

Allgemeine Hinweise

Nachdem wir in der ersten Hälfte des Semesters überwiegend theoretische Inhalte behandelt haben, soll dieses Wissen nun in einem Programmierprojekt praktisch angewendet werden. Dies ist ein wichtiger Schritt, um die am Anfang der Vorlesung mitgeteilten Kompetenzziele – insbesondere zum Entwurf und zur Realisierung eigener Webanwendungen – zu erreichen. Denn bisher haben wir zwar einzelne Teilaspekte in kleineren Übungen vertieft, wir haben uns aber noch nicht an die Umsetzung einer komplett neuen App gewagt. Doch erst durch die praktische Anwendung kannst du die notwendige Erfahrung sammeln, damit sich dein neues Wissen festigt.

In der beruflichen Praxis wirst du jedoch selten mal die Gelegenheit haben, einfach drauf los zu programmieren und am Ende des Tages (oder der Woche oder des Monats) zu schauen, ob was brauchbares dabei herausgekommen ist. Auch wirst du in der Praxis nicht immer alleine sondern sehr oft im Team mit anderen Entwicklern an einer Codebasis arbeiten. Um dich also bestmöglich auf den späteren Alltag vorzubereiten, müssen auch diese Aspekte in unserer Vorlesung berücksichtigt werden. Aus diesem Grund besteht eure Aufgabenstellung nicht einfach nur darin, eine Anwendung zu entwickeln, deren Quellcode am Semesterende benotet wird. Stattdessen sollt ihr sämtliche Schritte, die von der ersten Idee bis zum fertigen Produkt anfallen, im kleinen Rahmen einmal durchlaufen.

Bevor ihr also anfangt zu programmieren, müsst ihr erst eine Idee entwickeln und diese in einem Fachkonzept dokumentieren. Dieses bildet dann später die Grundlage für die Umsetzung eurer Idee. Es ist sozusagen der Vertrag, den ihr mit euch selbst und auch mit uns Dozenten schließt, um darin festzuhalten, was für eine Anwendung ihr entwickeln wollt und was sie alles leisten oder auch nicht leisten muss. Neben einer fachlichen Beschreibung der Anwendungsfunktionen enthält das Konzept deshalb auch andere Inhalte, die für die Umsetzung von Bedeutung sind:

- An welche Anwendergruppen richtet sich die neue Webanwendung?
- Welche Aufgaben und Probleme wollen die Anwender mit der Anwendung lösen?
- Wie soll die Anwendung später aussehen? (Skizzen mit UI-Mockups)
- Was muss die Anwendung im Einzelnen tun, um alle Anforderungen zu erfüllen?
- Soll die Anwendung im Browser, auf dem Server oder auf beiden laufen?
- Welche Technologien sollen bei der Implementierung eingesetzt werden?

Auf dieser Basis könnt ihr dann die Anwendung realisieren, was natürlich den Löwenteil der Arbeit ausmachen wird. Sobald die Programmierung jedoch abgeschlossen wurde, müsst ihr die App auch einmal durchtesten, um ihre Funktionstüchtigkeit zu beweisen. Das Vorgehen und die Ergebnisse dieses Tests müsst ihr somit ebenfalls dokumentieren, da nur so ein wirklicher Nachweis über die Qualität vorliegt und auch nur so dokumentiert ist, welche Fehler möglicherweise noch ausgebessert werden müssen.

Fachkonzept

Auf Moodle findet ihr ein Beispiel und eine Vorlage für das Fachkonzept. Nutzt diese, um euer eigenes Fachkonzept zu schreiben. Euer Konzept sollte zu jedem Punkt im Inhaltsverzeichnis entsprechende Informationen beinhalten und dabei die oben genannten Leitfragen vollständig beantworten. Umfang und fachliche Tiefe sollen in etwa dem Beispielkonzept entsprechen. Das Konzept muss in der ersten Projektwoche abgeschlossen und eingereicht werden.

Implementierung

Auf Basis des Fachkonzepts könnt ihr euch dann an die Realisierung der neuen Anwendung machen. Dabei solltet ihr jedoch folgende Spielregeln beachten:

- Alle Arten von Anwendungen sind erlaubt. Es gibt keine Einschränkungen, außer dass jede Gruppe sich eine eigene Anwendung ausdenken und umsetzen muss.
- Ebenso sind alle Technologien, Werkzeuge, Bibliotheken und Frameworks erlaubt.
- Umfang und Komplexität der Anwendung müssen in etwa der Beispielanwendung auf Moodle entsprechen.
- Ihr könnt so viel Code, wie ihr wollt, aus der Vorlage und den anderen Quellcodes auf Moodle übernehmen. Jedoch müssen mindestens 75% des Quellcodes von euch selbst erdacht und geschrieben werden, um die Prüfung zu bestehen.
- Das Grundgerüst der Anwendung kann vom gesamten Team gemeinsam erstellt werden.
- Jedes Gruppenmitglied muss darüber hinaus aber einen abgegrenzten Teil der Anwendung weitgehend eigenständig umsetzen. Dieser Bereich muss alle Themengebiete unserer Vorlesung berühren, so dass am Ende jeder Einzelne sowohl HTML- und CSS- als auch JavaScript-/Java-Code beigesteuert hat.
- Wenn eure Anwendung einen umfangreichen JavaScript-Anteil im Browser hat, müsst ihr keinen zusätzlichen Serverteil entwickeln und umgekehrt. Eines von beiden muss jedoch vorhanden sein, um die Anwendungslogik zur Verfügung zu stellen.
- Jedes Mitglied muss ungefähr die gleiche Menge Code zur Anwendung beitragen.
- Damit wir das nachvollziehen können, muss euer Code mit git verwaltet und auf GitHub öffentlich einsehbar sein. Der Code muss dabei so lange online bleiben, bis ihr eine Notenmitteilung erhaltet.
- Die Commit-Nachrichten sollten der zur Verfügung gestellten Vorlage entsprechen, damit wir eure Eigenleistung sinnvoll bewerten können.
- Der Code muss so dokumentiert sein, dass wir ihn bei der Bewertung auf unseren eigenen Rechnern laufen lassen können.

Testdokumentation

Auf Moodle findet ihr ein Beispiel und eine Vorlage für die Testdokumentation. Nutzt diese, um eure Anwendung durchzutesten und das Ergebnis der Tests sauber zu dokumentieren. Zu jedem Test gehört dabei eine Beschreibung, was getestet wird, wie ihr hierfür vorgegangen seid und welches Ergebnis ihr dabei erwartet habt. Dies wird dann dem tatsächlichen Ergebnis gegenübergestellt und mit Screenshots dokumentiert. Umfang und fachliche Tiefe sollen in etwa der Beispieldokumentation auf Moodle entsprechen. Ihr solltet jedoch ganz unabhängig davon alle Bereiche eurer Anwendung so gut wie möglich testen.

Leistungsnachweis

Auf Moodle findet ihr eine Vorlage zur Erbringung eurer Leistungsnachweise. Es handelt sich dabei um ein Dokument, in dem ihr zu jedem Artefakt (Konzept, Testdokumentation, Quellcodedateien) die Namen der beteiligten Projektmitglieder dokumentieren müsst. Dies benötigen wir, um euren individuellen Projektbeitrag bewerten zu können. Das Dokument muss daher wahrheitsgemäß und gewissenhaft ausgefüllt und zusammen mit den anderen Projektunterlagen eingereicht werden. Zur Kontrolle werden wir bei der Benotung Stichproben vornehmen und die Commit-Historie einzelner Quellcodedateien mit dem Dokument abgleichen.

Sollte sich bei der Bewertung herausstellen, dass die Aufwände im Team zu ungleich verteilt waren, wird die Endnote der Vorlesung aller Projektmitglieder wie folgt angepasst:

- **Gruppenmitglied mit dem größten Anteil:** Bekommt die volle Note
- **Gruppenmitglied mit dem zweitgrößten Anteil:**¹ Bekommt eine halbe Note schlechter
- **Alle anderen Gruppenmitglieder:** Bekommen eine ganze Note schlechter

Sollte sich darüber hinaus herausstellen, dass sich einzelne Mitglieder auf bestimmte Vorlesungsinhalte spezialisiert haben, und entsprechend der anderen Kapitel gar nichts oder nur sehr wenig beigetragen haben (zum Beispiel nur HTML oder nur CSS)², wird die bestenfalls erreichbare Endnote wie folgt gedeckelt:

- **Wenn alle drei Bereiche abgedeckt werden:** 1,0 ist durchaus möglich
- **Wenn nur zwei Bereiche abgedeckt werden:** Nicht besser als 2,5
- **Wenn nur ein Bereich abgedeckt wird:** Nicht besser als 3,5

Zwar ist es in der Praxis durchaus üblich, sich auf bestimmte Gebiete zu spezialisieren und die Aufgaben dementsprechend im Team zu verteilen. Im Sinne der Prüfungsgerechtigkeit müssen wir jedoch darauf achten, dass die Prüfung für alle Studierenden den gleichen Umfang, die gleichen Inhalte und das gleiche Niveau besitzt. In der Vorlesung müssen wir die Aufgaben daher leider anders verteilen, als es möglicherweise später im Berufsleben der Fall wäre.

¹ Nur, wenn die Gruppe aus mindestens drei Mitgliedern besteht.

² Jedes Gruppenmitglied muss sowohl HTML-Code als auch CSS-Code aus auch JavaScript-/Java-Code beisteuern. Der Umfang muss jeweils dem Umfang entsprechen, der auch von den anderen Mitgliedern erbracht wird.