
PRAKTIKUMSBERICHT

Herr
Hannes Steiner

Entwicklung einer Dokumenten-Scanner-App

**Digitalisierung der Verwaltung an der Hochschule
Mittweida**

2020

PRAKTIKUMSBERICHT

Entwicklung einer Dokumenten-Scanner-App

**Digitalisierung der Verwaltung an der Hochschule
Mittweida**

Autor:
Hannes Steiner

Studiengang:
Softwareentwicklung

Seminargruppe:
IF17wS-B

Erstprüfer:
Prof. Dr. Mark Ritter

Zweitprüfer:
N.N.

Mittweida, März 2020

Bibliografische Angaben

Steiner, Hannes: Entwicklung einer Dokumenten-Scanner-App, Digitalisierung der Verwaltung an der Hochschule Mittweida, 27 Seiten, Hochschule Mittweida, University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften

Praktikumsbericht, 2020

Dieses Werk ist urheberrechtlich geschützt.

I. Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Abkürzungsverzeichnis	IV
1 Einleitung	1
2 Hochschule Mittweida	3
3 Problemstellung	5
3.1 Digitalisieren der Klausur-Daten	5
3.2 Klausuren-Vorlage	5
3.3 Vorlage verbessern	6
3.4 Weitere Anmerkungen	6
4 Anforderungen	7
5 Konzept des Dokumenten-Scanners	9
6 Entwicklung des ersten Prototyps	11
6.1 Planung und Vorbereitung	11
6.2 Entwurf und Design	11
6.3 Implementierung	12
6.3.1 Template erstellen	12
6.3.2 Template speichern	12
6.3.3 Template verwenden	12
6.4 "Abnahme"	12
7 Weitere Entwicklung und Besonderheiten	15
7.1 App-Architektur	15
7.2 API	15
8 Grenzen der App	17
8.1 Probleme beim Erkennen von Dokumenten	17
8.2 Probleme der Klausur-Vorlage beim Scannen	17
8.3 Weitere	17

9	Templates	19
10	Ausblick.....	21
A	Tätigkeitsbericht	23
	Literaturverzeichnis	25

II. Abbildungsverzeichnis

III. Tabellenverzeichnis

IV. Abkürzungsverzeichnis

bzw. beziehungsweise, Seite 5

Fakultät CB Fakultät für angewandte Computer- und Biowissenschaften, Seite 5

OCR optical character recognition, deutsch: optische Zeichen Erkennung, Seite 9

1 Einleitung

Digitalisierung wird gängig als Integration von digitaler Technologie in den Alltag verstanden, und soll helfen Zeit einzusparen [1]. Mit diesem Gedanken initiierten die Mitarbeiter Holger Langner und Falk Schmitsberger der Hochschule Mittweida, das Projekt *Memo Space*. Im Zuge dessen sollen kleinere Forschungsergebnisse entstehen, die richtungsweisend für die Digitalisierung der Verwaltung von Lehr- und Forschungseinrichtung sind.

Eine der ersten Ideen ist es, die Arbeit von Klausur-Prüfern zu erleichtern. Diese müssen, nachdem die Klausuren kontrolliert wurden, die Zensur, sowie die Eckdaten der Studenten, in ein digitales Format bringen. Grund dafür ist, dass die Noten in das Notensystem der Einrichtung eingetragen werden müssen.

Im Rahmen eines Forschungspraktikums an der Hochschule Mittweida arbeiteten der Student Tobias Kallauke und der Verfasser, gemeinsam an einer Lösung zur Digitalisierung dieses Arbeitsschrittes. Beide arbeiteten zum einem Teil zusammen und auch getrennt an Aufgaben... Sie entwickelten... (Klausuren-Vorlage, App, Server, OCR? evtl.) / Dabei entstanden .../ Der Verfasser programmierte...

Was noch?

2 Hochschule Mittweida

3 Problemstellung

Hochschulmitarbeiter sitzen zum Ende eines Semesters über Tage an der Kontrolle von Klausuren. Diese Aufgabe muss stets mit hoher Konzentration erledigt werden, und lässt sich aber in den meisten Fällen nur schwer durch Maschinen ersetzen. Unter keinen Umständen dürfen bei der Bewertung Fehler vorkommen, was jedoch bei der kognitiven Last der Prüfer immer wieder passiert. Auch nach der Durchsicht der Prüfungsaufgaben ist eine hohe Achtsamkeit wichtig. Denn anschließend wird die Benotung in eine digitale Tabelle überführt. In diese muss die Matrikelnummer, der Vor- und Nachname, sowie die Note des Studenten eingetragen werden. Hier kommt es vor allem bei der Matrikelnummer und der Zensur auf die Richtigkeit jedes Zeichens drauf an.

3.1 Digitalisieren der Klausur-Daten

Für genau diesen Vorgang des Digitalisierens wird eine Lösung gesucht. Die Prüfer sollen so bequem und möglichst zeitsparend diese Aufgabe verrichten, ohne dabei ihre Aufmerksamkeitsspanne zu überlasten. Des Weiteren müssen die Ergebnisse der Prüfungen, sowie die Eckdaten der Studenten in ein geeignetes digitales Format gebracht werden, um es der Notenfreigabe weiterzuleiten. Darüber hinaus empfiehlt es sich, digitale Kopien der Klausuren abzuspeichern, da/um ... (warum genau, sollen die gespeichert werden? meine/unsere Idee Online Klausureneinsicht -> ins Fazit/Ausblick (hat viele "Probleme"))

3.2 Klausuren-Vorlage

- ist es eher ein Gestaltungsleitfaden "style guide" oder eine Klausuren-/Layout-Vorlage?

Eine Klausuren-Vorlage bzw. ein Gestaltungsleitfaden für Klausuren der Fakultät *Angeordnete Computer- und Biowissenschaften* (Fakultät CB) bietet außerdem die Möglichkeit der Kontrolle des Prüfers an. Genauer ist es durch das vorgegebene Layout der Klausur möglich, einen Teil der Arbeit des Prüfenden auf Fehler zu untersuchen. Auf dem Deckblatt der Klausur ist eine Tabelle, mit drei Zeilen und für jede Aufgabe eine Spalte. In der ersten Zeile befinden sich die Nummern der Aufgaben. In der zweiten, die zu erreichenden Punkte der Aufgabe. Und in der dritten Zeile trägt der Prüfer die erbrachten Punkte des Studenten ein. Unter der Tabelle befindet sich ein Feld für die erreichte Gesamtpunktzahl, sowie ein Feld für die aus den Punkten resultierende Note. Die Punkte pro Aufgabe, die Gesamtpunktzahl und die Zensur stehen in (Korrelation?) Relation zu einander, sodass aus den Punkten der Aufgaben die beiden anderen Felder errechnet werden und mit den Ergebnissen des Prüfers abgeglichen werden könnten.

Eine weiteres Merkmal der Klausuren-Vorlage ist ein Feld, für die vom Studenten erreichten Punkte über jeder Aufgabenstellung. Die dortige Angabe sollte mit der, in der Tabelle auf dem Deckblatt übereinstimmen und bietet somit noch eine weitere Möglichkeit der Kontrolle an.

3.3 Vorlage verbessern

Templates entwerfen, die optimiert für die Digitalisierungs-Prozesse sind ...

3.4 Weitere Anmerkungen

TODO:

Ferner soll bei der Lösung von der Anschaffung neuer Technologie und Geräte abgesehen werden. Grund dafür sind neben den Anschaffungskosten, die Idee, dass das Ergebnis des Forschungsprojekts in weiteren Lehr- und Forschungseinrichtung Anwendung finden sollte.

Was noch?

Soll hier schon rein, dass eine App, ein Webportal, etc. entstehen soll? Oder soll die Grund Idee zur Lösung in ein extra Kapitel?

4 Anforderungen

Es soll eine iOS-App entstehen, die - ios app:

- ios 13.0 > version, wegen swiftui und vision / vision kit -> bedeutet geräte ab dem 6s, aber nicht ältere (Prozessor ist entscheidend)
- auch iPads sind recht einfach möglich dank SwiftUI!!!
- vlt ein paar worte zu swift, swiftui, vision und visionkit ...
- dokumente ein scannen
- vorlagen erstellen, um wichtige Bereiche zu erfassen
- mit der vorlage dokumente einscannen
- auf dem gerät oder auf dem server die texte erkennen
- vorlagen auf einem server speichern und abrufen
- ergebnisse an den server schicken, nicht auf der App speichern
- Datenschutz/sicherheit beachten, auch bei Server und APIs
-

5 Konzept des Dokumenten-Scanners

Aus der Problemstellung heraus entstand die Idee einer Dokumenten-Scanner App. Diese soll die wichtigen Daten auf dem Deckblatt einer Klausur, wie Vor- und Nachname des Studenten, seine Matrikelnummer sowie die Note erkennen, digitalisieren und in ein für die Notenfreigabe geeignetes Format bringen. Die digitalisierten Daten sollen anschließend an einen Server gesendet werden, wo sie und die beim Einscannen entstandenen Bilder gespeichert werden. Bei Verwendung der Klausuren-Vorlage der Fakultät CB oder ähnlichen, die eine Punkteübersicht haben, soll es außerdem möglich sein, die Punkte sowie die Note auf Richtigkeit zu überprüfen.

Zur Digitalisierung und Kontrolle der Daten werden Scan-Vorlagen verwendet. Diese muss vor dem Einscannen der ausgefüllten Klausuren erstellt werden, um zu gewährleisten, dass die richtigen Daten digitalisiert werden. Beim Erstellen einer Scan-Vorlage geht man wie folgt vor:

- Zuerst wird jede Seite der Klausur fotografiert. Dabei wird in jedem Bild das Dokument erkannt, vom Hintergrund getrennt oder genauer gesagt ausgeschnitten und perfekt ausgerichtet. (Kontrast erhöht -> Schrift besser zu erkennen, ...)
- Anschließend markiert man diejenigen Regionen auf jeder Seite, die zu digitalisieren und/oder kontrollieren sind.
- Des Weiteren kann man aus allen markierten Regionen diese auswählen, die in Relation stehen. Diesen Relationen werden danach noch weitere Eigenschaften zugeteilt, um sie später bei der Kontrolle richtig zu analysieren. So eine Eigenschaft könnte sein, dass es sich um einen Vergleich zwischen zwei in Relation stehenden Regionen handelt. Z. B. ist die eine Region die Zelle in der Punkteübersichts-Tabelle, in der die erreichten Punkte zu Aufgabe 1 rein geschrieben werden sollen und die zweite Region die Stelle über der Aufgabenstellung von Aufgabe 1, in der ebenfalls die erreichten Punkte eingetragen werden sollen. Weitere Beispiele für solche Relationen und Eigenschaften sind die Summanden für die erreichte Gesamtpunktzahl und die Summe selbst, oder auch noch die daraus resultierende Note.
- Als letztes speichert man die Vorlage ab, welche dann automatisch an einen Server gesendet wird, so dass andere genauso diese Vorlage nutzen können.

Nach der Erstellung einer Vorlage kann das Dokument digitalisiert werden. Dazu scannt man die Seiten in derselben Reihenfolge, wie in der Vorlage ein. Die App digitalisiert mithilfe der Vorlage und optical character recognition (OCR) den Inhalt in den vordefinierten Regionen. Im Anschluss an die Texterkennung kann das Resultat noch mal überprüft und korrigiert werden, bevor es an den Server zum Abspeichern gesendet wird.

... noch mehr? (Auf Tobias warten...)

Ein wichtiger Vorteil einer App gegenüber anderen Lösungen ist, dass so gut wie jeder Mitarbeiter an einer Lehr- und Forschungseinrichtung ein eigenes oder Zugang zu einem Smartphone oder Tablet hat.

- Einspeisung von erwarteten Ergebnissen noch mit rein nehmen? (Noten: ,0 ,3 oder ,7; Seminargruppen und Klausur Teilnehmer.)
- Noch mehr zu OCR sagen: was es ist, wie es geht, komplett eigenes Kapitel?

6 Entwicklung des ersten Prototyps

Vor Beginn des Praktikums saßen die Praktikum-Betreuer Falk Schmidberger, Robert Manthey der Hochschule Mittweida, mit Tobias Kallauke und mir zusammen. Wir diskutierten, welches Thema für das Praktikum geeignet ist, wie eine Art Durchführbarkeits- / Machbarkeitsstudie und entschieden uns für den Dokumenten-Scanner. Da ich privat Erfahrung in Swift sowie iOS-Entwicklung gesammelt habe, sollte ich damit anfangen den Scanner als iOS-App zu entwickeln. Wir machten uns auch direkt noch Gedanken über die Anforderungen an die App und definierten, wie das zukünftige Software-System mit Server und Backend aussehen könnte.

6.1 Planung und Vorbereitung

In der aller ersten Woche habe ich mich hauptsächlich mit der Problemstellung auseinander gesetzt, Ideen entwickelt, Problemanalyse betrieben und einen kleinen Prototypen entwickelt. Ich begann damit die Dokumentation von *Vision*, *VisionKit* und *PhotoKit* von *Apple* an zu schauen. Danach hatte ich einen guten Überblick darüber, was ich noch zu entwickeln habe und was mir Frameworks abnehmen können. Zum einen war das Erkennen und gerade ziehen von Dokumenten in Echtzeit in der Kamera, sowie die Texterkennung auf Bildern schon in den Frameworks implementiert. Nebenbei entdeckte ich eine Beispiel-App von *Apple* [2], in der es um Erkennen von Objekten in Standbildern ging. Diese schaute ich mir an, um das Vision Framework genauer zu verstehen, sodass ich mich gut eingearbeitet habe.

- Scrum-Prinzip mit rein, ...

6.2 Entwurf und Design

Ich begann anschließend damit, mir Gedanken zu dem Workflow der App zu machen, erstellte einen minimalen Projektplan, als Leitfaden für die Prototypen Entwicklung und die spätere Weiterentwicklung. Außerdem fertigte ich für jede View ein grobes Design an, sodass es möglichst schnell und effizient mit SwiftUI umzusetzen war. Danach setzte ich ein Repository in GitHub auf und begann mit dem Programmieren des Prototyps.

- Nochmal umschreiben: Redux mit rein

6.3 Implementierung

Einzelne Schritte als eigene Section mit Verlauf der jeweiligen Entwicklung: erst das, dann kam das, dann musste es so, ...

6.3.1 Template erstellen

6.3.2 Template speichern

6.3.3 Template verwenden

...

Die Beispiel-App von *Apple* nahm ich mir als Vorbild für die Umsetzung der Texterkennung.

...

6.4 "Abnahme"

Abnahme ...

Weitere Punkte die in Kapitel 5 rein könnten/sollten:

- beschreiben wie die einzelnen Views/Seiten nun aussehen und funktionieren? oder eher den Prozess der Entwicklung?
- Bild mit Wireframe aller Views und deren Workflow?
- AppState erklären? (Single source of truth) -> später AppStore mit mehreren States
- erklären deskew / Dokument ausrichten?
- Umrechnung der Bilder vom Template aus und zuschneiden der neuen Regionen erklären
- Attribute hinzufügen (den Vorgang) -> Rechteck einzeichnen...

- Verwendete Programme, Sprache, etc.
- ein paar Worte, wie gut die Entwicklung lief (Simulator vs. echtes Gerät), wo sind die Grenzen des Simulators (Keine Kamera), wo waren Probleme mit dem echten Gerät...

7 Weitere Entwicklung und Besonderheiten

7.1 App-Architektur

- redux like app store mit states erklären? (single source of truth)

7.2 API

- auf die API eingehen? oder nur tobias?

7.3

-

8 Grenzen der App

8.1 Probleme beim Erkennen von Dokumenten

- Gleich-farbiger Hintergrund
- Wenig Licht
- Hintergrund mit starken Kanten
- Runde Ecken, keine Ecken
- Starke Kanten im Bild (schwarzer Kreditkartenstreifen)
- zu jedem möglichem vlt. dann ein Beispiel Bild

8.2 Probleme der Klausur-Vorlage beim Scannen

- Schrift zu klein
- zu wenig Platz
- zu viel Platz
- Überschneidungen
- ...

8.3 Weitere ...

- ...

9 Templates

- Template und den Entwicklungsprozess vorstellen,
- wieso weshalb warum muss das nun so aussehen?
- Was kann an der neuen Vorlage immer noch optimiert werden?
- Welche Probleme konnten behoben werden?

10 Ausblick

- Es müssen nicht nur Klausuren sein, sondern alles mögliche (Krankenscheine, Urlaubsscheine, was auch immer, nur DB muss angepasst werden. (oder automatisches generieren von DB-Tabellen an Hand der Template Attribute))
- Server mit Bildern als Klausuren-Einsicht nutzen -> allerdings viele Probleme (Klausuren würden kopiert werden -> Profs mehr Arbeit, keinen direkten Kontakt zum Prof wegen Fragen, Verbesserungen oder Anmerkungen, ...)
- QR-Code-Idee um Vorlagen weg zu lassen -> Programm oder Plugin (Word/LaTex) was die QR-Codes dann automatisch erstellt und richtig einfügt. (Wichtige Daten sind da hinterlegt, Vor- und Nachteile von QR-Codes, ...)
- Ist die App nun schneller als der normale Umgang bleibt offen -> Bachelorarbeit knüpft da an...

Anhang A: Tätigkeitsbericht

Datum	Tätigkeit
24.02. - 01.03.	<p>Ich habe mich mit der Problemstellung auseinander gesetzt, Ideen gesammelt, Problemanalyse betrieben und einen kleinen Prototypen entwickelt. Dazu erstellte ich einen minimalen Projektplanung, arbeitete mich in die Frameworks <i>Vision</i> und <i>VisionKit</i> ein und setzte eine Versionsverwaltung auf. Zusätzlich suchte ich nach einer passenden App-Architektur, die geeignet für das deklarative GUI-Framework <i>SwiftUI</i>, sowie für asynchrone Aufgaben, wie z. B. API-Aufrufe ist. Dabei stieß ich auf <i>Cleancode Architecture</i> und <i>Redux</i>.</p> <p>Am Ende der Woche konnte man in der App Vorlagen mit einer Seite erstellen. Das heißt man konnte ein Foto machen, aus welchem das Dokument rausgeschnitten und anschließend ausgerichtet wurde. Weiter war es möglich Regionen auf dem Dokument zu markieren und diese in der Vorlage abspeichern. Ansonsten konnten die Vorlage schon dazu benutzt werden, um die Regionen auf dem neuen Foto heraus zu schneiden.</p>
02.03. - 08.03.	<p>In dieser Woche habe ich die Texterkennung auf den berechneten Regionen eines neuen Fotos implementiert, den Workflow sowie viele andere Kleinigkeiten in der App verbessert und alle Fehler der letzten Woche behoben, sodass ich neue Dinge implementieren konnte. Zudem probierte ich CI sowie Lint für das Projekt aus. Da CI für eine iOS-App mit <i>GitHub Actions</i> schwer aufzusetzen war und ab April etwas kosten würde, lies ich es sein. Des Weiteren pflegte ich das Projekt Management durch <i>Issues</i> und <i>Project-Boards</i> in GitHub. Anschließend programmierte ich den App-Workflow so um, dass nun mehr als eine Seite aufgenommen und analysiert werden konnte.</p> <p>Abgesehen von neuem Quellcode fing ich an den Praktikumsbericht zu schreiben und arbeitet mich dafür in \LaTeX und die Bachelorarbeits-Vorlage für \LaTeX der Hochschule Mittweida ein.</p>
09.03. - 15.03.	<p>Zu Beginn der dritten Woche schaute ich mir Möglichkeiten für serverseitiges OCR an. Genauer sammelte ich Informationen zu dem Framework Vapor und Swift unter Linux. Jedoch funktionieren die Frameworks <i>Vision</i> und <i>CoreML</i> von <i>Apple</i> unter Linux nicht, weshalb sich IronOCR als beste Option herausstellte. Ich entwickelte ein Datenbankmodell, mithilfe der in der App verwendeten Datentypen und erstellte dazu noch eine JSON-Struktur die später für die APIs verwendet werden könnte. Außerdem gab es ein Meeting, in dem wir unseren aktuellen Stand präsentieren sollten, um weitere Schritte und Aufgaben zu planen. Bis zum Ende der Woche arbeitete ich weiter an meinem Beleg und schrieb den Datenfluss in der App um. Nun ähnelt er sehr dem Redux-Model.</p>

Datum	Tätigkeit
16.03. - 22.03.	Anfangs habe ich weiter an meinem Praktikumsbericht geschrieben, neue Issues hinzugefügt und bearbeitet. Außerdem gepflegte ich die Dokumentation und betrieben Projekt Management, um nun Links zwischen Regionen hinzuzufügen. Dabei entstanden neue Views und der Redux-Store musste dadurch angepasst werden. Es kam eine Erweiterung für die Texterkennung hinzu, so dass man durch die Auswahl eines Datentyps, das Resultat der Erkennung verbessern konnte. Des Weiteren habe ich bis zum Ende der Woche die Vergleich-Links vollständig implementiert und die App auf Fehler und Abstürze kontrolliert, sowie den Beleg um einige Kapitel erweitert.
23.03. - 30.03.	...

Literaturverzeichnis

[1] Michael Graf, Partner bei PwC

<http://www-cs-faculty.stanford.edu/~uno/abcde.html>

[2] Apple Developer - Sample Code, Detecting Objects in Still Images

https://developer.apple.com/documentation/vision/detecting_objects_in_still_i

Erklärung

Hiermit erkläre ich, dass ich meine Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit noch nicht anderweitig für Prüfungszwecke vorgelegt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Mittweida, 13.03.2020