie
A2.8	Die Benutzerführung erfolgt auf Deutsch
A2.9	Workspaces werden automatisch mit den nötigen Dateien angelegt

4

Analyse verwandter Software

Für viele der in Kapitel 3 aufgeführten Anforderungen existieren bereits Systeme, die diese Aufgaben wenigstens teilweise erfüllen. In diesem Kapitel werden eben diese Systeme, insbesondere unter den Aspekten der Anforderungsanalyse und der Einbindbarkeit in die bereits bestehende AI-Lab Infrastruktur, analysiert. Dabei fokussiert sich, die Analyse vor allem auf die für das AI-Lab relevanten Features der jeweiligen Systeme fokussieren. Das System lässt sich basierend auf den gegebenen Anforderungen in drei grundlegende Komponenten aufteilen. Ein Lernmanagement-System, welches das Verwalten von Nutzern, Gruppen und Veranstaltungen übernimmt, ein Jupyter Notebook Dienst, der es den Nutzern ermöglicht Jupyter Notebooks zu bearbeiten und auszuführen und ein Speicher System, auf dem die Arbeitsbereiche der Nutzer bzw. Gruppen verwaltet und gespeichert werden.

4.1 Lernmanagement-Systeme

Lernmanagement-Systeme (kurz: LMS) sind Softwaresysteme, die dazu dienen, Lehr- und Lernprozesse digital zu unterstützen. Insbesondere bieten Lernmanagement-Systeme die Infrastruktur zur Verwaltung von Lernmaterialien sowie zur Sammlung und Übersicht von Lernerfolgen der Nutzer. Lernmanagement-Systeme werden ebenfalls für administrative Zwecke eingesetzt, beispielsweise häufig für das Ein- und Ausschreiben in Lehrveranstaltungen. Ein wichtiger Aspekt von Lernmanagement-Systemen ist die Sammlung und Auswertung der Lernerfolge. Dies bietet den Lehrenden eine gesammelte Übersicht über den individuellen, aber auch den gesammelten Wissensstand der Lernenden, so können Wissenslücken identifiziert und gegebenenfalls die Lehrstrategie angepasst werden (Watson und Watson, 2007). Dies ist für diese Arbeit jedoch von geringerer Relevanz.

Lernmanagement-Systeme weisen häufig ein Rollen-System auf, um Rechte in- und außerhalb der Veranstaltungen zu verwalten. Innerhalb eines LMS können in den meisten Fällen Nutzeraccounts für Lehrende und Lernende angelegt werden. Lernende können zu Veranstaltungen, häufig auch als "Kurse" bezeichnet, hinzugefügt werden und auf deren Inhalte zugreifen. Lehrende können Veranstaltungen erstellen und deren Inhalte verwal-

ten. Aufgrund dieser Funktionalitäten eignen sich Lernmanagement-Systeme, um viele der in 3.4 diskutierten Features, insbesondere in Bezug auf Nutzer- und Rollenverwaltung, umzusetzen. Im folgenden Abschnitt werden zwei gängige LMS, Moodle und ILIAS, deshalb im Bezug auf die Anforderungen genauer beschrieben und analysiert.

Moodle

Moodle ist ein Open-Source Software-Paket, welches an vielen Hochschulen, unter anderem auch der Universität zu Lübeck, für die Verwaltung von Lehrinhalten und Kursen verwendet wird (moodle.org, 2020a). Es bietet standardmäßig eine Reihe an Funktionalitäten, die durch das Installieren von Drittanbieterplugins vielfältig erweitert werden können (moodle.org, 2019a).

Aufbau der Software

Moodle wurde in der Webanwendungssprache PHP entwickelt und basiert standardmäßig auf dem LAMP-Stack. Bei LAMP handelt es sich um eine Server-Architektur mit einem Linux-basierten Betriebssystem, einem Apache Webserver, einer MySQL Datenbank und der Programmiersprache PHP (Luber, 2019). Moodle unterstützt neben MySQL auch PostgreSQL, MSSQL, Oracle und SQLite. Es ist ebenfalls möglich, Moodle auf einem Windows Server mit einem Microsoft IIS Webserver zu installieren (moodle.org, 2019b).

Moodle ist in verschiedene Kategorien, sogenannte "Kontexte", aufgeteilt (moodle.org, 2014). Ein Kontext gibt an, in welchem Bereich sich der Nutzer auf der Moodle Seite befindet. Wie Abbildung 4.1 zu entnehmen ist, lässt sich Moodle in eine Hierarchie aus Kontexten aufteilen. Diese Kontexte können als Komponenten des Systems betrachtet werden und beschreiben den Aufbau von Moodle. Vereinfacht können in einem Moodle System-Kontext Nutzer und Moodle Kurse angelegt werden, wobei es auch möglich ist, Unterkategorien für Kurse zu erstellen. In einem Kurs können entweder Aktivitätsmo-

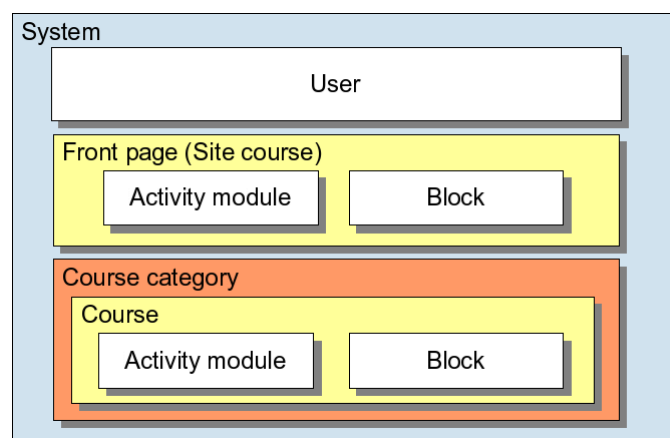


Abbildung 4.1: Exemplarischer Aufbau des Moodle-Systems in Bezug auf den Kontext der jeweiligen Komponenten (moodle.org, 2011)

dule, Blöcke oder Kurskategorien angelegt werden. In Kurskategorien können wiederum Aktivitätsmodule oder Blöcke angelegt werden und sie werden häufig verwendet, um einen Kurs in Themen zu unterteilen. Blöcke sind mehr oder weniger statische Elemente einer Moodle Seite und werden häufig verwendet, um Informationen in der Seitenleiste, je nach Design und Moodle Version auf der linken oder rechten Seite der Website, darzustellen. In einem Block kann beispielsweise ein Kalender mit kommenden Terminen dargestellt werden. Aktivitätsmodule liefern die Hauptfunktionalitäten eines Moodle Kurses und es gibt standardmäßig 14 vorinstallierte Aktivitätsmodul-Plugins (moodle.org, 2020b). So lässt sich beispielsweise über die "Aufgabe"-Aktivität eine Onlineabgabe mit einer Deadline für eine Übung erstellen.

Rollen und Rechte

Ein Kurs kann mehrere Teilnehmer haben, die je nach Zuweisung der Rolle unterschiedliche Rechte haben. Moodle stellt standardmäßig 8 Rollen zur Verfügung (moodle.org, 2013), welche sich wiederum in 5 systemweite und 3 kursweite Rollen aufteilen lassen. Die Systemrollen sind:

- *Administrator*: Der Administrator hat uneingeschränkten Zugriff auf die Moodle Plattform. Er kann die Installation von Plugins verwalten, Kurse erstellen und löschen, neue Nutzer anlegen und löschen, etc.
- *Manager*: Manager haben sehr ähnliche Rechte wie der Administrator, jedoch können diese von einem Administrator genau spezifiziert werden, um höhere Sicherheit bei unerfahreneren Moodle Nutzern zu gewährleisten.
- *Kursersteller*: Kursersteller können in Moodle Kurse anlegen und verwalten, haben aber im Gegensatz zu Managern keine weiteren Rechte.
- *Authentifizierter Nutzer*: Ein authentifizierter Nutzer kann sich in bereits existierende Kurse einschreiben und innerhalb dieser Kurse auf alle Inhalte zugreifen.
- *Gast*: Ein Gast ist ein unangemeldeter Nutzer und kann somit nur bereits erstellte Kurse ansehen. Gäste können sich nicht in Kurse einschreiben und somit auch nicht auf die Inhalte der Kurse zugreifen.

Innerhalb eines Kurses kann Nutzern, unabhängig von deren Systemrolle, von einem Administrator, einem Manager oder dem entsprechenden Kursersteller eine der folgenden drei Kursrollen zugewiesen werden, welche nur innerhalb des jeweiligen Kurs-Kontexts gültig sind.

- *Trainer*: Trainer können innerhalb eines Kurses Inhalte verwalten, Bewertungen vornehmen und Teilnehmer und Gruppen verwalten.
- *Trainer ohne Bearbeitungsrecht*: Diese Rolle kann im Gegensatz zum Trainer keine Kursinhalte bearbeiten oder andere Kurseinstellungen ändern, kann jedoch Bewertungen einsehen und vornehmen.
- *Teilnehmer*: Teilnehmer können auf Kursinhalte zugreifen und an Kursaktivitäten teilnehmen.

Relevante Features

Innerhalb von Kursen können in Moodle Gruppen erstellt werden, welche beispielsweise für gemeinsame Abgaben von Übungsaufgaben verwendet werden können. Die Teilnehmer einer Gruppe müssen standardmäßig manuell von einem Trainer oder einem Nutzer mit mehr Rechten hinzugefügt werden, das Drittanbieterplugin "Group Choice" (Dunand, 2020) bietet Teilnehmern jedoch die Möglichkeit, sich selbst in eine von einem Trainer erstellte Gruppe einzuschreiben. Mit dem Drittanbieterplugin "BigBlueButtonBN" kann die Open-Source, Video- und Sprachchat Plattform "BigBlueButton" in einen Kurs eingebunden werden. Mit diesem Plugin kann auf einem BigBlueButton Server ein kursweiter Raum und gruppenspezifische Räume erstellt werden. Trainer können allen Räumen beitreten, wohingegen Teilnehmer auf den kursweiten Raum und den der jeweiligen Gruppe beschränkt sind (Fred Dixon, 2020).

Moodle unterstützt außerdem das Einrichten und Verwenden von "Single-Sign-On" (SSO) Diensten wie OAuth 2 (moodle.org, 2018), welche es den Nutzern erlauben, auf unterschiedlichen Plattformen mit dem gleichen Account angemeldet zu sein.

Evaluation und Fazit

Insgesamt bietet Moodle viele Funktionalitäten, die von den Anwendungsfällen in Kapitel 3 und den Anforderungen in Tabellen 3.4 und 3.5 gefordert werden. So sind die Anforderungen für das Erstellen und Verwalten von Nutzern, Kursen, Gruppen, Abgaben für Übungen und das Bewerten eingereichter Übungen bereits gedeckt (A1.1 bis A1.5, A1.15 und A1.16). Das Verwalten von Nutzerrechten über ein Rollen-System (A1.3) ist ebenfalls vergleichbar mit den in 3.4 aufgeführten Anforderungen. Die Verwendung von einem "Single-Sign-on"-Dienst (A2.1) vereinfacht zusätzlich die Integration in das bestehende System, da alle Dienste des "KI-Lab Lübeck" Projektes über einen SSO-Dienst laufen sollen. Moodle ist Open-Source und unterstützt dadurch die Einbindung selbst entwickelter Plugins, was die Implementierung der fehlenden Funktionalitäten über ein solches Plugin vereinfacht. Da die Universität zu Lübeck seit mehreren Jahren Moodle als Zentrale Verwaltungsplattform für Vorlesungen verwendet, sind die meisten erwarteten Nutzer bereits mit der Bedienung vertraut und es wäre denkbar, ein "AI-Lab" Moodle Plugin in das Moodle der Universität zu integrieren. Einige Anforderungen, wie das eigenständige Einschreiben in Gruppen (A1.6) und die Verwendung eines Videokonferenztools (A2.6) können ebenfalls über die oben erwähnten Drittanbieterplugins realisiert werden. Moodle verfügt allerdings auch über eine große Menge an Features, die nicht in den Anforderungen vorgesehen sind. So kann das Moodle den Nutzern, vor allem bei Anwendungsfällen wie dem Hackathon, als zu kompliziert oder überflüssig vorkommen, da der Hackathon beispielsweise nur die Gruppenfunktion und Nutzerverwaltung von Moodle benötigt.

Aus den aufgezählten Gründen bietet das Moodle System ein fundiertes und flexibles Gerüst, welches sich mit anderen Systemen kombinieren lässt und viele der in der Anforderungsanalyse aufgeführten Features bereits implementiert. Somit ist es für die Verwen-

dung auf dem AI-Lab geeignet.

ILIAS

Ein anderes, weit verbreitetes Lernmanagement-System ist die Software ILIAS, kurz für "Integriertes Lern-, Informations- und Arbeitskooperations-System". ILIAS ist Open-Source und bietet standardmäßig Funktionen für Kurs-Management, Lernmodule, Tests und Prüfungen, Portfolios, Umfragen, Wikis und Blogs. Diese Funktionen können zusätzlich durch die Installation von Drittanbieterplugins erweitert werden (ilias.de, 2020a).

Aufbau der Software

Die Verwendung von Kursen, Gruppen, Kurskategorien (in ILIAS, Sitzungen genannt) und Aktivitätsmodulen (hier nur Modulen) ist stark mit der von Moodle vergleichbar und wird deshalb nicht näher diskutiert. Eine Installation von ILIAS setzt sich aus verschiedenen Komponenten, sogenannten Objekten, zusammen. Alle Objekte werden in ILIAS in dem "Magazin" angelegt, eine hierarchische Struktur, vergleichbar mit Ordnern in Betriebssystemen (siehe Abbildung 4.2). Es gibt vier verschiedene Objektarten (ilias.de, 2020b):

- *Container-Objekte*: Container-Objekte können andere Objekte enthalten und dessen Zugriff kontrollieren. Darunter fallen beispielsweise Kurse und Gruppen, dessen Zugang durch ein Passwort geschützt werden kann und in denen wiederum andere Objekte erstellt werden können.
- *Lernobjekte*: Unter die Lernobjekte fallen alle Module, die sich explizit auf die Lehre und Lernaktivitäten beziehen, beispielsweise Tests, Umfragen und Übungen, aber auch Foren, Chats und Dateien.
- *Pool-Objekte*: Pool-Objekte sind Sammlungen an Daten und Dateien, auf die immer wieder zugegriffen werden kann. Beispielsweise kann ein Fragenpool für Tests oder Umfragen erstellt werden, auf den die entsprechenden Lernobjekte zugreifen können.
- *Plugin-Objekte*: Als Plugin-Objekt wird alles bezeichnet, was sich nicht in die Lernobjekte einordnen lässt, beispielsweise Abstimmungsfunktionen oder Video- und Sprachchat Plugins, wobei ILIAS eine große Bibliothek an Plugins bereitstellt (ilias.de, 2020d).

Rollen und Rechte

Wie in Moodle, gibt es auch in ILIAS verschiedene Rollen und dazugehörige Rechte. Dabei lässt sich hier ebenfalls nach globalen und lokalen Rollen unterscheiden. Die standardmäßig definierten globalen Rollen sind:

- *Anonymous*: Eine spezielle Rolle, die kein Login erfordert und nur geringe Rechte hat. Anonyme Nutzer können im Magazin nur auf speziell freigegebene Objekte zugreifen.

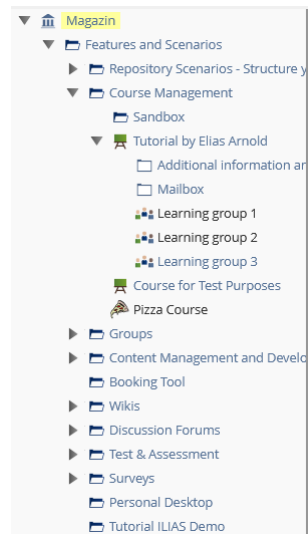


Abbildung 4.2: Beispielhafter Aufbau eines Magazins in ILIAS

- *Gast*: Gäste müssen sich im System Registrieren und können auf das Magazin zugreifen, sie haben jedoch keine Leserechte für Objekte wie Kurse, Wikis oder Gruppen.
- *User*: User sind die häufigsten Nutzer im System. Sie haben volle Leserechte auf die Inhalte im Magazin und können mit einigen Inhalten wie Wikis und Foren interagieren.
- *Administrator*: Administratoren haben volle Zugriffs- und Bearbeitungsrechte auf alle Objekte im Magazin. Sie können ebenfalls Plugins installieren und Nutzer anlegen und verwalten.

Zusätzlich zu diesen Rollen können Administratoren eigens definierte globale Rollen anlegen und genau einstellen, welche Bearbeitungsrechte für welche Objekte diesen zugewiesen werden sollen. Jedes angelegte Objekt hat ebenfalls lokale Rollen. Dabei wird zwischen automatisch generierte und manuell angelegte Rollen unterschieden. Die automatisch generierten Rollen werden beim Erstellen eines Objekts, wie der Name vermuten lässt, automatisch generiert. Diese Rollen hängen von der Art des Objekts ab, so werden für ein Kurs-Objekt beispielsweise die Rollen "Kursadministrator", "Kurstutor" und "Kursteilnehmer" generiert. Der Ersteller eines Objekts kann aber auch neue lokale Rollen erstellen und diese Nutzern zuweisen, sollten die automatisch generierten Rollen nicht ausreichen (Sabelfeld, 2010; ilias.de, 2020e).

Fazit und Vergleich

ILIAS implementiert ähnlich wie Moodle viele der in 3.4 und Tabellen 3.4 und 3.5 aufgeführten Anforderungen, wie das Erstellen und Verwalten von Gruppen, ein Rollensystem, ein Kurssystem und die Abgabe und Bewertung von Übungen (A1.1 bis A1.6, A1.14 und A1.16). Zusätzlich lassen sich Drittanbieterplugins installieren und eigene Plugins entwickeln. Die Einbindung von OpenID Connect bzw. OAuth 2 (A2.1) wird in ILIAS ebenfalls unterstützt (ilias.de, 2020c), wodurch ILIAS und Moodle effektiv die gleichen An-

forderungen erfüllen. Die beiden Systeme unterscheiden sich jedoch in der Umsetzung einiger Anforderungen, so muss zwar in beiden Systemen ein Plugin für die Verwendung eines Sprach- und Videochats installiert werden (A2.6), jedoch können Teilnehmer in ILIAS Gruppen ohne Hilfe eines Plugins selbstständig beitreten (A1.6). Das "Magazin" von ILIAS ist zusätzlich für Nutzer, die keine Erfahrung mit beiden Systemen haben, intuitiver, da es an die Baumstruktur eines Ordnersystems erinnert. Es ist jedoch anzumerken, dass ein Großteil der Nutzer des Systems Studenten und Mitarbeiter der Universität zu Lübeck sind und somit im Rahmen des Lehrbetriebs bereits Erfahrungen mit Moodle sammeln konnten.

4.2 Jupyter Notebook Dienste

Jupyter Notebooks sind seit einigen Jahren die bevorzugte Programmierumgebung im Bereich Data-Science und maschinelles Lernen. Sie wurden ursprünglich entwickelt, um eine Open-Source Alternative zu der Software Mathematica zur Verfügung zu stellen. Aufgrund des von Jupyter verwendeten Python Kernels und der Möglichkeit des Einschubs von Fließtext, Grafiken und Formeln in den Quellcode, eignen sich Jupyter Notebooks insbesondere für die Verwendung in der Lehre. Neben C++ ist Python die einzige Programmiersprache, welche die Verwendung des Tensorflow Frameworks unterstützt. Tensorflow ist ein Software-Framework, das Funktionen für Implementierung und Training künstlicher Intelligenz bereitstellt (Tensorflow, 2020). Wie in 3.4 aufgeführt, sollen für alle Anwendungsfälle Jupyter Notebooks zur Bearbeitung der jeweiligen Aufgaben verwendet werden. Im folgenden Abschnitt werden daher zwei Softwareprodukte, JupyterHub und CoCalk, analysiert und verglichen, welche die Bearbeitung von Jupyter Notebooks ermöglichen.

Jupyterhub

JupyterHub ist eine Open-Source Webplattform, die es Nutzern ermöglicht, eine vorkonfigurierte Programmierumgebung in Form eines Jupyter Notebooks in einem Webbrowser zu bearbeiten und auszuführen. JupyterHub wird bereits an einigen Hochschulen eingesetzt, da es sich durch die Integrierung von Jupyter Notebooks besonders für die Lehre und die Arbeit im Bereich Data-Science eignet (Randles u. a., 2017).

Aufbau der Software

Die JupyterHub Software kann in drei prinzipielle Komponenten unterteilt werden. Eine Datenbank für Nutzerdaten, einen Authentifizierungsservice und einen sogenannten "Spawner" für das Erstellen von JupyterLab und Jupyter Notebook Server Instanzen, sogenannten "Single-User Servern" (Jupyter, 2016c). Dieser grundlegende Aufbau ist in Abbildung 4.3 dargestellt.

In der Nutzerdatenbank werden relevante Daten der Nutzeraccounts gespeichert. Der Authentifizierungsservice authentifiziert die Identität eines Nutzers. JupyterHub bietet

hier verschiedene Optionen für den Service, so kann neben dem standardmäßig installierten PAM-basierten Authentifikator unter anderem auch der Token-basierte OAuth Authentifikator verwendet werden. Dadurch wird weiter die gewünschte Einbindung eines Single-Sign-On Services ermöglicht (Jupyter, 2016a).

Hat sich ein Nutzer angemeldet und ist authentifiziert, kann er über den Spawner einen JupyterLab oder Jupyter Notebook Single-User Server anfordern. In der AI-Lab Infrastruktur greift dieser Server auf ein dafür implementiertes Ordnersystem auf dem Netzwerkspeicher zu. JupyterHub bietet auch hier mehrere Möglichkeiten diese Server zu erzeugen. So lassen sich mit dem "DockerSpawner" (Jupyter, 2016b) Single-User Server in separaten Docker Containern spawnen, auf denen die Nutzer dann arbeiten können. Mit den Docker Containern kann, sofern es gewünscht ist, eine gerechte Ressourcenaufteilung unter den Nutzern gewährleistet werden. Docker Container bieten außerdem einen hohen Grad an Sicherheit, da nicht von außerhalb auf einen Container zugegriffen werden kann. Es ist außerdem möglich, JupyterHub so zu konfigurieren, dass auf jedem gespawnten Server die gleichen Softwarepakete vorinstalliert sind. Über einen Proxy Server werden Nutzer entweder zum Hub oder zu einem JupyterLab Server weitergeleitet.

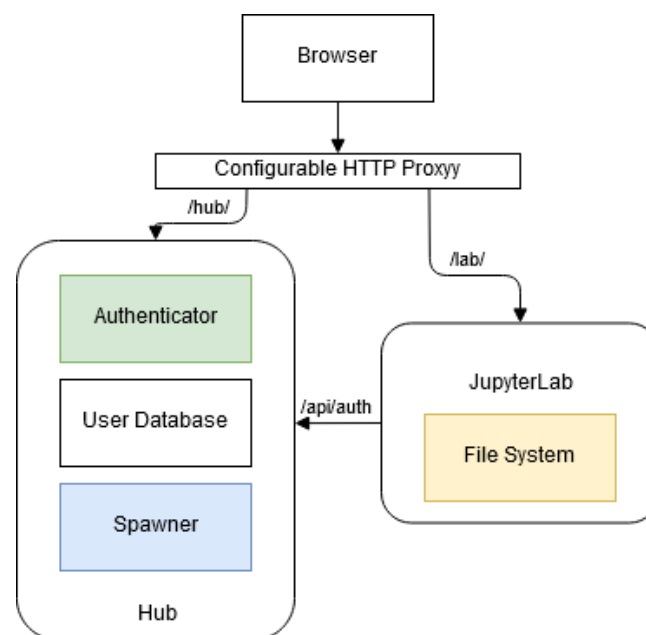


Abbildung 4.3: Aufbau des AI-Lab JupyterHub Servers, basierend auf (Jupyter, 2016c). Links der eigentliche Hub mit Spawner-, Authentifizierungs- und Nutzerdatenbankkomponenten. Rechts ist der vom Spawner erstellte JupyterLab Server, welcher auf dem AI-Lab zusätzlich ein simples Ordnersystem enthält abgebildet. Oben in der Abbildung ein Proxy, der die Weiterleitung verwaltet

JupyterLab

JupyterLab ist eine Open Source Webserver Software, die eine Entwicklungsumgebung in einem Webbrowser zur Verfügung stellt. Wie bei etablierten Entwicklungsumgebungen üblich, bietet JupyterLab verschiedenste Entwicklertools. So kann neben einer Auswahl an Texteditoren eine Konsole, ein Datei-Browser und der Notebook Editor verwendet werden (Jupyter, 2018). JupyterLab unterstützt laut Entwickler über 40 Programmiersprachen, darunter Python, R, Scala und Julia und erlaubt die Erweiterung der Funktionalitäten durch Drittanbieterplugins (Jupyter, 2020). JupyterLab ist eine Weiterentwicklung der Jupyter Notebook Server, wobei weiterhin beide Formate von JupyterHub unterstützt werden.

Eines der Kernfeatures von JupyterLab sind die Jupyter Notebooks. Dabei handelt es sich um Dateien, die eine Mischung aus Code, erklärendem Text, Bildern, mathematischen Gleichungen und interaktiven Visualisierungen enthalten können. Die einzelnen Komponenten des Notebooks werden dynamisch in sogenannte "Cells" (Zellen) geschrieben. Ein beispielhafter Aufbau eines Jupyter Notebooks ist in Abbildung 4.4 dargestellt. Un-

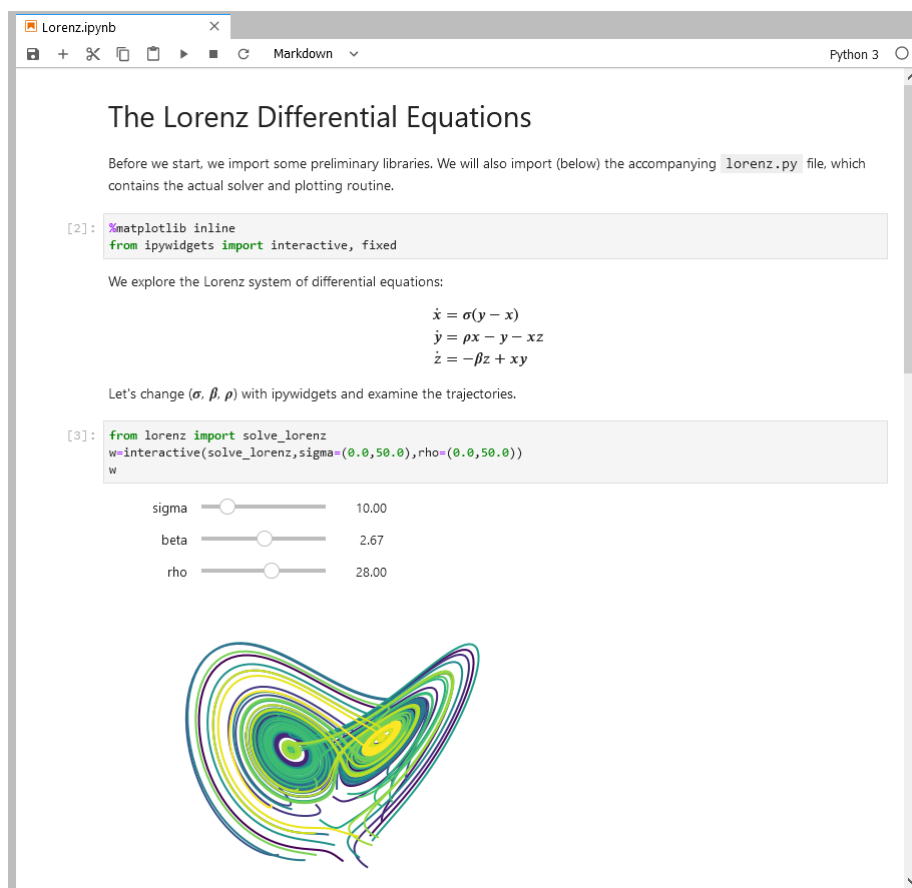


Abbildung 4.4: Beispielhafter Ausschnitt eines Jupyter Notebooks, in dem Differentialgleichungen im Lorenz System visualisiert werden. Die im unteren Teil des Notebooks dargestellte Grafik kann durch die darüberliegenden Slider manipuliert werden.

ter Verwendung der Auszeichnungssprache Markdown können der Code und die Visualisierungen mit Textblöcken aufgearbeitet und erklärt werden. Innerhalb der Textblöcke können Bilder und mathematische Gleichungen mit der Auszeichnungssprache $\text{\LaTeX}$ integriert werden. Der Code eines Notebooks kann entweder als Ganzes oder Zellenweise ausgeführt werden. Durch den Einsatz verschiedenster Pakete können aus dem Output des Codes interaktive Visualisierungen generiert werden. Der Code in den Notebooks wird vorwiegend in Python geschrieben und über die IPython Umgebung ausgeführt, wobei auch Julia und R Kernels verwendet werden können. Die Notebooks werden im JSON Format mit der Dateiendung `.ipynb` gespeichert, was das Konvertieren zu anderen, statischen Dateiformaten wie `html` oder `pdf` ermöglicht (Jupyter, 2015).

Evaluation und Fazit

JupyterHub bietet aufgrund des Spawners, eine skalierbare Grundlage, die es ermöglicht, die Ressourcen eines Rechners oder Rechnersystems unter mehreren Nutzern aufzuteilen. Somit ist JupyterHub für den Einsatz in Systemen mit einer großen Menge an Nutzern geeignet.

Zusätzlich erfüllt JupyterHub einen Großteil der in 3.4 und Tabellen 3.4 und 3.5 aufgeführten Anforderungen. So lassen sich mit JupyterHub JupyterLab Server realisieren, in denen über die Konsole alle gängigen Python Pakete installiert werden können (A2.5). Es ist außerdem möglich, JupyterHub so zu konfigurieren, dass die am häufigsten verwendeten Pakete, wie Tensorflow, Pytorch etc., bereits vorinstalliert sind. Durch das Verwenden von Docker Containern ist ebenfalls eine faire Ressourcenaufteilung und Kapselung der Workspaces unter den einzelnen Nutzern realisierbar (A2.2 und A2.4). OAuth wird von JupyterHub unterstützt und kann somit an ein Single-Sign-On System angebunden werden (A2.1). JupyterHub ist außerdem Open-Source, wodurch keine Lizenzgebühren anfallen und fehlende Features gegebenenfalls implementiert werden können. A1.10, A1.11 und A1.14, sowie A2.3 und A2.7 werden von JupyterHub bzw. JupyterLab gedeckt. Es gibt jedoch keine Unterstützung für kollaboratives Arbeiten (A1.9), was bedeutet, dass Nutzer, die an dem gleichen Dokument arbeiten, Änderungen erst sehen, wenn ein Speichervorgang ausgeführt wird. Änderungen, die nicht gespeichert wurden, werden von einem anderen Speichervorgang überschrieben. Insgesamt eignet sich JupyterHub, mit Ausnahme des nicht-kollaborativen Arbeiten, aufgrund der flexiblen Struktur und der bereits bestehenden Implementierung auf der AI-Lab Infrastruktur für die Verwendung der hier diskutierten Architektur.

CoCalk

Die zweite Jupyter Notebook Software, die hier analysiert wird ist CoCalk, eine virtuelle, online Entwicklungsumgebung zum Erstellen und Verwalten von Workspaces im Bereich von Data-Science und maschinellem Lernen. Sie unterstützt die Entwicklung in Jupyter Notebooks sowie die kollaborative Nutzung der meisten ihrer Funktionen (Sagemath, 2020g).

Aufbau und relevante Features

CoCalk teilt Workspaces in abgetrennte Bereiche, sogenannte "Projekte", auf. Hinter den Kulissen ist ein Projekt nicht mehr als ein Docker Container, auf dem eine Linux Umgebung mit der vorinstallierten CoCalk Software läuft. Alle in CoCalk erstellten Dateien werden dabei im /home Verzeichnis der Linux Umgebung gespeichert (Sagemath, 2020f). Der Ersteller eines Projekts kann Kollaborateure hinzufügen, welche volle Schreib- und Leserechte innerhalb des Projekts haben. Auf der Linux Umgebung sind alle Nutzer unter dem "user" Account angemeldet. Kollaborateure können ebenfalls neue Kollaborateure hinzufügen und andere entfernen, sie können jedoch nicht den Ersteller des Projekts entfernen (Sagemath, 2020a).

CoCalk bietet neben den Jupyter Notebook und JupyterLab Servern von Jupyter, eine von CoCalk selbst entwickelte Implementierung eines Jupyter Notebook Servers. Diese CoCalk Notebook Server erlauben es mehreren Nutzern, gleichzeitig die gleichen Funktionalitäten zu verwenden. So lassen sich $\text{\LaTeX}$ Dokumente, Python und R Dateien, Linux Terminals und Jupyter Notebooks kollaborativ bearbeiten (Sagemath, 2020c; Sagemath, 2020e; Sagemath, 2020d). Diese kollaborative Implementierung unterstützt standardmäßig 14 Kernels, darunter Python 3, Julia, R und den hauseigenen SageMath Kernel. Es ist aber auch möglich, eigene Kernels auf der Linux Umgebung zu installieren und für Jupyter Notebooks zu verwenden. Neben den in 4.2 beschriebenen Funktionen von Jupyter Notebook Servern verfügt die CoCalk Implementierung über einen projekt-internen Textchat, mit dem die Nutzer eines Projekts untereinander kommunizieren können und eine "Time Travel" Funktion, mit der eine detaillierte Bearbeitungshistorie der einzelnen Dokumente angezeigt werden kann (Snyder, 2017).

Fazit und Vergleich

Es gibt zwei Möglichkeiten CoCalk in das bestehende System einzubinden. Da die Software Open-Source ist, kann sowohl die von CoCalk selbst verwendete Server-Software als auch ein Docker Image verwendet werden. Die Server-Software umfasst mehrere hundert Gigabyte und enthält Funktionen, welche für die hier diskutierte Software nicht benötigt werden. Dementsprechend müssten große Modifikationen an dem Quellcode vorgenommen werden, um CoCalk in das bestehende System zu integrieren.

Es steht zusätzlich ein CoCalk Docker Image zur Verfügung, mit dem die Kernfunktionen von CoCalk in einem Docker Container verwendet werden können. Das Image trennt die darin erstellten Projekte nicht voneinander, da in diesem Docker Container wiederum keine neue Docker Instanz für ein neues Projekt erstellt wird. Somit könnten Nutzer unterschiedlicher Projekte, ohne Anpassung der Software, auf alle anderen Projekte zugreifen. Aus diesem Grund ist es außerdem nicht möglich die Ressourcen des Servers unter den einzelnen Projekten aufzuteilen (Sagemath, 2020b). Das Image könnte mithilfe von Docker Swarms oder Kubernetes, ähnlich wie JupyterHub, für Nutzer der Software zur Verfügung gestellt werden, somit könnte beispielsweise für jede Abgabegruppe einer Übung ein CoCalk Docker Image gestartet werden. Diese Funktionen werden jedoch

nicht von CoCalk bereit gestellt und müssten somit eigens entwickelt werden.

Obwohl CoCalk eine der wenigen Softwarelösungen für Jupyter Notebooks ist, die das kollaborative Bearbeiten von Notebooks ermöglicht, mangelt es an Skalierbarkeit auf größere Systeme. Zusätzlich zu einer lückenhaften Dokumentation bietet CoCalk, im Vergleich zu Jupyter, eine weitaus kleinere Plugin-Bibliothek und ist außerdem aufgrund der komplexeren Software weniger flexibel in Bezug auf die Konfiguration der Software. Das Jupyter Ecosystem besteht hingegen aus vielen kleinen Bauteilen, die je nach Anwendungsgebiet zusammengesetzt werden können, was JupyterHub zu einer weitaus flexibleren Lösung macht. Insgesamt stellt die Verwendung von CoCalk einen weitaus größeren Entwicklungsaufwand als die vergleichbare JupyterHub Software dar.

4.3 Software zur Umsetzung von Programmierwettbewerben

Für die Anwendungsfälle "Hackathon" und "CTF-Competition" existieren, wie in 1.2 bereits erwähnt, schon Softwarelösungen, die einige Features der Anwendungsfälle umsetzen. Im Folgenden werden diese kurz beschrieben und für die Architektur relevante Features und Mängel dargestellt.

Kaggle

Kaggle ist eine online Plattform und Community für Datenwissenschaftler und KI-Entwickler. Neben einer großen Ansammlung an für die Verwendung in KI und Data-Science geeigneten Datensätzen werden auf Kaggle Wettbewerbe, sogenannte Competitions, veranstaltet. Ziel einer Competition ist es, ein bestimmtes Programmierproblem häufig im Bereich von KI oder Data-Science zu lösen. Dafür können entweder die benötigten Dateien von Kaggle heruntergeladen werden, um das Problem lokal zu bearbeiten oder das Problem kann auf einem simplen Jupyter Notebook Server über Kaggle gestartet werden und im Browser bearbeitet werden. Wenn eine Lösung erfolgreich programmiert wurde, kann sie je nach Spezifikation der Aufgabe hochgeladen werden, wobei die hochgeladene Datei noch einmal automatisch auf Konformität zu dem Abgabeformat überprüft wird. Die Lösungen können entweder einzeln oder als Team abgegeben werden. Nachdem die Lösungen anhand eines Skripts evaluiert wurden, können die Ergebnisse über ein Leaderboard eingesehen werden. Auf dem Leaderboard werden alle teilnehmenden Teams nach ihrer Punktzahl platziert (Kaggle, 2020).

Kaggle selbst ist nicht Open-Source und kann somit nicht direkt in das diskutierte System eingebunden werden. Zusätzlich ist die von Kaggle verwendete Jupyter Notebook Programmieroberfläche im Vergleich zu dem auf dem AI-Lab verwendeten JupyterLab in Bezug auf die Funktionen eher eingeschränkt, so kann beispielsweise nur mit den Programmiersprachen R und Python gearbeitet werden und der Zugriff auf eine Konsole ist ebenfalls nicht möglich. Allerdings gibt Kaggle einen guten Einblick in das Zusammenspiel mehrerer Systeme für die Entwicklung von KI-Anwendungen, insbesondere für einen Wettbewerb, vergleichbar mit den Hackathon- und Competition-Anwendungsfällen.

Es ist beispielsweise möglich, direkt auf die von Kaggle bereitgestellte Sammlung an öffentlichen Datensätzen zuzugreifen, vergleichbar mit der vom AI-Lab verwendeten DKAN Plattform. Features wie das Leaderboard und die Evaluation der eingereichten Lösungen sind vergleichbar mit den gestellten Anforderungen und es kann sich dementsprechend an dem Ansatz von Kaggle orientiert werden.

CTF-Implementierung des ITS

Am 4. Juli 2020 veranstaltete das Institut für IT-Sicherheit die jährliche CTF-Competition. Anders als in den vorherigen Instanzen wurde diese Competition auf der Hardware des AI-Labs ausgeführt. Zu diesem Zweck wurden mittels eines Docker Spawners Jupyter Notebook Server Instanzen erstellt, auf denen die Teilnehmer die entsprechenden Aufgaben bearbeiten konnten. Für das Training wurden dann NVidia Docker Container erzeugt in denen dann das Training und die Evaluation auf der AI-Lab Hardware ausgeführt wurden. Zur Bewertung der Evaluation wurde eine angepasste Version der Software "root-the-box" verwendet (Mayman, 2016).

Feedback der Teilnehmer ergab, dass eine starke Erhöhung der Bearbeitungszeit von ursprünglich etwa 30 Minuten auf mehrere Stunden sowie eine Aufteilung der Runden auf mehrere Tage gewünscht sei. Neben diesem Feedback gab es auch einige technische Schwierigkeiten, so konnten die GPUs des DGX-2 Clusters nicht voll ausgelastet werden, da scheinbar einige Treiber nicht die richtige Version aufwiesen.

5

Realisierung der Software

Aus der in Kapitel 4 durchgeführten Softwareanalyse wird deutlich, dass für die Realisierung der Software eine Kombination aus Moodle und JupyterHub die gewünschten Anforderungen erfüllen kann. Moodle bietet sich insbesondere dadurch an, dass ein Großteil der erwarteten Nutzer durch die Verwendung im Lehrbetrieb der Universität bereits Erfahrungen mit dem System sammeln konnten. Außerdem implementiert Moodle bereits einige Kernfunktionen der zu entwickelnden Software, wie die Aufteilung der Nutzer in Gruppen und Rollen, die Erstellung von Veranstaltungen in Form von Kursen und die Möglichkeit der Abgabe und Bewertung von Übungen. Moodle lässt sich außerdem problemlos an das Single-Sign-On System von keycloak anbinden und bietet aufgrund des modularen Aufbaus der Software die Möglichkeit, ein Plugin zu entwickeln, welches die Moodle Plattform mit den anderen Komponenten des AI-Lab Systems verknüpft. Es gibt insgesamt 58 verschiedene Moodle Plugin Arten, wobei Aktivitätsmodule und Blöcke die am häufigsten verwendeten Plugin Typen sind (moodle.org, 2020d). Aktivitätsmodule können im Kontext einer Website als einzelne Seite innerhalb einer Veranstaltung angesehen werden, weshalb sich für die Entwicklung eines Plugins dieser Art entschieden wurde.

JupyterHub wird verwendet, da bereits eine Implementierung davon auf dem AI-Lab eingesetzt wird, welche durch einige Änderungen in der Konfiguration für die hier beschriebenen Zwecke genutzt werden kann. Ein Nachteil bei der Verwendung von JupyterHub ist, dass im Gegensatz zu der ebenfalls diskutierten Software CoCalk mit JupyterHub nicht kollaborativ gearbeitet werden kann. Mehrere Nutzer können zwar über JupyterLab auf die gleichen Dateien zugreifen, diese werden jedoch nur nach einem Speichervorgang in JupyterLab synchronisiert und nur die Änderungen von der speichernden Instanz werden übernommen. Diese Funktionen werden zwar von CoCalk ermöglicht, jedoch wurde sich gegen die Verwendung der Software entschieden, da hierfür ein weiteres System auf dem AI-Lab implementiert werden müsste, inklusive eines Docker Swarms oder Kubernetes Clusters für die Instanzierung von CoCalk.

5.1 Interaktion der Systemkomponenten

Die gewählten Komponenten müssen in das bestehende AI-Lab System integriert werden. Abbildung 5.1 zeigt, wie dies für Moodle realisiert wurde und stellt zusätzlich die Interaktion zwischen den verschiedenen Komponenten dar. Die betrachteten Komponenten sind hier das Moodle System, inklusive Kurs- und Aktivitäts-Instanzen, der JupyterHub Server, der Authorisations-Server, die Startseite des AI-Labs, die DKAN Plattform, der Netzwerkspeicher und der Webbrowser des Nutzers.

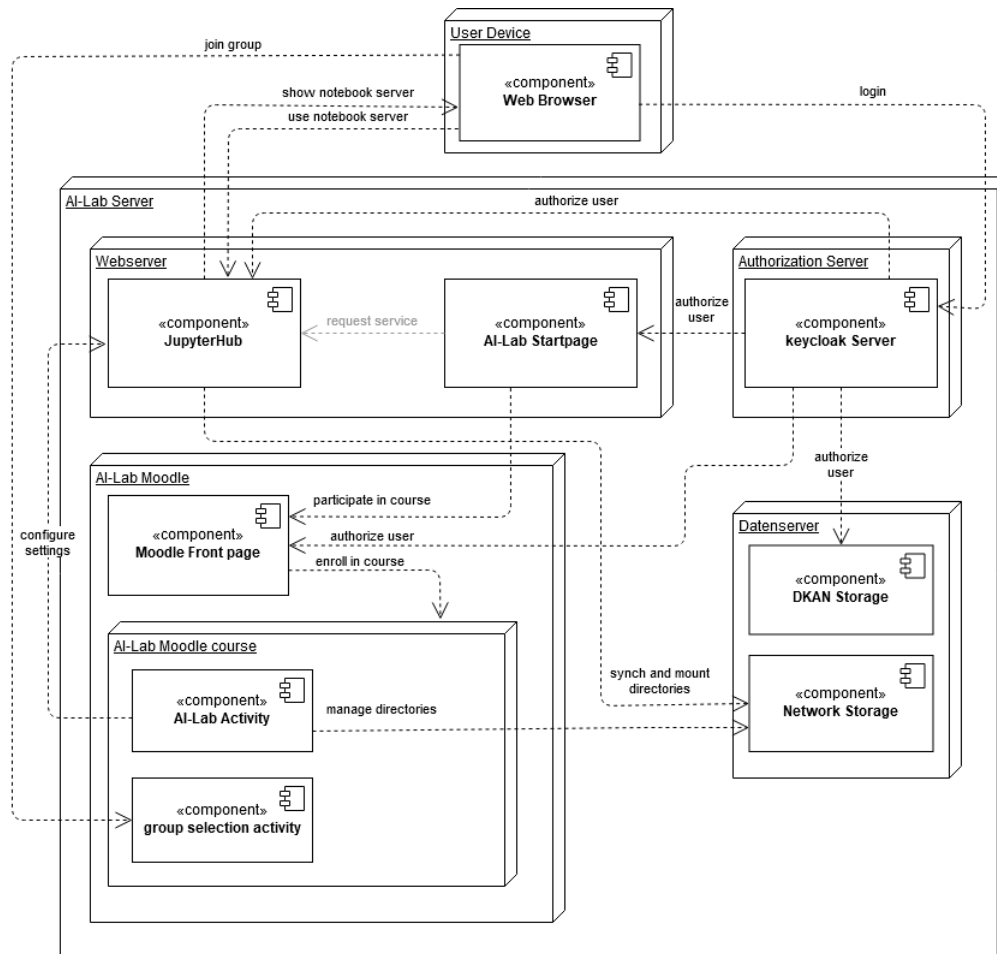


Abbildung 5.1: Komponentendiagramm, welches das Zusammenspiel der einzelnen Komponenten des AI-Labs mit der Moodle Komponente modelliert.

Die "AI-Lab Startpage" Komponente fungiert dabei als zentraler Ausgangspunkt des Systems. Von der Startpage aus kann auf das Moodle System zugegriffen werden und ein privater JupyterLab Server kann gestartet werden. Zusätzlich kann auf das Open-Storage System DKAN zugegriffen werden, welches für das Speichern von Test-, Trainings- und Validierungsdatensätzen verwendet wird. Bevor aber auf diese Komponenten zugegriffen werden kann, muss sich der Nutzer über das Authorisierungsprogramm "keycloak"

verifizieren. Ein über keycloak angemeldeter Nutzer ist für alle Komponenten des Systems verifiziert. Die Verifizierung muss nicht über die Startpage erfolgen, sondern kann beispielsweise auch über das Moodle System oder die DKAN Plattform direkt stattfinden. Verifizierte Nutzer können dann von der Startseite der Moodle Website aus einem Kurs beitreten. Innerhalb eines Kurses gibt es eine AI-Lab Aktivität und eine Gruppenauswahl Aktivität, wobei als "Aktivitäten" Instanzen eines Moodle Plugins bezeichnet werden. Über die Gruppenauswahl Aktivität kann sich ein Nutzer in eine Gruppe eintragen. Die AI-Lab Aktivität kann auf dem Netzwerkspeicher die für die Veranstaltung benötigten Ordner anlegen und Dateien wie das Codegerüst in die entsprechenden Ordner kopieren. Sie kann ebenfalls JupyterHub für die Teilnehmer der Veranstaltung so konfigurieren, dass die auf dem Netzwerkspeicher erstellten Verzeichnisse den JupyterLab Servern angehängt werden. Über JupyterHub kann dann ein JupyterLab Single-User Server angefordert werden. In diesem Fall werden die nötigen Verzeichnisse gemounted und der JupyterLab Server wird gestartet, mit dem der Nutzer über einen Webbrowser interagieren kann. Wenn in dem JupyterLab Server ein Speichervorgang durchgeführt wird, wird das angehängte Verzeichnis mit dem auf dem Netzwerkspeicher synchronisiert. Über JupyterLab kann außerdem auf die privat angelegten Projekte und Dateien zugegriffen werden. Hinter den Kulissen werden alle an JupyterLab angehängten Verzeichnisse über einen einzelnen Synology DiskStation Manager Nutzer verwaltet und direkt über das Linux-System des Netzwerkspeichers den Servern angehängt.

5.2 Modellierung der Nutzerinteraktionen

Nachdem in 5.1 die einzelnen Komponenten des Systems und deren Verknüpfung diskutiert wurde, kann nun anhand dieser Komponenten die Interaktion der Nutzer mit dem System modelliert werden. Hierzu wurden die Rollen der Nutzer auf die eines "Teilnehmers" und eines "Moderators" vereinfacht. Die folgenden Abläufe beschreiben, wie die Nutzer im Idealfall mit dem System interagieren, wenn eine neue Veranstaltung erstellt und ausgeführt werden soll.

Vorbereitung

Bevor an einer Veranstaltung teilgenommen werden kann, muss ein Moderator dieser Veranstaltung einige Vorbereitungen treffen. In Abbildung 5.2 wurden die Interaktionen der Systemkomponenten und die des Moderators anhand eines Sequenzdiagramms modelliert.

Als Startpunkt der Interaktion wird hier von der Login-Seite der AI-Lab Moodle Komponente ausgegangen. Auf dieser Login-Seite fordert der Nutzer eine Anmeldung über das AI-Lab an. Dies führt zu einer Weiterleitung zu der keycloak Seite des Digital Hub Lübeck, wo die Anmeldedaten nun eingegeben werden können. Nach erfolgreicher Anmeldung wird der Moderator wieder zu der Moodle Seite weitergeleitet, wo dieser jetzt angemeldet ist.

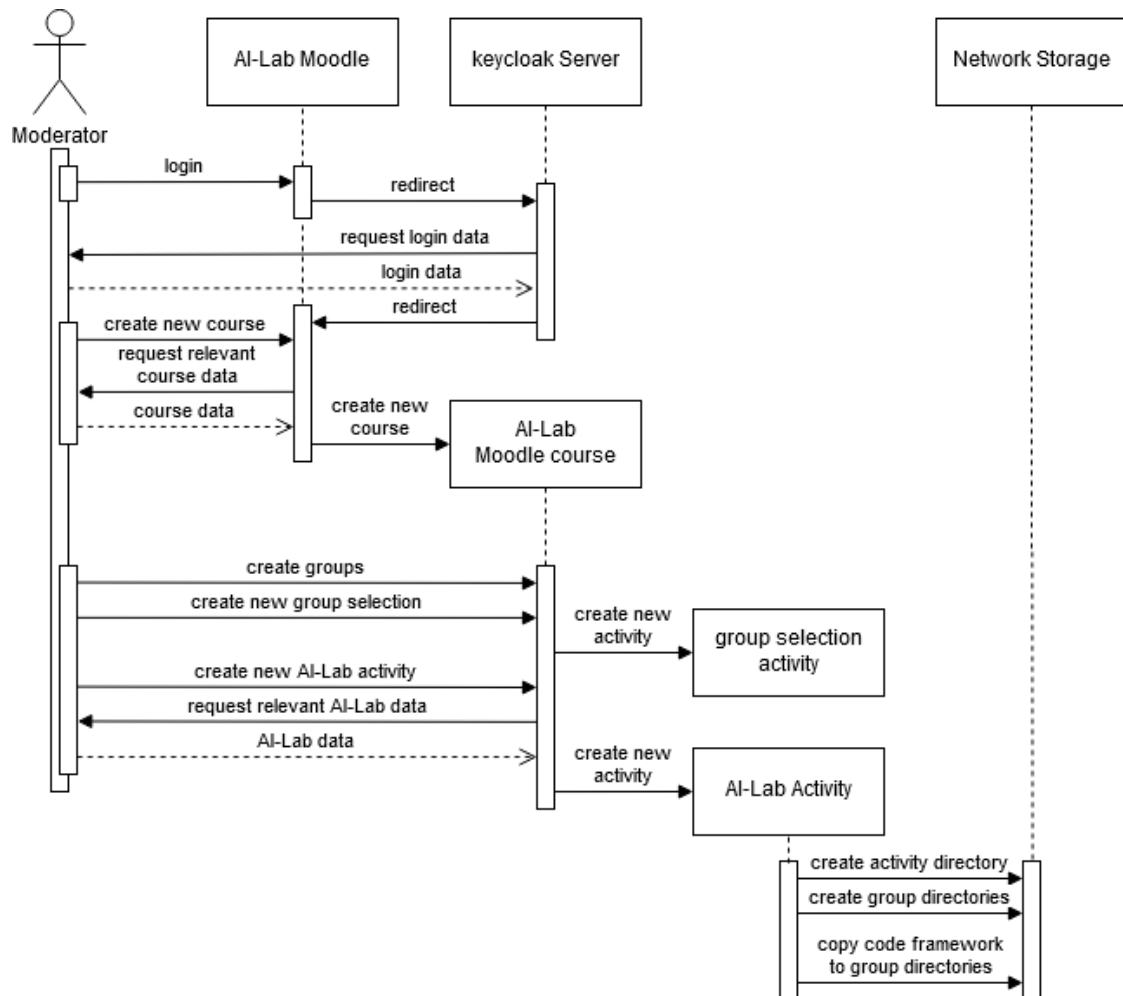


Abbildung 5.2: Sequenzdiagramm, welches die Anmeldung und Vorbereitung einer Veranstaltung aus Sicht eines Moderators modelliert. Die verschachtelten Aktivierungsbalken in der Lebenslinie des Akteurs repräsentieren, mit welchem System interagiert wird.

Der Moderator kann nun einen neuen Moodle Kurs anlegen. Dafür müssen einige Einstellungen für die Veranstaltung festgelegt werden. Es müssen mindestens der Name und der Kurzname der Veranstaltung festgelegt werden, zusätzlich können optionale Einstellungen wie der Gültigkeitszeitraum und ein Einschreibeschlüssel festgelegt werden. Es kann ebenfalls eingestellt werden, ob Gruppen in dem Kurs verwendet werden sollen. Alle hier getroffenen Einstellungen können zu einem späteren Zeitpunkt, sofern nötig, noch geändert werden.

Nachdem der Kurs erstellt wurde, müssen Gruppen angelegt werden. Hier bietet es sich an, mithilfe des Drittanbieterplugins "group choice" eine Aktivität zur Gruppenauswahl zu erstellen, sodass sich die Teilnehmer eigenständig in eine Gruppe einschreiben können. Es ist für Moderatoren jedoch auch möglich die Gruppen manuell mit Teilnehmern zu füllen. Nun muss noch die eigentliche AI-Lab Aktivität erstellt werden. Hier müssen

ebenfalls einige Einstellungen ausgewählt werden, wie der Name der Aktivität und ein Gültigkeitszeitraum für diese. Außerdem kann hier ein Code-Gerüst für die Veranstaltung hochgeladen werden. Auch bei der Aktivität ist es möglich die getroffenen Einstellungen zu einem späteren Zeitpunkt zu ändern. Wurden alle Einstellungen gesetzt, wird innerhalb des Moodle Kurses von Moodle eine neue AI-Lab Aktivität erstellt und es wird automatisch ein Ordner für diese Aktivität auf dem Netzwerkspeicher erstellt. In dem Aktivitätsordner wird dann für jede Gruppe ein Ordner angelegt, in den das Code-Gerüst kopiert wird.

Nachdem der Moodle Kurs erstellt wurde, können Teilnehmer diesem beitreten. Der Ablauf dazu ist in Abbildung 5.3 anhand eines Sequenzdiagramms modelliert. Die Anmel-

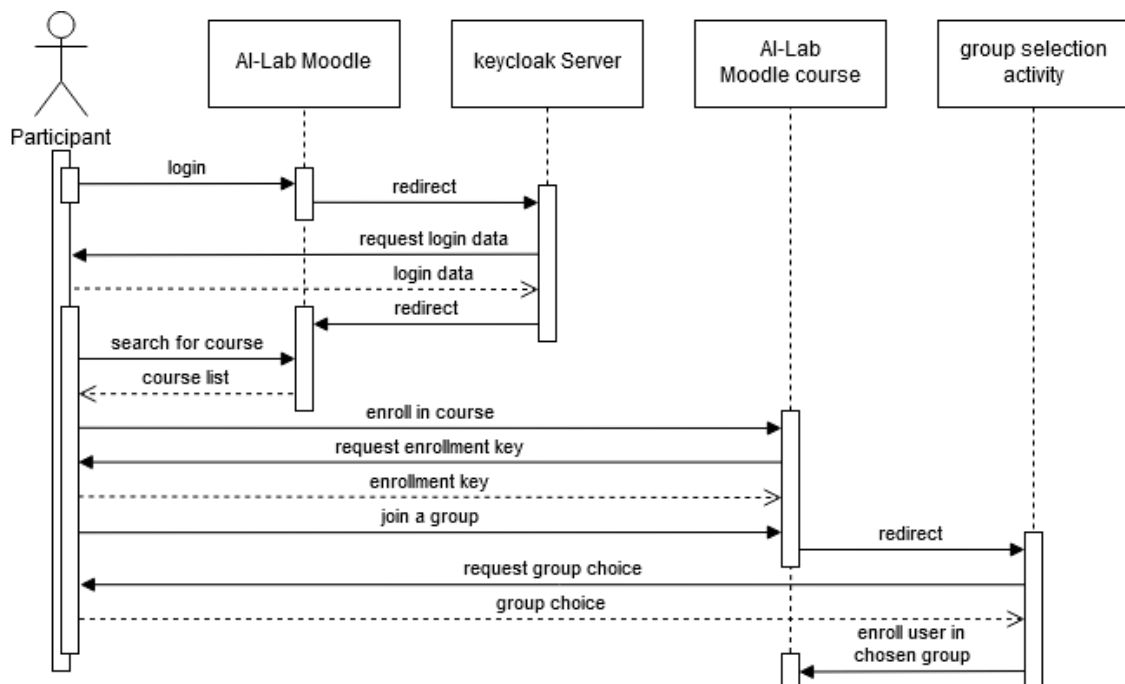


Abbildung 5.3: Sequenzdiagramm, welches die Anmeldung und Vorbereitung einer Veranstaltung aus Sicht eines Teilnehmers modelliert. Die verschachtelten Aktivierungsbalken in der Lebenslinie des Akteurs repräsentieren, mit welchem System interagiert wird.

derung der Teilnehmer auf Moodle erfolgt analog zu der Anmeldung der Moderatoren, welche zuvor bereits genauer beschrieben wurde. Nach erfolgreicher Anmeldung können die Teilnehmer über die Suchfunktion des Moodles nach der gewünschten Veranstaltung suchen. Wenn die Veranstaltung bzw. der Kurs gefunden wurde, kann der Teilnehmer dem Kurs gegebenenfalls nach Eingabe eines Einschreibeschlüssels beitreten und auf die Aktivitäten des Kurses zugreifen. Falls die zuvor erwähnte Gruppenauswahl Aktivität verwendet wird, kann der Teilnehmer nun einer Gruppe beitreten.

Durchführung einer Veranstaltung

Nachdem die beschriebenen Vorbereitungen ausgeführt wurden, kann die eigentliche Veranstaltung beginnen. Die verschiedenen Interaktionen hierfür sind im Folgenden wieder nach Moderator und Teilnehmer getrennt aufgeteilt.

Abbildung 5.4 modelliert die Interaktionen eines Moderators mit dem System als Sequenzdiagramm. Wenn alle Teilnehmer angemeldet und in Gruppen eingetragen sind, kann der Moderator über die AI-Lab Aktivität JupyterHub konfigurieren. Je nach Rolle des Nutzers wird festgelegt, welches Verzeichnis dem JupyterLab Server angehängt wird. Nun können sowohl die Moderatoren als auch die Teilnehmer ihren Jupyter Workspace

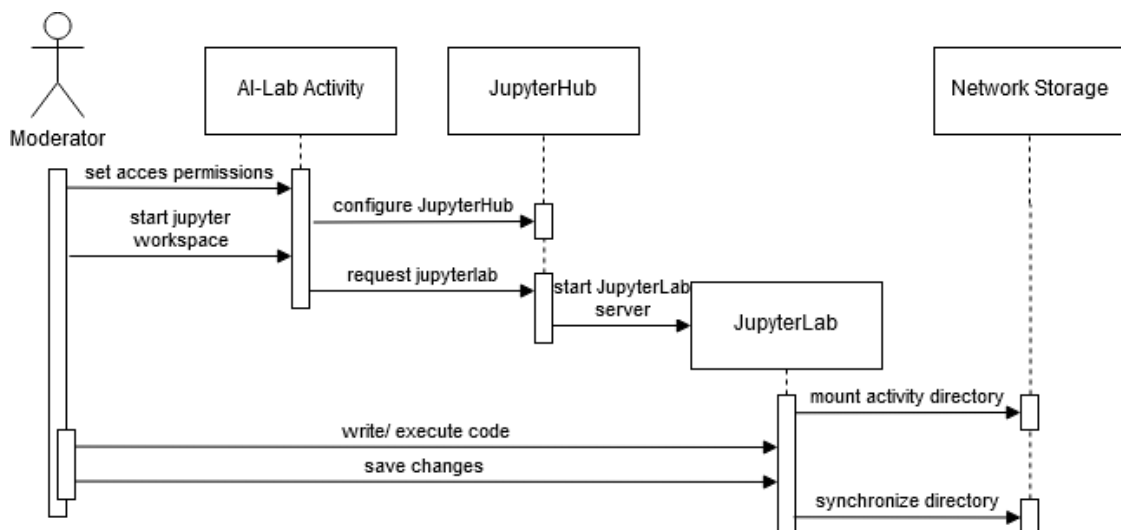


Abbildung 5.4: Sequenzdiagramm, welches die Interaktion mit einer Veranstaltung aus Sicht eines Moderators modelliert. Die verschachtelten Aktivierungsbalken in der Lebenslinie des Akteurs repräsentieren, mit welchem System interagiert wird.

starten. Abbildung 5.5 modelliert hier den Ablauf für Teilnehmer als Sequenzdiagramm, wobei dieser in vielen Punkten analog zu dem für Moderatoren ist. Wenn ein Jupyter Workspace angefordert wurde, leitet die AI-Lab Aktivität den Nutzer zu JupyterHub weiter. Hier wird dann ein JupyterLab Server, mit dem entsprechenden Gruppenverzeichnis, gestartet, zu dem der Nutzer weitergeleitet wird. Nun kann der Nutzer beliebig mit den Funktionen von JupyterLab interagieren, wie bereits in 4.2 beschrieben. Wenn der Nutzer Änderungen in JupyterLab speichert, wird das angehängte Verzeichnis mit dem Verzeichnis auf dem Netzwerkspeicher synchronisiert. Für die Moderatoren wird hingegen das ganze Aktivitätsverzeichnis inklusive der sich darin befindenden Gruppenverzeichnisse angehängt. Dementsprechend können die Moderatoren nicht nur auf alle Arbeitsbereiche der Teilnehmer zugreifen, um diesen gegebenenfalls zu assistieren, sie können auch in dem Aktivitätsverzeichnis Skripte und Notebooks zur Evaluation der Lösungen ausführen. Dadurch muss nicht jede Lösung einzeln evaluiert werden, da über alle Lösungen iteriert werden kann.

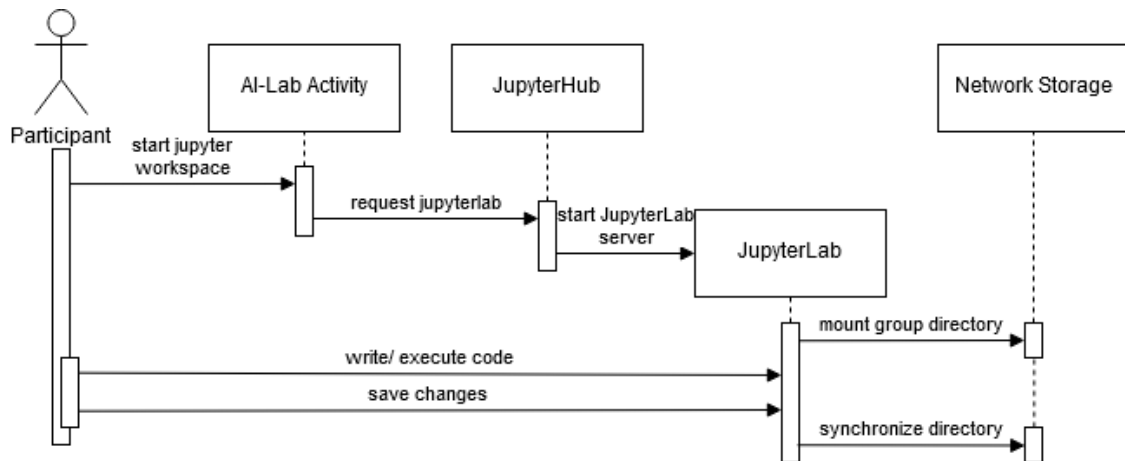


Abbildung 5.5: Sequenzdiagramm, welches die Interaktion mit einer Veranstaltung aus Sicht eines Teilnehmers modelliert. Die verschachtelten Aktivierungsbalken in der Lebenslinie des Akteurs repräsentieren, mit welchem System interagiert wird.

5.3 Praktische Implementierung

Nachdem in 5.1 und 5.2 die Funktionen und Abläufe der Software modelliert wurden, kann diese nun anhand des Modells implementiert werden. Dabei liegt der Schwerpunkt in der Implementierung eines Moodle Plugins, welches den Netzwerkspeicher und JupyterHub miteinander verknüpft und die nötigen Verzeichnisse auf dem Netzwerkspeicher anlegt.

Das Moodle Plugin

Die Entwicklung eines Moodle Plugins erfolgt primär in PHP, wobei die Einbindung von Javascript Dateien ebenfalls möglich ist. Wie zuvor erwähnt, wurde sich für die Entwicklung eines Aktivitätsmodul Plugins entschieden. Aktivitätsmodule sind in sechs grundlegende Bereiche unterteilt, welche jedes Aktivitätsmodul benötigt, um in Moodle eingebunden zu werden. Es gibt einen Datenbank Ordner, in dem alle nötigen Dateien zum Konfigurieren der Datenbankfelder eines Moduls abgelegt werden, einen Sprachen Ordner, in dem Dateien für die Übersetzung der angezeigten Texte in verschiedene Sprachen gespeichert werden und eine `version.php` Datei, in der die aktuelle Version des Plugins und die benötigte Moodle Version festgehalten werden. Diese Dateien enthalten mehr oder weniger statische Informationen, die von dem Moodle-System für das Installieren und Aktualisieren des Plugins benötigt werden. In der Datei `mod_form.php` werden Eingaben, die für das Erstellen und Aktualisieren einer Plugin-Instanz nötig sind, definiert. In dieser Datei werden ebenfalls Funktionen zur Validierung dieser Eingaben implementiert. Nach der Validierung werden die Eingaben an `lib.php` weitergereicht, eine Datei, die Funktionen enthält, in denen definiert wird, was passiert, wenn eine neue Instanz des Plugins erstellt wird, eine bestehende Instanz aktualisiert wird oder eine Instanz gelöscht wird. Zuletzt gibt es noch die Anzeigedateien `view.php`, welche den eigentlichen Inhalt des Moduls darstellt und `index.php`, in der eine Auflistung der Instan-

zen des Plugins in dem entsprechenden Kurs dargestellt wird (moodle.org, 2020c). Die Struktur dieses Grundgerüsts ist in Abbildung 5.6 dargestellt.

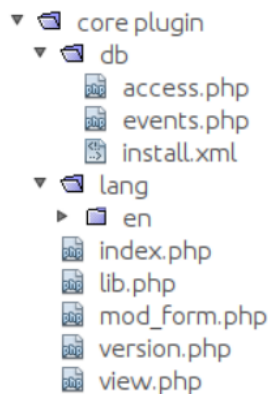


Abbildung 5.6: Grundlegende Dateihierarchie eines Aktivitätsmoduls in Moodle.

Neben den bereits erwähnten essentiellen Dateien wurden einige "Hilfsdateien", welche Skripte und Funktionen für nicht-systemrelevante Features, wie Templates zum Darstellen von Seiteninhalten, enthalten, implementiert. Die Klasse `SynologyAPIInterface` implementiert Funktionen, mit denen die Synology Storage Server API verwendet werden kann, welche zum Erstellen der Aktivitäts- und Gruppenverzeichnisse sowie zum Kopieren des Codegerüsts und Auslesen eines Scoreboards verwendet wird. Um die nötigen Zugriffsrechte auf Synology zu gewährleisten, muss ein Moodle Administrator die entsprechenden Anmeldedaten in den globalen Einstellungen des Plugins angeben. Die Datei `settings.php` stellt dafür die entsprechenden Eingabefelder zur Verfügung. Mit der Klasse `leaderboard.php` lässt sich ein Leaderboard von dem Synology Server herunterladen und in Diagramm- und Tabellenform anzeigen.

Die Synology API

Um auf dem Netzwerkspeicher die nötigen Dateien und Ordner für eine Veranstaltung anzulegen, wurde die offizielle Entwicklerdokumentation der Synology FileStation API verwendet (Inc., 2013). Der grundlegende Workflow mit der API lässt sich in drei Schritte aufteilen. Zuerst muss sich mit einem gültigen Account über die API angemeldet werden, dann können API Aufrufe ausgeführt werden und zuletzt muss eine Abmeldung des Nutzers über den entsprechenden API Aufruf stattfinden.

Nach einigem Testen wurde festgestellt, dass die Offizielle FileStation API Dokumentation nicht mehr aktuell war. So wurde der Einstiegspunkt für die API beispielsweise auf `entry.cgi` vereinheitlicht und einige Methodennamen wurden geändert. Um trotzdem die API nutzen zu können, wurden die benötigten Operationen manuell in der Synology Oberfläche ausgeführt. Über die Netzwerkanalyse Funktion des Browsers konnten so die getätigten POST und GET Anfragen ausgelesen werden und mithilfe des Syntax der Dokumentation in funktionierende API Aufrufe übersetzt werden.

Für die Ausführung der entsprechenden API-Aufrufe wird der gleiche Synology User wie für JupyterHub verwendet und ein Ordner für Veranstaltungen wurde angelegt. In diesem Ordner werden dann von dem Synology User die Verzeichnisse für die einzelnen Veranstaltungen angelegt.

Jupyterhub

Über ein typescript Skript wird auf Keycloak für jeden Nutzer, der an einem Hackathon teilnimmt, ein entsprechendes Attribut gesetzt. Es wird ebenfalls ein Attribut gesetzt, welcher Gruppe der Nutzer angehört und an welchem Hackathon dieser teilnimmt. In der Datei `jupyter_config.py` des JupyterHub Servers werden dann die entsprechenden Attribute ausgelesen und der Gruppenordner in dem dazugehörigen Hackathonordner wird den "Mounts" des DockerSpawners hinzugefügt. Für Moderatoren wird die Gruppe auf "mod" gesetzt, was dazu führt, dass in der Konfiguration der ganze Veranstaltungsordner angehängt wird. Das Skript wurde speziell für den "Maritimen Hackathon" am 10.09.2020 erstellt, weshalb hier nur von Hackathon Veranstaltungen die Rede ist, es ist jedoch leicht zu erkennen, wie es auch für andere Veranstaltungen verwendet werden kann.

6

Evaluation

Das in Kapitel 5 ausgearbeitete System konnte anhand von zwei Tests evaluiert werden. Zuerst wurden, um die Eignung des Konzepts zu testen, die Abläufe des Softwaremodells per Hand beim "Maritimen Hackathon" ausgeführt und getestet. Es wurde ebenfalls eine implementierte Version des Moodle Plugins getestet und evaluiert.

6.1 Test des Konzepts am Hackathon

Am 10.9.2020 fand die erste Instanz des "Maritimen Hackathons" statt. Zu diesem Anlass wurden die in 5.2 ausgearbeiteten Schritte "manuell" auf dem AI-Lab ausgeführt, um zu testen, ob das ausgearbeitete Konzept praktikabel ist. Aus technischer Sicht hat der Ablauf größtenteils problemlos funktioniert, jedoch musste der Single-User Server eines Nutzers neu gestartet werden, wenn dieser zur Laufzeit einem Hackathon hinzugefügt wurde, da der betreffende Nutzer andernfalls die hinzugefügten Verzeichnisse nicht sehen konnte. Es wurde sowohl auf keras und pytorch Code-Gerüsten gearbeitet und es konnte mit beiden Frameworks problemlos trainiert werden. Es wurde angemerkt, dass die Gruppeneinteilung zu Beginn der Veranstaltung viel Zeit in Anspruch genommen hat. Ebenfalls wurde festgestellt, dass das manuelle Kopieren der Code-Gerüste in die entsprechenden Gruppenverzeichnisse gerade bei einer größeren Gruppenmenge als mühselig empfunden wurde. Diese beiden Aufgaben werden durch das Moodle Plugin automatisiert, es ist somit anzunehmen, dass diese Kritikpunkte bei zukünftigen Veranstaltungen entfallen.

6.2 Evaluation des Prototyps

Der Prototyp setzt in Kombination mit den restlichen Komponenten des AI-Labs einen Großteil der in Tabellen 3.4 und 3.5 aufgeführten Anforderungen um. Aus diesen Tabellen werden A1.1 bis A1.5, A1.15, A1.16 und A2.1 durch die Features von Moodle direkt umgesetzt, wobei A1.6 und A2.6 durch die Verwendung von Drittanbieterplugins integriert werden können. A1.8 kann durch die bereits bestehende DKAN Plattform des AI-Labs ebenfalls abgehakt werden. Durch JupyterHub und Docker werden die nötigen Funktio-

nalitäten für A2.3, A2.4 und A2.5, sowie A1.10, A1.11 und A1.14 gedeckt. Dadurch, dass Moderatoren in ihrem JupyterLab Server Zugriff auf alle Gruppenordner gewährt wird, können ebenfalls A1.12 und A1.13 umgesetzt werden. A2.2, A2.9 und A1.7, das Anlegen der Workspaces auf dem Netzwerkspeicher und die Zuweisung der Workspaces zu den entsprechenden Nutzern, wird von dem Moodle Plugin und JupyterHub übernommen. Ebenfalls kann das Plugin ein Scoreboard (A1.17), in Form einer JSON Datei, vom Netzwerkspeicher einlesen und in Moodle als Rangliste, in Form einer Tabelle und eines Balkendiagramms, darstellen. A2.8 wurde nur teilweise umgesetzt, wobei es durchaus möglich ist, eine mobile Version des Plugins zu verwenden, die dafür nötigen Klassen wurden jedoch nicht implementiert, da davon ausgegangen wurde, dass nicht an mobilen Geräten, wie Smartphones, programmiert wird. A2.9 konnte ebenfalls nur teilweise umgesetzt werden, so steht zwar für das Moodle Plugin eine deutsche und eine englische Version zur Verfügung, JupyterHub wird jedoch nur in englischer Ausgabe zur Verfügung gestellt. A1.9, das kollaborative Bearbeiten der Notebooks, konnte nicht umgesetzt werden, da JupyterHub nicht die dafür notwendigen Funktionen zur Verfügung stellt und sich aufgrund der in 4.2 aufgeführten Gründe gegen die Verwendung von CoCalk entschieden wurde.

Der Prototyp kann die in 6.1 erwähnten Kritikpunkte in Bezug auf das Kopieren und Anlegen der Verzeichnisse sowie die Auswahl der Gruppen automatisieren. Nach Rücksprache mit den Veranstaltern des Hackathons, wurde sich gegen die Verwendung eines wie in 4.1 beschriebenen Videochat Plugins entschieden. Stattdessen wurde ein externes Tool verwendet, da die Teilnehmer aus der Industrie Schwierigkeiten haben könnten in Moodle zu navigieren und somit auch damit dem Videochat beizutreten. Insgesamt konnte ein Prototyp implementiert werden, welcher die verschiedenen Komponenten des AI-Labs miteinander verknüpft und einen Großteil der aufgeführten Anforderungen umsetzt.

7

Zusammenfassung und Ausblick

In dieser Arbeit wurde unter Anwendung gängiger Softwareentwicklungsverfahren eine Architektur zur Verwendung des AI-Labs für drei Anwendungsfälle aus der Lehre entwickelt. Zu diesem Zweck wurde zunächst eine Anforderungsanalyse für die Anwendungsfälle durchgeführt, wobei festgestellt wurde, dass Lernmanagementsysteme viele der aufgestellten Anforderungen bereits erfüllen und eine gute Grundlage für das System bieten könnten. Dementsprechend wurden verschiedene Lernmanagementsysteme unter dem Aspekt der Integrierung in das AI-Lab System analysiert, wobei sich letztendlich für die Verwendung der Open Source Plattform Moodle entschieden wurde. Es wurden ebenfalls mögliche Jupyter Notebook Dienste analysiert, da Jupyter Notebooks für die Bearbeitung der Aufgaben aller Anwendungsfälle verwendet werden sollen. Es wurde sich schließlich für die Verwendung von JupyterHub entschieden, da es bereits Bestandteil des AI-Lab Systems ist und somit keine weitere Komponente dem System hinzugefügt werden muss. Insgesamt wurde sich für die Entwicklung eines Moodle Plugins entschieden, welches die Verzeichnisse auf dem Netzwerkspeicher verwaltet und JupyterHub so konfiguriert, dass die entsprechenden Netzwerkverzeichnisse in den von JupyterHub erstellten JupyterLab Single-User Servern verwendet werden.

Das Moodle Plugin wurde letztendlich auch Form eines Prototypen implementiert und getestet. Mithilfe des Plugins konnten so viele der diskutierten Anforderungen umgesetzt werden. Viele repetitive Arbeitsabläufe wie das Anlegen von Workspaces und das Kopieren der Code-Gerüste konnten ebenfalls automatisiert werden. Es konnte also eine Softwarearchitektur für die Verwendung der AI-Lab Infrastruktur in der Lehre entwickelt und prototypisch implementiert werden.

7.1 Ausblick

Das System weist noch einige Schwächen auf. So ist es entgegen der Anforderungsanalyse nicht möglich, die Notebooks kollaborativ zu bearbeiten. Dies könnte durch die Einbindung von CoCalk, wie in 4.2 beschrieben, umgesetzt werden. Es wäre ebenfalls denkbar, das Plugin an das universitäre Moodle anzubinden, um den Nutzern die Verwirrung in der Verwendung zweier Moodle Plattformen zu ersparen. Da das Uni-Moodle jedoch

nur für Studenten und Mitarbeiter der Universität zugänglich ist, müssten die außer-universitären Teilnehmer zukünftiger Veranstaltungen weiterhin eine separate Moodle Plattform verwenden. Für Übungen und Veranstaltungen, die über einen längeren Zeitraum stattfinden, wäre es sinnvoll das Konfigurieren von JupyterHub weiter zu automatisieren, sodass ein Moderator nicht bei jedem Gruppenwechsel oder neuem Teilnehmer die entsprechenden Konfigurationen setzen muss. Die Konfiguration könnte beispielsweise immer ausgeführt werden, wenn ein Nutzer einer Gruppe beitrifft. Außerdem ist das Einlesen des Scoreboards in dem Prototypen noch sehr rigide. Hier könnte flexibler Code geschrieben werden, um das Einlesen unterschiedlicher Wertearten zu ermöglichen.

Das hier präsentierte System bietet eine Grundlage zur Umsetzung der drei vorgestellten Anwendungsfälle. Das bedeutet auch, dass es, um für alle Anwendungsfälle nutzbar zu sein, gegebenenfalls noch spezifischer an diese angepasst werden muss. So lässt sich ein Hackathon mit dem System realisieren, genauso wie Übungen, wenn die entsprechenden unterstützenden Moodle-Funktionen, wie die Abgabe-Aktivität verwendet werden. Um jedoch eine Competition zu realisieren, müssen noch einige Funktionen implementiert werden. Während die "Runden" durch die Verwendung mehrerer Instanzen des Plugins realisiert werden könnten, müsste bei jeder Runde JupyterHub neu konfiguriert werden, sodass Verzeichnisse von vorherigen Runden nicht mehr verwendet werden können. Die Bewertung der Rundenergebnisse sowie das Bereitstellen der Modelle aus vorherigen Runden zum Angriff sollte dafür ebenfalls automatisiert werden.

Literatur

DKAN (2020). *DKAN Documentantation*. dkan, <https://demo.getdkan.org/modules/contrib/dkan/docs/index.html>.

Dunand, N. (Juli 2020). *Activities: Group choice*. moodle, https://moodle.org/plugins/mod_choicegroup.

Firesmith, D.G. (1999). Use case modeling guidelines. In *Proceedings of Technology of Object-Oriented Languages and Systems - TOOLS 30* (Cat. No.PR00278), S. 184–193.

Fred Dixon, J.F. (Juli 2020). *Activities: BigBlueButtonBN*. moodle, https://moodle.org/plugins/mod_bigbluebuttonbn.

ilias.de (2020a). *Über ILIAS*. ILIAS, <https://www.ilias.de/open-source-lms-ilias/>.

ilias.de (2020b). *Dokumentation für Dozenten*. ILIAS, https://docu.ilias.de/ilias.php?ref__id=5237&obj__id=71152&cmd=layout&cmdClass=illmpresentationgui&cmdNode=ba&baseClass=ilLMPresentationGUI.

ilias.de (2020c). *OpenID Connect support*. ILIAS, https://docu.ilias.de/goto_docu_wiki_wpage_4827_1357.html.

ilias.de (2020d). *Plug-ins*. ILIAS, https://docu.ilias.de/goto_docu_cat_1442.html.

ilias.de (2020e). *Rollen- und Rechteverwaltung*. ILIAS, https://docu.ilias.de/ilias.php?ref__id=21&obj__id=1&cmd=layout&cmdClass=illmpresentationgui&cmdNode=ba&baseClass=ilLMPresentationGUI.

Inc., S. (2013). *Synology File Station Official API*. Synology, https://global.download.synology.com/download/Document/Software/DeveloperGuide/Package/FileStation/All/enu/Synology_File_Station_API_Guide.pdf.

Inc., S. (2020). *DiskStation Manager*. Synology, <https://www.synology.com/de-de/knowledgebase/DSM/help>.

ISO/IEC/IEEE (2018). *ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering*. ISO/IEC/IEEE 29148:2018(E), 1–104.

Joseph Maria, I., Thirupathi, D., Rajendran, R. und Velumani, B. (2019). Technologies, Challenges and Tools for Digital Learning. In 2019 IEEE Tenth International Conference on Technology for Education (T4E), S. 268–269.

Jupyter, P. (2015). *The Jupyter Notebook Format*. nbformat, <https://nbformat.readthedocs.io/en/latest/>.

Jupyter, P. (2016a). *Authenticator*. jupyterhub, <https://jupyterhub.readthedocs.io/en/stable/reference/authenticators.html>.

Jupyter, P. (2016b). *Spawners*. jupyterhub, <https://jupyterhub.readthedocs.io/en/stable/reference/spawners.html>.

Jupyter, P. (2016c). *Technical Overview*. jupyterhub, <https://jupyterhub.readthedocs.io/en/stable/reference/technical-overview.html>.

Jupyter, P. (2018). *JupyterLab Documentation*. jupyterlab, <https://jupyterlab.readthedocs.io/en/stable/index.html>.

Jupyter, P. (Juli 2020). *Project Jupyter*. jupyter, <https://jupyter.org/>.

Kaggle (2020). *Titanic: Machine Learning from Disaster*. Kaggle, <https://www.kaggle.com/c/titanic/overview/frequently-asked-questions>.

Keycloak (2020). *About Keycloak*. Keycloak, <https://www.keycloak.org/about>.

Kotoyna, G., Sommerville, I., Wiley, J. u. a. (1999). *Requirements engineering: processes and techniques*.

Luber, S. (März 2019). *Was bedeutet LAMP?* Cloudcomputing Insider, <https://www.cloudcomputing-insider.de/was-bedeutet-lamp-a-808284/>.

Mayman, D. (2016). *root-the-box*. root-the-box, <https://root-the-box.com/>.

moodle.org (2011). *File:Moodle-contexts-1.8.png*. moodle, <https://docs.moodle.org/dev/File:Moodle-contexts-1.8.png>.

moodle.org (2013). *Standard roles*. moodle, https://docs.moodle.org/39/en/Standard_roles.

moodle.org (2014). *Context*. moodle, <https://docs.moodle.org/39/en/Context>.

moodle.org (Apr. 2018). *OAuth 2 authentication*. moodle, https://docs.moodle.org/39/en/OAuth_2_authentication.

moodle.org (2019a). *Features*. moodle, <https://docs.moodle.org/39/en/Features>.

moodle.org (2019b). *Installation von Moodle*. moodle, https://docs.moodle.org/39/de/Installation_von_Moodle.

- moodle.org (2020a). *About Moodle*. moodle, https://docs.moodle.org/39/en/About_Moodle.
- moodle.org (2020b). *Activities*. moodle, <https://docs.moodle.org/39/en/Activities>.
- moodle.org (Juli 2020c). *Activity Modules*. moodle, https://docs.moodle.org/dev/Activity_modules.
- moodle.org (Juli 2020d). *Plugin Types*. moodle, https://docs.moodle.org/dev/Plugin_types.
- OAuth (2020). *OAuth 2.0*. oauth, <https://oauth.net/2/>.
- OpenID (Jan. 2020). *Specification*. openid, <https://openid.net/connect/>.
- Randles, B.M., Pasquetto, I.V., Golshan, M.S. und Borgman, C.L. (2017). Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study. In 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), S. 1–2.
- Sabelfeld, D. (2010). *Rechte- und Rollenmanagement in ILIAS*. lernenzweinnull, <http://lernenzweinnull.de/wp-content/uploads/2010/07/Rollenverwaltung-ILIAS.pdf>.
- Sagemath, I. (2020a). *About collaborators*. CoCalc, <https://doc.cocalc.com/project-settings.html>.
- Sagemath, I. (2020b). *CoCalc Docker Image*. CoCalc, <https://doc.cocalc.com/docker-image.html>.
- Sagemath, I. (2020c). *Online Jupyter Notebooks*. CoCalc, <https://cocalc.com/doc/jupyter-notebook.html>.
- Sagemath, I. (2020d). *Online LaTeX Editor*. CoCalc, <https://cocalc.com/doc/latex-editor.html>.
- Sagemath, I. (2020e). *Online Linux Environment*. CoCalc, <https://cocalc.com/doc/linux.html>.
- Sagemath, I. (2020f). *Projects*. CoCalc, <https://doc.cocalc.com/project.html>.
- Sagemath, I. (2020g). *What is CoCalc?* CoCalc, <https://doc.cocalc.com/>.
- Snyder, H. (Mai 2017). *New SMC Jupyter Client Released*. CoCalc Blog, <https://blog.cocalc.com/jupyter/2017/05/05/jupyter-rewrite-for-smc.html>.
- Tensorflow (2020). *About Tensorflow*. Tensorflow, <https://www.tensorflow.org/?hl=de>.
- Watson, W. und Watson, S.L. (2007). An argument for clarity: What are learning management systems, what are they not, and what should they become.

A

Anwendungsfälle

Tabelle A.1: Anwendungsfallbeschreibung für den Übungsbetrieb

Use Case Name	Vorlesung Medical Deep Learning – Praktische Übung
Author	IMI, Lasse Hansen (hansen@imi.uni-luebeck.de)
Priority	Mittel – Hoch – Sehr Hoch
Pre-Conditions	
Post-Conditions	
Brief Description	Im Rahmen der Vorlesung „Medical Deep Learning“ findet ebenfalls eine praktische Übung statt. Diese besteht aus zweiwöchentlichen Programmieraufgaben in Pytorch. Diese Aufgaben sollen den Studenten inklusive Daten und Codegerüst als Jupyter Notebooks zur Verfügung gestellt werden. Die Studenten sollen die Aufgaben direkt im Notebook bearbeiten und ihre Lösungen am Ende der Bearbeitungszeit einreichen können.
Participating Actors	ProfessorIn, Wissenschaftliche MitarbeiterInnen, HiWis, Studierende
Event Flow	Die praktischen Übungen werden von den beteiligten Mitarbeitern konzipiert und getestet. Danach sollten die Übungsaufgaben inklusiver aller benötigten Daten in einem speziellen Kursbereich auf den AI-Lab Server geladen werden. Zu den entsprechenden Übungszeiten sollten dann Jupyter Notebook Server bereitgestellt werden, so dass sich Studierende einloggen und die Aufgaben direkt im Browser (mit GPU Unterstützung) bearbeiten können. Am Ende der Übungszeit sollten Studierende ihre Lösungen final einreichen können. Wissenschaftliche Mitarbeiter und HiWis können dann auf die eingereichten Lösungen zugreifen und eine Bewertung vornehmen.
Alternative Flow	Die Kursmaterialien sollen auch heruntergeladen werden können, so dass Studierende privat an den Übungen arbeiten können. Entsprechend sollte auch ein Upload von lokal erstellten Lösungen zur Verfügung stehen.
Exceptions	
User Interface	
Further Conditions	

Tabelle A.2: Anwendungsfallbeschreibung für den Hackathon

Use Case Name	Hackathon auf dem AI-Lab Server
Author	Max Gosau (max.gosau@student.uni-luebeck.de)
Priority	Mittel – Hoch – Sehr Hoch
Pre-Conditions	Im Vorfeld werden den Angemeldeten Teilnehmern Zugangsinformationen zugeschickt.
Post-Conditions	
Brief Description	Im Rahmen der Digitalen Woche Kiel soll für Teilnehmer aus der Industrie ein Hackathon (Programmier-Challenge mit Wettbewerbscharakter) im Bereich Bilderkennung veranstaltet werden. Der Hackathon soll etwa einen halben Tag dauern.
Participating Actors	Organisationsteam des Hackathons (Moderatoren), Angestellte aus Mittelständigen Unternehmen in der IT Branche (Teilnehmer), ggf. Studenten (Teilnehmer)
Event Flow	Die Moderatoren legen einen Hackathon an und laden ein Codegerüst, eine Beschreibung der Aufgabe, sowie Trainings und Validierungsdatensätze auf den open-data Server des Ki-Labors. Die Teilnehmer legen sich ein zeitlich begrenztes Konto an und treten der Hackathon Veranstaltung bei. In der Veranstaltung schreiben sich die Teilnehmer in Teams von ca. 2-4 Personen ein in denen ein Bilderkennungsproblem (bspw. Klassifizierung von Schiffen) bearbeitet wird. Dabei können alle Mitglieder eines Teams kooperativ an dem gleichen Dokument (z.B. Jupyter notebook) arbeiten. Die Moderatoren können die Dokumente aller Teams einsehen und bearbeiten, die Teams können nur auf ihrem eigenen workspace zugreifen. Die Teams können, je nach Erfahrung, das Codegerüst bearbeiten oder eine eigene Lösung programmieren. Jeder Teilnehmer kann den programmierten Code ausführen und zum Training einreichen. Am Ende speichern die Teams ihre Lösung in Form eines Textdokuments in ihrem workspace und von einem Moderator wird ein Testdatensatz hochgeladen. Die Moderatoren testen die eingereichten Lösungen mit einem Evaluationsskript und stellen einen Gewinner fest. Nach Ablauf des Gültigkeitszeitraums des Accounts wird dieser automatisch gelöscht.

Alternative Flow	Gegebenenfalls sollen die Gruppen mit Studenten der Universität aufgestockt werden.
Exceptions	
User Interface	
Further Conditions	Neben dem Python Kernel sollen einige Pakete vorinstalliert sein: numpy, sklearn, torch, torchvision, torchsummary, tensorflow, tensorboard. Die Teilnehmer können außerdem über die Konsole andere Python Pakete installieren. Jedem Team werden die gleichen Rechenressourcen zuge- teilt. (bspw. 1 GPU pro Team). Wenn der Entwicklungszeitraum es erlaubt soll ein Video- konferenztool in die Anwendung eingebunden werden. In der Anwendung gibt es, neben Sprach- und Viedochatfunktionen die Möglichkeit der Bildschirmübertragung. Neben einem „Hauptraum“ soll es für jede Gruppe einen Raum geben.

Tabelle A.3: Anwendungsfallbeschreibung für CTF-Competitions

Use Case Name	Competitions on AI-Lab Server
Author	Max Gosau max.gosau@student.uni-luebeck.de
Priority	Mittel – Hoch – Sehr Hoch
Pre-Conditions	
Post-Conditions	
Brief Description	Es soll in regulären Intervallen (jährlich) eine Competition im Bereich KI ausgeführt werden. Dazu werden in mehreren Runden KI-Probleme von Teams bearbeitet und anhand einer Bewertungsmetrik auf Genauigkeit getestet. Die Probleme werden als Jupyter Notebook zusammen mit einem Codegerüst von den Moderatoren zur Verfügung gestellt.
Participating Actors	Mitarbeiter des ITS (Moderatoren), Studenten (Teilnehmer), externe Betriebe (Teilnehmer)
Event Flow	Von den Moderatoren wird ein Jupyter Notebook, öffentliche Trainings – und Testdatensätze, versteckte Datensätze zur Überprüfung und Bewertung der Lösungen, sowie eine Bewertungsmetrik in Form eines Python Skripts bereitgestellt. Die Teilnehmer bilden Teams, in denen die Probleme bearbeitet werden und melden sich auf der Plattform an. Die Competition ist in mehrere, 30-Minütige Runden aufgeteilt. In einer Runde wird von den Teams ein Problem bearbeitet (ein Modell trainiert) und ein Angriff auf ein Modell einer anderen Gruppe aus der vorherigen Runde ausgeführt. Ziel eines Angriffes ist es eine gegebene Menge an Datensätzen auf Verwendung im Training des Modells zu überprüfen. Also: Gibt es eine Menge an Datensätzen die nicht korrekt klassifiziert werden kann. Ein Erfolgreicher Angriff führt zu Punkteabzug beim Team dessen Modell angegriffen wurde. In dem bereitgestellten Jupyter Notebook wird ein Modell programmiert und trainiert. Jeder Trainingsauftrag des Teams erzeugt eine neue Docker Instanz, in der das Modell trainiert wird. Auf Anfrage des Teams wird das Modell in einer separaten, vertrauenswürdigen Instanz mithilfe des Python Skripts und dem Bewertungsdatensatz auf Genauigkeit geprüft und basierend auf diesen Ergebnissen mit Punkten bewertet.

	Während der Bearbeitung soll es im Notebook möglich sein ein scoreboard anzuzeigen in dem Statistiken zu den einzelnen Teams zu sehen sind (z.B. die Genauigkeit des letzten Modells). Im Notebook soll außerdem ein Tensorboard Interface anzeigbar sein, um den Trainingsfortschritt live zu verfolgen. Durch eine Virtuelle Ebene (Abstraktionsebene) soll mit einem Wrapper Tensor-Flow auf dem eigene PC „simuliert“ werden (Es soll möglich sein Tensor-Flow Methoden aufzurufen). Über diese Abstraktionsebene sollen ebenfalls Log-Messages von Tensorflow dargestellt werden. Da einige Probleme sehr Rechenintensiv sein können, soll der DGX Rechner verwendet werden.
Alternative Flow	Es soll auch Teilnehmern außerhalb der Universität möglich sein an der Competition teilzunehmen. Dazu müssen diese einen Account anlegen können und der Server muss von außerhalb des Uni- Netzes erreichbar sein.
Exceptions	
User Interface	
Further Conditions	

B

Anforderungsdokument

Anforderungsdokument - KI-Labor Webanwendung

Max Gosau

September 15, 2020

Contents

1	Einleitung	2
1.1	Zweck des Dokuments	2
1.2	Konventionen	2
2	Allgemeine Beschreibung	2
2.1	Produktperspektive	2
2.2	Produktfunktionen	2
2.3	Begriffe	3
2.4	Nutzer der Software	3
3	Spezifische Anforderungen	4
3.1	funktionale Anforderungen	4
3.1.1	Verwaltung von Nutzern	4
3.1.2	Verwaltung von Veranstaltungen	5
3.1.3	Übungen	8
3.1.4	Hackathon	10
3.1.5	Competition	12
3.1.6	Allgemeines zum Bearbeiten von Lösungen	15
3.1.7	Training und Ausführung von Modellen	15
3.1.8	Bewertung und Evaluation	16
3.1.9	Hoch- und Runter-laden von Dateien	17
3.2	Nicht-funktionale Anforderungen	18
3.2.1	Sicherheit	18
3.2.2	Technik	19
3.2.3	Kompatibilität	19
3.2.4	Ergonomie	19

1 Einleitung

1.1 Zweck des Dokuments

Ziel dieses Dokumentes ist die Beschreibung einer formalen Anforderungsanalyse an eine Webanwendung für das KI-Labor, welche in der Lehre eingesetzt werden soll. Erst wird in 2 ein Überblick über die Software gegeben, anschließend werden in 3 die einzelnen Anforderungen genauer spezifiziert und klassifiziert.

1.2 Konventionen

Dieses Dokument richtet sich grob nach dem IEEE Standard 29148:2018 für die Erstellung von Anforderungsdokumenten in der Softwareentwicklung [1].

2 Allgemeine Beschreibung

2.1 Produktperspektive

Die Software soll, ähnlich wie kaggle.com [2] Funktionen zur Bearbeitung von KI bzw. Deep Learning Problemen zur Verfügung stellen. Die Software soll in der Lehre eingesetzt werden, weshalb ein Kurssystem vergleichbar mit dem von moodle [3] verwendet wird.

2.2 Produktfunktionen

Die Kernfunktion der Software ist die Bearbeitung von KI- und Deep learning-Problemen. Der Nutzer soll dabei die Programmierung, das Training und das Testen eines Modells vom Browser aus steuern können.

Die Software lässt sich dabei in drei übergeordnete Anwendungsfälle aufteilen:

Übungsaufgaben: In Begleitung zu Lehrveranstaltungen sollen Übungsaufgaben im Bereich KI bearbeitet werden. Die programmierten Lösungen sollen trainiert, getestet und bewertet werden.

Competition: Das ITS veranstaltet eine jährliche Competition, in der Teams in mehreren Runden gegeneinander antreten, um das beste Neuronale Netz zu trainieren. In jeder Runde erfolgt außerdem ein "Angriff" auf die Modelle der anderen Teams.

Hackathon: Im Rahmen der Digitalen Woche Kiel soll ein Hackathon (Coding Challenge mit Wettbewerbscharakter) stattfinden. Hier treten mehrere Teams gegeneinander an um ein Bilderkennungsproblem zu lösen.

2.3 Begriffe

Begriffserklärungen für häufig verwendete Begriffe.

Veranstaltung: Eine Veranstaltung ist vergleichbar mit einem Kurs auf moodle.com. Nutzer können einer Veranstaltung beitreten. Nur Nutzer die einer Veranstaltung beigetreten sind, haben Zugriff auf die darin veröffentlichten Übungen, Competitions oder Hackathons. Veranstaltungen sind zeitlich begrenzt und werden nach einem gewissen Zeitraum wieder gelöscht.

Rollen: Für die Software gibt es verschiedene Rollen, die jeweils mit verschiedenen Rechten verbunden sind. Es bieten sich vier Rollen an:

Administrator: Der Administrator kann Veranstaltungen anlegen und Nutzer verwalten.

Veranstaltungsverwalter: Personen von jedem Institut, das die Software verwenden möchte. Veranstaltungsverwalter können Veranstaltungen erstellen und in diesen „Nutzern“ Rollen zuweisen

Moderator: In einer Veranstaltung kann ein Administrator Moderatoren festlegen, welche für die Verwaltung der Veranstaltung zuständig sind. Ein Moderator kann bspw. Übungen erstellen, Teams bilden etc.

Teilnehmer: Teilnehmer haben die wenigsten Rechte. Sie können Veranstaltungen beitreten, Aufgaben bearbeiten und abgeben etc. Teilnehmer werden die größte Gruppe der Nutzer ausmachen.

Eine genauere Beschreibung welche Rechte welcher Rolle zugewiesen sind ist den jeweiligen Anforderungen in 3 zu entnehmen.

Angriff: Parallel zu der Bearbeitung einer Lösung kann in einer Runde einer Competition ein Angriff auf ein Modell eines anderen Teams durchgeführt werden. Ziel eines Angriffes ist es herauszufinden, ob es eine Menge an Datensätzen gibt, die von dem Modell des anderen Teams nicht korrekt klassifiziert werden kann. Ein erfolgreicher Angriff führt zu Punkteabzug beim Team dessen Modell angegriffen wurde.

2.4 Nutzer der Software

Studenten: Studenten werden Die Software zum bearbeiten von Übungsaufgaben und Competition Problemen nutzen.

Professoren und Wissenschaftliche Mitarbeiter: Professoren und Wissenschaftliche Mitarbeiter der Universität werden die Software nutzen, um Veranstaltungen für Lehrzwecke zu erstellen und zu verwalten.

Teilnehmer aus der Industrie: Für die Competition und den Hackathon ist es Vorgesehen, dass auch Teilnehmer aus der Industrie teilnehmen können. Insbesondere ist zu erwarten, dass diese einen Hintergrund in IT haben.

Annahmen: Es ist von den Teilnehmergruppen anzunehmen, dass diese bereits Erfahrungen im Bereich IT und Programmierung aufweisen.

3 Spezifische Anforderungen

3.1 funktionale Anforderungen

In diesem Teil erfolgt eine Auflistung und Erklärung der aus Nutzergesprächen ermittelten Anforderungen an die Funktionen des Systems

3.1.1 Verwaltung von Nutzern

Anlegen eines Accounts

Funktion: Ein Potentieller Nutzer kann einen Account anlegen.

Eingaben: E-Mail Adresse, Name, Passwort

Verarbeitungsschritte:

- Prüfe, ob E-Mail Adresse valide ist (bspw. über eine whitelist), wenn nicht, zeige Fehlermeldung
- Sende einen Bestätigungslink an E-Mail Adresse
- Ohne Bestätigung ist kein Login möglich

Ausgaben: Der Account wird erfolgreich angelegt oder eine Fehlermeldung wird angezeigt.

Anmelden auf der Website

Funktion: Ein Nutzer kann sich auf der Website anmelden.

Voraussetzungen: Der betreffende Nutzer hat einen Account angelegt.

Eingaben: E-Mail Adresse, Passwort

Verarbeitungsschritte:

- Prüfe ob E-Mail Adresse und Passwort valide sind, wenn nicht, zeige Fehlermeldung an

Ausgaben: Der Nutzer hat sich erfolgreich angemeldet oder eine Fehlermeldung wird angezeigt.

Löschen von Accounts anderer Nutzer

Funktion: Inaktive oder unerwünschte Accounts von Nutzern können gelöscht werden.

Voraussetzungen: Der betreffende Nutzer hat die Rolle Administrator

Eingaben: Passwort, zu löschende Nutzer

Verarbeitungsschritte:

- Fordere vor Ausführung des Löschen Eingabe des Administrator Passworts
- Wenn das Passwort falsch ist, zeige eine Fehlermeldung an

Ausgaben: Die Accounts der Nutzer wurden gelöscht oder eine Fehlermeldung wird angezeigt.

Ändern von Accountdaten

Funktion: Der Nutzer kann Accountdaten ändern oder seinen Account löschen

Voraussetzungen: Der betreffende Nutzer hat einen Account angelegt und ist angemeldet.

Eingaben: Name, Passwort, Löschung

Verarbeitungsschritte:

- Fordere erneute Eingabe des alten Passworts um Änderungen zu bestätigen, wenn das Passwort falsch ist, gebe Fehlermeldung aus
- Bei Löschung des Accounts zeige eine Warnung an

Ausgaben: Der Nutzer hat die Accountdaten erfolgreich geändert oder eine Fehlermeldung wird angezeigt.

3.1.2 Verwaltung von Veranstaltungen

Anlegen einer Veranstaltung

Funktion: Ein Administrator oder Veranstaltungsverwalter kann eine neue Veranstaltung erstellen.

Voraussetzungen: Der betreffende Nutzer hat die Rolle Administrator oder Veranstaltungsverwalter.

Eingaben: Name der Veranstaltung, Kennwort zum Beitreten, Gültigkeitszeitraum der Veranstaltung

Verarbeitungsschritte:

- Prüfe ob der Veranstaltungsname verfügbar ist, wenn nicht, zeige Fehlermeldung an
- Prüfe ob Gültigkeitszeitraum bereits abgelaufen ist, wenn nicht, zeige Fehlermeldung an

Ausgaben: Die Veranstaltung wird angelegt oder eine Fehlermeldung wird angezeigt.

Automatisches Löschen einer Veranstaltung

Funktion: Das System kann eine Veranstaltung nach Ablauf des Gültigkeitszeitraums löschen.

Voraussetzungen: Der Gültigkeitszeitraum der Veranstaltung wurde überschritten.

Eingaben: Die Veranstaltung und alle damit zusammenhängenden Dateien.

Verarbeitungsschritte:

- Sende im Voraus eine Warnung an den Ersteller der Veranstaltung
- Lösche alle in der Veranstaltung angelegten Dateien

Ausgaben: Die Veranstaltung wurde gelöscht.

Manuelles Löschen einer Veranstaltung

Funktion: Ein Administrator oder ein Veranstaltungsverwalter kann eine Veranstaltung löschen.

Voraussetzungen: Der Betreffende Nutzer hat die Rolle Administrator oder Veranstaltungsverwalter.

Eingaben: Die zu löschende Veranstaltung und alle damit zusammenhängenden Dateien.

Verarbeitungsschritte:

- Zeige eine Warnung an ob die Veranstaltung gelöscht werden soll
- Wenn ja, lösche alle in der Veranstaltung angelegten Dateien

Ausgaben: Die Veranstaltung wurde gelöscht oder nichts passiert.

Suchen einer Veranstaltung

Funktion: Nutzer können über eine Such-leiste Offene Veranstaltungen suchen.

Voraussetzungen: Der betreffende Nutzer ist angemeldet.

Eingaben: Der Name der Veranstaltung

Verarbeitungsschritte:

- Gebe den Namen oder einen Teil des Namens der Veranstaltung in eine Such-Leiste ein
- Wenn Veranstaltungen mit dem Namen existieren, zeige diese an
- Wenn nicht, zeige eine Fehlermeldung an

Ausgaben: Zu dem Such-Begriff passende Veranstaltungen werden angezeigt oder eine Fehlermeldung wird angezeigt.

Beitreten einer Veranstaltung

Funktion: Nutzer können einer Veranstaltung beitreten.

Voraussetzungen: Der betreffende Nutzer ist angemeldet und kennt das Kennwort der Veranstaltung.

Eingaben: Das Kennwort der Veranstaltung

Verarbeitungsschritte:

- Prüfe ob das Kennwort korrekt ist, wenn nicht, zeige Fehlermeldung an

Ausgaben: Der Nutzer tritt der Veranstaltung bei oder eine Fehlermeldung wird angezeigt.

Vergabe von Rollen in einer Veranstaltung

Funktion: Administratoren oder Veranstaltungsverwalter können Nutzern in einer Veranstaltung Rollen zuweisen.

Voraussetzungen: Der betreffende Nutzer hat die Veranstaltung erstellt und der Nutzer, dem eine Rolle zugewiesen werden soll, ist der Veranstaltung beigetreten.

Eingaben: Nutzer, dem eine Rolle zugewiesen werden soll, Art der Rolle

Ausgaben: Dem Nutzer wird die entsprechende Rolle zugewiesen.

Erstellen von Abgabegruppen bzw. Teams

Funktion: Moderatoren können Abgabegruppen bzw. Teams im Rahmen einer Veranstaltung erstellen.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Größe der Gruppe(n), Name(n) der Gruppe(n), ggf. Anzahl der Gruppen

Ausgaben: Eine neue Gruppe bzw. mehrere neue Gruppen werden angelegt.

Beitreten von Abgabegruppen bzw. Teams

Funktion: Nutzer können Abgabegruppen bzw. Teams im Rahmen einer Veranstaltung beitreten

Voraussetzungen: Der betreffende Nutzer ist der betreffenden Veranstaltung beigetreten.

Eingaben: Auswahl einer Gruppe, der der Nutzer beitreten möchte.

Verarbeitungsschritte:

- Prüfe ob noch freie Plätze in der Gruppe sind, wenn nicht, zeige eine Fehlermeldung an

Ausgaben: Der Nutzer tritt der Gruppe bei oder eine Fehlermeldung wird angezeigt.

3.1.3 Übungen

Erstellen einer Übung

Funktion: Moderatoren können im Rahmen einer Veranstaltung eine Übung erstellen.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Abgabefrist, Veröffentlichungsdatum, Titel der Übung, Beschreibung der Übung, Codegerüst als Jupyter Notebook, ggf. Hilfsskripte in Python, Testdaten, Trainingsdaten, ggf. Validierungsdaten zum Testen der Lösung, ggf. Pythonskript zur Evaluation

Verarbeitungsschritte:

- Prüfe ob Abgabefrist bzw. Veröffentlichungsdatum bereits vergangen sind, wenn ja, zeige Fehlermeldung an
- Prüfe ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus
- Prüfe ob Pflichtfelder (Abgabefrist, Veröffentlichungsdatum, Titel) ausgefüllt sind, wenn nicht, zeige Fehlermeldung an

Ausgaben: Eine Übung wurde in einer Veranstaltung erstellt oder eine Fehlermeldung wird angezeigt.

Bearbeiten der Parameter einer Übung

Funktion: Das Bearbeiten der Parameter einer Übung, um Korrektur von Fehlern o.ä. die bei der Erstellung der Übung aufgetreten sind zu erlauben.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Abgabefrist, Veröffentlichungsdatum, Titel der Übung, Beschreibung der Übung, Codegerüst als Jupyter Notebook, Hilfsskripte in Python, Testdaten, Trainingsdaten, Pythonskript zur Evaluation. *Alle Eingaben sind hier optional*

Verarbeitungsschritte:

- Prüfe ob Abgabefrist bzw. Veröffentlichungsdatum bereits vergangen sind, Wenn ja, zeige Fehlermeldung an
- Prüfe ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus

Ausgaben: Die Parameter einer Übung werden geändert oder eine Fehlermeldung wird angezeigt.

Löschen einer Übung

Funktion: Ein Moderator kann eine erstellte Übung Löschen.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Die zu löschende Übung

Verarbeitungsschritte:

- Zeige eine Warnung zur Bestätigung der Löschung an
- Wenn bestätigt, lösche die Übung und alle damit zusammenhängenden Dateien und Lösungen der Nutzer

Ausgaben: Die Übung wurde gelöscht oder nichts passiert.

Bearbeiten einer Übung

Funktion: Teilnehmer können eine Übungsaufgabe bearbeiten.

Voraussetzungen: Das Veröffentlichungsdatum der Übung wurde erreicht oder überschritten. Die Abgabefrist der Übung wurde nicht überschritten.

Eingaben: Erstellen von Hilfsdateien (*optional*), hochladen von Hilfsdateien (*optional*), Code zur Lösung der Übung (Bearbeitung des Codegerüsts).

Verarbeitungsschritte:

- Prüfe ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus

Ausgaben: Die Übung wurde bearbeitet oder eine Fehlermeldung wird angezeigt.

Eine bearbeitete Übung zur Bewertung einreichen

Funktion: Teilnehmer können eine bearbeitete Übung zur Bewertung einreichen.

Voraussetzungen: Das Veröffentlichungsdatum der Übung wurde erreicht oder überschritten. Die Abgabefrist der Übung wurde nicht überschritten.

Eingaben: Das trainierte Modell und die dazugehörige Implementierung, die bewertet werden soll.

Ausgaben: Die Übung wird zur Bewertung eingereicht.

3.1.4 Hackathon

Erstellen eines Hackathons

Funktion: Moderatoren können im Rahmen einer Veranstaltung einen Hackathon erstellen.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Abgabefrist, Veröffentlichungsdatum, Titel des Hackathons, Beschreibung, Codegerüst, Hilfsskripte in Python, Testdaten, Trainingsdaten, Validierungsdaten zum Testen der Lösung, Pythonskript zur Evaluation

Verarbeitungsschritte:

- Prüfe, ob Abgabefrist bzw. Veröffentlichungsdatum bereits vergangen sind, wenn ja, zeige Fehlermeldung an
- Prüfe, ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus
- Prüfe, ob Pflichtfelder (Abgabefrist, Veröffentlichungsdatum, Titel) ausgefüllt sind, wenn nicht, zeige Fehlermeldung an

Ausgaben: Ein Hackathon wurde in einer Veranstaltung erstellt oder eine Fehlermeldung wird angezeigt.

Bearbeiten der Parameter eines Hackathons

Funktion: Bearbeiten der Parameter eines Hackathons um Korrektur von Fehlern o.ä., die bei der Erstellung der Übung aufgetreten sind, zu erlauben

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Abgabefrist, Veröffentlichungsdatum, Titel der Übung, Beschreibung der Übung, Codegerüst, Hilfsskripte in Python, Testdaten, Trainingsdaten, Pythonskript zur Evaluation. *Alle Eingaben sind hier optional*

Verarbeitungsschritte:

- Prüfe, ob Abgabefrist bzw. Veröffentlichungsdatum bereits vergangen sind, wenn ja, zeige Fehlermeldung an
- Prüfe, ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus.

Ausgaben: Die Parameter eines Hackathons werden geändert oder eine Fehlermeldung wird angezeigt.

Löschen eines Hackathons

Funktion: Ein Moderator kann einen erstellten Hackathon Löschen.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Der zu löschende Hackathon

Verarbeitungsschritte:

- Zeige eine Warnung zur Bestätigung der Löschung an
- Wenn bestätigt, lösche den Hackathon und alle damit zusammenhängenden Dateien und Lösungen der Nutzer

Ausgaben: Der Hackathon wurde gelöscht oder nichts passiert.

Bearbeiten eines Hackathons

Funktion: Teilnehmer können Die Aufgabe des Hackathons bearbeiten.

Voraussetzungen: Das Veröffentlichungsdatum des Hackathons wurde erreicht oder überschritten. Die Abgabefrist wurde nicht überschritten.

Eingaben: Code zur Lösung des Problems (Bearbeitung des Codegerüsts).

Ausgaben: Die Aufgabe des Hackathon wurde bearbeitet.

Eine bearbeitete Lösung des Hackathons zur Bewertung einreichen

Funktion: Teilnehmer können eine bearbeitete Lösung des Hackathons zur Bewertung einreichen.

Voraussetzungen: Das Veröffentlichungsdatum des Hackathons wurde erreicht oder überschritten. Die Abgabefrist wurde nicht überschritten.
item[] **Eingaben:** Textdatei mit Ergebnissen

Ausgaben: Die Lösung wird zur Bewertung eingereicht.

Zugriff auf Working Directories der Teilnehmer

Funktion: Ein Moderator kann auf die Working Directories aller Teams Zugreifen und die Lösungen bearbeiten.

Vorraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der Veranstaltung.

Videokonferenztool *optional*

Funktion: Ein Videokonferenztool kann von den Teilnehmern verwendet werden. Es können ebenfalls Untergruppen für die Teams erstellt werden.

Vorraussetzungen: Der betreffende Nutzer hat ein Mikrofon und ggf. eine Webcam.

3.1.5 Competition

Erstellen einer Competition

Funktion: Moderatoren können im Rahmen einer Veranstaltung eine Competition erstellen

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Anfangsdatum, Rundenanzahl, Länge einer Runde, Titel der Competition, Beschreibung der Competition, Codegerüst als Jupyter Notebook, Hilfsskripte in Python, Testdaten, Trainingsdaten, Validierungsdaten zum Testen der Lösung, Pythonskript zur Evaluation der Lösungen.

Verarbeitungsschritte:

- Prüfe, ob Anfangsdatum bereits vergangen ist, wenn ja, zeige Fehlermeldung an
- Prüfe, ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus
- Prüfe, ob die Pflichtfelder (Anfangsdatum, Rundenanzahl, Länge einer Runde) ausgefüllt sind, wenn nicht, gebe Fehlermeldung aus

Ausgaben: Eine Competition wird in der Veranstaltung erstellt oder eine Fehlermeldung wird angezeigt.

Bearbeiten der Parameter einer Competition

Funktion: Bearbeiten der Parameter einer Competition, um Korrektur von Fehlern o.ä., die bei der Erstellung der Competition aufgetreten sind, zu erlauben.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Anfangsdatum, Rundenanzahl, Länge einer Runde, Beschreibung der Competition, Codegerüst als Jupyter Notebook, Hilfsskripte in Python, Testdaten, Trainingsdaten, Validierungsdaten zum Testen der Lösung, Pythonskript zur Evaluation der Lösungen. *Alle Eingaben sind hier optional*

Verarbeitungsschritte:

- Prüfe, ob Anfangsdatum bereits vergangen ist, wenn ja, zeige Fehlermeldung an
- Prüfe, ob die hochgeladenen Dateien das richtige Format haben, wenn nicht, gebe entsprechende Fehlermeldung aus

Ausgaben: Die Einstellungen bzw. Parameter der Competition werden geändert oder eine Fehlermeldung wird angezeigt.

Löschen einer Competition

Funktion: Ein Moderator kann eine erstellte Competition Löschen.

Voraussetzungen: Der betreffende Nutzer hat mindestens die Rolle Moderator in der betreffenden Veranstaltung.

Eingaben: Die zu löschende Competition

Verarbeitungsschritte:

- Zeige eine Warnung zur Bestätigung der Löschung an
- Wenn bestätigt, lösche die Competition und alle damit zusammenhängenden Dateien und Lösungen der Nutzer

Ausgaben: Die Competition wurde gelöscht oder nichts passiert.

Bearbeiten der Lösung einer Runde der Competition

Funktion: Das Bearbeiten einer Lösung eines Problems in einer Runde der Competition.

Voraussetzungen: Die Competition befindet sich in einer Runde.

Eingaben: Erstellen von Hilfsdateien (*optional*), Code zur Lösung des Problems (Bearbeitung des Codegerüsts).

Ausgaben: Die Übung wurde bearbeitet.

Ausführung eines Angriffs in einer Runde der Competition

Funktion: Das Ausführen eines Angriffs in einer Runde der Competition.

Voraussetzungen: Ein trainiertes Modell einer anderen Gruppe existiert. Die Competition befindet sich mindestens in der zweiten Runde.

Eingaben: Angriffsdatensatz (Teilmenge der Trainingsdaten)

Verarbeitungsschritte:

- Das Modell wird mit dem Angriffsdatensatz ausgeführt und mit dem Evaluierungsskript bewertet.
- Wenn der ausgeführte Angriff besser war, als vorherige Angriffe des Teams auf das gleiche Modell, werden entsprechend einer Metrik dem Angegriffenen Team Punkte abgezogen.

Ausgaben: Die Bewertung des Angriffes und ggf. die abgezogenen Punkte des Angegriffenen Teams.

Anzeigen eines Scoreboards während der Competition

Funktion: Während der Competition soll ein Scoreboard mit den Punkten der verschiedenen Teams angezeigt werden.

Eingaben: Die Punkte der verschiedenen Teams.

Ausgaben: Eine Übersicht des Punktestandes der verschiedenen Teams.

3.1.6 Allgemeines zum Bearbeiten von Lösungen

Daten und Dokumente für die Teilnehmer

Funktion: Die zur Bearbeitung einer Lösung benötigten Daten und Dokumente werden den Teilnehmern in einem Working Directory zur Verfügung gestellt.

Seperate Freigabe der Test-, Trainings- und Validierungsdaten

Funktion: Die Test-, Trainings- und Validierungsdaten sollen getrennt voneinander verfügbar gemacht werden.

Cooperative Bearbeitung der Lösung in einem Team

Funktion: Ein Team arbeitet auf auf dem gleichen Working Directory. Mehrere Teammitglieder können das gleiche Dokument gleichzeitig bearbeiten. Es kann nicht auf die Working Directories anderer Teams zugegriffen werden.

3.1.7 Training und Ausführung von Modellen

Erstellen eines Trainingsauftrags

Funktion: Der Nutzer kann einen Trainingsauftrag für ein programmiertes Modell an den Server stellen.

Eingaben: das zu trainierende Modell.

Verarbeitungsschritte:

- Der Trainingsauftrag wird an den Server gestellt
- Falls ein Evaluationsskript vorhanden ist, wird das trainierte Modell bewertet

Ausgaben: Das trainierte Modell oder eine Fehlermeldung falls das Training fehlgeschlagen ist. Ggf. wird die Bewertung(z.B. in Form der Genauigkeit) angezeigt.

Übersicht über den Trainingsfortschritt und Fehlermeldungen in Echtzeit

Funktion: Der Trainingsfortschritt eines Modells und dabei auftretende Fehlermeldungen können in Echtzeit verfolgt werden.

Voraussetzungen: Ein Modell befindet sich im Training.

Eingaben: Daten zum Trainingsfortschritt vom Server, Fehlermeldungen.

Verarbeitungsschritte:

- Aufbereitung der Daten und Fehlermeldungen vom Server

Ausgaben: Der Aufbereitete Trainingsfortschritt des Modells und bzw. oder aufgetretene Fehlermeldungen.

Ausführen eines Trainierten Modells

Funktion: Ein trainiertes Modell kann auf einem Datensatz ausgeführt werden.

Voraussetzungen: Ein trainiertes Modell und ein Datensatz (z.B. Testdaten) ist verfügbar ist verfügbar.

Eingaben: Das trainierte Modell, ein Datensatz

Verarbeitungsschritte:

- Das trainierte Modell wird auf dem Datensatz ausgeführt

Ausgaben: Die Klassifizierungen des Modells.

3.1.8 Bewertung und Evaluation**Bewerten eines Trainierten Modells**

Funktion: Das automatische Bewerten eines trainierten Modells.

Voraussetzungen: Ein Evaluationsskript und Datensatz ist für die Übung bzw. Competition verfügbar.

Eingaben: Das trainierte Modell, ein Evaluationsskript, ein Evaluationsdatensatz.

Verarbeitungsschritte:

- Das trainierte Modell wird mit dem Evaluationsskript auf den Evaluationsdaten ausgeführt

Ausgaben: Die Bewertung des Modells.

Vergabe von Punkten in einer Competition

Funktion: Einem Team in einer Competition werden Punkte angerechnet oder abgezogen.

Voraussetzungen: Ein Angriff wurde auf ein Modell des Teams ausgeführt oder ein trainiertes Modell des Teams wurde bewertet.

Eingaben: Die Bewertung eines Angriffes oder des Trainings.

Verarbeitungsschritte:

- Dem Team werden entsprechend der Bewertung und basierend auf einem Punktevergabesystem Punkte angerechnet oder Abgezogen

Ausgaben: Der neue Punktestand des Teams.

Die zur Bewertung eingereichten Lösungen anzeigen

Funktion: Die zur Bewertung eingereichten Lösungen einer Übung oder eines Hackathons anzeigen.

Voraussetzungen: Der Nutzer hat mindestens die Rolle Moderator in der Veranstaltung.

Eingaben: Die eingereichten Lösungen der Teilnehmer.

Verarbeitungsschritte:

- Prüfe ob mindestens eine Lösung eingereicht wurde, wenn nicht, zeige Fehlermeldung an

Ausgaben: Liste mit Übersicht über die eingereichten Lösungen oder eine Fehlermeldung wird angezeigt.

3.1.9 Hoch- und Runter-laden von Dateien**Herunterladen von Dateien einer Übung**

Funktion: Nutzer können die zur Bearbeitung bereitgestellten Dateien einer Übung und die eigens bearbeiteten Dateien einer Übung herunterladen.

Eingaben: Die Dateien der Übung.

Verarbeitungsschritte:

- Die Dateien werden als .zip Archiv zusammengefasst.

Ausgaben: Die Dateien als .zip Archiv Download.

Herunterladen des trainierten Modells

Funktion: Nutzer können das trainierte Modell herunterladen.

Voraussetzungen: Ein Modell wurde von einem Nutzer der Gruppe bzw. des Teams trainiert.

Eingaben: Das trainierte Modell.

Ausgaben: Das trainierte Modell als Download.

Herunterladen der eingereichten Lösungen

Funktion: Die zur Bewertung eingereichten Lösungen einer Übung können heruntergeladen werden.

Voraussetzungen: Der Nutzer hat mindestens die Rolle Moderator in der Veranstaltung.

Eingaben: Die Lösung die Heruntergeladen werden soll.

Verarbeitungsschritte:

- Die Dateien werden als .zip Archiv zusammengefasst.

Ausgaben: Die Dateien als .zip Archiv Download.

Hochladen eines bereits trainierten Modells

Funktion: Ein Bereits trainiertes Modell soll im Pytorch Format hochgeladen werden können.

Voraussetzungen: Der Nutzer hat mindestens die Rolle Moderator in der Veranstaltung.

Eingaben: Das trainierte Modell.

Verarbeitungsschritte:

- Prüfe, ob das Modell das richtige Format hat, wenn nicht, zeige Fehlermeldung an.

Ausgaben: Das Modell wird hochgeladen oder eine Fehlermeldung wird angezeigt.

3.2 Nicht-funktionale Anforderungen

3.2.1 Sicherheit

Trennung von Training und Evaluation: Um die Modelle der Teams in einer Competition vor ungewollten Angriffen zu schützen, soll jeder Trainingsauftrag einen eigenen Docker Container generieren, in dem das Training stattfindet. Die Evaluation des trainierten Modells soll in einer separaten, vertraulichen Docker Instanz stattfinden.

Zugriff außerhalb des Uni-netzes: Es soll Nutzern möglich sein das System von Rechnern außerhalb des Uni-Netzes zu verwenden.

Datenschutz: Da Accounts, Passwörter und E-Mail Adressen verwaltet werden, müssen die in Deutschland geltenden Bestimmungen zum Schutz von Nutzerdaten eingehalten werden.

3.2.2 Technik

Hardware: Sowohl der DGX Rechner als auch die Dell Rechner sollen zum Training genutzt werden können.

Speicherbedarf: Für einen Trainings bzw. Testdatensatz müssen mindestens 1 GB freier Speicher zur Verfügung stehen.

Software: Zum bearbeiten bzw. erstellen von Lösungen sollen Jupyter Notebooks und der Jupyter Editor verwendet werden. *Bei dem Hackathon noch unklar*

Software auf dem Server: Es soll ein Linux Server mit Docker und einer Python Umgebung verwendet werden.

Python Pakete: Die Python Pakete numpy, sklearn, torch, torchvision, torchsummary, tensorflow, tensorboard sollen installiert sein.

Wrapper zur Simulation von Tensorflow auf dem Rechner des Nutzers: Mittels einer Abstrakten Ebene soll ein Wrapper implementiert werden, der Tensorflow auf dem Rechner des Nutzers simuliert und es somit erlaubt Tensorflow Befehle direkt zu verwenden.

3.2.3 Kompatibilität

Browser: Das System soll von allen herkömmlichen Browser Anwendungen benutzbar sein.

3.2.4 Ergonomie

Sprachausführung: Die Benutzerführung erfolgt auf deutsch

Format der Downloads: Das System soll alle Downloads als .zip Datei zur Verfügung stellen

References

- [1] Ieee standard 29148:2018. <https://standards.ieee.org/standard/29148-2018.html>. Accessed: 2020-06-10.
- [2] kaggle.com. <http://www.kaggle.com>. Accessed: 2020-06-05.
- [3] moodle.com. <https://moodle.com/>. Accessed: 2020-06-07.