Bachelor Thesis

Spektrometer App

Anbindung Spektrometer an mobiles Device

Andreas Lüscher, Raphael Bolliger

Fachhochschule Nordwestschweiz
Hochschule für Technik

Dozent: Martin Gwerder
Auftraggeber: Andreas Hueni

Windisch, 6. März 2017

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Ziel der Arbeit . . . 1
1.2 Hilfestellungen . . . 1
1.3 Erreichtes . . . 1
1.4 Leserführung . . . 1

2 Theoretischer Teil **2**
2.1 Einleitung . . . 2
2.2 Messverfahren . . . 2
2.3 Berechnungsablauf . . . 2
2.4 Berechnungen . . . 2
2.4.1 Dark Current . . . 2
2.4.2 White Reference . . . 2
2.4.3 Radiance . . . 2

3 Umsetzung **3**
3.1 Einleitung . . . 3
3.2 Anforderungen . . . 3
3.3 Technologien . . . 3
3.3.1 Entwicklungsumgebung . . . 3
3.3.2 Programmiersprache . . . 3
3.3.3 Abhängigkeiten . . . 3
3.4 Architektur . . . 3
3.4.1 Grobarchitektur . . . 3
3.4.2 Core . . . 4
3.4.3 Service . . . 4
3.4.4 iOS App . . . 4
3.5 Core . . . 4
3.5.1 Connection . . . 4
3.5.2 Input/Output . . . 4
3.5.3 Calculations . . . 4
3.5.4 Error Handling . . . 4
3.6 Service . . . 4
3.6.1 Instrument Store . . . 4
3.6.2 Command Manager . . . 4
3.6.3 File Write Manager . . . 4
3.7 iOS App . . . 4
3.7.1 App Delegate . . . 4
3.7.2 Core Data . . . 4
3.7.3 Views . . . 4
3.7.4 Controllers . . . 5
3.7.5 Components . . . 5

3.7.6 View Store 5
3.7.7 Service 5

Abbildungsverzeichnis

Zusammenfassung

Das Hauptziel des Projektes, ist es eine mobile Applikation zu erstellen, welche die RS3 Desktoplösung von ASD ablöst. RS3 ist eine Software, welche die Verbindung zu einem Spektralmessgerät herstellt, Messungen auslösen sowie die Resultate anzeigen kann. Das bisherige System besteht aus einem Laptop inklusive RS3 Software, welche jeweils auf ein Spektrometer abgestimmt ist. Neu soll eine mobile App ausreichen, um mehrere Spektrometer ansprechen zu können. Die App soll Forschende unterstützen, Messungen direkt vor Ort zu beurteilen und verwalten zu können.

1 Einleitung

1.1 Ziel der Arbeit

Das geologische Institut der Universität Zürich betreibt zur Forschung vier Spektroradiometer der Firma ASD Inc. aus Colorado in den USA. Zu jedem Spektrometer liefert ASD einen Notebook mit installierter Software um das Spektrometer zu bedienen und Messungen auszuführen. Ziel dieser Arbeit ist es die Software RS3 von ASD mit einer modernen Applikation für ein mobiles Device abzulösen. Das Projektteam hat sich gemeinsam mit dem Kunden dazu entschieden die Applikation für iOS Geräte zu entwickeln.

1.2 Hilfestellungen

Zur Umsetzung konnten verschiedenen Hilfestellungen in Anspruch genommen werden. ASD bietet auf der ihrer Webseite einen Download mit einem Developer Kit an. In dieser Dokumentation ist beschrieben wie interessierte Entwickler mittels eines TCP Servers der auf dem Spektrometer betrieben wird, selbst Applikationen entwickeln können. Die Dokumentation enthält ausführliche Informationen zu Verbindungsparameter, Rückgabetypen und Dateiformaten.

Weiter konnten wir auf das GitHub Repository der SPECCHIO Datenbank zurückgreifen. In dieser Applikation wurde verschiedenste Berechnungen und Manipulationen mit Spektralfiles bereits in Java ausprogrammiert.

1.3 Erreichtes

Die Applikation wurde mit den definierten Grundanforderungen vollständig umgesetzt. Der Benutzer kann, sofern das iPad mit dem Spektrometer über WLAN verbunden ist, das Spektrometer bedienen und Messungen ausführen. Es wurde speziell darauf geachtet die Messungen einfacher und für den Benutzer intuitiver zu gestalten.

1.4 Leserführung

2 Theoretischer Teil

2.1 Einleitung

2.2 Messverfahren

2.3 Berechnungsablauf

2.4 Berechnungen

2.4.1 Dark Current

2.4.2 White Reference

2.4.3 Radiance

3 Umsetzung

Dieses Kapitel beschreibt die Umsetzung des Produktes. Die Nachfolgenden Abschnitte werden Schrittweise detaillierter und beschreiben den Aufbau sowie die Designentscheidungen des Softwarecodes.

3.1 Einleitung

3.2 Anforderungen

3.3 Technologien

3.3.1 Entwicklungsumgebung

3.3.2 Programmiersprache

Das Projekt wurde in der Programmiersprache Swift 3 umgesetzt und ist mit iOS 10 und höher kompatibel. Es wurde darauf geachtet, dass alle Abhängigkeiten ebenfalls in Swift umgesetzt sind. Dies erleichtert die Weiterentwicklung da kein, für ungeübte Entwickler, meist schwer lesbarer Objective C Code zum Einsatz kommt.

3.3.3 Abhängigkeiten

3.4 Architektur

3.4.1 Grobarchitektur

Um weite Teile des Sourcecodes erneut nutzen zu können, wurde das Projekt in zwei Teile aufgeteilt. Der Core beinhaltet den gesamten Code, welcher Systemunabhängig ist.

Der restliche Code befindet sich im Ordner Spektrometer. Dieser ist iOS spezifisch und kann nicht einfach auf andere Plattformen portiert werden.

3.4.2 Core

3.4.3 Service

3.4.4 iOS App

3.5 Core

3.5.1 Connection

3.5.2 Input/Output

File Writer

File Reader

Spectrometer Parser

3.5.3 Calculations

3.5.4 Error Handling

3.6 Service

3.6.1 Instrument Store

3.6.2 Command Manager

3.6.3 File Write Manager

3.7 iOS App

3.7.1 App Delegate

3.7.2 Core Data

3.7.3 Views

Alle ViewController sind in der Datei Main.storyboard enthalten. Dies war eine bewusste Entscheidung, um Entwicklern einen guten Überblick über den gesamten UI Ablauf zu ermöglichen. Einzig das Design für eine Zelle der Messübersichtstabelle wurde in eine eigne XIB-Datei ausgelagert.

Die Anordnung der Controls wurde im Storyboard gelöst. Kleinere Merkmale wurden jeweils im Code angepasst, indem von bestehenden Controls abgeleitet wurde.

3.7.4 Controllers

Settings

Einstellungen, welche pro Applikation verfügbar sein müssen, werden in den sogenannten UserDefaults gespeichert. Dieser Speicher, kann sämtliche serialisierbaren Objekte speichern.

3.7.5 Components

3.7.6 View Store

3.7.7 Service

Validation

Für die Validierung wurde von jedem benutzen Control eine Ableitung erstellt und und das BaseValidationControl Protokoll implementiert. Dieses Protokoll enthält ein Property isValid welches den Gültikeitszustand des Objektes enthält.

In jedem ViewController, muss nun nur noch der ValidationManager aufgerufen werden und die Hauptview übergeben werden. Dieser ValidationManager prüft nun alle Subviews, welchedas Protokoll implementieren.

Um die Validation für einen neuen ViewController hinzuzufügen, kann folgendermassen vorgangen werden: 1. Erstellen Sie einen neuen ViewController 2. Fügen Sie ein Control hinzu und leiten sie von einem bestehenden ValidationControl ab. Oder erstellen Sie eine neue Ableitung eines Controls und implementieren Sie das BaseValidationControl Protokoll. 3. Rufen Sie die validateSubViews Methode des ValidationMangers auf.

File Browser