Marvin Prang 10.11.23

Hausaufgabe 3

1 shall = soll should = sollte

user = tutor system = klipsias

Wenn der user sich einloggt soll ~~das~~ klipsias eine Verifizierung durchführen und Passwort / Nutzername mit der Datenbank abgleichen.

Danach sollte klipsias den User auf eine Startseite weiterleiten.

Dort sollte ~~das~~ klipsias alle Kurs Räume des aktuellen Semesters auflisten, die der User Zugriff hat.

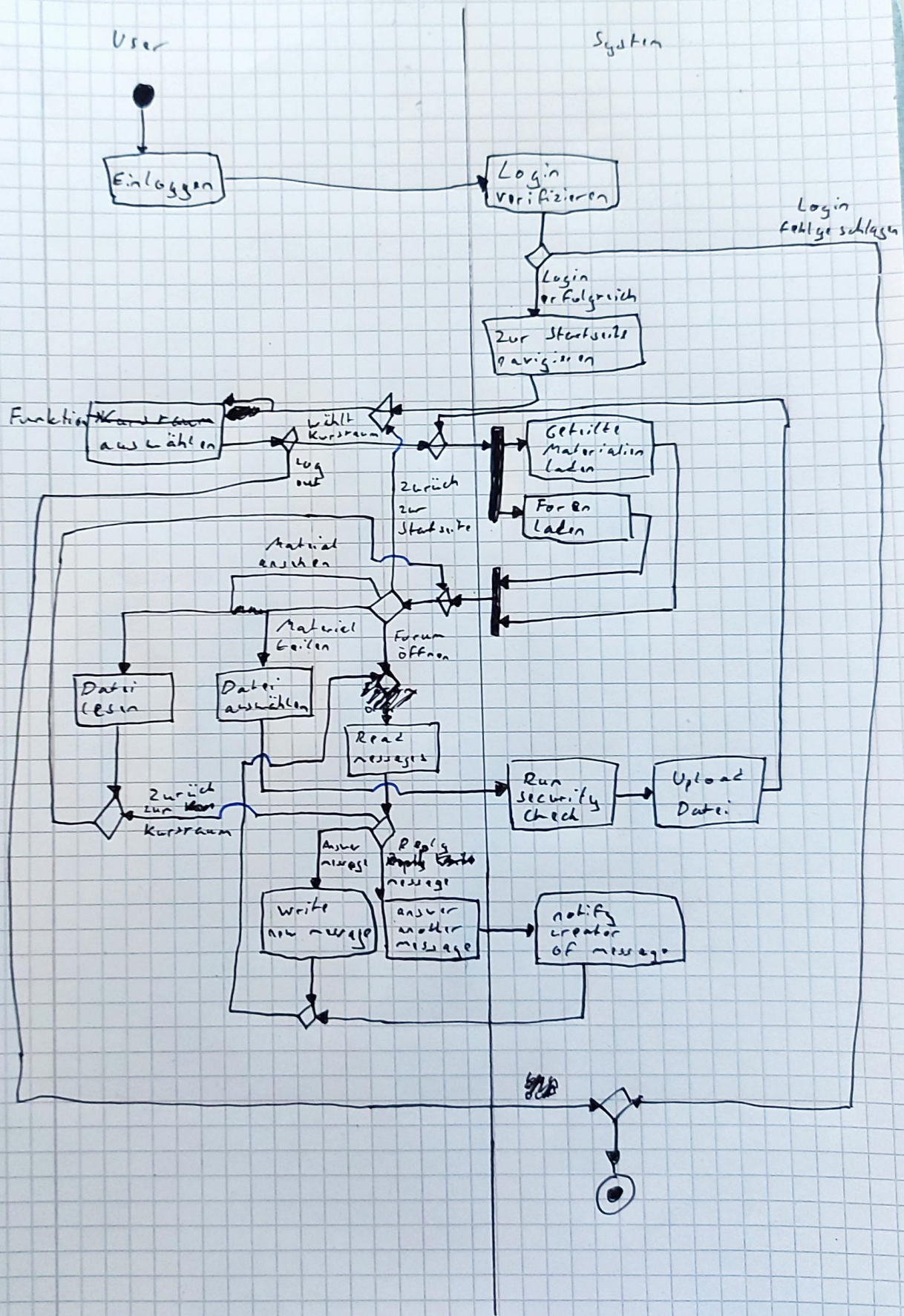
~~Das System~~

klipsias sollte Funktion die nicht mit Kurs Räumen ~~zusammen~~ zusammenhängen in einem Dropdown menü anbieten.

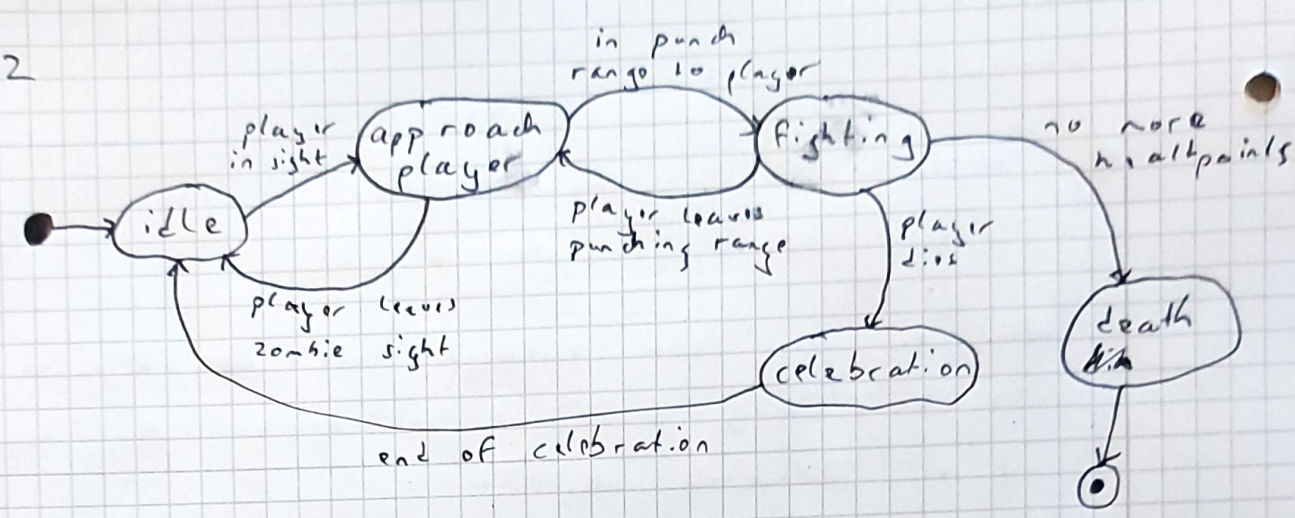
In einem Kursraum soll klipsias den User die verfügbaren geteilten Materialien und Foren anzeigen

~~Wenn~~ Teilt der User Materialien mit dem Kursraum soll klipsias einen Security Check für die Materialien ausführen

Antwortet der User einer Person im Forum soll klipsias diese benachrichtigen.


User
System
Einloggen
Login verifizieren
Login fehlgeschlagen
Login erfolgreich
Zur Startseite navigieren
Funktion
Kurs auswählen
Wählt Kursraum
Log out
Zurück zur Startseite
Gewählte Materialien laden
Foren laden
Material ansehen
Material teilen
Forum öffnen
Datei lesen
Datei auswählen
Read message
Run security check
Upload Datei
Zurück zum Kursraum
Answer message
Reply message
Write new message
answer another message
notify creator of message

2



3 a) Model - View - Control Architecture

Die Logik und das Datenmanagment ist auf verschiedene Components aufgeteilt. Dabei enthält Article die relevanten Daten, Page zeigt die Daten an von Articles an. Und über Bidscheduler werden Bids abgegeben die entscheiden welche Daten bei Page visualisiert werden

b) Es wird in 3 Komponenten aufgeteilt, die alle miteinander arbeiten ~~[illegible]~~ aber unabhängig voneinander ~~[illegible]~~ sind. So sind die ~~[illegible]~~ Daten unabhängig davon was angezeigt wird

c) Extensibility: Neue Views hinzuzufügen ist bei dieser Architektur schwierig aufgrund der ineinandergreifenden Interfaces.

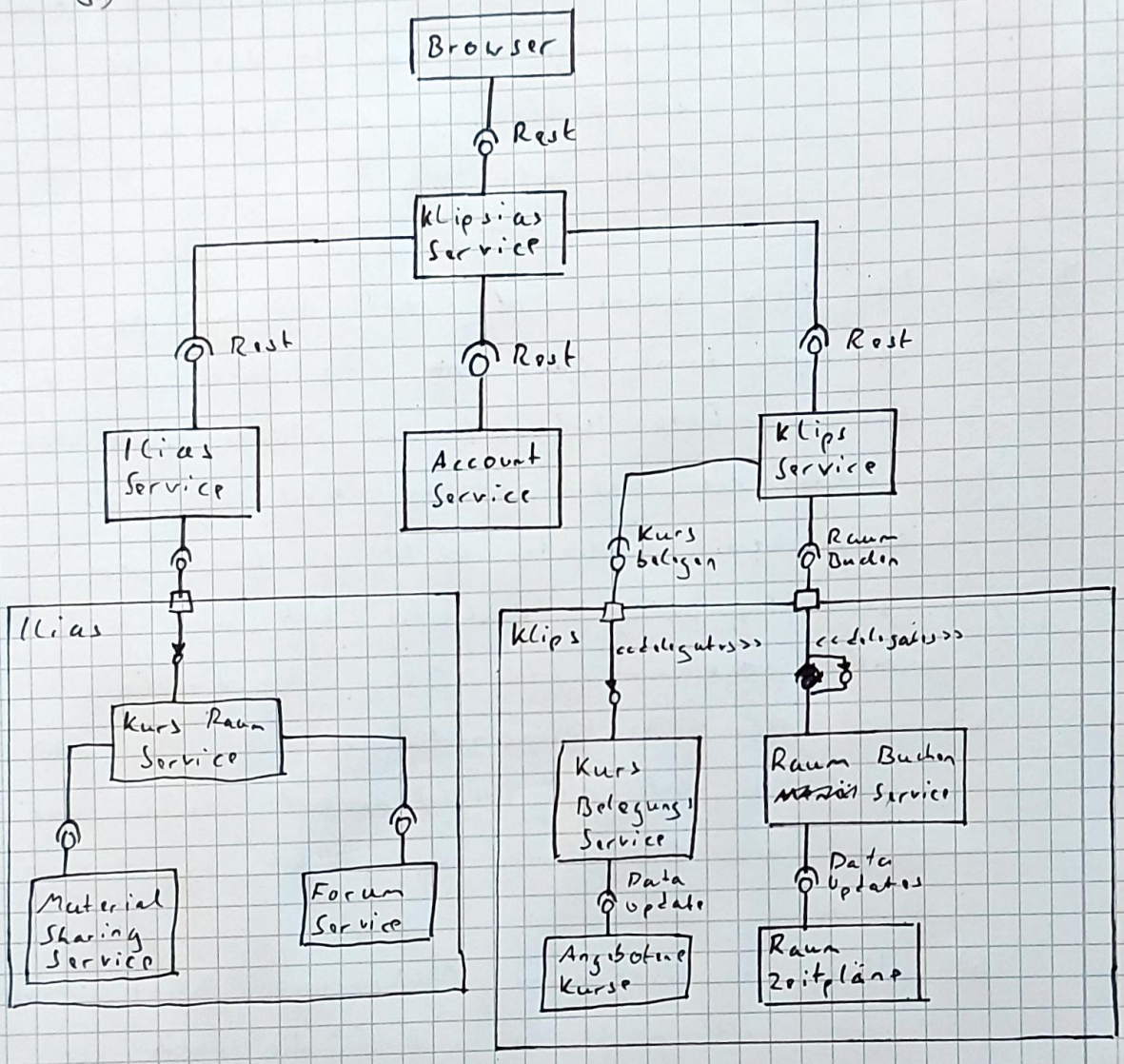
Modifiability: Durch das Routerbrechen komplexer Muster in nur drei Komponenten sind Änderungen und Wiederverwendbarkeit vorhanden.

Understandability: Durch das Unterbrechen häufig komplexer System in das MVC wird der Einstieg in das Verstehen des Codes erleichtert.

4 a) Das View würde als Presentation gemappt, bspw. zeigt die Daten für den User an. Data entspricht dem Model, dort werden die Daten be- und verarbeitet sowie gespeichert. Controller würde als Logik gemappt, jedoch unterscheidet sich diese beim mehr voneinander. In MVC wird die Logik auf alle 3 Komponenten verteilt, für PAD wird sie auf die eine Komponente zentriert was auch als Kommunikator von Client zu Daten dient.

b) Wenn ein Programm Offline und Online nutzbar sein soll macht es Sinn die Logik auf Client und Server zu verteilen, so muss der Client alle funktionale Logiken enthalten um auch offline funktionsfähig zu sein. Die Online Logik wirkt dann die Logik der Monetarisierung ab, bspw. Premium-Account, Store, Guthaben. Als Beispiel könnte ein Computerspiel sein, so sollte es online und offline spielbar sein (Client Logik). Über die Server Logik wird dann beispielsweise Multiplayer und/oder in Game Store / Währung geregelt.

5)



~~Edit Direction~~ Task

Für klipsias habe ich mich für den Aufbau einer Microservices Architektur entschieden, da bereits schon zwei Systeme existieren die so miteinander verbunden werden können (vor allem wenn es eventuell möglich ist auf Komponenten von Ilias und Campus Online zuzugreifen anstatt ein komplettes System zu nutzen und so nicht von null entwickelt werden muss).