M223: Multiuser Applikation

Probe IPA

Inhaltsverzeichnis

[2 Vorwort 4](#_Toc89775615)

[3 Organisation der Arbeitsergebnisse 4](#_Toc89775616)

[4 Teil 1 – Umfeld und Ablauf 4](#_Toc89775617)

[4.1 Aufgabenstellung 4](#_Toc89775618)

[4.2 Projektaufbauorganisation 5](#_Toc89775619)

[4.3 Mittel und Methoden 6](#_Toc89775620)

[4.4 Vorkenntnisse 6](#_Toc89775621)

[4.5 Vorarbeit 6](#_Toc89775622)

[4.6 Deklaration der benützten Firmenstandards 6](#_Toc89775623)

[4.7 Zeitplan 7](#_Toc89775624)

[4.8 Arbeitsprotokoll 8](#_Toc89775625)

[4.8.1 29.11.2021 – Tag 1 8](#_Toc89775626)

[4.8.2 30.11.2021 – Tag 2 10](#_Toc89775627)

[4.8.3 01.12.2021 – Tag 3 11](#_Toc89775628)

[4.8.4 06.12.2021 – Tag 4 12](#_Toc89775629)

[4.8.5 07.12.2021 – Tag 5 13](#_Toc89775630)

[5 Teil 2 15](#_Toc89775631)

[5.1 Kurzzusammenfassung 15](#_Toc89775632)

[5.2 Informieren 15](#_Toc89775633)

[5.2.1 Wertebereich 15](#_Toc89775634)

[5.3 Planen 15](#_Toc89775635)

[5.3.1 Zeitplan **Fehler! Textmarke nicht definiert.**](#_Toc89775636)

[UMLs **Fehler! Textmarke nicht definiert.**](#_Toc89775637)

[5.3.2 Testing 20](#_Toc89775638)

[5.4 Entscheiden 22](#_Toc89775639)

[5.4.1 Versionsverwaltung 22](#_Toc89775640)

[5.4.2 Projekt / Technologie 22](#_Toc89775641)

[5.4.3 Design 24](#_Toc89775642)

[5.5 Realisieren 24](#_Toc89775643)

[5.5.1 Persistenz 24](#_Toc89775644)

[5.5.2 Komponenten 24](#_Toc89775645)

[5.6 Kontrollieren 27](#_Toc89775646)

[5.6.1 Testing 27](#_Toc89775647)

[5.7 Auswerten 30](#_Toc89775648)

[5.7.1 Zeitplan (Soll/Ist Vergleich) 30](#_Toc89775649)

[5.7.2 Reflexion 30](#_Toc89775650)

[5.8 Glossar 31](#_Toc89775651)

[5.9 Quellenverzeichnis 31](#_Toc89775652)

[5.10 Abbildungsverzeichnis 31](#_Toc89775653)

[5.11 Tabellenverzeichnnis 31](#_Toc89775654)

[5.12 Anhang 31](#_Toc89775655)

# Vorwort

# 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 |
| 1.0 | 08.12.2021 | * Arbeitsjournal Tag 6 * Überarbeiten ganze Dokumentation * Reflexion * Anhang | LF |
| 0.8 | 07.12.2021 | * Realisieren erstellt * Testing durchgeführt * Arbeitsjournal Tag 5 | LF |
| 0.6 | 06.12.2021 | * Arbeitsjournal Tag 4 | LF |
| 0.4 | 01.12.2021 | * Planen erweitern * Use Case in Textform * Testing erstellt * Arbeitsjournal Tag 3 | LF |
| 0.2 | 30.11.2021 | * UML‘s (UseCase, Wireframe) * Ddl erstellt * Informieren, Planen und Entscheiden erstellt * Arbeitsjournal Tag 2 | LF |
| 0.0 | 29.11.2021 | * Dokumentation inkl. Struktur erstellt * Gantt Planung * Arbeitsjournal Tag 1 | LF |

# Teil 1 – Umfeld und Ablauf

## 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
  + Schrieben 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 besitz, 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.

## Projektaufbauorganisation

Mehr infos zu personen

**Auftraggeber:** Remo Steinmann

**Auftragnehmer:** Lara Felix

**Experte:** Remo Steinmann

**Nebenexpert:** Ruwen Wiederkehr und Merjem Hamza

## 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.

## 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.

## Vorarbeit

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

## Deklaration der benützten Firmenstandards

Es wird die aktuelle Version der Firmenstandards Siemens Mobility benutzt.

## Zeitplan



### 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 |  |

## Arbeitsprotokoll

### 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.

### 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.

### 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 Zeit angaben

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.

### 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 gleichzeit 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

### 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 | OK |
| 13.00 – 14.00 | Dokumentation Realisieren - Teil | OK |
| 14.00 – 16.00 | Gespräch mit HEX, Dokumentation Realisieren – Teil, Testing durchführen | OK |
| 16.00 – 16.30 | Dokumentation überarbeiten und Projektjournal Tag 5 abschliessen | OK |

Gemacht

* Projektjournal Tag 5
* Projektjournal Tag 4 überarbeiten
* Gantt Planung nachführen
* Tab\_userstory abfüllen
* Gespräch HEX (Remo Steinmann)
  + Versionierung – Änderungen nur Änderung von Dokumentation protokollieren
  + Planung nur einmal drin
  + Spezifische Tagesplanung bei Arbeitsjournal bei den vorherigen Tagen nicht im Nachhinein einfügen.
  + Für Morgen Adapter mitnehmen. Testen ob es mit Biemer funktioniert für Präsentation
* Inputs validieren
* Dokumentation Realisieren Teil
* Testing durchgeführt

Probleme

Ich konnte heute fast alles erledigen, was ich geplant habe. Leider nahm das abfüllen der Tab\_UserStory sowie die Funktionen des Zeit Counters mehr Zeit ein als geplant. Deshalb musste ich am Abend zu Hause noch die Dokumentation ein wenig überarbeiten, sodass ich am letzten Tag, also morgen, keinen Stress habe und genügend Zeit habe, alles nochmals anzuschauen.

Die Änderungstipps, welche ich von Remo im Gespräch erhalten habe, setzte ich ebenfalls noch nach dem Gespräch um. So brauchte ich noch mehr ungeplante Zeit. Beim Gespräch ging der Anschluss an den Bildschirm nicht. Für das muss ich morgen noch meine Adapter von zu Hause aus mit nehmen, in der Hoffnung das der Funktioniert.

Gelerntes

Ich konnte heute erneut meine Java-Kenntnisse auffrischen. Zudem brauchte ich heute seit langem wieder mal Regex in Java. Dies bringt mir im Hinblick auf die IPA viel, da ich dort sehr wahrscheinlich auch den Input in den Feldern überprüfen muss und dies ebenfalls mit Regex machen werden.

# Teil 2

## Kurzzusammenfassung

Ausgangslage

Umsetzung

Ergebniss

## 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

### Wertebereich

Es werden folgende Wertebereiche festgelegt. Zudem soll die Eingabe bzw. das Absenden von keiner Eingabe nicht möglich sein. Falls die E- Mail Adresse bereits gebraucht wurde, soll dies ebenfalls mit einem Fehlermeldung an den User mitgeteilt werden.

|  |  |
| --- | --- |
| Was | Wertebereich |
| E-Mail Adresse | muss @ und . beinhalten |
| Passwort | min. 8 Zeichen |
| Satz 1 | 0-25 Zeichen, Buchstabe, Zahlen, Sonderzeichen (. , ! ? -) |
| Satz 2 | 0-25 Zeichen, Buchstabe, Zahlen, Sonderzeichen (. , ! ? -) |

Ziele der Aufgabenstellung 19

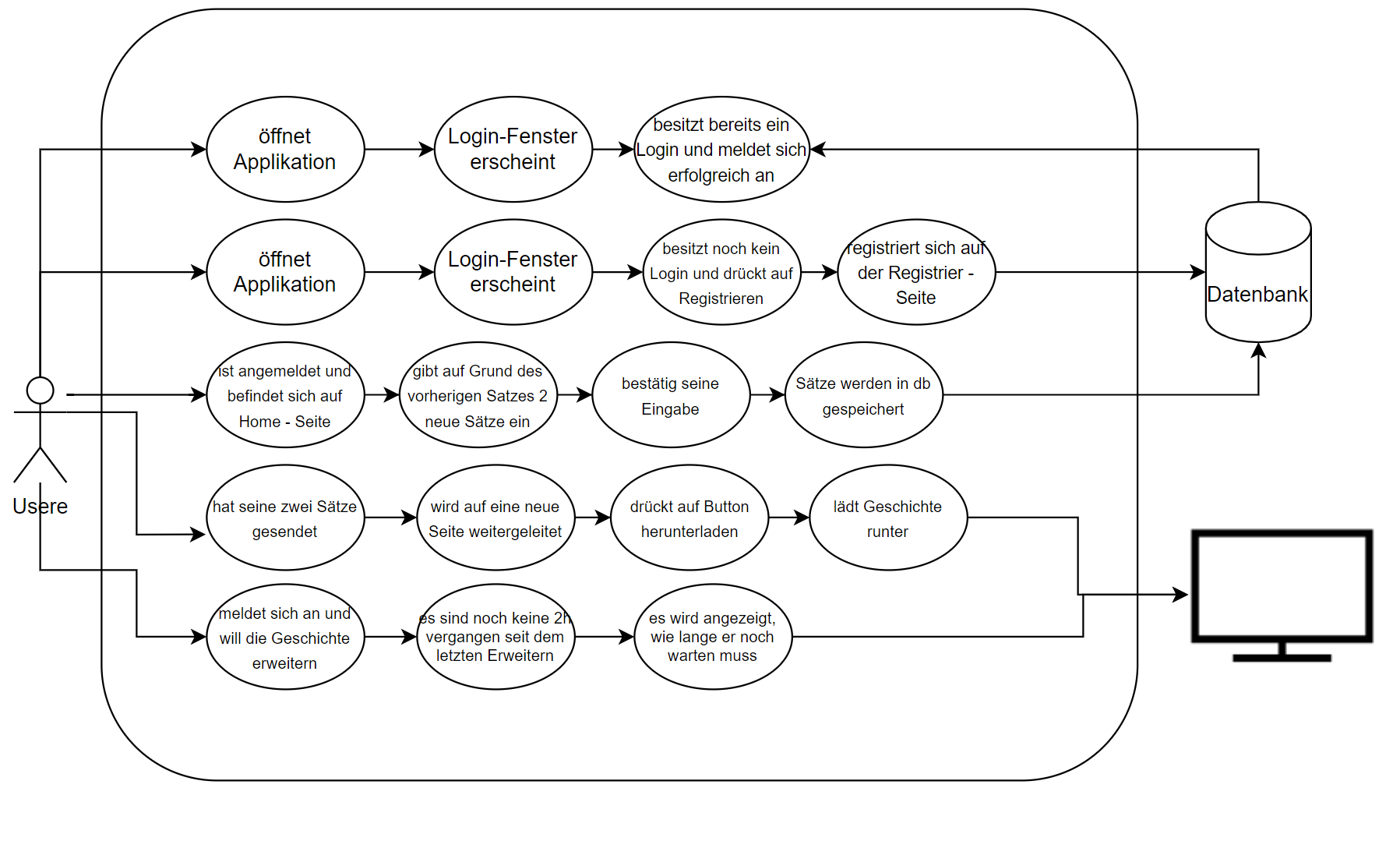
Vorgaben 19

Fragen 19

## Planen

Im nächsten Schritt werden die Funktionen sowie der Aufbau der Applikation definiert.

### UseCase



### UseCase Textform

|  |  |
| --- | --- |
| **UC1 --> Login** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 Benutzter 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 Benutzter 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 inerhalb aks eingegeben werden. Die Eingabe wird mittels «senden»-Button bestätigt und in der Datenbank abgespeichert. |
| Nachbedingung | keine |
| **UC4 --> Geschichte herunterladen** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |
| **UC5 --> Geschichte erweitern bevor 2h vorbei sind** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |

### Backend

Ursprünglich wäre das Backend mit Perl umgesetzt worden. Jedoch werde ich dies nicht machen, da es keine grossen Verarbeitungen der Daten gibt. Die Datenbank abfragen werden direkt aus dem Frontend mit Java gemacht.

#### Datenbank

Die Datenbank ist eine mySQL Datenbank und wird mittels xampp lokal gehostet.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

|  |  |  |
| --- | --- | --- |
| **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 |

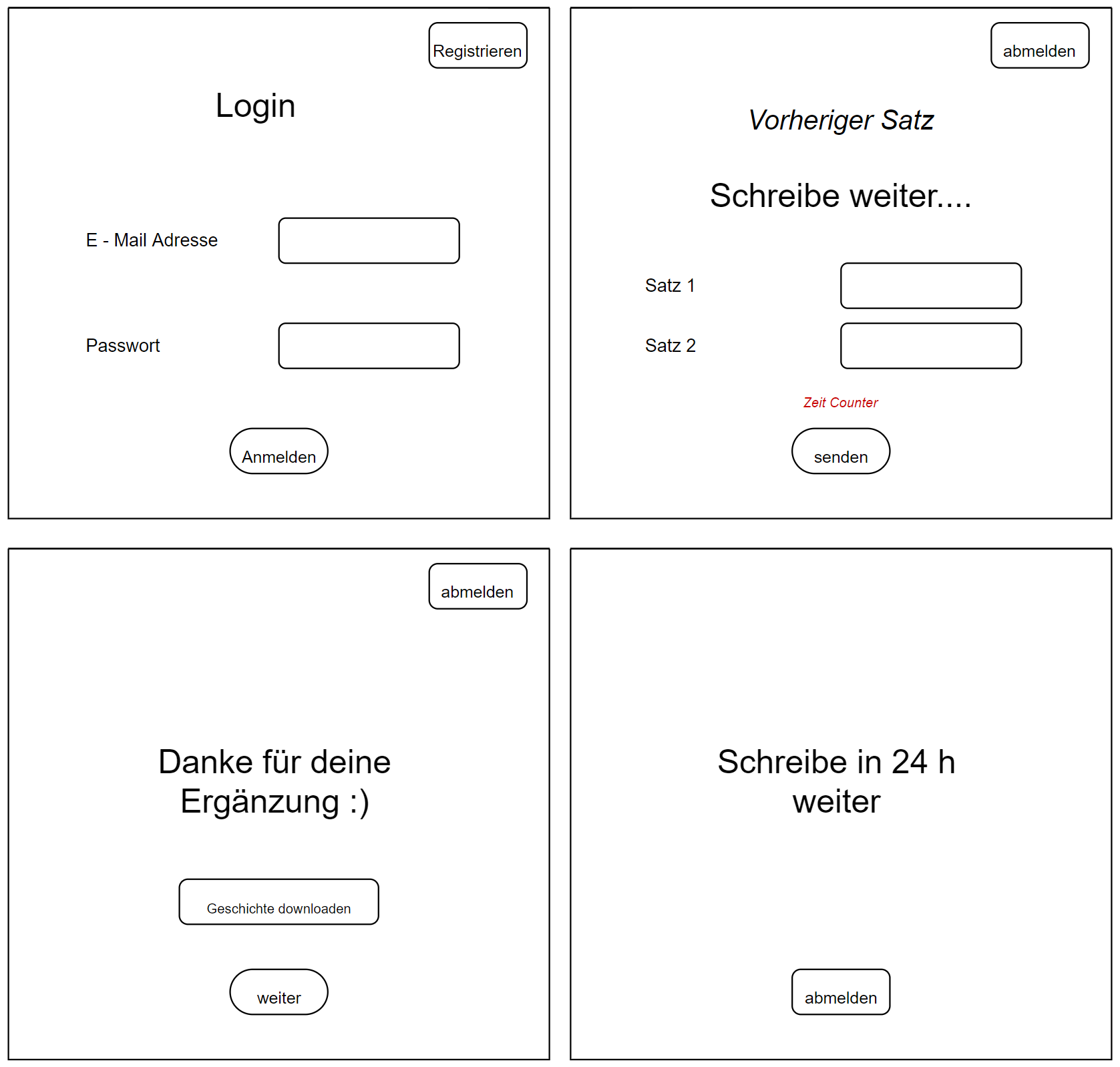
**tbl\_User ->(1:n) tbl\_userstory**Ein Benutzer kann mehrere Einträge in der Geschichte erstellen. Ein Eintrag (an zwei Sätzen) kann nur einem User zugeteilt werden.

### Frontend

Das Frontend wird mittels Java erstellt und verwaltet alle Funktionalitäten der Applikation.

#### Wireframe

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



2

3

4

2

1

Falls man sich anmeldet, obwohl die 2h 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.



### Testing

|  |  |
| --- | --- |
| **TC1 --> Login** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |
| **TC1.2 --> Login** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |
| **TC2 --> Registrierung** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzter 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 Werteberich ein und bestätigt seine Eingabe via. «registrieren»- Button. User wird in der DB gespeichert. User wird zur Hauptseite weitergeleitet. |
| Nachbedingung | keine |
| **TC2.1 --> Registrierung** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzter 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 Werteberich 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 |
| **TC3 --> Geschichte erweitern** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 inerhalb aks eingegeben werden. Die Eingabe wird mittels «senden»-Button bestätigt und in der Datenbank abgespeichert. |
| Nachbedingung | keine |
| **TC3.1 --> Geschichte erweitern** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 nicht valide ist. Daten werden nicht in der Datenbank erfasst. Zeit counter |
| Nachbedingung | keine |
| **TC4--> Geschichte herunterladen** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |
| **TC5 --> Geschichte erweitern bevor 2h vorbei sind** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |

## Entscheiden

### 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 umgsetzt sondern lokal, deshalb wird nicht das Siemensinterne Git verwendet sondern das kommerzielle GitHub.

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.

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

### 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.

### Design

Die Applikation berücksichtigt wegen mangelnder Zeit Barrierefreiheit nicht.

**Navigation**

Oben rechts im Fenster gibt es jeder Zeit die Möglichkeit, sich abzumelden. Beim Fenster Login kann man zum Regsitrieren wechseln. Bei der Registration zum Login. Die Navigation wird simpel gehalten. Die Buttons, mit welche man auf die nächste Seite kommt, befinden sich unten in der Mitte.

**Geschichte**

Die Geschichte wird beim herunterladen in ein Textfile gespeichert Nach jeder Storyline (zwei Sätze eines Users) wird ein Umbruch gemacht. Falls ein Gamemaste die Geschichte herunterlädt, wird am Schluss einer Storyline der User angezeigt, welcher die Line verfasst hat. Die Geschichte wird im Downlaodsordner abgelegt.

**Farb-/Schriftlegende**

Schrift:

* Tahoma
* Times New Roman

Farbe:

* Schwarz
* Weiss
* Rot

**Namenskonvention**

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

## Realisieren

Das Projekt inkl. Dokumentation ist unter dem folgenden Link erreichbar: [GIT - REPO](https://github.com/laraaaf/M223.git)

### Persistenz

Es werden alle Informationen in der Datenbank gespeichert. Die Passwörter werden gehashed. Die Multiuser Funktionalität ist gegeben, da sich diverse User anmelden können und auf den gleichen Datenstand zugreifen.

### Komponenten

Bei jedem Fenster ausser beim Login und Registration, gibt es die Möglichkeit sich abzumelden. Beim Fenster Login kann man zur Registration switchen und beim Fenster Registrieren zum Login Fenster.

#### Login

Die Login Seite dient als Entrypoint der Applikation. Um sich einzuloggen muss man E-Mail Adresse sowie Passwort eingeben. Falls die Anmelde Informationen falsch sind, kommt ein Pop-UP Fenster mit der Information, dass Passwort oder E-Mail Adresse falsch ist. Falls man sich anmeldet, und die letzte Erweiterung die man gemacht hat, noch nicht 2h zurückliegt, kommt ein Fenster mit der Angabe, wie lange man noch warten muss.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### Registrierung

Falls man noch kein Login besitzt, kann man sich über das Registrierungsfenster registrieren. Man braucht eine E-Mail Adresse gemäss dem definierten Wertebereich oben. Zudem wird ein Passwort gesetzt welches gehasht in die Datenbank gespeichert wird. Nach erfolgreichem registrieren, wird man zur Hauptseite (Geschichte erweitern) weitergeleitet.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### Geschichte erweitern

Es wird einem der letzte Satz der Geschichte angezeigt. Falls noch kein Satz in der Geschichte vorhanden ist, wird einem dies mitgeteilt und man schreibt die ersten zwei Sätze der Geschichte. Ansonsten erweitert man die Geschichte aufgrund des letzten Satzen um zwei neue. Seine Eingabe muss man mittels des Senden – Button abschicken. Diese werden in der Datenbank gespeichert. Zudem wird in der tab\_userstory der Vorgang dokumentiert. Das heisst, es wird abgespeichert welcher User welche zwei Sätze erfasst hat. Nach dem Absenden wird der User auf die nächste Seite heruntergeladen.

Ein Bild, das Text, Screenshot, Bildschirm enthält.

Automatisch generierte Beschreibung

#### Geschichte herunterladen

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungAuf dieser Seite hat man die Möglichkeit, die bereits geschriebene Geschichte herunterzuladen. Diese wird in einem Text File in den Downloads Ordner abgelegt. Durch den Button «Weiter» gelangt man auf die nächste Seite. Falls der User ein Game-Master ist, wird ihm in der heruntergeladenen Geschichte angezeigt, welcher User was geschrieben hat.

#### Zeit Counter

Die zwei neuen Sätze müssen innerhalb von zwei Minuten erfasst und abgeschickt werden. Falls dies nicht geschieht, wird man wieder auf die Login Seite geleitet. Durch diese Funktion, soll die Spontanität der Geschichte gefördert werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Kontrollieren

Die Testings sind von den UseCases abgeleitet. Durch das ist gewährleistet, dass alle UseCases ebenfalls erfüllt sind. Pro TestCase gibt es unter TestCase, welche falsch Eingaben bzw. Grenz – Extrem Werte berücksichtigen.

### Testing

Alle Testings sind erfolgreich ausgefallen.

|  |  |
| --- | --- |
| **TC1 --> Login** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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.  E-Mail: [testing@user.ch](mailto:testing@user.ch)  Passwort: Passwort123! |
| Nachbedingung | keine |
| Resultat | Login – Informationen werden überprüft. Login – Daten sind korrekt. User wird zur Hauptseite weitergeleitet. |
| Status | OK |
| **TC1.2 --> Login** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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.  E-Mail: [testing@user.ch](mailto:testing@user.ch)  Passwort: Passwort12! |
| Nachbedingung | keine |
| Resultat | Es kommt eine Information, dass die Eingabe nicht richtig war. |
| Status | OK |
| **TC2 --> Registrierung** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzter 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 Werteberich ein und bestätigt seine Eingabe via. «registrieren»- Button. User wird in der DB gespeichert. User wird zur Hauptseite weitergeleitet.  E-Mail: [testing@[deinName].ch](mailto:testing@[deinName].ch)  Passwort: Testing[deinName]! |
| Nachbedingung | keine |
| Resultat | User wird in der DB gespeichert. User wird zur Hauptseite weitergeleitet. |
| Status | OK |
| **TC2.1 --> Registrierung** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist noch nicht als Benutzter 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 Werteberich 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.  E-Mail: [testing.ch](mailto:testing@[deinName].ch)  Passwort: hallo |
| Nachbedingung | keine |
| Resultat | Es erscheint eine Information, dass die Registrierungsdaten nicht valide sind. Daten werden nicht in der Datenbank erfasst. |
| Status | OK |
| **TC3 --> Geschichte erweitern** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 2 Minuten eingegeben werden. Die Eingabe wird mittels «senden»-Button bestätigt und in der Datenbank abgespeichert.  Satz1: Testing Satz.  Satz2: Yuhui es funktioniert! |
| Nachbedingung | keine |
| Resultat | Die Eingabe wird mittels «senden»-Button bestätigt und in der Datenbank abgespeichert. |
| Status | OK |
| **TC3.1 --> Geschichte erweitern** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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.  Satz1: hallo[]  Satz2: halo ich bin ein viiiiiiiiiiiiiel zu langer Satz. |
| Nachbedingung | keine |
| Resultat | Es erscheint eine Information, dass die Eingabe nicht valide ist. Daten werden nicht in der Datenbank erfasst. |
| Status | OK |
| **TC3.2 --> Geschichte erweitern** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter registriert. User ist angemeldet und befindet sich auf der Hauptseite |
| Beschreibung | User sieht den letzten Satz der Geschichte. Er schreibt uns sendet seine Sätze nicht innerhalb der vorgesehenen Zeit von 2 Min. Er wird nach Ablauf der zwei Minuten auf die Anmelde Seite weitregeleitet. Bereits eingegebenes wird nicht abgespeichert. |
| Nachbedingung | keine |
| Resultat | Es erscheint eine Information, dass die Eingabe nicht valide ist. Daten werden nicht in der Datenbank erfasst. |
| Status | OK |
| **TC4--> Geschichte herunterladen** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |
| Resultat | Geschichte wird als Text File in den Downloadordner gespeichert. |
| Status | OK |
| **TC5 --> Geschichte erweitern bevor 2h vorbei sind** | |
| Bedingung | Applikation wurde betriebsbereit auf dem System installiert. User ist bereits als Benutzter 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 |
| Resultat | Es erscheint eine Seite mit der Zeit Angabe, wie lange der User warten muss, bis er weiterschreiben darf. |
| Status | OK |

## Auswerten

### 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.

### Reflexion

## Glossar

## Quellenverzeichnis

<https://reactnative.dev/>

<https://www.themoviedb.org/$>

## Abbildungsverzeichnis

## Tabellenverzeichnnis

## Anhang

*Sind auf dem Git-Repo ersichtlich.*

Testing

Gantt-Diagram