HF-ICT

Diplomarbeit

Kassenzettelverwaltung mit iOS-App

Timo Dörflinger

19. Juni 2018

Inhaltsverzeichnis

1 Einleitung

In meiner Diplomarbeit erstelle ich eine App zur Kassenzettel-Verwaltung. Für meinen Auftraggeber und mich persönlich sind die Kassenzettel verlorene Daten, ausser man verwendet Sie zur persönlichen Analyse bzw. zum Führen eines Haushaltsbuchs. Daher erstelle ich in meiner Diplomarbeit eine iOS-App, mit welcher ein Kassenzettel fotografiert werden kann. Aus diesem erstellten Abbild werden dann gewisse Werte bzw. Daten mit Hilfe eines bereits bestehenden OCR-Frameworks ausgelesen und zur weiteren Verarbeitung zwischengespeichert. Hier kann also unter anderem der Gesamtbetrag ausgelesen und mit einem gesetzten monatlichen Budget verrechnet werden. Dies soll dann auch kategorisiert werden können. Daraus ist ersichtlich, was in dem laufenden Monat bereits in den verschiedenen Kategorien ausgegeben wurde. Die Kategorien können jeweils individuell erstellt werden.

2 GitHub-Repository

3 Projektmanagement - Scrum

Die Diplomarbeit ist ein wichtiges Projekt mit einem geringen zeitlichen Rahmen. Um in diesem eng gesetzten Rahmen die Aufgaben und deren Verteilung im Überblick zu haben, habe ich mich entschieden die Diplomarbeit mit einer Projektmanagement-Methode zu führen. Da ich bereits in einem zuvor geführten Projekt mit HERMES5 schlechte Erfahrungen in einem Software-Engineering-Projekt gemacht hatte, habe ich mich entschieden, die Diplomarbeit mittels Scrum zu führen.

Da ich, wie bereits erwähnt, einen engen Zeitrahmen von 60 Stunden für die Diplomarbeit habe, wird ein Scrum-Sprint das komplette Projekt abdecken. Um die Diplomarbeit bestmöglich nach Scrum führen zu können, habe ich diverse unterstützende Programme herausgesucht. Bei der Evaluierung bin ich auf zwei interessante Programme gestossen.

Nach der ersten Suche hatte ich iceScrum `https://www.icescrum.com` entdeckt. Es sah einfach aus und war für ein Projekt kostenlos. Das Management findet in der Cloud des Anbieter statt. Bei dem ersten Test nach der Anmeldung musste ich aber feststellen, das manche Seiten lediglich auf Französisch zur Verfügung stehen. Da ich aber so gut wie keine Französischkenntnisse habe, empfand ich das als nicht sonderlich hilfreich und dachte es ist besser nach einer Alternative zu suchen.

Nach einer weiteren Suche bin ich dann auf OpenProject gestossen. OpenProject ist eine *Open source project management software* mit Scrum integriert. OpenProject bietet eine Community-Version an, die kostenlos als auch lokal verwendet werden kann und auch die Darstellung des Programs auf ihrer Homepage hat mich angesprochen. Daher habe ich es getestet und empfinde es als die richtige Lösung für mich und die Diplomarbeit.

Ich habe das OpenProject in meiner lokalen Docker-Umgebung auf dem MacBook installiert. Dafür bin ich der Anleitung von OpenProject gefolgt und bin nach diese Anleitung prompt in ein Problem gelaufen, dass mich zeitlich etwas aufgehalten hatte. Der Anleitung zu folge, sollte für eine produktive Nutzung eine erweiterte Installation gemacht werden. So kann gesichert werden, dass bei einem Neustart des Container keine Daten verloren gehen und auch die Log-Dateien lokal in einer selbst gesetzten Ordnerstruktur auf dem System zu finden ist.

Dafür habe ich einen Ordner in dem GitHub-Repository erstellt, in dem dann diese Daten und Logs gesichert werden sollen. Dafür habe ich den angegebenen Konsole-Befehl für Docker, mit der angepassten Ordnerstruktur,

ausgeführt.

```
1 docker run -d -p 8080:80 --name diplomopenproject -e SECRET_KEY_BASE=secr
2   -v /Users/Timo/diplom-kassenzettelverwaltung/projektmanagement/openproj
3   -v /Users/Timo/diplom-kassenzettelverwaltung/projektmanagement/openproj
4   -v /Users/Timo/diplom-kassenzettelverwaltung/projektmanagement/openproj
5   openproject/community:7
```

Dieser ist auch erfolgreich durchgelaufen und hat den Container erfolgreich erstellt. Die Webseite konnte ich aber dann nicht im Localhost http://localhost:8080 aufrufen. Nach langem suchen und versuchen habe ich dann den zuvor aufgeführten Befehl wie folgt angepasst:

```
1 docker run -it -p 8080:80 --name diplomopenproject -e SECRET_KEY_BASE=sec
2   -v /Users/Timo/diplom-kassenzettelverwaltung/projektmanagement/openproj
3   -v /Users/Timo/diplom-kassenzettelverwaltung/projektmanagement/openproj
4   -v /Users/Timo/diplom-kassenzettelverwaltung/projektmanagement/openproj
5   openproject/community:7
```

Mit diesem Befehl konnte ich den Container erfolgreich erstellen und dann auch im Browser darstellen. Danach konnte ich den Container mit dem Befehl *docker stop diplomopenproject* den Container stoppen und mit *docker start diplomopenproject* den Container wieder starten und es wieder im Browser darstellen.

Nach der Installation ist bereits ein Administrator-Account erstellt, der für den ersten Login verwendet werden kann. Die Zugangsdaten stehen in der Installations-Anleitung. Nachdem Login mit dem Admin muss zuerst einmal ein neues Projekt erstellt werden. In diesem Projekt können nun *Work packages*, also Arbeitspakete definiert werden, welchen nur einzelne Aufgaben oder ganze Meilensteine zugewiesen werden können. Da diese Einstellungen bzw. Erstellungen der Arbeitspakete als Projekt-Administrator mit dem bereits installierten Administrator definiert werden, habe ich noch einen weiteren Benutzer mit meinem Namen hinzugefügt. Diesem Benutzer werden dann die Arbeitspakete und deren enthaltener Aufgaben und Meilensteine zugewiesen.

4 Evaluierung des Datenbank-Frameworks

Im zuvor durchgeführten Projekt hatte ich bereits die Vorstellung, zwischen dem ausgelesenen Text aus den Kassenzettel-Abbildern und der Benutzeroberfläche, eine Datenbank bereitzustellen. In diese Datenbank können dann

die ausgelesenen Endbeträge der Kassenzettel, mit dem jeweiligen Erfass-Datum und noch weiteren Daten zwischengespeichert werden. So hatte ich mich bereits während dem ersten Projekts bezüglich einer lokalen Datenbank für eine iOS-App schlau gemacht. Hier sind mir nach diversen Berichten *SQ-Lite* und *Apple Core Date* im Gedächtnis geblieben. Bei der aktuellen Evaluierung bin ich dann noch auf eine weitere Möglichkeit gestossen, die Daten lokal in der App abspeichern zu können. Diese Option heisst *Realm* und ist ein externes Framework für einen Datenbank-Aufbau in einer App, mit einer steigenden Begeisterung bei Programmierern.

Da leider nicht genug Zeit in der Diplomarbeit übrig war, konnte ich diese drei Optionen selbst nicht mit einem Testaufbau testen. Die nachfolgenden Vergleiche beziehen sich daher auf verschiedene Berichte, die diese drei Optionen verglichen haben.

4.1 SQLite

SQLite ist die schlankere Form von MySQL und damit ein vollumfängliches Datenbank-Framework für iOS das auf Tabellen und Beziehungen (Relations) aufbaut, wie es wir es im Datenbank-Unterricht gelernt haben. SQLite kann mit einer grossen Menge Daten umgehen und die Daten können mit einfachen SQL-Querys (Abfragen) eingefügt oder aufgerufen werden. Jedoch sind laut den verschiedenen Internet-Beiträgen haben diese Querys als iOS-Framework keine hohe Performance und ist wohl recht aufwändig in der Implementierung in die iOS-App.

4.2 Apple Core Data

Apple Core Data ist der von Apple entwickelte Datenbank-Aufbau. Wobei Core Data anders als die üblichen Datenbank-Aufbauten gehandhabt wird. Core Data selbst ist keine Datenbank. Es baut auf SQLite auf und verwaltet die Daten als Objekte. Das macht die Verarbeitung von Daten recht schnell. Allerdings ist der Aufbau von Core Data recht kompliziert und verlangt eine grosse zeitliche Einarbeitung.

4.3 Realm

Realm ist ein noch relativ junges Framework und verarbeitet die Daten, ähnlich wie Core Data, in Objekten. Es ist aber ein relativ schlankes Framework, welches eine sehr hohe Performance bietet. Auch die Verwendung von Realm kann mit weniger Code-Zeilen verwendet werden, als mit Core

Data. Was ich als einen weiteren Vorteil sehe, ist dass Realm auch Plattformübergreifend verwendet werden kann. Soll die App also später in einer produktiven Umgebung auf das Android-System migriert werden, können aus dem Realm-Aufbau der iOS-App einfach die Datenbank-Dateien in einen Android-Realm-Aufbau übernommen werden.

Aus den oben beschriebenen Vergleichen werde ich die Datenbank für dieses Projekt in Realm umsetzen.

5 Quellenverzeichnis

Anleitung für OpenProject in Docker `https://www.openproject.org/docker/`
Video-Anleitung für Scrum Nutzung in `https://www.openproject.org/collaboration-software-features/scrum-agile-project-management/`

6 Bilderverzeichnis