

HS'BI

Hochschule
Bielefeld
University of
Applied Sciences
and Arts



Web Engineering WS 2024/25 Einführungsveranstaltung

Patrick Poppe, Sarah Flohr, Florian Fehring
09.10.2024

INHALT

1. Vorstellung der Teilnehmer
2. Termine
3. Bewertungskriterien
4. Projekt 1
5. Projekt 2

DOZENTEN

Gruppenleitung Energy Guide:

Patrick Poppe

patrick.poppe1@hsbi.de

Gruppenleitung Weserdatenbank:

Sarah Flohr

sarah.flohr@hsbi.de

Administration:

Florian Fehring

florian.fehring@hsbi.de

BLITZRUNDE

Je ein Satz zu:

- Name
- Studiengang
- Erfahrung und Vorkenntnisse in Webtechnologien
- Gruppenwunsch: Gruppe am Dienstag oder Mittwoch

TERMINE

Zeit	15.10	22.10	29.10	05.11	12.11	19.11	26.11	03.12	10.12	17.12	07.01	14.01	21.01	28.01
	A	A	SWK	F	MS1	F	MS2	F	MS3	F	AB	F	SWÜ	PB
08:45	*	*	WD G1	*	WD G1	*	WD G1	*	WD G1	*	*	*	WD G1	-
09:30	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	WD G2	-
10:15	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	WD G3	-
Pause 11:00 – 11:30														
11:30	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	EG G1	-
12:15	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	EG G2	-
13:00	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	EG G3	-

WD = Weserdatenbank
EG = Energy Guide
* = unbenotete Teamabnahme mit WD G1 mittwochs um 12:30 Uhr per Zoom
A = Anforderungsanalyse
SWK = Softwarekonzept
F = Feedbackstunde
MS1, MS2, MS3 = Meilensteine
AB = Agiler Baustein
SWÜ = Softwareübergabe
PB = Abgabe Projektbericht

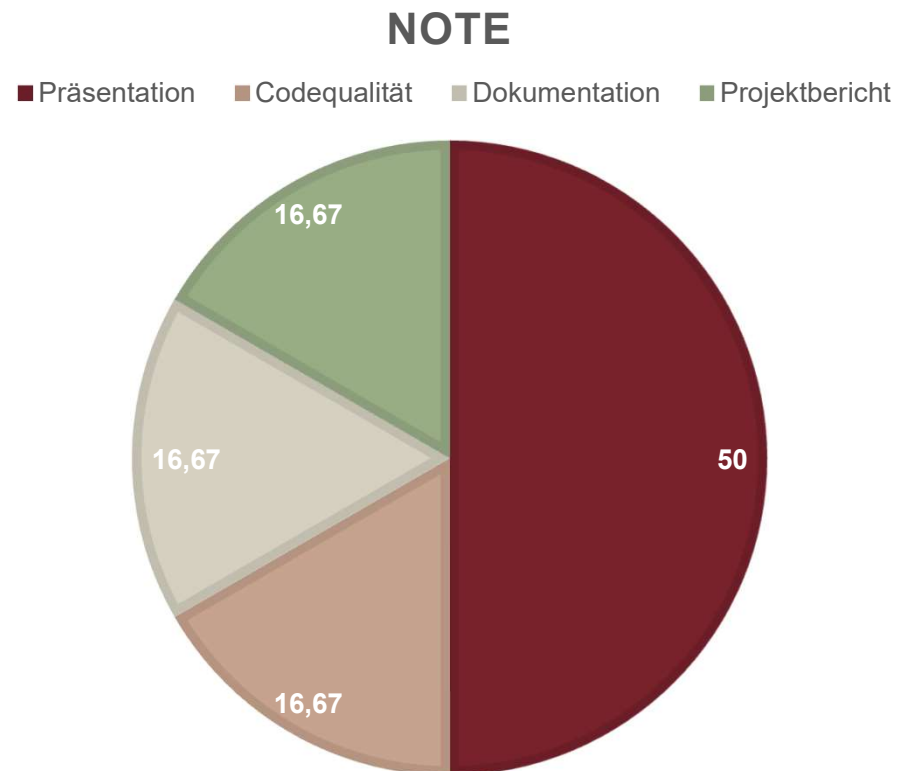
Rot: Seminarpräsentation mit Anwesenheitspflicht für alle WD Gruppen in D320 (benotet)

Lila: Seminarpräsentation mit Anwesenheitspflicht für alle EG Gruppen in D320 (benotet)

Schwarz: entsprechendes Team ist anwesend in D320 (unbenotet)

Blau: Seminarpräsentation mit Anwesenheitspflicht für alle in H10 (benotet)

BEWERTUNGSKRITERIEN ALLGEMEIN



BEWERTUNGSBAUSTEINE

SW-Konzept (UML)

- **Note 1** Präsentation

Meilenstein 1

- **Note 2** Präsentation + Codequalität + Dokumentation + (Software-Demo)

Meilenstein 2

- **Note 3** Präsentation + Codequalität + Dokumentation + (Software-Demo)

Meilenstein 3

- **Note 4** Präsentation + Codequalität + Dokumentation + (Software-Demo)

Softwareübergabe

- **Note 5** Präsentation + Codequalität + Dokumentation + Software-Demo

Projektbericht

- **Note 6**

BESTEHENSGRENZE

Wann bestehen Sie den Kurs nicht?

- eine **Seminarpräsentation** wird **nicht gehalten** oder **mit 5,0 bewertet**.
- **Präsentation** oder **Codequalität** oder **Dokumentation** ist **nicht vorhanden** oder wird **mit 5,0 bewertet**.
- der **Projektbericht** wird **nicht abgegeben** oder **mit 5,0 bewertet**.
- Sie fehlen mehr als **zweimal unentschuldigt**.
- Sie haben **kein Feature implementiert** oder Ihr **Feature** wird mit *nicht bestanden* bewertet.

Jede Note ist eine **Teamnote**, die die gesamte Gruppe erhält.

Wir behalten uns vor, Sie ggfs. auch individuell zu benoten, sollten wir der Meinung sein, dass die Teamnote nicht Ihre Leistung widerspiegelt.

Um das Modul zu bestehen, müssen Sie **mindesten ein Feature implementieren** und dürfen **nicht mehr als zweimal unentschuldigt fehlen**.

Sollten Sie an einem Tag mit Seminarpräsentation fehlen, wird Ihre Präsentation an einem gesonderten Termin nachgeholt.

BEWERTUNGSKRITERIEN SOFTWARE-KONZEPT

Inhaltliche Vollständigkeit und Themenbezug

- Kurze Projektbeschreibung
- Theoretische Grundlagen
 - Motivation und Zielsetzung
 - Relevanz des Themas und der Daten
- Anforderungsanalyse
 - UseCases / UserStories etc. inklusive Priorisierung
 - Mock-Ups
- Konzeptionelle Dokumentation
 - Ggfs. Klassendiagramme / Sequenzdiagramme
 - Ggfs. Komponentendiagramm
 - Tipp: Passen Sie die konzeptionelle Dokumentation an Ihre Anwendung an!

Organisation und Vorgehen

- Vorstellung der Gruppenmitglieder
- Zuteilung der Features
- Risikoanalyse
- Meilensteinplan

BEWERTUNGSKRITERIEN PRÄSENTATION

Inhaltliche Vollständigkeit und Themenbezug je nach aktuellen Projektstand

- Priorisierung der inhaltlichen Konzepte oder Meilensteine

Übersichtlichkeit und Leserlichkeit

- Passende Präsentationsform
- Schriftgröße
- Passende und hilfreiche Grafiken
- Sinnvolle und inhaltliche Strukturierung (roter Faden)
- Orientierungshilfen während des Vortrages

Vortragsweise

- Verständlichkeit (Lautstärke, Wortwahl, didaktische Methodik)
- Spannungsbogen, Aufmerksamkeitsgewinn beim Publikum
- Gleichmäßige und passende Aufteilung der Vortragsteile in der Gruppe

Einhalten der zeitlichen Vorgaben

- 30 Minuten (+/- 3 min) pro Team
- Gleichmäßige Redeanteile der Teammitglieder

BEWERTUNGSKRITERIEN SOFTWARE-ÜBERGABE

Konzept

- Sinnvolle Verwendung und Erklärung von Software-Entwicklungsmustern
- Sorgfalt bei der Auswahl von wieder-verwendeten Komponenten. (Langlebigkeit, Wartbarkeit etc.)
- Erweiterbarkeit (Nutzung von Konzepten wie Strategy-Pattern, Plugins,...)
- UseCases, Komponenten- und Klassendiagramme

Implementierung

- Sinnvolle Fehlerbehandlung
- Abstraktion vom Anwendungsfall wo möglich und Wiederverwendbarkeit vom Code
- Userinterface
- Intuitives Design und Einfachheit der Anwendung
- Anwendung ist auf Anwenderkreis abgestimmt

Software

- Erfüllung der geforderten Funktionalität
- Konfigurierbarkeit variabler Einstellungen
- Datenbankdesign
- Lauffähige, stabile Softwareversion

BEWERTUNGSKRITERIEN CODEQUALITÄT

- Effizienz und Performance des Codes
- Wiederverwendbarkeit des Codes (Modularität)
- Sinnvolle und Aussagekräftige Kommentare
- Konsistenz in Namensgebung und Formatierung
- Skalierbarkeit (Anwendung ist leicht zu erweitern, wenn die Nutzeranforderung wachsen)
- Redundanzen vermeiden
- Robustheit
- Sinnvolle Fehlerbehandlung
- Sprechende Bezeichnungen für Klassen, Funktionen, Variablen etc.
- Sauberkeit des Codes (keine unnötigen Importe, Formatierung, Codezeilen etc.)
- Sinnvolle Strukturierung

BEWERTUNGSKRITERIEN DOKUMENTATION

- Vollständigkeit
- Klarheit und Verständlichkeit
- Konsistente Struktur und standardisierte Formatierungen
- Aktualität (Dokumentation spiegelt den aktuellen Stand des Codes wider)
- Ggfs. Erklärung komplexer Algorithmen und Logiken
- Edge Cases (Dokumentation zu seltenen oder kritischen Anwendungsfällen sowie Fehlerbehandlung)
- Installations- und Konfigurationshinweise (wie wird die Anwendung installiert und ausgeführt, Anweisungen zur Einrichtung der Entwicklungsumgebung)
- Ggfs. Wichtige Designentscheidung (um später Modifikation und Wartung zu erleichtern)
- API-Dokumentation (API-Endpunkte, API-Nutzung, Beispiele für API-Calls)
- Fehlermeldungen

BEWERTUNGSKRITERIEN PROJEKTBERICHT

Inhaltliche Vollständigkeit und Themenbezug

- Inhaltsverzeichnis, Abbildungsverzeichnis, Tabellenverzeichnis, Abkürzungsverzeichnis etc.
- Motivation und Themenbeschreibung
- Übersicht über ihre Feature sowie AB und welches Gruppenmitglied welches Pflicht-Feature bearbeitet hat
- Theoretische Grundlagen
- Konzeptionelle Arbeiten und eine aussagekräftige Auswahl an UseCases und SW-Architektur, Klassendiagramme, Sequenzdiagramm (alle textuell/inhaltlich beschrieben)
- Aussagekräftige Code-Ausschnitte, um Besonderheiten hervorzuheben
- Ausblick, Fazit
- Quellenverzeichnis
- Anlagen (Meilensteinplan, SW-Konzept mit kompletten UseCases, Sequenzdiagrammen, Klassendiagrammen etc.)

Übersicht und Leserlichkeit des Berichts

- Schriftgröße 10 p im Fließtext, Überschriften dürfen leicht größer sein
- Passende und hilfreiche Grafiken
- Inhaltliche Kohärenz
- Seitennummern
- Abbildungen, Tabellen sind nummeriert, Bild- bzw. Abbildungsunterschrift

Schreibweise

- Verständliche Ausdrucksweise
- Grammatik und Orthographie

Einhalten der Vorgaben der Seitenzahl (ohne Anhang)

- 40-50 Seiten

EIN ERFOLGREICHES PROJEKT

Minimalanforderungen:

- Präsentation
- Codequalität
- Dokumentation
- Projektbericht
- Jeder ein Feature
- Agiler Baustein: Mind. ein weiteres Feature

Erweitern und verbessern Sie Ihre Features!

- Chance, sich von den anderen Gruppen abzuheben und besondere Innovation sowie Kreativität zu zeigen.

ALLGEMEINES

- Die Web-Anwendung wird mit React programmiert
- Sie nutzen TypeScript
- Gruppengröße: 4-5 Personen

SEMINARPRÄSENTATION

Am Vortag der Seminarpräsentation **bis 23:00 Uhr:**

- Sie laden Ihren Vortrag als **PDF** im **ILIAS** hoch
- Sie setzen einen **annotierten Git-Tag**. Der Name des Tags enthält Ihren Gruppennamen sowie den aktuellen Meilenstein. Also bspw. „G1MS1“, „G2MS1“, „G3MS2“ etc.
- Die Dokumentation machen Sie in der Read-Me als Markdown

Git-Tag: <https://git-scm.com/book/de/v2/Git-Grundlagen-Taggen>

Markdown: <https://commonmark.org/>

PROJEKT 1 – ENERGY GUIDE – PATRICK POPPE

Anzeige von Energiewerten

Folgende Parameter können abgerufen werden

- Strom – Leistung
- Strom – Verbrauch
- Pneumatik – Leistung
- Pneumatik – Verbrauch
- Absaugung – Leistung
- Absaugung – Verbrauch
- Eventdaten
- Zeitstempel

ANFORDERUNGEN ALLGEMEIN

Die Erfassung der Energiedaten ist zur Überwachung der Maschine und des Energieverbrauches wichtig

Ziel: Web-Applikation

- Einlesen und Exportieren der Daten
- Grafische Darstellung der Messwerte sowie Events
- Rollenbasiertes Zugriffsmanagement (Admin Panel)

Datenbank:

- Eigene Wahl

FEATURES (PFLICHT)

Graphische Datenauswertung

Als Nutzer möchte ich, dass die Messwerte in einer Verlaufsgrafik dargestellt werden. Die Events sollen ebenfalls in der Grafik dargestellt werden.

Einlesen und Exportieren von Daten

Als Nutzer möchte ich, dass csv-Dateien eingelesen werden können. Die Daten sollen ebenfalls als csv-Datei wieder exportiert werden können.

Rollenbasiertes Zugriffmanagement

Als Admin möchte ich, dass der Zugriff beschränkt wird. Nur bestimmten Nutzern soll es möglich sein, Maschinen hinzuzufügen oder zu löschen. Bestimmte User sollen nur für bestimmte Maschinen Daten hochladen können.

Filter- und Suchfunktion

Als Nutzer möchte ich nach Programmen suchen und die Messwerte nach verschiedenen Kriterien (z.B. Programmname, Energieverbrauch) filtern können.

Mehrere Maschinen

Als User möchte ich die Daten von verschiedenen Maschinen speichern und abrufen können.

FEATURES (WEITERE IDEEN)

Dashboard

Anzeige von Kenndaten über mehrere Maschinen hinweg (z.B. Prozentzeit der Maschine in Programm) +
Anzeige von Kenndaten für eine Einzelmachine

Datenexport

Als Nutzer möchte ich, dass die Daten auch als JSON, XML oder PDF ausgegeben werden, um
verschieden Anwendungsfälle zu berücksichtigen.

Mobile Optimierung

Als Nutzer möchte ich eine optimierte Darstellung auf meinen mobilen Geräten. (Responsive Design)

Benutzeraktivitäten-Log

Als Admin möchte ich eine automatische Protokollierung der Aktivitäten von Nutzern ausgegeben
bekommen (z.B. wenn Daten eingegeben oder bearbeiten wurden)

Vergleich

Als User möchte ich verschiedene Programme grafisch nebeneinander / übereinander vergleichen können

FEATURES (WEITERE IDEEN)

Personalisierte Dashboards

Als Nutzer möchte ich mein eigenes Dashboard erstellen und individuell konfigurieren (z.B. verschiedene Diagrammtypen oder spezifische Maschinen)

Berichtexport

Als Nutzer möchte ich, dass wichtige Kennzahlen (z.B. Anteil der Maschine in Nebenzeiten) in einem übersichtlichen PDF-Bericht ausgegeben werden

Trendanalyse

Als Nutzer möchte ich Programme mit dem gleichen Programmnamen nebeneinander darstellen und eine Trendanalyse der verbrauchten Energie durchführen

Maschinenvergleich

Als User möchte ich verschiedene Maschinen miteinander vergleichen (z.B. Programm von Maschine A mit Programm von Maschine B)

Vergleich

Als User möchte ich verschiedene Programme grafisch nebeneinander / übereinander vergleichen können

PROJEKT 2 – WESERDATENBANK – SARAH FLOHR

Die Weserdatenbank Qualitätsüberwachung der Weser

Folgende Parameter können abgerufen werden:

- Chlorid und weitere Ionen
- Hydrologische Daten
- Kontinuierlich gemessene Kenngrößen
- Metalle
- Nährstoffe
- Organische Schadstoffe
- Summenparameter



ANFORDERUNGEN ALLGEMEIN

Die Erfassung der Gewässergüte ist wichtig, um die Wasserqualität zu überwachen und Ökosysteme zu schützen. Sie hilft, Schadstoffe frühzeitig zu erkennen und Trinkwasserressourcen zu sichern.

Ziel: Web-Applikation

- Einlesen und Exportieren der Daten
- Kartendarstellung von Messwerten sowie Messstationen
- Graphische Auswertung von Messwerten
- Rollenbasiertes Zugriffmanagement (Admin Panel)

Datenbank:

- PostgreSQL

FEATURES (PFLICHT)

Kartendarstellung

Als Nutzer möchte ich, dass einzelne Messwerte sowie Messstationen auf einer Karte angezeigt werden.

Graphische Datenauswertung

Als Nutzer möchte ich, dass der Chlorid-Gehalt des Wassers als Diagramm im zeitlichen Verlauf dargestellt wird.

Einlesen und Exportieren von Daten

Als Nutzer möchte ich, dass csv-Dateien eingelesen werden können. Die Daten sollen ebenfalls als csv-Datei wieder exportiert werden können.

Rollenbasiertes Zugriffmanagement

Als Admin möchte ich, dass der Zugriff beschränkt wird. Nur bestimmten Nutzern soll es möglich sein, Messstationen hinzuzufügen oder zu löschen.

Filter- und Suchfunktion

Als Nutzer möchte ich nach Station suchen und die Messwerte nach verschiedenen Kriterien (z.B. Zeitraum, Schadstofftyp) filtern können.

FEATURES (WEITERE IDEEN)

Benachrichtigungssystem

Als Nutzer möchte ich eine Warnmeldung erhalten, wenn ein Messwert kritisch überschritten wird.

Historische Datenanalyse

Als Nutzer möchte ich die Wasserqualität über verschiedene Zeiträume hinweg mit einer Trendanalyse dargestellt bekommen.

Mobile Optimierung

Als Nutzer möchte ich eine optimierte Darstellung auf meinen mobilen Geräten. (Responsive Design)

Kommentarfunktion

Als Nutzer möchte ich Notizen und Kommentare zu bestimmten Messwerten oder Stationen hinzufügen können, um beispielsweise auffällige Beobachtungen zu notieren.

FEATURES (WEITERE IDEEN)

Dashboards

Als Nutzer möchte ich ein Überblicks-Dashboard mit interaktiven Diagrammen, z.B. für Durchschnittswerte oder regionale Unterschiede.

Datenexport

Als Nutzer möchte ich, dass die Daten auch als JSON, XML oder PDF ausgegeben werden, um verschieden Anwendungsfälle zu berücksichtigen.

Wetterdaten-Integration

Als Nutzer möchte ich, dass die lokalen Messdaten mit Wetterdaten (z.B. Open-Meteo) verknüpft werden, um Zusammenhänge zwischen Wetter und Wasserqualität aufzuzeigen.

Benutzeraktivitäten-Log

Als Admin möchte ich eine automatische Protokollierung der Aktivitäten von Nutzern ausgegeben bekommen (z.B. wenn Daten eingegeben oder bearbeiten wurden)

FEATURES (WEITERE IDEEN)

Geografische Clusteranalyse

Als Nutzer möchte ich Hotspots oder Muster in den Messwerten über Regionen hinweg identifizieren können.

Statistische Auswertung

Als Nutzer möchte ich eine statistische Auswertung der Werte (Standardabweichungen, Medianen, Quartilen etc.) zur tieferen Analyse der Wasserqualität.

Personalisierte Dashboards

Als Nutzer möchte ich mein eigenes Dashboard erstellen und individuell konfigurieren (z.B. verschiedene Diagrammtypen oder spezifische Stationen)

NÜTZLICHE LINKS

<https://www.fgg-weser.de/weser-datenbank>

<https://datenbank.fgg-weser.de/weserdatenbank/#/data-groups>

<https://www.postgresql.org/>

<https://www.chartjs.org/>

<https://mui.com/>

<https://sap.github.io/ui5-webcomponents/>

<https://react-leaflet.js.org/>