Dennis Schäppi

dennis.schaeppi@siemens.com

Individuelle produktivarbeit

Internes soziales Netzwerk

Probelauf

# Inhaltsverzeichnis

[1 Inhaltsverzeichnis 1](#_Toc528724410)

[2 Management Summary 2](#_Toc528724411)

[3 Informieren 3](#_Toc528724412)

[4 Planen 4](#_Toc528724413)

[5 Entscheiden 5](#_Toc528724414)

[6 Realisieren 6](#_Toc528724415)

[7 Kontrollieren 7](#_Toc528724416)

[8 Auswerten 8](#_Toc528724417)

# Management Summary

## Aufgabenstellung

### Beschreibung der Applikation

Kleines soziales Netzwerk, bei dem ein Benutzer sich registrieren kann. Er muss bei der Registrierung einen Benutzernamen, eine E-Mail-Adresse, den Vornamen, Nachnamen und ein Passwort in Textform angeben. Freiwillig ist das Setzen eines Profilbildes. Einloggen kann sich der Benutzer ausschliesslich nur mit dem Benutzernamen oder E-Mail und mit dem Textpasswort über ein Login-Fenster mit zwei Eingabefeldern. Als eingeloggter Benutzer kann er einen Textbeitrag verfassen. Er kann beim Erstellen eines Beitrags einen Text mit begrenzter Zeichenzahl schreiben und den Beitrag veröffentlichen. Er kann den Beitrag auch wieder löschen. Dazu kann er seine und andere Beiträge mit «Gefällt mir» markieren, indem er beim Beitrag auf den entsprechenden Knopf drückt und nur mit Text unter dem Beitrag kommentieren. Die Kommentare können auch wieder vom Ersteller gelöscht werden. Die Benutzer sehen alle Beiträge von allen Benutzern. Andere Benutzer können andere Beiträge melden indem sie beim Beitrag auf die Melden-Option klicken. Die Benutzerrollen (Admin, Troubleshooter und Konsument) haben ihr eigenes Panel.

Der Zweck der Applikation ist es, in einem kleinen Unternehmen den globalen Informationsaustausch einfach und abgegrenzt zu ermöglichen.

### Kriterien

#### Funktioneller Umfang

* Multiuser-Applikation
* Benutzerrollen
  + Konsument (Registrierter Benutzer)
    - Konsument kann sich registrieren
      * Pflicht:
        + Benutzername
        + Passwort und Passwort wiederholen

Wird verschlüsselt

* + - * + Passwort wiederholen
        + E-Mail-Adresse
        + Vorname
        + Nachname
      * Kann:
        + Profilbild
    - Konsument kann sich einloggen
      * Textfeld für Benutzername oder E-Mail
      * Passworttextfeld für Passwort
    - Konsument kann sein Profil bearbeiten
      * Profilbild ändern
      * Vorname
      * Nachname
      * Benutzername
      * Passwort
      * E-Mail-Adresse
    - Konsument kann Beiträge in Form von Text verfassen, seine eigenen bearbeiten und löschen.
      * Nur Text im Beitrag
      * Bearbeiten mit Option direkt beim Beitrag
      * Löschen mit Option direkt beim Beitrag
    - «Gefällt-Mir-Funktion» und Kommentarfunktion für alle Konsumenten bei allen Beiträgen
      * Kommentar nur in Form von Text
      * Gefällt-Mir direkt beim Beitrag an- und abwählbar
    - Beiträge können gemeldet werden
      * Werden von Troubleshooter verwaltet
  + Admin
    - Admin hat Zugriff auf alle Hintergrundfunktionen (Eigenes Admin-Panel mit allen Funktionen auf einer Seite):
      * Beiträge ansehen
      * Beiträge löschen
      * Verwaltung der gemeldeten Beiträge
      * Benutzer hinzufügen/löschen/sperren
  + Troubleshooter
    - Troubleshooter hat nur Zugriff auf die Verwaltung der gemeldeten Beiträge (Eigenes Panel)
      * Beiträge ansehen
      * Beiträge löschen
      * Benutzer sperren
  + Passant (Nicht eingeloggter Benutzer)
    - Passant hat keinen Zugriff auf die Funktionen des Konsumenten und wird direkt zur Login-Seite weitergeleitet

#### Technischer Umfang

* Datenbankverbindung (MySQL/JDBC)
* Muss objektorientiert realisiert werden
* Produkt ist eine Java Servlet Page (JSP)

Die einzelnen Schritte müssen geplant, durchgeführt und dokumentiert werden.

### Mittel und Methoden

* 1 Laptop
* Entwicklungsumgebung Eclipse
* Ausgabesprachen HTML und CSS für GUI
* Programmiersprachen Java und JavaScript für die Hintergrundfunktionen
* MySQL-Datenbank für die Datenablagerung
* SQL für die Datenabfrage und Datenmanipulation
* GitHub für das Sichern der Backups für das Projekt

### Vorarbeiten

Im Eclipse muss die Erweiterung für Webapplikationen und der Tomcat-Server installiert, auf die eiwandfreie Funktionalität getestet und konfiguriert werden.

Dokumentation hat die Grundstruktur und der Zeitplan ist erstellt und die einzelnen Schritte geplant.

## Vorkenntnisse

Während des 3. Lehrjahrs bekam ich einen vertieften Einblick in die Webentwicklung mit Java, HTML, CSS und JavaScript mit Verbindung zu MySQL.

|  |  |  |
| --- | --- | --- |
| Sprache | Anwendung | Kenntnis |
| HTML, CSS | In der gesamten Lehrzeit habe ich HTML, CSS immer wieder in Webapplikationen angewendet. | Sehr gut |
| SQL | Seit dem 2. Lehrjahr vertieft in Webapplikationen angewendet | Sehr gut |
| JavaScript | Seit dem 2. Lehrjahr in fast jeder Webapplikation vertieft angewendet | Gut |
| Java | Seit dem 3. Lehrjahr vertieft in Webapplikationen angewendet | Gut |

## Vorarbeiten

Die folgenden Arbeitsschritte wurden als Vorbereitung durchgeführt:

* Eclipse JSP-Erweiterung installiert
* Tomcat-Server für Eclipse installiert
* Einrichten von GitHub mit Eclipse
* Test-Verbindung zu Datenbank
* Test-Verbindung zu Tomcat-Server

## Projektmanagement

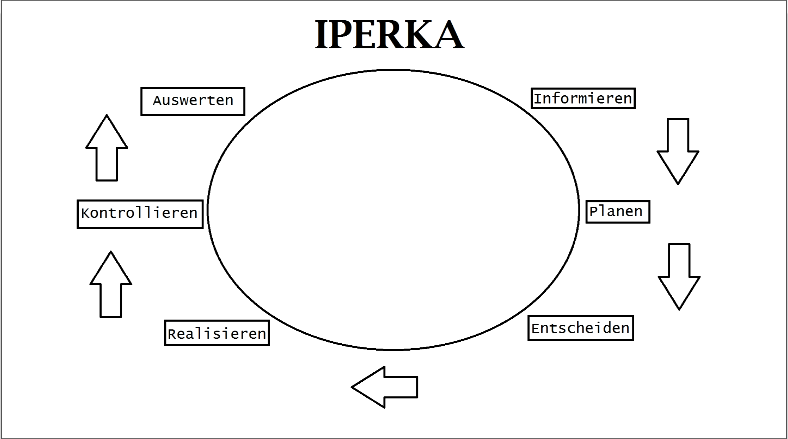
Selber zeichnen!

Abbildung 1 - IPERKA-Modell[[1]](#footnote-1)

In der oberen Grafik (Abbildung 1) sieht man den Strukturierten Ablauf von IPERKA. Der Ablauf ist klar strukturiert, dh. dass zuerst alle Schritte genau geplant werden, bevor sie dann durchgeführt werden. Somit kann man im Verlaufe des Projekts keine neuen Arbeitsschritte hinzufügen und neu planen.

## Zeitplan

### Meilensteine

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Meilenstein | Aufgaben | Datum |
| 1 | Informieren | Die Dokumentation beinhaltet den kompletten Teil, des Informierens. | 31.10.2018 |
| 2 | Planen & Entscheiden | Die gesamte Planung ist in der Dokumentation vorhanden. Dazu gehören das GUI-Mockup, das Klassendiagramm und das Datenbankdesign. | 01.11.2018 |
| 3 | Realisieren | Die Aufgaben orientieren sich an den Teil des Realisierens im IPERKA-Modell |  |
| 4 | Kontrollieren | Die Aufgaben orientieren sich an den Teil des Kontrollierens im IPERKA-Modell |  |
| 5 | Auswerten und Abgabe | Die Aufgaben orientieren sich an den Teil des Auswertens im IPERKA-Modell  Abgabe der Probe-IPA |  |

## Organisation

Die verantwortliche Fachkraft ist für die Zulassung der Aufgabenstellung zuständig. Die Experten sind für die Bewertung dieser Dokumentation, der Präsentation und für das Fachgespräch zuständig.

## Projektübersicht

Diese Dokumentation orientiert sich an den einzelnen Phasen meiner gewählten Projektmanagementmethode:

* Informieren
* Planen
* Entscheiden
* Realisieren
* Kontrollieren
* Auswerten

Diese zusätzlichen Informationen sind in dieser Dokumentation auch vorhanden:

* Arbeitsprotokoll
* Anhang

## Lehrbetrieb

Siemens Schweiz AG  
Freilagerstrasse 40  
8047 Zürich

## Involvierte Personen

|  |  |
| --- | --- |
| Lernender | Dennis Schäppi |
| Verantwortliche Fachkraft / Hauptexperte | Remo Steinmann |
| Berufsbildner | Jonas Knoll |

# Informieren

## Ausgangslage

Die Applikation wird von Grund auf neu Programmiert. Die Entwicklungsumgebung wurde so eingerichtet, dass man direkt mit der Realisierung anfangen könnte.

## Umgebung

* Lokal auf privatem Gerät
* Datenbank läuft lokal unter XAMPP
* Git-Repository auf GitHub

## Rechtschreibung und Grammatik

Für die Rechtschreibung wird das Korrekturprogramm von MS Word verwendet. Zusätzlich lese ich diese Dokumentation auch mehrmals durch und korrigiere alle Fehler manuell.

## Technologien

* Entwickelt mit Java, HTML, CSS und JavaScript
* MySQL

## UseCase

Dieses Use-Case-Diagramm wurde mit MS Visio erstellt und als PNG-Bild exportiert. Visio konnte ich von Microsoft kostenlos herunterladen, da ich mich als TBZ-Schüler mich dort anmelden konnte und wir bekamen sämtliche Office-Programme kostenlos. Visio habe ich schon während meiner ganzen Lehrzeit angewendet und bin mit diesem Tool am meisten vertraut.

Für die Erklärung dieses UseCase-Diagramms nehme ich den Bezug zur Beschreibung in der Applikation in der Aufgabenstellung:

*«Er muss bei der Registrierung einen Benutzernamen, eine E-Mail-Adresse, den Vornamen, Nachnamen und ein Passwort in Textform angeben. Freiwillig ist das Setzen eines Profilbildes. Einloggen kann sich der Benutzer ausschliesslich nur mit dem Benutzernamen oder E-Mail und mit dem Textpasswort über ein Login-Fenster mit zwei Eingabefeldern. Als eingeloggter Benutzer kann er einen Textbeitrag verfassen. Er kann beim Erstellen eines Beitrags einen Text mit begrenzter Zeichenzahl schreiben und den Beitrag veröffentlichen. Er kann den Beitrag auch wieder löschen. Dazu kann er seine und andere Beiträge mit «Gefällt mir» markieren, indem er beim Beitrag auf den entsprechenden Knopf drückt und nur mit Text unter dem Beitrag kommentieren. Die Kommentare können auch wieder vom Ersteller gelöscht werden. Die Benutzer sehen alle Beiträge von allen Benutzern. Andere Benutzer können andere Beiträge melden indem sie beim Beitrag auf die Melden-Option klicken. Die Benutzerrollen (Admin, Troubleshooter und Konsument) haben ihr eigenes Panel.»*

Das UseCase-Diagramm befindet sich auf der nächsten Seite.

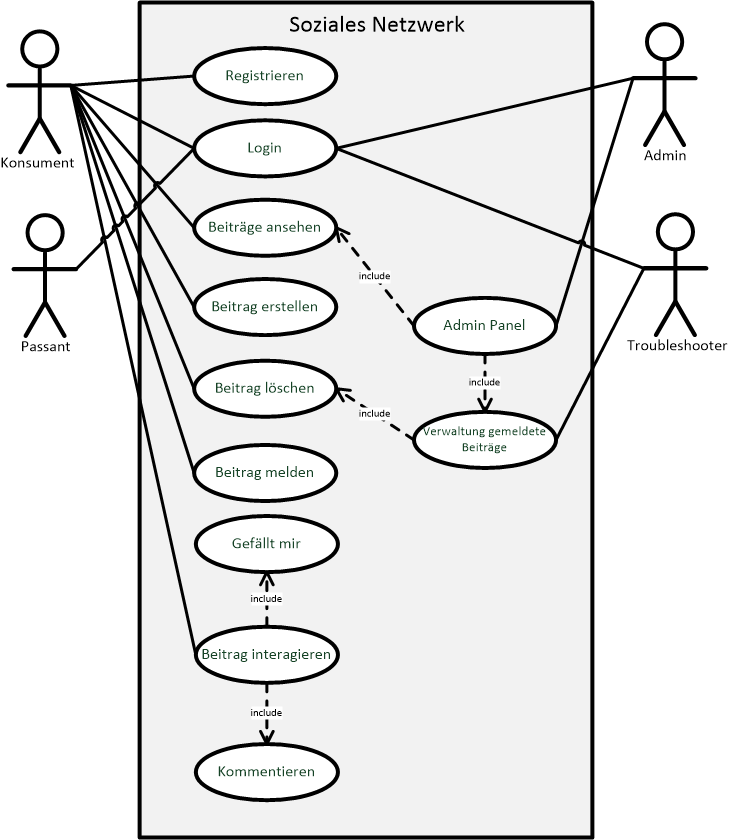


Abbildung 2 - UseCase-Diagramm

## Datenbank

Damit ich das Datenbankmodell erstellen kann, muss ich zuerst analysieren, was alles in die Datenbank kommt und wie es aufgebaut ist. Folgende Tabellen konnte anhand des UseCase-Diagramms (Abbildung 2) definieren:

|  |  |  |  |
| --- | --- | --- | --- |
| Tabelle | Spalte | Datentyp | Einschränkung |
| Benutzerrolle | ID Benutzerrolle | Ganzzahl |  |
| Bezeichnung | Text | Max. 20 Zeichen |
| Berechtigungen | ID Berechtigung | Ganzzahl |  |
| Bezeichnung | Text | Max. 20 Zeichen |
| Zuteilung Berechtigung | ID Zuteilung | Ganzzahl |  |
| FS Benutzerrolle | Ganzzahl |  |
| FS Berechtigung | Ganzzahl |  |
| Benutzer | ID Benutzer | Ganzzahl |  |
| Benutzername | Text | Max. 20 Zeichen |
| Vorname | Text | Max. 30 Zeichen |
| Nachname | Text | Max. 30 Zeichen |
| Email | Text | Max. 50 Zeichen |
| Passwort (Verschlüsselt) | Text | Max. 255 Zeichen |
| FS Benutzerrolle | Ganzzahl |  |
| Gesperrt | Ganzzahl | Nur 0 oder 1 |
| Beitrag | ID Beitrag | Ganzzahl |  |
| Titel | Text | Max. 30 Zeichen |
| Beschreibung | Text | Max. 300 Zeichen |
| Datum des Posts | Datum/Uhrzeit |  |
| FS Benutzer | Ganzzahl |  |
| Gemeldet | Ganzzahl | Nur 0 oder 1 |
| Kommentare | ID Kommentar | Ganzzahl |  |
| FS Beitrag | Ganzzahl |  |
| FS Benutzer | Ganzzahl |  |
| Text | Text | Max. 100 Zeichen |
| Datum des Kommentars | Datum/Uhrzeit |  |
| Gemeldet | Ganzzahl | Nur 0 oder 1 |
| Gefällt Mir | ID Gefällt Mir | Ganzzahl |  |
| FS Benutzer | Ganzzahl |  |
| FS Beitrag | Ganzzahl |  |

Alle Primärschlüssel sind rot markiert und alle Fremdschlüssel sind blau markiert.

## Berechtigungen

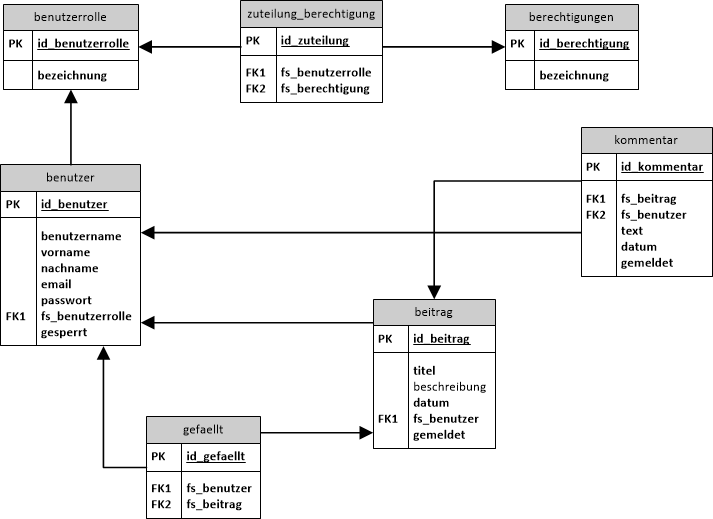
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Passant | Konsument | Admin | Troubleshooter |
| Login-Seite |  |  |  |  |
| Beitragsseite |  |  |  |  |
| Mein Profil |  |  |  |  |
| Registrieren-Seite |  |  |  |  |
| Neuer Beitrag erstellen |  |  |  |  |
| Admin Control Panel |  |  |  |  |
| Troubleshooter Panel |  |  |  |  |

# Planen

## Datenbank ERM

Nach dem ich mich über die nötigen Tabellen und Felder informieren konnte, konnte ich nun das Datenbankmodell erstellen, damit ich beim Realisieren nur noch nach diesem Modell die Datenbank aufsetzen kann. Um das Datenbankmodell besser verstehen zu können, habe ich folgende Legende:

|  |  |
| --- | --- |
| PK | Primärschlüssel |
| FK | Fremdschlüssel |
|  | Die Beziehungsverknüpfung endet mit einem Pfeil, wenn sie auf den Primärschlüssel verweist. (1) |
|  | Die Beziehungsverknüpfung endet mit einer Linie, wenn sie auf die Fremdschlüssel verweist. (n) |
| Normal | Die normal gedruckten Wörter zeigen, dass die Werte in der Spalte NULL sein dürfen. |
| **Fett** | Die fett gedruckten Wörter zeigen, dass die Werte in der Spalte nicht NULL sein dürfen. |



## Stored Procedures

Damit SQL-Funktionen und Java-Code sauber getrennt sind, verwende ich für das Aufrufen der Funktionen Stored Procedures.

|  |  |  |
| --- | --- | --- |
| Name | Beschreibung | Seite |
| loadBenutzer | Lädt alle Benutzerdaten | Login/Mein Profil |
| register | Schreibt den neuen Benutzer in die Datenbank | Registrieren |
| loadBeitraege | Lädt alle nicht gemeldeten Beiträge | Beiträge |
| loadAlleBenutzer |  | Admin Control Panel |
| loadGemeldeteBeitraege | Lädt alle gemeldeten Beiträge | Troubleshooter Panel |
| setGefaellt | Erfasst ein «Gefällt Mir» beim Beitrag | Beiträge |
| entferneGefaellt | Entfernt ein «Gefällt Mir» beim Beitrag | Beiträge |
| meldeBeitrag | Setzt den Beitrag auf gemeldet | Beiträge |
| sperreBenutzer | Setzt den Benutzer auf gesperrt | Admin Control Panel |
| entsperreBenutzer | Setzt den Benutzer auf nichtgesperrt | Admin Control Panel |
| loescheBeitrag | Löscht den Beitrag aus der Datenbank | Beiträge |

## Erlaubte Zeichen in den Feldern

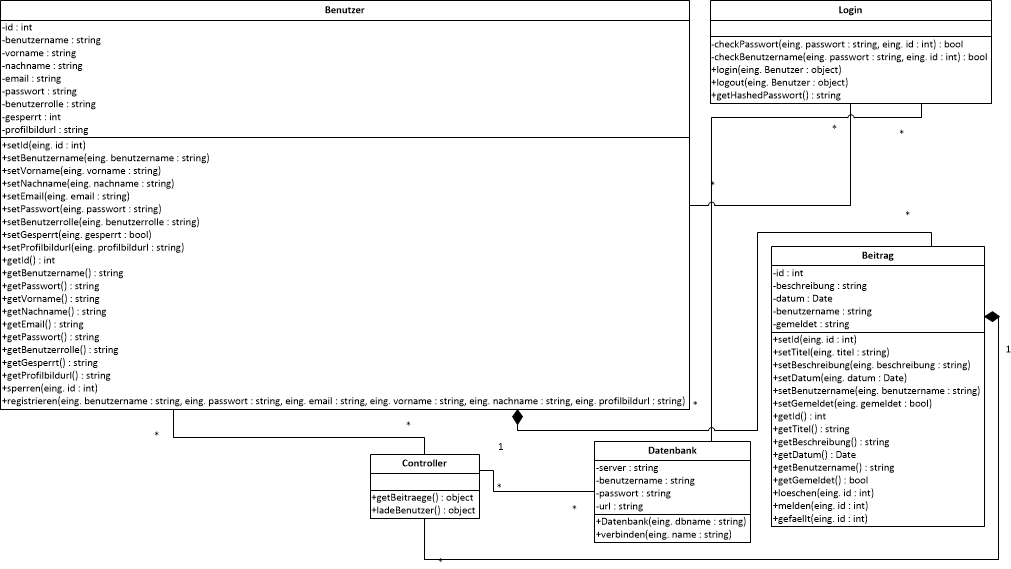
Bei der Registrierung hat der Benutzer bestimmte Einschränkungen bei der Zeicheneingabe:

|  |  |
| --- | --- |
| Feld | Erlaubte Zeichen |
| Benutzername | Nur Buchstaben (ohne äöü) und Zahlen |
| Vorname | Nur Buchstaben |
| Nachname | Nur Buchstaben |
| Email | Nur Buchstaben (ohne äöü), Zahlen, Punkt, Binde- und Unterstrich und @ |
| Passwort | Nur Buchstaben, Zahlen und folgende Sonderzeichen: ?!$&%-\_:. |
| Passwort wiederholen | Nur Buchstaben, Zahlen und folgende Sonderzeichen: ?!$&%-\_:. |

## Back-End Klassendiagramm

Ich habe mich am UseCase-Diagramm orientiert und so das Klassendiagramm erstellt. Zuerst habe ich die Klassen grob definiert und dann für jede Klassen die relevanten Funktionen aufgelistet:

* Benutzer
  + Die Klasse Benutzer speichert alle Benutzerdaten und beinhaltet die Optionen, welche mit dem Benutzer angestellt werden können.
    - Registrieren
    - Sperren
    - Identifizieren
* Beitrag
  + Die Klasse Beitrag speichert alle Informationen eines Beitrags und beinhaltet die Optionen, welche mit dem Beitrag angestellt werden können:
    - Melden
    - Löschen
    - Bearbeiten
    - Gefällt Mir
* Login
  + Login
  + Logout
  + Benutzerdaten überprüfen
  + Passwort verschlüsseln
* Controller
  + Beiträge aus Datenbank laden
  + Benutzer der Beiträge und angemeldeter Benutzer identifizieren
* Datenbank
  + Verbindung zur Datenbank herstellen



## GUI Mockup

### Registrieren

Auf dieser Seite (Abbildung 3) kann der Benutzer folgende Daten eingeben:

* Benutzername (Textfeld)
* Vorname (Textfeld)
* Nachname (Textfeld)
* Email (Textfeld)
* Passwort (Passworttextfeld)
* Passwort wiederholen (Passworttextfeld)

Der Benutzer ist auch bei der Registration dazu in der Lage, ein Profilbild für sein Profil auszuwählen.

Mit einem Klick auf «Senden kann er das Formular abschicken und der Benutzer wird registriert.

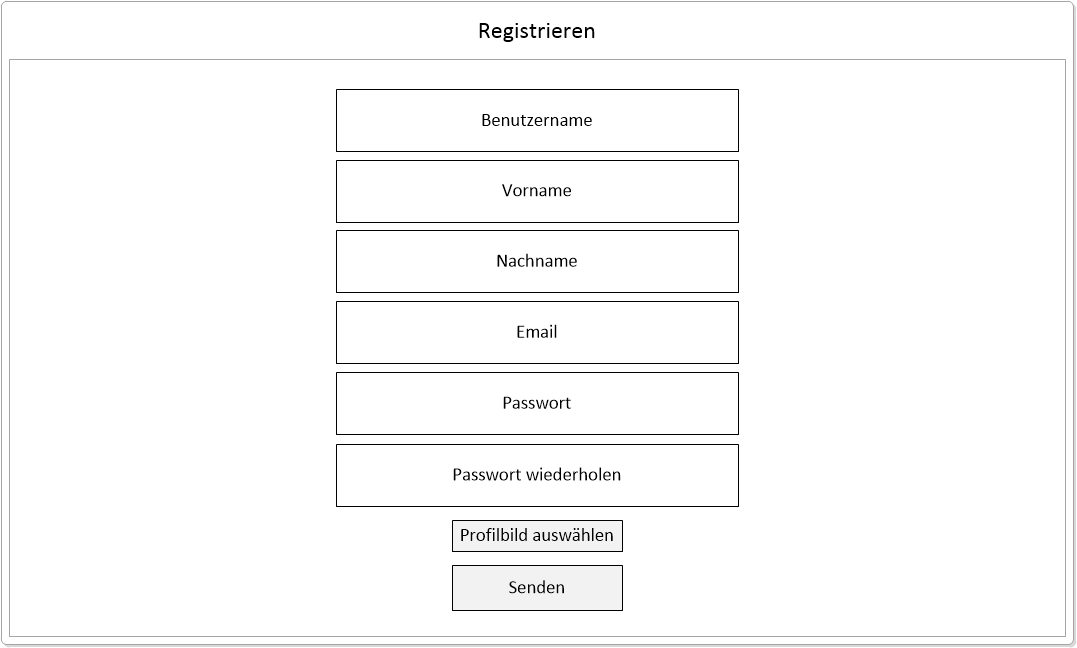


Abbildung - GUI Registrieren

### Login

Auf dieser Seite (Abbildung 4) kann der registrierte Benutzer folgende Daten eingeben:.

* Benutzername oder E-Mail-Adresse (Textfeld)
* Passwort (Passworttextfeld)

Mit einem Klick auf «Login» werden zuerst die angegebenen Daten überprüft und dann bei korrekter Prüfung angemeldet.

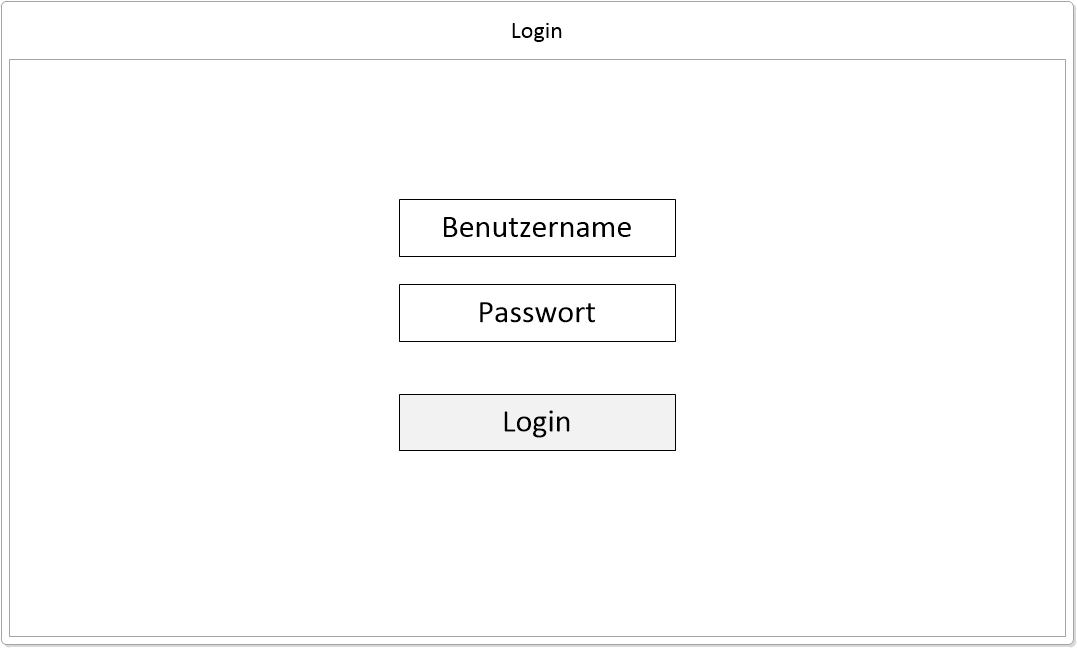


Abbildung - GUI Login

### Beiträge

Auf dieser Seite (Abbildung 5) werden alle Beiträge aufgelistet, die Konsumenten erstellt haben. Ganz oben links beim Beitrag wird angezeigt, wer ihn gepostet hat und wann er gepostet wurde. Oben rechts werden die Optionen angezeigt, jedem Benutzer das Melden und dem Ersteller das Löschen. Unten kann der Benutzer den Beitrag mit «Gefällt Mir» markieren. Ganz unten kann der Benutzer in ein Kommentarfeld (Textfeld) einen Kommentar verfassen.

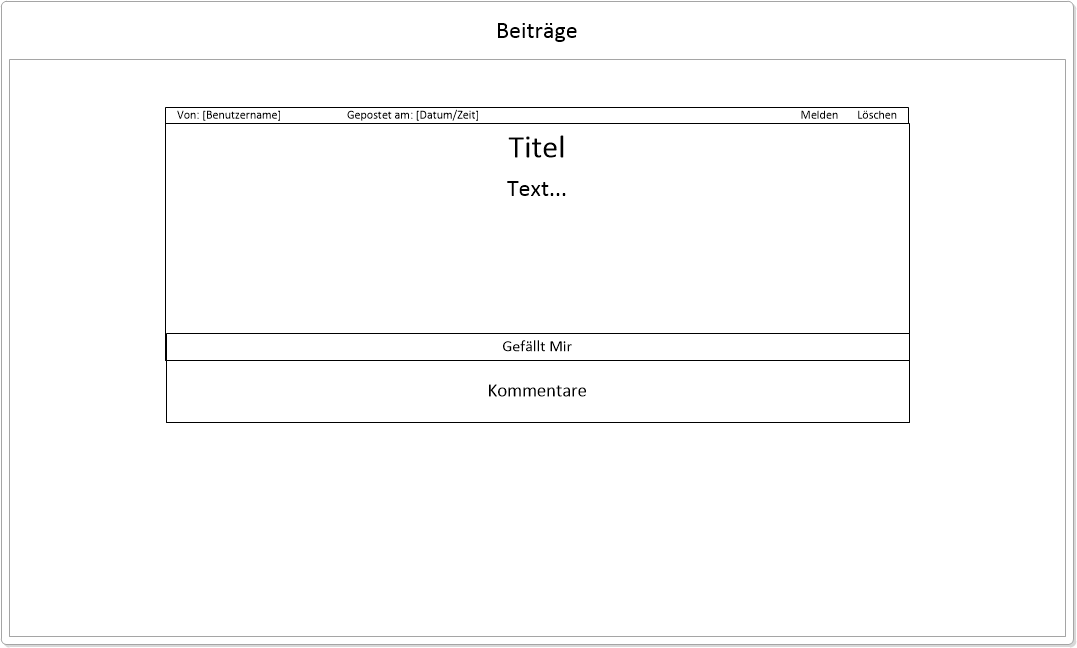


Abbildung - GUI Beiträge

### Neuer Beitrag

Auf dieser Seite (Abbildung 6) kann der Konsument einen neuen Beitrag erstellen. Er kann folgendes definieren:

* Titel (Textfeld)
* Text (Textfeld)

Nachdem der Benutzer den Beitrag fertig geschrieben hat, kann er auf «Erstellen» klicken und der Beitrag wird gepostet.

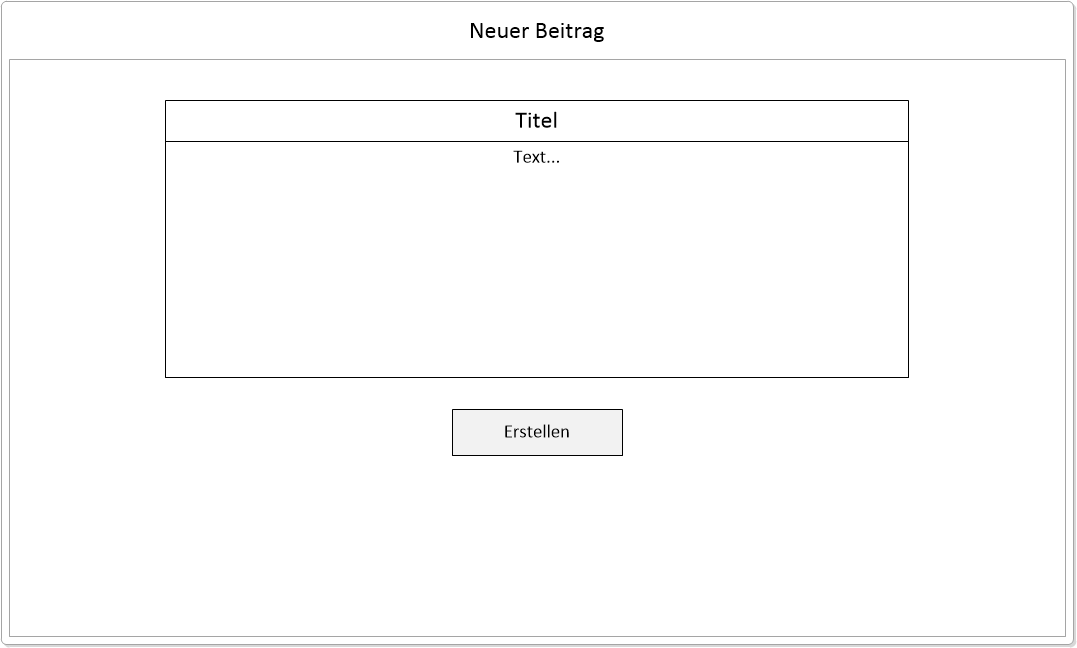


Abbildung - GUI Neuer Beitrag

### Mein Profil

Auf dieser Seite (Abbildung 7) kann der Benutzer sein Profil bearbeiten, dabei kann er folgendes ändern:

* Benutzername (Textfeld)
* Vorname (Textfeld)
* Nachname (Textfeld)
* Profilbild (Image)

Mit einem Klick auf «speichern» kann der Benutzer seine Änderungen speichern.

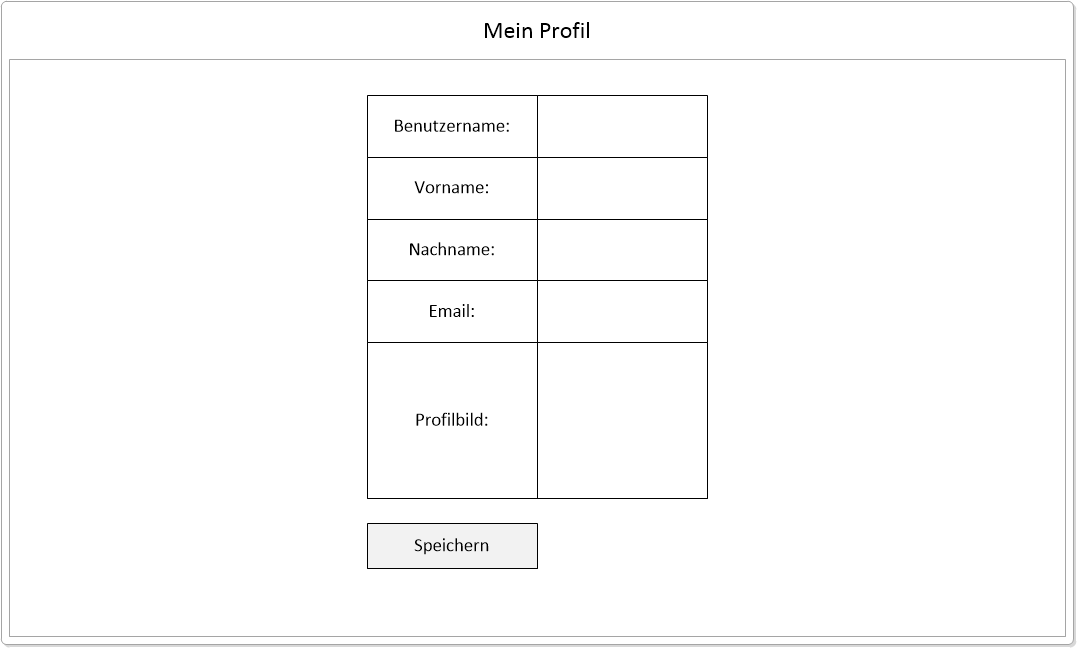


Abbildung - GUI Mein Profil

### Admin Control Panel

Auf dieser Seite (Abbildung 8) kann nur der Admin zugreifen und er sieht eine Auflistung aller Benutzer und eine Liste aller Beiträge. Bei den Beiträgen sieht er auch, ob dieser gemeldet wurde. Der Admin kann Benutzer sperren und Beiträge löschen.

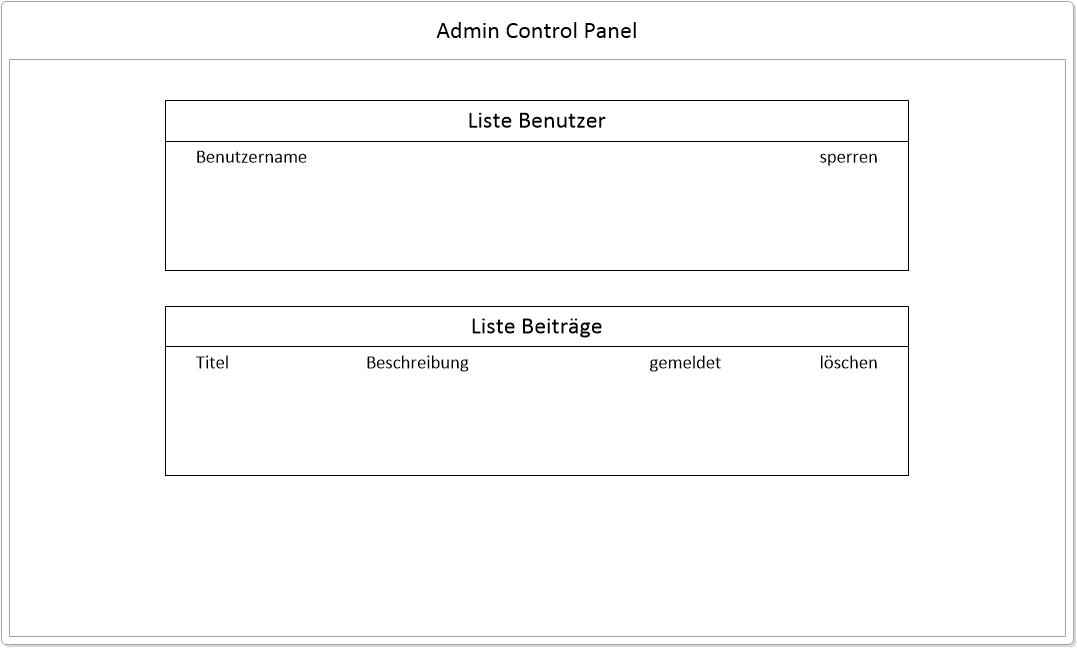


Abbildung - GUI Admin Control Panel

### Troubleshooter Panel

Auf dieser Seite (Abbildung 9) kann nur der Troubleshooter zugreifen und er sieht eine Liste mit allen gemeldeten Beiträgen. Er kann diese löschen.

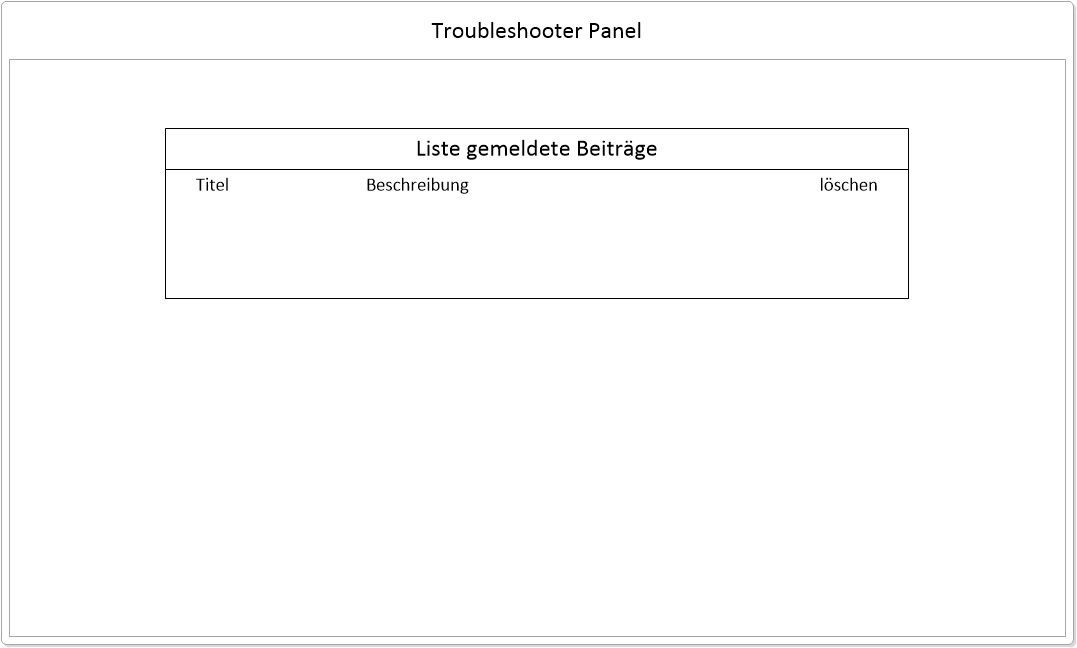


Abbildung - GUI Troubleshooter Panel

# Entscheiden

## Datenbankumgebung

Ich habe die folgenden 2 Möglichkeiten:

* MySQL
* MSSQL

Kriterien:

|  |  |
| --- | --- |
| Kriterium | Gewichtung |
| Bedienbarkeit | 1x |
| Kosten | 2x |
| Unterstützung Java | 3x |

Meine Entscheidung:

|  |  |  |
| --- | --- | --- |
|  | **MySQL** | **MSSQL** |
| Bedienbarkeit | 1 | 1 |
| Kosten | 2 | 0 |
| Unterstützung Java | 3 | 3 |
| **Punkte** | **6** | **4** |

Ich habe mich für MySQL entschieden, weil ich die Probe-IPA auf einem privaten Gerät mache und deshalb eine kostenlose Datenbankumgebung haben will.

## Entwicklungsumgebung

Ich habe bis jetzt mit den folgenden 2 Entwicklungsumgebungen gearbeitet:

* Eclipse
* Visual Studio

Kriterien:

|  |  |
| --- | --- |
| Kriterium | Gewichtung |
| Bedienbarkeit | 1x |
| Kosten | 2x |
| Unterstützung Java | 3x |

Meine Entscheidung:

|  |  |  |
| --- | --- | --- |
|  | **Eclipse** | **Visual Studio** |
| Bedienbarkeit | 1 | 1 |
| Kosten | 2 | 0 |
| Unterstützung Java | 3 | 3 |
| **Punkte** | **6** | **4** |

Ich habe mich für Eclipse entschieden, weil ich die Probe-IPA auf dem privaten Gerät mache und deshalb eine kostenlose Entwicklungsumgebung will.

## Dynamisch und statisch

Ich musste mich für die Art entscheiden, wie ich diese Applikation realisieren werde. Ich hatte die Wahl zwischen einer statischen oder dynamischen Webapplikation.

Kriterien:

|  |  |
| --- | --- |
| Kriterium | Gewichtung |
| Datenbankverbindung | 2x |
| Unterstützung Java | 3x |

Meine Entscheidung:

|  |  |  |
| --- | --- | --- |
|  | **Statisch** | **Dynamisch** |
| Datenbankverbindung | 0 | 2 |
| Unterstützung Java | 0 | 3 |
| **Punkte** | **0** | **5** |

Ich habe mich für eine dynamische Applikation entschieden, weil nur hier das Verbinden zu einer Datenbank möglich ist und bei Dynamisch auch eine serverseitige Programmiersprache eingebunden werden kann.

# Realisieren

## Datenbank aufsetzen

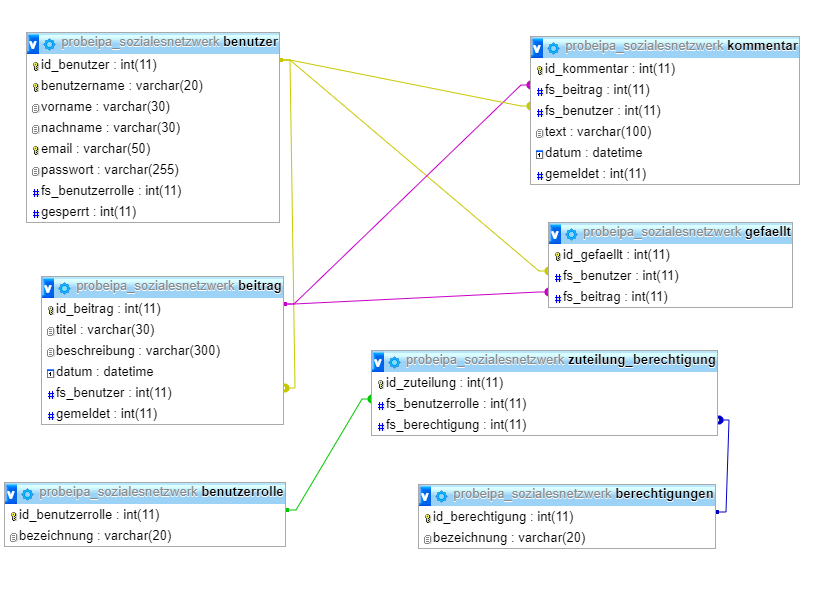
Als erstes fing ich damit an, die Datenbank aufzusetzen. Die Angaben zur Datenbank:

|  |  |
| --- | --- |
| Name der Datenbank: | probeipa\_sozialesnetzwerk |
| Server: | localhost |
| Benutzer: | root |
| Passwort: |  |

Damit es keine Komplikationen mit dem Fremdschlüsseln gibt, habe ich die Tabellen in dieser Reihenfolge erzeugt:

1. Benutzerrolle
2. Berechtigungen
3. Benutzer
4. Beitrag
5. Kommentar
6. Gefaellt
7. Zuteilung\_Berechtigung

Danach musste ich noch die nötigen Beziehungen von Fremdschlüssel zu Primärschlüssel setzen und das ERM sah dann so aus:



Nachdem ich alle diese Tabellen erzeugt habe, musste ich die Grunddaten bei der Tabelle «Berechtigungen» einschreiben, denn das kann später nicht über das GUI gemacht werden.

Folgende Daten waren notwendig für die Tabelle «Berechtigungen», dabei orientiere ich mich aus der Berechtigungsmatrix aus dem Kapitel 3.7:

* Beitrag enfernen
* Beitrag melden
* Beitrag erstellen
* Beitrag bearbeiten
* Benutzer sperren

Auch die Grunddaten für die Tabelle «Benutzerrolle» war notwendig:

* Konsument
* Troubleshooter
* Admin

## Implementierung Frontend

Als erstes habe ich alle HTML-Elemente bei den Seiten hinzugefügt, ohne das CSS zu berücksichtigen. Im zweiten Schritt habe ich das CSS dem Design der Mockups angepasst und dazu habe ich die Backend-Funktionen implementiert.

## Implementierung Backend

Weil ich im Zeitplan ein wenig im Rückstand bin, habe ich mich im ersten Schritt nur auf die wesentlichen Funktionen beschränkt. Zuerst habe ich nachdem Klassendiagramm alle Java-Klassen hinzugefügt.

### Login

Das Login ist das Erste, was ich implementiert habe. Nachdem ich die Klasse erstellt habe, konnte ich anfangen, diese mit Funktionen auszurüsten.

# Kontrollieren

# Auswerten

# Arbeitsprotokolle

## Arbeitsprotokoll vom 31.10.2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Phase | Tätigkeiten | Soll | Ist |
| 1 | Informieren | Aufgabenstellung | 1.5h | 1h |
| 2 | Informieren | Management Summary | 1.5h | 1.5h |
| 3 | Planen | Zeitplan | 1h | 1h |
| 4 | Dokumentation | Informieren dokumentiert und ergänzt | 2h | 3h |
| Total | | | 6h | 6.5h |

### Erfolge und Misserfolge

Heute war mein erster Tag bei der Probe-IPA. Am Morgen erstellte ich einen Zeitplan und überarbeitete meine Aufgabenstellung, da diese noch zu wenig ausführlich war. Die verantwortliche Fachkraft bestätigte mir, dass die Aufgabenstellung so in Ordnung war. Der Start war ein wenig turbulent, da wir am Morgen zuerst den Zeitplan erstellen mussten und erst danach mit der Dokumentation anfangen konnten, weil die verantwortliche Fachkraft dies so haben wollte. Deshalb musste ich die Informieren- und Planen-Phase ein wenig mischen, jedoch half mir der Zeitplan sehr, mich an den bevorstehenden Aufgaben zu orientieren.

### Probleme

Ich habe den Zeitaufwand beim Informieren unterschätzt und plante deshalb auch, heute das GUI-Design, das Use-Case-Diagramm und das Klassendiagramm fertig zu haben.

### Überzeiten

Heute musste ich eine halbe Stunde Überzeit machen, weil ich den Teil des Informierens unbedingt fertig haben wollte.

### Reflexion

Der heutige Tag verlief im Grossen und Ganzen relativ gut. Ich bin im Zeitplan im Rückstand. Ich hätte heute bereits das GUI-Design, das Use-Case-Diagramm und das Klassendiagramm fertig haben sollen.

Die Planung werde ich morgen fertig haben und mit einem kleinen Teil in der Realisierung beginnen können.

## Arbeitsprotokoll vom 01.11.2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Phase | Tätigkeiten | Soll | Ist |
| 1 | Informieren | UseCase-Diagramm | 1h | 1h |
| 2 | Planen | Datenbankmodell | 1h | 2h |
| 3 | Planen | Klassendiagramm | 1.5h | 2h |
| 4 | Planen | GUI Mockup | 1h | 1h |
| 5 | Dokumentation | Planen Grafiken hinzugefügt und teilweise dokumentiert. | 1h | 1h |
| 6 | Expertengespräch | Hilfreiche Tipps bekommen | 0.5h | 0.5h |
| Total | | | 6h | 7.5h |

### Erfolge und Misserfolge

Der zweite Tag der Probe-IPA verlief sehr produktiv. Ich hatte das erste Gespräch mit dem Experten und habe mir alles Wichtige notiert bei der Besprechung, um diese Arbeit noch mehr zu optimieren. Das Gespräch verlief gut. Leider konnte ich ihm noch nicht viel zeigen, aber dafür war das Gespräch nicht zu lange und ich konnte wieder zurück an die Arbeit. Ich konnte das UseCase- und Klassendiagramm, das Datenbankmodell und das GUI Mockup erstellen. Beim Datenbankdesign habe ich sehr viel Zeit verloren, weil ich im Visio immer zwischen den Optionen wechseln musste. Das Gleiche war auch beim Klassendiagramm der Fall. Das Mockup konnte ich schnell erstellen und ich kam beim Mockup gut durch. Die Dokumentation musste ich laufend ergänzen und ich wurde mit dem auch nicht ganz fertig, was bedeutet, dass ich noch mehr Zeit verliere. Ich konnte nicht mit der Entscheidungsphase beginnen.

### Probleme

Das Erstellen des Datenbankmodells und des Klassendiagrams war ein Problem, denn Visio verschob mir alle Elemente immer wieder herum, so dass ich die Dokumente mehrmals wiederherstellen musste. Ich habe die gesamte Zeitplanung unterschätzt und es wird voraussichtlich ein Teil beim Realisieren wegfallen.

### Überzeiten

Heute musste ich anderthalb Stunden Überzeit machen, weil ich beim Datenbankmodell und beim Klassendiagramm den Zeitaufwand unterschätzt habe.

### Reflexion

Der heutige Tag verlief am Anfang gut und am Ende nicht mehr so gut, weil mir ein Teil der Dokumentation fehlte und ich nicht mit dem Entscheiden beginnen konnte, was heute hätte fertig werden müssen. Am Morgen hatte ich ein gutes Expertengespräch, welches nicht zu lange dauerte. Morgen werde im ersten Block die Dokumentation für das Planen und Entscheiden abschliessen und werde im zweiten und dritten Block mit dem Realisieren starten. In dem ich heute sehr viele hilfreiche Diagramme erstellt habe, werde ich genau wissen, was wo implementiert wird und das beschleunigt den Prozess beim Realisieren. Ich darf auch nicht vergessen, immer wieder zu dokumentieren, denn das ist das Wichtigste bei dieser Arbeit. Ich habe geplant, dass ich in der letzten Stunde nur noch dokumentieren werde und das Arbeitsjournal des Tages zu führen.

## Arbeitsprotokoll vom 02.11.2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Phase | Tätigkeiten | Soll | Ist |
| 1 | Entscheiden | Visual Studio oder Eclipse  Dynamisch oder statisch  SQL oder MSSQL | 0.5h | 0.5h |
| 2 | Realisieren | GUI Frontend erstellt | 2h | 1.5h |
| 3 | Realisieren | Backend Login implementiert | 2.5h | 3.5h |
| 4 | Dokumentation | Entscheidung und Prozess dokumentiert | 1h | 1h |
| Total | | | 6h | 6.5h |

### Erfolge und Misserfolge

Heute konnte ich das Planen abschliessen, ich musste noch etwas Kleines im Klassendiagramm ändern. Danach konnte ich direkt mit dem Entscheiden beginnen, wo ich zuerst für die Datenbankumgebung, für die Entwicklungsumgebung und für dynamische oder statische Website. Ich fing danach mit dem Realisieren an, gleich mit dem Aufsetzen der Datenbank, was ohne Probleme funktioniert hat. Ich hatte ein schönes ERM von der Planung vor mir und wusste genau, was ich tun musste. Beim Realisieren hatte ich heute auch einige Misserfolge, denn ich hatte bereits beim Login Probleme mit dem Backend. Der grösste Misserfolg heute war, dass ich zu lange an einem Punkt stecken geblieben bin, dass ich nicht weiterging. Das Gute beim Realisieren heute war, dass ich alle Seiten mit den nötigen HTML-Elementen hinzufügen konnte, ohne das CSS und den Hintergrundfunktionen.

### Probleme

Beim Implementieren des Backends für das Login gab es Probleme bei der falschen Eingabe, ein Error. Ich beharrte darauf, dass ich den Fehler finden musste, dass ich vergessen habe, die anderen fälligen Punkte zu realisieren. Die Folge: Stress und Rückstand. Das Realisieren nach dem Klassendiagram hat einige Abweichungen und deshalb verlangsamt sich dieser Prozess auch, wenn ich die Funktionen ändern muss.

### Überzeiten

Heute musste ich eine halbe Stunde Überzeit machen, weil ich beim Login zu sehr am Backend hängengeblieben war.

### Reflexion

Der heutige Tag verlief beim Entscheiden gut und danach eher schlechter. Ich hing zu lange am Login mit dem Backend fest, so dass ich vergessen habe, dass an einer IPA nicht alles perfekt sein muss, sondern entweder realisiert oder dokumentiert. Ich werde im nächsten Schritt alle Nebenfunktionen streichen und nur das Wichtigste machen. Jetzt, wo ich die erste Datenbankabfrage gemacht habe, habe ich eine Vorlage für alle anderen Abfragen, die bevorstehen, hier werde ich viel Zeit sparen können und jetzt bin ich wieder mit Programmieren routinierter, denn ich habe in den letzten zwei Tagen fast nur Word, Excel und Visio gesehen. Ich werde aber auch genügend Zeit für das Kontrollieren und Auswerten investieren, am besten einen halben oder ganzen Tag.

# Anhang

1. Quelle: https://projekte.bbbaden.ch/2010/12/23/projekttag-der-film/iperka/ [↑](#footnote-ref-1)