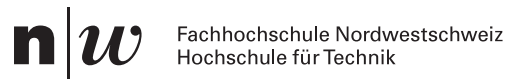


Bachelor Thesis

Spektrometer App

Anbindung Spektrometer an mobiles Device

Andreas Lüscher, Raphael Bolliger



Dozent: Martin Gwerder
Auftraggeber: Andreas Hueni

Windisch, 22. Februar 2017

Inhaltsverzeichnis

1	Ausgangslage und Problemstellung	1
1.1	Ausgangslage	1
1.2	Problemstellung	1
2	Umsetzung	2
2.1	Voraussetzungen	2
2.2	Architektur	2
2.3	UI	2
2.4	Validierung	2
2.5	Einstellungen	3
2.6	Gespeicherte Daten	3
2.7	Indico Format	3
2.8	Raw	3
2.9	Reflectance	3
2.10	Radiance	3

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 Ausgangslage und Problemstellung

1.1 Ausgangslage

Das ist die Ausgangslage Das geologische Institut der Universität Zürich

1.2 Problemstellung

Das ist die Problemstellung

2 Umsetzung

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

2.1 Voraussetzungen

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.

2.2 Architektur

Um weite Teile des Quellcodes erneut nutzen zu können, wurde das Projekt in 2 Teile aufgeteilt. Der Core-Teil 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.

2.3 UI

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 eigene XIB-Datei ausgelagert.

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

2.4 Validierung

Für die Validierung wurde von jedem benutzten Control eine Ableitung erstellt und das BaseValidationControl Protokoll implementiert. Dieses Protokoll enthält ein Property isValid welches den Gültigkeitszustand 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, welche das Protokoll implementieren.

Um die Validation für einen neuen ViewController hinzuzufügen, kann folgendermaßen vorgegangen 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 ValidationManagers auf.

2.5 Einstellungen

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

2.6 Gespeicherte Daten

Alle Dateien, welche auf dem Gerät gespeichert sind,

2.7 Indico Format

Um Messungen speichern zu können, bietet die Firma ASD ein Format namens Indico File Format an. Diese Projektarbeit stützt sich ebenfalls auf dieses Format.

2.8 Raw

Bei sogenannten Raw Messungen, werden die Dateien welche vom Spektrometer gesendet werden, ohne weitere Manipulationen gespeichert.

2.9 Reflectance

2.10 Radiance