Dokumentation Projektarbeit

ÜK-318

Giannopoulos Kalyca

|  |  |  |
| --- | --- | --- |
| **Datum** | **Änderungen** | **Autor** |
| 15.11.2017 | Initialisierung | Kalyca Giannopoulos |
| 20.11.2017 | Einleitung, Zweck, Anforderungen, Use Cases, Aktivitätendiagramme hinzufügen | Kalyca Giannopoulos |
| 21.11.2017 | Umsetzung, Probleme, Mockup, Programmierrichtlinien, Installationsanleitung, Testfälle hinzufügen & überarbeitungen | Kalyca Giannopoulos |

Inhaltsverzeichnis

[Einleitung 3](#_Toc499039574)

[Zweck des Dokuments 3](#_Toc499039575)

[Anforderungen 3](#_Toc499039576)

[Was wurde umgesetzt? 4](#_Toc499039577)

[Probleme? Was konnte nicht umgesetzt werden? 4](#_Toc499039578)

[Mockup 4](#_Toc499039579)

[Plan: 4](#_Toc499039580)

[Programmierrichtlinien 6](#_Toc499039581)

[Naming Conventions 6](#_Toc499039582)

[Declaration 6](#_Toc499039583)

[Comments 6](#_Toc499039584)

[Statements (New Lines, Indentation) 7](#_Toc499039585)

[Use Cases 7](#_Toc499039586)

[Aktivitätendiagramme 8](#_Toc499039587)

[Testfälle (Systemtests) 9](#_Toc499039588)

[Testfall A001 Verbindung suchen 9](#_Toc499039589)

[Testfall A002 mehrere Verbindungen gleichzeitig 10](#_Toc499039590)

[Testfall A003 Abfahrtstafel 10](#_Toc499039591)

[Installationsanleitung 11](#_Toc499039592)

# Einleitung

Im Programm Swiss Transport geht es darum, Verbindungen mit Hilfe des Fahrplans des ÖVs zwischen zwei Stationen zu suchen und diese anzuzeigen.

# Zweck des Dokuments

In diesem Dokument möchte ich die Umsetzung dieses Projekts erläutern. Welchen Anforderungen gestellt waren welche erfüllt wurden und welchen nicht. Welche Vorarbeiten erledigt wurden.

# Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Beschreibung** | **Priorität** |
| A001 | Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | 1 |
| A002 | Als ÖV-Benutzer möchte ich die aktuellen, d.h. mindestens die nächsten vier bis fünf Verbindungen zwischen den beiden gefundenen und ausgewählten Stationen sehen, damit ich weiss wann ich zur Station muss, um den für mich idealen Anschluss zu erwischen. | 1 |
| A003 | Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrtstafel haben kann. | 1 |
| A004 | Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste SuchResultate erscheinen, damit ich effizienter nach Stationen suchen kann. | 2 |
| A005 | ALs ÖV-Benutzer möchte ich nicht nur aktuelle Verbindungen suchen können, sondern auch solche zu einem beliebigen anderen Zeitpunkt, damit ich zukünftige Reisen planen kann. | 2 |
| A006 | Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. | 3 |
| A007 | Als ÖV-Benutzer möchte Stationen finden, die sich ganz in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann. | 3 |
| A008 | Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können. | 3 |

# Was wurde umgesetzt?

Umgesetzt wurden A001 - A003. Das Programm kann eine Verbindung zwischen zwei Orten finden und gibt diese aus. Bei der Ausgabe ist Abfahrtszeit, Abfahrtsort, Zielort, Ankunftszeit und Dauer ersichtlich. Man kann sich zudem eine Abfahrtstafel erstellen, bei welcher man aufgrund eines Abfahrtsortes, alle Verbindungen in einem bestimmten Zeitraum sieht.

# Probleme? Was konnte nicht umgesetzt werden?

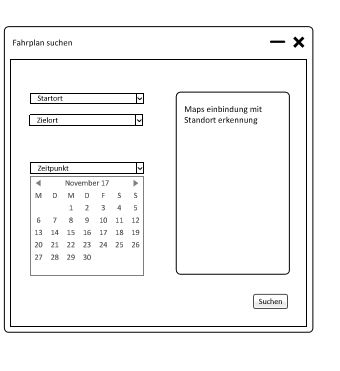
Bei A004 konnte ich nicht umsetzen da ich nicht herausfinden konnte wie ich Autocomplete anwenden kann.

Für A005 existieren die Felder, diese sind jedoch noch nicht implementiert.

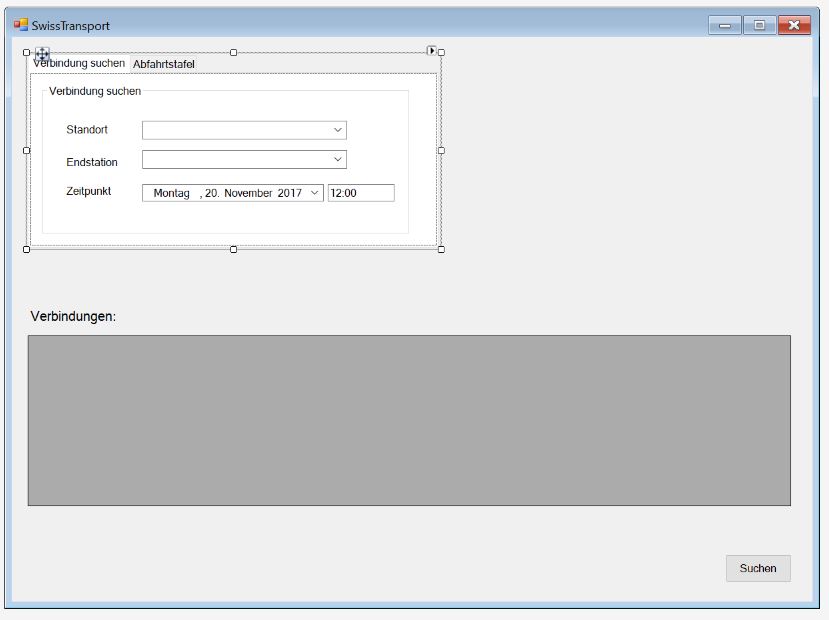
A006 – A008 konnte nicht umgesetzt werden, da die Zeit nicht reichte. Ich habe zudem, bis auf das wenig in der Schule, noch nie mit C# (WPF) gearbeitet und allgemein keine Erfahrung mit GUI.

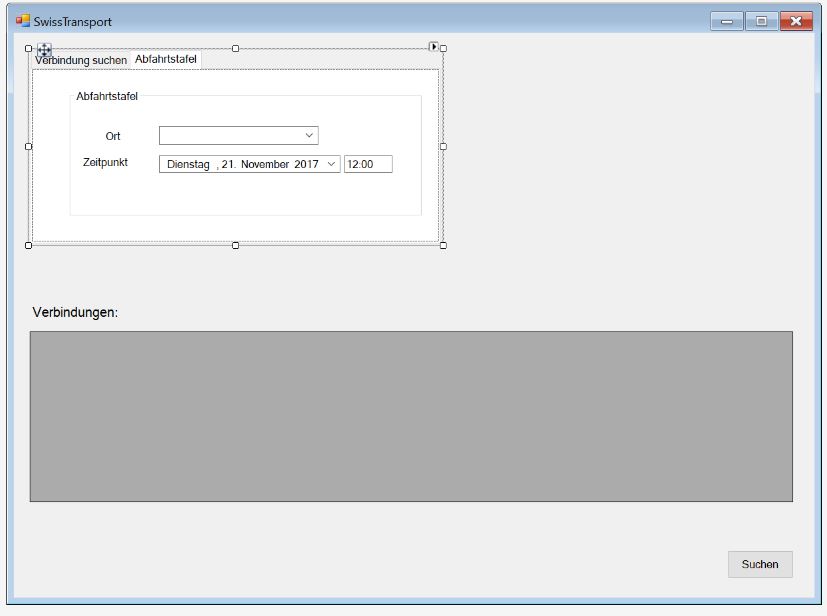
# Mockup

### Plan:

****

**Tatsächlich:**

****

****

# Programmierrichtlinien

### Naming Conventions

* Variablen (global, lokal)
  + global
    - \_camelCase
    - Beispiel: string \_resultat
  + lokal
    - \_camelCase
    - Beispiel: string \_resultat
* Eigenschaften
  + keine Eigene gebraucht
* Methoden / Funktionen
  + PascalCase ausser EventHandler
  + Beispiel: private string ReplaceString(string \_anfang)
* Klassen
  + keine Eigene gebraucht
* GUI-Controls
  + Button: btnName
  + TextBox: txtName
  + Label: lName
  + DataGridView: dgvName
  + groupBox: gbEingabe
  + comboBox: cbName
  + tabControl: tcName
  + TapPage: tpName
  + DateTimePicker: dtpName

### Declaration

* Lokale, globale (=Klassenvariablen) Variablen Wo? Wie?
  + globale
    - beim Erstellen
  + lokale
    - beim Erstellen

### Comments

* Wo?
  + oberhalb und nebendran
* Was?
  + alles
* Wie?
  + /// (Summary)
  + // (Einzeiliges)
  + /\* \*/ (Mehrzeiliges)

### Statements (New Lines, Indentation)

* if-else , for, foreach, while, etc
  + Klammern {} auf neuer Zeile

# Use Cases

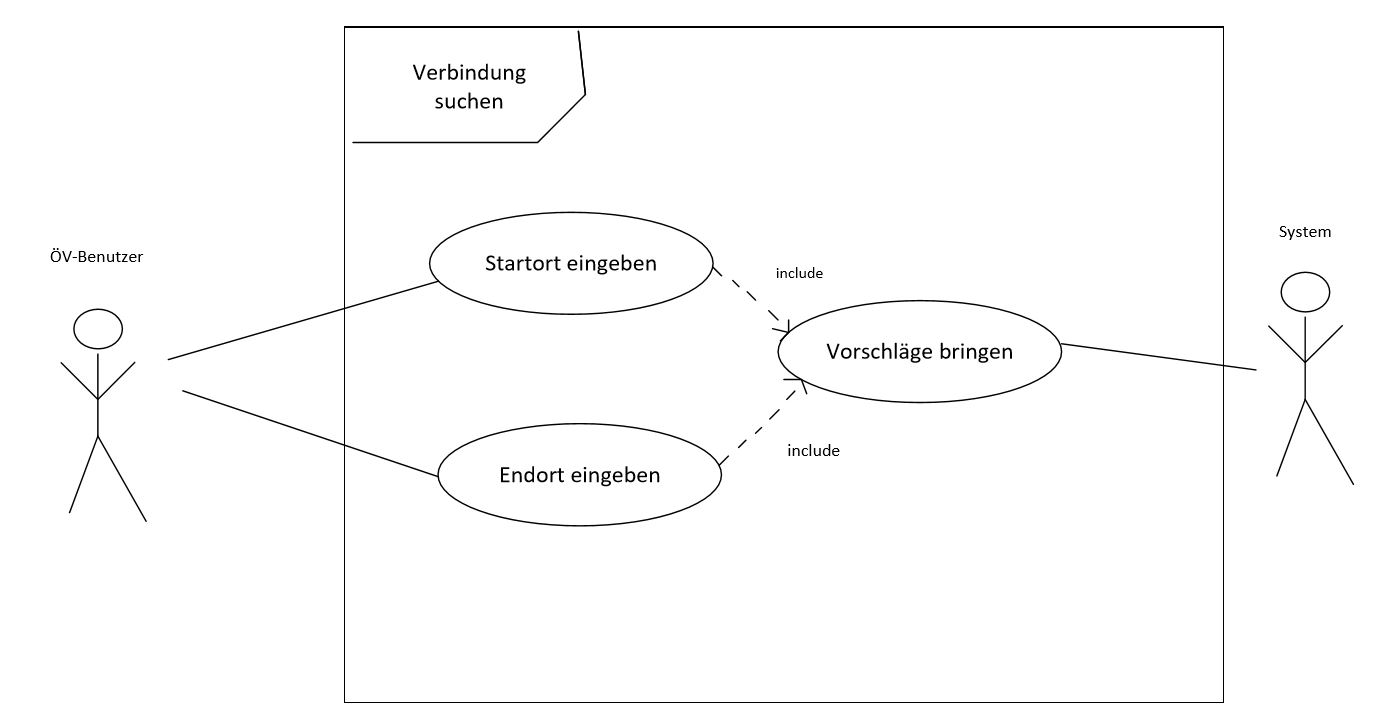


Abbildung A001

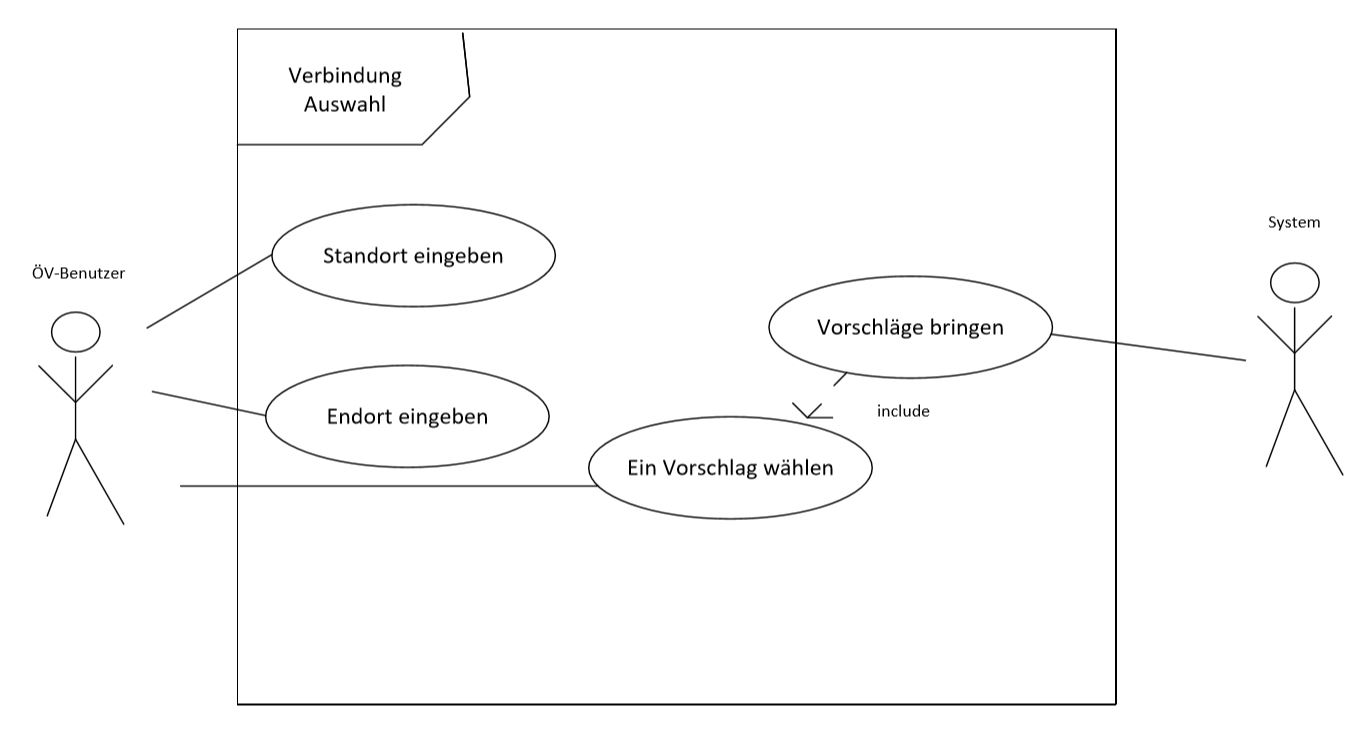


Abbildung A002

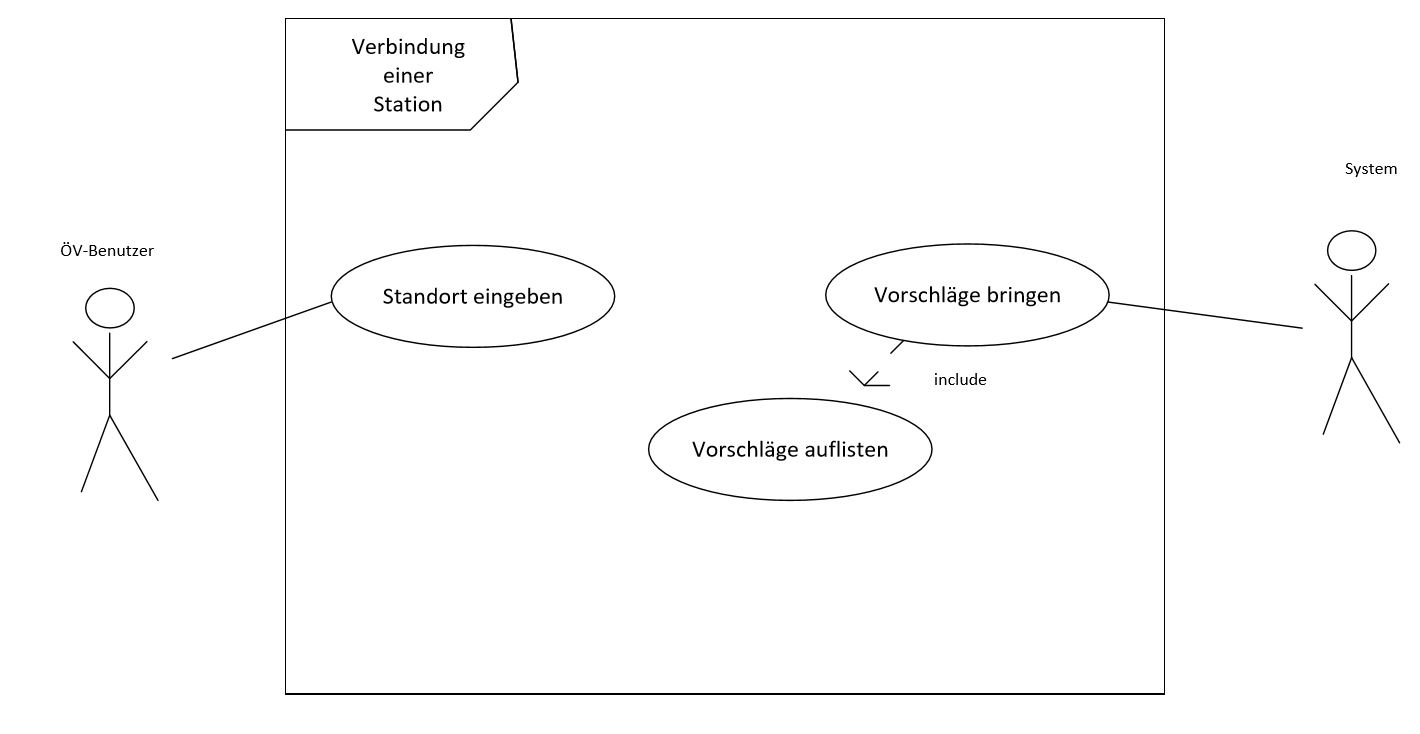


Abbildung A003

# Aktivitätendiagramme

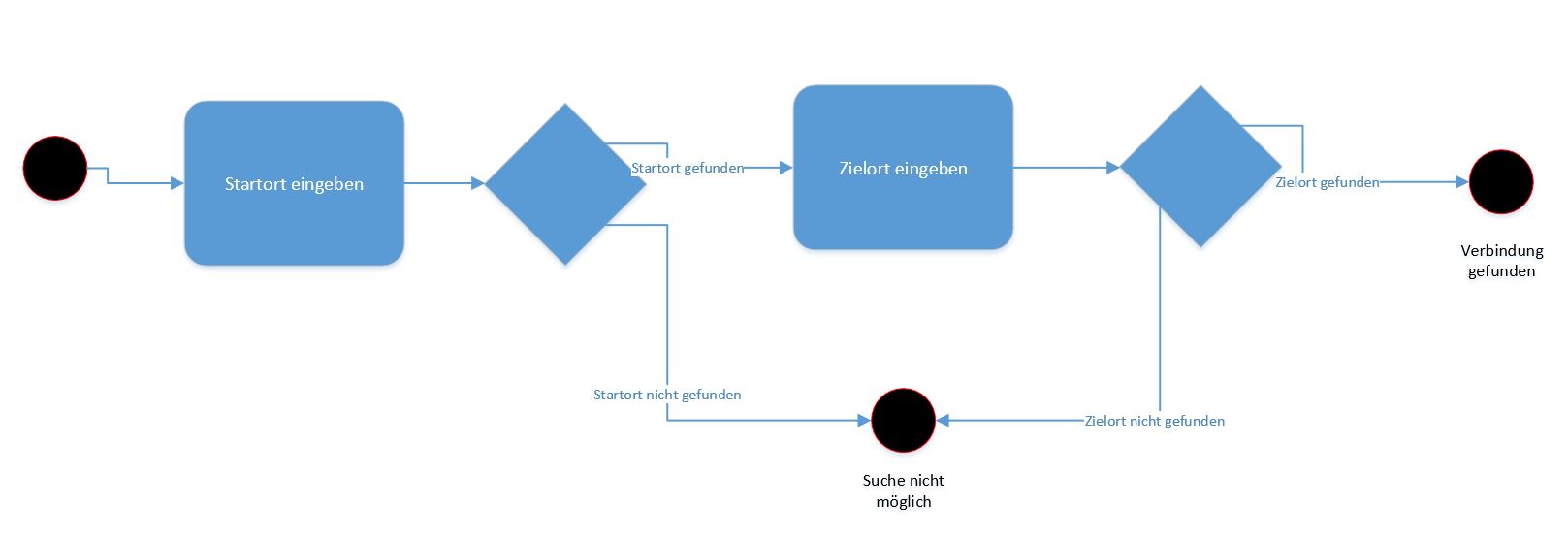


Abbildung A001

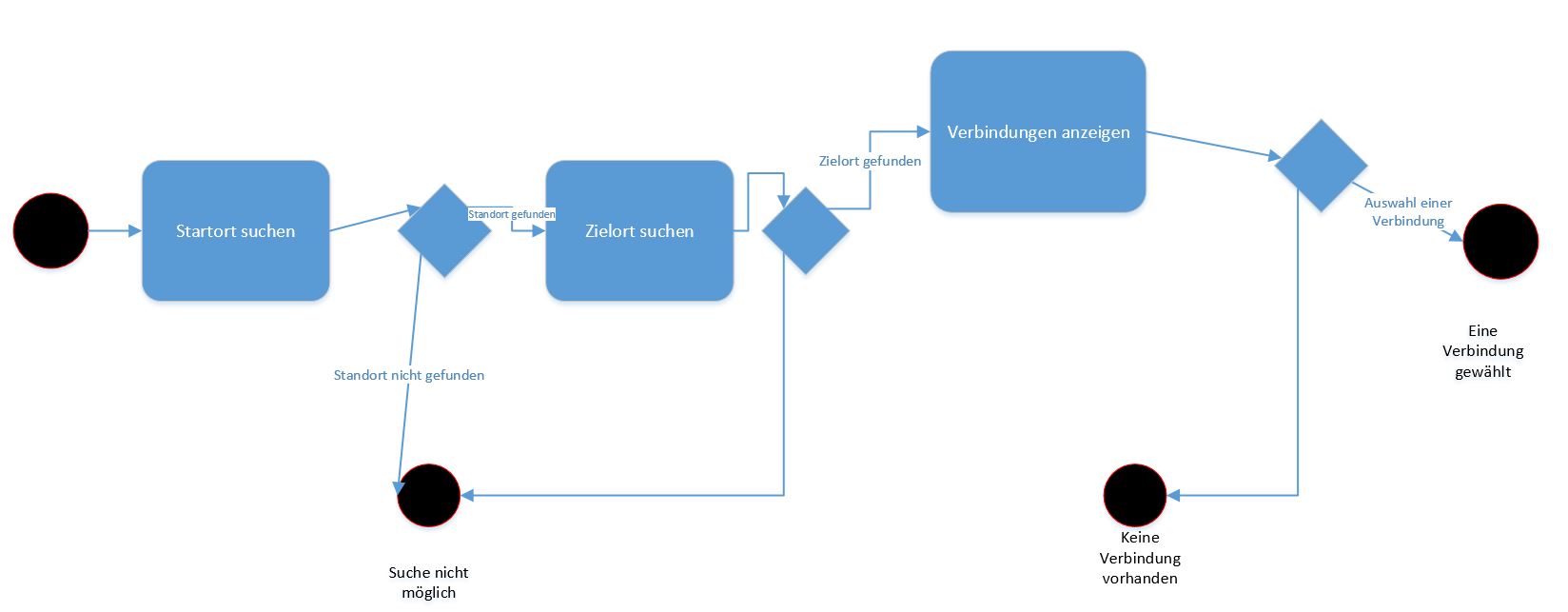


Abbildung A002

# Testfälle (Systemtests)

## Testfall A001 Verbindung suchen

**Vorbedingungen**

Was muss erfüllt sein, damit der Testfall durchgeführt werden kann?

- Programm starten (.exe)

- Tab **Verbindung suchen** auswählen

**Testszenario**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Bei Standort Surs eingeben, Dropdown drücken | Vorschläge von Haltestellen mit Surs |
| 2 | Sursee auswählen | Standort mit Sursee ausgefüllt, Dropdown verschwindet wieder |
| 3 | Bei Endstation Luz eingeben, Dropdown drücken | Vorschläge von Haltestellen mit Luz |
| 4 | Luzern auswählen | Endstation mit Luzern ausgefüllt, Dropdown verschwindet wieder |
| 5 | Button Suchen drücken | Erscheinen einer Tabelle bei Verbindungen: mit 4 Verbindungen und den Infos (Abfahrtszeit, Abfahrtsort, Zielort, Ankunftszeit, Dauer) |

## Testfall A002 mehrere Verbindungen gleichzeitig

**Vorbedingungen**

Was muss erfüllt sein, damit der Testfall durchgeführt werden kann?

* Programm starten (.exe)
* Tab **Verbindung suchen** auswählen

**Testszenario**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Bei Standort Ebi eingeben, Dropdown drücken | Vorschläge von Haltestellen mit Ebi |
| 2 | Ebikon auswählen | Standort mit Ebikon ausgefüllt, Dropdown verschwindet wieder |
| 3 | Bei Endstation Luz eingeben, Dropdown drücken | Vorschläge von Haltestellen mit Luz |
| 4 | Luzern auswählen | Endstation mit Luzern ausgefüllt, Dropdown verschwindet wieder |
| 5 | Button Suchen drücken | Erscheinen einer Tabelle bei Verbindungen: mit 4 Verbindungen und den Infos (Abfahrtszeit, Abfahrtsort, Zielort, Ankunftszeit, Dauer) |

## Testfall A003 Abfahrtstafel

**Vorbedingungen**

Was muss erfüllt sein, damit der Testfall durchgeführt werden kann?

* Programm starten (.exe)
* Tab **Abfahrtstafel** auswählen

**Testszenario**

|  |  |  |
| --- | --- | --- |
| **Schritt** | **Aktivität** | **Erwartetes Resultat** |
| 1 | Bei Ort Luz eingeben, Dropdown drücken | Vorschläge von Haltestellen mit Luz |
| 2 | Luzern auswählen | Ort mit Luzern ausgefüllt, Dropdown verschwindet wieder |
| 3 | Button Suchen drücken | Erscheinen einer Liste bei Verbindungen: mit 42 e  Einträgen und den Infos (Abfahrtszeit, Abfahrtsort welcher gleich wie Ort oben sein sollte und Zielort) |

# Installationsanleitung