

PA 2023  
Abschlussarbeit

Projekt-Überwachung

Alain Hoch

blue office AG

Turbistrasse 10

6280 Hochdorf

Inhalt

[1 Vorwort 4](#_Toc126139933)

[2 Organisation der Arbeitsergebnisse 4](#_Toc126139934)

[3 Umfeld und Ablauf 4](#_Toc126139935)

[3.1 Aufgabenstellung 4](#_Toc126139936)

[3.2 Aufbau 4](#_Toc126139937)

[3.3 Funktionen 4](#_Toc126139938)

[4 Projektorganisation 5](#_Toc126139939)

[5 Mittel und Methoden 5](#_Toc126139940)

[6 Vorkenntnisse 5](#_Toc126139941)

[7 Vorarbeiten 6](#_Toc126139942)

[8 Arbeiten in den letzten 6 Monaten 6](#_Toc126139943)

[9 Hilfestellung 6](#_Toc126139944)

[10 Zeitplan 7](#_Toc126139945)

[10 Arbeitsjournal 9](#_Toc126139946)

[10.1 Tag 1: 31.01.23 9](#_Toc126139947)

[10.2 Tag 2 (01.02.23) 9](#_Toc126139948)

[10.3 Tag 3 (Datum) 9](#_Toc126139949)

[10.4 Tag 4 (Datum) 10](#_Toc126139950)

[10.5 Tag 5 (Datum) 10](#_Toc126139951)

[10.6 Tag 6 (Datum) 10](#_Toc126139952)

[10.7 Tag 7 (Datum) 10](#_Toc126139953)

[10.8 Tag 8 (Datum) 11](#_Toc126139954)

[10.9 Tag 9 (Datum) 11](#_Toc126139955)

[10.10 Tag 10 (Datum) 11](#_Toc126139956)

[11 Projekt 11](#_Toc126139957)

[11.1 Zusammenfassung (Kurzfassung) 11](#_Toc126139958)

[11.1.1 Ausgangslage (kurze Ausgangssituation) 11](#_Toc126139959)

[11.1.2 Umsetzung 12](#_Toc126139960)

[11.1.3 Ergebnis 12](#_Toc126139961)

[11.2 Einleitung 12](#_Toc126139962)

[11.3 Informieren 12](#_Toc126139963)

[11.3.1 Ziele der Aufgabenstellung 12](#_Toc126139964)

[11.3.2 Vorgaben 12](#_Toc126139965)

[11.3.3 Fragen 12](#_Toc126139966)

[11.4 Planen 12](#_Toc126139967)

[11.4.1 Realisierungskonzept 12](#_Toc126139968)

[11.4.2 Klassen-Diagramm 13](#_Toc126139969)

[11.4.3 Datenbank-Diagramm 14](#_Toc126139970)

[11.4.4 Testkonzept 15](#_Toc126139971)

[11.5 Entscheiden 18](#_Toc126139972)

[11.5.1 Varianten 18](#_Toc126139973)

[11.5.2 Entscheid 18](#_Toc126139974)

[11.6 Realisieren 18](#_Toc126139975)

[11.6.1 Projektumgebung 18](#_Toc126139976)

[11.6.2 Datenbank 18](#_Toc126139977)

[11.6.3 .. Alles was zur Realisierung benötigt wird hier auflisten: 18](#_Toc126139978)

[11.7 Kontrollieren 18](#_Toc126139979)

[11.7.1 Testprotokoll 18](#_Toc126139980)

[11.7.2 Testbericht 18](#_Toc126139981)

[12 Reflexion 18](#_Toc126139982)

[13 Glossar 18](#_Toc126139983)

[14 Abbildungsverzeichnis 18](#_Toc126139984)

[15 Quellenverzeichnis 18](#_Toc126139985)

[16 Anhang 18](#_Toc126139986)

[16.1 Anleitung 18](#_Toc126139987)

[16.2 MSSQL 18](#_Toc126139988)

[16.3 VB.Net 18](#_Toc126139989)

# Vorwort

Diese Dokumentation gehört zur PA von Alain Hoch bei der Firma blue office AG. In dieser Dokumentation ist der Ablauf der Arbeit beschrieben und das Vorgehen dahinter. Die Dokumentation ist in 2 Teile gegliedert. Im ersten Teil wird die detaillierte Aufgabenstellung und der Ablauf der Arbeit aufgezeigt. Es ist ersichtlich mit welchen Mitteln gearbeitet wurde und welche Vorkenntnisse vorhanden sind. Im zweiten Teil kommt dann die Projekt Dokumentation in der die eigentliche Arbeit beschrieben wird, was für Probleme aufgetreten sind und wie die Arbeit getestet wurde. Durch die ganze Arbeit hinweg wurde mit der Projektplanungsmethode IPERKA gearbeitet. Die Reflexion und das Fazit stehen am Schluss.

# Organisation der Arbeitsergebnisse

Die Dokumentation sowie die restlichen Dateien, die nicht im Visual Studio bearbeitet werden können, werden bei jedem Zwischenspeichern oder manuell ins OneDrive gespeichert, das auf meinem geschäfts-Account läuft. Für die Office Dateien wird dort automatisch eine Versionierung durchgeführt, welches bei jedem Zwischenspeichern durchgeführt wird. Die Restlichen Daten werden von mir manuell auf GitHub hochgeladen und dort Versionisiert. Man kann auf GitHub jedes einzelne Hochladen einen so genannten «Push» nachverfolgen und nachschauen was geändert worden ist. Das Projekt wird im Visual Studio programmiert, dies kann mittels internem Terminal und Git-CLI auch Updates machen auf das zugewiesene Projekt.

# Umfeld und Ablauf

## Aufgabenstellung

### Titel der Arbeit

Projekt-Überwachung

### Ausgangslage

Die Firma blue office AG benötigt für Neukunden-Projekte eine Übersicht, bzw. eine Überwachung der Offerierten Dienstleistungsstunden mit den erbrachten Leistungen. Es soll erkennbar sein, ob die erbrachten Dienstleistungen den offerierten Aufwand - aus der Auftragsbestätigung - übersteigen, bevor die Dienstleistungsrechnungen erstellt sind, bzw. die Supportfälle abgeschlossen sind)

### Detaillierte Aufgabenstellung

Hintergrundinformationen: In den Auftragsbestätigungen sind Arbeitsstunden (Dienstleistungsartikel vom Typ «Leistung») erfasst. In den Service-Anfragen sind die «Bearbeitungszeit-Einträge» der geleisteten Stunden hinterlegt, diese Stunden werden automatisch in die Zeiterfassung geschrieben. Es müssen nun die Stunden aus der Zeiterfassung, die dieser Service-Anfrage zugeordnet sind, ausgewertet werden. Allgemein: Es muss ein Add-In für blue office® erstellt werden. Als Grundlage der Erstellung des Add-Ins soll das interne Dokument «Kurzbeschreibung AddIn-Erstellung.pdf» und «Beschreibung AddIn-Möglichkeiten.pdf» verwendet werden. Das Projekt, bzw. die Add-In DLL soll einen sprechenden Namen erhalten. Z.B. boProjektAnalyse.dll. Der Präfix «bo» ist die Abkürzung für «blue office». Aufgabe Adressverwaltung: Add-In erstellen, das in der Adressverwaltung als Control (im Form-Designer) eingefügt werden kann. Es muss eine Tabelle in der Datenbank erstellt werden, die die Zuordnung der Adressen mit den Aufträgen und den Service-Anfragen speichert. Es müssen pro Adresse verschiedene Aufträge und Service-Anfragen hinzugefügt und entfernt werden können. Es dürfen nur Aufträge und Service-Anfragen hinzugefügt werden, die auch der entsprechenden Adresse zugeordnet sind. Alle zugeordneten Aufträge und Service-Anfragen sollen direkt auf dem Control ersichtlich sein. Die Arbeitsstunden aus dem Auftrag (SOLL) sollen nun mit den geleisteten Stunden aus der Zeiterfassung (IST) gegenübergestellt werden. • Summe aller SOLL-Stunden • Summe aller IST-Stunden, diese soll zuzüglich noch pro Verrechnungstyp unterteilt werden in: - Summe aller IST-Stunden «Verrechnen» - Summe aller IST-Stunden «Warten» - Summe aller IST-Stunden «Kulanz» - Summe aller IST-Stunden «Garantie» - Summe aller IST-Stunden «Nicht Verrechnen» Folgendes soll visuell ersichtlich sein (Z.B. rot eingefärbt oder als Prozentanzeige): - IST-Stunden (nur Verrechen+Warten) ggü. SOLL-Stunden «Kundensicht» - IST-Stunden (alle) ggü. SOLL-Stunden «blue office Sicht» Wenn IST-Stunden 90% der SOLL-Stunden erreichen, soll das ebenso visuell ersichtlich sein. Z.B. orange eingefärbt. (Farben ggf. einstellbar) Ein Auftrag, oder Service Anfrage darf nur einmal einer Adresse hinzugefügt werden. Ein Auftrag oder eine Service-Anfrage, soll direkt über die Auflistung(en) geöffnet werden können. Doppelklick und via Kontextmenu Layout: Die Darstellung überlassen wir Alain Hoch. Der SOLL / IST Vergleich muss jedoch gut ersichtlich platziert werden. Das Einfügen von Aufträgen und Service Anfragen soll logisch angeordnet sein. Z.B. via Button oberhalb der entsprechenden Liste und auch via Kontextmenu. Für Auflistungen der Aufträge und Service-Anfragen, müssen die Controls wie «boSearchList» oder «boGrid», aus der blue office API verwendet werden. Bei Auflistung von Aufträgen, muss mindestens die Auftrags-Nr. (A.XXXX), Bezeichnung und Datum aufgeführt werden. Bei Auflistungen von Service-Anfragen, muss mindestens die Service-Nr. (SR.XXXX) und der Titel, Erstellungsdatum aufgeführt werden. Aufgabe Service-Anfrage: Hier muss ein Control erstellt werden das im Form-Designer bei den Service-Anfragen hinterlegt werden kann. Hierbei geht es darum, wenn eine Service-Anfrage geöffnet ist, dass man die SOLL/IST Stunden aller dazugehörigen Aufträge und Zeiterfassungs-Einträge sieht. Es kann sehr minimal gehalten werden und soll nur die entsprechenden Stunden ausweisen. Ausreichend wäre auch eine Prozentanzeige von IST-Stunden (nur Verrechen+Warten) >= SOLL-Stunden. Auch hier überlassen wir Alain Hoch die Darstellung. Das «Control» soll aber nicht zu gross sein und die Kennzahl «Prozentanzeige» enthalten. Evtl. auch mit Farben arbeiten bei kritischen Werten. Benutzerhandbuch: Es soll ein Benutzerhandbuch für blue office Mitarbeiter erstellt werden.

**Optional:** • Im Control der Service-Anfrage soll die Möglichkeit bestehen, das Control der Adressverwaltung aufzurufen. Das soll in einem Dialog (eigener Form) geschehen, das die Möglichkeit besteht weiter Aufträge oder Service-Anfragen hinzuzufügen.

**Prio 1.** • Verwaiste Einträge «Leichen» Wenn z.B. ein Auftrag oder eine Service-Anfrage gelöscht wird, weiss das Add-In nichts davon. D.H. es gibt in der Tabelle Einträge, die keinen Treffer mehr haben. Diese könnten bereinigt werden, wenn man die Adresse wechselt.

**Prio 2.** • (Löschen ist bei blue office nicht erlaubt, wäre aber bei anderen Kunden denkbar)

## Aufbau

Die Grafische Darstellung ist mir selber überlassen, jedoch für die Auflistung der Daten und der Belege und Service Requests, muss ich eine boSearchList verwenden (für Belege und SR’s) und für die Daten der SR’s werde ich ein boGrid verwenden, das sind alles Grids und Listen, welche von unserem API zur Verfügung gestellt werden, was jedoch einen Kern aus der .Net Framework hat.

## Funktionen

Die Arbeitsstunden aus dem Auftrag (SOLL) sollen nun mit den geleisteten Stunden aus der Zeiterfassung (IST) gegenübergestellt werden.

• Summe aller SOLL-Stunden

• Summe aller IST-Stunden, diese soll zuzüglich noch pro Verrechnungstyp unterteilt werden in:

- Summe aller IST-Stunden «Verrechnen»

- Summe aller IST-Stunden «Warten»

- Summe aller IST-Stunden «Kulanz»

- Summe aller IST-Stunden «Garantie»

- Summe aller IST-Stunden «Nicht Verrechnen»

Folgendes soll visuell ersichtlich sein (Z.B. rot eingefärbt oder als Prozentanzeige): - IST-Stunden (nur Verrechen+Warten) ggü. SOLL-Stunden «Kundensicht» - IST-Stunden (alle) ggü. SOLL-Stunden «blue office Sicht»

# Projektorganisation

**Lehrbetrieb und Durchführungsort:**

blue office AG

Turbistrasse 10

6028 Hochdorf

041 911 07 11

info@blue-office.ch

**Kandidat:**

Alain Hoch

Schulhausstrasse 15

6030 Ebikon

079 613 46 89

alain.hoch@hotmail.com

**Berufsbildner/Lehrfirma:**

Frank Melber

Bénédict-Schule Luzern

Inseliquai 12B

6005 Luzern

**Verantwortliche Fachkraft:**

Marcel Ruckstuhl

blue office AG

Turbistrasse 10

6028 Hochdorf

m.ruckstuhl@blue-office.ch

Hauptexperte

Michael Estermann

# Mittel und Methoden

- Programmiersprache: VB.NET

- Auflistungen (Grids): Wenn immer möglich Listen aus der blue office API (blueoffice.controls.dll) verwenden. "boSearchList" und "boGrid"

- Ressourcen wie Icons. Falls entsprechende Grafik vorhanden, immer diese der blue office API (blueoffice.common.dll) verwenden - Datenbankzugriff: Erfolgt immer über die blue office API. (blueoffice.common.dll und blueoffice.DAL.dll)

# Vorkenntnisse

- blue office Klassen (API)

- Erstellung von blue office Add-In

- Erfahrungen in der Applikation blue office

- Kenntnisse der internen Abläufe

# Vorarbeiten

- Raster für Dokumentation

- Vorlage für Anleitung

# Arbeiten in den letzten 6 Monaten

- e-Shop Import, Code von VB6 zu VB.NET "übersetzt".

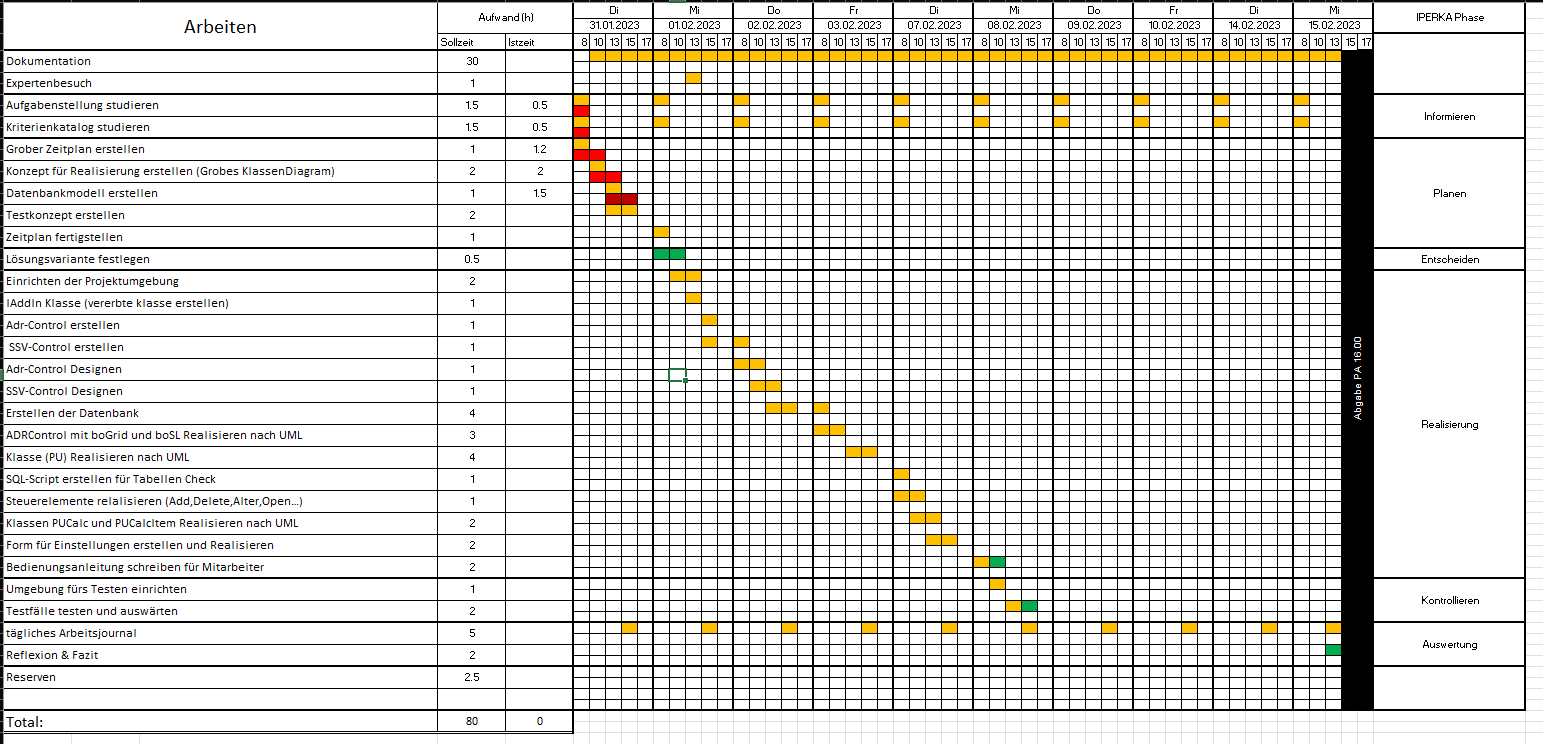
- GDPDU Export

- Datenimport aus andere Datenbanken

# Hilfestellung

Anselm: DB- Relationen und Darstellung mit diesen durch Anselm und Marcel.

# Zeitplan



# Arbeitsjournal

## Tag 1: 31.01.23

|  |  |
| --- | --- |
| Tätigkeiten | Aufgabenstellungen Studiert, Kriterien-Katalog studiert, Zeitplan erstellt, Klassen- DB-Diagramm erstellt, Testkonzept erstellt |
| Erreichte Ziele | Für heute habe ich alles erreicht, was ich mir vorgenommen habe, ich wollte die Planung heute so gut wie möglich vorbereiten.   * Zeitplanung * Diagramme * Testkonzepte |
| Probleme | Ich musste etwas genauer hinschauen beim Datenbank-Diagramm und beim Klassen-Diagramm, diese waren nicht einfach zu erstellen. Jedoch denke ich habe ich einen guten Weg gefunden diese Grafisch dar zu stellen. |
| Hilfestellung | Marcel hatte mir einen alten boMandanten gegeben und geholfen zu installieren, um für Morgen gewappnet zu sein, wenn es an die DB geht. |
| Ausserplanmässige Arbeiten | E-Mail beantwortet. |
| Reflexion | Im Ganzen bin ich gut in die PA gestartet, ich konnte mich super konzentrieren und war zielstrebig auf dem Weg unterwegs, welcher ich mir heute geplant habe, bislang läuft es gut.  Wie gesagt, bei den Diagrammen tat ich mich etwas schwerer als gedacht, jedoch bin ich mit dem Ergebnis zufrieden.  Den Zeitplan musste ich zwischendurch wieder anpassen, da ich 1-2 Fehler gefunden gehabt habe. Diese waren jedoch schnell behoben und störten mich nicht weiter an der PA. |

## Tag 2 01.02.23

|  |  |
| --- | --- |
| Tätigkeit | Datenbank-Diagramm neu erstellt, Zeitplan vervollständigt, Lösungsvariante Festgelegt, Projekt Umgebung eingerichtet, Alle Klassen erstellt jedoch nur ctlADR und ctlSSV implementieren angefangen. Experten Besuch |
| Erreichte Ziele | Alles, was ich mir für heute vorgenommen hatte. Unteranderem wurde der Meilenstein «Lösungsvariante festlegen» erreicht. |
| Probleme | Ich hatte ein Problem mit meinem Datenbank-Diagramm, es störte mich gestern Abend schon, wie ich es gemacht hatte, daher habe ich es heute Morgen mit Fremdhilfe von Anselm neu erstellt. |
| Hilfestellung | Darstellung des DB-Diagramms und Besprechung wie die Tabellen verbunden werden sollen durch Anselm. |
| Ausserplanmässige Arbeiten | Keine |
| Reflexion | Da mich das ganze gestört hatte, wie ich das DB-Diagramm gemacht hatte, habe ich es erneut erstellt. Ich bin froh habe ich das gmacht, somit hatte ich die Relationen auch besser verstanden wie ich sie machen sollte. Weiter verlief der Tag nach Plan, das ganze hatte mich nicht weit nach hinten geworfen, im Gegenteil ich lag sehr gut im zeitrahmen und konnte so etwas vorarbeiten. Ich habe am Schluss noch Par zielen an der Dokumentation gearbeitet und bin mit dem Heutigen Tag sehr zufrieden.  Zum Experten-Besuch war ich etwas nervös, jedoch kam die Entspannung sehr schnell als ich merkte, dass das alles nicht so wild ist.  Weiter werde ich morgen versuchen weiter konzentriert an der Arbeit zu sein, um im Zeitplan zu bleiben. |

## Tag 3 02.02.23

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme | GitHub |
| Hilfestellung | keine |
| Ausserplanmässige Arbeiten | Reparieren von GitHub |
| Reflexion |  |

## Tag 4 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

## Tag 5 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

## Tag 6 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

## Tag 7 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

## Tag 8 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

## Tag 9 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

## Tag 10 (Datum)

|  |  |
| --- | --- |
| Tätigkeit |  |
| Erreichte Ziele |  |
| Probleme |  |
| Hilfestellung |  |
| Ausserplanmässige Arbeiten |  |
| Reflexion |  |

# Projekt

## Zusammenfassung (Kurzfassung)

### Ausgangslage (kurze Ausgangssituation)

### Umsetzung

### Ergebnis

## Einleitung

## Informieren

### Ziele der Aufgabenstellung

Es soll eine Ansicht generiert werden, bei welcher Aufträge mit Leistungen erfasst werden können. Dazu werden noch sogenannte Service Requests erfasst. In diesen werden die geleisteten Stunden geschrieben welche erbracht wurden. Für die Stunden gibt es verschiedene Stadien in welche sie gesetzt werden können. (Verrechnet, nicht verrechnet, Warten, Kulanz, Garantie)

Wie die Namen schon verraten, werden nicht immer Beträge verrechnet.   
Um die geleisteten Stunden im Auge behalten zu können, wird nun diese Ansicht generiert. Sie soll eine Übersicht darstellen, in welcher man gleich erkennt wie weit ein Auftrag ist, im Vergleich die im Auftrag erfassten und die schon geleisteten Stunden.

Die einzelnen Werte, von den oben benannten Stadien sollen zusammengefasst aufgezeigt werden und leicht lesbar sein. Dabei soll ein Soll / Ist vergleich gemacht werden, bei welchem Farblich gekennzeichnet wird, ob sich der Vergleich noch im «grünen Bereich» befindet oder nicht mehr.

Zusätzlich soll Grafisch eine Anzeige gestaltet werden, welche eine prozentuale Anzeige anzeigt.

### Vorgaben

- Korrekte Darstellung der Daten bzw. Stunden (geleistet / soll)

- Farblich dynamische Veränderung der Ist Zahl, sobald ein gewisser Prozentualer Wert erreicht wurde (vom Benutzer einstellbar)

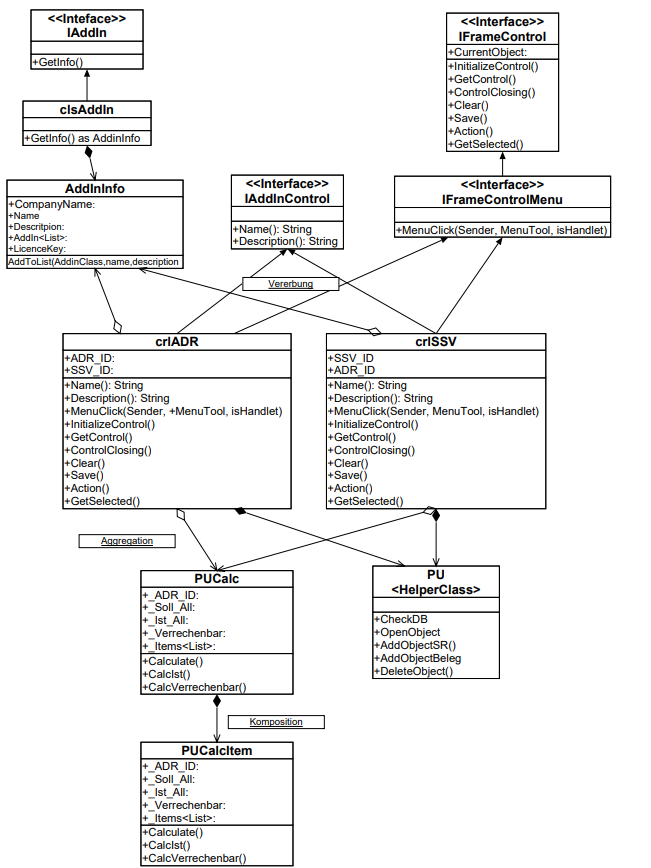
### Fragen

## Planen

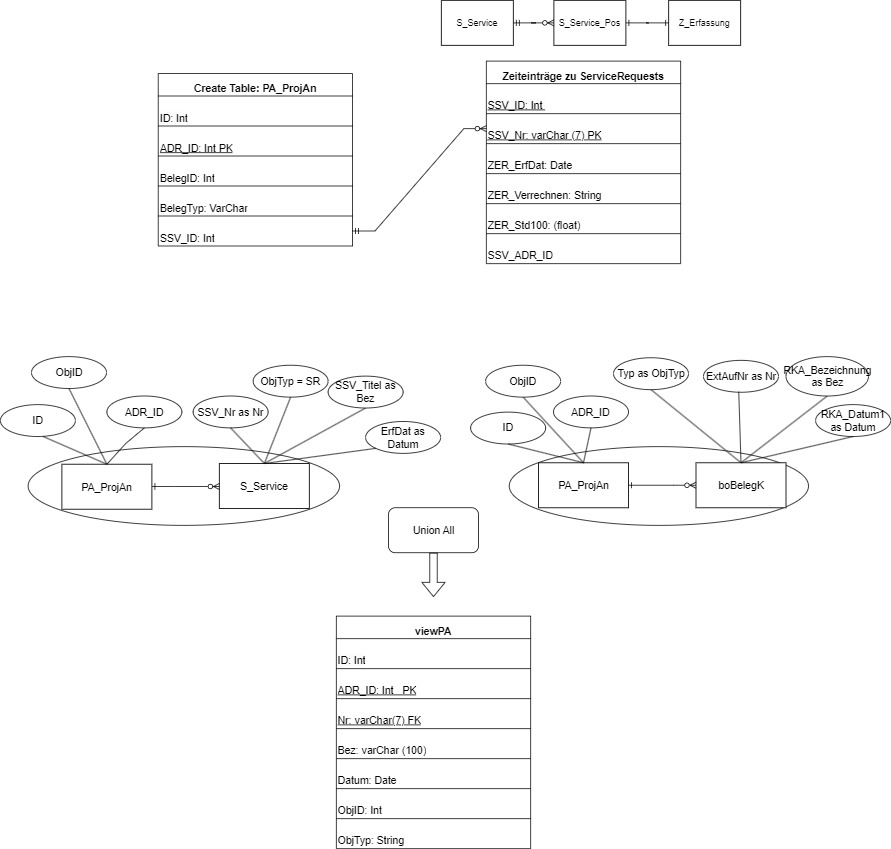
### Realisierungskonzept

-IPERKA beschreiben!

### Klassen-Diagramm



### Datenbank-Diagramm



### Testkonzept

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 1 |
| **Bezeichnung** | | Add-in lässt sich in der Adresse öffnen |
| **Situation** | **Von-Zustand** | Ungeöffnet (Adress Verwaltung) |
| **Signal** | Klick auf Register Projekt Analyse |
| **Eingaben** |  | - |
|  | - |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Das Register geht auf und die ganze Projekt Analyse (boGrid,boSL mit Buttons und Progress bar sind ersichtlich) |
| **Anzeige** | Noch nichts |
| **Meldungen** | Nichts |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 2 |
| **Bezeichnung** | | Befüllen der boSL |
| **Situation** | **Von-Zustand** | Leere Liste |
| **Signal** | Mit Button „Add“ Aufträge Dialog öffnen und Aufträge einfügen können |
| **Eingaben** |  | Auswahl von Aufträgen im Dialog |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Wieder zurück auf der Project Analyse |
| **Anzeige** | Aufträge welche ausgewählt wurden sollen dann angezeigt werden in der boSL. |
| **Meldungen** | Keine. |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 3 |
| **Bezeichnung** | | Füllen von boGrid |
| **Situation** | **Von-Zustand** | Leer |
| **Signal** | SR-Dialog öffnen |
| **Eingaben** |  | Auswahl der SR; welche von der SQL Abfrage direkt auf den Auftrag abgestimmt werden |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Nach dem Schliessen des Dialogs soll wieder die Projektanalyse angezeigt werden. |
| **Anzeige** | Anzeigen der SR’s im boGrid welche vom Benutzer ausgewählt wurden |
| **Meldungen** | keine |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 4 |
| **Bezeichnung** | | Daten in dem boGrid |
| **Situation** | **Von-Zustand** | Aufträge und SR’s wurden in de boSL ausgewählt. Nach dem Schliessen des Dialogs sollen dann die richtigen Daten angezeigt werden auf dem boGrid. |
| **Signal** | Schliessen von Dialog |
| **Eingaben** |  | Die Eingaben vom Dialog welche SR’s oder A geöffnet und verwendet werden sollen. |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Daten von (Verrechnen, nichtverrechnen, warten, Kulanz ist, soll) sollen korrekt angezeigt werden |
| **Anzeige** | // |
| **Meldungen** | Keine. |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 5 |
| **Bezeichnung** | | Daten sollen noch da sein, nach dem Schliessen der Projekt Analyse. |
| **Situation** | **Von-Zustand** | Geschlossener Projekt Analyse (Adressverwaltung) |
| **Signal** | Öffnen der PA |
| **Eingaben** |  | Keine |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Dass die Daten, welche bim Schliessen von der PA noch vorhanden waren, immer noch da sind. (Abfrage auf SQL Tabelle und Daten ablesen) |
| **Anzeige** | // |
| **Meldungen** | Keine |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 6 |
| **Bezeichnung** | | Vergleich der Zahlen |
| **Situation** | **Von-Zustand** | Daten werden angezeigt |
| **Signal** | Keins |
| **Eingaben** |  | Keine |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Das angezeigte Daten korrekt ausgewertet sind. |
| **Anzeige** | Korrekte Daten |
| **Meldungen** | Keine. |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 7 |
| **Bezeichnung** | | Verfärbung der Ist Zahl bei gewisser % Erreichung |
| **Situation** | **Von-Zustand** | Neutral orange Schriftfarb |
| **Signal** | Sobald ein gewisser (vom Benutzer eingestellter Prozentsatz erreicht wurde, soll sich die IST Zahl farblich verändern (Rot) |
| **Eingaben** |  | Keine |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Verfärbte IST-Zahl |
| **Anzeige** | ROT Ist Zahl |
| **Meldungen** | Keine |

|  |  |  |
| --- | --- | --- |
| **Testnummer** | | 7 |
| **Bezeichnung** | | Verfärbung der Ist Zahl bei gewisser % Erreichung |
| **Situation** | **Von-Zustand** | Neutral schwarzer Schriftfarb |
| **Signal** | Sobald ein gewisser (vom Benutzer eingestellter Prozentsatz erreicht wurde, soll sich die IST Zahl farblich verändern (Orange) dieser wird 20% bevor die vom Benutzer eingestellte Rote Verfärbung erreicht wird automatisch vollzogen. |
| **Eingaben** |  | Keine |
|  |  |
| **Erwartetes**  **Resultat** | **Ziel-Zustand** | Verfärbte IST-Zahl |
| **Anzeige** | Orange Ist-zahl |
| **Meldungen** | Keine |

## Entscheiden

### Varianten

Da in der Aufgabenbeschreibung festgelegt wurde, dass das ganze Projekt als ein AddIn gestaltet werden soll, ist die Variante somit schon festgelegt.

Was zur Alternative gestanden hätte:   
Das Projekt als ein eigenes Control zu implementieren, welches dann fix verwendet wird.   
Jedoch ist es das Ziel, dieses Projekt am Schluss auch zu verkaufen an interessierte Kunden und daher ist ein AddIn eine bessere Wahl.

### Entscheid

Das Projekt ProjektAnalyse wird als Add-In realisiert. Dabei wird intern der Formdesiner angesteuert, bei welchem eingestellt werden kann, wie das Add-In (Control) aussehen und wo es platziert werden soll. Das ist eine schon implementierte Funktion unseres ERP’s welche hier angesteuert wird. Es macht den Umgang mit unseren Add-Ins und die dazugehörigen Controls einfach.

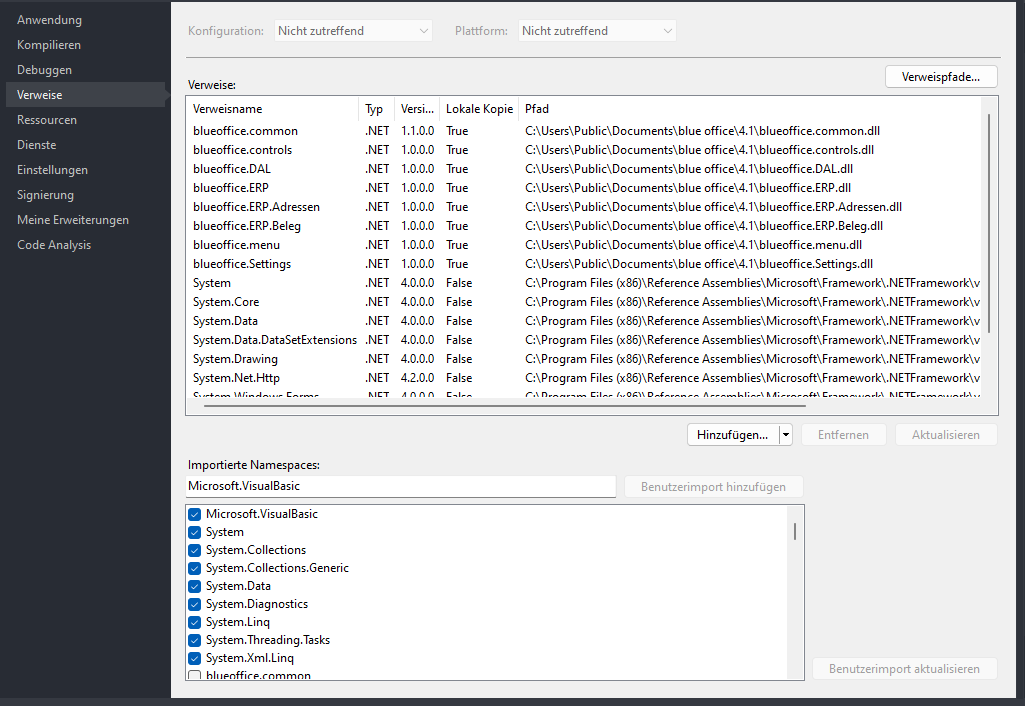
Der Entscheid wurde mir abgenommen, da im Auftrag schon definiert wurde, das ganze als Add-In zu gestalten. Den Formdesigner zu verwenden ist dann logisch.

## Realisieren

### Projektumgebung

Die Projektumgebung ist mir Wohl bekannt. Wir haben auf unseren Computern Folder, welche die Daten vom ganzen Programm blue office enthalten, das ist unsere Entwicklerumgebung, in welcher wir jegliche Projekte verweisen und dort hinaus starten wir auch das Kompilieren und Debuggen.

Es enthällt alle benötigten DLL’ s. Daher müssen wir intern im Visual Studio jegliche verweise darauf machen, welche wir wie folgt machen:

Schritt Eins ist doubleclick auf My Project:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

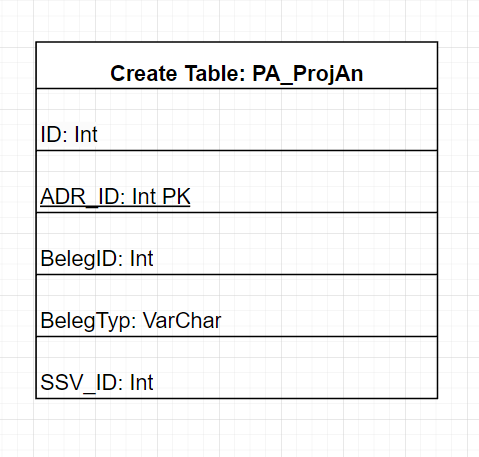


Dabei landen wir hier MyProject und dann auf Verweise Hierbei müssen wir in den Entwickler Ordner hinein und auf alle benötigten DLL’s verweisen. Somit kann ich später mit den benötigten NameSpaces arbeiten.

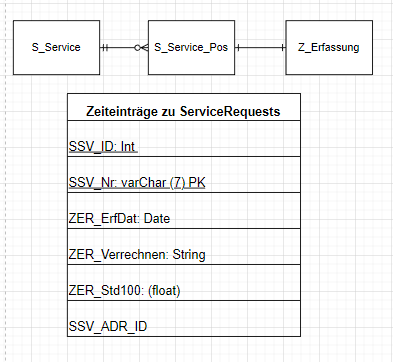
Weiter habe ich Das Projekt auf GitHub hochgeladen, damit immer eine Versionierung gemacht wird, sobald ich es hochlade, das heisst konkret; bei jedem Hochladen auf GitHub die Veränderungen dokumentiert werden und ich diese dann später nachschauen kann was genau geändert worden ist, oder ich kann zu diesem Stand zurückspringen, zu welchem ich möchte.

GitHub ist eigentlich nichts weiteres als eine Grosse Datenbank, welche Codes abspeichert. Von dieser Plattform aus kann man Code herunterladen oder hochladen.

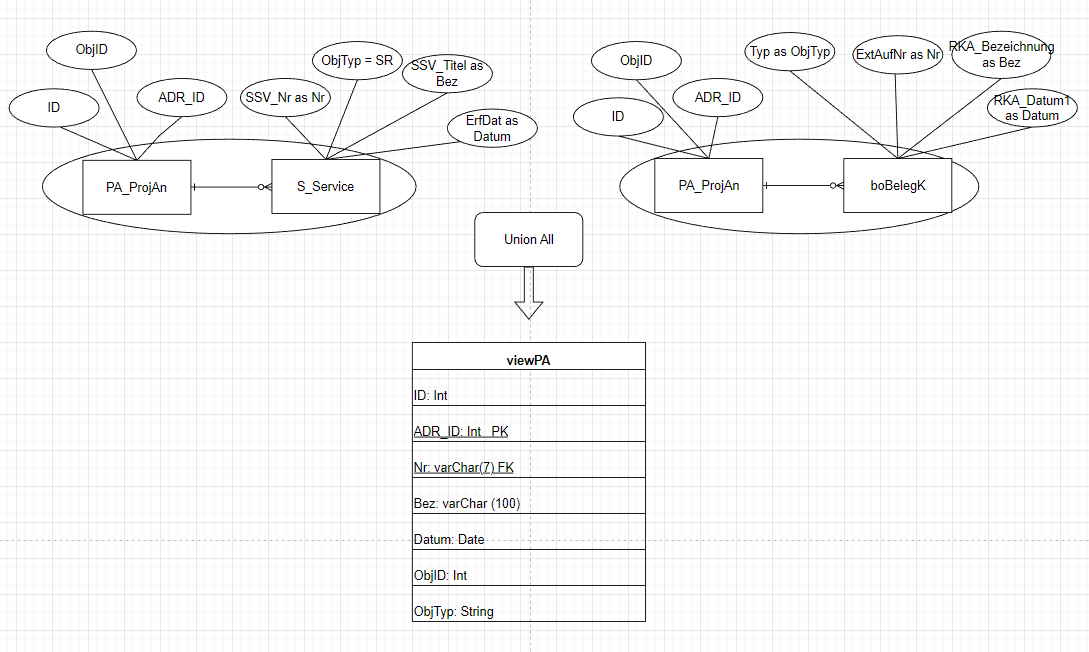
### Datenbank



Eigener Tabelle erstellt um die eingegebenen Daten vom welche in der Projektanalyse eingegeben wurden abzuspeichern und wieder abrufbar zu machen, sobald die Projektanalyse aufgerufen wird.



Abfrage von Zeiterfassung



Union All zwischen PA\_ProjAn und boBelegK | Pa\_ProjAn und S\_Service

### .. Alles was zur Realisierung benötigt wird hier auflisten:

## Kontrollieren

### Testprotokoll

### Testbericht

# Reflexion

# Glossar

# Abbildungsverzeichnis

# Quellenverzeichnis

# Anhang

## Anleitung

## MSSQL

## VB.Net