

Projekt: Paketlieferservice

Pflichtenheft

Projektbezeichnung	Paketlieferservice
Projektleiter	Gabriel Goller
Erstellt am	17.12.2020
Letzte Änderung am	18.12.2020
Status	in Bearbeitung
Aktuelle Version	2

Änderungsverlauf

Nr.	Datum	Version	Änderung	Autor
1	17.12.2020	1	Algorithmen	Lukas S.
2	18.12.2020	2	Formatierung	Gabriel G.

Team

Rolle(n)	Name	Github	E-Mail
Projektleiter	Gabriel Goller	@kaffarell	stgolgab@bx.fallmerayer.it
Programmierer	Daniel Bosin	@Kurusu003	stbosdan@bx.fallmerayer.it
Programmierer	Lukas Schatzer	@zLuki	stschluk2@bx.fallmerayer.it
Programmierer	Philipp Olivotto	@zbaakez	stoliphi@bx.fallmerayer.it
Designer	Alex Unterleitner	@LaxeChef	stuntale@bx.fallmerayer.it

Einlesen der Adjazenzmatrix:

Zum Einlesen der Adjazenzmatrix aus einer CSV-Datei, benutzen wir die Klasse CSVReader. Dabei wird die CSV-Datei geöffnet und die Matrix wird in einem zweidimensionalen Array gespeichert. Mit einer Methode gibt die Entfernung zwischen zwei Orten zurück.

Die grafische Darstellung als UML-Klassendiagramm finden Sie in /UML.

to be continued

Berechnen der bestmöglichen Route:

Hier müssen wir unterscheiden, aus wie vielen Orten die Route besteht:

Mit folgender Formel kann man sich die Anzahl der möglichen Routen berechnen::

$$\frac{(n-1)!}{2}$$

Bei relativ wenigen Orten verwenden wir ein Brute-force Algorithmus, um mit 100%iger Sicherheit die effizienteste Route zu bestimmen.

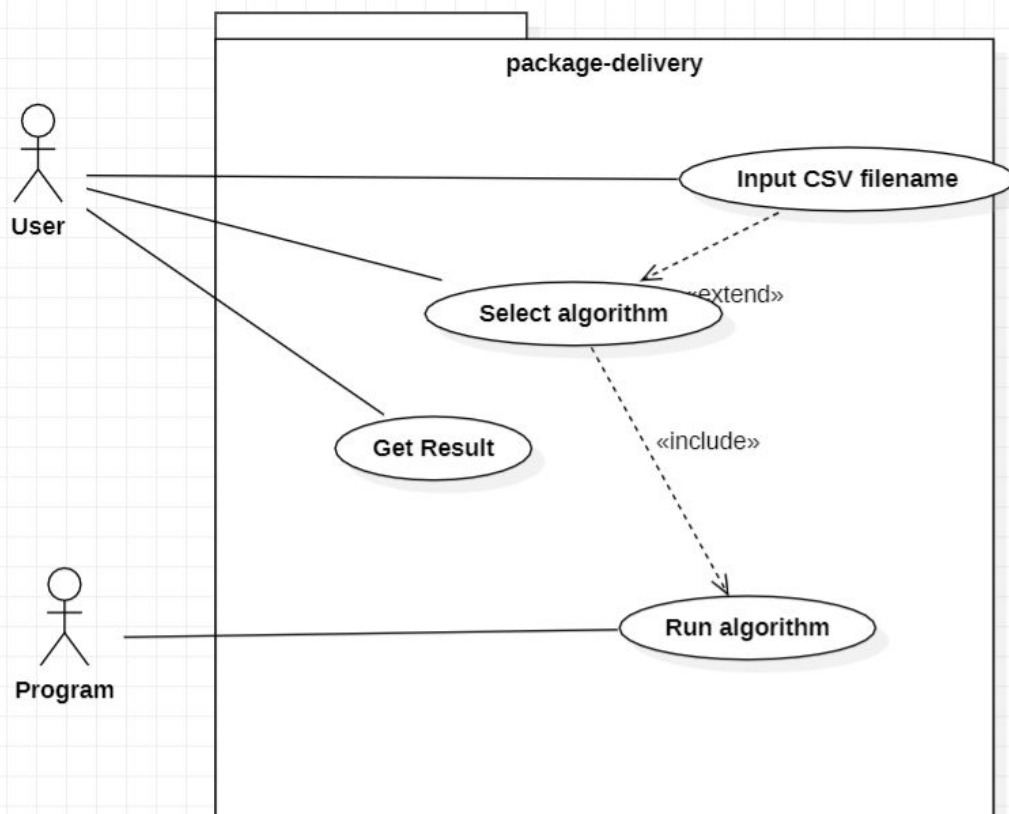
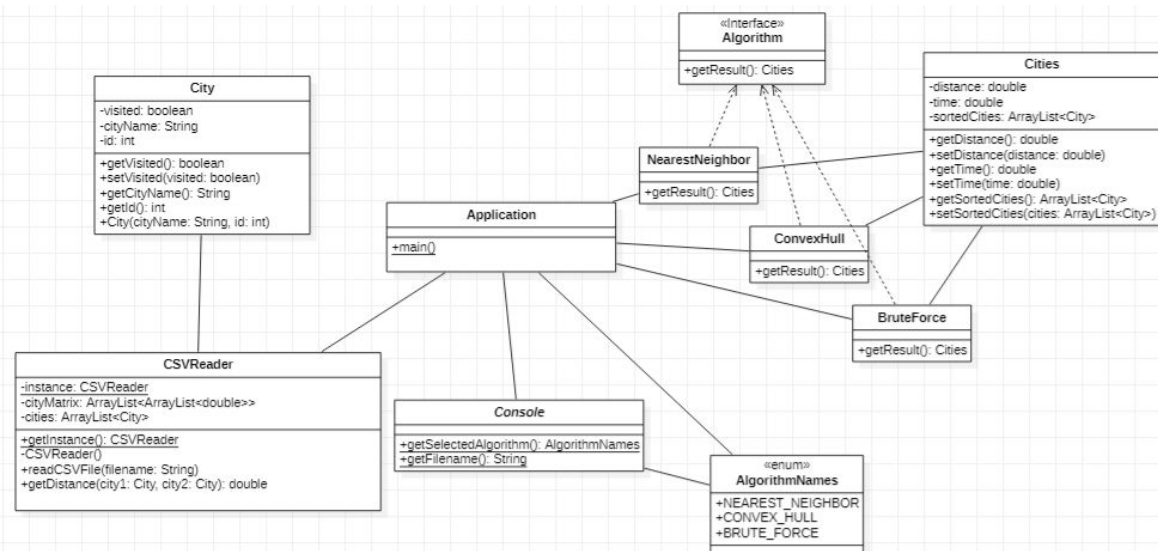
Bei vielen Orten würde ein Brute Force Algorithmus viel zu lange dauern, deshalb verwenden wir ein heuristisches Verfahren. Hier eignen sich zwei Algorithmen sehr gut, der Nearest-Neighbor Algorithmus und der Convex Hull Algorithmus.

Beide haben Vor- und Nachteile, Nearest-Neighbor ist schnell und einfach zu implementieren, Convex Hull ist etwas langsamer, deutlicher schwerer zu implementieren, dafür liefert er aber etwas bessere Ergebnisse.

Wir haben drei Klassen für die Algorithmen erstellt:

- Bruteforce

- NearestNeighbor
- ConvexHull



<https://tspvis.com/>

<https://stackoverflow.com/questions/11703827/brute-force-algorithm-for-the-traveling-salesman-problem-in-java>

