Swiss Travel Guide

Boss Info AG

info@bossinfo.ch

15. November 2017

Version 1.0/DLA

# Inhaltsverzeichnis

[Swiss Travel Guide 1](#_Toc498510132)

[1 Inhaltsverzeichnis 2](#_Toc498510133)

[2 Einleitung 3](#_Toc498510134)

[3 Ausgangslage 3](#_Toc498510135)

[4 Der Kunde «Traveller» ist mit folgenden Anforderungen zur Firma «Boss Info AG» gekommen: 3](#_Toc498510136)

[5 Ziel 3](#_Toc498510137)

[6 Richtlinien und Informationen 3](#_Toc498510138)

[6.1 Entwicklerrichtlinien 3](#_Toc498510139)

[6.1.1 Naming Conventions 3](#_Toc498510140)

[6.1.2 Declaration 4](#_Toc498510141)

[6.1.3 Comments 4](#_Toc498510142)

[6.1.4 Statements (New Lines, Indentation) 4](#_Toc498510143)

[7 Anforderungen 4](#_Toc498510144)

[7.1 Anforderungen Kunde 4](#_Toc498510145)

[7.1.1 Usecases des Kunden 4](#_Toc498510146)

[7.1.2 Anforderungskatalog 5](#_Toc498510147)

[7.2 Anforderungen 3t Personen 5](#_Toc498510148)

[8 Designe 5](#_Toc498510149)

[8.1 GUI-Designe 5](#_Toc498510150)

[8.2 UML-Diagramme 5](#_Toc498510151)

[8.2.1 Klassendiagramm 5](#_Toc498510152)

[9 Schnittstellenbeschrieb 5](#_Toc498510153)

[9.1 Benutzersicht 5](#_Toc498510154)

[9.2 Code 7](#_Toc498510155)

[10 DTA Setup 7](#_Toc498510156)

# Einleitung

Im ÜK Modul 318 behandeln wir die Themen Analysieren und objektbasiert programmieren mit Komponenten. Im Verlauf des Kurses haben wir den Auftrag bekommen eine Lösung für einen fiktiven Kunden zu erstellen. Dieses Dokument dient als Dokumentation des Kurses / des Projektes. Es soll eine Zusammenfassung von Dokumentation, Benutzerhandbuch, Projektkonzept, Richtlinien und Hintergrundinformationen sein. Das Dokument richtet sich an Entwickler, End-User und 3t Personen. Die Abschnitte werden separat eingeleitet und in der Einleitung spezifisch an eine Lesergruppe gerichtet werden.

# Ausgangslage

# Der Kunde «Traveller» ist mit folgenden Anforderungen zur Firma «Boss Info AG» gekommen:

* Desktop-Anwendung welche folgende Punkte erfüllt:
  + Anhand von Start/End-Station einen Fahrplan abrufen.
  + Anhand einer Station eine Abfahrtstafel abrufen.
  + Standort anzeigen anhand des Stationsname
  + Abgerufene Resultate verbreiten über E-Mail.
  + Stationen suche anhand von Koordinaten/Ortsangaben.

Die Anforderungen werden durch den Kunden in verschieden Prioritäten eingeteilt.

# Ziel

Der Kunde soll bis 21.11.2017 einen ersten funktionsfähigen Prototyp erhalten. Der Prototyp soll ohne Entwicklerumgebung startbar sein. Die Basis soll in C# für Windows-Betriebssysteme entwickelt werden. Das Ziel ist das im Prototyp alle Anforderungen der Priorität 1 abgebildet sind. Alle weitern Anforderungen sind nicht erfolgskritisch für die Umsetzung des Prototypen.

# Richtlinien und Informationen

Dieser Abschnitt richtet sich an Entwickler welche die Applikation erweitern/ kontrollieren wollen. Im Abschnitt werden Richtlinien für die Qualität der Entwicklung hinterlegt, sowie weitere hilfreiche Informationen zum Entwicklungsablauf.

## Entwicklerrichtlinien

### Naming Conventions

Alle Namen werden auf Englisch geschrieben.

Alle Variablen, Eigenschaften, Methoden, Klassen, GUI-Controls werden PascalCase geschrieben.

Klassen werden Klassenname underline c geschrieben

Variablen werden Variablenname underline g für globale und underline l für lokale geschrieben

Eigenschaften werden Eigenschaftenname underline p geschrieben

Methoden werden Methodenname underline m geschrieben

GUI-Controls werden Controlname underline Objektart geschrieben

### Declaration

Variablen werden immer initialisiert. Lokale Variablen werden am Anfang der jeweiligen Funktion deklariert. Globale Variablen am Anfang der Klasse auf welche sie sich beziehen.

### Comments

Grundsätzlich wird nur kommentiert was vom Namen her nicht klar ist. Alle public Methoden werden mit /// kommentiert. Änderungen werden wie folgt dokumentiert //Laufnummer für eine Zeile. //- - Laufnummer und //++Laufnummer für mehrzeilige Anpassungen.

Wenn Änderungen vorgenommen wurden werden diese am Anfang der jeweiligen Klasse oder der Main Methode wie folgt dokumentiert: Laufnummer 4Leerschläge Datum(dd.mm.yyyy)/Firmenkürzel / Entwicklerkürzel Leerschlag Bindestrich Leerschlage Beschreibung der Anpassung. Wenn mehrere Anpassungen gemacht wurde mit der gleichen Laufnummer dann wird das Datum angepasst und die weitere Anpassung auf eine neue Zeile geschrieben. Jede Anpassung startet mit Bindestrich Leerschlag und alle Bindestriche sollten untereinander angeordnet sein. Zwischen den einzelnen Laufnummerblöcken wird eine Zeile abstand gelassen.

### Statements (New Lines, Indentation)

Schleifen usw werden wie folgt geschrieben:

if (Car\_l == Bike\_l)

{

RunPlane\_m;

}

# Anforderungen

Dieser Abschnitt behandelt die Anforderungen welche vom Kunden sowie 3t Personen an das Projekt gestellt werden.

## Anforderungen Kunde

### Usecases des Kunden

Folgende Usecases wurden mit dem Kunden zusammen definiert:

### Anforderungskatalog

Folgender Anforderungskatalog mit Prioritätenliste wurde erstellt:

## Anforderungen 3t Personen

# Designe

Dieser Abschnitt befasst sich mit der Planung des Designs der Applikation. Er richtet sich an Entwickler sowie 3t Personen welche sich für den Projektablauf interessieren.

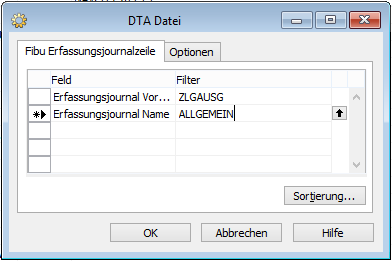
## GUI-Designe

## UML-Diagramme

### Klassendiagramm

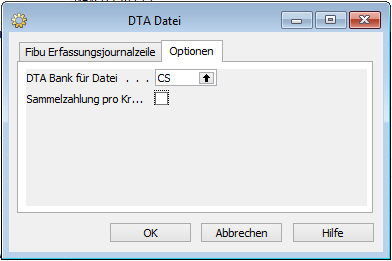
# Schnittstellenbeschrieb

## Benutzersicht

Im folgenden Abschnitt wird beschrieben wie die Schnittstelle für den Endbenutzer aussieht und was man alles beachten muss.

2.

1.



4.

3.

1. Hier wird der Erfassungsjournal Vorlagenname definiert. Im Normalfall wird dies direkte anhand des aktuellen Erfassungsjournal definiert und muss nicht bearbeitet werden.
2. In dieser Zeile wird hinterlegt wie das Erfassungsjournal heisst das bearbeitet werden soll. Auch dies wird anhand des aktuellen Erfassungsjournal automatisch hinterlegt.
3. Hier wird die Bank angegeben von welcher aus die Zahlungen gemacht werden sollen. Die Bankinformationen werden auf der Tabelle DTA Setup hinterleget.
4. Wenn man dieses Feld anwählt wird aus den einzelnen Posten des Erfassungsjournals eine Sammelzahlung erstellt. Das heisst pro Kreditorenbank und Zahlungsart zusammengefasst.

Wenn dann in diesem Fenster «OK» gedrückt wird dann erstellt die Schnittstelle eine Textdatei im Ordner welcher im DTA Setup hinterlegt wurde. Siehe Kapitel 4. DTA Setup.

## Code