***Christen-Travel Projekt Dokumentation***

# Analyse:

## Aktuelle Situation:

* Programm welche die Fahrplandaten des Schweizerischen öffentlichen Verkehrs benutzt
* Programm mit dem man Verkehrsverbindungen zwischen zwei Stationen suchen kann

## Ziel definieren:

* Eine funktionierende Applikation
* Anforderungen sollen Erfüllt sein
* Qualität-Ziele sollen erfüllt sein
* Hohe Qualität soll vorhanden sein

## Anforderungen ermitteln:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Beschreibung | Priorität | Erfüllt |
| A001 | Als Christen-Travel-Benutzer möchte ich Start- und Endstation mittels Textsuche suchen können, damit ich nicht alle Stationsnamen auswendig lernen muss. | 1 |  |
| A002 | Als Christen-Travel-Benutzer möchte ich die aktuellen, d.h. mindestens die nächsten 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 Christen-Travel-Benutzer möchte ich sehen, welche Verbindungen ab einer bestimmten Station vorhanden sind, damit ich bei mir zuhause eine Art Abfahrstafel haben kann. | 1 |  |
| A004 | Als Christen-Travel-Benutzer möchte ich, dass schon während meiner Eingabe erste Suchresultate erscheinen, damit ich effizienter nach Stationen suchen kann. | 2 |  |
| A005 | Als Christen-Travel-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 Christen-Travel-Benutzer möchte ich sehen, wo sich eine Station befindet, damit ich mir besser vorstellen kann, wie die Situation vor Ort aussieht. | 3 |  |
| A007 | Als Christen-Travel-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 |  |

## Qualitätsziele:

Funktionalität:

* Angeforderte Funktionalitäten sollen entsprechend laut Besprechung umgesetzt werden.
* Die Funktionen sollen korrekt umgesetzt sein.
* Die Funktionsweisse soll durch Testfälle sichergestellt sein.

Zuverlässigkeit:

* Die Software soll zuverlässig funktionieren und tun, was man von ihr erwartet.
* Die Software soll in Fehlerfällen den Benutzer korrekt auf Fehler hinweisen.
* Nach Auftreten eines Fehlers sollte die Software, wenn möglich, normal weiterlaufen.

Benutzbarkeit:

* Die Bedienung der Software soll selbsterklärend sein.
* Benutzereingaben sollen validiert werden.
* Die grafische Benutzeroberfläche soll übersichtlich gestaltet sein.
* Das GUI soll sich an UI Standards halten.

Wartbarkeit:

* Der Source Code soll sich an die Programmierrichtlinien halten.
* Variablen, Klassen, Methoden und andere Elemente sollen sprechende Namen haben.
* Der Source Code soll verständlich kommentiert sein, insbesondere sollen alle öffentlichen Methoden inklusive ihrer Parameter kommentiert sein.
* Es soll einfach und problemlos möglich sein, Änderungen an der Software zu machen. Dazu gehört: kein repetitiver Code, kein Spaghetticode, Beachtung des Kohäsionszprinzip und eine möglichst lose Kopplung der einzelnen Module und Klassen.

Portierbarkeit:

* Die Software soll einfach zu installieren sein.
* Die Software soll auch einfach wieder entfernt werden können.
* Die Software soll nach der Installation auch auf einem Rechner ohne Entwicklungsumgebung lauffähig sein.
* Zusammen mit der Software soll eine Installationsanleitung ausgeliefert werden.