

# Raspberry Poolüberwachung

## DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von  
Sebastian Egger  
Florian Wilflingseder

Betreuer:  
Michael Wagner  
Gerald Köck

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

S. Egger & F. Wilflingseder

Zur Verbesserung der Lesbarkeit wurde in diesem Dokument auf eine geschlechtsneutrale Ausdrucksweise verzichtet. Alle verwendeten Formulierungen richten sich jedoch an alle Geschlechter.

# Zusammenfassung

Zusammenfassung unserer genialen Arbeit. Auf Deutsch. Das ist das einzige Mal, dass eine Grafik in den Textfluss eingebunden wird. Die gewählte Grafik soll irgendwie eure Arbeit repräsentieren. Das ist ungewöhnlich für eine wissenschaftliche Arbeit aber eine Anforderung der Obrigkeit. *Bitte auf keinen Fall mit der Zusammenfassung verwechseln, die den Abschluss der Arbeit bildet!*



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Umfeldanalyse</b>	<b>2</b>
<b>3</b>	<b>Technologien</b>	<b>3</b>
3.1	Raspberry Pi . . . . .	3
3.2	MQTT . . . . .	3
3.3	.Net . . . . .	3
<b>4</b>	<b>Zusammenfassung</b>	<b>5</b>
	<b>Literaturverzeichnis</b>	<b>IV</b>
	<b>Abbildungsverzeichnis</b>	<b>V</b>
	<b>Tabellenverzeichnis</b>	<b>VI</b>
	<b>Quellcodeverzeichnis</b>	<b>VII</b>
	<b>Anhang</b>	<b>VIII</b>

# 1 Einleitung

Unsere Diplomarbeit wurde in das Leben gerufen um einen Pool überwachen zu können. Aktuell sind mehrere Geräte für die Überwachung eines Pools erforderlich. Ziel unserer Diplomarbeit ist es diese Funktionalitäten zuverlässig in einem Gerät kosteneffizient zusammenzuführen und über ein UI der SP-Anwendung wird ein 360 Grad Blick auf die Geschehnisse im Pool ermöglicht.

## **2 Umfeldanalyse**

## 3 Technologien

### 3.1 Raspberry Pi

### 3.2 MQTT

### 3.3 .Net

Projekte

Das Backend setzt sich aus 12 Projekten zusammen, welche in C# .Net 5 geschrieben sind.

- `CommonBase`

In `CommonBase` befinden Klassen und Methoden, die wiederverwendbar sind um Code verdoppelung zu vermeiden.

- `CSharpCodeGenerator.Logic`

`CSharpCodeGenerator.Logic` dient zum automatischen generieren von den Entitäten in `SnQPoolIot.Logic`, `SnQPoolIot.WebApi` und `SnQPoolIot.AspMvc`. In `SnQPoolIot.Contracts` werden Entitäten als Interfaces angegeben. Wenn danach die Solution gebuildet wird, werden für die Entitäten Klassen und die dazugehörigen Controller angelegt. Es werden auch Beziehung und Keys zwischen den Entitäten generiert.

- `SnQPoolIot.Adapters`

`SnQPoolIot.Adapters` bietet uns einen direkten Zugriff auf die Logic. Der Zugriff auf die Logic kann dadurch entweder direkt erfolgen oder per Rest über die WebApi.

- `SnQPoolIot.WebApi`

Der Zugriff auf die Messwerte wird durch Rest-Zugriffe in `SnQPoolIot.WebApi` providet. Auf die Daten kann aber nur zugegriffen werden, wenn man sicher vorher mit einem Account einloggt.

- SnQPoolIot.Contracts

SnQPoolIot.Contracts beinhaltet alle notwendigen Schnittstellen und Enumerationen des Projektes. Hier werden die Entitäten als Interfaces angelegt, welche von CSharp-CodeGenerator.Logic als Klassen und Controller in den oben genannten Projekten generieren.

- SnQPoolIot.Logic

SnQPoolIot.Logic ist das Kernstück des Projektes. Durch die Logic können wir alle Daten von der Datenbank holen. Die Datenbank auf welche die Logik zugreift ist ein SqlServer und der Zugriff und das Erzeugen der Datenbank wird mittels Entityframework

- SnQPoolIot.Transfer

SnQPoolIot.Transfer verwaltet die Transferobjekte für den Datenaustausch zwischen den Layers.

- SnQPoolIot.AspMvc

SnQPoolIot.AspMvc ist ein Ersatz für das Frontend. Hier werden die Funktionen wie zum Beispiel einloggen eines Users oder anzeigen von Messwerten dargestellt.

- SnQPoolIot.ConApp

In SnQPoolIot.ConApp werden User mit verschiedenen Rechten angelegt, die für die Authentifizierung benötigt werden.

.Net 5 vs .Net 6 Unser Backend wurde wie oben bereits erwähnt mit .Net 5 geschrieben, weil die Umstellung zu .Net 6 einiges mit sich bringt. Der Hauptgrund warum wir uns für .Net 5 entschieden haben, ist die Unterscheidung der Nullability von Objekten und die dadurch ausgelösten Warnings. In .Net 5 werden durch Objekte die Null sein könnten keine Warnings angezeigt und man erspart sich etliche Zeit, wenn man sich nicht durch die Warnings käpfen muss. Jedoch im Vergleich zu .Net 5 hat .Net 6 eine bessere Leistung, denn bei einer Ausführung in der Cloud sinkt es die Computekosten, welches ich als riesigen Vorteil sehe. Einen weiteren Nachteil finde ich ist die Lesbarkeit des Code. Hier ein Beispiel zu diesem Thema:

[Foto von ConApp .Net 5] [Foto von ConApp in .Net 6]



## **4 Zusammenfassung**



# Literaturverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Quellcodeverzeichnis

# Anhang