2019

David Gataric

CSS Versicherung

Ennetbürgerstrasse 4

6374 Buochs

2.5.2019

Concheck ihr ÖV-Ratgeber



Inhalt

[Einführung 2](#_Toc7788984)

[Zweck des Dokuments 2](#_Toc7788985)

[Was wurde umgesetzt? 2](#_Toc7788986)

[A001 und A004 2](#_Toc7788987)

[A002 und A005 2](#_Toc7788988)

[A003 3](#_Toc7788989)

[A006 4](#_Toc7788990)

[Route sehen 4](#_Toc7788991)

[Gestellte Anforderungen 5](#_Toc7788992)

[Funktionalitäten die fehlen 6](#_Toc7788993)

[Bekannte Bugs 6](#_Toc7788994)

[Use Case 6](#_Toc7788995)

[Aktivitätendiagramme 7](#_Toc7788996)

[Mockups 8](#_Toc7788997)

[Testfälle 10](#_Toc7788998)

[Installation Guide 11](#_Toc7788999)

[Deinstallation Guide 13](#_Toc7789000)

# Einführung

Im ÜK M318 lernten wir das Objektorientierte Programmieren. Um uns zu prüfen, wurde uns der Auftrag erteilt, eine App zu erstellen. Diese App funktioniert eigentlich wie die SBB App. Durch die API von Opendata kamen wir an unsere Daten. Ziel des Programmes war es, eine Abfahrtstafel anzuzeigen, einige Verbindungen von Station zu Station zu zeigen und natürlich noch mehr, aber zu diesen Anforderungen kommen wir im Verlauf der Dokumentation.

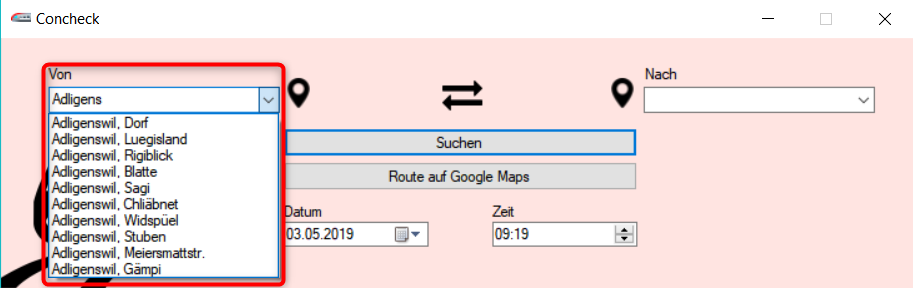
# Zweck des Dokuments

Dieses Dokument soll den Werdegang der App aufzeigen. Zudem soll dieses Dokument offene Fragen zur App beantworten. Zusätzlich sind hier alle Test Cases enthalten, mit denen die App getestet wurde.

# Was wurde umgesetzt?

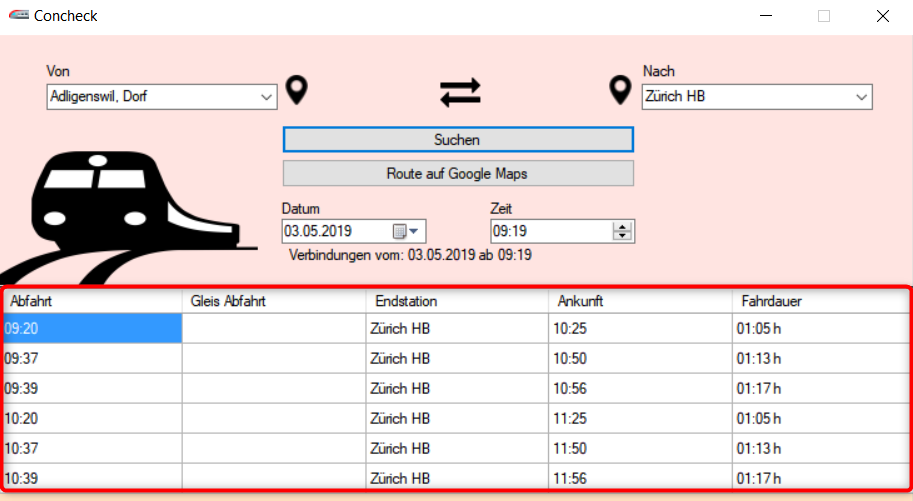
## A001 und A004

Da die Anforderungen 1 und 4 beide etwas mit der Textsuche zu tun hatten, nahm ich diese Anforderungen gleich zusammen. Dafür habe ich Comboboxen benutzt. Anfangs benutzte ich die Autocompletion Eigenschaft der Combobox, jedoch erwies sich dies als nicht performant, da ich nicht alle Stationen beim Appstart in die Combobox laden konnte. Deshalb machte ich dies mit der Liste selbst. Die Combobox List wird beim KeyUp Event geleert und danach werden die Stationsnamen, welche dem Text Input gleichen in die Liste gepackt.



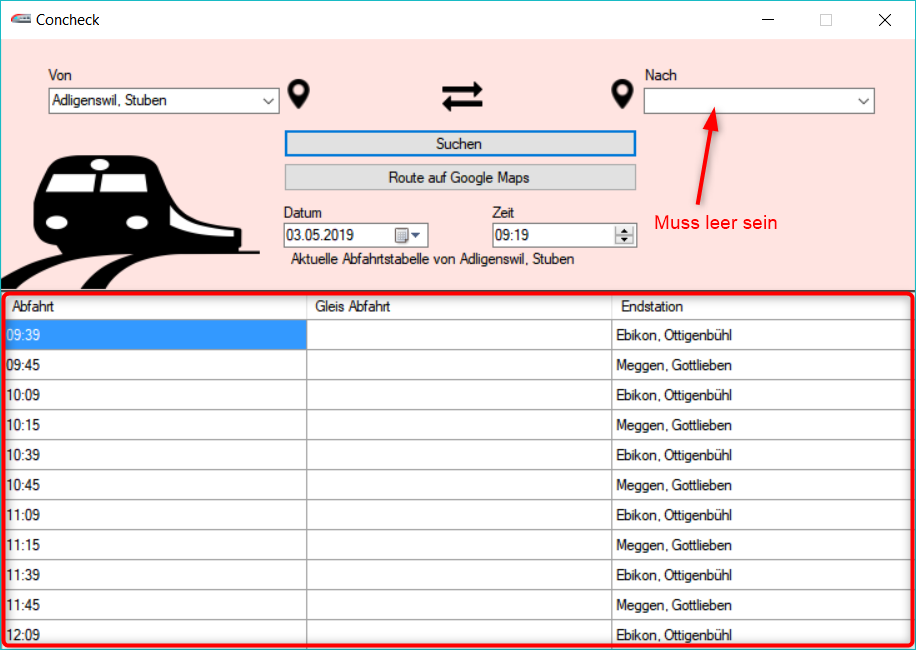
## A002 und A005

Die Zweite Anforderung war, die Verbindungen zwischen den beiden Stationen anzuzeigen.  
Dazu habe ich eine DataGridView erstellt, in welcher die nächsten 6 Verbindungen angezeigt werden.  
Zuerst wurden standardmässig durch die API nur 4 Verbindungen gezeigt, jedoch änderte ich das Limit auf 6 Verbindungen. Da die fünfte Anforderung auch zu der zweiten passte, kombinierte ich auch diese zwei Anforderungen. Ich erstellte also zwei DateTimePicker und erweiterte den API Aufruf. Ich musste nicht viel ändern ausser den Parametern und dem Rest-Aufruf. Nachdem dies alles gemacht war funktionierte diese Funktion. Zur Übersicht habe ich auch noch ein Label gemacht, welches dem User anzeigt, wann die Verbindungen sind. Um Erwartungskonform zu sein,  
habe ich wie die SBB App die Suchumkehrung Funktion eingebaut. Diese wird mit einem Mausklick auf das Pfeilbild getriggert.



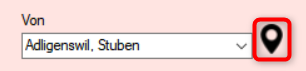
## A003

Bei der dritten Anforderung gab es einige Wege, die zum Ziel führen. Ich habe mich dazu entschieden  
die Anzeigetafel in einem neuen DataGridView anzuzeigen. Je nach Benutzung der App wird die Abfahrtstafel eingeblendet und die Verbindungstafel ausgeblendet. Um nach einer Abfahrtstafel zu suchen, muss man nur die Von Verbindung eintragen. Wenn man dann auf Suchen drückt, dann werden die nächsten Abfahrten im DataGridView angezeigt.



## A006

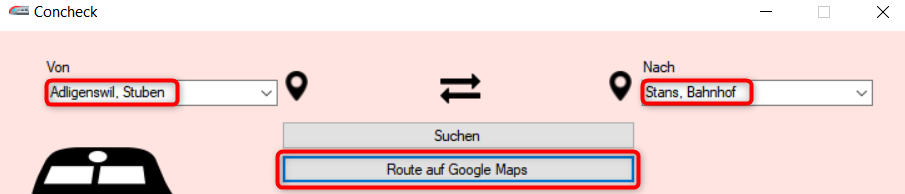
Wenn der User nicht wüsste, wie es bei einer der 2 Stationen aussehen würde, wäre dies nicht so gut. Deshalb habe ich die Anforderung 6 verwirklicht, indem ich eine Google Maps search Funktion eingebaut habe. Nun kann der Benutzer entweder die Von- oder die Nach Station auf Google Maps suchen. Es wird lediglich ein Google Maps Link konstruiert, bei dem auch noch der Text aus der Combobox hinzugefügt wird. Nachdem man auf die Picturebox drückt, wird der Standardbrowser aufgemacht und die Google Maps Seite der Station wird aufgemacht.

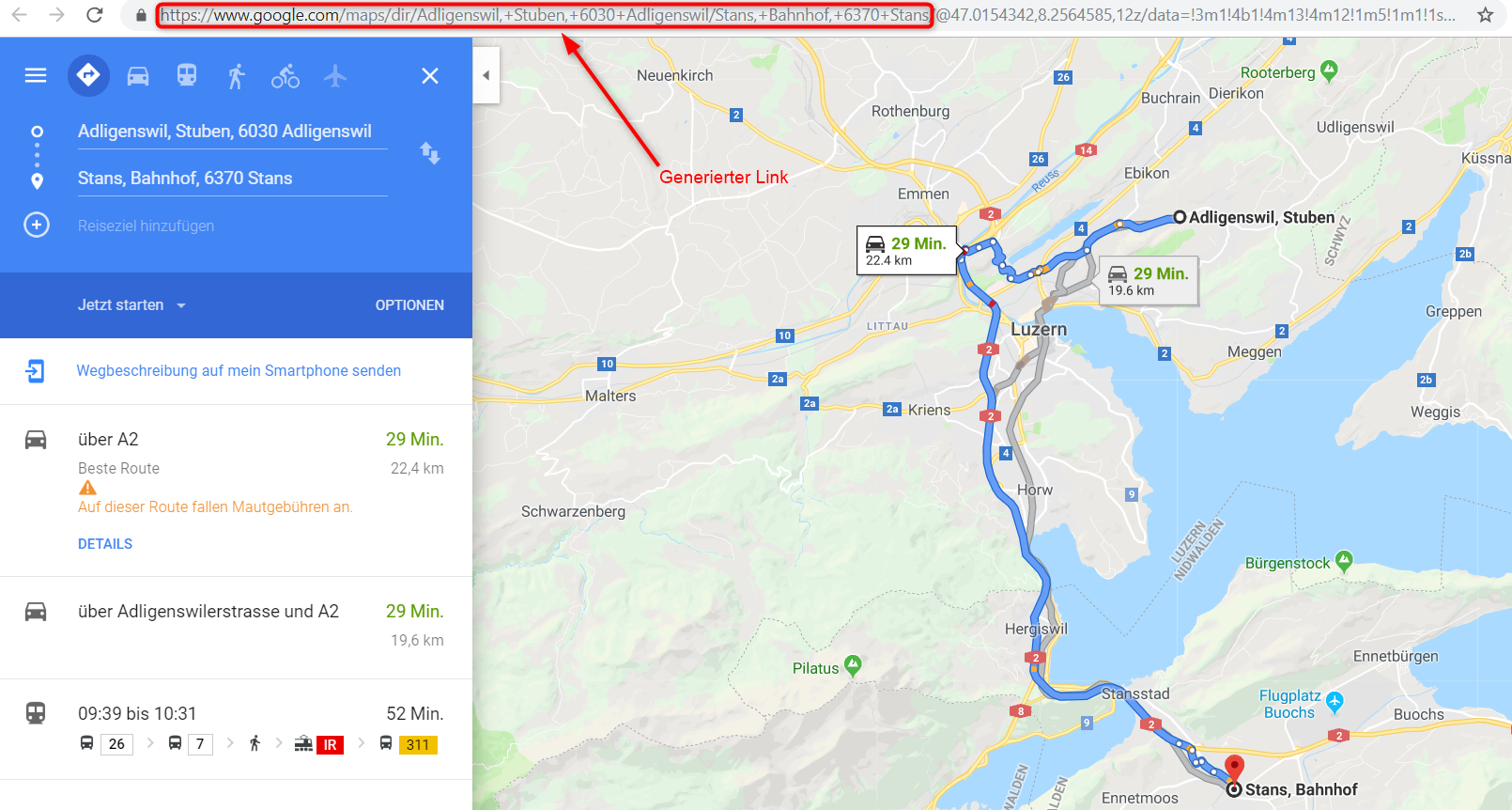




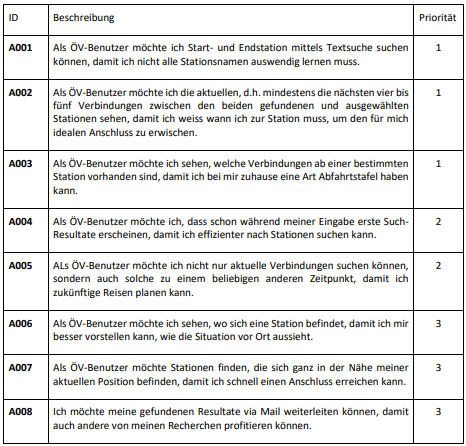
## Route sehen

Damit ich eine Route auch gleich sehen kann, habe ich eine Funktion eingebaut, welche es mir ermöglicht, die Route zwischen den zwei eingegebenen Stationen zu sehen. Dazu müssen beide Comboboxen gefüllt sein und der Route auf Google Maps Button gedrückt werden. Danach wird man auf den Google Maps Routenplaner navigiert.





# Gestellte Anforderungen

Hier sieht man alle Anforderungen, welche man machen musste, sowie konnte.

# Funktionalitäten die fehlen

Da mir die Zeit sowie das Know-How fehlten, konnte ich die siebte Anforderung nicht erledigen.  
Ich wusste nicht wie ich meine jetzige Position ausfindig machen konnte. Obwohl ich dies später im Internet gefunden hatte, wusste ich trotzdem nicht, wie ich nun damit die nächstliegende Haltestelle finden konnte. Alle anderen Anforderungen konnte ich erledigen.

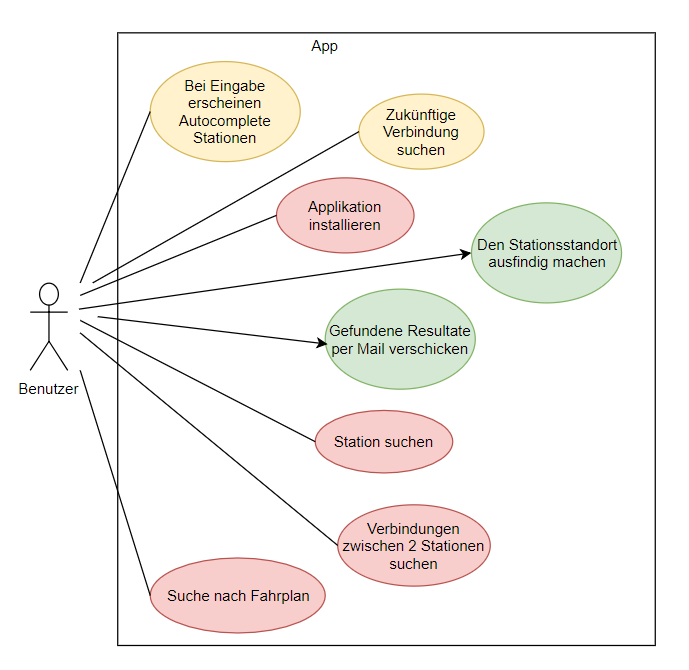
Eine Anforderung konnte ich so halb lösen. Es geht hier um die Mail Versendung. Ich konnte keine gute Variante verwirklichen ausser, dass nur Outlook und Gmail User eine Mail verschicken können.

# Bekannte Bugs

Die mir bekannten Bugs sind nun hier aufgelistet.  
- Das Autocomplete ist nicht so schnell

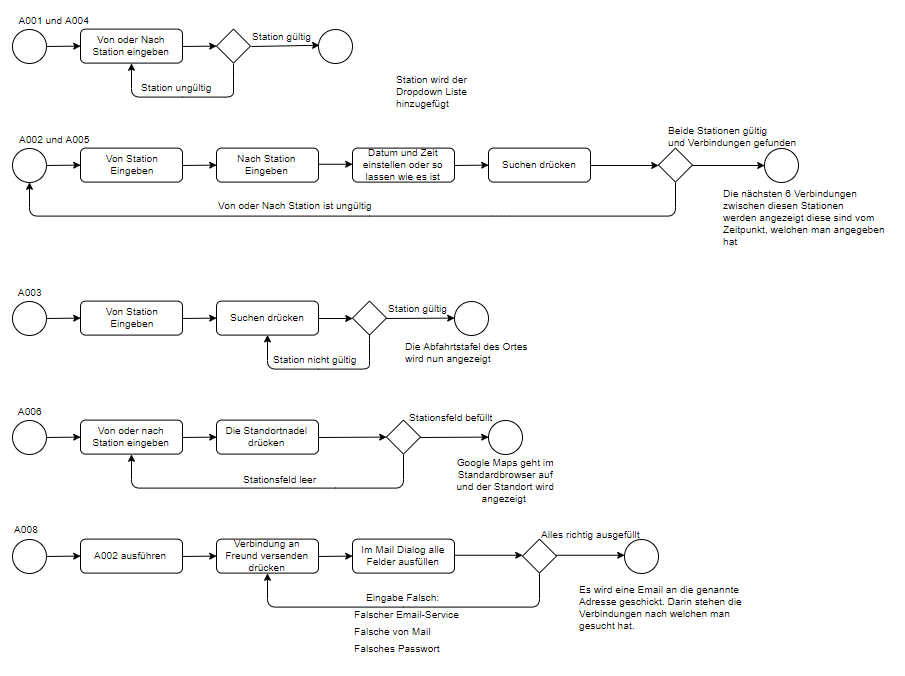
# Use Case

Mit dem Use Case Diagramm soll aufgezeigt werden, was der User alles machen kann. Ich habe die Tasks 1 bis 5 in einem Use Case festgehalten.



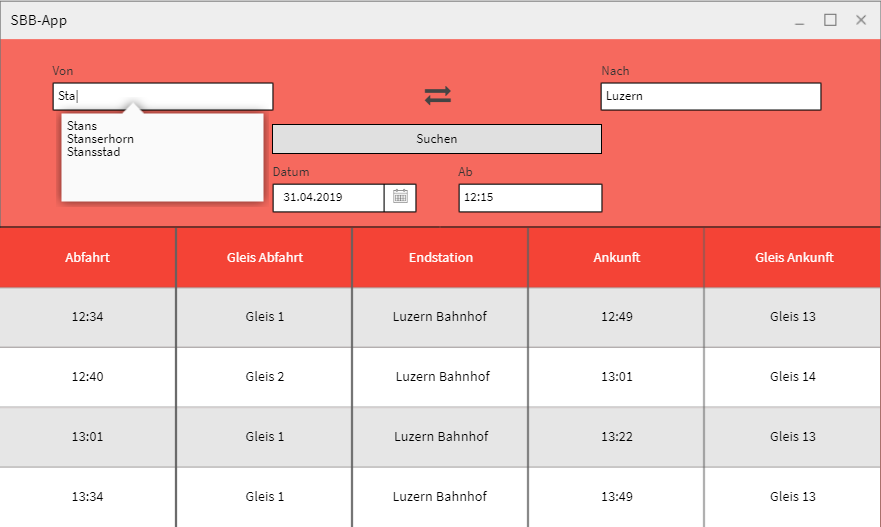
# Aktivitätendiagramme

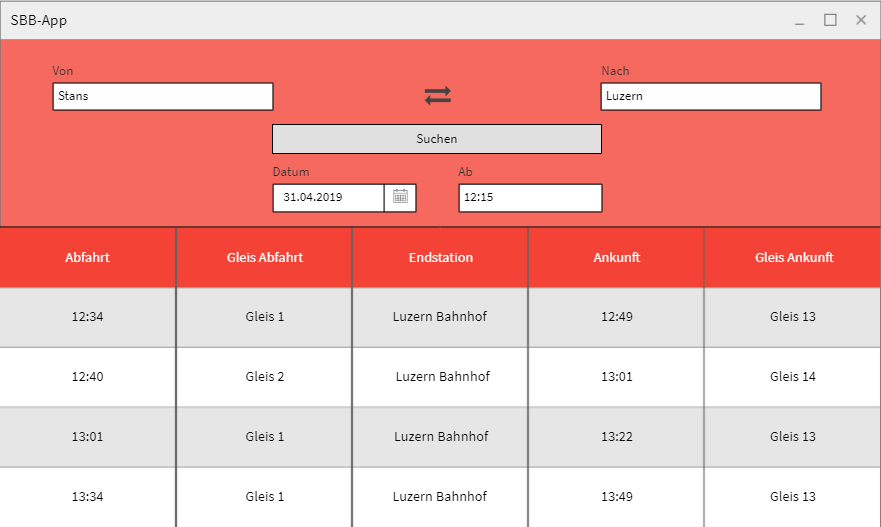
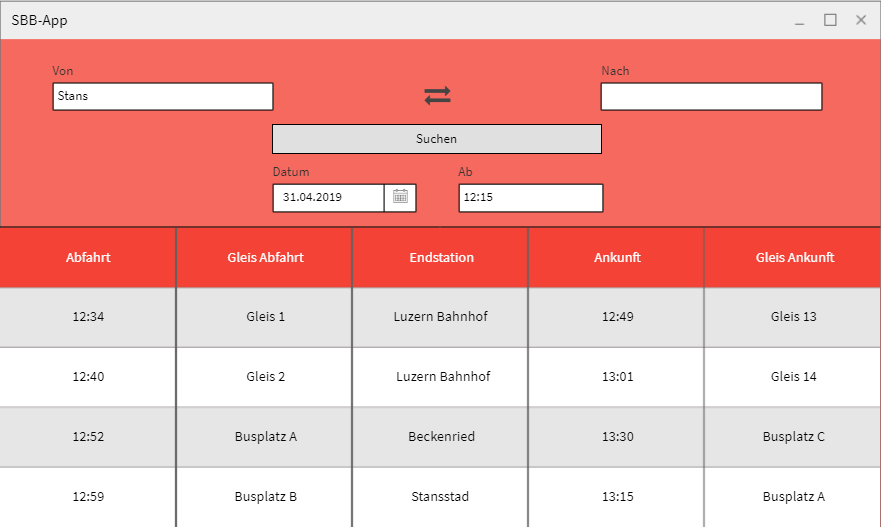
Die Aktivitätendiagramme sollen die Logik simpel aufzeigen. Das Ziel ist es, für jeden Task aufzuzeigen, was bei einem Fall passieren kann.



# Mockups

Die Mockups wurden erstellt, um mir einen Überblick über das Appdesign zu erschaffen. Für jede Anforderung habe ich jeweils ein Mockup erstellt.





# Testfälle

|  |  |
| --- | --- |
| Given | User will nach Luzern, aber er kennt die genaue Endstation nicht. |
| When | User trägt Von Station ein und beginnt mit dem tippen von der Nach Station. |
| Then | Während dem tippen wird die Combobox aufgeklappt und der User sieht alle Stationen, welche seiner Eingabe gleichen. |

Anmerkung: Der Test verläuft erfolgreich, denn der Autocomplete zeigt mir alle Stationen, welche meiner Eingabe gleichen. z.B Luzern, -> Luzern, Bahnhof oder Luzern, Schwanenplatz

|  |  |
| --- | --- |
| Given | User möchte von Basel nach Zürich reisen, er will die nächsten Verbindungen sehen. |
| When | Der User gibt die Von Station sowie die Nach Station ein und drückt danach auf Suchen. |
| Then | Die DataGridView connections bekommt die Daten von den nächsten 6 Verbindungen und der User sieht diese. |

Anmerkung: Dieser Test verlief erfolgreich, nach dem Suchen bekomme ich genau die nächsten 6 Verbindungen ab dem jetzigen Zeitpunkt.

|  |  |
| --- | --- |
| Given | User möchte die Abfahrtstafel von Genf sehen |
| When | Der User gibt die Von Station ein |
| Then | Die DataGridView board bekommt die Daten von den nächsten Abfahrten und der User sieht diese. |

Anmerkung: Dieser Test verlief erfolgreich, nach dem Suchen bekomme ich die nächsten Abfahrten ab dem jetzigen Zeitpunkt.

|  |  |
| --- | --- |
| Given | Der User möchte in 3 Monaten um 12:00 Uhr von Luzern nach Zürich reisen |
| When | Der User gibt die Von Station sowie die Nach Station danach gibt er das Datum im DateTimePicker und die Zeit im TimePicker ein. Schlussendlich drückt er auf suchen. |
| Then | Die DataGridView bekommt die Daten von den nächsten 6 Verbindungen, ab der Eingegebenen Zeit. |

Anmerkung: Der Test war erfolgreich, die Daten werden angezeigt und die Zeiten sind korrekt.

|  |  |
| --- | --- |
| Given | User möchte sehen wo die Station ist. |
| When | Der User seine Station eingibt und danach auf die Mapnadel drückt. |
| Then | Der Standardbrowser öffnet sich und Google Maps zeigt die Eingegebene Station an. |

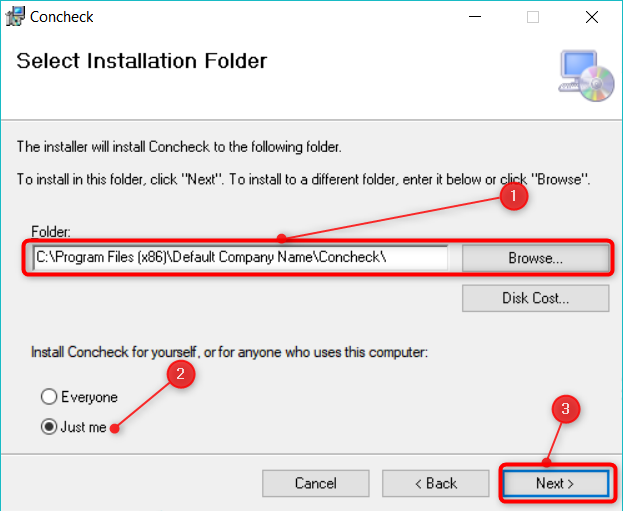
Anmerkung: Test verlief erfolgreich, Google Maps wird geöffnet und die Stationsdaten werden angezeigt.

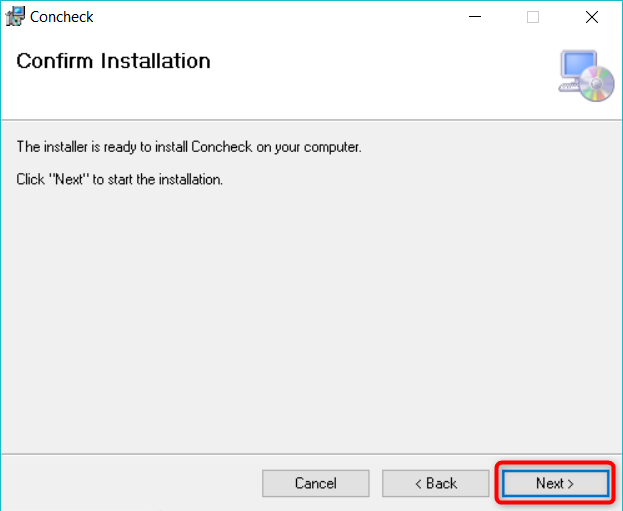
# Installation Guide

Im ConcheckInstallation Ordner hat man zwei Dateien, einerseits Concheck.msi und setup.exe.  
Eine dieser Dateien muss man nun Doppelklicken.

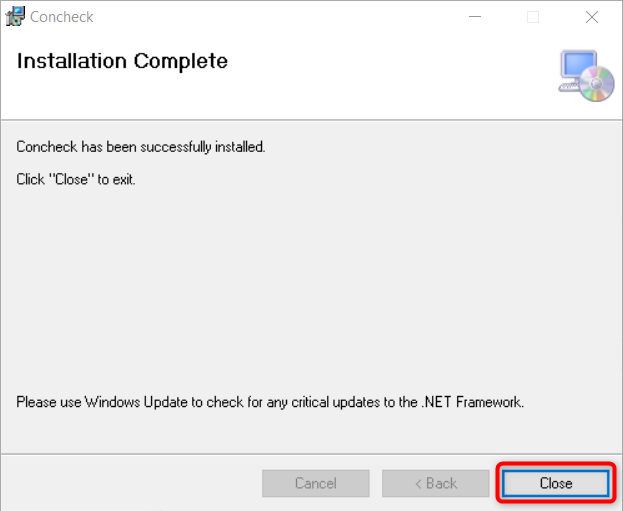
1. Next drücken



2. Speicherort auswählen, Installation nur für mich auswählen und Next drücken

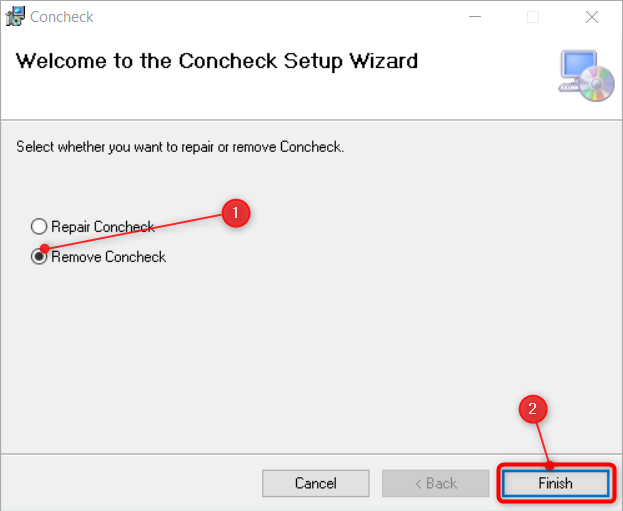
3. Wieder Next drücken

4. Warten bis Installation fertig ist. Falls nun die Frage kommen sollte, ob das Installationstool Änderungen am Gerät machen darf, dann klicke auf Ja.

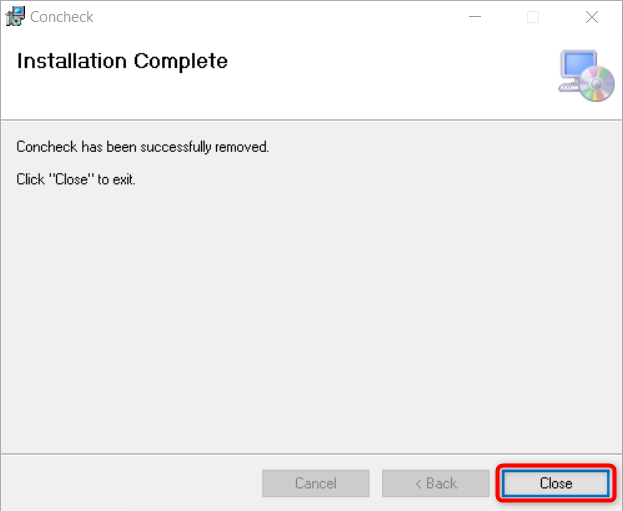
5. Auf Close drücken und Concheck geniessen

# Deinstallation Guide

Falls du leider Concheck deinstallieren möchtest, dann musst du diese Schritte befolgen.  
Du solltest also als erstes die setup.exe oder die Concheck.msi Datei öffnen.

1. Remove Concheck auswählen und Finish drücken

2. Warten bis Deinstallation fertig ist. Falls nun die Frage kommen sollte, ob das Installationstool Änderungen am Gerät machen darf, dann klicke auf Ja.

3. Auf Close drücken