

Software Requirements Specification

SRS

Rate Presentations

Vincent Schauer, Lucas Thiess, Christoph Vorlen, Daniel Ziegler

Inhalt

1.	Beschreibung/Zweck/Ziele	3
1.1.	Aktuelle Basis	3
1.2.	Aktuelles Seitenlayout	3
1.2.1.	Homepage	3
1.2.2.	Projektbeschreibung	4
1.2.3.	Login	4
1.2.4.	Als Vortragender eingeloggt	4
1.2.5.	Als Student eingeloggt	5
1.3.	Das wird umgesetzt	5
2.	Funktionale Anforderungen	6
2.1.	Must Have	6
2.2.	Should Have	6
2.3.	Could Have	6
2.4.	Won't Have	6
3.	Nicht Funktionale Anforderungen	6
3.1.	Must Have	6
3.2.	Should Have	6
3.3.	Could Have	6
4.	Use Case Diagramm	7

1. Beschreibung/Zweck/Ziele

Bei diesem Projekt handelt es sich um eine Website, die dazu verwendet wird, die Präsentationen von Präsentationsteams zu bewerten – direkt nach deren Präsentation. Dabei bezieht sich die grundsätzliche Idee auf Studenten, es könnte allerdings in Zukunft auch für die Arbeitswelt Einsatz finden. In den weiteren Beschreibungen bezieht sich alles auf Studenten und Lehrende.

Jeder Student des Publikums bewertet die gerade gesehene Präsentation zusätzlich zur Bewertung des/der Lehrende(n). Dies ermöglicht einerseits den Präsentationsteams eine Möglichkeit einfach Feedback zu erhalten. Andererseits erhält der/die Lehrende eine Rückmeldung, wie andere Personen im Publikum die Präsentation gerade empfunden haben.

Ziel des Projekts im Fach „Software Engineering“ ist es, den Prototypen aus dem Fach „Web Technologies“ weiterzuentwickeln.

1.1. Aktuelle Basis

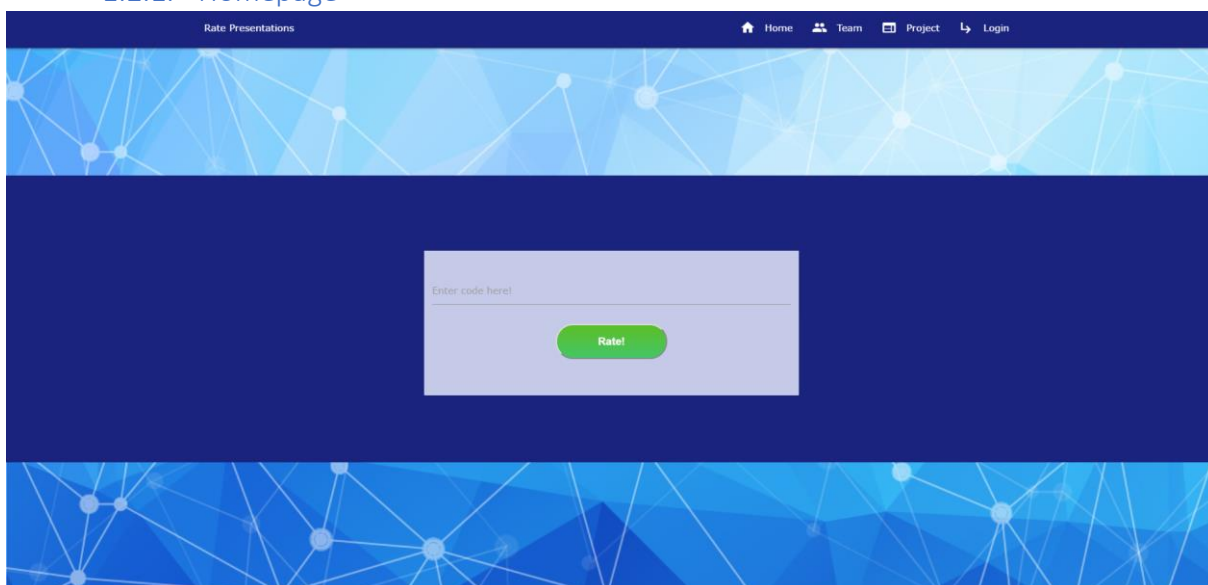
Der aktuelle Entwicklungsstand umfasst folgende Punkte und dient als Basis für das weitere Projekt:

- Vollständiges Design der Website mit MaterializeCSS
- Login und Logout funktioniert für Studentenaccounts und Lehreraccounts
- Lehrende können Accounts erstellen
- Studenten können für eine Präsentation in die Datenbank eingefügt werden
- Einige Funktionen sind vom Konzept her da, allerdings noch nicht vollständig funktionsfähig und wurden für die Präsentation der Website im Fach „Web Technologies“ mit angenommenen Werten befüllt und ausgegeben oder als Prototyp bezeichnet und nicht weiter dargestellt (Teile der Funktionen wurden daher zu diesem Zeitpunkt (SS2019) hard-gecoded)
- Der Fragebogen zum Bewerten ist vollständig implementiert
- Lehrer- und Studentenbereich ist vom Konzept her vorhanden und implementiert

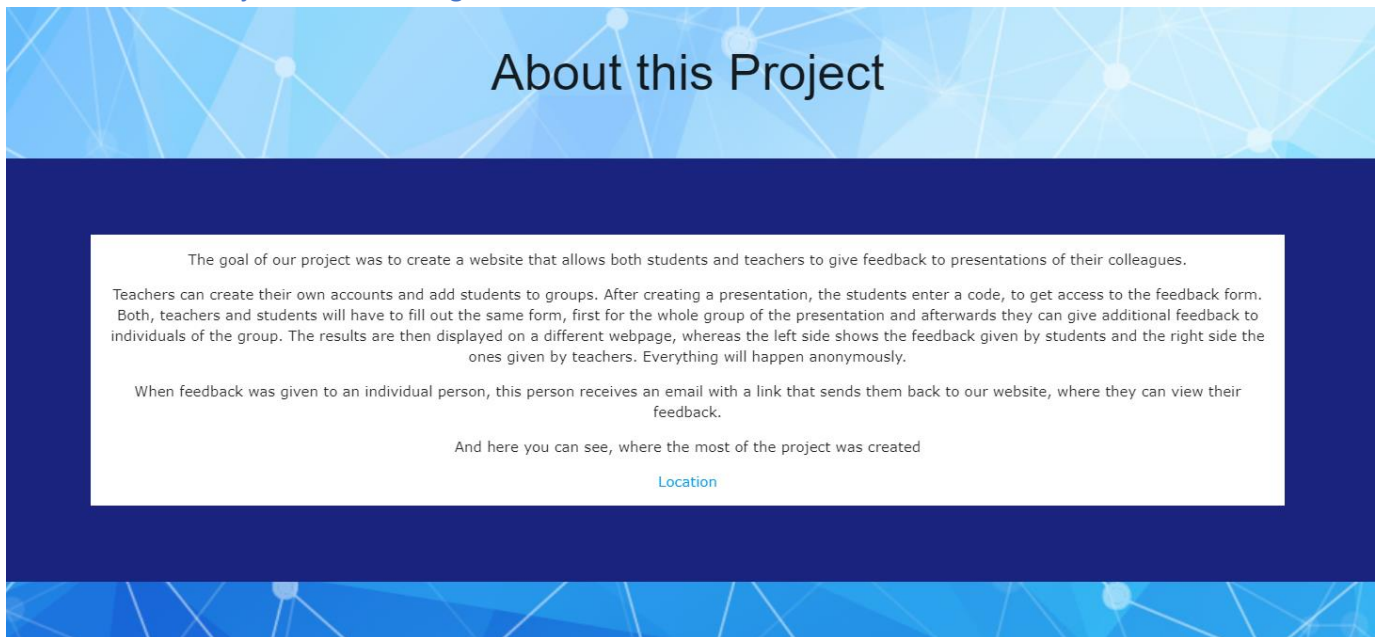
In den folgenden Bildern sind die Website zum Zeitpunkt vor Start des Projektes im Fach „Software Engineering“ dargestellt:

1.2. Aktuelles Seitenlayout

1.2.1. Homepage



1.2.2. Projektbeschreibung



About this Project

The goal of our project was to create a website that allows both students and teachers to give feedback to presentations of their colleagues.

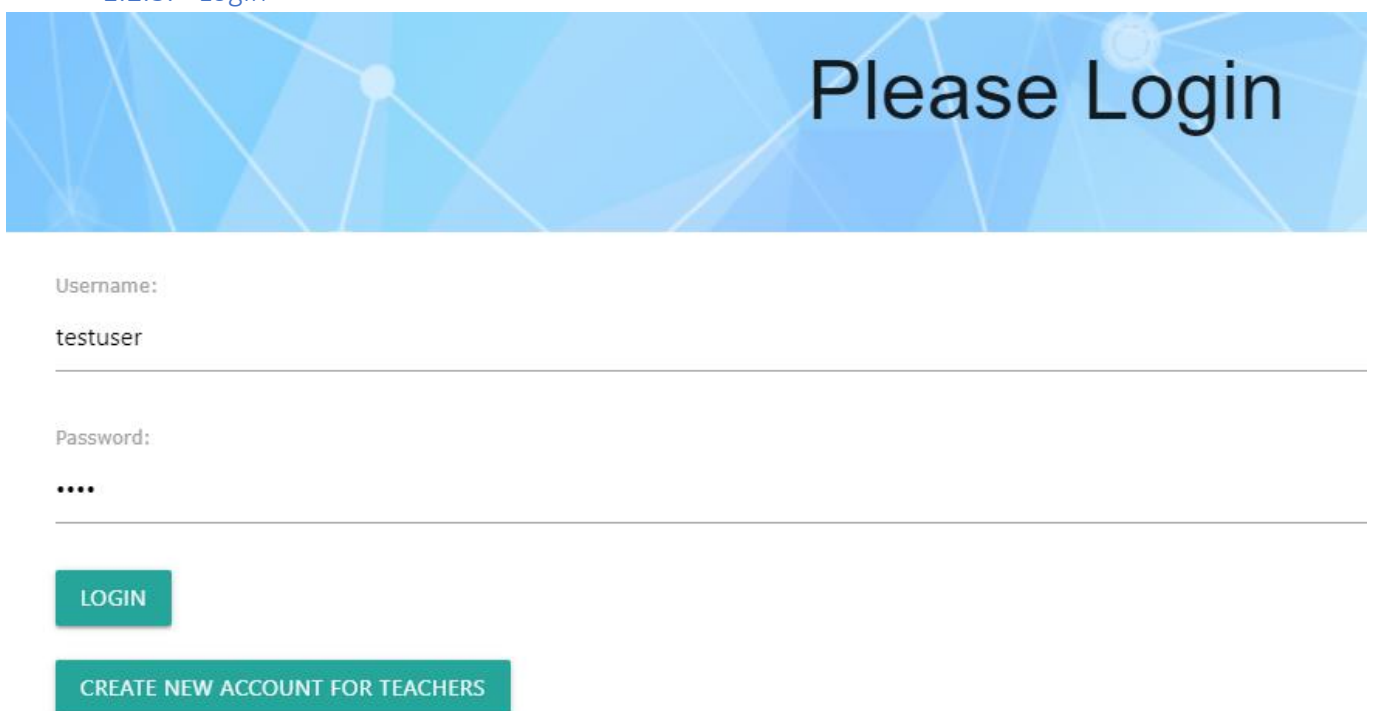
Teachers can create their own accounts and add students to groups. After creating a presentation, the students enter a code, to get access to the feedback form. Both, teachers and students will have to fill out the same form, first for the whole group of the presentation and afterwards they can give additional feedback to individuals of the group. The results are then displayed on a different webpage, whereas the left side shows the feedback given by students and the right side the ones given by teachers. Everything will happen anonymously.

When feedback was given to an individual person, this person receives an email with a link that sends them back to our website, where they can view their feedback.

And here you can see, where the most of the project was created

[Location](#)

1.2.3. Login



Please Login

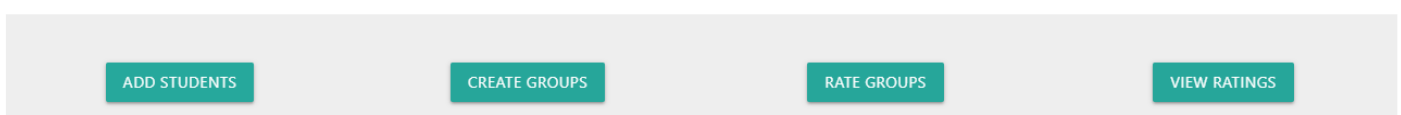
Username: testuser

Password:

LOGIN

CREATE NEW ACCOUNT FOR TEACHERS

1.2.4. Als Vortragender eingeloggt



ADD STUDENTS CREATE GROUPS RATE GROUPS VIEW RATINGS

1.2.5. Als Student eingeloggt

A screenshot of a web application interface. In the center, there is a dark teal rectangular button with the text 'VIEW RATINGS' in white, uppercase letters. The button is set against a light gray background.

1.3. Das wird umgesetzt

Folgende Anforderungen sollen im Rahmen des Projektes von „Software Engineering ILV“ umgesetzt werden:

- Das Design mittels Materialize soll in Bootstrap umgesetzt werden. Grund dafür ist, dass es für Bootstrap viel mehr Dokumentationen gibt, um bei Fehlermeldungen besser Lösungen zu finden
- Website soll sicher gemacht werden
 - SQL injection sicher
 - Password encrypted
- Datenbankstruktur überarbeiten
- Nicht funktionsfähige bzw. nicht vollständig implementierte Funktionen ausarbeiten
 - Logik für das Anzeigen der berechneten Ratings (View Ratings) implementieren
- Im Bereich der Lehrenden – Zufallscode der Präsentation anzeigen lassen
- Website online stellen
- Studenten hinzufügen – Studenten werden beim Hinzufügen der Lehrer in der Datenbank gespeichert und erhalten ein Zufallspasswort – per Email erhält der Student eine Benachrichtigungsmail mit einem Link, wo er das erstellte Zufallspasswort verändern kann

Im nächsten Schritt werden die hier angeführten Anforderungen in funktionale und nicht funktionale Anforderungen unterteilt.

2. Funktionale Anforderungen

2.1. Must Have

- **Feature zum Erstellen von Gruppen**
Studenten lassen sich bereits in die Datenbank speichern und in einer Tabelle darstellen. Diese Studenten sollen ausgewählt werden können und einer Gruppe zugeordnet werden können.
- **Zufallscodes erstellen**
Wenn der erste Punkt funktioniert, soll für eine dargestellte Gruppe, einen ausgewählten Fragebogen/Ratingbogen und ein definiertes Datum ein Zufallscod erstellt werden. Dieser soll für den Lehrenden beim Name der Gruppe und dem Datum dargestellt werden.
- **Logik vom Anzeigen von bewerteten Präsentationen anpassen**
Dieser Teil wurde beim Prototypen hardgecodet und soll vollkommen funktionstüchtig eingebaut werden
- **Bewertungen darstellen**
Auswertung der abgegebenen Bewertungen der Studenten und Lehrenden

2.2. Should Have

- User können Passwörter zurücksetzen
- Website online stellen um diese im großen Rahmen vorzustellen

2.3. Could Have

- Präsentationen können in einen Kalender hinzugefügt werden

2.4. Won't Have

- Raumplan der FH Campus – dabei ist gemeint, dass anstehende Präsentationen und der Raum in dem diese stattfinden auf der Website angezeigt werden

3. Nicht Funktionale Anforderungen

3.1. Must Have

- Neues modernes Design
- Zufallscodedarstellung – Beschreibung bei funktionalen Anforderungen

3.2. Should Have

- Website online veröffentlichen

3.3. Could Have

- Website gegen SQL-Injections sicher machen
- Nur auf gewisse Seiten zugreifen können, wenn (richtiger/jeweiliger User) eingeloggt

4. Use Case Diagramm

