

Software Projekt - Excel als SQL Datenbank

Autoren: Rijad Žužo , Séverin Müller

Dozent: Ulrich Hauser

May 9, 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	1
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

Die Motivation für dieses Projekt stand aus dem Beruflichen Leben. Ein Excel File das zum einfachen auflisten einiger Daten erstellt wurde, kann schnell in eine richtige Datenbank auswachsen.

1.1 Ausgangssituation

Das auslesen eines größeren Excel Files ist deutlich langsamer als einer normalen SQL Datenbank der gleichen größe, deshalb ist es sinnvoll das Excel File in eine SQL Tabelle umzuwandeln. Die Konversion erfordert zusätzliche Software Kenntnisse die in dieser Situation unwahrscheinlich sind.

Mit der x2dB Applikation wurde das Problem gelöst. Die Umwandlung ist unnötig und somit kann das Excel File bestehend bleiben.

1.2 Lösungsidee

Die Idee ist es jetzt eine Applikation mit simplem User-Interface zu entwickeln, welches im Hintergrund das Excel File wie eine SQL Tabelle behandelt und sich über entsprechende Treiber mit dem File verbindet. Das ermöglichen uns die Microsoft.Office.Core und Microsoft.Office.Excel.Interop Treiber.

Durch das benutzen der Applikation hat der User einen begrenzten Einfluss auf das File, welches die Kollision bei zbsp. einem Shared File vermeidet. Die Verbindung dauert nur so lange bis die Daten gelesen/geschrieben wurden und danach wird das File wieder freigestellt.

Für zusätzliche Datensicherheit wird nach entsprechender Zeit ein Backup File erstellt und einsetzbar sein. Dies ist natürlich nicht eine Endlösung aber eine guter fix für solche problematische Legacy Situationen.

2 Anforderungsliste

Für die Applikation stellen sich jetzt gewisse Anforderungen- Da es sich um eventuell sehr wichtige Daten handelt, darf es in diesem Prozess keine Datenverluste und ähnliche Ausfälle geben.

2.1 Muss-Anforderungen

Die Applikation muss:

M1: Verbindung zum Excel File erstellen.

M2: Aus dem File Auslesen und Einschreiben können.

M3: Nach dem Auslesen und Einschreiben die Verbindung trennen.

M4: Nach gewisser Zeit einen Backup erstellen.

2.2 Soll-Anforderung

Die Applikation soll:

S1: Leicht Bedienbar sein.

S2: Ein angenehmes User-Interface haben.

S3: Den User vom Excel File abgrenzen.

2.3 Wunsch-Anforderung

Die Applikation könnte:

W1: Eine Benutzungsanleitung haben.

W2: Mehrere Sprachen unterstützen.

3 Projektumgebung

3.1 Entwicklungsprozess

Da es sich um eine Applikation handelt welche eng mit dem User verbunden ist kann es auch //TODO SCRUM

3.2 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.

3.3 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

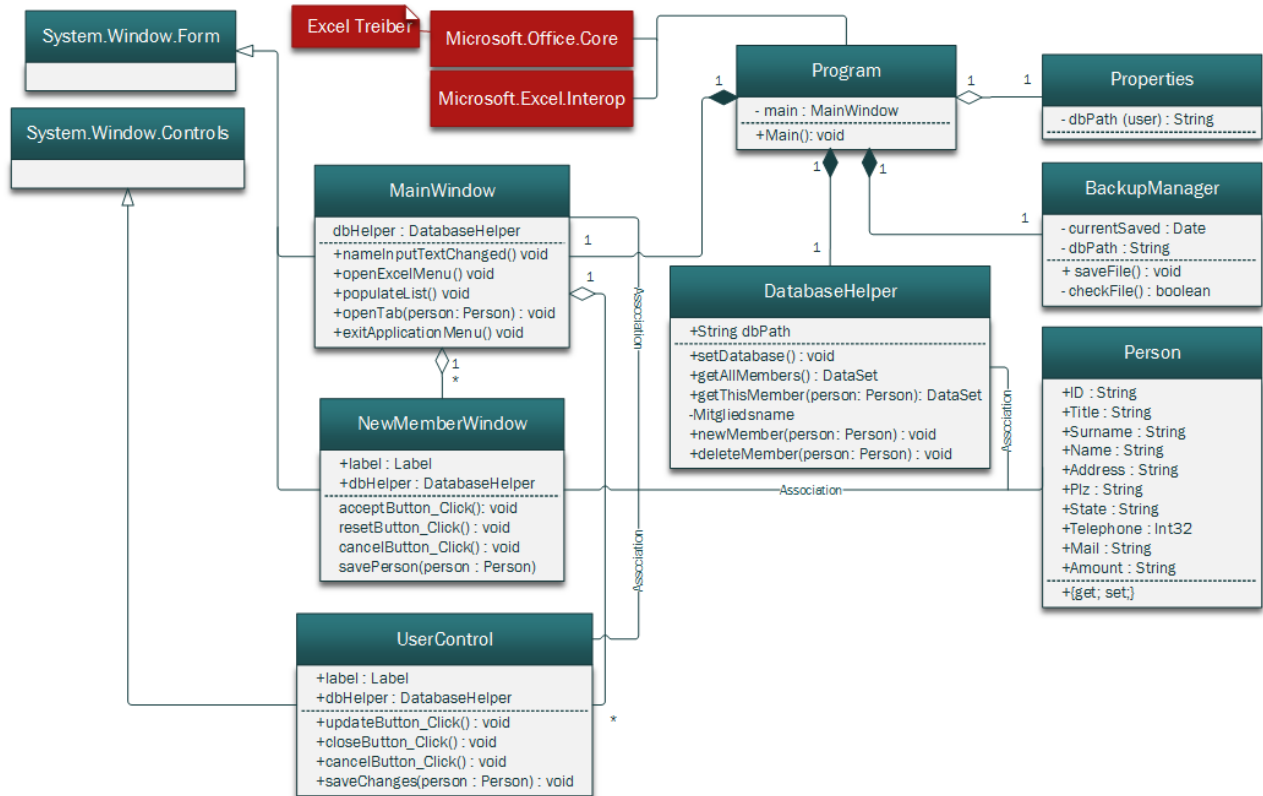
4.4 Sprint Review

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



1.a Klassendiagramm

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