



# Projektbericht:

## *FLOATING CITY*

Gruppe: Informatik 4bINFT - INF | PP | 2020

# Inhalt

|                       |   |
|-----------------------|---|
| 1. Einleitung.....    | 3 |
| 2. Projekt.....       | 4 |
| Projektziel.....      | 4 |
| Projektumfeld.....    | 4 |
| Vorgehen.....         | 4 |
| Probleme.....         | 4 |
| 3. Entwurf.....       | 5 |
| Modelaufbau.....      | 5 |
| Datenbankdesign.....  | 5 |
| Hardware.....         | 6 |
| Simulationsdaten..... | 6 |
| Settings.....         | 7 |
| 4. Fazit.....         | 7 |

## 1. Einleitung

Dieser Projektbericht dient als Ergänzung unserer Onlinedokumentation des Projekts im Rahmen unser Lehrausbildung.

Das Thema ist in aller Munde. Klima, Klimaschutz und Energiegewinnung. An der Universität Innsbruck wurde im Zuge dessen das Konzept für eine „schwimmende Stadt“ ausgearbeitet. Im Prinzip handelt es sich dabei um ein Stadt welche, im besten Fall, energieautonom ist.

Es handelt sich um eine Stadt die auf „Luftkissen“ gebaut wird in welchen während des Tages Wasser, durch Einpumpen, gespeichert werden kann. Sollte es am Abend/in der Nacht zu Verbrauchsspitzen kommen, kann Wasser aus diesen Tanks wieder abgelassen werden um somit Generatoren anzutreiben und Strom zu gewinnen.

## 2. Projekt

### PROJEKTZIEL

Um erste Simulationen zu diesem Projekt fertigzustellen, wurden wir damit beauftragt das ganze Szenario anhand eines Modells durchzuspielen.

Dazu sind wir aufgrund der momentanen Coronakrise allerdings nicht mehr gekommen. Somit haben wir Simulationsdaten etc. generieren müssen, was unseren Arbeitsablauf ein wenig umgestaltet hat.

### PROJEKTUMFELD

Das Projekt fand im Rahmen unserer Lehrausbildung statt. Da wir nun kurz vor unserem Abschluss stehen und bald zur Lehrabschlussprüfung antreten werden, hat uns dieses Projekt in vielen Bereichen gefördert und gefordert, was für uns eine gute Vorbereitung für die LAP bedeutete.

### VORGEHEN

Nach einem ersten Brainstorming konnten wir die einzelnen Interessensgebiete und Schwerpunkte unserer Gruppenmitglieder absprechen. Somit konnten wir dann auch bei der Aufgabenverteilung gut managen, da wir wussten wer welche Stärken vorzuweisen hatte.

Über Azure Devops regelten wir die Einteilung von Workitems , sowie unser Projektmanagement. Damit wurde gewährleistet, dass jedes Mitglied unserer Gruppe zu jeder Zeit auf dem selben Stand ist.

### PROBLEME

Wir wurden vor verschiedenste Problem gestellt. Was uns aber wahrscheinlich am meisten Zeit gekostet hat war ein Merge-Konflikt, da wir alle noch relativ unerfahren hinsichtlich Azure-Devops gewesen sind. Doch auch hier konnten wir einiges an Erfahrung sammeln.

### 3. Entwurf

Wir haben uns für ein ASP.NET MVC – Anwendung entschieden, da wir Personen mit guten HTML- sowie Bootstrap und AJAX-Kenntnissen in unserer Gruppe haben, was uns bei der Gestaltung der Grafiken für den Enduser enorm weitergeholfen hat. Bei der Datenbankgenerierung entschieden wir uns für Code-First. Bei der Datenbank selbst handelt es sich um eine SQL-Express DB. Es war uns ebenfalls wichtig, sämtlichen Funktionen auch für Mobilgeräte anschaulich zu gestalten.

#### MODELAUFBAU

Prinzipiell haben wir uns für einen relativ einfachen Modelaufbau entschieden, da dieser, aus unserer Sicht, ausreichend war. Dies spiegelte sich später auch in verschiedenen Tests unserer Anwendung wider.

Es wurde ebenfalls auf eine Userhierarchie geachtet um etwaige Funktionen bestimmten Usern vorzuenthalten bzw. zu ermöglichen.

#### DATENBANKDESIGN

An dieser Stelle haben wir uns für eine table-per-class-inheritance Strategie entschieden. Das bedeutet, dass für jede Klasse in der Vererbungskette eine eigene Tabelle angelegt wird. Diese sind über einen eindeutigen Primary-Key zuweisbar und per Foreign-Key werden Verbindungen zwischen den Klassen hergestellt.

Auch die verschiedenen Beziehungstypen(one-to-one, one-to-many und many-to-many) mussten bei diesem Projekt berücksichtigt werden. Die Beziehungen werden im Code untereinander mit Listen bzw. Objekten dargestellt.

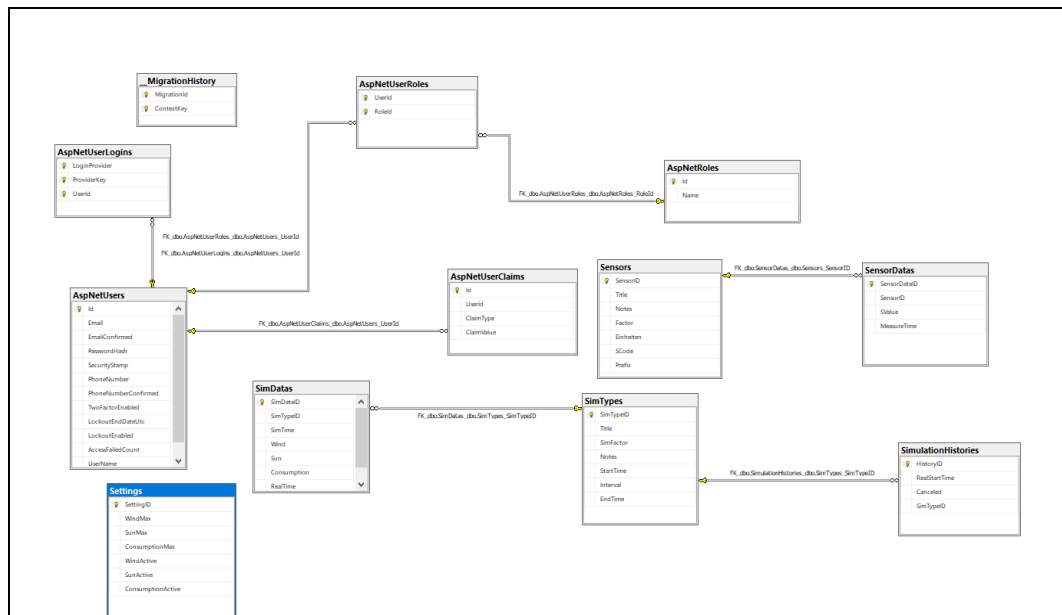
Hier der daraus resultierende (gekürzte)Code für die Tabelle „Sensor“:

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
public int SensorID { get; set; }

[Display(Name = "Sensoren")]
public string Title { get; set; }

//one-to-many Verbindung zw. Sensor und SensorData
public virtual ICollection<SensorData> SensorDatas { get; set; }
```

Abbildung unseres genauen Datenbankaufbaus, inklusive Standard-ASP.NET User-Management:



## HARDWARE

Für die Übertragung der Daten der Controller in unserem Maßstabsmodell der Stadt, haben wir uns für ein Raspberry Pi 4 entschieden. Die Schnittstelle von Raspberry Pi zu unserem Webserver wurde im Zuge dessen von zwei unserer Gruppenmitglieder programmiert. Wir entschieden uns für eine Serialisierung der Daten per JSON.

Ebenfalls sollten bei dem beendeten Projekt viele Controller in dem Maßstabsmodell vorhanden sein. Diese Aufgabe wurde von unseren Elektroniker-Kollegen übernommen. Aufgrund der momentanen Umstände war es uns aber leider nicht möglich mit diesem Modell zu arbeiten.

## SIMULATIONS DATEN

Ursprünglich war es geplant, die Daten direkt von den Controllern zu bekommen. Jedoch brauchten wir vorerst Simulationsdaten die wir selbst generieren konnten. Dies wurde per Konsolen-Applikation gelöst. Für erste Tests haben wir uns ebenfalls per VM einen Webserver eingerichtet.

Ein großer Punkt war die zeitliche Manipulation der Simulationen, damit diese beispielsweise in 10-facher Geschwindigkeit durchgeführt werden könnten.

Einblicke in die Daten der bisherigen Simulationen sind ebenso möglich wie die Erstellung von Diagrammen für Live-Abbildungen der momentan laufenden Simulation.

## SETTINGS

Für die Zukunft sind ebenfalls noch einige Benutzereinstellungen geplant. Vorerst ist es für die Benutzer jedoch möglich Faktoren und Werte für die jeweiligen Simulationen zu setzen.

Weitere Settings sollen die Benutzerfreundlichkeit in Zukunft noch verbessern. z.B.: Das Einfügen von Default-Werten etc...

## 4. Fazit

Vorab ist gleich zu sagen, dass das Projekt an dem Maßstabsmodell leider nie getestet werden konnte aufgrund der momentanen Corona-Krise. Aus diesem Grund haben wir uns viel mit dem Thema der Datenerzeugung auseinandergesetzt. Auch auf den Gebieten , JS, AJAX oder Bootstrap konnten wir unseren Horizont sehr erweitern.

Ebenfalls wurde viel an Datenbankdesigns und Verbesserungen vorgenommen, was uns zwar einiges an Zeit gekostet hat, die wir uns dadurch in Zukunft aber sparen werden.

Besonders angenehm war auch die Zusammenarbeit in der Gruppe, vor allem wenn es darum ging Wissenslücken bei anderen zu Füllen, war immer jemand zu erreichen um das Problem gemeinsam anzugehen.