

## **M223: Multiuser Applikation**

Probe IPA

# 1 Inhaltsverzeichnis

2	Vorwort .....	4
3	Organisation der Arbeitsergebnisse .....	4
4	Teil 1 – Umfeld und Ablauf .....	4
4.1	Aufgabenstellung .....	4
4.2	Projektaufbauorganisation .....	5
4.3	Mittel und Methoden .....	6
4.4	Vorkenntnisse .....	6
4.5	Vorarbeit .....	6
4.6	Deklaration der benützten Firmenstandards .....	6
4.7	Zeitplan .....	7
4.8	Arbeitsprotokoll .....	8
4.8.1	29.11.2021 – Tag 1 .....	8
4.8.2	30.11.2021 – Tag 2 .....	9
4.8.3	01.12.2021 – Tag 3 .....	10
4.8.4	06.12.2021 – Tag 4 .....	11
4.8.5	07.12.2021 – Tag 5 .....	12
5	Teil 2 .....	13
5.1	Kurzzusammenfassung .....	13
5.2	Informieren .....	13
5.2.1	Wertebereich .....	13
5.3	Planen .....	13
5.3.1	Zeitplan .....	14
	UMLs .....	15
5.3.2	Testing .....	19
5.4	Entscheiden .....	22
5.4.1	Versionsverwaltung .....	22
5.4.2	Projekt / Technologie .....	22
5.4.3	Design .....	23
5.5	Realisieren .....	23
5.5.1	Persistenz .....	23
5.5.2	Komponenten .....	24
5.6	Kontrollieren .....	24
5.6.1	Testing .....	24
5.7	Auswerten .....	24
5.7.1	Zeitplan (Soll/Ist Vergleich) .....	24

5.7.2	Reflexion .....	24
5.8	Glossar .....	25
5.9	Quellenverzeichnis .....	25
5.10	Abbildungsverzeichnis .....	25
5.11	Tabellenverzeichnis .....	25
5.12	Anhang.....	25

## 2 Vorwort

## 3 Organisation der Arbeitsergebnisse

Der Stand des Projekts wird mindestens nach jedem Tag auf Git abgelegt. Vorzugsweise wird die Arbeit auch nach jeder grösseren Änderung, zum Beispiel beim Erreichen eines Meilensteins, abgelegt. In der Versionentabelle unten wird nur die abgelegte Version eines Tages (Ende des Tages) festgehalten. Die anderen Versionen, können auf Git eingesehen werden.

Version	Datum	Änderung	Autor
<b>0.8</b>	06.12.2021	<ul style="list-style-type: none"> <li>Grundstruktur aller Fenster fertig</li> <li>Grundfunktionen erstellt (Login/Registrierung, Geschichte erweitern und Geschichte herunterladen)</li> <li>Arbeitsjournal 4</li> </ul>	LF
<b>0.6</b>	01.12.2021	<ul style="list-style-type: none"> <li>Planen erweitern</li> <li>Use Case in Textform</li> <li>Testing erstellt</li> <li>Anfangen Grundaufbau Fenster</li> <li>Arbeitsjournal 3</li> </ul>	LF
<b>0.4</b>	30.11.2021	<ul style="list-style-type: none"> <li>UML's (UseCase, Wireframe)</li> <li>Ddl erstellt</li> <li>update Projektdokumentation (Informieren, Planen und Entscheiden)</li> <li>Arbeitsjournal Tag 2</li> </ul>	LF
<b>0.2</b>	29.11.2021	<ul style="list-style-type: none"> <li>Dokumentation inkl. Struktur erstellt</li> <li>Gantt Planung</li> <li>Arbeitsjournal Tag 1</li> </ul>	LF

## 4 Teil 1 – Umfeld und Ablauf

### 4.1 Aufgabenstellung

Im Rahmen des ÜK Moduls 223 soll eine Multiuser Applikation erstellt werden.

Folgende Rahmenbedingungen wurde durch den Modulleiter Remo Steinmann vorgegeben.

- Objektorientierte Multi-User-Applikation
- Umfang gemäss Planung (ca. 3 Tage Entwicklung, 2.5 Tage Dokumentation; 6h pro Tag)
- Zentrale Datenbank
- Mehrere Clients müssen gleichzeitig auf den gleichen Datenbestand zugreifen
- Zentrale Benutzer- und Rechte-Verwaltung

Für dies werde ich das bekannte Spiel «Faltpapier-Geschichte» digitalisieren. Dieses Spiel funktioniert wie folgt:

1. Spieler 1 schreibt zwei Sätze
2. Spieler 2 sieht den zweiten Satz des Spieler 1
3. Aufgrund dessen schreibt er ebenfalls zwei Sätze
4. Spieler 3 sieht nur den zweiten Satz von Spieler 2
5. Aufgrund dessen schreibt er ebenfalls zwei Sätze

Und so weiter.

Das Erstellen der zwei Sätze wird Zeitbegrenzt sein, sodass man nicht zu lange studiert und die Geschichte spontaner wird. Zudem kann jeder User nur 1-mal alle 2h die Geschichte um zwei Sätze erweitern. So schreibt der User nicht eine Geschichte allein und es schützt vor Spam-Angriffen. Nach dem Absenden der erstellten Sätze, gibt es die Möglichkeit den aktuellen Stand Geschichte herunterzuladen.

Es gibt folgende Rollen und ihre Berechtigungen.

- **Writer**
  - Standardrolle, wenn man sich neu registriert.
  - Einloggen und ausloggen
  - Schreiben von 2 Sätzen
  - Herunterladen der Geschichte
- **Game Master**
  - Wird in der DB manuell von einem Writer zu einem Game Master migriert
  - Einloggen und ausloggen
  - Sieht aktueller Stand der Geschichte
  - Sieht welcher User was Geschrieben hat

Alle User müssen sich beim Start der Applikation über ein Login Fenster authentifizieren. Die Authentifizierung wird mittels Mailadresse und Passwort durchgeführt. Es gibt keine weiteren Möglichkeiten der Authentifizierung. Falls der User noch kein Login besitzt, kann er sich über den Registrierungs-Button als neuen User anmelden. Es ist nicht möglich, als einen Gast mitzuschreiben.

Bei den Userdaten werden nur die Mailadresse sowie das Passwort in der Datenbank gespeichert.

Es werden für die gesamte Projektdokumentation die Standards des Betriebs bzw. der Abteilung verwendet. So wie die Standards Programmiersprachen der Abteilung (Perl und Java). Alle Daten werden in einer zentralen relationalen MySQL-Datenbank abgespeichert.

Folgendes wird in diesem Projekt nicht berücksichtigt.

- Keine Profiländerungen des Benutzers
- Änderungen am Datenbestand durch einen anderen Client wird nicht automatisiert im eigenen Programm aktualisiert
- Die Applikation muss nur auf dem Gerät, welches während der Arbeit benutzt wird, laufen.
- Andere Betriebssysteme oder Geräte werden nicht berücksichtigt.
- Die Versionierung wird nur mittels Gitlab gestattet.

## 4.2 Projektaufbauorganisation

### Mehr infos zu personen

**Auftraggeber:** Remo Steinmann

**Auftragnehmer:** Lara Felix

**Experte:** Remo Steinmann

**Nebenexperte:** Ruwen Wiederkehr und Merjem Hamza

### Versionierung?

### 4.3 Mittel und Methoden

Folgende Mittel werden in diesem Projekt gebraucht:

- Programmiersprache: Java, Perl, SQL
- Datenbank: MySQL
- Betriebssystem: Windows 10
- Versionierungssystem: GitHub Repository und Git
- Diagramme (UML's): diagrams.net
- DIE: Visual Studio Code

Als Projektmanagementmethode wird IPERKA benutzt.

### 4.4 Vorkenntnisse

Ich arbeite in meiner Abteilung hauptsächlich mit den Programmiersprachen Java und Perl. Das verbinden von Frontend (Java) und Backend (Perl) ist mir ebenfalls durch die Arbeit in meiner Abteilung bekannt. Jedoch stehen dort diverse Template für das Frontend mittels Java zur Verfügung. Alles vom Grunde auf neu aufzubauen wird für mich Neuland sein. Das Einbinden einer Datenbank ist mir ebenfalls schon bekannt, dies jedoch auch wieder nur mittels Librarys.

Eine Multiuserapplikation an sich habe ich noch nie erstellt.

Die oben aufgeführten Mittel habe ich alle schon mehrmals eingesetzt.

### 4.5 Vorarbeit

Für das Erstellen des Projektantrags, machte ich mir schon ein paar Gedanken bezüglich des groben Aufbaus. Dies wird im Schritt Planen erwähnt.

### 4.6 Deklaration der benützten Firmenstandards

Es wird die aktuelle Version der Firmenstandards Siemens Mobility benutzt.

## 4.7 Zeitplan

		Montag - 29.11.2021			Dienstag - 30.11.2021			Mittwoch - 01.12.2021			Montag - 06.12.2021			Dienstag - 07.12.2021			Mittwoch - 08.12.2021			
		2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h		
Informieren	Vorkonzept/Projektbeschreibung	soll																	Meilensteine	
		ist																		
Planen	Projektdokumentation aufbau erstellt	soll																	<div>Informieren/Planen und Entscheiden abgeschlossen</div> <div>Grundaufbau der Applikation fertig (Datenbank und Layout)</div> <div>Grundfunktionen der Applikation fertig (Geschichte erweitern und Login)</div> <div>Dokumentation und Applikation abgeschlossen</div> <div>Soll</div> <div>Ist</div>	
		ist																		
	UML's und Mockups erstellt	soll																		
		ist																		
	Gantt - Planung erstellt	soll																		
		ist																		
Entscheiden	Testkonzept erstellt	soll																		
		ist																		
Entscheiden	finale Varianten evaluiert	soll																		
		ist																		
Meilenstein ein	Informieren/Planen und Entscheiden abgeschlossen	soll																		
		ist																		
Realisieren	Grundstruktur Datenbank erstellt	soll																		
		ist																		
	Layout (Grundstruktur) Hauptseite erstellt	soll																		
		ist																		
	Login implementieren und testen	soll																		
		ist																		
	Funktionalität "Geschichte erweitern" implementiert und getestet	soll																		
		ist																		
Realisieren	Funktionalität "Geschichte Export" implementiert und getestet	soll																		
		ist																		
Realisieren	Funktionalität "Zeit Counter" implementiert und getestet	soll																		
		ist																		
Meilenstein	Grundaufbau der Applikation fertig (Datenbank und Layout)	soll																		
		ist																		
Meilenstein	Grundfunktionen der Applikation fertig (Geschichte erweitern und Login)	soll																		
		ist																		
Kontrollieren/Auswerten	Testing durchgeführt	soll																		
		ist																		
Kontrollieren/Auswerten	Reflexion geschrieben	soll																		
		ist																		
Allgemeines	Dokumentation nachführen	soll																		
		ist																		
	Journal weitergeführt	soll																		
		ist																		
Allgemeines	Experten Gespräch	soll																		
		ist																		
Meilenstein ein	Dokumentation und Applikation abgeschlossen	soll																		
		ist																		

## 4.8 Arbeitsprotokoll

### 4.8.1 29.11.2021 – Tag 1

#### Geplant

- Projektdokumentation Aufbau erstellen
- Gantt - Planung erstellen
- Projektjournal Tag 1
- Vorkonzept / Projektbeschreibung
- Gespräch mit HE (Remo Steinmann)

#### Gemacht

- Gantt – Planung
- Projektdokumentation Aufbau
- Projektjournal Tag 1
- Einlesen in FARbeit\_2021 und Kriterienkatalog
- Aufgabenstellung
- Projektaufbauorganisation
- Erstes Gespräch mit Hauptexperten (Remo Steinmann)

#### Probleme

Bei der eigentlichen Umsetzung von den oben genannten Punkten traten keine Probleme auf. Jedoch kam ich nicht so gut voran wie gedacht.

#### Fazit

Folgendes wurde im Gespräch besprochen

- Gantt Planung
  - Schritte mehr aufteilen (z.B. Login planen, Login erstellen, Login testen)
  - Legende im Excel – File und nicht nur in Textform in der Dokumentation
  - Meilensteine Soll/Ist Vergleich zum Beispiel Meilenstein als eigenen Task
  - Jeden Iperka – Schritt aufteilen und nicht zusammenfassen
  - Schriftgrösse von Tasks grösser → Soll/Ist – Boxen im Vergleich zu gross
- Aufgabenstellung
  - Gut, kommt von abgegebener Aufgabenstellung umgeändert in die Dokumentation
- Projektdokumentation Aufbau
  - Glossar, Quellenverzeichnis, Abbildungsverzeichnis und Tabellenverzeichnis kommen in den Punkt Anhang und wird als Punkt 6 gekennzeichnet

Die oben genannten Verbesserungspunkte werden bis morgen Abend 30.11.2021 17.00 Uhr angepasst.



## 4.8.2 30.11.2021 – Tag 2

Geplant

- UML's Mockup
- Testkonzept
- Projektjournal Tag 2
- Varianten evaluieren
- Dokumentation nachführen
  - Teil 1
  - Informieren und Planen Teil 2

Gemacht

- Teil 1 nachgeführt
  - Mittel und Methoden
  - Vorkenntnisse
  - Vorarbeit
- Wireframes
- Ddl
- UseCase (muss noch überarbeitet werden)
- Projektjournal Tag 2

Probleme

Ich konnte mich, wie gestern, nicht so gut konzentrieren. Da wir uns in dieser Konstellation schon lange nicht mehr gesehen haben. Deshalb wurde viel geredet anstatt gearbeitet. Ich hoffe, dass ich dies morgen im Home Office aufholen kann. Falls dies nächste Woche immer noch der Fall sein sollte, werde ich mehr Musik hören, um konzentrierter zu arbeiten und nicht abgelenkt zu werden. Bei dem Use Case war ich mir wegen den Notationen nicht mehr sicher. Dies klärte ich jedoch mit Remo.

Gelerntes

## Input Remo

- Check-Liste für Gespräch mit HEX
  - Aufgeräumt
  - Infrastruktur geprüft (vor ort bzw. zu Hause)
  - HEX als Besucher anmelden
  - Gantt - Planung parat
  - Geht ca. 20 min.
  - Kriterienkatalog ausdrucken
  - Tipps umsetzen
- Use Case
  - Überblick über das Programm geben
  - Muss noch in Text erläutert werden
  - Akteur, Voraussetzung, Kurz Beschreib

Es ist wichtig, dass ich fokussierter arbeite. Dies ist mein Hauptproblem. Ich konnte viel aus diesem Tag mitnehmen insbesondere die Inputs von Remo haben mir auf Hinsicht dieses Projektes sowie auf die IPA sehr geholfen.

Konnte nicht umgesetzt werden und muss morgen nachgeholt werden

- Use Case überarbeiten mit Inputs von Remo
- Testkonzept (Können von Use Case abgeleitet werden)
- Evtl. weitere UML's einfügen bzw. bestehende überarbeiten.

## 4.8.3 01.12.2021 – Tag 3

Geplant

- Nachholen von Gestern
  - Use Case überarbeiten mit Inputs von Remo
  - Testkonzept (Können von Use Case abgeleitet werden)
  - Evtl. weitere UML's einfügen bzw. bestehende überarbeiten.
- Grundstruktur Datenbank erstellt (muss heute fertig sein)
- Layout (Grundstruktur) Hauptseite erstellen
- Login implementieren anfangen
- Funktionalität "Geschichte erweitern" anfangen
- Doku nachführen
- Projektjournal Tag 3
- Zwischengespräche mit Remo

Gemacht

- Projektjournal Tag 3
- Use Case in Textform
- Testkonzept (Können von Use Case abgeleitet werden)

Probleme

Ich konnte mich, wie gestern, nicht so gut konzentrieren. Da wir uns in dieser Konstellation schon lange nicht mehr gesehen haben. Deshalb wurde viel geredet anstatt gearbeitet. Ich hoffe, dass ich dies morgen im Home Office aufholen kann. Falls dies nächste Woche immer noch der Fall sein sollte, werde ich mehr Musik hören, um konzentrierter zu arbeiten und nicht abgelenkt zu werden. Bei dem Use Case war ich mir wegen den Notationen nicht mehr sicher. Dies klärte ich jedoch mit Remo.

Gelerntes

## Input Remo

- Arbeitsjournal
  - Planung innerhalb des Tages mit Zeitangaben

Es ist wichtig, dass ich fokussierter arbeite. Dies ist mein Hauptproblem. Ich konnte viel aus diesem Tag mitnehmen insbesondere die Inputs von Remo haben mir auf Hinsicht dieses Projektes sowie auf die IPA sehr geholfen.

Konnte nicht umgesetzt werden und muss morgen nachgeholt werden

- Use Case überarbeiten mit Inputs von Remo
- Testkonzept (Können von Use Case abgeleitet werden)
- Evtl. weitere UML's einfügen bzw. bestehende überarbeiten.

## 4.8.4 06.12.2021 – Tag 4

Geplant

- Grundaufbau der Applikation fertig (Datenbank und Layout)
- Projektjournal Tag 4
- Verbindung mit DB fixen
- Dokumentation erweitern

Zeit	Was	Status
8.30 – 9.00	Projektjournal Tag 4 erstellen (Geplant)	OK
9.00 – 11.00	Grundaufbau der Applikation fertig (Datenbank und Layout)	OK
11.00 – 12.00	Dokumentation erweitern	OK
13.00 – 14.00	Verbindung mit DB fixen	OK
14.00 – 16.00	Funktionalität "Geschichte erweitern" implementieren	OK
16.00 – 16.30	Dokumentation nachführen und Projektjournal Tag 4 abschliessen	OK

Gemacht

- Projektjournal Tag 4
- Grundaufbau der Applikation fertig (Login/registration, expand und download)
- Verbindung mit DB gefixt

Probleme

Beim Programmieren traten keine grosse Probleme auf. Wie ich die Zwischentabelle tab\_userstory richtig fülle, weiss ich noch nicht genau bzw. ist noch nicht fertig implementiert. Als erstes dachte ich, es wäre eine gute Idee, sie via einem Trigger zu füllen, welcher bei einem neuen Eintrag in die Story-Tabelle ausgelöst wird. Dort habe ich jedoch das Problem, dass ich nicht weiss, welcher User diesen neuen Eintrag erstellt hat. Deshalb wird der Eintrag in die tab\_userstory gleichzeitig mit dem Eintrag in die tab\_story erstellt. Ich habe heute eigentlich keine Zeit für die Dokumentation verwendet, jedoch bin ich gut bei der Applikation vorwärts gekommen, deshalb ist dies oke. Ich werde morgen mehr Zeit für die Dokumentation aufwenden um vor allem auch den Realisieren – Teil anzufangen, denn dieser ist momentan noch ziemlich mager.

Gelerntes

Ich konnte heute meine Java-Kenntnisse auffrischen. Ich arbeite zwar in meiner Abteilung mit Java, jedoch sind dort schon viele Templates und Librarys zur Verfügung. Das ganze also von Grunde auf neu aufzubauen, habe ich schon seit längerer Zeit nicht mehr gemacht.

Ich konnte heute viel fokussierter Arbeiten als die letzten Tage. Ich denke, dass liegt zum einen daran, dass die Anderen auch nicht so viel geredet haben und zum anderen habe ich die meiste Zeit Musik gehört. Die Kopfhörer habe ich auf «Geräuschunterdrückung» eingestellt. So konnte ich mich noch mehr fokussieren. Ich werde dies für den morgigen Tag auf jeden Fall übernehmen, sodass ich fokussierter Arbeiten kann.

Konnte nicht umgesetzt werden und muss morgen nachgeholt werden

- Arbeitsjournal Tag 4 nochmals überarbeiten
- Dokumentation erweitern

## 4.8.5 07.12.2021 – Tag 5

Geplant

- Dokumentation Realisieren-Teil
- Projektjournal Tag 5
- Projektjournal Tag 4 überarbeiten
- Gantt Planung nachführen
- Funktion Zeit-Counter
- Tab\_userstory abfüllen
- Inputs validieren
- Dokumentation überarbeiten
- Testing durchführen
- Gespräch mit HEX (Remo Steinmann)

Zeit	Was	Status
7.30 – 8.00	Projektjournal Tag 4 erstellen (Geplant), Projektjournal 4 überarbeiten und Gantt-Planung nachführen	OK
8.30 – 11.00	Tab_userstory abfüllen, Funktion Zeit-Counter	OK
11.00 – 12.00	Inputs validieren	
13.00 – 14.00	Dokumentation Realisieren - Teil	
14.00 – 16.00	Gespräch mit HEX, Dokumentation Realisieren – Teil, Testing durchführen	
16.00 – 16.30	Dokumentation überarbeiten und Projektjournal Tag 5 abschliessen	

Gemacht

- Projektjournal Tag 5
- Projektjournal Tag 4 überarbeiten
- Gantt Planung nachführen
- Tab\_userstory abfüllen
- 

ProblemeGelerntes

Konnte nicht umgesetzt werden und muss morgen nachgeholt werden

## 5 Teil 2

### 5.1 Kurzzusammenfassung

Ausgangslage

Umsetzung

Ergebnis

### 5.2 Informieren

Folgende Rahmenbedingungen wurde durch den Modulleiter Remo Steinmann vorgegeben

- Objektorientierte Multi-User-Applikation.
- Umfang gemäss Planung (ca. 3 Tage Entwicklung, 2.5 Tage Dokumentation; 6h pro Tag)
- Zentrale Datenbank
- Mehrere Clients müssen gleichzeitig auf den gleichen Datenbestand zugreifen
- Zentrale Benutzer- und Rechte-Verwaltung

#### 5.2.1 Wertebereich

Was	Wertebereich
<b>E-Mail Adresse</b>	muss @ und . beinhalten
<b>Passwort</b>	min. 8 Zeichen
<b>Satz 1</b>	0-15 Zeichen, Buchstabe, Zahlen, Sonderzeichen
<b>Satz 2</b>	0-15 Zeichen, Buchstabe, Zahlen, Sonderzeichen

Ziele der Aufgabenstellung 19

Vorgaben 19

Fragen 19

### 5.3 Planen

## 5.3.1 Zeitplan

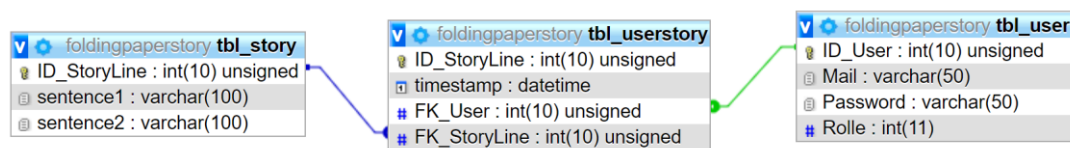
		Montag - 29.11.2021		Dienstag - 30.11.2021		Mittwoch - 01.12.2021		Montag - 06.12.2021		Dienstag - 07.12.2021		Mittwoch - 08.12.2021			
		2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h	2h		
Informieren/Planen/Entscheiden	Vorkonzept/Projektbeschreibung	soil												Meilensteine	
	ist														
	Projektokumentation aufbau	soil													
	ist														
	Storyboard/Mockup	soil												Informieren/Planen und Entscheiden abgeschlossen	
	ist														
	Planung	soil													
	ist														
	Testkonzept	soil												Grundaufbau der Applikation fertig (Datenbank und Layout)	
	ist														
	Grundstruktur Datenbank erstellt	soil													
	ist														
Realisieren	Login fertig	soil												Grundfunktionen der Applikation fertig (Geschichte erweitern und Login)	
	ist														
	Layout (Grundstruktur) Hauptseite fertig	soil													
	ist														
	Funktionalität "Geschichte erweitern"	soil												Dokumentation und Applikation abgeschlossen	
	ist														
	Funktionalität "Geschichte Export"	soil													
	ist														
	Funktionalität "Zeit Counter"	soil													
	ist														
	Testing	soil													
	ist														
Kontrollieren/Reflexion	Reflexion	soil													
	ist														
	Dokumentation nachführen	soil													
	ist														
Abgrenzen	Journal	soil													
	ist														
	Experten Gespräch	soil													
	ist														

## 5.3.1.1 Meilensteine

Meilenstein	Zeitpunkt - Soll	Status / Ist
Informieren, Planen und Entscheiden abgeschlossen	30.11.2021 - 17:00 Uhr	OK 01.12.2021 – 11.00 Uhr
Grundaufbau der Applikation fertig (Datenbank und Layout)	06.12.2021 - 10:00 Uhr	OK 06.12.2021 – 11.00 Uhr
Grundfunktionen der Applikation fertig (Geschichte erweitern und Login)	07.12.2021 - 10:00 Uhr	OK 06.12.2021 – 16.00
Dokumentation und Applikation abgeschlossen	08.12.2021 – 12.00 Uhr	

## UMLs

## 5.3.1.2 Datenbank



tbl_User: In dieser Tabelle werden Benutzer abgespeichert.		
Feld	Propertys	Beschreibung
ID_User	INT, Primary-Key, Not Null, Unique	-
Mail	varchar(50), Not Null	E- Mail des Benutzers
Password	varchar(50), Not Null	Passwort des Benutzers
Rolle	Int (11)	Rolle des Benutzer

**tbl\_user ->(1:n) tbl\_userstory**

Ein Benutzer kann mehrere Einträge in der Geschichte erstellen. Ein Eintrag (an zwei Sätzen) kann nur einem User zugeteilt werden.

tbl_userstory: In dieser Tabelle werden Benutzer abgespeichert.		
Feld	Propertys	Beschreibung
ID_User	INT, Primary-Key, Not Null, Unique	-
Mail	varchar(50), Not Null	E- Mail des Benutzers
Password	varchar(50), Not Null	Passwort des Benutzers
Rolle	Int (11)	Rolle des Benutzer

**tbl\_userstory ->(n:1) tbl\_story**

Ein Benutzer kann mehrere Einträge in der Geschichte erstellen. Ein Eintrag (an zwei Sätzen) kann nur einem User zugeteilt werden.

tbl_story: In dieser Tabelle werden Benutzer abgespeichert.		
Feld	Propertys	Beschreibung

ID_User	INT, Primary-Key, Not Null, Unique	-
Mail	varchar(50), Not Null	E- Mail des Benutzers
Password	varchar(50), Not Null	Passwort des Benutzers
Rolle	Int (11)	Rolle des Benutzer

**tbl\_User ->(1:n) tbl\_userstory**

Ein Benutzer kann mehrere Einträge in der Geschichte erstellen. Ein Eintrag (an zwei Sätzen) kann nur einem User zugeteilt werden.



## 5.3.1.3 Wireframe

Die unten dargestellten Wireframes sind nach ihrer chronologischen Reihenfolge angeordnet und nummeriert. Das Design wird schlicht gehalten.

The wireframes are arranged in a 2x2 grid, numbered 1 to 4 in the top-left corner of each screen.

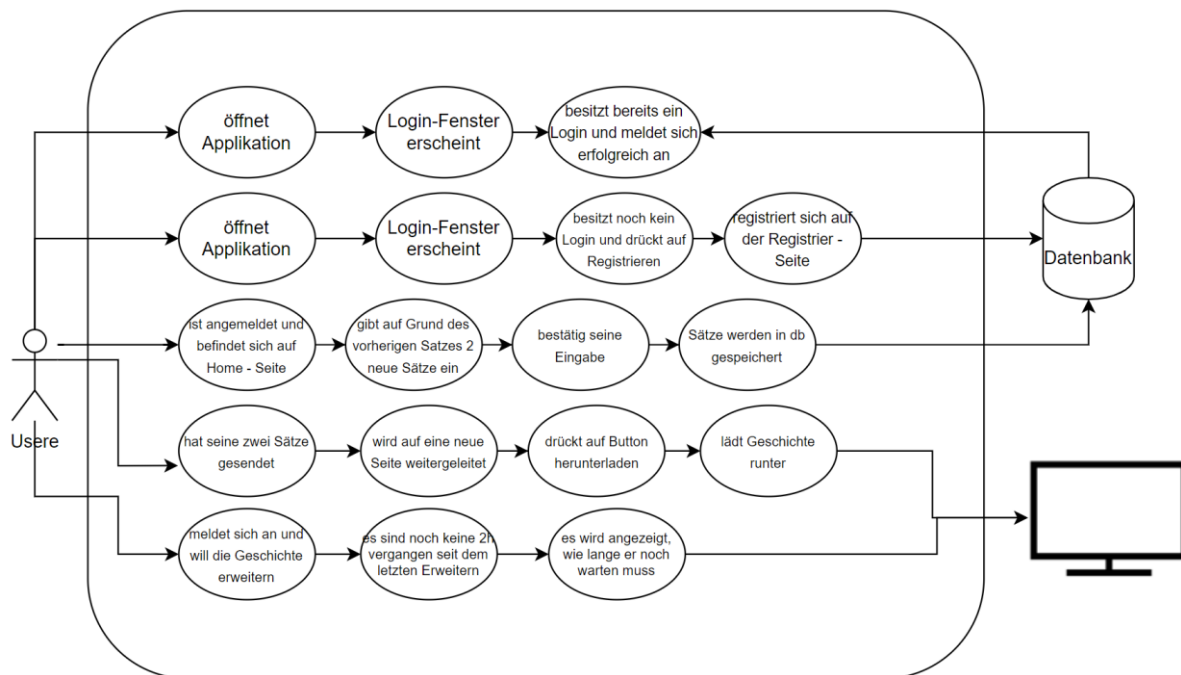
- Screen 1: Login**
  - Top right: [Registrieren](#)
  - Center: **Login**
  - Form fields: **E - Mail Adresse** and **Passwort**, each with a text input box.
  - Bottom center: [Anmelden](#)
- Screen 2: Vorheriger Satz**
  - Top right: [abmelden](#)
  - Center: **Vorheriger Satz** and **Schreibe weiter....**
  - Form fields: **Satz 1** and **Satz 2**, each with a text input box.
  - Bottom center: Zeit Counter and [senden](#)
- Screen 3: Danke für deine Ergänzung :)**
  - Top right: [abmelden](#)
  - Center: **Danke für deine Ergänzung :)**
  - Form fields: [Geschichte downloaden](#) and [weiter](#)
- Screen 4: Schreibe in 24 h weiter**
  - Center: **Schreibe in 24 h weiter**
  - Bottom center: [abmelden](#)

Falls man sich anmeldet, obwohl die **2h** 24h noch nicht vorbei sind, kommt folgendes Fenster, welches einem die verbleibende Wartezeit anzeigt. X steht in der Darstellung unten für die verbleibende Zeit. Das Registrierungsfenster ist gleich aufgebaut wie das Login Fenster.

The wireframes are arranged in a 2x2 grid, numbered 1 to 4 in the top-left corner of each screen.

- Screen 1: Schreibe in x weiter**
  - Center: **Schreibe in x weiter**
  - Bottom center: [abmelden](#)
- Screen 2: Registrierung**
  - Top right: [Login](#)
  - Center: **Registrierung**
  - Form fields: **E - Mail Adresse** and **Passwort**, each with a text input box.
  - Bottom center: [Registrieren](#)

## 5.3.1.4 UseCase



## 5.3.1.5 UseCase Textform

UC1 --> Login	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert.
Beschreibung	Startet Applikation. Es erscheint das Login – Fenster. User gibt korrektes Passwort und E- Mail – Adresse ein. Login – Informationen werden überprüft. Login – Daten sind korrekt. User wird zur Hauptseite weitergeleitet.
Nachbedingung	keine
UC2 --> Registrieren	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzer registriert.
Beschreibung	Startet Applikation. Es erscheint das Login – Fenster. User drückt auf Registrieren – Button oben rechts. Es erscheint das Registrierung – Fenster. User gibt seine E-Mail Adresse und Passwort ein und bestätigt seine Eingabe via. «registrieren»-Button. User wird in der DB gespeichert. User wird zur Hauptseite weitergeleitet.
Nachbedingung	keine
UC3 --> -Geschichte erweitern	

Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert. User ist angemeldet und befindet sich auf der Hauptseite
Beschreibung	User sieht den letzten Satz der Geschichte. Schreibt in die vorgesehenen Inputfelder zwei neue Sätze aufgrund des letzten Satzes der Geschichte. Die Eingabe der neuen Sätze muss innerhalb <b>aks</b> eingegeben werden. Die Eingabe wird mittels «senden»-Button bestätigt und in der Datenbank abgespeichert.
Nachbedingung	keine
<b>UC4 --&gt; Geschichte herunterladen</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert. User ist angemeldet und hat seine zwei Sätze abgesendet.
Beschreibung	User wird nach dem absenden der zwei Sätze auf die Herunterladen – Seite weitergeleitet. Durch den Download - Button hat der User die Möglichkeit die Geschichte herunterzuladen.
Nachbedingung	keine
<b>UC5 --&gt; Geschichte erweitern bevor 2h vorbei sind</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert und hat innerhalb der letzten zwei Stunden die Geschichte bereits erweitert.
Beschreibung	Bevor der User nach dem Anmelden auf die Hauptseite weitergeleitet wird, wird überprüft, ob er innerhalb der letzten zwei Stunden die Geschichte bereits erweitert hat. Falls dies der Fall ist, wird er nicht auf die Hauptseite weitergeleitet. Es erscheint eine Seite mit der Zeit Angabe, wie lange der User warten muss, bis er weiterschreiben darf.
Nachbedingung	keine

### 5.3.2 Testing

<b>TC1 --&gt; Login</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert.
Beschreibung	Startet Applikation. Es erscheint das Login – Fenster. User gibt korrektes Passwort und E- Mail – Adresse ein. Login – Informationen werden überprüft. Login – Daten sind korrekt. User wird zur Hauptseite weitergeleitet.

Nachbedingung	keine
<b>TC1.2 --&gt; Login</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert.
Beschreibung	Startet Applikation. Es erscheint das Login – Fenster. User gibt nicht korrektes Passwort und E- Mail – Adresse ein. Es kommt eine Information, dass die Eingabe nicht richtig war.
Nachbedingung	keine
<b>TC2 --&gt; Registrierung</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzer registriert.
Beschreibung	Startet Applikation. Es erscheint das Login – Fenster. User drückt auf Registrieren – Button oben rechts. Es erscheint das Registrierung – Fenster. User gibt eine valide Eingabe der Email Adresse und des Passworts, gemäss definierter Wertebereich ein und bestätigt seine Eingabe via. «registrieren»- Button. User wird in der DB gespeichert. User wird zur Hauptseite weitergeleitet.
Nachbedingung	keine
<b>TC2.1 --&gt; Registrierung</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzer registriert.
Beschreibung	Startet Applikation. Es erscheint das Login – Fenster. User drückt auf Registrieren – Button oben rechts. Es erscheint das Registrierung – Fenster. User gibt eine invalide Eingabe der Email Adresse und des Passworts, gemäss definierter Wertebereich ein und bestätigt seine Eingabe via. «registrieren»- Button. Es erscheint eine Information, dass die Registrierungsdaten nicht valide sind. Daten werden nicht in der Datenbank erfasst.
Nachbedingung	keine
<b>TC3 --&gt; Geschichte erweitern</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert. User ist angemeldet und befindet sich auf der Hauptseite

Beschreibung	User sieht den letzten Satz der Geschichte. Schreibt in die vorgesehenen Inputfelder zwei neue Sätze, gemäss definierten Wertebereich, aufgrund des letzten Satzes der Geschichte. Die Eingabe der neuen Sätze muss innerhalb <b>aks</b> eingegeben werden. Die Eingabe wird mittels «senden»-Button bestätigt und in der Datenbank abgespeichert.
Nachbedingung	keine
<b>TC3.1 --&gt; Geschichte erweitern</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert. User ist angemeldet und befindet sich auf der Hauptseite
Beschreibung	User sieht den letzten Satz der Geschichte. Schreibt in die vorgesehenen Inputfelder zwei neue Sätze, nicht gemäss definierten Wertebereich, aufgrund des letzten Satzes der Geschichte. Es erscheint eine Information, dass die Eingabe nicht valide ist. Daten werden nicht in der Datenbank erfasst. <b>Zeit counter</b>
Nachbedingung	keine
<b>TC4--&gt; Geschichte herunterladen</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert. User ist angemeldet und hat seine zwei Sätze abgesendet.
Beschreibung	User wird nach dem Absenden der zwei Sätze auf die Herunterladen – Seite weitergeleitet. Durch den Download - Button hat der User die Möglichkeit die Geschichte herunterzuladen.
Nachbedingung	keine
<b>TC5 --&gt; Geschichte erweitern bevor 2h vorbei sind</b>	
Bedingung	Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzer registriert und hat innerhalb der letzten zwei Stunden die Geschichte bereits erweitert.
Beschreibung	Bevor der User nach dem Anmelden auf die Hauptseite weitergeleitet wird, wird überprüft, ob er innerhalb der letzten zwei Stunden die Geschichte bereits erweitert hat. Falls dies der Fall ist, wird er nicht auf die Hauptseite weitergeleitet. Es erscheint eine Seite mit der Zeit Angabe, wie lange der User warten muss, bis er weiterschreiben darf.

Nachbedingung	keine
---------------	-------

## 5.4 Entscheiden

### 5.4.1 Versionsverwaltung

Für die Versionsverwaltung wird Git von GitHub gebraucht. Da ich dies für alle Projekte bei der Arbeit sowie für die Schule brauche bin ich mit Git bereits bekannt. Dieses Projekt wird jedoch nicht in der Siemens Umgebung umgesetzt sondern lokal, deshalb wird nicht das Siemensinterne Git verwendet sondern das kommerzielle GitHub.

GitHub Projektlink: <https://github.com/laraaaf/M223.git>

### 5.4.2 Projekt / Technologie

Folgende Mittel werden in diesem Projekt gebraucht:

- Programmiersprache: Java 11, SQL
- Datenbank: MySQL
- Betriebssystem: Windows 10
- Versionierungssystem: GitHub Repository und Git
- Diagramme (UML's): diagrams.net
- DIE: Visual Studio Code

Anders als im Projektauftrag beschrieben, wird Perl nicht verwendet. Dies ist so, da nicht viel im Backend verarbeitet werden muss. Es gibt lediglich einige Datenbank abfragen, welche gut direkt aus dem Java Frontend gemacht werden können. Falls die Daten zum Beispiel aus der DB oder vom Input mehr verarbeitet werden müssen, wäre ein effektives Backend mittels Perl vorzuziehen.

### 5.4.3 Design

Unsere APP berücksichtigt wegen mangelnder Zeit Barrierefreiheit nicht.

#### Navigation

Die Icons in der Navigation sind so gewählt, dass der User genau weiss, wo was ist und er sich schnell zurechtfindet.

Die Icons werden unabhängig der Bildschirmgrösse horizontal gleichmässig positioniert.

Zudem wird der aktuelle Tab durch dunklere Farbe gekennzeichnet.

#### Filme

Die Filme werden in jedem Tab in Kacheln angezeigt. Diese werden abhängig der Bildschirmgrösse zentriert.

Es wird der Titel und eine kurze Beschreibung des Filmes angezeigt.

Durch ein Dropdown Menü, kann man den Status auswählen (Plan to watch, watched) und die Bewertung abgeben.

#### Suchleiste/ Titel / Untertitel

Die Suchleiste, der Titel sowie der Untertitel sind auf jedem Tab vorhanden und werden ebenfalls zentriert und passen sich so der Bildschirmgrösse an.

#### Farb-/Schriftlegende

Schrift:

- Nunito (Google Fonts)

Farbe:

- #2F2B36
- #3C3746
- #6B6478
- #7840FF
- #C5ACFF
- #ECE4FF
- #7840FF14

#### Namenskonvention

- Variablen und Funktionen werden klein geschrieben (Camel Case)
- Files werden klein geschrieben

### 5.5 Realisieren

Das Projekt inkl. Dokumentation ist unter dem folgenden Link erreichbar: [GIT - REPO](#)

Eine Installationsanleitung ist im Readme.md ersichtlich.

#### 5.5.1 Persistenz

Es werden alle Informationen in der Datenbank gespeichert.

## 5.5.2 Komponenten

### 5.5.2.1 Login

### 5.5.2.2 Registrierung

### 5.5.2.3 Geschichte erweitern

### 5.5.2.4 Geschichte herunterladen

## 5.6 Kontrollieren

Da wir einiges nicht wie geplant umsetzen konnten. Konnten wir gewisse Punkte im Testing nicht erfolgreich erfüllen.

### 5.6.1 Testing

## 5.7 Auswerten

### 5.7.1 Zeitplan (Soll/Ist Vergleich)

Wir waren bis zum letzten Tag immer sehr gut in der Zeitplanung. Durch die Probleme am letzten Tag wurden wir sehr zurück geworfen und konnten so nicht alles erreichen wie gewünscht.

### 5.7.2 Reflexion

Wir sind sehr zufrieden mit unserem Projekt. Wenn wir unserem Projekt eine Note geben würde, werde es mindestens eine 5. Obwohl wir nicht alles so erreicht haben wie geplant haben wir viel Zeit und Mühe in dieses Projekt gegeben und für 2 Tage effektives Programmieren inklusive Doku sind wir zufrieden mit unserem Projekt. Wir waren immer sehr gut im Zeitplan und kamen gut und ohne grosse Probleme voran. Doch am letzten Tag hat uns das Kombinieren von Struktur und API einen Strich durch Rechnung gemacht und das Rendern der Filme uns mehr Zeit gekostet als gedacht. Wir vermuten, dass uns das API durch zu viele Anfragen immer wieder für eine kurze Zeit gesperrt hat. Jedoch steht auf der API Seite, dass das Rate limit disabled ist.

So konnten wir nicht alles erreichen wie wir wollten. Trotzdem versuchten wir das beste daraus zumachen. Durch die gewohnte zusammen Arbeit konnten wir eine gewisse Sicherheit und Kontrolle in den abschliessenden Stress bringen.

Wir können viel von diesem ÜK mitnehmen. Wir konnten unser Wissen bezüglich Projektarbeit und Aufbau einer Doku auffrischen und festigen. Da wir in der TBZ auch oft solche Projekte haben, war es für uns nicht so schwer. Wir können jedoch viel für die IPA mitnehmen und konnten in diesem ÜK nochmals genauer auf für die IPA wichtigen Punkte achten.



5.8 Glossar

5.9 Quellenverzeichnis

<https://reactnative.dev/>

[https://www.themoviedb.org/\\$](https://www.themoviedb.org/$)

5.10 Abbildungsverzeichnis

5.11 Tabellenverzeichnis

5.12 Anhang

*Sind auf dem Git-Repo ersichtlich.*

Testing

Gantt-Diagramm