

Measurement Library

Webapplikationen zu erstellen sind häufige Aufgaben eines Informatikers. Dies liegt vor allem daran, dass man recht einfach beginnen kann und dann die Applikation mit immer mehr Features ausstatten kann. So steigert sich der Schwierigkeitsgrad immer ein bisschen mehr. Um eine Webapplikation zu erstellen werden verschiedenste Bereiche benötigt (Design erstellen, Seiteninhalte grafisch darstellen, programmieren, Datenbank erstellen, ...). Das macht es zu einer idealen Aufgaben für einen Lernenden.

Projektauftrag

Der Auftrag war die Entwicklung einer Webapplikation, welche für die Verwaltung von elektronischen Messgeräten eingesetzt werden kann. Die Anforderung ist, dass man anhand der Applikation sehen kann, was für Messgeräte es in der Firma gibt und wer welches Messgerät ausgeliehen hat. Zudem war gefordert, dass man noch weitere Eigenschaften der Messgeräte speichern kann, wie z.B. der aktuelle Standort, ein Barcode für die eindeutige Kennzeichnung, etc. Die Applikation muss einem Administrator die notwendige Funktionalität zur Verfügung stellen, um neue Messgeräte zu erfassen oder ältere Geräte zu löschen.

Ablauf

Am Anfang ging es darum alle Anforderungen zu sammeln. Daraus habe ich dann eine Liste mit allen benötigten Funktionen erstellt. So konnte ich die Liste später als eine Art Checkliste durchgehen und schauen, welche Funktionen ich noch zu erstellen hatte. Als weiteren Schritt musste ich festlegen mit welcher Web-Technologie ich arbeiten wollte und ein Design für einen Prototypen erstellen.

Ich habe mir anhand der benötigten Funktionen überlegt, welche Seiten die Applikation umfassen muss und auf welchen Seiten die Funktionen platziert werden. So habe ich die Seitenstruktur der Applikation definiert. Nun ging es noch darum die Struktur der Datenbank zu definieren.

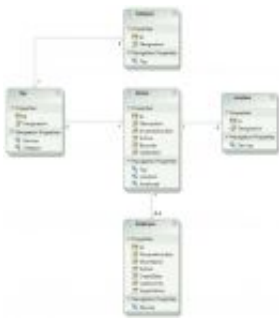
Als Nächstes ging es darum, anhand des Design und der Seitenstruktur, eine erste Version der Webapplikation zu erstellen. So habe ich mich entschieden, als Erstes die Seite zu erstellen auf welcher alle Geräte angezeigt werden. Damit ich jedoch auf der Seite überhaupt etwas anzeigen konnte, musste ich anhand der Datenbank-Struktur noch einen Datenbank erstellen und diese mit Testdaten füllen. Nachdem diese Seite funktioniert hat, entwickelte ich auch noch die anderen Seiten.

Als einer der letzten Schritte war ein ausführlicher Test der gesamten Applikation notwendig. Danach musste ich die erste Version meinem Auftraggeber präsentieren. Anhand des Feedbacks welches ich während dem Testen und nach der Präsentation erhalten hatte, arbeitete ich an einer zweiten Version, welche die neuen Anforderungen und weiteren Wünsche beinhaltet.

Fazit

Ich habe dieses Projekt eine sehr spannende Arbeit gefunden, da es eines der ersten Projekte war, welche ich von A-Z alleine durchführen konnte. Ich fand es spannend und motivierend eine Webapplikation zu entwickeln, welche später wirklich gebraucht wird und nicht nur ein „Lernprojekt“ war.

Juli 2013, Lukas Arnold, Informatiker 2.Lj



```
public class Message {
    // Model vars
    string text = string.Empty;
    MessageType type = MessageType.None;

    // View vars
    // GET or set the text of the message
    // Summary
    public string Text {
        get { return text; }
        set { text = value; }
    }

    // Summary
    // GET or set the type of the message
    // Summary
    public MessageType Type {
        get { return type; }
        set { type = value; }
    }

    // Summary
    // Summary
    // Creates a new Message with the following parameters
    // Summary
    // Updates the text of the message
    // Updates the type of the message
    public Message(string text, MessageType type) {
        Text = text;
        Type = type;
    }
}
```

```
.gridView .pagerStyle table
{
    border: 1px solid #000000;
    border-collapse: separate;
    background-color: #000000;
}

.gridView .pagerStyle a
{
    color: #333333;
    text-decoration: none;
}

.gridView .pagerStyle a:hover
{
    text-decoration: underline;
}

.gridView .editRowStyle
{
    background-color: #000000;
}

.gridView .rowStyle
{
    background-color: Silver;
}

.gridView tr
{
    border: 1px solid #000000;
}
```