

Fahrtroutenplanung

Pflichtenheft zum Softwareentwicklungsprojekt

Lehrveranstaltung „Software Engineering I und II“

1. Juli 2016

Entwickler: Paul Wolff, Dennis Hinterwimmer, Maxim Aisel

Auftraggeber: Prof. Dr.-Ing. habil. Hartmut Fritzsche

Bachelorstudiengang Medieninformatik
Hochschule für Technik und Wirtschaft Dresden

Inhaltsverzeichnis

1	Zielbestimmung	3
2	Produkteinsatz	4
2.1	Beschreibung des Problembereichs	4
2.2	Glossar	4
2.3	Modell des Problembereichs	5
2.4	Geschäftsprozesse	6
3	Produktfunktionen	7
3.1	Funktionale Anforderungen	7
3.2	Benutzeroberfläche	7
3.3	Produktdaten	7
4	Produktcharakteristika	8
4.1	Nichtfunktionale Anforderungen	8
4.2	Technische Produktumgebung	9
4.3	Gliederung in Teilprodukte	9
4.4	Anforderungen an die Entwicklungsumgebung	10
5	Anhänge	10

1 Zielbestimmung

Im Rahmen der Lehrveranstaltung Softwareengineering I bei Herrn Professor Fritzsche, erhielten wir als Gruppe aus drei Personen den Auftrag einen Fahrtoutenplaner zu entwerfen und zu entwickeln. Dieser muss in der Lage sein, ausgehend von einem vom Nutzer gewählten Lager, eine Route zu finden, die alle Lager miteinander verbindet. Die Route soll ähnlich einer Liste visualisiert werden und die einzelnen Lager nachfolgend anzeigen. Zusätzlich soll die Wegverbindung in Kilometern angezeigt werden. Die Daten liegen in Form einer Menge von Lagern, mit Verbindungen untereinander, vor.

2 Produkteinsatz

2.1 Beschreibung des Problembereichs

Der Fahrtoutenplaner soll es dem User ermöglichen auf einfachste Weise Orte miteinander zu verbinden und hierbei die gewünschte Route zu finden. Dies findet im Logistikbereich eine sinnvolle Anwendung. Das hier dargestellte Programm findet seinen Einsatz in der Lehre und Forschung. Es soll den User in die Lage versetzen, Strecken besser einzuschätzen und Routen zu planen und dient somit der Organisation und Logistik. Die Applikation soll zuverlässig und aktuell sein, hierzu wird eine Kommunikation zwischen Endgerät und Server aufgebaut. Da er Routenplaner sowohl zu Forschungszwecken, als auch zu Lehrzwecken dienen soll, wird eine möglichst intuitive Handhabung von Nöten sein.

2.2 Glossar

- Route
 - Eine festgelegter Streckenverlauf
- User
 - Der Benutzer des Programmes
- Lager
 - Ort zur Organisation von Rohstoff-, Betriebs- oder Fertigteilen
- Wegverbindungen
 - Streckenverbindung der Lager
- Pulldown-Menü
 - Menü zum ausklappen
- Applikation
 - Hiermit ist der Fahrtoutenplaner gemeint
- Button
 - Visueller und Virtueller Knopf zur Interaktion mit dem Programm

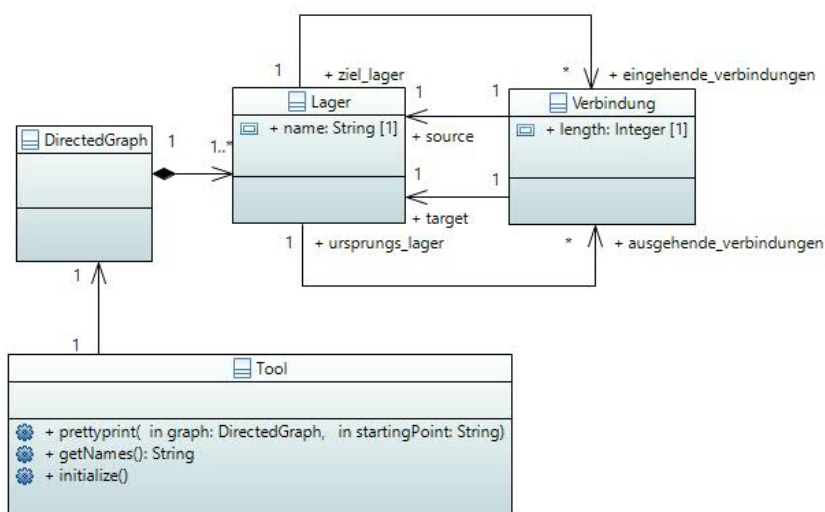
- Distanz
 - Entfernung von zwei oder mehr Lagern
- Server
 - Hard- und Softwaremäßige Lösung zum Bereitstellen der Daten über Netzwerk und Internet

2.3 Modell des Problembereichs

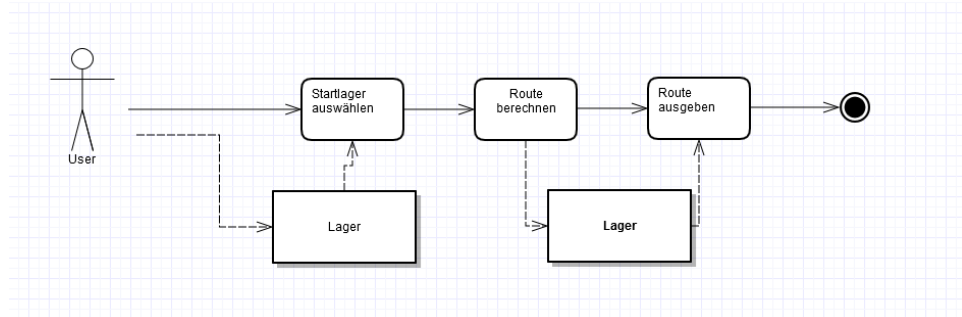
- Client:



- Server:



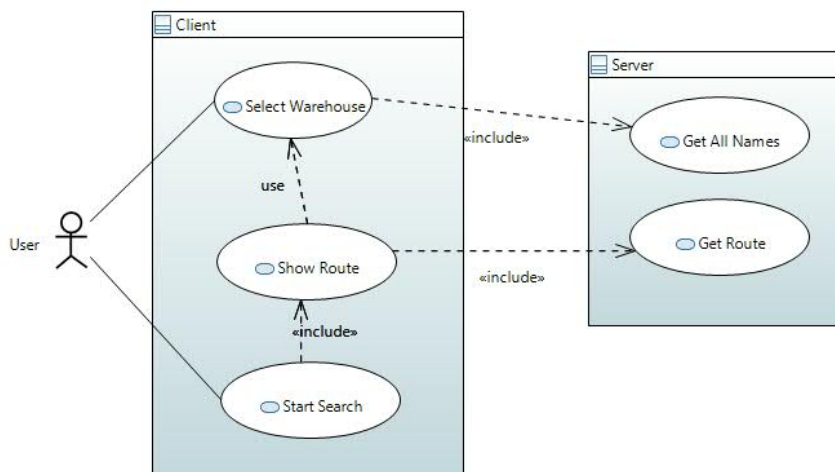
2.4 Geschäftsprozesse



3 Produktfunktionen

Anhand eines Dropdown Menüs ist der Nutzer in der Lage aus einer Liste vorgegebener Lager zu wählen, welches dann als Startpunkt der Routenplanung fungiert. Über einen Button kann er nach der Auswahl die Routenplanung starten, welche dann im Programm visualisiert wird. Die Visualisierung erfolgt über eine Liste, in der die Folge der Lager aufgeführt wird, außerdem wird die insgesamt zurückgelegte Strecke in Kilometern dargestellt.

3.1 Funktionale Anforderungen



3.2 Benutzeroberfläche

Eingabe:	Visualisierung:
<div> <div>Lager wählen ... ▾</div> <div> Lager 1 Lager 2 Lager 3 Lager 4 ... </div> </div> <div>Start</div>	<div> <div> Lager 3 Lager 1 Lager 5 ... </div> </div> <div>Gesamtlänge: 153km</div>

3.3 Produktdaten

Die Speicherung der Daten, insbesondere der verschiedenen Lager, wird mittels Objektserialisierung realisiert.

4 Produktcharakteristika

Das entwickelte Produkt ist "Open Source" und wird für Forschung und Lehre eingesetzt. Eine kommerzielle Nutzung des Produktes ist ausgeschlossen. Die entwickelte Software wird an der HTW Dresden benutzt.

4.1 Nichtfunktionale Anforderungen

Zeit:

Bezeichnung	Datum	Beschreibung
1.Meilenstein	27.01.2016	Entwurf des Pflichtenheftes
2.Meilenstein	09.05. - 14.05.2016	Kontrolle des Projektfortschritts / Überprüfung der Erfolgsaussichten
3.Meilenstein	17.06.2016	Übergabe vom Entwicklungsprojekt an den AG
Verteidigung	20.06. - 02.07.2016	Präsentation der entwickelten Anwendung

Genauigkeit: Entfernungen in km Die Länge der Fahrtroute in km ist zu berechnen

Organisatorische Randbedingungen: Subversion-Versionsverwaltung

Das Entwicklungsprojekt muss folgende Verzeichnisse enthalten:

- models : Modelle (UseCase-, Sequenz-, Klassen-, Paket-, Zustands-, Verteilungsdiagramme)
- src : Annotierte Quelltexte test : JUnit-Testskripte entsprechend der Struktur des src-Verzeichnisses scripts : Build-Skripte (ANT) mit Zielen zur Erzeugung - der API-Spezifikation, - des Produktes mit Quellen und - des Produktes ohne Quellen nach Anwendung eines Obfuscators Generierte API-Spezifikation
- doc : Generierte API-Spezifikation
- documentation : Pflichtenheft, Entwicklerdokumentation inklusive Anwenderdokumentation und Installationsanleitung (gemäß vorgegebener Gliederung). Die Dokumentation muss eine Beschreibung der Konfiguration auf einem Zielrechner enthalten (Deployment).
- thirdparty : Nicht selbst entwickelte, aber aus anderen Quellen nachgenutzte Artefakte (soll ggf. die gleiche Verzeichnisstruktur aufweisen wie das Projekt selbst)

Technische Randbedingungen: Zur Entwicklung wird ausschließlich "Open Source"-Software eingesetzt.

Qualitätsanforderungen

	wichtig	normal	unwichtig
Funktionalität	x		
Zuverlässigkeit		x	
Effizienz			x
Benutzerfreundlichkeit		x	
Wartbarkeit		x	

4.2 Technische Produktumgebung

"Das zu entwickelnde Softwaresystem soll als Client-Server-System mit einer Schichtenarchitektur realisiert werden." Quelle: [Lastenheft]

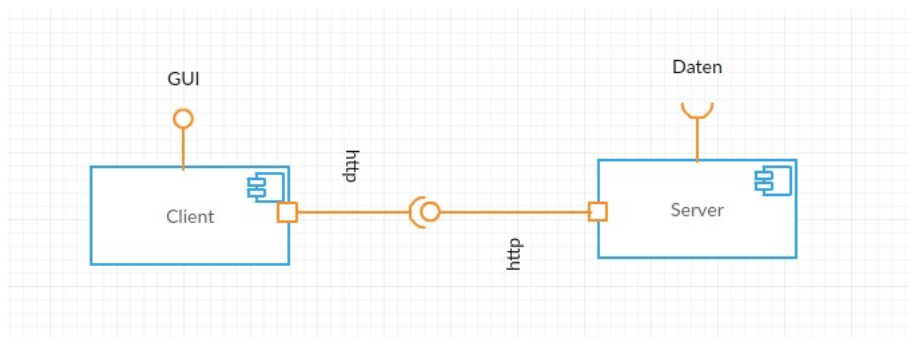
Software

- Client: Java unterstützung
- Server: Java unterstützung, ein Servlet

Hardware

- Client: Java fähiger Rechner mit Bildschirm, Internetzugang und freier Festplattenkapazität je nach Software-Umfang vorausgesetzt.
- Server: Java fähiger Rechner mit Internetzugang und freier Festplattenkapazität je nach Software-Umfang vorausgesetzt.

4.3 Gliederung in Teilprodukte



4.4 Anforderungen an die Entwicklungsumgebung

- **Java:** Als Implementierungssprache ist Java zu verwenden.
- **Eclipse-EE** mit Plug-in **subclipse**: Das Entwicklungsprojekt soll in Eclipse realisiert werden und eine Subversion Versionsverwaltung nutzen.
- **SVN-Repository:** soll unter `svn://141.56.139.110/se/Belege-2016/FAP-2` gepflegt werden.
- **Javadoc:** API-Spezifikationen sind mit Javadoc zu erstellen.
- **UML-Diagramme:** zur Erstellung wird Papyrus verwendet.
- **JUnit:** Testscripte entsprechend der Struktur des src-Verzeichnisses
- **Build-Skripte (ANT):** Erzeugung
 - der API-Spezifikatione
 - des Produktes mit Quellen
 - des Produktes ohne Quellen nach Anwendung eines Obfuscators

5 Anhänge