

Bericht zum Fachpraktikum

Zeitraum: 01.04. – 14.08.2020

Pauline Röhr

Matrikelnummer: 569735

Internationale Medieninformatik, FB4

pauline.roehr@yahoo.de

+49172 880 1756

Ausbildungsstelle:

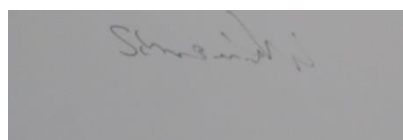
Babel GmbH

Erkrather Straße 224a, 40233 Düsseldorf

Ansprechpartner: Stefan Schweminski

s.schweminski@babel.com

07.09.2020



Inhalt

Inhalt.....	1
Zeit und Ausbildungsstelle	2
Dauer, Beginn und Ende des Fachpraktikums.....	2
Ausbildungsstelle und Einsatzbereiche	2
Eigene Aufgaben und Tätigkeiten	2
Art und Umfang der einzelnen Aufgaben.....	2
Agile Arbeiten.....	2
Content und seine Struktur.....	3
Zusammenarbeit mit dem Frontend.....	4
Anpassungen an der Content-Management-App.....	4
Implementierung von Logik	4
Fehlerbehebung	5
Sonarqube	5
Sonstiges	6
Eigene Arbeitsergebnisse	6
Anpassungen am CKEditor	6
Detailnavigation	8
Bilduploads.....	8
Art und Umfang der Unterstützung durch die Ausbildungsstelle	8
Bezüge zwischen Studium und Fachpraktikum.....	9
Nützlichkeit der bisherigen Lehrveranstaltungen	9
Nützlichkeit des Fachpraktikums für das weitere Studium.....	10
Zusammenfassung und Ausblick	10
Aufgetretene Probleme	10
Verbesserungsmöglichkeiten	11
Berufliche Perspektiven.....	11

Zeit und Ausbildungsstelle

Dauer, Beginn und Ende des Fachpraktikums

Vom 01.04.-14.08.2020, also für ca. 19 Wochen, habe ich mein Praktikum bei der Firma Babel GmbH abgeleistet. In diesem Zeitraum waren auch insgesamt 8 Urlaubstage enthalten.

Ausbildungsstelle und Einsatzbereiche

Babel GmbH agiert als Agentur und Berater für Online-Kommunikation, Digital Marketing und E-Commerce, seit 1991. Mit mehr als 100 Mitarbeitern in verschiedensten Bereichen – Administratoren, Software-Entwickler, Projektleiter, Grafik-Designer, Kaufmännische Angestellte – bietet Babel Ihren Kunden auf sie zugeschnittene Softwarelösungen, sowie ggfs. ein entsprechendes Hosting an. Diese Softwarelösungen basieren zu einem großen Teil auf Partnerschaften mit SAP-Hybris, JAHIA und CoreMedia. Mein Einsatzbereich im Praktikum bezog sich dabei hauptsächlich auf die Backend-Entwicklung am CoreMedia Content Management System für unseren Kunden „Wiener Tourismusverband“. Da ich schon seit Mai 2019 bei Babel als Werkstudentin tätig bin, war es mir möglich in der ersten Woche meines Praktikums ohne zusätzliche Einarbeitung schon am ersten SCRUM-Sprint dieses Projekts aktiv mitzuwirken.

Eigene Aufgaben und Tätigkeiten

Art und Umfang der einzelnen Aufgaben

Zum 31.03.2020 begann bei Babel die agile Entwicklungsarbeit und somit der erste Sprint im Projekt für den Kunden Wiener Tourismusverband (fortan kurz WTV). Zielsetzung ist es, die bestehenden Webseiten des Kunden inkl. des vorhandenen Contents in das Content-Management-System CoreMedia zu übertragen. Jegliche Entwicklungs-Aufgaben in meinem Praktikum stammen aus den User Stories dieses SCRUM-Projekts, deren Anforderungen größtenteils vom WTV gestellt wurden. Die Aufgaben habe ich mir häufig selbst ausgesucht oder gemeinsam im Entwicklungsteam entschieden, welche Aufgaben ich übernehmen sollte.

Agile Arbeiten

Für die Arbeiten in diesem Projekt wurde sich für ein SCRUM-Prozess entschieden, welcher erfordert, dass das Entwicklungsteam gemeinsam Sprints plant und Umsetzungen konzeptioniert. Im „Daily“-Meeting zu Beginn des Tages hat jede Person aus dem Entwicklungsteam kurz zusammengefasst, woran zuletzt gearbeitet wurde, welche Pläne für den kommenden Tag bestehen und ob es Probleme gibt, welche ggfs. von unserem SCRUM-Master behoben werden sollten.

Außerdem haben wir wöchentlich gemeinsam neue User Stories nach SCRUM-Standard geschätzt, damit wir diese im nächsten Sprint bearbeiten konnten. In diesen „Refinements“ haben wir gemeinsam die Umsetzung der Anforderungen konzeptioniert und als

Unteraufgaben in unserem Ticketsystem festgehalten. Die Dauer und den Aufbau dieses Meetings haben wir häufiger angepasst, da wir oftmals zu wenig Stories geschätzt hatten und hierfür viel Konzentration und Zeit erforderlich waren. Um uns dies etwas zu vereinfachen haben wir uns nach einigen Sprints darauf geeinigt, dass die Entwickler User Stories vorbereiten, sodass sie die Anforderungen zusammenfassen und schon Umsetzungsvorschläge mitbringen können.

Zum Sprintwechsel alle drei Wochen gab es außerdem drei weitere Meetings - Sprint Review, Sprint Planning und die Retrospektive – mit welchen wir meist einen ganzen Tag verbracht haben. Im Sprint Review hat das Entwicklungsteam dem Kunden fertiggestellte User Stories vorgestellt und anstehende Fragen zu den jeweiligen Umsetzungen beantwortet. Meist hat jeder Entwickler die Stories vorgestellt, die er selbst auch umgesetzt hat, weshalb ich auch bei fast jedem Sprint Review etwas vorstellen konnte. Im Sprint Planning ging es darum den neuen Sprint zu planen und sich ein Sprintziel zu setzen. Häufig haben wir hier erneut Stories geschätzt, um genügend User Stories einplanen zu können. Auch haben wir hier einen kleinen Rückblick auf den letzten Sprint geworfen, um zu überprüfen, welche Aufgaben wir eventuell noch nicht erledigt hatten und wo erneut mit dem Kunden kommuniziert werden musste. In der Retrospektive haben wir den beendeten Sprint ausgewertet. Alle Teammitglieder durften hier offen ihre Kritik äußern und anschließend über mögliche Maßnahmen diskutieren. Hieraus stammten starke Verbesserungen in unseren Kommunikations- und Arbeitsprozessen, wie z.B. die Anpassungen an unseren Refinement-Meetings.

Content und seine Struktur

CoreMedia ermöglicht einen Webauftritt bzw. den enthaltenen Content über ein Ordner- und Verlinkungssystem im sogenannten „Studio“ darzustellen und zu pflegen. Ich habe im Projekt die Aufgabe übernommen, die anfängliche Ordnerstruktur für alle erforderlichen Mandanten/Sprachvarianten aufzusetzen und Beispielcontent vorzubereiten. Dies war notwendig, um jegliches Testing zu ermöglichen und hat außerdem dem Kunden als Dokumentation gedient. Grundsätzlich war dies keine Entwicklungsarbeit, jedoch war es nötig präzise zu Planen und gegebene Standards von Coremedia einzuhalten, sodass Sprachvarianten auch als solche erkannt wurden.

Außerdem sollte in einigen User Stories die Redaktionelle Pflege des Rahmenlayouts ermöglicht werden. Dies beinhaltet beispielsweise die Anpassbarkeit von Verlinkungen, Linktexten oder Icons in der Footer Navigation, Footer Service Navigation oder der Service Navigation (siehe Header und Footer Bereich von <https://www.wien.info/de>), welche auf allen Seiten eines Mandanten sichtbar sind. Dafür habe ich Daten in der Content-Management-App (Studio) vorbereitet und anschließend den Aufbau dieser für das Frontend und den Kunden in unserem Projekt-Wiki dokumentiert. Meist habe ich für solche Aufgaben einen halben Arbeitstag gebraucht.

Zusammenarbeit mit dem Frontend

Die Frontend-Arbeiten im WTV-Projekt wurden von einer externen Firma übernommen, weshalb es die Aufgabe der Backend-Entwickler war, das Frontend Git-Repository, sowie die korrekte Anbindung der Frontend-Ressourcen vorzubereiten. Ich habe dafür beispielsweise die GitLab-CI Pipeline aufgesetzt und getestet, sodass unsere Frontend-Entwickler ihre Änderungen deployen und am Server testen konnten.

Außerdem habe ich viel mit den Frontend-Entwicklern kommuniziert, sodass sie auf von mir angelegten Content zugreifen konnten und um für ein besseres Verständnis der Coremedia-Datenstruktur zu sorgen.

Anpassungen an der Content-Management-App

In einem Großteil der User Stories, die ich im Laufe meines Praktikums bearbeitet habe, wurde die Anforderung gestellt, die Pflegebarkeit von bestimmten Content-Daten im Studio zu ermöglichen und anzupassen. Für solche Änderungen müssen häufig das Datenmodell erweitert und die Formulare der Studio-App angepasst werden. Während die Datenmodell Anpassungen eher einen geringeren Aufwand dargestellt haben, habe ich häufig viel Zeit (2-5 Tage jeweils) mit den Studio-Anpassungen verbracht.

Das Coremedia-Studio basiert auf ExtJS, worüber ich vor meinem Praktikum noch sehr wenig Wissen hatte. ExtJS ist aufgrund der API-Größe komplex und bietet nur wenige Ressourcen und Dokumentation, weshalb ich hierfür häufig meine Arbeitskollegen um Hilfe gebeten habe, oder im Notfall auf Trial-and-Error zurückgreifen musste. Trotzdem habe ich diese Aufgaben immer gerne übernommen, da ich sie als Herausforderung gesehen habe. Mit der Zeit habe ich viel über die Studio-Entwicklung gelernt, kleinere Aufgaben aus diesem Bereich sind fast schon trivial geworden. Ich habe durch diese Arbeiten das Debugging im Browser, JavaScript und den CKEditor besser kennen gelernt.

Implementierung von Logik

Das CoreMedia-Backend ist Spring basiert, weshalb die Implementierung von Logik meist in Java stattgefunden hat. Zu den Lösungen, die ich in Java umgesetzt habe, gehören:

- Facades, die die Übergabe von kalkulierten Daten an das Frontend ermöglichen
- Validatoren, die überprüfen ob erforderliche/korrekte Content-Daten im Studio gepflegt wurden
- Interceptors, die Prozesse im Studio erweitern, um bspw. Content korrekt zu initialisieren
- Richtext-Filter, die XML-haltigen Text für die Ausgabe als HTML vorbereiten

Hierbei habe ich besonders darauf geachtet firmeninterne, sowie allgemeingültige Code-Standards einzuhalten und gut zu dokumentieren/kommentieren, damit Arbeitskollegen und auch ich selbst zukünftige Anpassungen etwas leichter vornehmen können.

Zur Java-Programmierung gehörte jeweils auch das Erstellen von Unit-Tests, welche ich mit Spock/Groovy geschrieben habe. Obwohl ich im Laufe meines Praktikums einige Unit-Tests geschrieben habe, denke ich jedoch, dass ich an dieser Stelle noch viel zu lernen habe und hätte mir gewünscht, diese einfacher und schneller realisieren zu können.

Fehlerbehebung

Nachdem User Stories von unseren Entwicklern abgeschlossen wurden, gingen diese natürlich an die Quality Assurance, und nach erfolgreichem Testing an den Kunden zur Abnahme. Dabei sind immer mal Fragen, Änderungswünsche oder Fehler aufgekommen, die in Bug-Tickets festgehalten wurden. Diese Tickets waren meist recht schnell zu lösen, jedoch sind auch in seltenen Fällen erst zu diesem Zeitpunkt wirklich große Umsetzungsprobleme aufgefallen, welche dann im Team oder mit dem Kunden besprochen und gelöst werden mussten. Hierbei ging es auch immer um die Verhältnismäßigkeit – was ist der Use Case für die erfolgreiche Umsetzung des Tickets und wie viel Arbeit sollte man für diesen Use Case investieren.

Sonarqube

Zu Beginn meines Praktikums wurde vom Teamleiter für Coremedia-Entwicklung an mich herangetragen, dass all unsere Babel-Projekte beim Build in der Gitlab-Pipeline von Sonarqube auf Code-Qualität überprüft werden und dass auch das WTV-Projekt diesen sogenannten „Build-Breaker“ bräuchte. Die Aufgabe, dies umzusetzen habe ich als User Story in das Projekt getragen und auch gleich an mich genommen. Da das WTV-Projekt mit einer neueren (Java-)Version läuft, war es allerdings nötig eine neue Sonarqube-Instanz aufzusetzen.

In Zusammenarbeit mit Administratoren-Kollegen habe ich zunächst eine neue Instanz auf der dafür vorgesehenen Virtual Machine aufgesetzt und eingerichtet, um das WTV Projekt darin zu testen. Nach kurzer Zeit würde die zugehörige Datenbank jedoch gelöscht, da die verwendete VM nicht mehr den Sicherheitsstandards entsprach und schon bald ersetzt werden sollte. Da also auch die alte Sonarqube-Instanz gelöscht werden sollte, musste der Build-Breaker für alle Babel-Projekte angepasst werden.

Dabei fiel auf, dass die Testabdeckung in der neuen Version nicht mehr korrekt berechnet wurde. Um dies zu korrigieren habe ich, sowie einige andere Kollegen, recht lange recherchiert, getestet und probiert, bis wir zu den korrekten Maven-Anpassungen gelangt sind. Da diese Prozesse alle viel Kommunikation und Arbeiten von verschiedensten Personen erfordert haben, hat sich das Hinzufügen des Build-Breakers für WTV sehr lange hingezogen.

Als letzten Teil meiner Sonarqube zugehörigen Arbeiten habe ich das Gespräch mit unserem Kunden gesucht, um die von Ihnen gewünschten Qualitätsansprüche an unseren Code in Sonarqube zu integrieren und den Code auf den entsprechenden Standard zu bringen. Dabei habe ich viele der Code-Regeln etwas besser verinnerlichen können und die Chance gehabt meinen Code zu verbessern.

Sonstiges

Da ich in einem SCRUM-Projekt gearbeitet habe, wurde mir ermöglicht, verschiedenste Aufgaben, sowie die Verantwortung für diese zu übernehmen. Beispielsweise habe ich eine Zeit lang die End-of-Sprint-Deployments auf die „Staging“-Umgebung durchgeführt und dokumentiert. Es war wichtig, die Zeiträume jeweils mit dem Team und den Kunden zu kommunizieren, da jegliche Tests, sowie die Abnahme durch den Kunden auf dieser Umgebung stattgefunden haben.

Für meine Arbeitskollegen habe ich außerdem regelmäßig Code-Reviews durchgeführt, genauso wie ich meinen Code auch immer von mindestens einer Person prüfen lassen habe. Beim Code-Review habe ich nicht nur viel über mögliche Umsetzungen von meinen Kollegen gelernt, sondern auch immer einen guten Überblick über das Projekt und den bestehenden Code behalten.

Oft habe ich auch mit meinen Arbeitskollegen gemeinsam „gedebuggt“ oder nach Lösungen gesucht, wenn ich oder ein anderes Teammitglied nicht weiterwusste. Meist haben wir dann per Screensharing gemeinsam die Probleme untersucht, und wenn das Prinzip der „Debug-Duck“ nicht geholfen hat sind wir durch Diskussionen und gemeinsame Überlegungen immer zu einer Lösung gelangt.

In seltenen Fällen war es nötig Skripte zu schreiben, um bspw. bestehenden Content am Server im Batch anzupassen. Da ich bei Babel schon vor meinem Praktikum häufig verschiedene Skripte geschrieben habe ist mir dies sehr leichtgefallen – insbesondere mit Groovy.

Eigene Arbeitsergebnisse

Da ich an recht vielen kleinen User Stories gearbeitet habe, lässt sich die Menge meiner Arbeitsergebnisse nur schwer zusammentragen. Zielsetzung meines Praktikums war es nicht, einzelne Projekte oder vollständige Applikationen abzuschließen, sondern im Team am Fortschritt eines sehr großen Projektes zu arbeiten. Einige meiner abgeschlossenen Aufgaben lassen sich hier dennoch darstellen.

Anpassungen am CKEditor

Der CKEditor ist ein JavaScript-basierter Texteditor, der standardmäßig im Coremedia-Studio enthalten ist. Er bietet einige wichtige Standardfunktionen (Textformatierungen wie fett, kursiv; Verlinkungen, Listen), jedoch war es erforderlich diese Funktionen zu erweitern.

Ich habe beispielsweise die Anpassung von internen und externen Verlinkungen durchgeführt. Dafür war es notwendig die Dialoge (siehe Bild 1 und 2) jeweils anzupassen. Dazu musste ich diese jedoch zuerst in ihrer Originalform nachstellen, da der Originalcode mir nur in JavaScript zur Verfügung stand und die Anpassung in ExtJs-Form in ActionScript und mxml stattfinden musste. Durch Reverse Engineering ist es mir gelungen die Funktionsweise der Dialoge in ExtJs nachzustellen und anschließend jeweils die Funktionalität für sogenannte Button Links (erste Verlinkung in Bild 3) hinzuzufügen. Dort war es auch notwendig mich mit dem Frontend abzustimmen, um die korrekte Darstellung im Frontend via CSS zu garantieren.

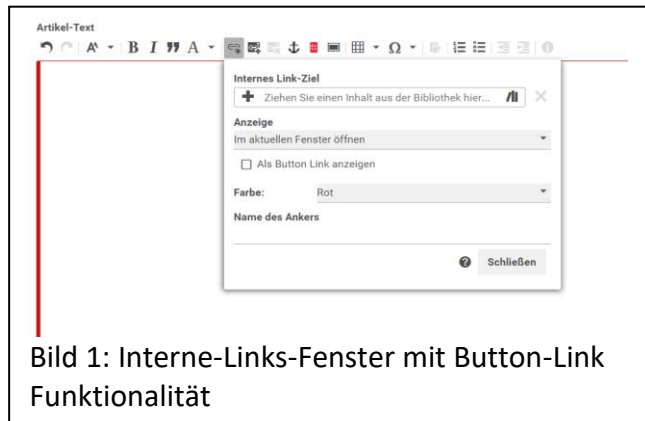


Bild 1: Interne-Links-Fenster mit Button-Link Funktionalität

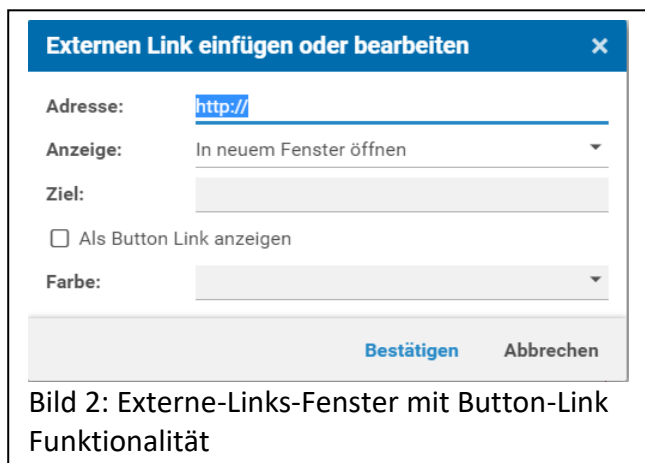


Bild 2: Externe-Links-Fenster mit Button-Link Funktionalität

Zusätzlich zu den Button Links musste hier auch beachtet werden, dass bestimmte Links zugehörige CSS-Klassen erhalten sollten. Beispielsweise sollten PDF-Downloads „pdf“ als CSS-Klasse erhalten oder externe Links „extern“, damit das Frontend dies entsprechend auf der Website anzeigen konnte (Bild 3, zweite und dritte Verlinkung). Das korrekte Auslesen des Linktypen habe ich in Java über einen „Richtext-Filter“ realisiert, der das xml der Im CKEditor erstellten Text-Daten gelesen und mit den entsprechenden CSS-Klassen bearbeitet.

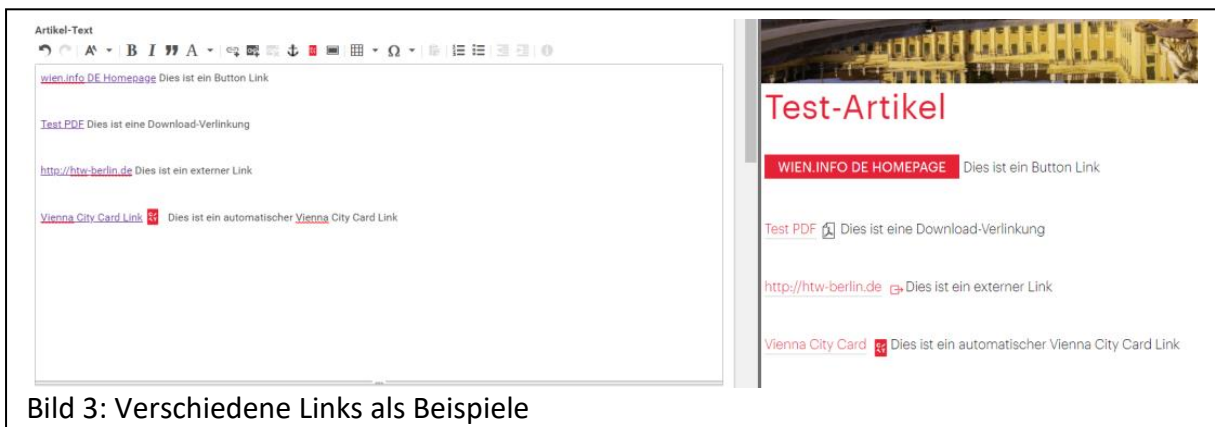


Bild 3: Verschiedene Links als Beispiele

Eine weitere Aufgabe aus diesem Bereich waren die „Vienna City Card“-Links. Dafür sollte ein Button in der Toolbar hinzugefügt werden, dessen Funktionalität darin bestand einen Link auf die Vienna-City-Card-Website in den Text zu fügen. Um diesen zu kennzeichnen sollte das Icon im Text-Editor und auf der Website direkt angezeigt werden (Bild 3, vierte Verlinkung). Auch hier musste ich mit einem Richtext-Filter arbeiten.

Detailnavigation

Die Detailnavigation war eine Aufgabe, bei der ich sowohl Backend- als auch Frontend-Aufgaben übernommen habe. Prinzipiell ging es hierbei darum, eine Daten-Vererbungslogik im Content aufzubauen. Es mussten zunächst nur drei Datenfelder im Studio via mxml/ActionScript hinzugefügt werden (Bild 4).

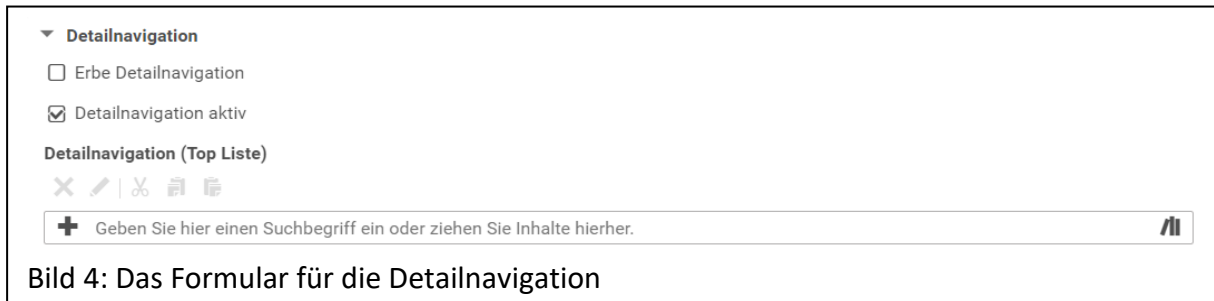


Bild 4: Das Formular für die Detailnavigation

Um die gespeicherten Daten aus diesen Feldern korrekt an das Frontend weiterzugeben, habe ich eine Facade-Klasse erstellt, die die Vererbungslogik der Detailnavigation enthalten hat und die darzustellenden Daten ausgegeben hat. Diese Daten habe ich in Freemarker- und Handlebars-Templates verwendet, um sie auf der Webseite anzeigen zu lassen (Bild 5). Hier musste ich erneut eng mit dem Frontend zusammenarbeiten, um die korrekte Listendarstellung zu erzielen.



Bild 5: Die Detailnavigation im Frontend

Bilduploads

Ein weiteres interessantes Thema, an dem ich viel gearbeitet habe, waren die Bilduploads. Es war gewünscht, dass Bilder beim Upload im Studio eine fortlaufende, einzigartige Bildnummer zugewiesen bekommen. Außerdem sollte aus den Bild-Metadaten das Copyright ausgelesen und in die Coremedia-Content-Daten geschrieben werden.

Um dies umzusetzen habe ich einen Upload-Interceptor in Java geschrieben, der den Prozess des Bild-Uploads erweitert hat. In diesem zusätzlichen Schritt war es möglich, die letzte verwendete Bildnummer zu inkrementieren, das Copyright aus den Upload-Daten zu lesen und beides in die Content-Daten zu schreiben. Um die Einzigartigkeit der Bildnummer zu garantieren, war es außerdem wichtig hierbei auf Thread-Sicherheit zu achten.

Art und Umfang der Unterstützung durch die Ausbildungsstelle

Babel hat mich zu einem großen Teil unterstützt, indem jegliche meiner Fragen so schnell wie möglich beantwortet wurden. Da ich zum Anfang meines Praktikums wenig Erfahrung mit der Studio-Entwicklung hatte wurde für mich und einen weiteren Kollegen eine zwei-tägige Schulung zum Thema veranstaltet. Der Schulungsleiter stand auch nach der Schulung immer

für Fragen zu Verfügung und hatte fast immer Lösungsansätze oder Beispielcode anzubieten. In anderen Themengebieten hat sich auch immer jemand gefunden, der mir schnell helfen konnte, falls ich offene Fragen hatte.

Außerdem habe ich von Babel, und auch im WTV-Projekt selbst, regelmäßiges Feedback zu meiner Arbeit erhalten, was mich motiviert hat. Auch wurde ich dabei unterstützt zusätzliche Aufgaben zu finden für die ich Verantwortung übernehmen konnte, um bspw. die Babel-Standards in das Projekt zu übertragen oder um allgemeingültige Lösungen im Projekt zu erforschen und weitertragen zu können.

Alle zwei Wochen hat außerdem die „Coremedia-Runde“ mit all unseren Coremedia-Backend-Entwicklern stattgefunden, in welcher neue Lösungen aus unseren Projekten vorgestellt und generelle Arbeitsweisen, sowie interne Standards besprochen und festgelegt wurden. Dabei konnte ich viele Informationen und neue Ideen sammeln, die mir immer mal wieder die Arbeit erleichtert haben. Eigene Ideen, Lösungen und Diskussionsthemen konnte man hier auch immer einbringen.

Auch unterstützt Babel mich und all meine Kollegen dabei in der Corona-Zeit sicher zu sein. Von Anfang an war es mir gestattet im Home-Office zu arbeiten, und es wurden wöchentliche Video-Meetings veranstaltet, um den Kontakt zu den Kollegen nicht zu verlieren. Auch wenn man sich gerne persönlich gesehen hätte, hat dies die angenehme Stimmung in der Firma gewahrt. Es wurden auch regelmäßige Updates von der Geschäftsführung an alle Mitarbeiter versendet, um uns über die Situation auf dem Laufenden zu halten und Möglichkeiten ins Büro zurückzukehren zu besprechen.

Bezüge zwischen Studium und Fachpraktikum

Nützlichkeit der bisherigen Lehrveranstaltungen

Aus meinem Medieninformatikstudium habe ich hauptsächlich die Java-Grundkenntnisse aus Informatik I anwenden können. Da Informatik III in meinem Jahrgang zu einem großen Teil ausgefallen ist konnte ich leider nur wenig Wissen zum Thema Softwarearchitektur und Entwurfsmuster mitbringen. Ein wenig konnte ich das fehlende Wissen aus diesem Kurs jedoch dank meines Teamleiters in der Backend-Entwicklung nachholen. Außerdem hat mir der Netzwerke-Kurs dabei geholfen Web-Technologien besser zu verstehen, welche ich in der Backend-Entwicklung natürlich auch kennen sollte. Jedoch hat auch dieses Wissen bisher wenig direkte Verwendung gefunden.

Eine Anwendung für meine bisher belegten grafisch orientierten Kurse (Computergrafik, Bildverarbeitung, Grundlagen digitaler Medien) konnte ich in der Backend-Web-Entwicklung leider nicht finden. Auch mein Wissen aus dem Datenbanken-Kurs konnte ich leider nicht verwenden, da die Datenbanken von den Administratoren aufgesetzt wurden.

Nützlichkeit des Fachpraktikums für das weitere Studium

Ich denke die wichtigste erlernte Fähigkeit aus meinem Praktikum für das Studium ist die Teamarbeit. Ich habe gelernt in einem Team Code zu entwickeln, Umsetzungskonzepte zu besprechen und Aufgaben zu planen. Ich habe viel über Git(-Lab) gelernt und erste Erfahrungen mit dem Aufsetzen von Projekten gemacht. Diese Fähigkeiten werden mir insbesondere im anstehenden Programmierprojekt im 5. IMI-Semester zugutekommen.

Außerdem habe ich durch die Agile Arbeitsweise viel darüber gelernt, die Komplexität, und den zeitlichen Aufwand von Programmier-Aufgaben einschätzen. Dies werde ich sicherlich im Studiumsalltag generell gut gebrauchen können, aber auch in Hinsicht auf meine Bachelorarbeit.

Zusätzlich habe ich einige unterschiedliche Web-Technologien angewendet, welche ich eventuell in den Web-Development Kursen des IMI-Studiengangs verwenden kann. Etwas Erfahrung mit einem Build-Tool wie Maven oder einem Container-System wie Kubernetes zu haben, kann sich auch an verschiedensten Stellen als hilfreich erweisen.

Zusammenfassung und Ausblick

Aufgetretene Probleme

Das größte Problem in meinem Praktikum war wohl Corona – jedoch denke ich, dass Babel und Ich das Beste aus der Situation gemacht haben. Ich habe im Home-Office gearbeitet, wie all meine anderen Kollegen auch, und ich denke auch das war eine Erfahrung wert. Ich fand es sogar erstaunlich, wie gut die Zusammenarbeit im Projekt auch ohne persönlichen Kontakt funktioniert hat.

Eine größere Herausforderung war auch die Erarbeitung von Umsetzungskonzepten in Zusammenarbeit mit mehreren Abteilungen. Dort sind wiederholt Wartezeiten auf Bestätigungen und auch Missverständnisse aufgetreten, die ich gerne vermieden hätte.

Für mich hat sich auch herausgestellt, dass eine 40-Stunden-Arbeitswoche unglaublich anstrengend, aber auch unglaublich belohnend sein kann. Es gab Wochen im Praktikum, in welchen ich mich immer nach der Arbeit müde und unzufrieden gefühlt habe, aber auch Wochen, in denen ich mich sehr über meine Fortschritte gefreut habe und am liebsten weiter gebastelt und gerätselt hätte. Ich hatte vorher noch nie Vollzeit gearbeitet und ich bin im Nachhinein sehr dankbar für meine 8 Urlaubstage. Für zukünftige Vollzeit-Jobs werde ich definitiv darüber nachdenken weniger Wochenstunden anzufragen.

Verbesserungsmöglichkeiten

Ich habe kein negatives Feedback für Babel, im Gespräch mit meinem Praktikumsbetreuer habe ich dies auch schon erwähnt. Babel hat eine unglaublich angenehme Arbeitsatmosphäre (sogar im Home-Office) und bietet sehr gute Arbeitsbedingungen.

Ich denke Verbesserungsmöglichkeiten gäbe es hauptsächlich im WTV-Projekt, jedoch arbeitet die Projektleitung dauerhaft daran bekannte Probleme zu beseitigen und hatte immer ein offenes Ohr, wenn neue Probleme an sie herangetragen wurden.

Berufliche Perspektiven

Da ich vor meinem Praktikum schon bei Babel als Werkstudent gearbeitet habe, werde ich dies auch weiterhin neben meinem Studium tun. Ich bin mittlerweile sehr gut eingearbeitet und bin der Meinung auch ein wichtiger Teil meines Projekt-Teams geworden zu sein. Als Werkstudentin bietet Babel mir großartige Arbeitsbedingungen und die Möglichkeit neben meinen Unikursen Praxiserfahrung zu sammeln. Ich freue mich darauf das WTV-Projekt weiter zu bearbeiten und gemeinsam mit dem Team den Live-Gang zu erreichen.

Durch mein Praktikum bzw. meine Arbeit bei Babel habe ich viel Erfahrung mit verschiedenen Backend-Web-Technologien gesammelt, und insbesondere Spring kenne ich nun recht gut. Auch habe ich nun viel mehr Erfahrung mit Java-Entwicklung vorzuweisen, weshalb mir das Praktikum sicherlich einige zukünftige Jobmöglichkeiten eröffnet. Sammle ich noch etwas mehr Erfahrung im Frontend-Bereich könnte ich sogar als Full-Stack-Entwicklerin arbeiten.