**Dokumentation Projektarbeit  
ÜK Modul 318**

****

Projekt TrafficSchedules

Autor: Simon Kaspar

Instruktor: Hanspeter Stalder

Dezember 2017

Inhalt

[2 Einleitung 3](#_Toc500858203)

[3 Zweck des Dokuments 3](#_Toc500858204)

[4 Funktionen 3](#_Toc500858205)

[4.1 Umgesetzte Funktionen 4](#_Toc500858206)

[4.2 Nachträglich geänderte Funktionen 4](#_Toc500858207)

[5 Diagramme 4](#_Toc500858208)

[6 Testfälle 7](#_Toc500858209)

[7 Installationsanleitung 9](#_Toc500858210)

[8 Quellen 9](#_Toc500858211)

# Einleitung

In diesem ÜK war das Ziel, mit der Hilfe von öffentlich zugänglichen Fahrplandaten ein Programm zu entwickeln, dass diese Daten mit einem möglichst benutzerfreundlichen und übersichtlichen GUI darstellt.

# Zweck des Dokuments

Diese Dokumentation soll meinen Denkprozess und Arbeitsprozess während der Projektarbeit veranschaulichen.

Zur Übersichtlichkeit habe ich versucht, mich an die gängigsten Richtlinien für Code zu halten, unter anderem:

* Sprechende Namen (Variablen, Methoden, Klassen)
* Möglichst viel lokale, wenig globale Variablen
* Keine Wiederholungen
* Funktionen Ordnen/gruppieren
* Kommentare für komplexe Methoden

Die Anforderungen lauteten:

* Start- und Endstation mittels Textsuche suchen können
* Die aktuellen Verbindungen zwischen zwei Stationen anzeigen
* Die Verbindungen von einer bestimmten Station in einer Abfahrtstafel ansehen können
* Während der Suche sollen Suchresultate erscheinen
* Es sollen nicht nur aktuelle Verbindungen gesucht werden, sondern Verbindungen sollen zu einem gewünschten Datum und einer gewünschten Zeit angezeigt werden.
* Eine Station soll auf einer Karte betrachtet werden können, damit man weiss, wie die Situation vor Ort aussieht.
* Weitere Stationen in meiner Nähe sollen angezeigt werden, damit ich schnell einen Anschluss erreichen kann.
* Die gefundenen Resultate sollen per E-Mail verschickt werden können.

# Funktionen

## Umgesetzte Funktionen

Gemäss den Anforderungen für das Projekt kann man mit dem Programm den Fahrplan abfragen. Dazu ist es möglich, die Namen der Stationen komplett einzutippen, oder auch Vorschläge anzeigen zu lassen mit Stationen mit ähnlichem Namen.

Die Verbindungen werden angezeigt mit Von, Bis, Abfahrt, Ankunft, Dauer und Gleis. Die Zeiten und Dauer der Verbindung werden in ein gut lesbares Format geparst. Die Abfahrten der Station werden mit Abfahrt, Nach, Nummer, Gleis und Netzbetreiber angezeigt. Per Knopfdruck können Start- und Endstation getauscht werden. Via DateTimePicker können das Datum und die Zeit der Abfahrt ausgewählt werden. Per Knopfdruck auf "Neue Suche" kann das Formular geleert werden, um eine neue Suche zu starten. Ausserdem kann per Enter Taste die Suche gestartet werden. Per Leertaste kann die ausgewählte Station aus der Liste bestätigt werden. Auf Wunsch wird ein Email an die gewünschte Adresse mit den Verbindungen oder der Abfahrtstabelle verschickt. Bei den Verbindungen wird der Standort des Abfahrtsortes auf einer Karte angezeigt. Bei der Abfahrtstabelle wird eine Karte der Umgebung der Station oder des Ortes angezeigt.

## Nachträglich geänderte Funktionen

Die Listbox für die Verbindungen wurde zu einem DataGridView geändert. Der Vorteil ist, dass man die Daten in mehreren Spalten anzeigen lassen kann und so eine bessere Übersicht hat. Aus Layout- und Platzgründen wurden die Elemente anders angeordnet.

Ich habe die SwissTransport API mehrere Male erweitert, um zusätzliche Daten zu erhalten.

Einerseits wird nun auch das Gleis bei der Abfahrtstabelle mit angezeigt. Andererseits wird bei den Verbindungen und der Abfahrtstabelle die Zeit und das Datum mit der Abfrage mitgeschickt, um passende Pläne zur gewünschten Zeit zu erhalten.

# Diagramme

Nachfolgend sehen Sie die Diagramme, die zur Planung des Projekts gedient haben.

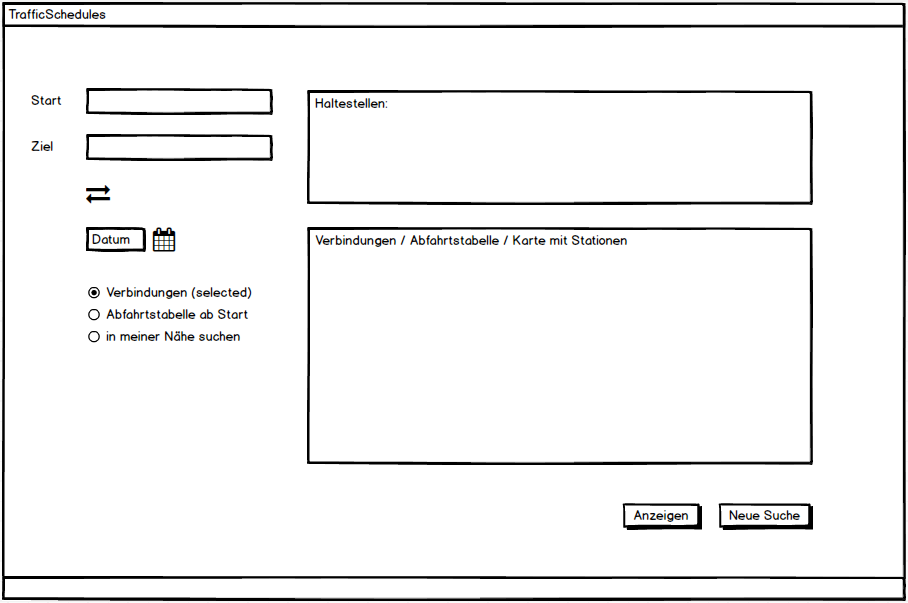
Der Vorentwurf der Benutzeroberfläche:

Abbildung : GUI Mockup

Man sieht eine Ähnlichkeit mit dem fertigen Programm, aber einige Elemente wurden geändert oder hinzugefügt. So ist die Karte als zusätzliches Element vorhanden und nicht im gleichen Feld wie die Verbindungen. Das Konzept einer ListBox als Vorschlagsliste für die beiden Eingabefelder habe ich beibehalten. Der RadioButton für in der Nähe suchen ist weggefallen, da die Karte nun direkt angezeigt wird und nicht erst nach dem Knopfdruck.

Das Use Case Diagramm, welches die Interaktionen des Benutzers mit dem System erläutert:

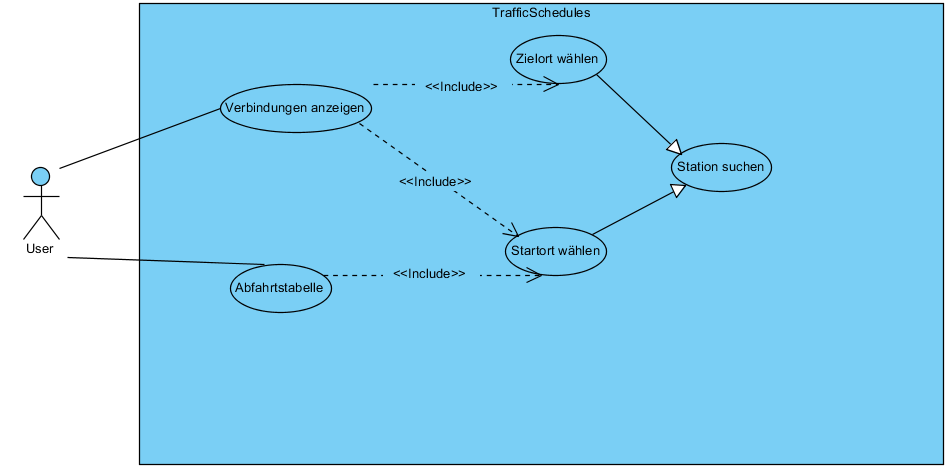


Abbildung : Use Case Diagramm (nur für die Anforderungen 1-3 mit der obersten Priorität)

Das Aktivitätsdiagramm, das einen typischen Programmablauf zeigt:

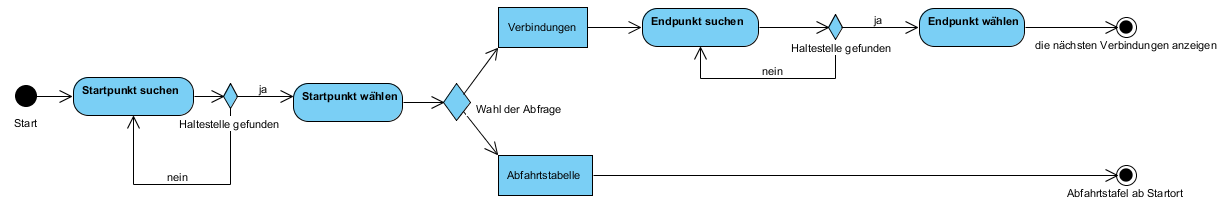


Abbildung : Aktivitätsdiagramm (nur für die Anforderungen 1-3 mit der obersten Priorität)

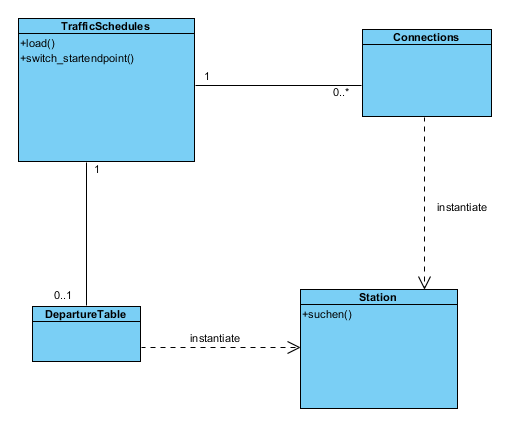
Das Klassendiagramm wie die Struktur aussehen könnte:

Abbildung : Klassendiagramm

Wie man im code-behind unschwer erkennen kann, habe ich das Konzept mit mehreren selbst erstellten Klassen wieder verworfen. Da die meisten Methoden auf Elemente aus dem Form zugreifen, machte es aus meiner Sicht keinen Sinn etwas in eine zweite Klasse auszulagern. Die Klassen Station(s) und Connection(s) wurden von der SwissTransport API bereits zur Verfügung gestellt. Die DepartureTable Klasse war nicht nötig, da die Funktionalität von einer Methode erfüllt wird. Ich habe allerdings versucht, möglichst viele kleinere Aufgaben in eigene Methoden auszulagern. Es ist allerdings eine neue Klasse hinzugekommen, da ich eine zweite Form-Klasse brauchte für den Abfragedialog für die E-Mail Adresse.

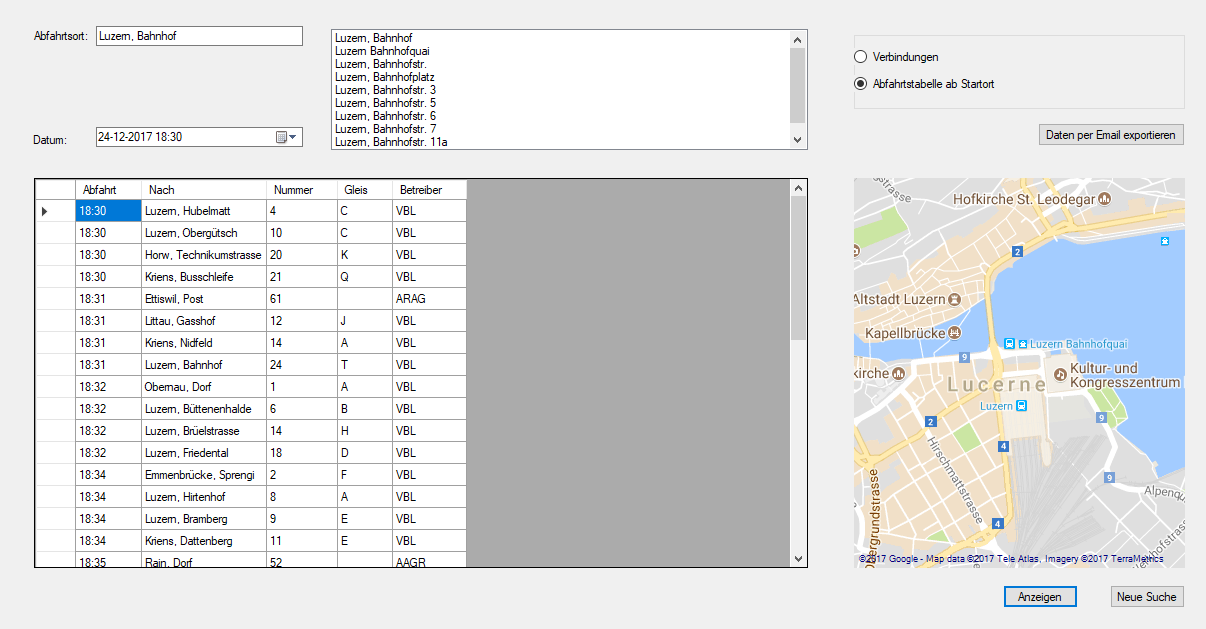


Abbildung : Screenshot der fertigen Applikation

# Testfälle

|  |  |  |
| --- | --- | --- |
| Test | Erwartetes Resultat | Tatsächliches Resultat |
| Station eingeben | Station wird im Textfeld angezeigt | Stationsname wird in der Listbox vorgeschlagen oder der genaue Userinput in der Textbox angezeigt |
| Station suchen | Station wird eingeblendet und eventuell gibt es noch weitere Vorschläge | Vorschläge mit Stationen mit ähnlichem Namen werden in der Listbox angezeigt |
| Datum eingeben in nächster Zeit | Datum wird im DateTimePicker übernommen | Datum wird übernommen, auch wenn das Jahr unrealistisch ist (z.B. 6666). Es werden natürlich keine Verbindungen angezeigt. |
| Datum eingebend, das in 2 Jahren ist | Datum wird im Fahrplan übernommen | Die Daten von OpenData reichen nur für das aktuelle und das letzte Fahrplanjahr. Daher werden keine Verbindungen gefunden, wenn das Datum darüber hinausreicht. |
| Zeit eingeben | Die Zeit wird bei der Abfrage übernommen. | Die Zeit wird bei der Abfrage übernommen. |
| Stationen vertauschen | Start und Zielhaltestelle werden vertauscht. | Start und Zielhaltestelle werden umgetauscht und aktualisierte Verbindungen angezeigt. |
| Verbindungen anzeigen lassen | Die nächsten verfügbaren Verbindungen werden angezeigt. | Die nächsten 4 Verbindungen werden angezeigt. |
| Abfahrtstabelle anzeigen lassen | Eine Abfahrtstabelle mit den nächsten Verbindungen ab dieser Station wird eingeblendet. | Standardmässig werden die nächsten 40 Abfahrten angezeigt. |
| korrekte Anzeige der Karte | Die Karte wird mit einem brauchbaren Zoom und mit der richtigen Haltestelle eingeblendet. | Die Karte der Abfahrtsstation wird angezeigt. Wenn nur der Ort – oder Stadtname gewählt ist, wird die Karte des Ortes angezeigt. Manchmal wird die Karte langsam aktualisiert. |
| Daten im DataGridView | Daten sind leserlich. Daten sind korrekt. Daten können nicht aus Versehen verändert werden. | Daten sind extra für Lesbarkeit formatiert und anscheinend korrekt. Das DataGridView ist auf read-only eingestellt. |
| Daten per E-Mail zuschicken lassen | Ich erhalte als Email den Inhalt des DataGridViews (Verbindungen oder Abfahrtstabelle) | Das Verbindungen/Abfahrten aus dem DataGridView werden als HTML-Tabelle formatiert verschickt. |
| Per Knopfdruck eine neue Suche starten | Das Formular wird auf den Ursprungszustand zurückgesetzt. | Alle Eingabeelemente werden geleert und der DateTimePicker auf die aktuelle Zeit und das aktuelle Datum eingestellt. |
| Mit Tabulator navigieren | Ich kann mit Tab zum nächsten Element navigieren. Dabei wird eine sinnvolle Reihenfolge eingehalten. | Da die Listbox Vorschläge für beide Textboxen bietet, je nachdem, welche aktiv ist, wird mit Tab erst durch die Textboxen und den DateTimePicker navigiert, damit man dort als Nächstes die gewünschte Station auswählen kann. |
| Eine Zeit einstellen, die es nicht gibt | Es kann nur eine Zeit innerhalb des 24h Systems eingegeben werden. | Zeiten ausserhalb des gültigen Formats können nicht eingegeben werden. |
| Ein Datum einstellen, dass es nicht gibt | Das Datum muss es auch tatsächlich geben. | Daten wie der 31. Juni können nicht gewählt werden, aber Jahrgänge die sehr weit zurück oder voraus liegen. |
| Eine Station eingeben, die es nicht gibt | Es werden keine Ergebnisse gefunden. | Es wird eine Fehlermeldung ausgegeben, dass keine Verbindungen gefunden wurden. |
| Fenstergrösse verändern | Die Elemente bleiben sinnvoll angeordnet. | Die Elemente orientieren sich gut bei Vergrösserung des Fensters. Bei Verkleinerung wird alles zu einer unschönen Masse von Elementen zusammengestaucht. |
| Eine unvollständige Abfrage machen | Die Abfrage wird nur ausgeführt, wenn alle nötigen Daten angegeben wurden. | Es erscheint eine Meldung, dass ungültige Stationen eingegeben wurden. |
| Eine Abfrage für ein schon vergangenes Datum machen | Es werden entweder die alten Verbindungen angezeigt oder keine gefunden. | Solange das Datum im Fahrplanjahr liegt, werden die Verbindungen angezeigt. |
| Eine private Adresse als Start und/oder Ziel einstellen | Es wird keine Verbindung gefunden oder Verbindungen mit Fussweg angezeigt. | Eine Meldung warnt, dass eine ungültige Adresse eingegeben wurde. |
| Station nur unvollständig eingeben und nicht per Klick vervollständigen und die Abfrage starten. | Es erscheint eine Fehlermeldung. | Der Stationsname wird automatisch mit dem nächstgelegenen Vorschlag komplettiert. |

# Installationsanleitung

Die TrafficSchedules.exe aus dem Application Ordner kann direkt ausgeführt werden, um das Programm zu starten.  
Voraussetzung dafür ist allerdings, dass das .NET Framework installiert ist (bei neueren Windows Versionen vorinstalliert).

Alternativ kann aus der Projekt Solution im Visual Studio unter Build -> Publish TrafficSchedules ein ausführbares Programm erstellt werden oder im Verzeichnis von TrafficSchedules unter bin/debug die TrafficSchedules.exe ausgeführt werden.

# Quellen

Karten bereitgestellt von Google und GMap.NET

Fahrplandaten von [opentransportdata.swiss](https://opentransportdata.swiss/de/dataset/gtfsrt)