

Softwareprojekt Wintersemester 2023/24
Gruppe WSP3 1



Lia	Lenckowski
Leon	Klausing
Matheus	Kolzarek
Noe	Baumann
Shutong	Sun
Til	Schmidt
Tjark-Morten	Thies
Yuri	Hassink



10. März 2024

Inhaltsverzeichnis

1	Einleitung	1
1.1	Entwicklungsumgebung	1
2	Team-Aufteilung	2
3	Komponentendiagramme	3
4	Verteilungsdiagramm	5
5	Klassendiagramme oder API-Beschreibung	6
5.1	API-Diagramme für Lehrer	6
5.2	API-Diagramme für SuS	10
6	Sequenzdiagramme	12
6.1	SuS-Login	12
6.2	SuS Produkt kaufen	
6.3	Lehrkraft Produkt hinzufügen	

Kapitel 1

Einleitung

1.1 Entwicklungsumgebung

Software	Version	URL
Python	3.11.X	https://www.python.org/
Docker	25.0.3	https://www.docker.com/
Git	2.44.0	https://git-scm.com/
VSCode	1.84.0	https://code.visualstudio.com/
Flask	3.0.2	https://flask.palletsprojects.com/en/3.0.x/
SQLite	3.45.1	https://www.sqlite.org/
nginx	1.21.3	https://www.nginx.com/
uWSGI	2.0.24	https://uwsgi-docs.readthedocs.io/en/latest/
Open JDK	11.0.22	https://www.oracle.com/java/technologies/downloads
Android Studio	2023.2.1	https://developer.android.com/studio

Tabelle 1.1: Entwicklungsumgebung

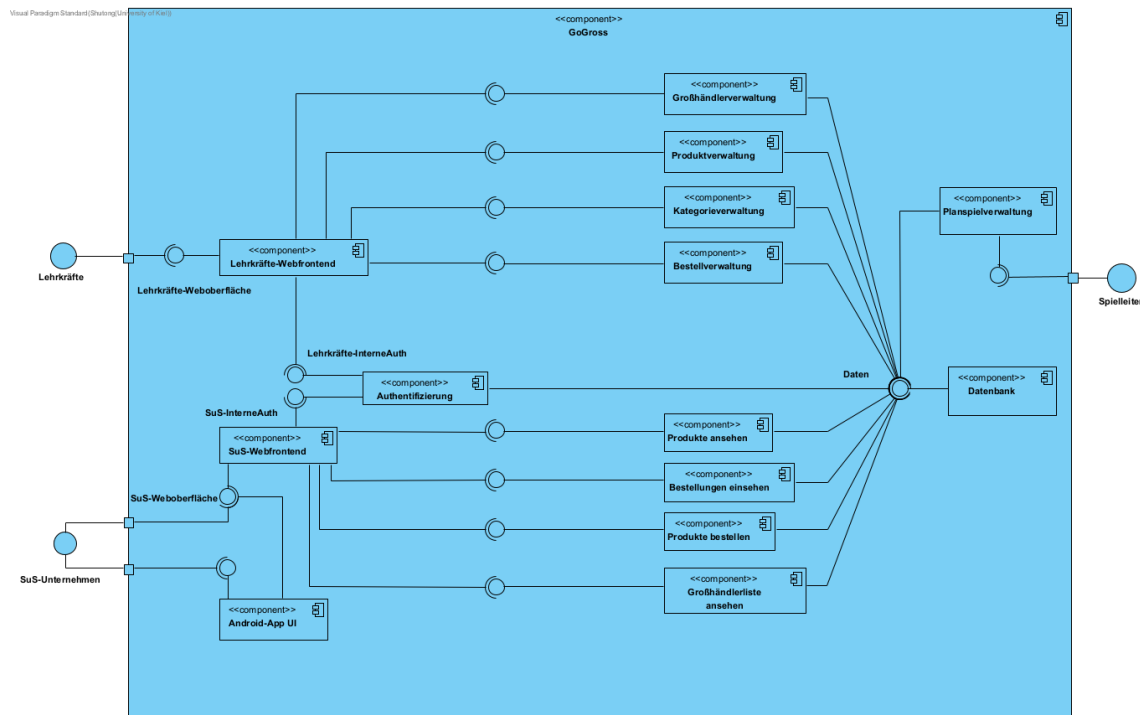
Kapitel 2

Team-Aufteilung

Name	Zuständigkeit
Lia	1. CI/CD Tests, 2. Backend
Leon	1. Frontend, 2. Testen
Matheus	Backend
Noe	1. Frontend APP, 2. Backend
Shutong	1. Frontend, 2. Backend
Til	1. Frontend 2. Backend
Tjark-Morten	1. Frontend 2. Backend
Yuri	Backend

Kapitel 3

Komponentendiagramme



- Der Spielleiter kann alle Planspiele über die Kommandozeile verwalten. Die Verwaltung umfasst die Erstellung, das Beenden sowie den Import und Export von Spielständen.
- Die Lehrkräfte nutzen eine über HTTPS gesicherte Weboberfläche, die über eine Authentifizierungskomponente gesichert ist. Nach erfolgreicher Authentifizierung können diese auf verschiedene Verwaltungskomponenten zugreifen. Diese ermöglichen das Management von Großhändlern, Produkten, Kategorien und Bestellungen. Wie zum Beispiel das Löschen, Hinzufügen, Importieren oder Exportieren.

- SuS-Unternehmen verwenden entweder eine Weboberfläche oder eine auf der Weboberfläche basierende Android-Anwendung, um sich zu authentifizieren und anschließend Zugriff auf verschiedene Bestellkomponenten zu erhalten. So können diese Großhändler, Produkte und Bestellungen ansehen sowie Bestellungen tätigen.
- Die Authentifizierungskomponente gewährleistet einen sicheren Zugang für Lehrkräfte und SuS-Unternehmen. Diese reguliert den Zugriff mithilfe der gespeicherten Benutzerdaten in der Datenbank und sichert somit die Integrität sowie Sicherheit des Systems.
- Das GoGross System wird von einer internen Datenbankkomponente gestützt, welche relevante Informationen und Beziehungen speichert. Diese Datenbank wird von den verschiedenen Komponenten zur Ausführung ihrer spezifischen Aufgaben abgefragt und verändert.

Kapitel 4

Verteilungsdiagramm

- Alle Komponenten des Systems werden auf dem Webserver des RBZ gehostet.
- Um unsere Web-Anwendung zu nutzen, muss Docker auf dem Webserver laufen.
- Alle Daten werden mittels SQLite auf dem Webserver gespeichert.
- Alle HTTPS-Anfragen werden mittels nginx auf den Webserver geleitet.

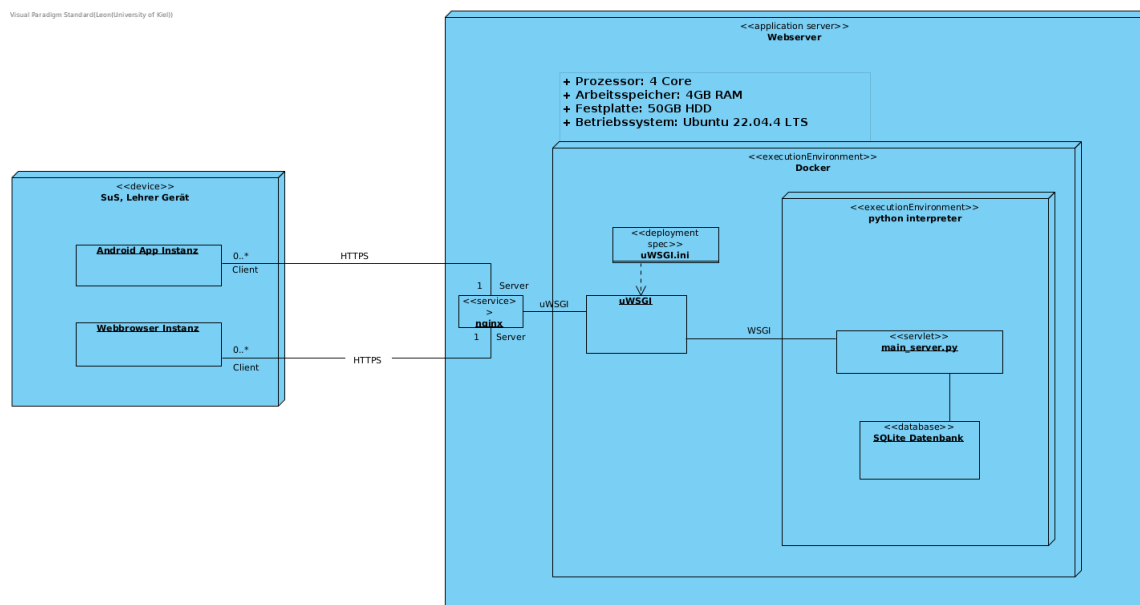


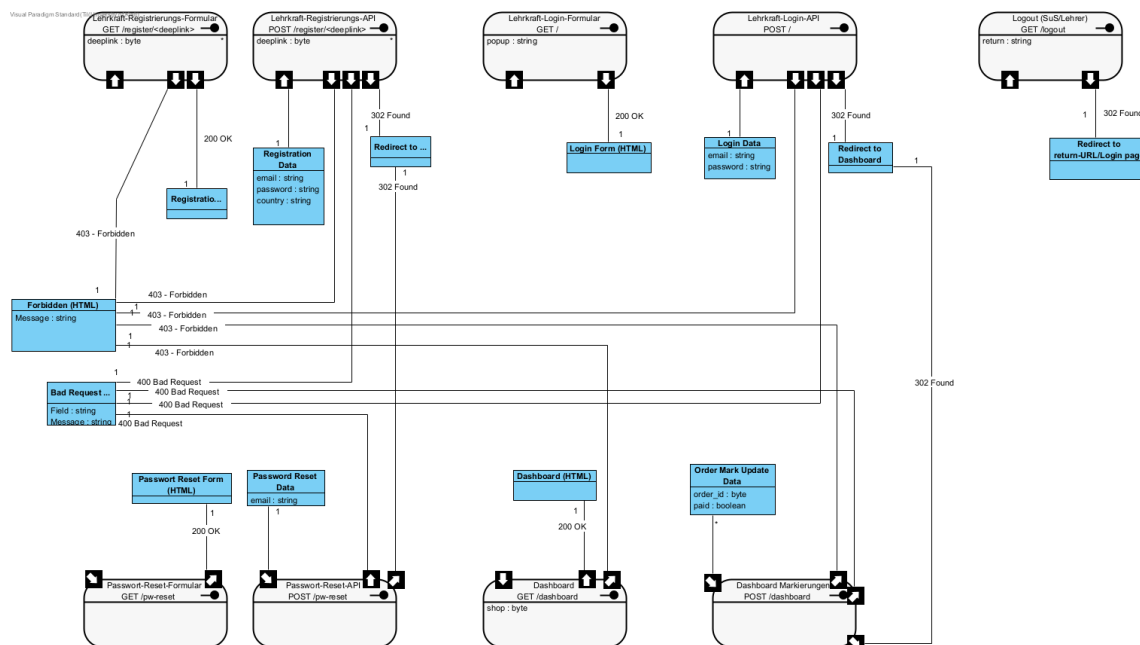
Abbildung 4.1: Verteilungsdiagramm

Kapitel 5

Klassendiagramme oder API-Beschreibung

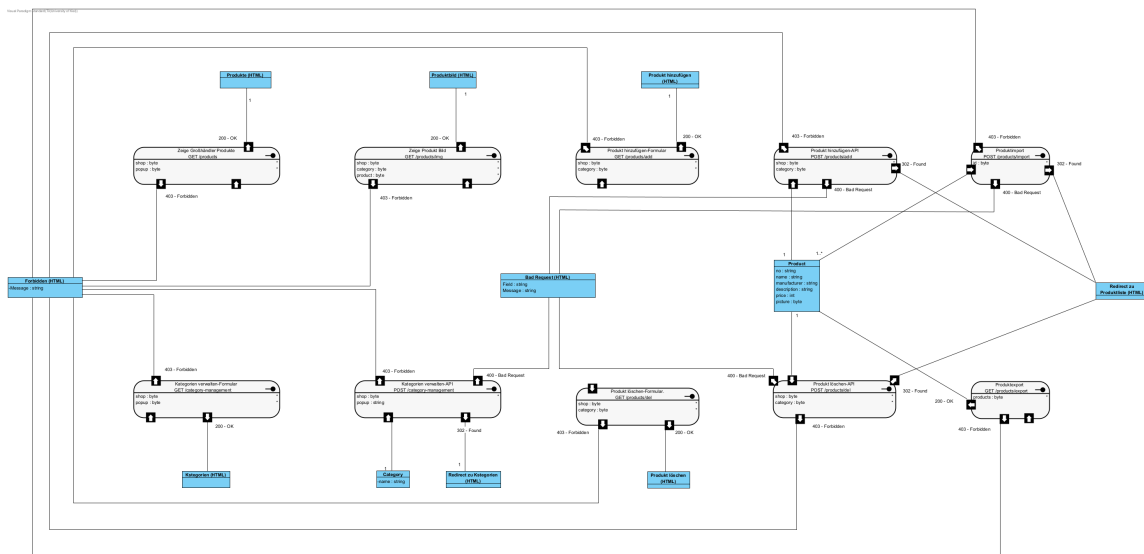
Auf die Diagramme folgt jeweils eine textuelle Beschreibung aller Endpunkte. Darin stellen Pfadteile in angle brackets (<x>) nötige, und Pfadteile in square brackets ([x]) optionale Parameter dar.

5.1 API-Diagramme für Lehrer



- /register/<deeplink>: Die Lehrer-Registrierung ist nur über deinen Deeplink erreichbar.
 - GET: Registrierungsformular.

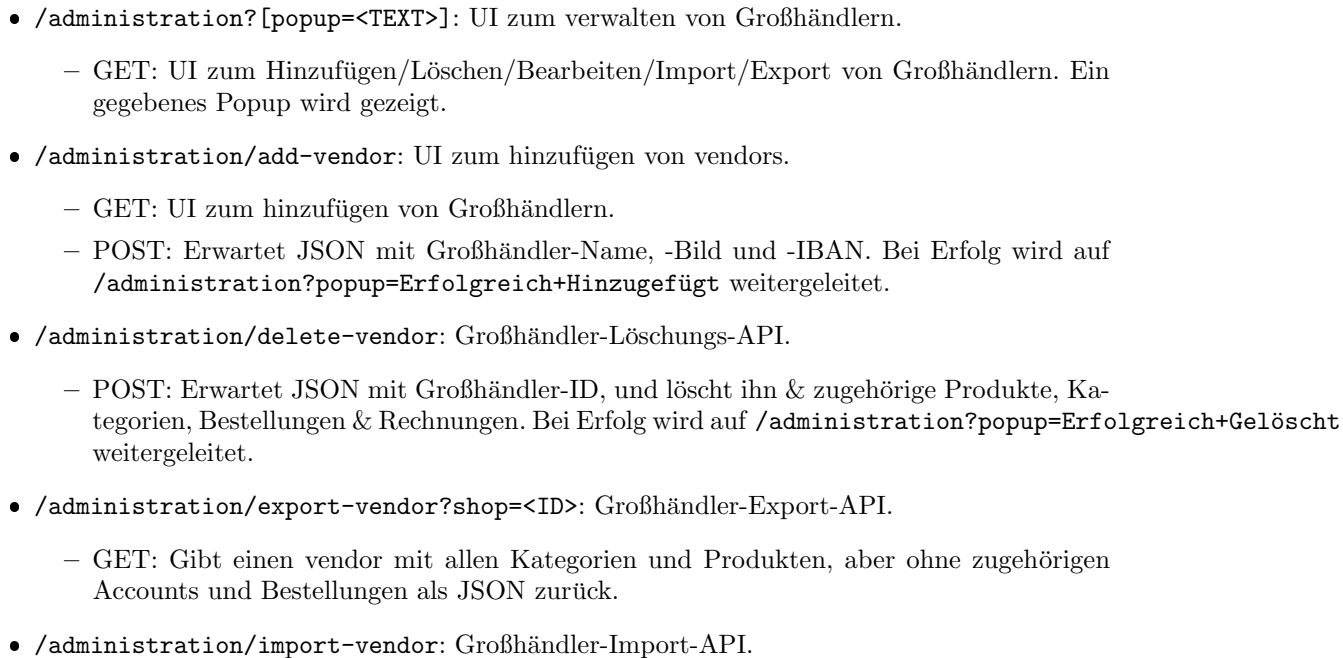
- POST: Hier wird ein JSON-Body mit den Daten aus dem Registrierungsformular erwartet. Erfolg führt zu einer Weiterleitung zu `/?popup=Registrierung+Erfolgreich`.
- `/?[popup=<TEXT>]`: Lehrkraft-Login.
 - GET: Login-Formular. Falls ein popup-Text gegeben wird, wird dieser angezeigt.
 - POST: Hier wird ein JSON-Body mit Login-Daten erwartet. Erfolg führt zu einer Weiterleitung zu `/dashboard`.
- `/pw-reset`: Lehrkraft-Passwort-Reset.
 - GET: Passwort-Reset-Formular.
 - POST: Hier wird ein JSON-Body mit der E-Mail, dessen zugehöriger Account sein Passwort zurückgesetzt haben soll, erwartet. Erfolg führt zu einer Weiterleitung zu `/`.
- `/logout?[return=<URL>]`: Logout.
 - GET: Aufruf führt zum Logout, sowohl von Lehrkraft- als auch SuS-Sessions. Falls angegeben, wird danach zur return-URL weitergeleitet, sonst zu `/`.
- `/dashboard?[shop=<ID>]`: Lehrkraft-Dashboard. Nur nach Login mit Lehreraccount erreichbar. Angabe des shop-parameters filtert die angezeigte Bestellungsliste auf Bestellungen bei dem gegebenen Shop.
 - GET: Lehrkraft-Dashboard. Es werden nur Daten zu den Großhändlern aus dem Land des Lehrkraftaccounts dargestellt.
 - POST: Hier wird ein JSON-Body erwartet, der für eine Untermenge an Bestellungen angibt, ob sie als bezahlt oder unbezahlt zu markieren sind. Erfolg führt zu einer Weiterleitung zu `/dashboard`.



Die folgenden Endpoints sind je nur mit einem Lehreraccount erreichbar, dessen Land zu evtl. angegebenen Großhändlern/Produkten passt.

- `/products?shop=<ID>[&category=<ID>&popup=<ID>]`: Produktlisten aus Lehrkraft-Sicht.

- GET: Zeigt die Produktliste des angegebenen Shops an. Je nach Parametern sind Produkte nach einer Kategorie gefiltert und/oder es wird ein Popup mit Text angezeigt.
- `/products/img?product=<ID>`: Produktbilder.
 - GET: Gibt als je natives Format das Bild des angegebenen Produktes zurück.
- `/products/export?products=<ID>[,ID2[,ID3[...]]]`: Produktexport.
 - GET: Gibt als JSON-Liste die angegebenen Produkte zurück.
- `/products/import?shop=<ID>`: Produktimport.
 - POST: Erwartet eine JSON-Liste an Produkten, die zu dem angegebenen Großhändler hinzugefügt werden.
- `/products/add?shop=<ID>`: Produkt-Hinzufügungs-UI.
 - GET: UI zum hinzufügen von einem Produkt.
 - POST: Erwartet JSON mit Produktdaten & Bild, und fügt es beim angegebenen shop in einer im JSON angegebenen Kategorie hinzu.
- `/products/del?shop=<ID>`: Produktlöschungs-API.
 - POST: Erwartet JSON mit Produkt-ID. Bei erfolg wird zu `/products?shop=ID&popup=Erfolgreich+Gelöscht` weitergeleitet.
- `/category-management?shop=<ID>[&popup=<TEXT>]`: UI fürs hinzufügen/löschen von Kategorien.
 - GET: UI zum Hinzufügen/Löschen von Kategorien, wobei man Änderungen erst speichern muss (was zu einem POST an dieselbe URL führt). Ein gegebenes Popup wird gezeigt.
 - POST: Erwartet als JSON eine Liste an hinzugefügten & gelöschten Kategorien. Bei Erfolg wird auf `/category-management?shop=ID&popup=Erfolgreich+Geändert` weitergeleitet.



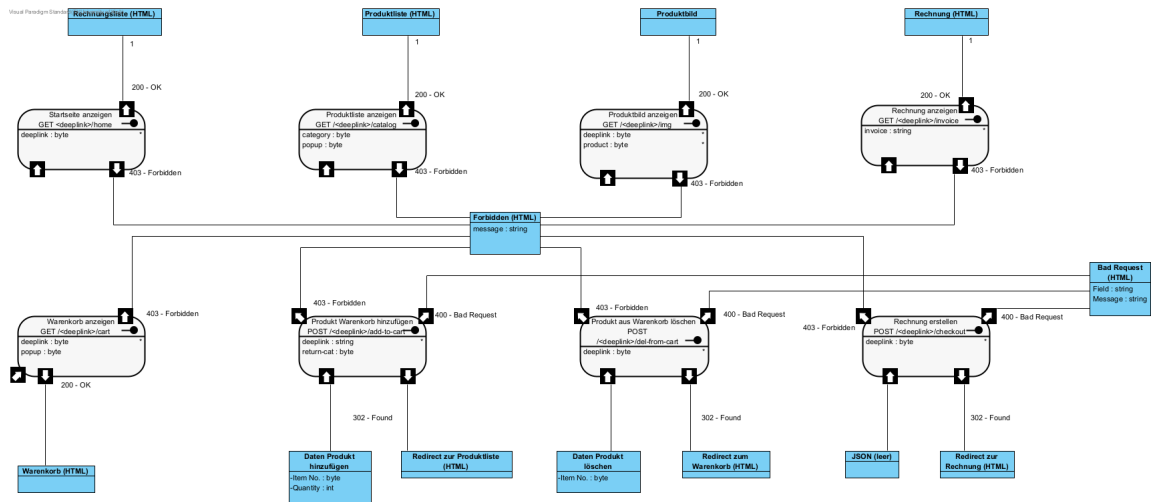
- POST: Erwartet eine Datei, die durch einen Großhändler-Export erstellt wurde. Bei Erfolg wird auf `/administration?popup=Erfolgreich+Importiert` weitergeleitet.
- `/administration/change-iban?shop=<ID>`: Großhändler-Bearbeitungs-UI.
 - GET: Großhändler-Bearbeitungs-UI. Dabei sind auf jeden fall IBAN, später vielleicht auch das Logo einstellbar.
 - POST: Erwartet JSON mit einer neuen IBAN und/oder einem neuen Logo. Bei Erfolg wird auf `/administration?popup=Erfolgreich+Geändert` weitergeleitet.

5.2 API-Diagramme für SuS

Hier verzichten wir aufs Darstellen von Registrierung/Login, da es im wesentlichen gleich wie bei Lehrkräften aufgebaut ist, außer das alle pfade einen deeplink als Präfix haben.

- `/<deeplink>`: Großhändlerübersicht-UI
 - GET: UI, welche die Übersicht aller Großhändler des zugehörigen deeplinks anzeigt.
- `/<deeplink>/invoice?id=<ID>`: Invoice-PDF
 - GET: Zeigt Invoice als PDF an.
- `/<deeplink>/catalog?[category=<ID>] [&popup=<TEXT>]`: Produktübersicht-UI
 - GET: UI zur Übersicht der Produkte eines Großhändlers. Ein gegebenes Popup wird gezeigt.
- `/<deeplink>/img?product=<ID>`: Produktbilder.
 - GET: Gibt als je natives Format das Bild des angegebenen Produktes zurück.
- `/<deeplink>/add-to-cart?[return-cat=<ID>]`: UI zum Hinzufügen von Produkten aus dem Warenkorb
 - POST: Erwartet JSON mit Produkt-ID und Anzahl. Bei Erfolg wird auf `/<deeplink>/catalog?[category=<ID>]` weitergeleitet.
- `/<deeplink>/cat?[popup=<TEXT>]`: Warenkorb-UI
 - GET: UI zur Ansicht des Warenkorbs. Ein gegebenes Popup wird gezeigt.
- `/<deeplink>/del-from-cart`: UI zum Löschen von Produkten aus dem Warenkorb
 - POST: Erwartet JSON mit Produkt-ID und löscht das Produkt. Bei Erfolg wird auf `/<deeplink>/car?popup=Erfolgreich+Gelöscht` weitergeleitet.
- `/<deeplink>/checkout`: Bestellen-UI
 - POST: Erwartet ein leeres JSON. Bei Erfolg wird auf die erstellte Invoice weitergeleitet.

Klassendiagramme oder API-Beschreibung

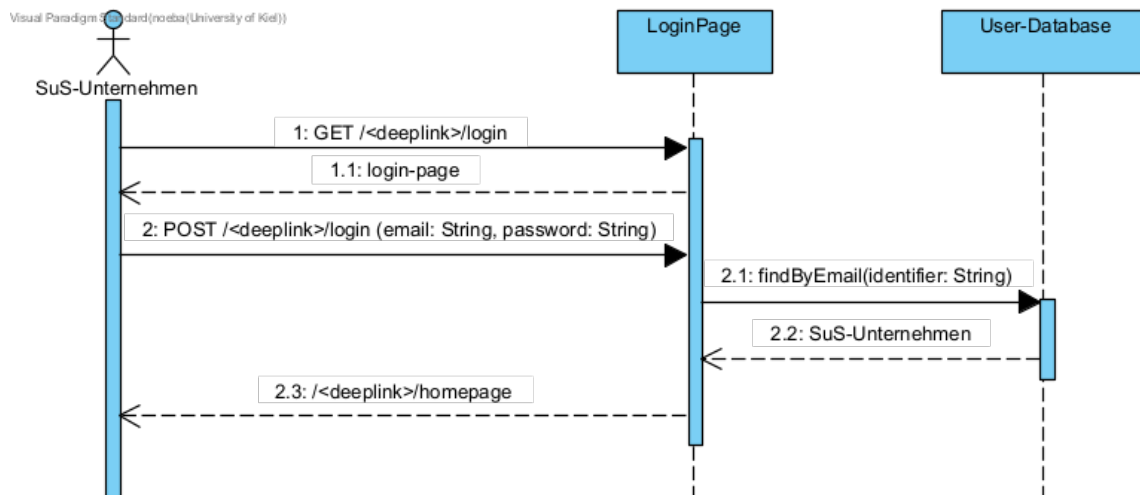


Kapitel 6

Sequenzdiagramme

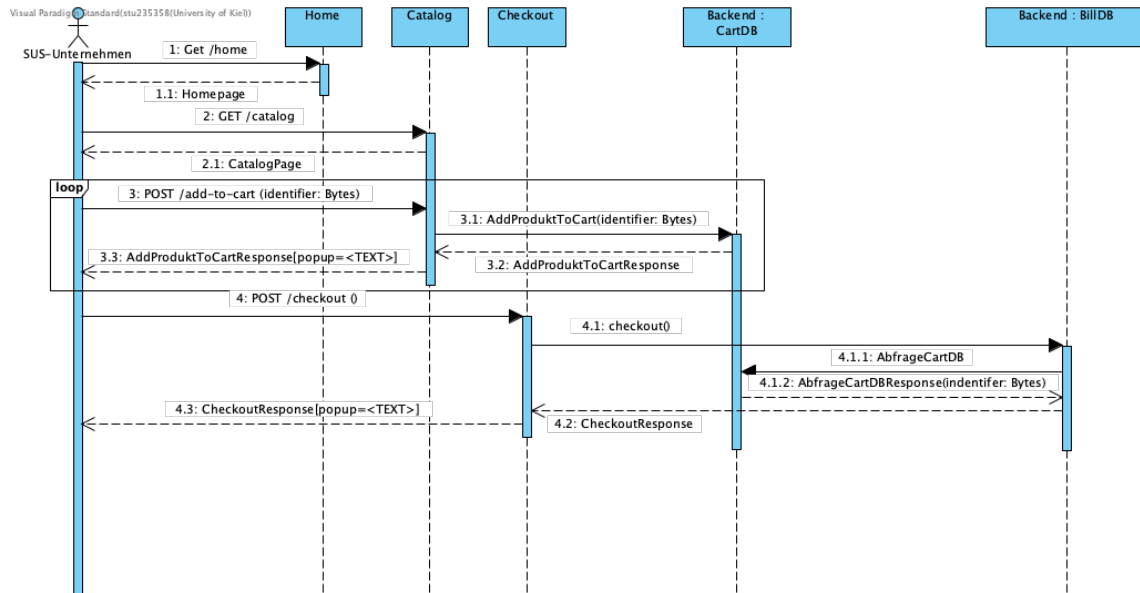
Es ist wichtig bei allen folgenden Sequenzdiagrammen zu beachten, dass die sonst als lifelines bekannten Objekte hier als üsagelineinterpretiert werden sollten.

6.1 SuS-Login



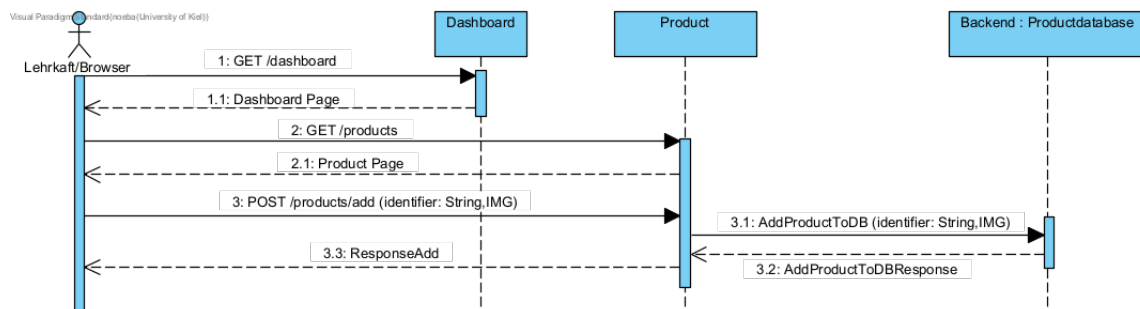
Um sich als SuS-Unternehmen einzuloggen wird vom oder von der Nutzer*in erst die LoginPage via Deeplink abgefragt. An diese werden danach die Logindaten gesendet, welche diese weiter an die Datenbank leitet um den oder die Nutzer*in zu authentifizieren. Ist die Authentifikation erfolgreich, so gibt die Datenbank den SuS-Unternehmen-Account zurück an die LoginPage und diese leitet den oder die Nutzer*in weiter zu der Shop-Homepage.

6.2 SuS Produkt kaufen



Dieses Sequenzdiagramm beschreibt den Prozess vom Produkt anschauen bis hin zum Kauf. Nachdem der oder die Nutzer*in die Homepage abgefragt und erhalten hat, fragt diese*r die Katalogseite ab. Hier kann nun ein Produkt in Warenkorb hinzugefügt werden. Produkt & session werden an eine Backend-Komponente weitergeleitet, welche den aktuellen Warenkorb für jeden Nutzer speichert. Werden mehr als nur eine Instanz des Produkts gleichzeitig dem Warenkorb hinzugefügt, so wird dieser loop entsprechend oft durchgeführt. Entscheidet sich der oder die Nutzer*in den Warenkorb zu bestellen, so schickt er/sie über die Weboberfläche einen Anfrage an einen Checkout-Endpoint, welcher diese an die Bill-Datenbank im Backend sendet. Die Bill-Datenbank fragt dann bei der Cart-Datenbank an, welche Objekte sich im Warenkorb befinden und generiert eine Rechnung, welche gespeichert und an den Nutzer zurückgeschickt wird.

6.3 Lehrkraft Produkt hinzufügen



Wir zeigen hier den Sequenzablauf für eine Lehrkraft, die ein Produkt ins System einführen möchte. Nach dem Login und dem navigieren zur Produkt-Hinzufügungs-Oberfläche folgen über POST die Daten über das Produkt. Die Lehrkraft wird dann wieder auf die Ursprungsseite weitergeleitet, wo per popup eine Meldung gezeigt wird, die das Hinzufügen des Produktes bestätigt.