

Dokumentation Projektarbeit

Fahrplan-Applikation

ÜK 318-1

Analysieren und objektbasiert
programmieren mit Komponenten

Lukas Bucher
Adligenswil, 04.12.2019

Eingereicht bei Markus Estermann

Inhaltsverzeichnis

1. Einleitung & Zweck	3
2. Funktionsumfang	3
2.1 Implementierte Funktionen	3
1.1 Fehlende Funktionen	3
3. Mockups	4
3.1 Verbindung	4
3.2 Abfahrtsplan	4
3.3 Informationen	5
3.4 Ausgabefenster	5
4. Use Cases	6
4.1 Use Case Diagramm	6
4.2 Use Case Beschreibungen	6
4.3 Aktivitätsdiagramm	8
5. Programmierrichtlinien.....	9
5.1 Naming Conventions	9
5.2 Declaration	9
5.3 Comments	9
6. Testing	9
6.1 Testfall «Hin- und Rückfahrt anzeigen lassen»	9
7. Installationsanleitung	10

1. Einleitung & Zweck

Der Zweck dieses Dokuments ist, meinen Arbeitsablauf zum Projekt Fahrplan-Applikation aufzuzeigen.

2. Funktionsumfang

2.1 Implementierte Funktionen

Ich habe mich an den in den Kursunterlagen aufgeführten Anforderungen orientiert.

Folgende Funktionen sind implementiert:

- Textsuche in Eingabefeldern inklusive Vervollständigung
- Verbindungen mit Abfahrts- und Ankunftsstation finden
- Abfahrtsplan mit Verbindungen ab Abfahrtsort finden
- Verbindungen nach eingegebener Uhrzeit finden
- Station in Google Maps anzeigen lassen
- Ergebnisse per Mail weiterleiten


1.1 Fehlende Funktionen


Die einzige fehlende Funktion nach den Anforderungen in den Kursunterlagen ist die Suche nach nahen gelegenen Stationen.


3. Mockups

3.1 Verbindung

Fahrplan







Abfahrtsort

list of stations

Datum

Uhrzeit

ab ☐ an


Ankunftsart


list of stations


search

3.2 Abfahrtsplan

Fahrplan







Abfahrtsort

list of stations

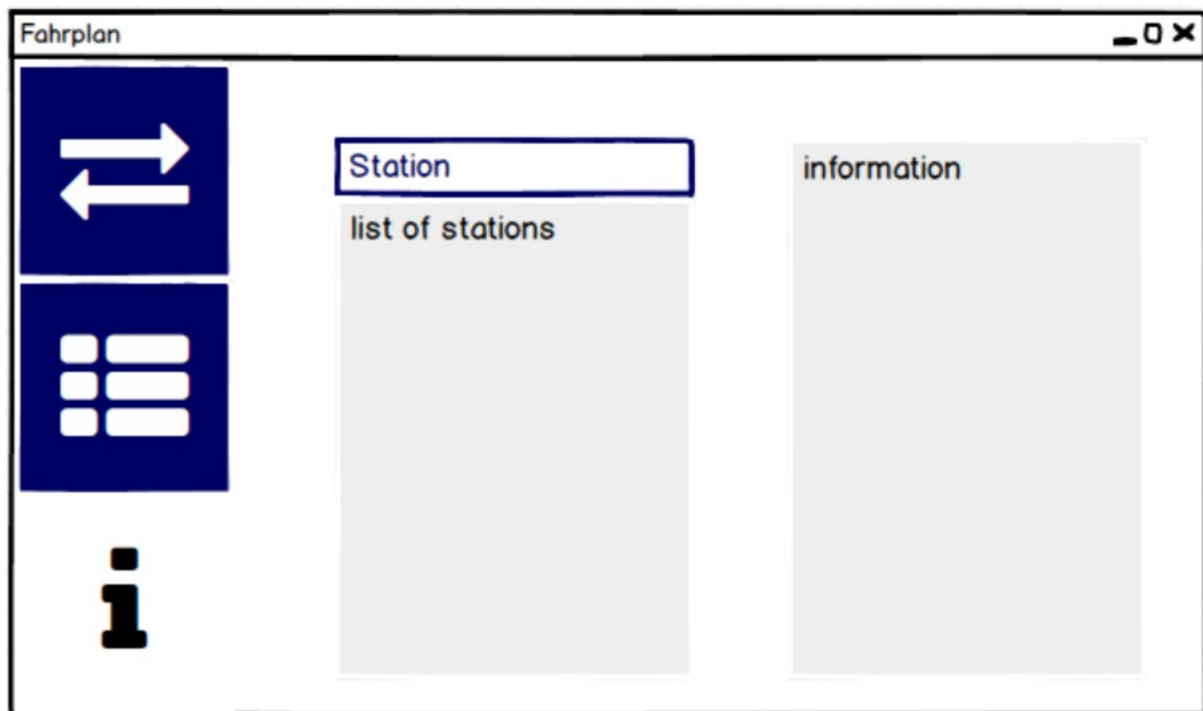
Datum

Uhrzeit

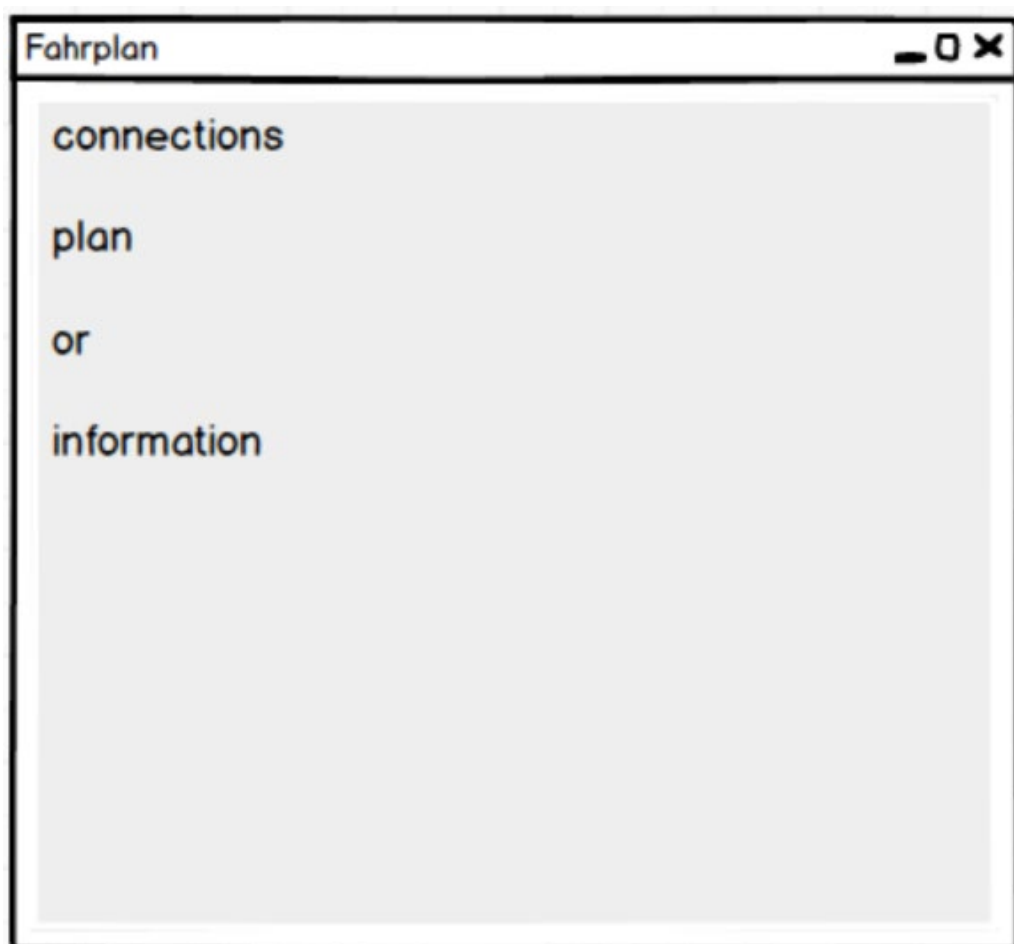
ab ☐ an

search

3.3 Informationen

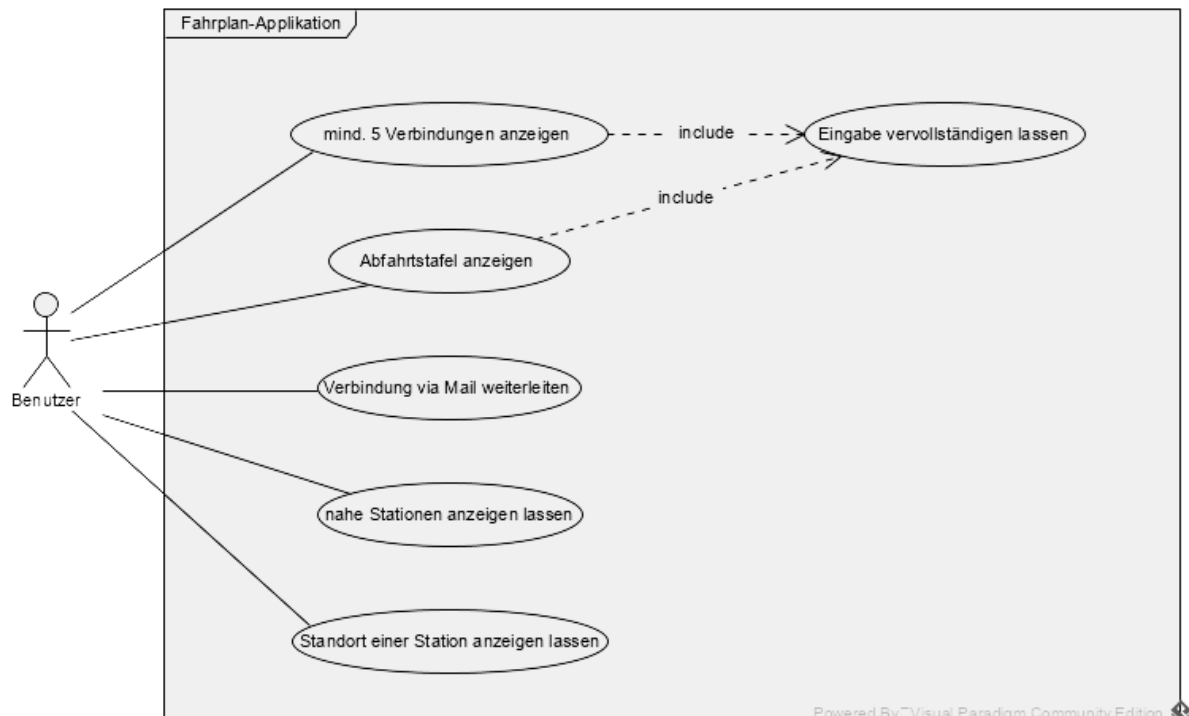


3.4 Ausgabefenster



4. Use Cases

4.1 Use Case Diagramm



4.2 Use Case Beschreibungen

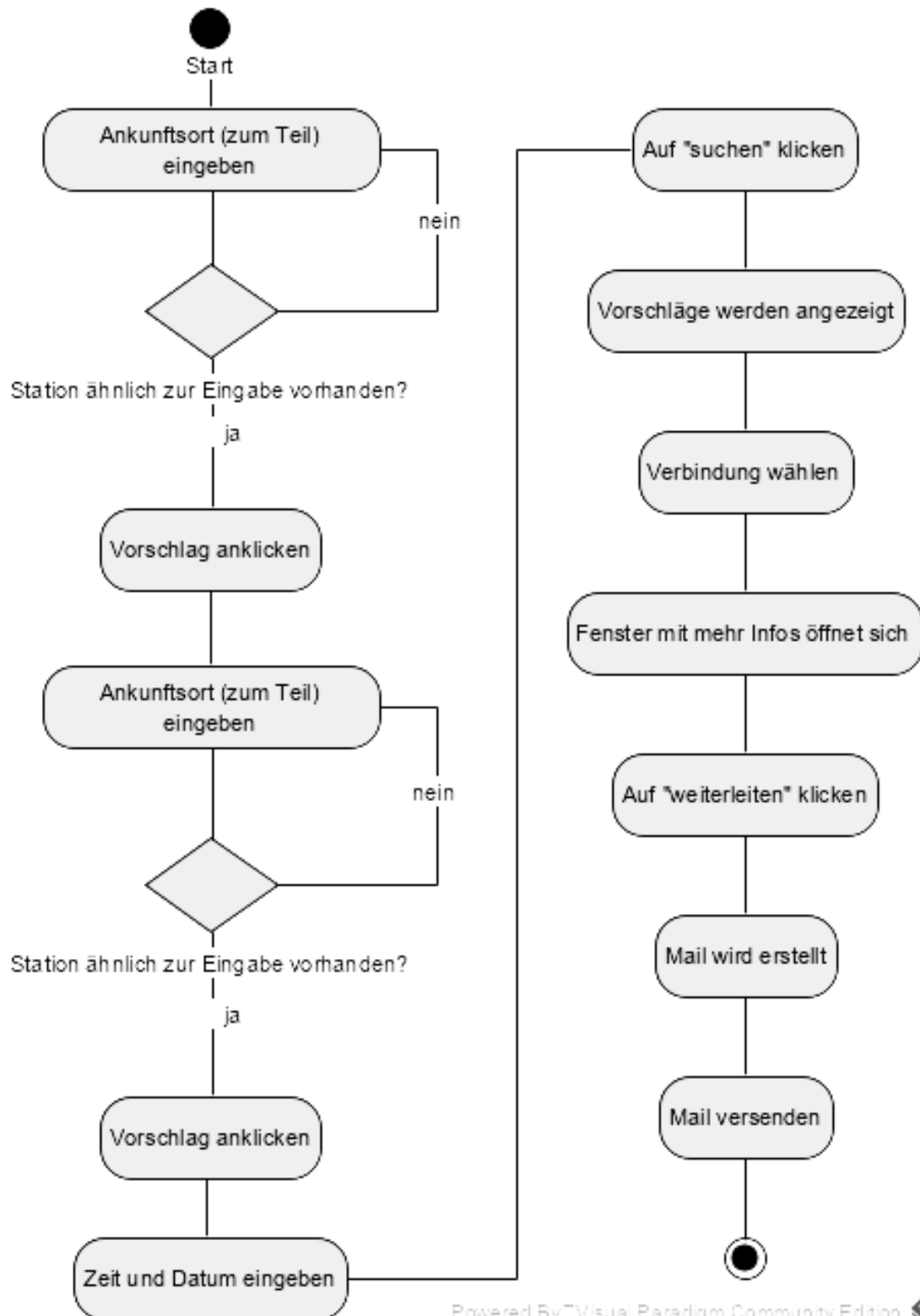
Use Case #01	Mind. 5 Verbindungen anzeigen
Beschreibung	Der Benutzer sucht eine Verbindung zwischen selbst gewähltem Abfahrts- und Ankunftsort. Es kann die aktuelle oder eine selbst gewählte Zeit bestimmt werden. Es sollen dann die nächsten fünf Verbindungen sichtbar sein.
Ablauf	<ol style="list-style-type: none">1. Abfahrtsort, Ankunftsort, Zeit eingeben2. Verbindung suchen3. Eine der angezeigten Verbindungen für mehr Infos anwählen
Ergebnis	Der Benutzer findet schnell eine passende Verbindung für seine Reise.
Use Case #02	Abfahrtstafel anzeigen

Beschreibung	Der Benutzer alle Verbindungen sehen, die zur gewählten Zeit vom gewählten Abfahrtsort fahren.
Ablauf	<ol style="list-style-type: none"> 1. Auf Abfahrtstafel-Maske wechseln 2. Abfahrtsort, Zeit eingeben 3. Verbindungen suchen 4. Eine der angezeigten Verbindungen für mehr Infos anwählen
Ergebnis	Der Benutzer sieht die Abfahrtstafel für den gewählten Abfahrtsort.

Use Case #03		Eingabe vervollständigen
Beschreibung	Der Benutzer möchte nach Stationen suchen können, damit er sich nicht alle Namen auswendig merken muss. Allenfalls kann die Vervollständigung bereits während dem Tippen geschehen.	
Ablauf	<ol style="list-style-type: none"> 1. Abfahrtsort / Ankunftsart eingeben 2. Eingabevervollständigung wird vorgeschlagen 3. Gesuchte Station anwählen 	
Alternativer Ablauf	<ol style="list-style-type: none"> 1. Auf Abfahrtstafel-Maske wechseln 2. Abfahrtsort eingeben 3. Eingabevervollständigung wird vorgeschlagen 4. Gesuchte Station anwählen 	
Ergebnis	Der Benutzer muss nicht den gesamten Namen der Station wissen und ihn abtippen. Er kann gemütlich einen Vorgeschlagenen Ort anklicken.	

4.3 Aktivitätsdiagramm

Dieses Aktivitätsdiagramm stellt den Ablauf des Benutzers dar, wenn er sich Verbindungen mit gewähltem Abfahrts- und Ankunftszeitpunkt anzeigen lässt und eine Verbindung anschliessend per Mail weiterleitet.



5. Programmierrichtlinien

5.1 Naming Conventions

- Variablen kurz, prägnant, logisch benennen
- camelCase verwenden
- Globale Variablen nicht redundant
- Methoden/Funktionen im Idealfall Verb und Nomen

5.2 Declaration

- Variablen immer direkt initialisieren
- Globale Variablen nach load Form deklarieren

5.3 Comments

- Code sollte selbsterklärend sein
- Kommentare nur wenn unbedingt nötig

6. Testing

6.1 Testfall «Hin- und Rückfahrt anzeigen lassen»

Schritt	Aktivität	Erwartetes Resultat	Tatsächliches Resultat	Status
1	User klickt in das Eingabefeld «von» und fängt an, seine gewünschte Start-Station einzugeben. → Stub...	In der Box unterhalb werden während dem Tippen Vorschläge angezeigt.	In der Box unterhalb werden während dem Tippen Vorschläge angezeigt.	✓
2	User klickt auf seine gewünschte Start-Station «Adligenswil, Stuben».	Der Stationsname wird in das Eingabefeld übernommen und die Vorschläge verschwinden.	Der Stationsname wird in das Eingabefeld übernommen und die Vorschläge verschwinden.	✓
3	User macht Schritte 1 & 2 mit der Ankunfts-Station «Luzern».	Programm verhält sich gleich wie bei den Schritten 1 & 2 beschrieben.	Programm verhält sich gleich wie bei den Schritten 1 & 2 beschrieben.	✓

4	User klickt auf den Button «Suchen».	Eine Liste mit den vier nächsten Verbindungen wird rechts angezeigt.	Eine Liste mit den vier nächsten Verbindungen wird rechts angezeigt.	✓
5	User klickt auf den Button «Rückfahrt».	Abfahrts- und Ankunftsstation werden getauscht und die nächsten vier Verbindungen werden angezeigt.	Abfahrts- und Ankunftsstation werden getauscht und die nächsten vier Verbindungen werden angezeigt.	✓

7. Installationsanleitung

Um die Applikation auszuführen, laden Sie das Zip-File vom Github-Repository herunter und führen Sie die Datei «Fahrplan\bin\Debug\Fahrplan.exe» aus.