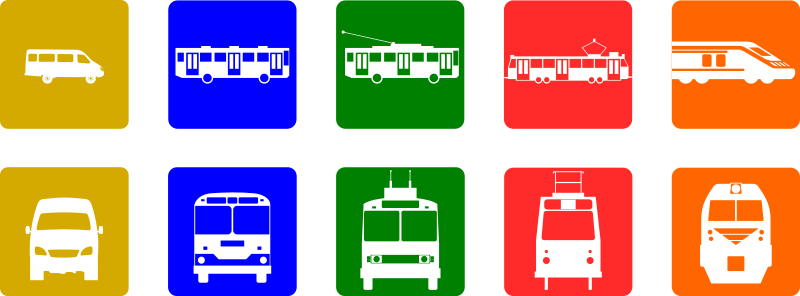
Dokumentation

**2016**

Michelle Meyer

Luzerner Kantonalbank AG

01.06.2016



Inhalt

[Analyse und Design 2](#_Toc452557360)

[Anforderungsanalyse 2](#_Toc452557361)

[Use Cases 2](#_Toc452557362)

[Aktivitätsdiagramm 3](#_Toc452557363)

[Klassendiagramm 4](#_Toc452557364)

[Sequenzdiagramm 4](#_Toc452557365)

[Applikation 5](#_Toc452557366)

[Umsetzung 5](#_Toc452557367)

[Portierbarkeit 6](#_Toc452557368)

[GitHub 6](#_Toc452557369)

[Testing 7](#_Toc452557370)

[A001 / A004 7](#_Toc452557371)

[A002 8](#_Toc452557372)

[A003 8](#_Toc452557373)

[A005 9](#_Toc452557374)

[A006 9](#_Toc452557375)

[A007 9](#_Toc452557376)

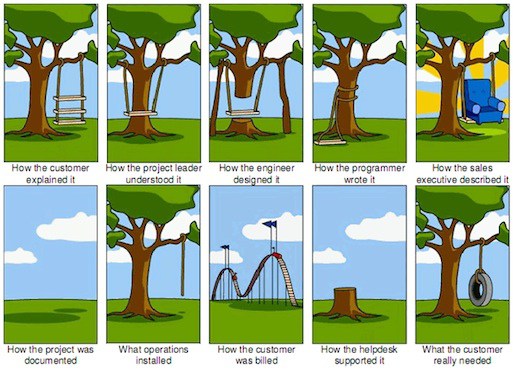
[A008 9](#_Toc452557377)

[Literaturverzeichnis 10](#_Toc452557378)

# Analyse und Design

## Anforderungsanalyse

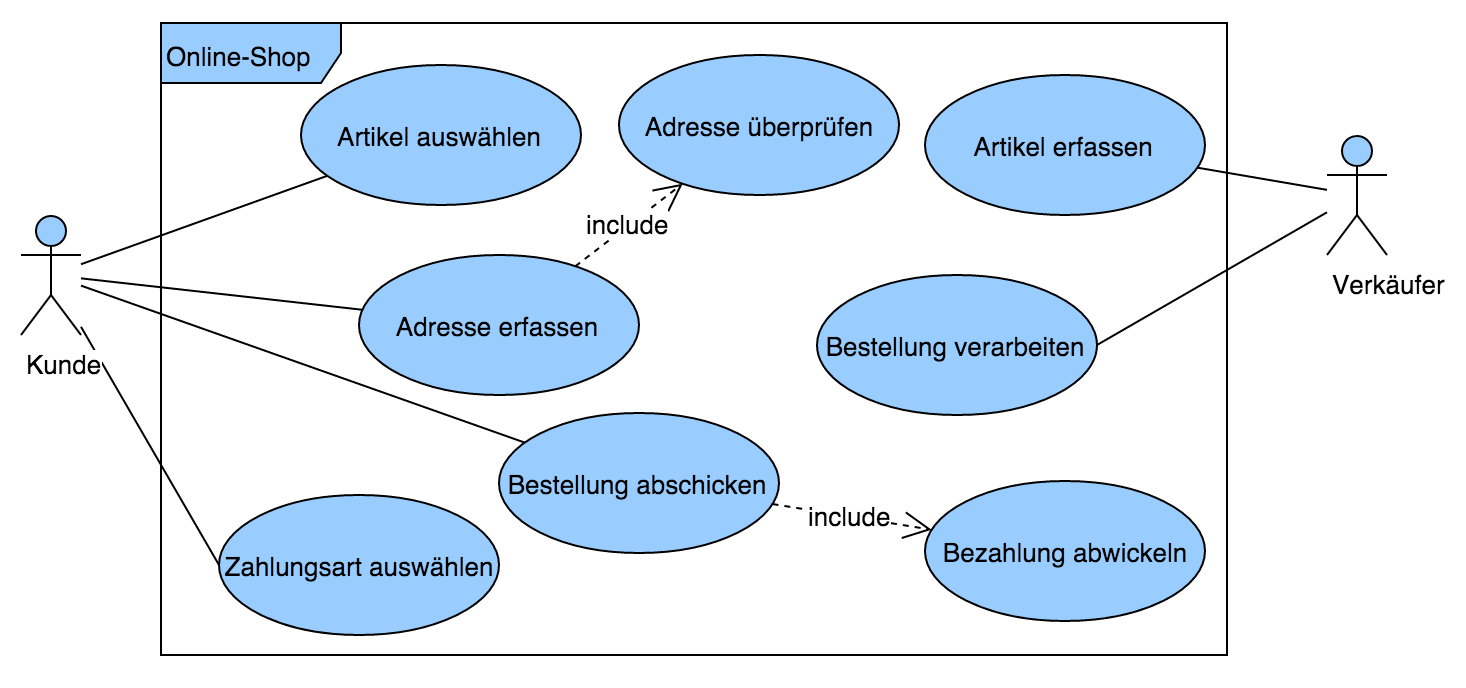
Die Anforderungsanalyse hilft bei dem Entwicklungsprozess einer Software. Es werden die Anforderungen des Kundens festgelegt und dokumentiert, damit immer wieder auf diese zurückgegriffen werden kann. So kann folgende Situation verhindert werden:



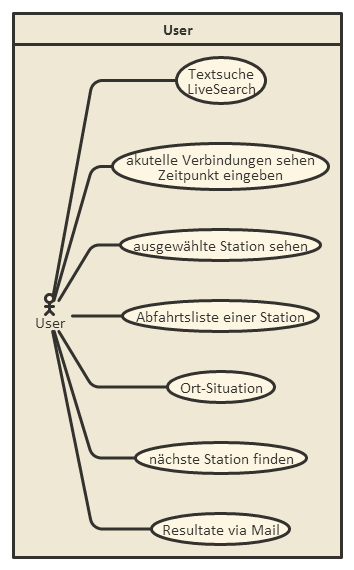
(Ward, 2010)

## Use Cases

Use Cases werden eingesetzt um die verschiedenen Funktionen aufzuzeigen, über die ein Programm verfügen muss. Man hat mindestens einen Actor(Aktor) und beliebig viele use cases(Anwendungsfälle), welche andere use cases includen(verwenden) können.

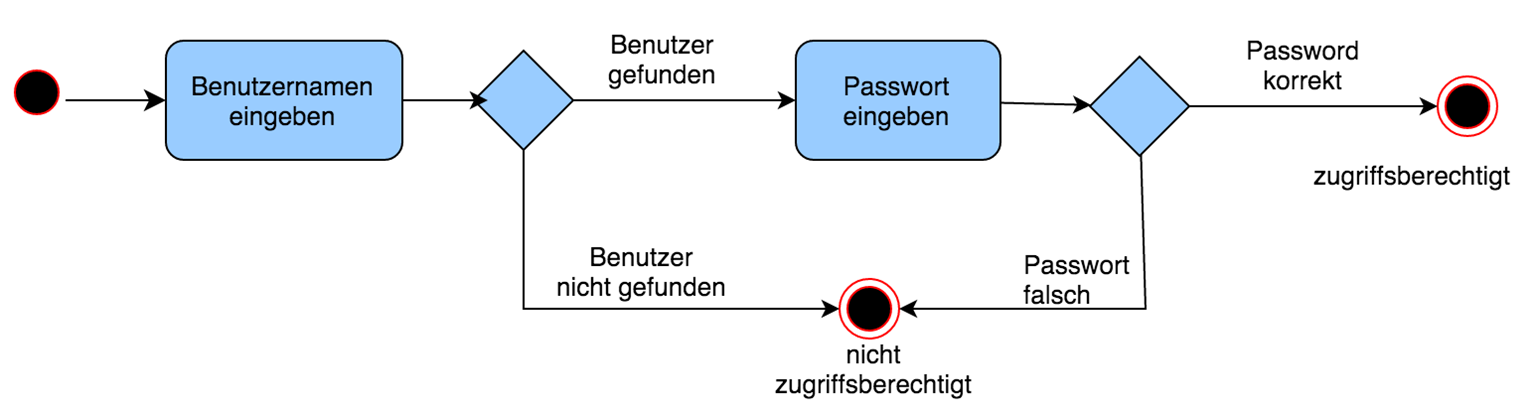


(Analyse und Design.pdf, 2016)

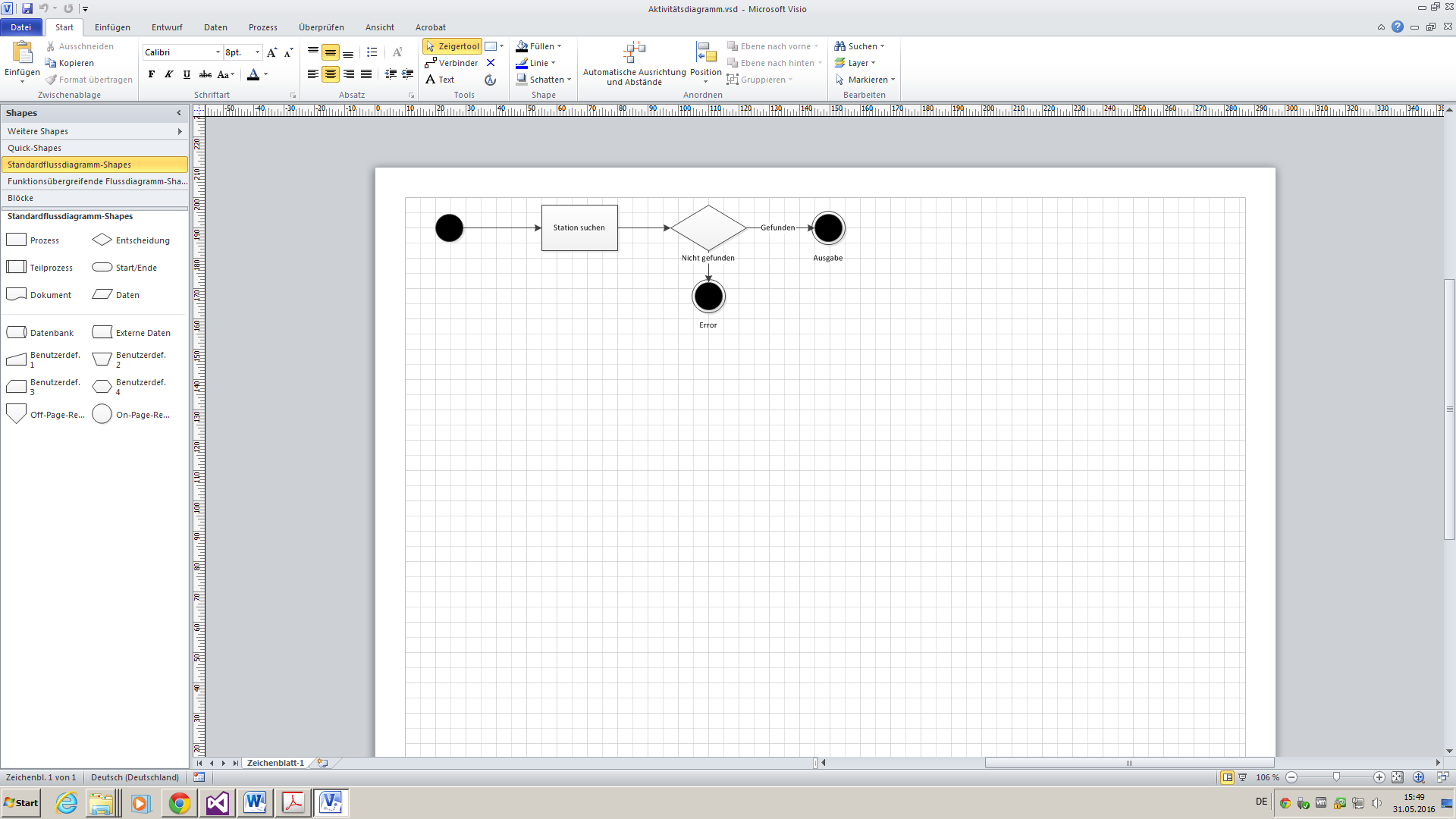
 (Meyer, 2016)

## Aktivitätsdiagramm

Das Aktivitätsdiagramm wird verwendet, um Daten- und Kontrollflüsse grafisch darzustellen. Es muss einen Startpunkt und einen oder mehrere Endpunkte haben. Dazwischen können verschiedene Aktivitäten stattfinden.

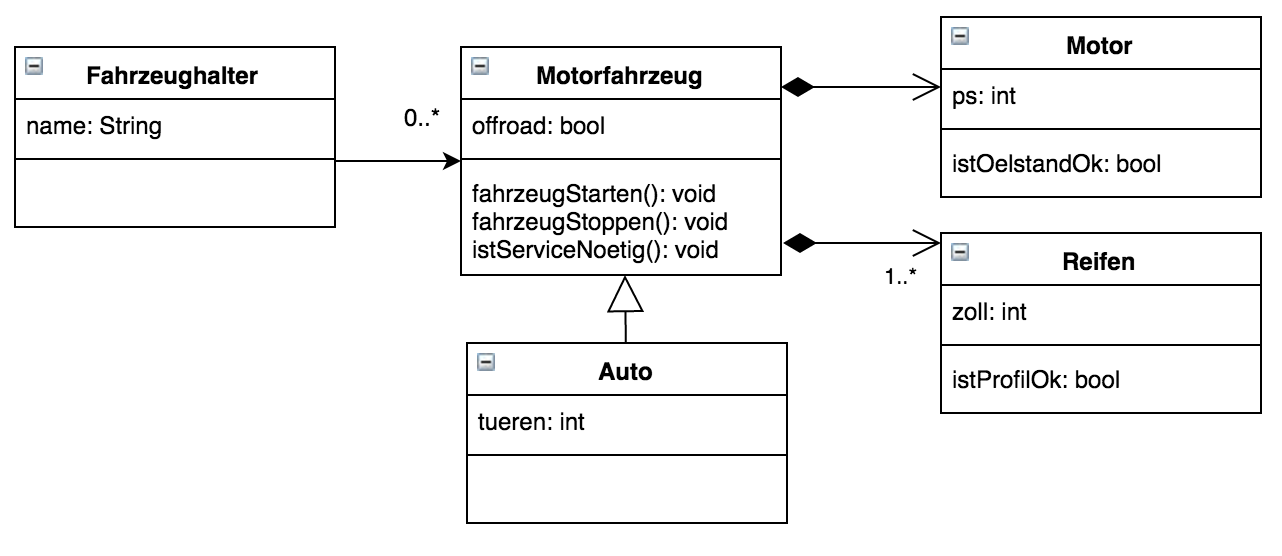


(Analyse und Design.pdf, 2016)

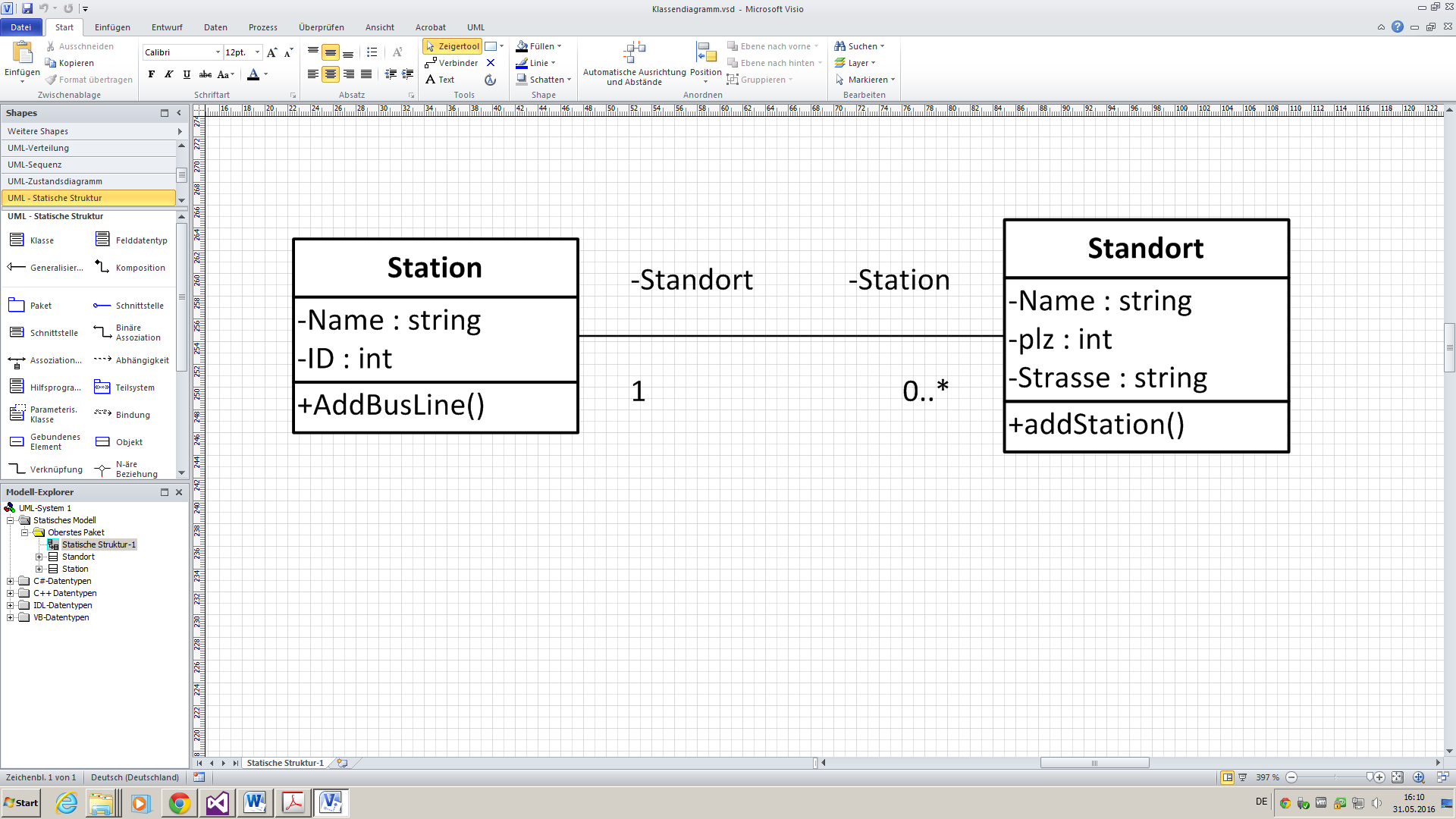
 (Meyer & Poblete, 2016)

## Klassendiagramm

Das Klassendiagramm wird benutzt, zur Darstellung von Klassen und ihrer Beziehungen in der OOP. Es können auch noch mehr Details herausgelesen werden. Behandelt in Modul 226.

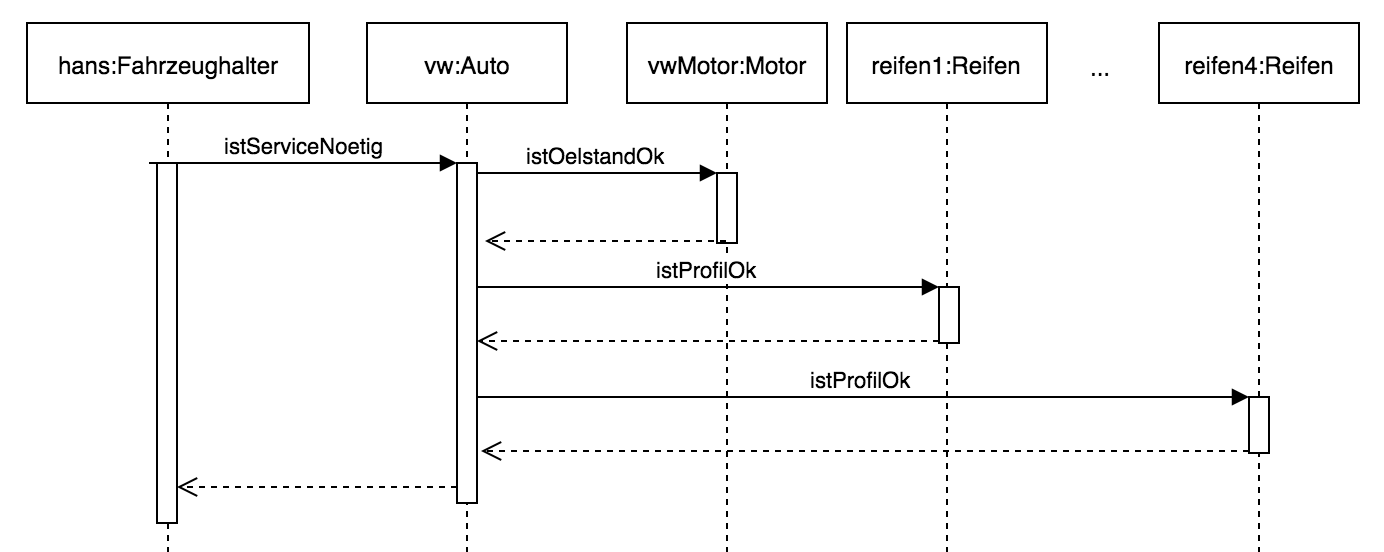


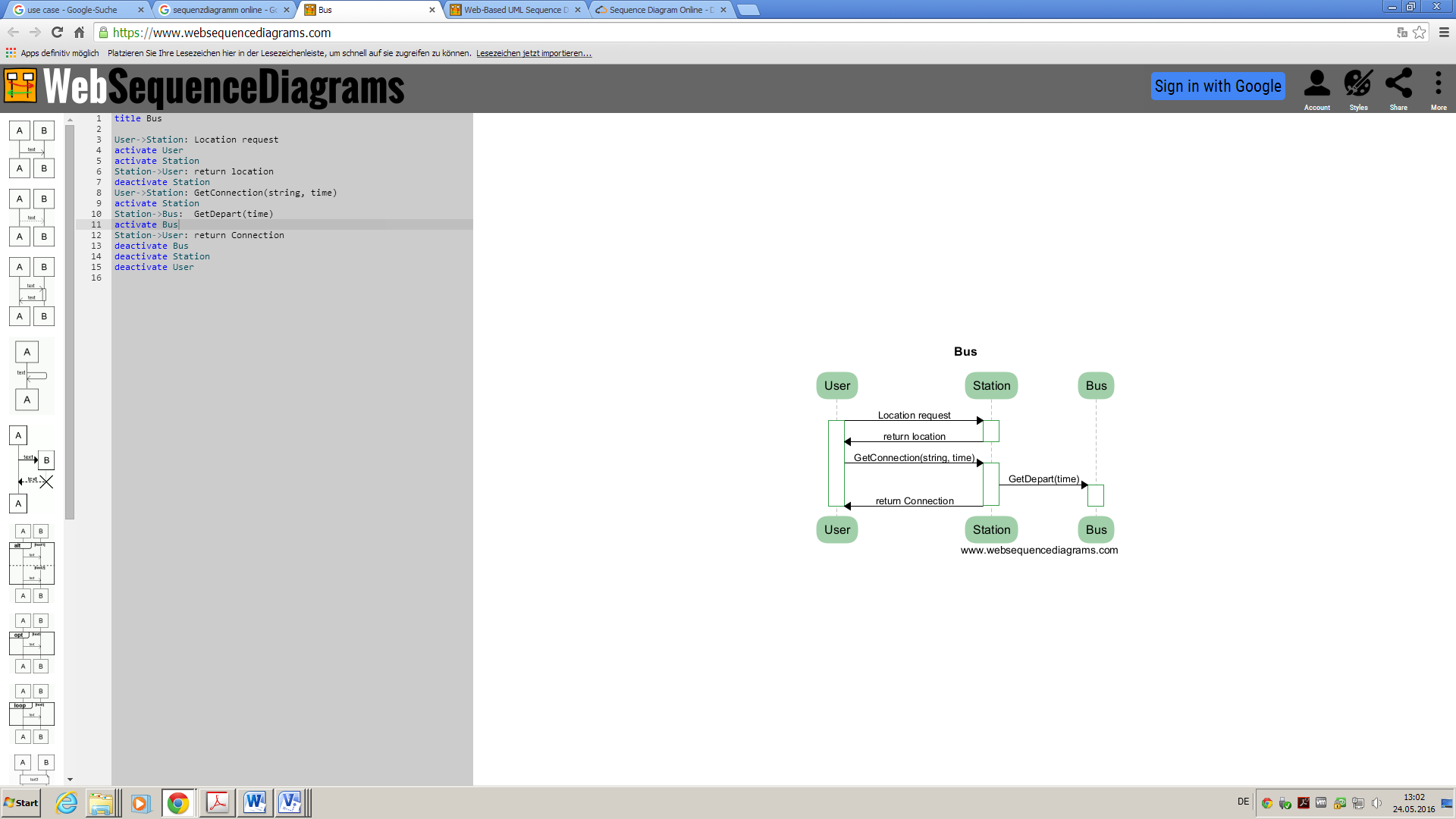
(Analyse und Design.pdf, 2016)

 (Meyer & Poblete, 2016)

## Sequenzdiagramm

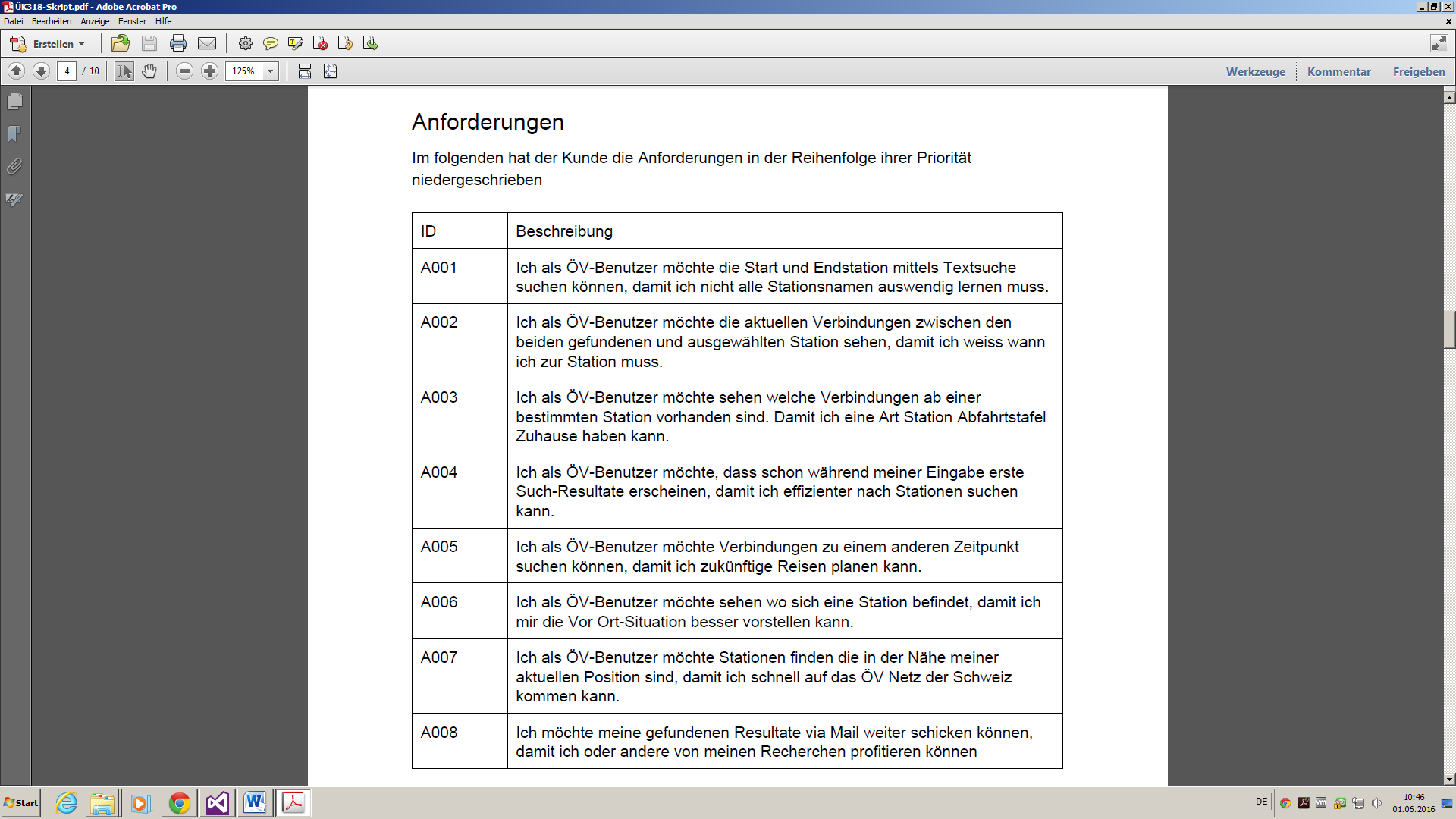
Das Sequenzdiagramm wird in der Informatik benutzt, um die Interaktionen zwischen Klassen aufzuzeigen.



(Meyer, 2016)

# Applikation

## Umsetzung

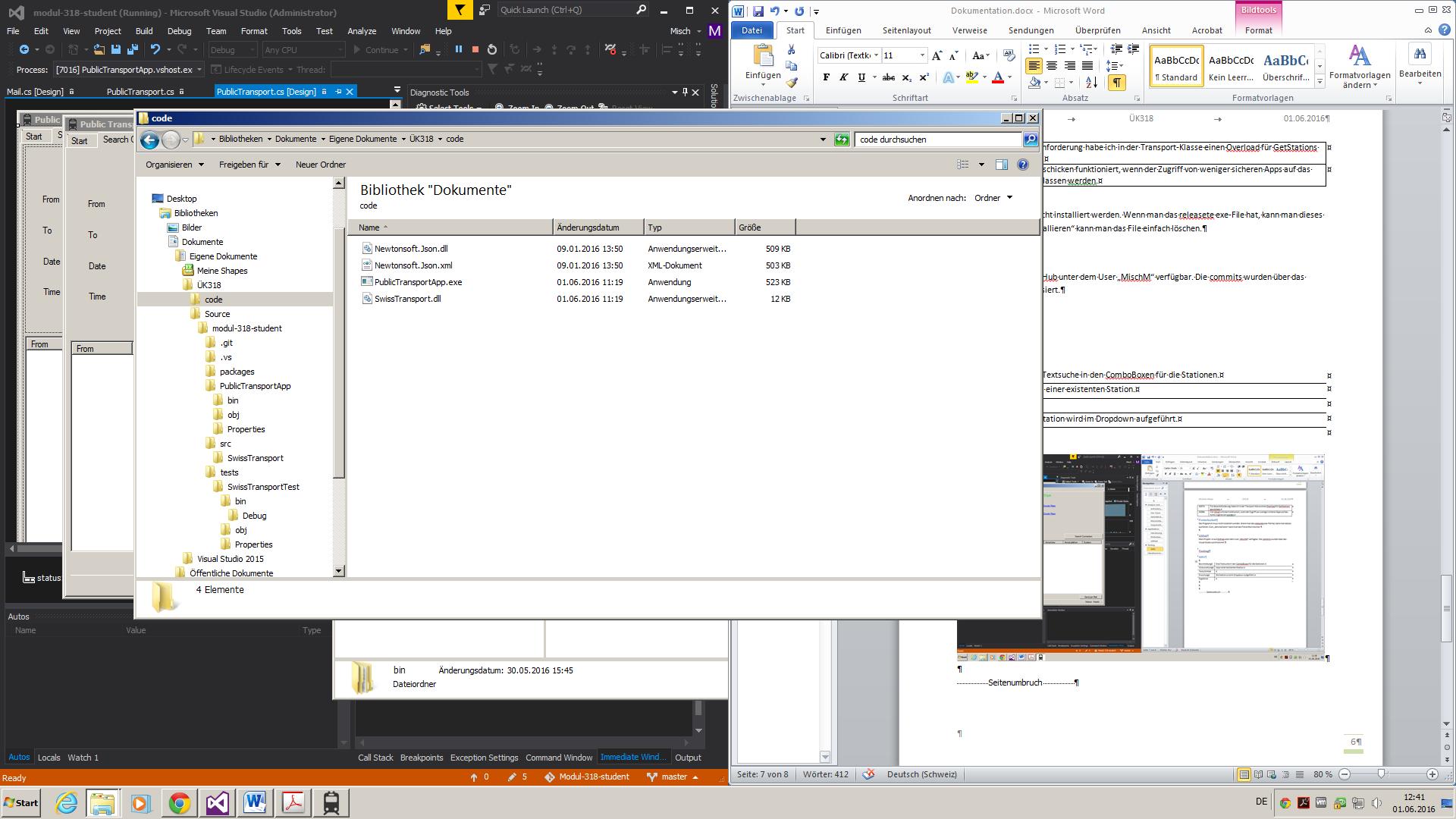
(ÜK318-Skript.pdf, 2016)

Es wurden alle Anforderungen umgesetzt. Nachfolgen einige spezifische Bemerkungen:

|  |  |
| --- | --- |
| A001 | Die Textsuche wird in drei Tabs in den ComboBoxen verwendet. Diese Anforderung wird zusammen mit A004 erfüllt. |
| A002 | Die Verbindungen werden in einer ListView angezeigt. |
| A003 | Die Abfahrtstafel der Stationen kann man im Stationboard-Tab sehen. |
| A004 | Siehe A001. Bei dieser Anforderung hatte ich zusätzliche Hilfe von einem Mitschüler(Lukas Dörig). Dieser riet mir, mit mehreren Threads zu arbeiten, da ich dies aber noch nie gemacht oder behandelt hatte, erklärte er mir wie es funktioniert. |
| A005 | Dafür habe ich eine Anpassung der Methode in der Transport-Klasse gemacht. Zusätzlich kann man zwischen Ankunfts-& Abfartszeit wechseln. |
| A006 |  |
| A007 | Für diese Anforderung habe ich in der Transport-Klasse einen Overload für GetStations geschriben. |
| A008 | Von Gmail schicken funktioniert, wenn der Zugriff von weniger sicheren Apps auf das Konto zugelassen werden. |

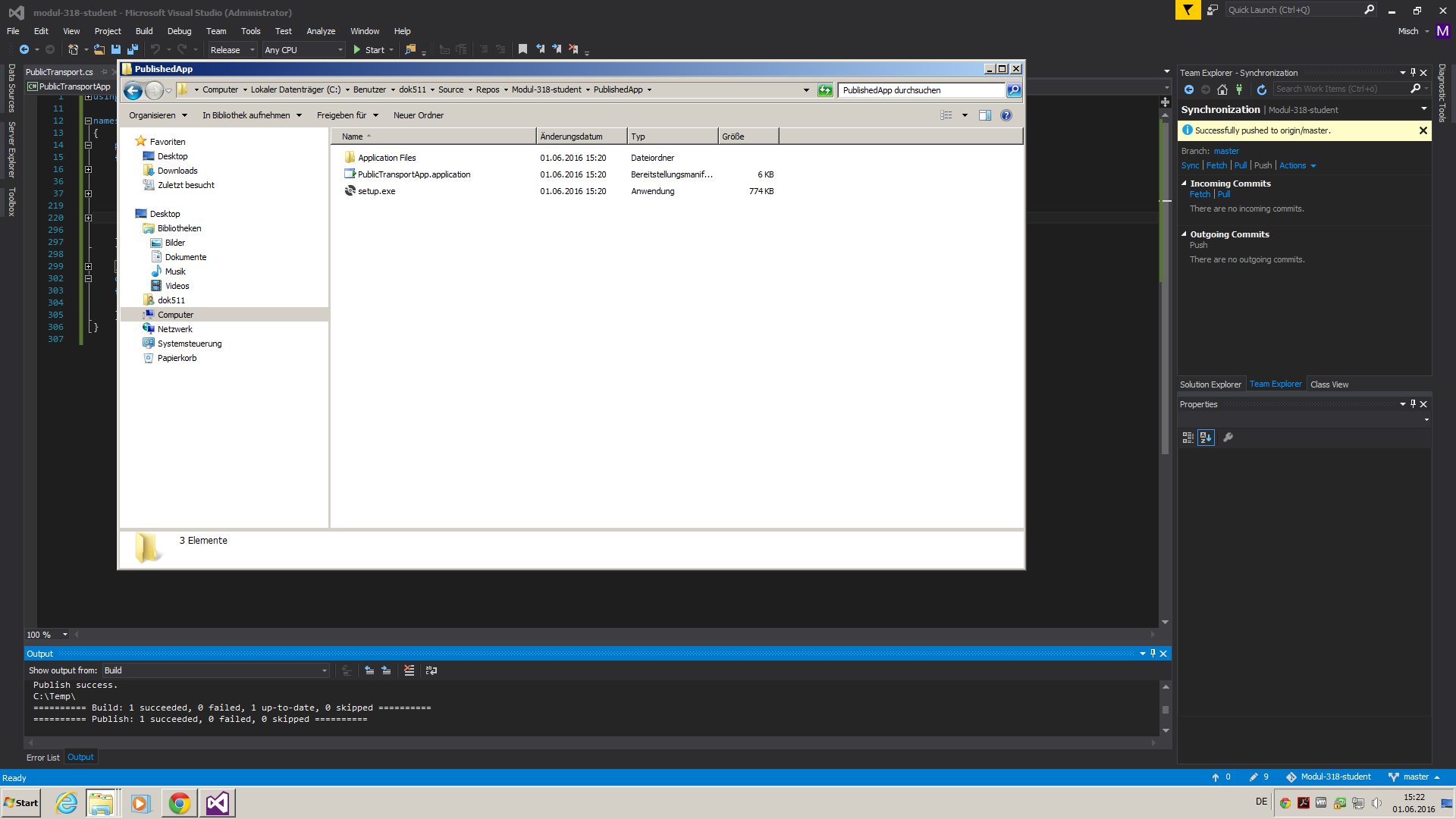
## Portierbarkeit

Das Programm muss nicht installiert werden. Wenn man das releasete exe-File von „Modul-318-student\PublicTransportApp\bin\Release“ und die anderen unten aufgeführten Dateien hat, kann man dieses in einen eigenen Ordner legen und ausführen. Zum „deinstallieren“ kann man das File einfach löschen.



### PublishedApp (Empfohlen)

Ich habe die App published und die Files in „Modul-318-student\PublishedApp“ abgelegt. Bei Klick auf die Setup.exe kann das Programm installiert werden. Deinstalliert wird wie gewohnt über die Systemsteuerung.



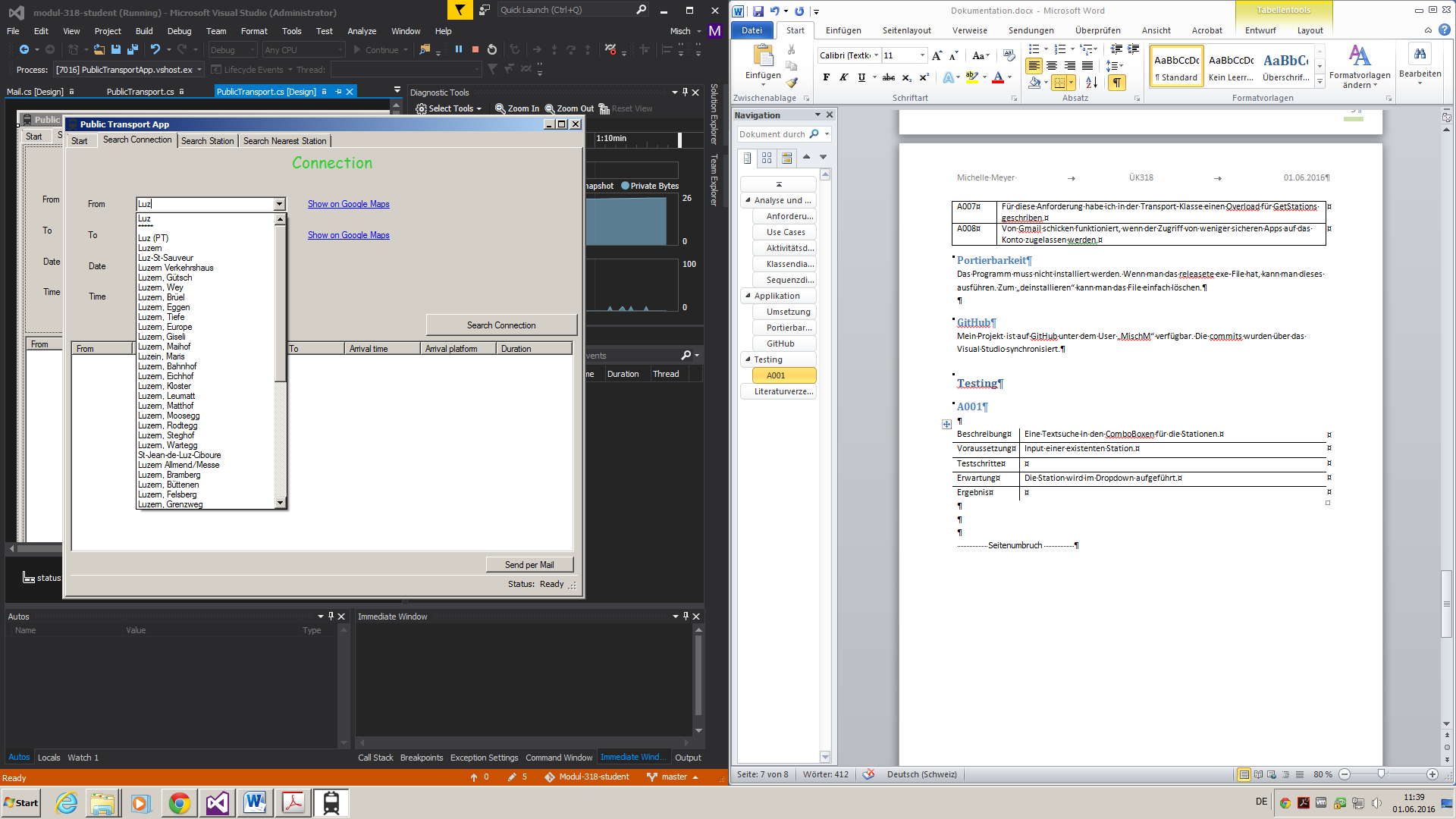
## GitHub

Mein Projekt ist auf GitHub unter dem User „MischM“ verfügbar (<https://github.com/MischM/modul-318-student> ). Die commits wurden über das Visual Studio synchronisiert.

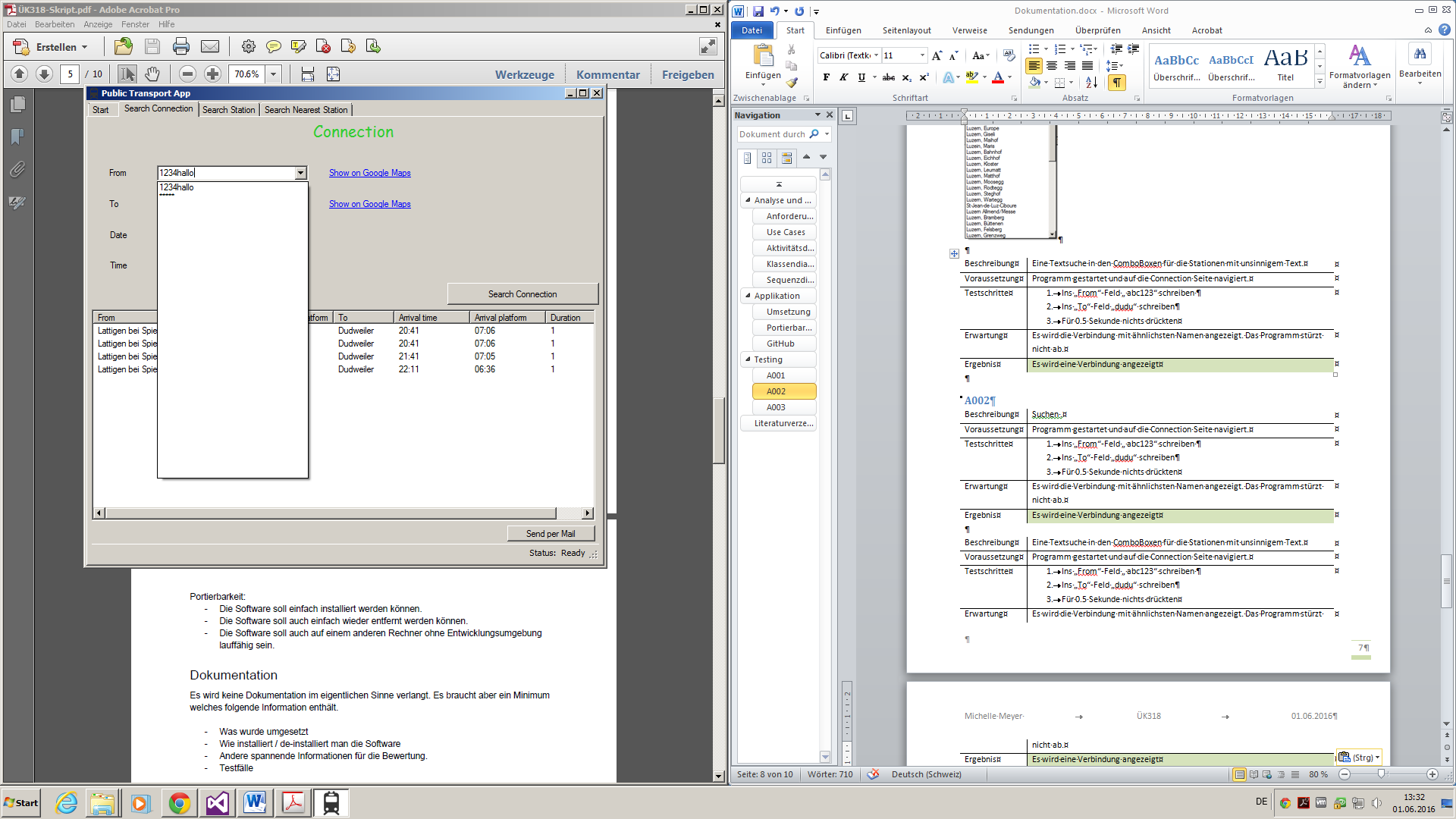
# Testing

## A001 / A004

|  |  |
| --- | --- |
| Beschreibung | Eine Textsuche in den ComboBoxen für die Stationen. |
| Voraussetzung | Programm gestartet und auf die Connection Seite navigiert. |
| Testschritte | 1. Ins „From“-Feld „ Luz“ schreiben 2. Für 0.5 Sekunde nichts drückten |
| Erwartung | Luzern wird im Dropdown aufgeführt, frühestens an dritter Stelle. |
| Ergebnis | Luzern ist an 4. Stelle im DropDown angezeitgt. |



|  |  |
| --- | --- |
| Beschreibung | Eine Textsuche in den ComboBoxen für die Stationen mit unsinnigem Text. |
| Voraussetzung | Programm gestartet und auf die Connection Seite navigiert. |
| Testschritte | 1. Ins „From“-Feld „ 1234hallo“ schreiben 2. Für 0.5 Sekunde nichts drückten |
| Erwartung | Das DropDown wird mit mindestens dem eingegebenen Text und den \*\*\* angezeigt. |
| Ergebnis | Eingabe und Sterne werden als Einträge angezeigt. |



## A002

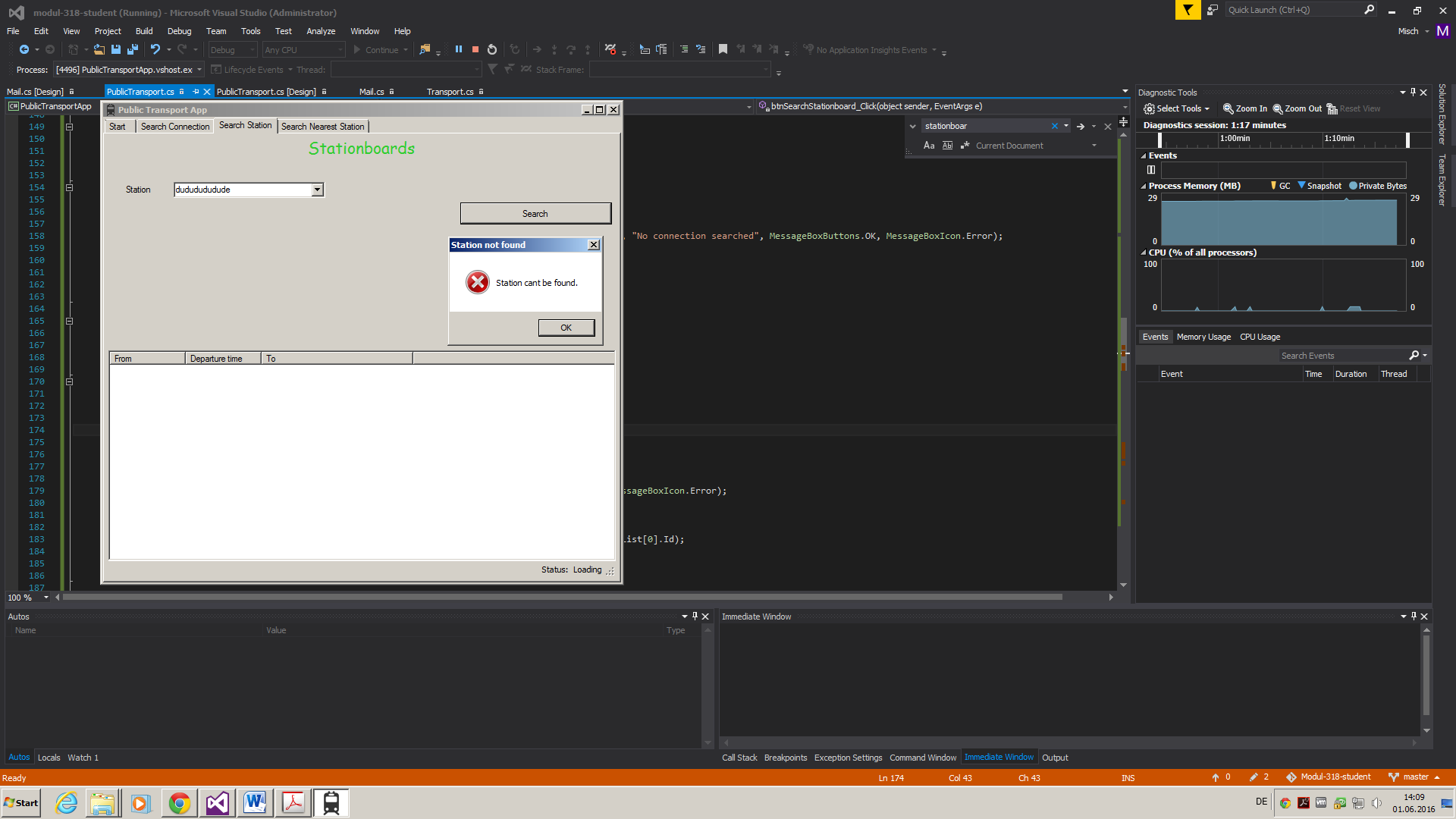
|  |  |
| --- | --- |
| Beschreibung | Suchen .einer Verbindung von zwei Stationen. |
| Voraussetzung | Programm gestartet und auf die Connection Seite navigiert. |
| Testschritte | 1. Ins „From“-Feld „ Sursee“ schreiben 2. Ins „To“-Feld „Luzern“ schreiben 3. „Search Connection“ drückten |
| Erwartung | Die ersten vier Verbindungen von Sursee nach Luzern werden in der ListView angezeigt. |
| Ergebnis | Die Verbindungen werden angezeigt |

|  |  |
| --- | --- |
| Beschreibung | Eine Textsuche in den ComboBoxen für die Stationen mit unsinnigem Text. |
| Voraussetzung | Programm gestartet und auf die Connection Seite navigiert. |
| Testschritte | 1. Ins „From“-Feld „ abc123“ schreiben 2. Ins „To“-Feld „dudu“ schreiben 3. Für 0.5 Sekunde nichts drückten |
| Erwartung | Es wird die Verbindung mit ähnlichsten Namen angezeigt. Das Programm stürzt nicht ab. |
| Ergebnis | Es wird eine Verbindung angezeigt |

## A003

|  |  |
| --- | --- |
| Beschreibung | Suchen einer Abfahrtstafel für eine existente Station. |
| Voraussetzung | Programm gestartet und in das Stationboard Tab navigiert. |
| Testschritte | 1. Ins „Station“-Feld Luzern eingeben 2. „Search“ drücken |
| Erwartung | Es werden die Verbindungen von Luzern aus angezeigt |
| Ergebnis | In der ListView wird die Abfahrtstafel angezeigt. |

|  |  |
| --- | --- |
| Beschreibung | Suchen einer Abfahrtstafel für eine inexistente Station. |
| Voraussetzung | Programm gestartet und in das Stationboard Tab navigiert. |
| Testschritte | 1. Ins „Station“-Feld „dudududude“eingeben 2. „Search“ drücken |
| Erwartung | Es erscheint eine Fehlermeldung. |
| Ergebnis | In der ListView wird die Abfahrtstafel angezeigt. |



## A005

|  |  |
| --- | --- |
| Beschreibung | Suchen einer Verbindung zu einem anderen Zeitpunkt für einen existenten Ort. |
| Voraussetzung | Programm gestartet und in das Connection Tab navigiert. |
| Testschritte | 1. Ins „From“-Feld „Sursee“eingeben 2. Ins „To“-Feld „Luzern“ eingeben 3. Time auf „16:15“ setzen 4. Radiobutton „Departure“ ist aktiviert 5. „Search Connection“ drücken |
| Erwartung | Keine der angezeigten Verbindungen ist vor 16:15 |
| Ergebnis | Die Verbindungen sind erst nach 16:15. Erste Abfahrt ist 16:18. |

## A006

|  |  |
| --- | --- |
| Beschreibung | Anzeigen einer validen Station auf Google Maps. |
| Voraussetzung | Programm gestartet und in das Connection Tab navigiert. |
| Testschritte | 1. Ins „From“-Feld „Sursee“eingeben 2. „Show on Google Maps“ drücken |
| Erwartung | Der Standort wird auf Google Maps im default Browser gezeigt. |
| Ergebnis | Der Browser wird geöffnet und die Station ist in Maps markiert. |

## A007

|  |  |
| --- | --- |
| Beschreibung | Anzeigen der naheliegensten Stationen. |
| Voraussetzung | Programm gestartet und in das Nearest Station Tab navigiert. |
| Testschritte | 1. Ins „Current address“-Feld „6210 Sursee, Kottenmatte 2“eingeben 2. „Search“ drücken |
| Erwartung | In der ListView werden Stationen nach Distanz geordnet angezeigt. |
| Ergebnis | Erwartung erfüllt. |

## A008

|  |  |
| --- | --- |
| Beschreibung | Senden einer Email. |
| Voraussetzung | Programm gestartet und bereitseine Verbindung gesucht, welche angezeigt wird. |
| Testschritte | 1. ImFormular alle erforderlichen Feld korrekt ausfüllen. 2. „Send“ drücken |
| Erwartung | Bestätigungs Message das gesendet wurde taucht auf. Email ist in der Inbox. |
| Ergebnis | Erwartungen erfüllt. |

# Literaturverzeichnis

Analyse und Design.pdf. (24. 05 2016).

Analyse und Design.pdf. (24. 05 2016).

Analyse und Design.pdf. (24. 05 2016).

Meyer, M. (24. 05 2016). Von www.websequencediagrams.com abgerufen

Meyer, M. (24. 05 2016). Von www.nomnoml.com abgerufen

Meyer, M., & Poblete, Y. (24. 05 2016).

Meyer, M., & Poblete, Y. (24. 05 2016).

ÜK318-Skript.pdf. (21. 05 2016).

Ward, J. (08. 07 2010). Von http://www.tamingdata.com/2010/07/08/the-project-management-tree-swing-cartoon-past-and-present/ abgerufen