

Pflichtenheft Badger

SWP-OP | 5AHWII | 2020/21

Mario Petrovic

HTL ANICHSTRASSE | WIRTSCHAFTSINGENIEURSWESEN & BETRIEBSINFORMATIK

Inhalt

Inhalt

Inhalt.....	1
Mitwirkende	2
Projektbeschreibung	2
Ziel	2
Web-Sockets.....	3
Ablauf des Projekts.....	3
Rahmenbedingungen	4
Technologien	4
Screenshots	5
Meilensteine.....	6

Mitwirkende

Name	Rolle	E-Mail
Sabo Rubner	Betreuung	s.koelloe@tsn.at
Mario Petrovic	Entwicklung	mpetrovic1@tsn.at

Projektbeschreibung

Badger ist eine web-basierende Plattform für Textkommunikation, welche von einem Schüler der Abschlussklasse der HTL Anichstrasse entwickelt wird. Als Inspiration dient hierbei die Chatting-App WhatsApp. Die Motivation für das Projekt ist die Sammlung von Erfahrung im Bereich Web-Development und Client-Server-Kommunikation.

Ziel

Schlussendlich soll eine Webseite entstehen, über die sich mehrere Nutzer mittels Accounts einloggen und miteinander kommunizieren können. Der Nutzer kann hierbei Kontakte zu anderen Nutzern einspeichern und aufrufen. Die Chatverläufe sowie die Nutzer sollen über eine Datenbank gespeichert werden und jederzeit aufrufbar sein. Während der Kommunikation soll es dem Nutzer möglich sein formatierte Texte und Bilder senden und empfangen zu können.

Web-Sockets

Web-Sockets dienen dazu, dass die Kommunikation zwischen Client und Server aus beiden Richtungen erfolgen kann. Normalerweise fragt bei HTTP nur der Client den Server nach Dateien. Dieser sendet diese dann zwar an den Client, jedoch geschieht dies nur auf Anfrage. Mittels Web-Sockets kann nun der Server selbstständig Daten ungefragt an einen Client senden.

Dies ist für dieses Projekt vor allem sinnvoll, da die Kommunikation zwischen zwei Usern über einen HTTP-Server läuft. Somit sendet ein User Daten an den Server, welcher diese dann ohne Anfrage gleich an einen anderen User sendet. In diesem Fall wird die Node.JS-Technologie Socket.io benutzt.

Socket.io besteht allgemein aus zwei Teilen:

- Ein Server, welcher das Node.JS-Modul socket.io integriert.
- Eine Client-Library, welches die Technologie auf die Browser-Seite ladet (in die HTML-Datei) namens socket.io-client. Diese Library ist im Node-Modul enthalten.

Ablauf des Projekts

Zu Beginn soll das Wichtigste erstellt werden: Die Grundfunktion. In diesem Fall wäre das die Kommunikation zwischen zwei Usern über einen Server. Somit wird sichergestellt, dass das Grundprinzip hinter dem Projekt funktioniert. Dazu wird ein Server benötigt, auf dem eine Webseite läuft, welcher deshalb als erstes erstellt wird.

Als nächstes soll die Datenbankeinbindung funktionieren und zeitgleich das Prinzip der User-Accounts erstellt werden.

Zuletzt wird endgültig die Webseite erstellt, welche die zuvor erstellten Funktionen einbindet.

Beim Ablauf wird nicht unbedingt strikt nach Plan entwickelt. In jedem Bereich wird abwechselnd weiterentwickelt, damit zu jedem Zeitpunkt ein Vorzeigemodell existiert, welches mit der Zeit verbessert wird.

Rahmenbedingungen

- Das Projekt soll im Zeitraum von September bis Jänner/Februar entwickelt und beim Betreuer eingereicht werden.

Technologien

- Visual Studio Code als Entwicklungsumgebung
- NodeJS und Express.js für die Aufsetzung des Web-Servers
- Die Library socket.io für die Echtzeit-Kommunikation zwischen Clients und dem Web-Sever
- Die API JQuery, für die einfachere Verwendung von JavaScript
- Das Zeichenprogramm DIA für die Erstellung eines Datenbankschemas
- Die Datenbankumgebung MySQL für die Speicherung von Nutzerdaten und Chatverläufen
- GitHub für die Dokumentation des Projekts
- HTML, CSS und JavaScript für die Entwicklung der Webseite

Screenshots

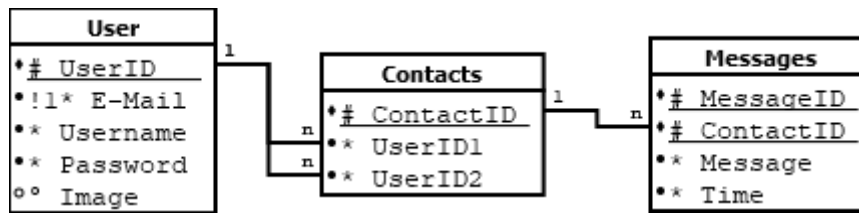


Abbildung 1: Datenbankschema

Meilensteine

Vorbereitung	
Einlesen in die unbekannten Technologien	28.09.2020
Verfassung des Pflichtenheftes	30.09.2020
Start des Projekts	30.09.2020
Entwicklung	
Entwicklung des Webseitenlayouts (HTML + CSS)	05.10.2020
Aufsetzen des NodeJS - Servers	19.10.2020
Kommunikation Client - Server	26.10.2020
Datenbankeinbindung	30.11.2020
Einbindung von Accounts	14.12.2020
Bonusziele	
Formatierung des Texts im Markdown-Format + Bilder	28.12.2020
Endphase	
Optimierung des Codes	25.01.2021
Abgabe	01.02.2021