

# Programmentwicklung Advanced II

## Modulübergreifende Projektarbeit

### NuGet

**Ausgangslage**

Im Projekt «Datenbank II» wurden u.a. folgende Anforderungen umgesetzt (nachfolgend der entsprechende Auszug aus dem Projektauftrag «Datenbank II»):

*Logging*

Der Kunde hat in einigen PoD eine SNMP-Monitoring-Lösung installiert, welche SNMP-Ereignisse automatisch an die zentrale Geräteverwaltung senden kann. Die SQL-Skripts für die Anbindung an diese Lösung sollen so einfach wie möglich gestaltet werden. Konkret bedeutet dies, dass eine Stored-Procedure *LogMessageAdd* zur Verfügung stehen muss, welche unter Angabe von PoD- und Hostname des Geräts, dem Level und der effektiven Nachricht, einen entsprechenden Logeintrag in der Datenbank erzeugt.

*Schnittstelle zu bestehendem Gerätemanagement-Werkzeug*

Die Firma verfügt über ein beliebtes Monitoring-Tool, das es erlaubt, Log-Einträge für Geräte im System zu visualisieren und ggf. zu quittieren. Das Tool erwartet eine Tabelle mit dem Namen *v\_logentries* und folgender Struktur:

| Attribut  | Datentyp  | Beschreibung   |
|-----------|-----------|--|
| id        | INT       | Eindeutige Identifikation des Logeintrags                  |
| pod       | TEXT      | Name des POD in welchem die Nachricht aufgetreten ist      |
| location  | TEXT      | Name der Location in welcher die Nachricht aufgetreten ist |
| hostname  | TEXT      | Hostname des Geräts dem die Nachricht zugeordnet ist       |
| severity  | INT       | Schwere des Logeintrags (größer = schwerwiegender)         |
| timestamp | TIMESTAMP | Zeitpunkt des Logeintrags                                  |
| message   | TEXT      | Lognachricht   |

Das Tool geht davon aus, dass diese Tabelle lediglich Einträge enthält, welche noch nicht quittiert wurden. Für das Quittieren erwartet das Tool, dass eine Stored Procedure *LogClear(IN Id INT)* zur Verfügung steht, welche die ID der zu quittierenden Logmessage erwartet.

Ein Testtool steht Ihnen zur Verfügung (siehe MonitorClient.zip).

|                       |  |
|-----------------------|--|
| <b>Auftrag</b>        | <p>Es soll ein WPF-Control erstellt werden, welches folgende Möglichkeiten bietet:</p> <ul style="list-style-type: none"><li>• Property für Angabe des ConnectionStrings zur Datenbank</li><li>• Button «Load» für das Laden der Daten. Dieser Button soll folgendes SQL-Statement ausführen:<br/><code>SELECT id, pod, location, hostname, severity, timestamp, message FROM v_logentries</code></li><li>• Daten, die von obiger Abfrage geliefert werden, sollen in einem Grid (z.B. DataGridView) visualisiert werden</li><li>• Es soll ein Datensatz markiert und über den Button «Confirm» bestätigt werden. Dies ruft die Stored Procedure <code>LogClear</code> auf. Anschliessend darf der Datensatz nicht mehr angezeigt werden.</li><li>• Button «Add» der die Erfassung neuer Logmeldungen ermöglicht. Wird dieser Button geklickt, sollen alle nötigen Daten abgefragt und anschliessend die Stored Procedure <code>LogMessageAdd</code> aufgerufen werden.</li></ul> <p>Anschliessend soll das WPF-Control als NuGet-Package auf <a href="http://www.nuget.org">www.nuget.org</a> zur Verfügung gestellt werden.</p> <p>Für die Datenbankanbindung soll ADO.NET (gemäss Modul «Datenbanken II») verwendet werden. Für die Businesslogik soll ein ViewModel für das Control erstellt werden, welches die komplette Logik enthält (d.h. es ist kein Model nötig). Das ViewModel soll auch im NuGet-Package integriert werden.</p> |
| <b>Lieferergebnis</b> | <p>Als Abgabe wird folgendes erwartet:</p> <ul style="list-style-type: none"><li>- PackageID, unter der das Package von nuget.org heruntergeladen und installiert werden kann.</li><li>- SourceCode</li></ul> <p>Am besten notieren Sie die PackageID in einer Textdatei und packen diese zusätzlich ins Zip des Sourcecodes.</p>  |
| <b>Bewertung</b>      | Gemäss Initialdokument   |
| <b>Termin</b>         | Gemäss Moodle-Abgabe   |