




WISE 2023

DOKUMENTATION
STUDENTS4STUDENTS

MARC ISSMER, DANIEL DESGRONTE, DANIEL KÜNKEL, AMANDA DE MOURA



Inhalt

1	Einleitung	3
1.1	Problemstellung	3
1.2	Ziele	3
2	Lösungsstrategie	4
2.1	Codekonventionen	4
2.1.1	Typescript Konventionen	4
2.1.2	Python Konventionen	4
2.2	Infrastruktur	5
2.2.1	Zugriffskategorien	7
2.3	Kryptografie	7
3	Bausteinsicht	10
4	Laufzeitsicht	14
5	Risiken und technische Schulden	14
5.1	Safety, Security, Privacy	14
6	Fazit	17
7	Quellen	19

Abbildungsverzeichnis

Abbildung 1: Überblick der ersten 3 Ebenen der Bausteinsicht	11
Abbildung 2: ER-Diagramm der Datenbank	12

URL des git-Repos

<https://github.com/dhbw-stuttgart-webengineering/programmentwurf-gruppe-2>

1 Einleitung

Die Plattform „Students4Students – DHBW-Edition“ wurde entwickelt, um die Studierendengemeinschaft an der DHBW Stuttgart zu unterstützen und stärken. Als User besteht die Möglichkeit sich über Anzeigen mit dem gesamten Netzwerk auszutauschen und auf Beiträge zu reagieren. Durch die Exklusivität des Netzwerkes, indem nur User, die eine gültige E-Mail-Adresse der DHBW Stuttgart besitzen, möchte „Students4Students“ mit diesem Ansatz eine Community fördern, die sich unterstützt und das anonyme Studierendenleben an der DHBW aufbrechen. Dieses Ziel wird verfolgt, indem die Kernwerte der Plattform auf dem Prinzip des Austauschs und der gegenseitigen Unterstützung basieren. In dieser Einleitung wird ein Überblick über das Projekt gegeben, die Problemstellung erläutert und die Ziele der Plattform vorgestellt.

1.1 Problemstellung

Die Studienzeit ist oft geprägt von finanziellen Engpässen, Stress und einem Mangel an Ressourcen. Hinzu kommt eine Besonderheit für DHBW-Studierende, nämlich die Anonymität. Diese Probleme haben eine starke Auswirkung auf jeden Studierenden. Dafür fehlt eine effiziente und zielgerichtete Plattform, die es den Studierenden an der DHBW Stuttgart ermöglicht, sich untereinander zu helfen und das Miteinander in den Fokus zu stellen.

1.2 Ziele

Die Schaffung einer Kommunikationsplattform für DHBW Stuttgart Studierende verfolgt vier Hauptziele:

1. Es soll eine lebendige und sich gegenseitig unterstützende Community geschaffen werden, die für den gemeinsamen Erfolg zusammenarbeitet. Durch den Austausch von Wissen, Fähigkeiten und Ressourcen kann somit jeder Einzelne profitieren.
2. Es soll eine bessere Allokation von Ressourcen zwischen den Studierenden stattfinden, indem untereinander getauscht wird. Die Idee ist, die vorhandenen Ressourcen in der Studierendengemeinschaft besser zu nutzen, um den finanziellen Druck, den viele Studierende erleben, zu mindern. Es soll also das Tauschen und die gegenseitige Hilfe gefördert werden.
3. Eine Förderung der Vernetzung der Studierenden untereinander. Es bestehen begrenzt Möglichkeiten für Studierende, andere, selbst im selben Jahrgang, kennen zu lernen. Über „Students4Students“ sollen soziale Kontakte innerhalb der DHBW gefördert werden.
4. Die Sicherheit und der Datenschutz der Anwendung steht im Fokus, um die Daten der Nutzer zu schützen und Unbefugten jeglichen Zugriff zu verhindern.

Zusammengefasst wird angestrebt, „Students4Students“ zu einer wertvollen Ressource für die Studierenden der DHBW Stuttgart zu machen. Es soll die Möglichkeit geboten werden, das akademische Leben und persönlichen Erfahrungen der Studierenden zu bereichern, indem eine Plattform geschaffen wird, die auf Gemeinschaft, Austausch und Unterstützung basiert.

2 Lösungsstrategie

2.1 Codekonventionen

Für jede Programmiersprache werden Formatregeln, sogenannte Codekonventionen, gesammelt, um einen einheitlichen Programmierstil beizubehalten. Dadurch entsteht ein gut strukturierter und lesbarer Code. Dieses Projekt nutzt im Frontend Typescript und im Backend Python (PEP 8). Diese werden zunächst beschrieben und die projektrelevanten Konventionen näher betrachtet.

2.1.1 TypeScript Konventionen

Variablennamen sollen bedeutsam sein, also eindeutig und kontextbezogen, sowie aussprechbar und in CamelCase-Schreibweise. Hierbei sollen diese auch explizit und auf das Nötigste beschränkt sein. Braucht eine Variable einen Kommentar, ist es kein guter Variablenname. (vgl. Anchiti, 2023)

CamelCase (bsp. hausOhneDach) soll zusätzlich für Methoden und Methodenparameter genutzt werden, PascalCase (bsp. HausMitDach) für Klassen- und Interfacenamen. Booleans sollen nicht negativ benannt werden und ein Präfix wie is, are oder has nutzen, wodurch der Datentyp eindeutig für andere Entwickler hervorgehoben wird. Bezüglich der Klammern gilt das Gebot OTBS (one true brace style), also, dass die zusammengehörigen Klammern auf derselben vertikalen Ebene stehen.

Kommentare sollen nur wenn nötig auftreten und im Normalfall sollte der Code für sich selbst sprechen. Wenn ein Kommentar geschrieben wird, dann wird dies auf Englisch gemacht und mit einem Leerzeichen vor dem Text. Offensichtlich sollen keine leeren Kommentare existieren. Diese erfolgen dann immer vor Methoden, d.h. außerhalb geschweiften Klammern.

2.1.2 Python Konventionen

„Konsistenz mit dem Style Guide ist wichtig. Konsistenz innerhalb eines Projekts ist wichtiger. Konsistenz innerhalb eines Moduls oder einer Funktion ist am wichtigsten.“ (vgl. van Rossum, 2001)

Zunächst wird Fokus auf das Code Layout gelegt. Hier sollen Argumente oder Parameter, die in die nächste Zeile gelangen, einen Unterschied von 4 Leerzeichen zu dem Rest haben. Des Weiteren sollen Argumente immer in einer vertikalen Ausrichtung mit den übergeordneten Argumenten stehen.

Top-level Funktionen und Klassendefinitionen sollen mit zwei leeren Zeilen umgeben sein, Methodendefinitionen innerhalb einer Klasse mit einer. Zusätzlich sollen alle Identifier ASCII-only Identifier nutzen und auf Englisch formuliert sein. Imports sind auf separate Zeilen zu initiieren. Des Weiteren sind single- und double-quoted Strings in Python gleichgestellt.

Kommentare bestehen in PEP8 aus ganzen englischen Sätzen, wobei das erste Wort großgeschrieben wird (außer es ist ein Identifier). Docstrings werden nicht weiter berücksichtigt.

Pakete und Module sollen kurze, kleingeschriebene Namen haben. Unterstriche können bei Modulen für erhöhte Lesbarkeit genutzt werden, bei Paketen wird das nicht empfohlen. Klassennamen sollen in CapWords (wie PascalCase) geschrieben werden. Konstanten werden in Großbuchstaben geschrieben (bsp. TOTAL).

2.2 Infrastruktur

Realisiert wird das Projekt „Students4Students“ durch eine logische Dreiteilung in Frontend, Backend und Datenbank. Das Frontend wurde mittels React.js, das Backend mittels Django und die Datenbank mittels MySQL realisiert. Im Folgenden werden die einzelnen Komponenten beschrieben und die Zusammenhänge aufgezeigt.

Django ist ein leistungsstarkes Python Open-Source-Web-Framework. Es stellt eine umfangreiche Sammlung von Tools und Bibliotheken zur Verfügung, um häufige Webentwicklungsaufgaben zu vereinfachen und dadurch die Entwicklungszeit zu kürzen und ein sauberes und pragmatisches Design zu fördern. Django bedient sich bei dem Aufbau von Projekten dem Model-View-Template (MVT), eine leicht abgewandte Version des MVC-Musters. Die Grundidee ist die Trennung von Daten (Model), Präsentation (View) und Anwendungslogik (Controller/Template). Zusätzlich bietet Django ein leistungsfähiges Object-Relational Mapping (ORM) Framework. Dieses vereinfacht die Einbindung von Datenbanken in ein Django-Projekt, indem sog. Model Objects für Datentabellen erstellt werden, deren Attribute den Spalten entsprechen. Allein durch diese Objekte wird dann in der Datenbank die entsprechende Tabelle erstellt und man kann in Python Datenbankoperationen über Objektmethoden ausführen. Weitere für dieses Projekt interessante Features von Django sind eine URL-Verwaltung, eine leistungsstarke Template-Engine, eingebaute Sicherheits- und eine automatisch generierte Admin-Oberfläche. Zudem kommt die Wiederverwendung von Apps, ein REST-Framework, eine Community, Erweiterbarkeit, Skalierbarkeit und Performanceoptimierung.

React.js ist eine Open-Source JavaScript-Bibliothek zur Entwicklung von Benutzeroberflächen für Webanwendungen. Es basiert auf dem Konzept von Komponenten, also die Zerlegung der Benutzeroberfläche in wiederverwendbare, eigenständige Einheiten, die eine bestimmte

Funktion oder Ansicht darstellen. Logisch stehen Komponenten in einem Projekt hierarchisch in Beziehung, um komplexe Benutzeroberflächen zu realisieren. React.js verwendet ein Virtual DOM (Document Object Model) um die Leistung zu optimieren. Es wird eine virtuelle Repräsentation des DOM erstellt, wodurch nur die geänderten Komponenten aktualisiert werden müssen, was zu schnelleren Aktualisierungen und einer verbesserten Benutzeroberflächenleistung beiträgt.

Um das Debuggen und die Verwaltung einer Anwendung zu erleichtern, folgt React.js dem Konzept des Einweg-Datenflusses. Daten fließen also von der übergeordneten Komponente zu den untergeordneten Komponenten. React.js ermöglicht die Verwendung von JSX, eine spezielle Syntax, die HTML-ähnlichen Code direkt in JavaScript schreiben lässt. Weitere Eigenschaften von React.js sind ein Komponentenlebenszyklus, die Reconciliation der DOM-Repräsentation mit der tatsächlichen DOM, Komponentenwiederverwendung, den React-Router, Zustandsverwaltung und eine Community bzw. ein Ökosystem. Somit ist React.js eine leistungsstarke und flexible Wahl für die Entwicklung moderner Webanwendungen mit einer dynamischen Benutzeroberfläche.

MySQL ist ein Relationales Datenbankverwaltungssystem (RDBMS), das für die Speicherung und Verwaltung von Daten in verschiedenen Anwendungen verwendet wird. Das bedeutet, dass das RDBMS Daten in tabellenartigen Strukturen organisiert, die als Tabellen oder Relationen bezeichnet werden. Diese bestehen aus Zeilen und Spalten, wobei jede Zeile einen Datensatz repräsentiert und jede Spalte ein Attribut darstellt. MySQL verwendet Structured Query Language (SQL) als Abfragesprache, um auf die Datenbank zuzugreifen und Daten zu manipulieren. Entwickler können SQL-Befehle verwenden, um Daten abzurufen, einzufügen, zu aktualisieren und zu löschen (CRUD-Befehle). Es werden eine Vielzahl von Datentypen angeboten, darunter Integer, Float, Text, Datum/Zeit und mehr, um die Art der Daten zu spezifizieren, die gespeichert werden können. Weitere Eigenschaften sind die Verwendung von Primärschlüssel und Indizes, die Unterstützung von Transaktionen und das Angebot von Sicherheitsfunktionen. MySQL ist also eine zuverlässige und skalierbare Open-Source-Datenbank, die häufig in Webanwendungen oder in anderen Systemen verwendet wird.

Docker ist eine Open-Source-Plattform zur Containerisierung, die die Bereitstellung und Verwaltung von Anwendungen in leichtgewichtigen, isolierten Containern auf Softwareebene ermöglicht. Container in dieser Instanz sind in sich geschlossene und isolierte Pakete, die eine Anwendung und alle ihre Abhängigkeiten enthält, was die Portabilität und Wiederholbarkeit von Anwendungen verbessert.

Innerhalb des Projektes entsteht folgender Aufbau: Django wird für die Entwicklung des Backends verwendet, in dem die Anwendungslogik, Autorisierung und Geschäftslogik entwickelt wird. React.js wird verwendet, um die Benutzeroberfläche zu erstellen und

verwalten. Es wird das Einweg-Datenflusskonzept verwendet, um Daten aus dem Backend (Django) anzuzeigen und interaktive Benutzeroberflächenkomponenten zu erstellen. Die Kommunikation zwischen Backend und Frontend erfolgt über die RESTful API. Docker wird verwendet, um das Frontend und Backend, als auch die MySQL-Datenbank in isolierten Containern zu verpacken. Dies ermöglicht eine einfache Bereitstellung und Skalierung der Anwendung in verschiedenen Umgebungen, von der Entwicklung bis zur Produktion. Durch diese Konstellation lässt sich das Projekt effizient und modern gestalten.

2.3 Zugriffskategorien

Zugriffskategorien sind Gruppen oder Klassen von Benutzern, die verschiedene Berechtigungen und Zugriffsrechte auf digitale Ressourcen oder Daten innerhalb eines Systems oder einer Plattform haben, um die Sicherheit und Verwaltung dieser Ressourcen zu steuern. Bei Students4Students sind die einzigen User die Studierenden der DHBW Stuttgart.

2.4 Kryptografie

Die Kryptografie bezeichnet die Wissenschaft von der Geheimhaltung von Nachrichten. Die generelle Vorgehensweise hierbei ist die Verschlüsselung von Daten in sog. Ciphertext und die erneute Entschlüsselung. Ciphertext bezeichnet einen verschlüsselten Text, der mithilfe eines Verschlüsselungsalgorithmus namens cipher aus Klartext umgewandelt wird. Die vier Ziele der Kryptographie sind Vertraulichkeit, Datenintegrität, Authentifizierung und Schutz vor Nichtanerkennung. Um diese Ziele zu erreichen, werden verschiedene Verfahren eingesetzt, unter anderen das Symmetrische und Asymmetrische Verfahren. (vgl. Egly 2007)

Schützenswerte Daten beziehen sich im Allgemeinen auf personenbezogenen Daten und deren Verarbeitung. Hinsichtlich der vorliegenden Projektarbeit sind die Datenbereiche gemeint, in denen persönlichen Daten, wie die studentische Mail, der Name und das Passwort, die bei der Erstellung eines Kontos verarbeitet werden. Diese müssen verschlüsselt werden.

Aufgrund der Nichtwertung der Cybersecurity des Projektes wird nur fiktiv ein geeigneter Verschlüsselungsalgorithmus und eine Schlüssellänge gewählt. Um das dritte Ziel der Kryptographie zu erreichen, die Authentifizierung, wäre die Multi-Factor Authentication (MFA) geeignet, um die Benutzerkonten zu schützen. Zusätzlich wird eingeschränkt, auf was jeder Benutzer Zugriff hat, d.h. eine Art Zugriffseinschränkung wird implementiert. Für Daten in Transit werden geeignete HTTPS Protokolle implementiert, sodass versichert werden kann, dass der Datenstrom zwischen Nutzer und Plattform verschlüsselt bleibt. Hierzu wird ein ordentliches SSL/TLS Zertifikat für das Domain genutzt. Kennwörter werden nicht im plaintext gespeichert, sondern ein starkes Passwort-Hashing Algorithmus wird verwendet. Für sensible Daten, die auf dem Server gespeichert bleiben, ist das Advanced Encryption Standard (AES) geeignet. Hierfür reicht das AES-256, ein symmetrisches Verschlüsselungsverfahren. Sie sind

hat formatiert: Schriftart: (Standard) Arial

in der Regel effizienter und schneller als asymmetrische Verschlüsselungsalgorithmen, wenn es sich um die Verschlüsselung großer Datenmengen handelt. AES-256 hat eine Länge von 256 Bits und wird als sehr stark und für die meisten Anwendungen ausreichend angesehen.

Benutzerauthentifizierungstoken und Sitzungen sind sichergestellt und nicht anfällig für häufige Angriffe wie Session-Fixierung, Cross-Site-Scripting (XSS) oder Cross-Site-Request-Forgery (CSRF). Die Verschlüsselungsbibliotheken und -abhängigkeiten werden auf dem neuesten Stand gehalten, da Sicherheitslücken oft in neuen Versionen behoben werden. Zudem wird in regelmäßigen Abständen Sicherheitsaudits und Penetrationstests durchgeführt, um eventuelle Schwachstellen zu identifizieren und vorzubeugen. Des Weiteren werden die Datenschutzvorschriften der DSGVO eingehalten.

2.5 Secure Product Design

Durch das Secure Product Design wird sichergestellt, dass alle Produkte die vorgegebenen Sicherheitsanforderungen erfüllen oder übertreffen und dass alle Sicherheitsentscheidungen zu einem erhöhten Sicherheitsniveau für das zu entwickelnde Produkt führen. Damit das erfolgen kann, werden vier Security Principles implementiert: die Principles of Least Privilege and Separation of Duties, of Defense-in-Depth, of Zero Trust und of Security-in-the-Open. Diese werden im Folgenden kompakt zusammengefasst.

1. The Principles of Least Privilege and Separation of Duties

Benutzer sollten nur die nötigsten Zugriffsrechte haben, um ihre Aufgaben zu erfüllen. Dadurch wird unbefugter Zugriff verhindert und dies reduziert das Risiko unbefugten Zugriffs auf sensible Daten oder Systeme. Um Betrug und Fehler zu minimieren, sollten verschiedene Personen unterschiedliche Aufgaben in Transaktionen übernehmen. Es gewährleistet, dass keine einzelne Person die volle Kontrolle über alle Aspekte einer Transaktion hat.

2. The Principle of Defense-in-Depth

Sicherheit sollte auf mehreren Ebenen (physisch, Netzwerk, Anwendung, Daten) implementiert werden, um Angriffe abzuwehren.

3. The Principle of Zero Trust

Alle Benutzer und Geräte werden als nicht vertrauenswürdig behandelt und müssen sich authentifizieren, um auf sensible Daten zuzugreifen. Somit können nur autorisierte Benutzer Zugriff gewährt werden.

4. The Principle of Security-in-the-Open

Entwickler in der Open-Source-Softwareentwicklung sollten sich der Sicherheit ihres Codes bewusst sein und Sicherheitspraktiken anwenden. Zusammenarbeit mit Sicherheitsexperten ist wichtig. Dies beinhaltet die Verwendung sicherer

Codierungspraktiken, die Suche nach Sicherheitslücken und die Verwendung sicherer Entwicklungstools.

Wichtig für diese Prinzipien sind bestimmte Bereiche, sog. Security Focus Areas. Diese beziehen sich auf Kontext, Komponenten, Verbindungen, Code und Konfiguration.

1. Kontext

Die Sicherheitsaspekte im Zusammenhang mit dieser Anwendung umfassen die Eingliederung in die Organisation, die beteiligten Abteilungen und deren Gründe für die Nutzung. Dies schließt die Art der enthaltenen Daten und das damit verbundene Risikoprofil ein. Die Schaffung des Sicherheitskontexts beinhaltet die Bedrohungsmodellierung, die die Integration sicherheitsrelevanter Elemente während des Produktentwurfs bei jeder Produktlieferung beinhaltet. Ebenso umfasst sie die Durchführung eines Business Impact Assessments, um die richtigen Sicherheitsstufen für ein bestimmtes Produkt während der Produktentstehung festzulegen.

2. Komponenten

Dieser Bereich betrachtet die Bibliotheken und externen Dienste, die in der Anwendung verwendet werden, und wie sie sicher gehalten werden. Hierbei werden sichere Designmuster und einsatzbereite Komponenten aus der Golden Path / Paved Road-Dokumentation genutzt und die Auswahl dieser Komponenten analysiert. Es werden auch kommerzielle Aspekte wie Lizenzierung und Wartung berücksichtigt.

3. Verbindungen

Es wird untersucht, wie die Anwendung mit anderen Komponenten und Diensten interagiert und wo die Daten gespeichert sind und wie auf sie zugegriffen wird. Hierbei werden auch mögliche Segregationen von Daten oder Umgebungen je nach den Anforderungen an die Produktsicherheitsstufen berücksichtigt.

4. Code

Sicherheitsaspekte im Code umfassen Input-Validierung, Fehlerbehandlung, Authentifizierung und Autorisierung, Kryptographie, das Prinzip der Mindestprivilegierung, sicheres Speichermanagement, Vermeidung von fest codierten Geheimnissen, Sicherheitstests, regelmäßige Code-Audits und Aktualisierung des Codes gemäß den besten Sicherheitspraktiken.

5. Konfiguration

Die sichere Konfiguration der Anwendung ist entscheidend und sollte das Prinzip der Mindestprivilegierung, Verteidigung in der Tiefe, Sicherheit durch Voreinstellung, den Schutz sensibler Daten und die Möglichkeit eines sicheren Ausfalls berücksichtigen.

Es sollte sichere Kommunikationsprotokolle verwenden und regelmäßige Updates sowie Sicherheitsincident-Response-Pläne einschließen, um auf Sicherheitsvorfälle angemessen reagieren zu können.

2.6 CSS-Framework

Ein CSS-Framework ist eine Bibliothek, die ein einfaches, standardkonformes Webdesign unter Verwendung von CSS(-Klassen) ermöglicht. Diese Bibliotheken bieten verschiedene Module und Werkzeuge, die den Umgang mit CSS vereinfachen. Im vorliegenden Projekt werden Bootstrap und Tailwind genutzt, um einfach standardisierte Komponentenstyles zu nutzen und den CSS-Code selbst zu schreiben.

3 Bausteinsicht

3.1 Scope and Content

Die technische Realisierung des Projektes ist der Abbildung 1 zu entnehmen. Die Datenbank, das Backend, sowie das Frontend werden über Docker Container ausführbar. Dabei hat die logische Trennung der Komponenten in eigenständige Container den Sinn eine spätere Einführung des Systems auf eigenständigen Servern zu simulieren. Auf einer höheren Abstraktionsebene lassen sich zwei Blöcke bilden in Datenbank als eigenständiges Modul, sowie ein Modul bestehend aus Backend und Frontend, in Abbildung 1 als Students4Students beschrieben. Es bestehen zwei Zugriffs Kategorien auf das Modul Students4Students, als User und als Admin.

Die Level-1-Betrachtung zeigt das Modul Students4Students als Whitebox S4S (s. Abb. 1). Das Frontend realisiert über einen mit dem backend verbundenen Docker-Container, besitzt ein Paket namens Index. Das Backend nutzt ein Paket, namens Index. Die Level-2-Betrachtung ermöglicht nun eine genaue Betrachtung dieser beiden Elemente als Querschnitt. Diese beinhaltet eine Main, welche die Schnittstelle zu React.js bildet. In der src-File sind die CSS, Pages, Setup, Logic, App, Layout, Assets, App Test und Component Klassen hinterlegt. Die Backend Communication Test Klasse kommuniziert mit dem Backend und prüft die Verbindung zwischen Backend und Frontend.

Die Backend Whitebox beinhaltet Models namens Ad, Favorites, User Authentication, Degree, Category und Faculty. Das ER-Diagramm (siehe Abb. 2) der vorliegenden Datenbank wird im Folgenden erläutert.

Die UserAuthentication sendet die Informationen an den User, um die Validierung des Benutzers zu garantieren. Der User bezieht Informationen von Ad und Degree, um den User fertigzustellen. Category sendet seine Daten an Ad und Faculty an Degree. Somit wird das

hat formatiert: Schriftart: (Standard) Arial, Schriftfarbe: Text 1

Formatiert: Block, Zeilenabstand: 1,5 Zeilen

Benutzerkonto fertiggestellt. Zusätzlich stehen Ad und User in konstanter Rücksprache zueinander.

Die Datenbank als zweites Modul, beinhaltet verschiedene Tables die in Beziehung zueinanderstehen, wie aus Abbildung 2 zu entnehmen. Das ER-Diagramm (Abb. 2) zeigt ebenfalls die verschiedenen Parameter der einzelnen Tables.

Die technische Realisierung des Projektes ist der Abbildung 1 zu entnehmen.

In der Abbildung ist der Zugriff von jeweils Admin und User aufgeführt, sowie Django, Docker und React. In der nächsten Ebene, Level 1, wird verdeutlicht, dass das Frontend mit Docker und React kommuniziert, sowie ein Paket namens Index besitzt. Das Backend kommuniziert mit Django und besitzt ein Paket mit dem Namen Components.

Auf Level 2 sind jeweils die Whiteboxes des Front- und Backends als Querschnitt dargestellt. Das zentrale Element des Frontends ist die src-File. Diese beinhaltet eine Main, welche die Schnittstelle zu React.js bildet. In der src-File sind die CSS, Pages, Setup, Logic, App, Layout, Assets, App Test und Component Klassen hinterlegt. Die Backend Communication Test Klasse kommuniziert mit dem Backend und prüft die Verbindung zwischen Backend und Frontend.

Die Backend Whitebox beinhaltet Models namens Ad, Favorites, User Authentication, Degree, Category und Faculty. Das ER-Diagramm (siehe Abb. 2) der vorliegenden Datenbank wird im Folgenden erläutert.

Die UserAuthentication sendet die Informationen an den User, um die Validierung des Benutzers zu garantieren. Der User bezieht Informationen von Ad und Degree, um den User fertigzustellen. Category sendet seine Daten an Ad und Faculty an Degree. Somit wird das Benutzerkonto fertiggestellt. Zusätzlich stehen Ad und User in konstanter Rücksprache zueinander.

hat formatiert: Schriftart: (Standard) Arial, 11 Pt.

hat formatiert: Schriftart: 11 Pt.

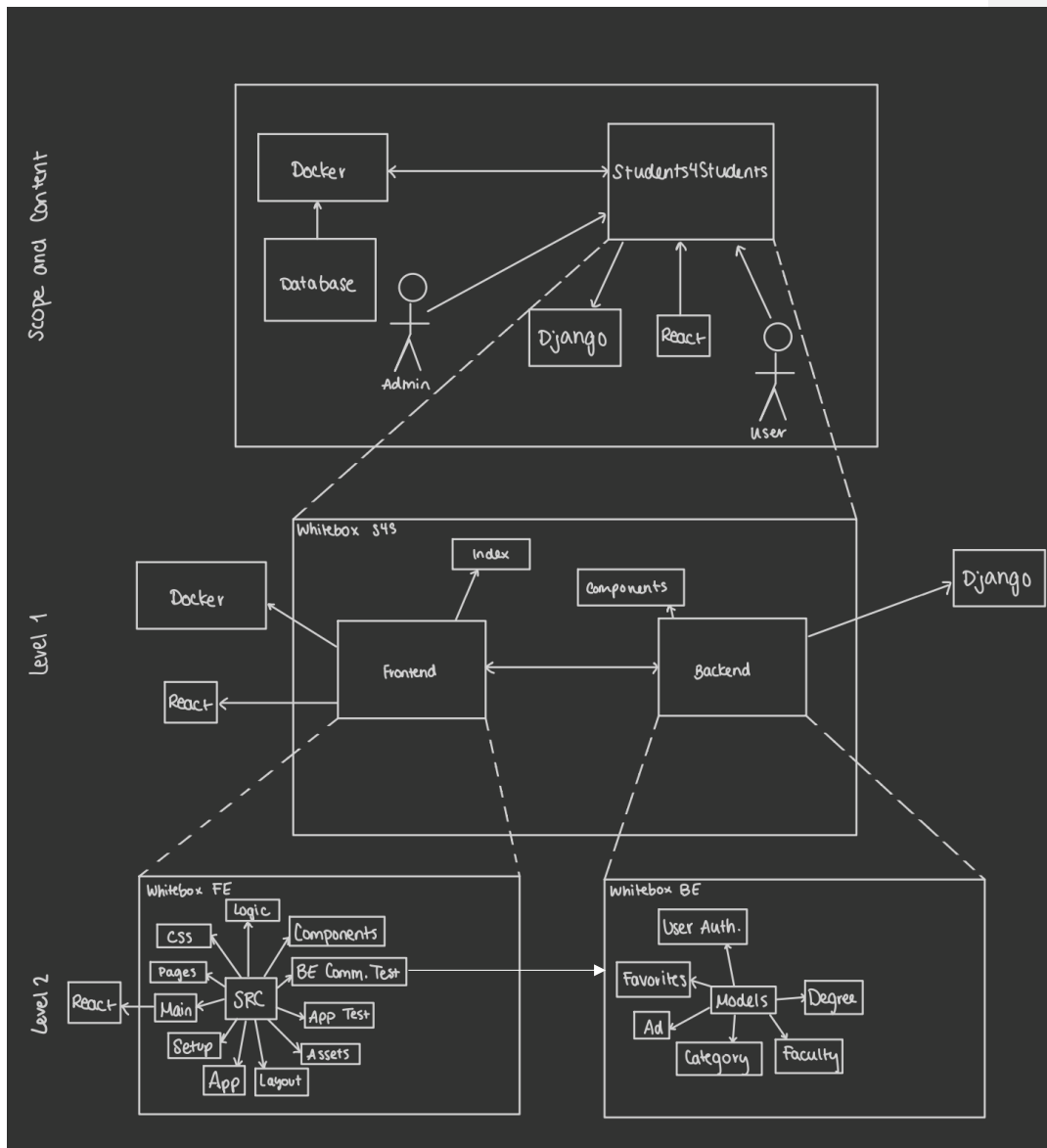


Abbildung 1: Überblick der ersten 3 Ebenen der Bausteinsicht

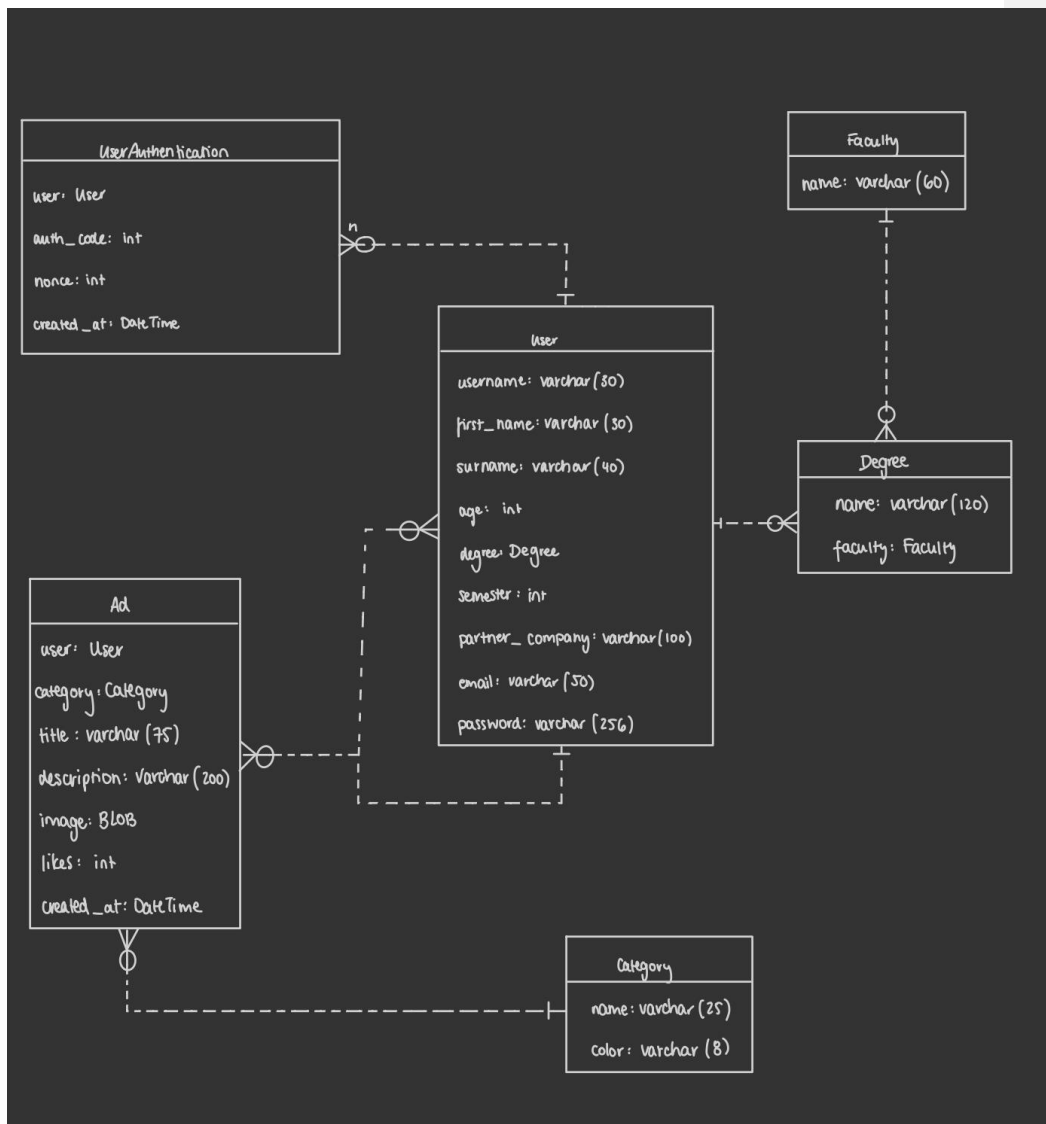


Abbildung 2: ER-Diagramm der Datenbank

4 Laufzeitsicht

Eine Laufzeitsicht impliziert eine Art Algorithmus, der zur Laufzeit eingesetzt wird. Das Projekt wurde ohne den Einsatz eines solchen Algorithmus realisiert. Die Funktionalitäten werden durch die Interaktion von mehreren kleineren Komponenten, die immer kleine Algorithmen ausführen, welche im größten Teil in Frameworks und Paketen übernommen werden. Somit ergibt es wenig Sinn, eine Laufzeitsicht zu konzipieren.

hat formatiert: Schriftfarbe: Text 1

5 Risiken und technische Schulden

5.1 Safety, Security, Privacy

Safety, Security und Privacy werden oft als Synonyme verstanden, obwohl diese Begriffe verschiedene Thematiken behandeln. Im Folgenden werden die Begriffe definiert und im Kontext des Projektes angewandt.

Safety meint den Schutz vor einer Fehlfunktion, die durch einen Fehler des Entwicklers entstanden ist. Dieser Fehler kann zur Bedrohung von Menschen oder auch anderen Maschinen führen. Security befasst sich mit dem Schutz von Software oder Anlagen gegen Angriffe auf die Vertraulichkeit, Integrität und Verfügbarkeit von Informationen. Letztlich bezieht sich Privacy auf den Schutz von Daten, die in einem System erstellt werden und die Verhinderung von unbefugtem Zugriff auf zB. personenbezogene Daten.

Um diese Sicherheitsthematiken angemessen in einem Projekt zu behandeln, existieren bereits lösungsorientierte Methoden. Bei Safety-Themen muss die gegebene IST-Funktionalität mit der spezifizierten SOLL-Funktionalität verglichen werden, das bedeutet, dass der gewünschte Zustand jeder Funktionalität spezifiziert und festgehalten werden muss. Hier werden z.B. Test-Driven Development, Testautomatisierung und Pareto-Prinzipien im Testen genutzt.

Bezüglich Security müssen die Schutzziele vorher definiert sein. Methoden wie Functional Testing, Vulnerability Scanning, Systematic Fuzzing, Penetration Testing, CVE, CAPEC und CWE sind nützlich, um das Testen der Sicherheit zu erweitern, sowie bestehende Risikomanagementsysteme für Softwaresysteme.

Bei Privacy wird durch die Verwendung von standardisierten Designentscheidungen und Techniken (Authentifizierung, Zero-Knowledge-Beweis) korrekte Anonymisierung und Pseudonymisierung gewährleistet und somit die Software geschützt.

5.2 CVE

CVE steht für "Common Vulnerabilities and Exposures". Es handelt sich um eine öffentliche Sammlung von Informationen zu bekannten Sicherheitsschwachstellen in Software und

Hardware. Jede CVE-Nummer identifiziert eine spezifische Sicherheitsschwachstelle und ermöglicht es, auf einfache Weise über Sicherheitsprobleme zu kommunizieren und Lösungen zu finden. Die CVE-Nummer wird von einer Organisation, der MITRE Corporation, vergeben und dient der Standardisierung und Vereinfachung der Sicherheitskommunikation und -koordination. Aufgrund der Übersichtlichkeit werden nur die 3 schwerwiegendsten Schwachstellen aufgelistet und erläutert.

Für Django gibt es 99 identifizierte Sicherheitslücken. Der extremste Fall ist CVE-2014-0474, mit einem CVSS Score von 10. Die Modellfeldklassen (1) FilePathField, (2) GenericIPAddressField und (3) IPAddressField in Django vor 1.4.11, 1.5.x vor 1.5.6, 1.6.x vor 1.6.3 und 1.7.x vor 1.7 Beta 2 Führen Sie die Typkonvertierung nicht ordnungsgemäß durch, was entfernten Angreifern unbestimmte Auswirkungen und Vektoren im Zusammenhang mit der „MySQL-Typumwandlung“ ermöglicht. Diese drei Felder wurden aktualisiert, um ihre Argumente vor der Abfrage in die richtigen Typen umzuwandeln. Darüber hinaus werden Entwickler von benutzerdefinierten Modellfeldern jetzt über die Dokumentation gewarnt, um sicherzustellen, dass ihre benutzerdefinierten Feldklassen die entsprechenden Typkonvertierungen durchführen, sowie Benutzer der Abfragemethoden `raw()` und `extra()` – die es dem Entwickler ermöglichen, Roh-SQL oder SQL-Fragmente bereitzustellen - Es wird empfohlen, sicherzustellen, dass vor der Ausführung von Abfragen entsprechende manuelle Typkonvertierungen durchgeführt werden.

Danach folgt CVE-2016-9013 mit einem CVSS-Score von 9.8: Django 1.8.x vor 1.8.16, 1.9.x vor 1.9.11 und 1.10.x vor 1.10.3 verwenden ein hartcodiertes Passwort für einen temporären Datenbankbenutzer, der beim Ausführen von Tests mit einer Oracle-Datenbank erstellt wird, was es für Remote-Angreifer einfacher macht, um Zugriff auf den Datenbankserver zu erhalten, indem das Versäumnis ausgenutzt wird, manuell ein Kennwort im TEST-Wörterbuch der Datenbankeinstellungen anzugeben. Dieser Benutzer wird normalerweise nach Abschluss der Testsuite gelöscht, jedoch nicht, wenn die Option `manage.py test --keepdb` verwendet wird oder wenn der Benutzer eine aktive Sitzung hat (z. B. eine Verbindung eines Angreifers). Für jeden Testlauf wird nun ein zufällig generiertes Passwort verwendet.

Formatiert: Standard

Zuletzt gibt es CVE-2019-14234. Ein Problem wurde in Django 1.11.x vor 1.11.23, 2.1.x vor 2.1.11 und 2.2.x vor 2.2.4 entdeckt. Aufgrund eines Fehlers bei der flachen Schlüsseltransformation wurden Schlüssel- und Indexsuchen für `django.contrib.postgres.fields.JSONField` und Schlüsselsuchen für `django.contrib.postgres.fields.HStoreField` einer SQL-Injection unterzogen. Dies könnte beispielsweise durch die manipulierte Verwendung von „OR 1=1“ in einem Schlüssel oder Indexnamen ausgenutzt werden, um alle Datensätze mithilfe eines entsprechend gestalteten Wörterbuchs mit Wörterbucherweiterung zurückzugeben, da die `**kwargs` an die `QuerySet.filter()` Funktion übergeben werden.

Bei React.js gibt es nur eine Schwachstelle, nämlich CVE-2018-6341 mit einem CVSS-Score von 6.1. React-Anwendungen, die mithilfe der ReactDOMServer-API in HTML gerendert wurden, konnten beim Rendern vom Benutzer bereitgestellte Attributnamen nicht umgehen. Dieses Fehlen von Escape-Funktionen könnte zu einer Cross-Site-Scripting-Schwachstelle führen. Dieses Problem betraf die Nebenversionen 16.0.x, 16.1.x, 16.2.x, 16.3.x und 16.4.x. Es wurde in 16.0.1, 16.1.2, 16.2.1, 16.3.3 und 16.4.2 behoben.

Bei MySQL gibt es 92 Sicherheitslücken. Es gibt zwei CVEs mit einem Score von 10, nämlich CVE-2004-0627 und CVE-2003-0628. Bei CVE-2004-0627 ermöglicht die Funktion `check_scramble_323` in MySQL 4.1.x vor 4.1.3 und 5.0 entfernten Angreifern, die Authentifizierung über eine verschlüsselte Zeichenfolge der Länge Null zu umgehen. Laut NGSSoftware Security Advisory wurde diese Schwachstelle in Version 4.1.3 (Beta) und Version 5.0 (Alpha) behoben.

Bei CVE-2004-0628 ermöglicht der stapelbasierte Pufferüberlauf in MySQL 4.1.x vor 4.1.3 und 5.0 entfernten Angreifern, einen Denial-of-Service (Absturz) auszulösen und möglicherweise beliebigen Code über eine lange Scramble-Zeichenfolge auszuführen. Hier ist die Lösung dieselbe wie bei der davor genannten CVE.

Zuletzt folgt CVE-2003-0780 mit einem CVSS-Score von 9. Pufferüberlauf in `get_salt_from_password` von `sql_acl.cc` für MySQL 4.0.14 und früher sowie 3.23.x ermöglicht es Angreifern mit ALTER TABLE-Berechtigungen, beliebigen Code über ein langes Passwortfeld auszuführen. Nach Angaben des Software Engineering Institute der Carnegie Mellon University ist dieses Problem in den MySQL-Versionen 3.23.58 und 4.0.15 behoben. Führen Sie ein Upgrade durch oder wenden Sie einen Patch an, wie von Ihrem Anbieter empfohlen.

6 Fazit

Die Entwicklung und Implementierung der Plattform "Students4Students – DHBW-Edition" ist ein bedeutender Schritt in Richtung Stärkung und Unterstützung der Studierendengemeinschaft an der DHBW Stuttgart. Das Projekt wurde initiiert, um den Herausforderungen, denen Studierende während ihrer Studienzeit gegenüberstehen, entgegenzuwirken und eine lebendige Gemeinschaft zu fördern. Während der Entwicklung sind mehrere Probleme aufgetreten, die sich als Lerninstanz präsentierten.

Erfolgreich war die benutzerfreundliche Gestaltung der Anwendung, um einen intuitiven Umgang für alle Studierenden zu gewährleisten, unabhängig von ihrem technischen Hintergrund. Die Konzeption und Planung der Plattform waren solide, wodurch klare Ziele für die Schaffung einer funktionalen Plattform definiert wurden. Die Fokussierung auf Zusammenarbeit und gegenseitige Unterstützung als Grundprinzipien ist ein starker Ausgangspunkt.

Herausforderungen sind gehäuft in der Anfangsphase entstanden. Das Erlernen der Nutzung von verschiedenen Ressourcen war teils kompliziert. Beispiele dafür war der Aufbau der containerbasierten Architektur mit Docker, sowie die kontinuierliche Einbindung der Softwarekomponenten, vor allem auch in Hinsicht des Zusammenspiels der Docker-Container. Aufgrund der hohen Komplexität des Aufbaus wurde erhöhter Fokus auf die zentralen Funktionen der Anwendung gelegt, wodurch jedoch die Basis für Erweiterungen besteht.

Die Lessons Learned lassen sich definieren wie folgt:

1. Frühzeitige Nutzereinbindung: Frühzeitige Einbindung der Nutzer in den Entwicklungsprozess kann dazu beitragen, ihre Bedürfnisse besser zu verstehen und die Akzeptanz der Plattform zu steigern. Durch die Kürze der gegebenen Zeit zur Realisierung des Projektes konnte erst gegen Ende das Projekt getestet werden, was zu unnötigen Komplikationen geführt hat.
2. Der Zeitaufwand muss ordentlich eingeschätzt werden. Funktionalitäten, die anfangs als einfach eingeschätzt wurden, wurden oft unterschätzt und somit hat sich der Fortschritt deutlich verlangsamt.
3. Effektive Kommunikation: Eine klare und effektive Kommunikationsstrategie im Team ist entscheidend, um die Plattform voranzutreiben und die Erwartungen von allen Beteiligten zu erfüllen.

Insgesamt wird "Students4Students – DHBW-Edition" erfolgreich dazu beitragen, eine unterstützende Gemeinschaft zu schaffen und den Austausch von Ressourcen innerhalb der Studierendengemeinschaft zu fördern. Durch die identifizierten Lessons Learned kann die

Plattform weiterentwickelt werden, um den Bedürfnissen der Studierenden noch besser gerecht zu werden.

7 Quellen

Anchiti, Andrea: "TypeScript coding standards", in: Internetseite Github, URL: <https://gist.github.com/anichitiandreea/e1d466022d772ea22db56399a7af576b>, Abgerufen am 23.10.2023

CVE: „Djangoproject >> Django: Security Vulnerabilities“, in: Internetseite SecurityScorecard, URL: https://www.cvedetails.com/vulnerability-list/vendor_id-10199/product_id-18211/Djangoproject-Django.html, Abgerufen am 27.10.2023

Feldfunktion geändert

Django: „News & Events“, in: Internetseite Django, URL: <https://www.djangoproject.com/weblog/2014/apr/21/security/>, Abgerufen am 27.10.2023

DSGVO: „Begriffsbestimmungen“, in: Internetseite DSGVO-Gesetz, URL: <https://dsgvo-gesetz.de/art-4-dsgvo/>, Abruf am 20.10.2023

Egly, Uwe: „Kryptographie, Allgemeine Einführung“, in: Internetseite TU Wien, URL: http://www.kr.tuwien.ac.at/education/krypto_slides/ws10/slides-intro.pdf, Abruf am 20.10.2023

OWASP: „Secure Product Design Cheat Sheet“, in: Internetseite OWASP Cheat Sheet Series, URL: https://cheatsheetseries.owasp.org/cheatsheets/Secure_Product_Design_Cheat_Sheet.html, Abruf am 25.10.2023

Feldfunktion geändert

Rafail, Jason A: „MySQL fails to validate length of password field“, in: Internetseite Carnegie Mellon University, URL: <https://www.kb.cert.org/vuls/id/516492>, Abgerufen am 27.10.2023

Feldfunktion geändert

Van Rossum, Guido: "PEP8 – Style Guide for Python Code", in: Internetseite Python Enhancement Proposals, URL: <https://peps.python.org/pep-0008/>, Abgerufen am 23.10.2023

Feldfunktion geändert

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt. Zudem bestätige ich: Aus den benutzten Quellen direkt oder indirekt übernommene Gedanken habe ich als solche kenntlich gemacht. Diese Arbeit wurde bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Datum: 6.11.2023

Namen: Daniel Desgronte, Daniel Künkel, Marc Issmer, Amanda de Moura