

Software Projekt - Excel als SQL Datenbank

Autoren: Rijad Žužo, Séverin Müller

Dozent: Ulrich Hauser

May 23, 2015



Inhaltsverzeichnis

1	Motivation	1
1.1	Ausgangssituation	1
1.2	Lösungsidee	1
2	Anforderungsliste	1
2.1	Muss-Anforderungen	1
2.2	Soll-Anforderung	2
2.3	Wunsch-Anforderung	2
3	Projektumgebung	3
3.1	Entwicklungsprozess	3
3.2	Programmiersprache	3
3.3	Verwaltungssystem	3
4	Projekt Planung	4
4.1	Product Backlog	4
4.2	Sprints	4
4.3	Sprint Stand-up Meetings	4
4.4	Sprint Review	4
5	Ablauf der Entwicklung	5
5.1	Modellierung	5
5.1.1	Klassen Diagramm	5
5.1.2	Aktivitäts-Diagramm	5
5.1.3	Sequenz Diagramm	5
5.2	Stunden Journal	5
5.3	GUI	5
5.4	Database Handling	5
5.5	Backup Management	6
5.6	Testen	6
6	Ziel	7
6.1	Resultat	7

1 Motivation

Das Projekt wurde von uns in der Freizeit für den Bosnischen Club St. Gallen erarbeitet. Diese führen seit langem eine Excel-Liste für die Mitgliederverwaltung.

Für Sie war dies das einfachste Werkzeug, jedoch gab es immer wieder Probleme, wie zum Beispiel das ungewollte löschen ganzer Zeilen, oder das verrutschen in den Zeilen / Spalten. Wir wollten jedoch nicht eine komplett neue Datenbank erarbeiten, und entschieden uns, Ihnen ein Werkzeug für die Verwaltung der vorhandenen Excel Tabelle zur Verfügung zu stellen, welches eine einfache, intuitive GUI zur Verfügung stellt.

1.1 Ausgangssituation

Nebst den Problemen bei unaufmerksamer Bearbeitung, ist das Auslesen eines grösseren Excel Files deutlich langsamer als bei einer SQL Datenbank der gleichen bzw. einer vielfachen Grösse, deshalb ist es sinnvoll das Excel File in eine SQL Tabelle umzuwandeln und so zu Verarbeiten.

Die Konversion würde zusätzliche Software Kenntnisse erfordern, so griffen wir auf Vorhandene Office Werkzeuge zurück.

Mit unserer x2dB Applikation wurde das Problem gelöst. Die Umwandlung ist dank ODBC Connector unnötig und somit kann das Excel File bestehend bleiben.

1.2 Lösungsidee

Wir legten vor allem Wert auf ein simples User-Interface mit allen nötigen Funktionen. Die Software verarbeitet das Excel File im Hintergrund als SQL Tabelle und verbindet sich mit entsprechenden Treibern über ein File das via GUI (Dateibrowser) eingebunden werden kann. Das ermöglichen uns die Microsoft.Office.Core und Microsoft.Office.Excel.Interop Treiber.

Mit unserer Applikation hat der User einen begrenzten Einfluss auf das File und kann so weniger Schaden am File anrichten. Schaden können auch gleichzeitige Lese- / Speicherzugriffe auf ein Shared File auf einem Netzwerklaufwerk. In unserer Applikation dauert die Verbindung nur kurz, bis die Daten gelesen/geschrieben wurden und danach wird das File wieder freigegeben.

Für zusätzliche Datenintegrität wird jeweils ein Backup des Files erstellt und kann nötigen Falls zurück gespielt werden. Dies ist nicht die 'ultimative Lösung', aber wir konnten so die Sicherheit bei Veränderungen am Excel File erhöhen und trotzdem ein 'für jeden lesbares' Dateiformat weiter verwenden; dass zum Beispiel auch mit einem USB Stick übertragen und auf heimischen Computern (weiter-)bearbeitet werden kann.

2 Anforderungsliste

Um die Bedürfnisse der Verwalter dieser Excel Liste bestmöglich abdecken zu können, haben wir uns mit Ihnen zusammen gesetzt und die nachfolgenden Anforderungen definiert.

2.1 Muss-Anforderungen

Die Applikation muss:

M1: Das vorhandene Excel File als Datenbasis verwenden.

M2: Das Excel File lesen und beschreiben können.

M3: Nach dem Lesen bzw. Schreiben die Verbindung trennen.

M4: In regelmässigen Abständen Backups erstellen.

2.2 Soll-Anforderung

Die Applikation soll:

S1: Leicht Bedienbar sein.

S2: Ein intuitives, einfaches User-Interface haben.

S3: Dem User das Handling des Excel Files abnehmen.

2.3 Wunsch-Anforderung

Die Applikation könnte:

W1: Eine Benutzungsanleitung haben.

W2: Mehrere Sprachen unterstützen.

W3: Dem User Hilfe anbieten.

3 Projektumgebung

3.1 Entwicklungsprozess

Wir entschieden uns für den iterativen Scrum-Prozess. Es erschien uns Ideal, da wir jede Woche Dienstag mindestens zwei Lektionen Zeit hatten. Somit entschieden wir uns, dass die Sprints 7 Tage dauerten und nicht wie gewöhnlich 30. Wir teilten die Rollen wie folgt auf:

Scrum Master: Rijad Zuzo

Management: Séverin Müller

Product Owner & Entwicklung Rijad Zuzo, Séverin Müller

Customer Verein

3.2 Ablauf

Die Stand-up Meetings erfolgten immer am Anfang der Lektionen am Dienstag bei einem Kaffee oder bei schönem Wetter kurz draussen. Dank des Online-Tools Trello.com konnten wir die Aufgabenpakete sogleich zuteilen und die Meetings waren dank unserer Vorbereitung sehr speditiv.

3.3 Programmiersprache

Die Excel Software läuft ausschließlich auf Microsoft Windows Betriebssystemen, deshalb haben wir uns für die Windows Spezifische Programmiersprache C# entschieden.

Diese in Kombination mit der IDE Visual Studio ermöglichten uns eine angenehme Entwicklung sowohl der Funktionalität und auch des GUI's.

3.4 Verwaltungssystem

Die Auswahl des Verwaltungssystem war sehr einfach da wir die Software als Open Source ausgeben wollten. Wir haben uns für das Git System entschieden und hosteten unser code auf der Website GitHub.

4 Projekt Planung

4.1 Product Backlog

4.2 Sprints

4.3 Sprint Stand-up Meetings

4.4 Sprint Review

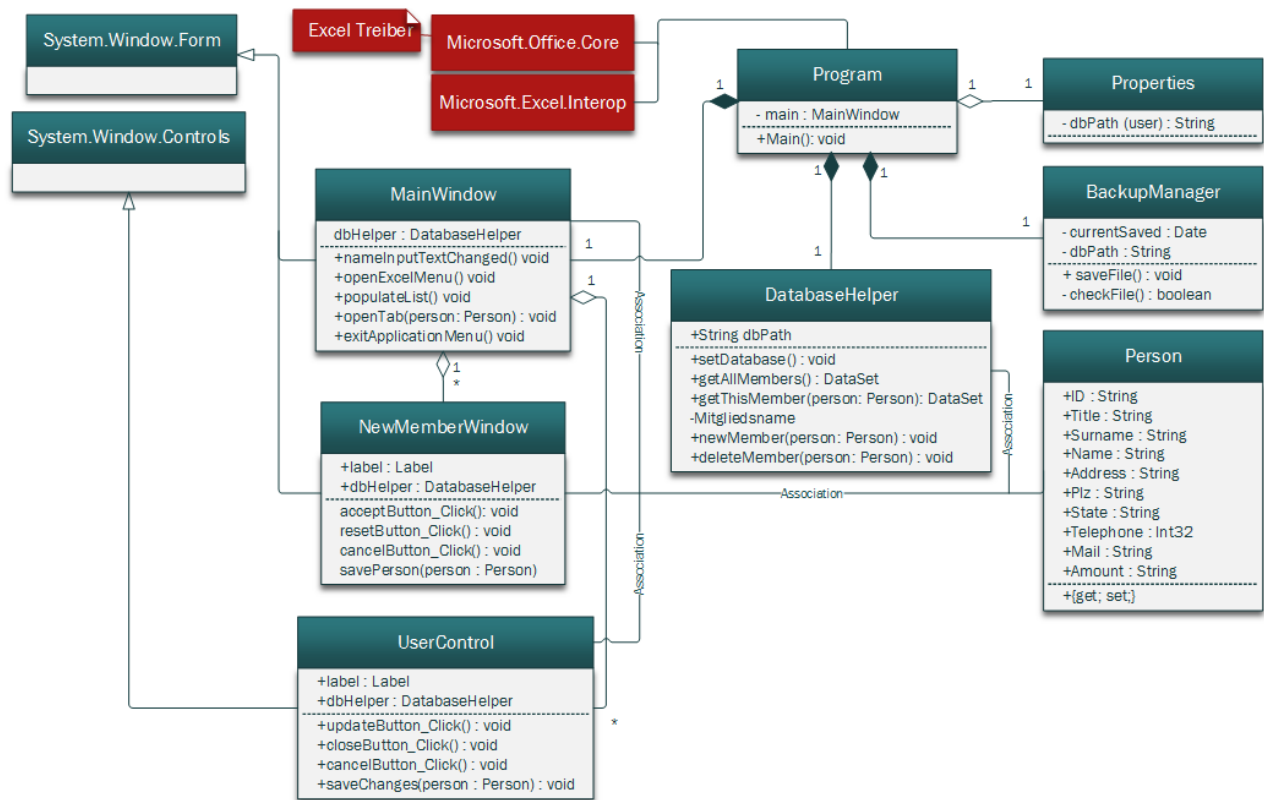


Figure 1: Klassendiagramm

5 Ablauf der Entwicklung

5.1 Modellierung

Die erste Phase der Entwicklung war das Strukturieren der Software. Wir wollten ein klares Design haben das uns die Weiterentwicklung in der Zukunft ermöglicht.

5.1.1 Klassen Diagramm

5.1.2 Aktivitäts-Diagramm

5.1.3 Sequenz Diagramm

5.2 Stunden Journal

Bild oder tabelle rekreiren von der Stunden template vom Sevi.

5.3 GUI

Vorgang bei Gestaltung des GUI's, bedenken usw.

5.4 Database Handling

Schwierigkeiten bei den Referenzen, Voraussetzungen zur Verbindung usw.

5.5 Backup Management

Ansätze für ein effizientes Backuping :).

5.6 Testen

Wie beim Testen vorgegangen wurde!

6 Ziel

6.1 Resultat