

Dokumentation, SBB Fahrplan

Projektarbeit Fahrplan

des

ÜK Moduls 318

geschrieben von

Martina Brawand

Inhaltsverzeichnis

1. EINLEITUNG	3
2. USER STORYS	4
3. MOCKUP	5
4. USE CASE	6
5. USE CASE BESCHREIBUNG	6
6. AKTIVITÄTSDIAGRAMM	8
7. PROGRAMMIERRICHTLINEN	9
8. TESTPROTOKOLL	11
9. INSTALLATION	11

1. Einleitung

Im aktuelle ÜK (M318, Analysieren und Objektbasiert programmieren) arbeiten wir an einem Projekt zur Realisierung eines Fahrplanes. Dieses Projekt wird in der Entwicklungsumgebung Visual Studio bearbeitet.

Dieses Dokument ist dazu da meine Schritte bis zum Endprodukt zu beschreiben und darzustellen.

Mein Fahrplan ermöglicht die Eingabe von zwei Standorten inkl. Datum & Uhrzeit. Das Programm zeigt anschliessend 4-5 Verbindungen an, welche der User für seinen Auswahl Prozess der idealen Verbindung nutzen kann.

2. User Storys

Story ID	Beschreibung	Umgesetzt	Priorität
01	Stationen anzeigen Als ÖV-Benutzer möchte ich Start- und Endstation mittels Textsuche anzeigen lassen können, damit ich die Station, die ich möchte, finden kann.	Ja	1
02	Verbindungsanzeige Als ÖV-Benutzer möchte ich mindestens die nächsten 4 bis 5 Verbindungen zwischen den beiden gefundenen und ausgewählten Stationen sehen, um den für mich idealen Anschluss auszuwählen.	Ja	1
03	Verbindungen ab einzelner Station Als ÖV-Benutzer möchte ich sehen, welche Verbindungen ab einer einzelnen bestimmten Station vorhanden sind, um alle Abfahrten zu kennen.	Ja	1
04	Autokomplete Als ÖV-Benutzer möchte ich, dass schon während meiner Eingabe erste Such-Resultate erscheinen, damit ich effizienter nach Stationen suchen kann.	Ja	2
05	Suche nach Datum & Zeit Als ÖV-Benutzer möchte ich nicht nur zur aktuellen Verbindung suchen können, sondern auch solche zu einem beliebigen anderen Zeitpunkt, damit ich zukünftige Reisen planen kann.	Ja	2
06	Karten Feature Als ÖV-Benutzer möchte ich sehen, wo sich eine Station befindet (auf einer Karte), damit ich mir besser vorstellen kann, wie ich zu dieser Station kommen kann.	Nein	2
07	Suche ausgehend von meiner Position Als ÖV-Benutzer möchte ich Stationen finden, die sich in der Nähe meiner aktuellen Position befinden, damit ich schnell einen Anschluss erreichen kann.	Nein	3
08	Weiterleitung per Mail Ich möchte meine gefundenen Resultate via Mail weiterleiten können, damit auch andere von meinen Recherchen profitieren können.	Nein	3

3. Mockup

Meine Vorstellung vom GUI wurde im Programm Balsamiq gezeichnet.
Das Endresultat sieht deutlich anders aus als meine Erwartungen.

Window Name

Fahrplan

Von: Nach:

Richtung Luzern	12.10	12.40
Richtung Luzern	12.15	12.45
Richtung Luzern	12.20	12.50
Richtung Luzern	12.30	12.55

01: Start und Endstation mittels
Textsuche suchen können.

Window Name

Fahrplan

Von: Nach:

Abfahrt:

Richtung Luzern	12.10	12.40
Richtung Luzern	12.15	12.45
Richtung Luzern	12.20	12.50
Richtung Luzern	12.30	12.55

02: Die nächsten 4-5 Verbindungen
anzeigen.

Window Name

Fahrplan

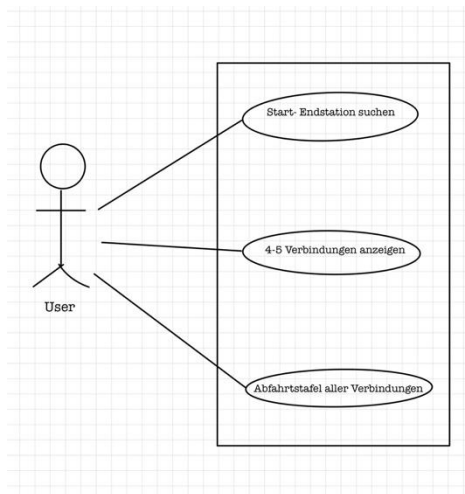
Von: Nach:

Abfahrt:

Richtung Luzern	12.10	12.40
Richtung Luzern	12.15	12.45
Richtung Luzern	12.20	12.50
Richtung Luzern	12.30	12.55

03: Verbindungen ab einer bestimmten
Station anzeigen.

4. Use Case



Das Use Case Diagramm für die Anforderung 01 bis 03 wurden im Programm Goodnotes gezeichnet. Es stellt die Anforderungen des Users an das Fahrplanprogramm dar.

5. Use Case Beschreibung

Use Case - Verbindung suchen 01

Beschreibung	Start und Endstation mittels Textsuche suchen
Akteur	User
Auslöser	Eine Abfrage
Vorbedingungen	Das Programm muss installiert sein
Ablauf	

- Programm öffnen
- Abfahrt und Zielort eingeben
- Suchen anklicken

Alternativer Ablauf Keiner

Ergebnisse Anzeige des Fahrplans

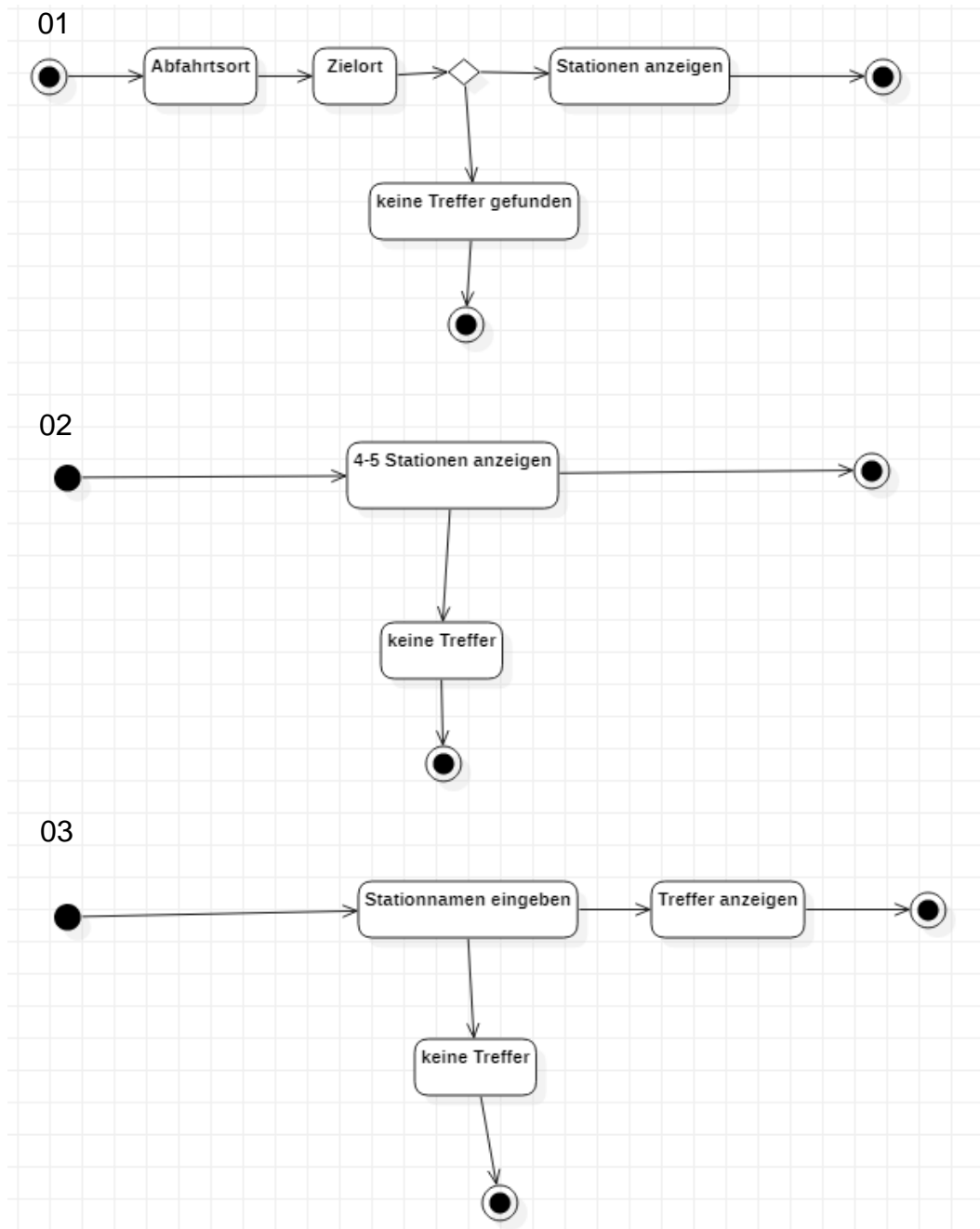
Use Case - Verbindungen anzeigen 02

Beschreibung	Der Benutzer möchte die nächsten 4-5 Verbindungen anzeigen lassen
Akteur	User
Auslöser	Eine Abfrage
Vorbedingungen	Der User muss das Programm installiert haben
Ablauf	<ul style="list-style-type: none">• Programm öffnen• Abfahrt und Zielort eingeben• Aktuelle Uhrzeit und Datum auswählen• Suchen klicken
Alternativer Ablauf	keiner
Ergebnisse	Anzeige des Fahrplans

Use Case - Aktuelle Verbindungen sehen 03

Beschreibung	Der Benutzer möchte alle Verbindungen von einer bestimmten Station sehen
Akteur	User
Auslöser	Eine Abfrage
Vorbedingungen	Der User muss das Programm installiert haben
Ablauf	<ul style="list-style-type: none">• Programm öffnen• Abfahrt und Zielort eingeben• Aktuelle Uhrzeit und Datum auswählen• Suchen klicken
Alternativer Ablauf	keiner
Ergebnisse	Anzeige des Fahrplans

6. Aktivitätsdiagramm



Das Aktivitätsprogramm für 01, 02 und 03 diese wurden im StarUML gezeichnet.

7. Programmierrichtlinien

Naming Conventions

- Soll immer auf Englisch sein, kein mix (einheitlich)
- Generell PascalCase verwenden.
- Constants in UPPER_CASE

Declarations

- Constructor
- Properties
- Methoden

Comments

- In einer separaten Zeile

Beispiel

```
public class Car {  
    private int speed; ///private variable declaration  
    public int wheels; ///public variable declaration  
  
    /*...constructor, etc...*/  
  
    public void speedUp() {  
        ///local variable declaration, in line assignment, only seen within speedUp method  
        int speedIncrease = 10;  
        speed += speedIncrease;  
    }  
}
```

Comments

```
C# Copy  
  
// First, in class Program, define the delegate type and a method that  
// has a matching signature.  
  
// Define the type.  
public delegate void Del(string message);  
  
// Define a method that has a matching signature.  
public static void DelMethod(string str)  
{  
    Console.WriteLine("DelMethod argument: {0}", str);  
}
```

Kommentare können mit `//`, `/* */` und `#` geschrieben werden.

Statements (Beispiele)

if else

```
static void IfStatement(string[] args)
{
    if (args.Length == 0)
    {
        Console.WriteLine("No arguments");
    }
    else
    {
        Console.WriteLine("One or more arguments");
    }
}
```

foreach

```
static void ForEachStatement(string[] args)
{
    foreach (string s in args)
    {
        Console.WriteLine(s);
    }
}
```

for

```
static void ForStatement(string[] args)
{
    for (int i = 0; i < args.Length; i++)
    {
        Console.WriteLine(args[i]);
    }
}
```

do while

```
static void DoStatement(string[] args)
{
    string s;
    do
    {
        s = Console.ReadLine();
        Console.WriteLine(s);
    } while (!string.IsNullOrEmpty(s));
}
```

switch case

```
static void SwitchStatement(string[] args)
{
    int n = args.Length;
    switch (n)
    {
        case 0:
            Console.WriteLine("No arguments");
            break;
        case 1:
            Console.WriteLine("One argument");
            break;
        default:
            Console.WriteLine($"{n} arguments");
            break;
    }
}
```

while

```
static void WhileStatement(string[] args)
{
    int i = 0;
    while (i < args.Length)
    {
        Console.WriteLine(args[i]);
        i++;
    }
}
```

int

```
static void Expressions(string[] args)
{
    int i;
    i = 123;           // Expression statement
    Console.WriteLine(i); // Expression statement
    i++;               // Expression statement
    Console.WriteLine(i); // Expression statement
}
```

8. Testprotokoll

Durchgeführt am: 08.01.2021

Durchgeführt von: Martina Brawand

Getestet von: Martina Brawand

Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	Der User füllt die "Angaben" bei der "Verbindungssuche" aus "Von: Nach:" und klick anschliessend auf Suchen	Die Daten Werden Angezeigt "Datum, Abfahrtszeit, usw.	Die Daten Werden Angezeigt "Datum, Abfahrtszeit, usw.	✓
2	Der User füllt zusätzlich die "Erweiterte Angaben", "Datum" und "Zeit" aus und klick anschliessend auf Suchen	Die Daten Werden Angezeigt "Datum, Abfahrtszeit, usw.	Die Daten Werden Angezeigt "Datum, Abfahrtszeit, usw.	✓
3	Der User füllt den "Abfahrtsort" bei der "Abfahrtsübersicht" aus und klick anschliessend auf Suchen	Die Daten "Abfahrt", "Richtung" und "Linie" werden angezeigt.	Die Daten "Abfahrt", "Richtung" und "Linie" werden angezeigt.	✓

9. Installation

Führe das Programm «Setup» aus. Nun erfolgt die Installation automatisch.

Für die Deinstallation gebe in der Windows Suchleiste «Programme hinzufügen oder entfernen» ein. Wähle das entsprechende Programm und deinstalliere es.