

Webentwicklung (WE) - FWPM

Informationen Studienarbeit und Prüfverfahren

Im Folgenden wird der erwartete Inhalt sowie die Bewertungskriterien für die Studienarbeit, sowie der Verlauf der Präsentation beschrieben.

Prüfverfahren

Das Prüfverfahren basiert auf einer Studienarbeit, in der die Inhalte der Vorlesung durch die Studenten praktisch verarbeitet werden müssen.

Die tatsächliche Note bildet sich aus dem abgegebenen Quellcode und einer Präsentation der Inhalte der Studienarbeit.

Quellcode und Code Freeze

Ein großer Bestandteil der Gesamtnote ist der im Repository liegende Code und die damit verbundenen Metadaten des Projekts. **Es werden alle im von der TH bereitgestellten git Repository befindlichen Dateien zur Notenbildung herangezogen.** Daher ist empfohlen auch eventuelle Architekturschaubilder, Webserver Konfigurationen, Projektmanagementunterlagen, etc. im Repository verfügbar zu machen.

Um die Funktionalität der Anwendung nachvollziehbar zu machen, **muss zwangsläufig ein Dump (Export) der Datenbank Teil des Repositories sein.** Dieser muss die Struktur der Datenbank enthalten und sollte auch bereits vorhandene Daten beinhalten.

Ein Datenbank Dump kann mit Tools wie z.B. *mysqldump* oder *MySQL Workbench* erzeugt werden und ist eine reine Textdatei die MySQL Befehle enthält.

Der späteste Abgabetermin ist Mitternacht am Vortag des kommunizierten Prüfungsdatums. Spätere commits auf das Repository z.B. kurz vor dem Präsentationstermin werden nicht mehr berücksichtigt!

Falls Drittanbieterbibliotheken benutzt werden, ist für die Benotung übrigens nicht erheblich ob sie die nutzen, die in der Vorlesung vorgestellt wurden oder sie z.B. ein anderes ORM wählen.

Präsentation der Studienarbeit

Jede(r) Prüfungsteilnehmer(in) hat in einer kurzen, 15 minütigen Präsentation die Möglichkeit auf ihre/seine Studienarbeit einzugehen und diese zu erklären.

Dabei ist ein kurzer Überblick über das Ziel der Anwendung und ihr Funktionsumfang sinnvoll.

Weiter sollte auf eventuelle Besonderheiten (sowohl Funktionalität als auch Code) eingegangen werden um Punkte aufzuzeigen an denen sich die Anwendung positiv hervorhebt.

Die Präsentation trägt zu einem Teil zur Gesamtnote bei (siehe **Bewertungsschema**) wobei hauptsächlich Wert auf den Inhalt und nicht auf den Präsentationsstil gelegt wird.

Die Präsentation findet am kommunizierten Prüfungsdatum statt und wird eventuell in einer separaten Abstimmung in Zeitslots unterteilt.

Erwartete Inhalte der Studienarbeit

Für die Studienarbeit soll jede(r) Prüfungsteilnehmer(in) selbstständig ein Webprojekt im in der Vorlesung behandelten Technologie-Stack HTML, CSS, PHP (+ JS falls gewollt) umsetzen.

Bei dem Projekt geht es weniger um den tatsächlichen Nutzen oder die Menge der Features sondern mehr um die fachliche Finesse mit der diese umgesetzt sind.

Eine saubere Architektur und Nutzung moderne Programmieransätze (z.B. Design Patterns) sind wichtiger als eine schön gestaltete Oberfläche.

Das Einbinden von Fremdbibliotheken ist explizit erlaubt, allerdings sollte es bei spezialisierten Bibliotheken wie z.B. Hilfsfunktionen zur Validierung von Nutzereingaben bleiben. Keine Frameworks oder Komplettlösungen für gesamte Teilbereiche oder Schichten einer Anwendung wie z.B. eine komplette Benutzerverwaltung. Im Zweifelsfall bitte die Nutzung mit dem Dozenten abstimmen.

Der Funktionsumfang kann einfach gehalten werden, allerdings sollten sich alle Themen aus der Vorlesung wiederfinden.

Dabei wird auf bestimmte Kernthemen besonders Wert gelegt.

Das sind unter anderem, aber nicht exklusiv Themen wie:

- Eine Persistenzschicht mit Nutzung einer Datenbank
- Das Ermöglichen und Verarbeiten von Nutzereingaben
- Eine Architektur die auf einem Schichtenmodell (z.B. MVC) basiert
- Die Möglichkeit zum Login und eine einfache Rechteverwaltung
- Ein einfaches Oberflächendesign auf Basis von HTML und Stylesheets

Konkretes Beispiel ist ein Blog. Angemeldete Nutzer können über ein Formular Beiträge erstellen.

Diese lassen sich einzeln oder auch als Liste anzeigen und durch den Originalautor auch wieder löschen und editieren.

Das Ganze ist einigermaßen optisch ansprechend.

Das wäre eine minimale Anwendung im Sinne der Vorlesung.

Weitere Themen die in der Vorlesung behandelt wurden sind nicht zwangsweise notwendig, verbessern aber die allgemeine Wahrnehmung des Projekts und damit die Note.

Das sind die Themen:

- Erweiterung um Webservices (Bereitstellung oder Anbindung)
- Nutzung von Javascript z.B. zur asynchronen Kommunikation mit dem Client
- Aktives Usability Engineering und Oberflächenoptimierungen
- Qualitätssichernde Maßnahmen z.B. Nutzung von umfangreichen CI Pipelines

Wer Weiteres plant (was gern gesehen wäre) kann sich beliebig spielen.

Eine gute Suchfunktion, Inhalte optional über einen Webservice, ein Pluginsystem für neue Funktionalität, Adminfunktionen, ein erweitertes Rechtesystem, vom Nutzer frei wählbare Oberflächentemplates, ...

Der Fantasie sind keine Grenzen gesetzt.

Bewertungsschema

Im Folgenden sind Kriterien samt Beispielen aufgelistet, die zur Bildung der Note herangezogen werden. In Klammern dahinter eine grobe Gewichtung.

Die Gewichtung ist mit Vorsicht zu genießen, da z.B. eine Anwendung die aufgrund von Fehlern in der Implementierung einfach nicht funktioniert in mehreren Kriterien zu Abstrichen führen kann.

Kriterien

Zweckmäßigkeit (15%)

Ist die Implementierung effizient gewählt für den Einsatzzweck? Sind die gemachten Abstriche bei der Effizienz durch eine bessere Architektur zu rechtfertigen?

Funktionalität (15%)

Funktionieren die Angedachten Funktionen der Anwendung? Hat die Anwendung eine gute Usability? Lässt sie sich durchgängig nutzen?

Umfang (25%)

Bietet die Anwendung einen plausiblen Umfang für eine Studienarbeit? Sind alle Kernthemen gut integriert? Wurden zusätzliche Themen, z.B. ein Webservice implementiert?

Optik (10%)

Ist die Anwendung optisch ansprechend? Verhält sich die Oberfläche fehlerresistent auf mobilen Endgeräten? Existiert ein durchgängiges Design?

Sicherheit (10%)

Sind Maßnahmen zur Sicherung der Anwendung getroffen worden? Wird sinnvoll mit Nutzereingaben und der Ausgabe dynamischer Inhalte umgegangen?

Architekturelle Qualität (25%)

Basiert lose auf den SOLID Prinzipien und dem ISO/IEC 9126 bzw. 25000 Standard und setzt sich aus den Kriterien Wartbarkeit, Benutzbarkeit, Zuverlässigkeit zusammen.

Wäre meine Anwendung durch eine dritte Partei leicht erweiterbar? Habe ich eine klare Trennung in Komponenten? Existiert eine Schichtenarchitektur?

Verteilung

Die obigen Kriterien gelten sowohl für Analyse des Quellcodes als auch die Präsentation. Sie fließen ungefähr in einem Verhältnis von 80/20 in die Note ein.