



PFLICHTENHEFT

Bartolini 2.0 GmbH

Projekt: Paketlieferservice

Erstellung eines Programms für einen Paketboten, damit er den kürzesten Weg beim Ausliefern der Pakete zurücklegt und damit Zeit einspart

Pflichtenheft

Projektbezeichnung	Bartolini 2.0
Projektleiter	Silas Demez
Erstellt am	18.12.2021
Letzte Änderung am	04.01.2021
Status	in Bearbeitung
Aktuelle Version	1.5

Änderungsverlauf

Nr.	Datum	Version	Geänderte Kapitel	Art der Änderung	Autor
1	18.12.2020	1.1	Alle	Erstellung	Silas Demez
2	03.10.2018	1.2	3-8	Ergänzung	Silas Demez
3	13.01.2021	1.3	3,6,8	Ergänzung	Silas Demez
4	14.01.2021	1.4	3,6	Ergänzung	Silas Demez
5	02.02.2021	1.5	1,3,5,6	Ergänzung	Gabriel Valentin

Inhalt

1	Einleitung	3
2	Allgemeines	3
2.1	Ziel und Zweck des Dokuments	3
2.2	Ausgangssituation	3
2.3	Teams und Schnittstellen	3
2.4	Begriffserklärungen	4
3	Konzept	4
3.1	Ziel(e) des Anbieters	4
3.2	Ziel(e) und Nutzen des Anwenders.....	4
3.3	Zielgruppe(n)	4
3.4	Systemarchitektur	5
4	Funktionale Anforderungen	5
4.1	Einlesen der Route	5
4.2	Auffinden einer möglichst kurzen Tour.....	5
5	Nichtfunktionale Anforderungen	5
6	Rahmenbedingungen	6
6.1	Zeitplan	6
6.2	Technische Anforderungen	7
6.3	Problemanalyse	7
6.4	Qualität	7
7	Liefer- und Abnahmebedingungen.....	7
8	Anhang	7
8.1	Quellen	7

1 Einleitung

Bartolini 2.0 stellt ein Programm dar, welches den Zweck hat dem Paketboten die Arbeit zu erleichtern. Das Programm berechnet den kürzesten Weg zwischen gegebene Wegpunkte, zu denen der Paketbote hinfahren muss, um die Pakete zuzustellen.

2 Allgemeines

2.1 Ziel und Zweck des Dokuments

Dieses Pflichtenheft beschreibt die Aufgaben und Funktionen, zu denen sich die Programmierer unter Leitung des Projektmanagers Silas Demez verpflichten. Bei Nichterfüllung der Anforderungen muss der Auftragnehmer 404 Not Found GmbH die Folgen tragen.

2.2 Ausgangssituation

Die Firma Bartolini 2.0 GmbH, welche bis vor kurzem noch mit traditionellen Mitteln Pakete zustellte, hat sich dazu entschieden ihre Firma zu modernisieren. Neben neuer Ausrüstung gehört auch die Einführung einer Hilfssoftware für Paketboten dazu, damit diese den schnellsten und kürzesten Weg zum Zustellen der Pakete vorgegeben bekommen.

2.3 Teams und Schnittstellen

Das Team von 404 Not Found GmbH besteht aus 5 Mitgliedern. Unten sind diese mit deren Rollen und Kontaktdaten angeführt:

Rolle(n)	Name	GitHub	E-Mail
Projektleiter/-manager	Silas Demez	SilasDemez	stdemsil@bx.fallmerayer.it
Chefingenieur, Programmierer	Gabriel Valentin	Gavaii	stvalgab@bx.fallmerayer.it
Web Developer, Programmierer	Noa Pichler	Git_Goat	stpicnoa@bx.fallmerayer.it
Programmierer	Nathan Obexer	obenat	stobenat@bx.fallmerayer.it
Programmiererin	Anna Kaserer	annaksrr	stkasann@bx.fallmerayer.it

Da das Programm in Java entwickelt wird, ist die Schnittstelle privat und kann nur von denen bearbeitet werden, die den Zugriff auf den Quellcode haben.

2.4 Begriffserklärungen

Siehe [Glossar](#)

3 Konzept

3.1 Ziel(e) des Anbieters

404 Not Found GmbH will ein benutzerfreundliches und einfaches Programm entwickeln. Dabei soll dieses Programm so schnell wie möglich den besten Weg liefern.

Das Ganze soll so funktionieren:

1. Der Paketbote hat eine CSV-Datei, in der die Punkte im Format einer Adjazenzmatrix eingetragen sind. Aus dieser Datei kann man den Abstand zwischen den verschiedenen Punkten auslesen.
2. Der Paketbote übergibt dem Programm die CSV-Datei (übergibt den Dateinamen/Dateipfad)
3. Eine Methode des Programms wandelt die Adjazenzmatrix in eine Liste von Punkten um. Diese werden im Format eines 2-dimensionalen Arrays abgespeichert.
4. Das Programm wählt anhand der Anzahl der Punkte den dazu geeigneten Algorithmus
5. Das 2-dimensionale Array, welches die Punkte beinhaltet wird dem ausgewählten Algorithmus mitgegeben
6. Der Algorithmus berechnet den kürzesten Weg
7. Das Programm gibt die Orte in geordneter Reihenfolge aus

Siehe auch:

- [Sequenzdiagramm](#)

3.2 Ziel(e) und Nutzen des Anwenders

Der Anwender gibt dem Programm eine Reihe von Orten und bekommt die kürzeste Strecke zwischen den Punkten zurück. Das Programm soll dem Anwender hilfreich sein, die Pakete, die er in seinem Lieferwagen mitbringt, effizient und schnell auszuliefern.

3.3 Zielgruppe(n)

Die Zielgruppe bzw. die Anwender sind die Paketboten. Diese müssen währendem sie fahren sehen, wo sie als nächstes hinfahren müssen. Das heißt also, dass das Programm die Aufmerksamkeit der Paketboten nicht zu sehr anziehen soll. So wäre z.B.: eine überdimensionale Schriftgröße wichtig.

3.4 Systemarchitektur

Das System besteht aus mehreren Klassen. Jede Klasse ist für jeweils einen Schritt zuständig. So ist die Klasse ‚ReadCSV‘ für das Einlesen der Datei zuständig, worin sich die Punkte befinden. Weiters gibt es die Klasse ‚NearestNeighbor‘ und ‚GrahamAlgorithmus‘, welche für den gleichnamigen Algorithmus zuständig sind. Die Klasse, die alles zusammenführt, ist die ‚Application‘ – Klasse.

Siehe auch:

- [Algorithmenvergleich](#)
- [Klassendiagramm](#)

4 Funktionale Anforderungen

Unten sind die geforderten Anforderungen angeführt

4.1 Einlesen der Route

Das Programm soll eine Möglichkeit anbieten eine CSV-Datei einzulesen, welche eine Liste der zu besuchenden Orte und alle Entfernungen zwischen den Orten enthält (Adjazenzmatrix).

4.2 Auffinden einer möglichst kurzen Tour

Eine Tour oder Hamiltonkreis sollte so kurz wie möglich sein. Dafür können verschiedene Algorithmen verwendet werden. Empfohlen ist die Verwendung von heuristischen Verfahren (Ameisenalgorithmus oder ein genetischer Algorithmus).

5 Nichtfunktionale Anforderungen/Zusatzfeatures

Falls alle notwendige Anforderungen erledigt sein sollten, wären diese Zusatzfeatures vom Arbeitgeber gefragt:

- Interaktive Website, basierend auf Java, die eine Weltkarte darstellt. Diese Webseite soll die Möglichkeit besitzen Punkte manuell zu setzen (Mausinput). Die Punkte sollen dem Algorithmus gesendet werden, der die möglichst kürzeste Distanz berechnet.
- Mehrere Algorithmen sollten dem Arbeitgeber zur Verfügung gestellt werden, um die Distanz effizienter und schneller zu berechnen oder die Distanz genauer zu berechnen.

- Das Programm sollte möglichst gut dokumentiert sein. Die Kommentierung sollte möglichst aus Java Docs bestehen, wobei die Java Docs auf der Website gehostet/heruntergeladen werden können.

6 Rahmenbedingungen

Die Entwicklung des Projekts erfolgt auf Grundlage des [Wasserfallmodells](#) in zwei Meilensteinen:

1. Meilenstein - Vorbereitungen und System-Architektur

- Recherche und Analyse des allgemeinen Problems
- Erstellung eines Glossars zu den wichtigsten Begriffen
- Beschreibung einer allgemeinen Einteilung der Lösungsansätze
- Wahl eines exakten Algorithmus und beispielhafte Anwendung an einem kleinen Beispiel (Implementation in Java)
- Auswahl eines heuristischen Algorithmus inklusive Begründung
- Erstellung Pflichtenheft, Zeitplanung
- System-Architektur

2. Meilenstein - Tests und Implementation

- JUnit-Tests
- Integrationstests
- Implementation
- Javadoc und Webseite

6.1 Zeitplan

404 Not Found GmbH verpflichtet sich dazu die Meilensteine innerhalb dieses Zeitpunktes fertig zu haben:

- 1. Meilenstein – 15. Januar
- 2. Meilenstein – 27. Januar

Pro Meilenstein nimmt sich jedes Mitglied des Entwicklerteams ca. 6 Stunden Zeit, um am Projekt weiterzuarbeiten.

404 Not Found GmbH nutzt zur Zeitplanung das Aufgabenverwaltungstool [Trello](#). Damit kann den verschiedenen Teammitgliedern eine Aufgabe zugewiesen werden.

Will man den Fortschritt bei der zugewiesenen Aufgabe überwachen, kann man auf [GitHub](#) nachschauen.

Siehe auch:

- [Zeitplan](#)

Siehe [Glossar](#) für eine Kurzbeschreibung von Trello und GitHub.

6.2 Technische Anforderungen

Damit das ganze Programm funktioniert, ist ein Computer mit installiertem Java notwendig. Mobilgeräte wie z.B.: Handys und Tablets sind dabei nicht eingeschlossen. Außerdem müssen verschiedene Bibliotheken eingebunden werden.

Sollte man die Website benutzen, dann sollte die Effizienz der Algorithmen gleich bleiben, der Vorteil ist, dass man nichts herunterladen/einstellen muss. Die Website funktioniert, aber nur unter der Bedingung, dass ein Server mit den Technischen Anforderungen die Website hostet.

6.3 Problemanalyse

Beim Einlesen der Orte werden wahrscheinlich die meisten Probleme auftreten, da der Anwender manchmal eine falsche Eingabe macht. Geplant ist, dass die falschen Eingaben entdeckt werden und dass System den Anwender auffordert die Eingabe auszubessern. Die Kommunikation zwischen der Website und dem Java-Programm wird wahrscheinlich nicht problemlos geschehen. Außerdem könnten Probleme bei der Konvertierung der Koordinaten zu Adjazenzmatrix und umgekehrt entstehen.

6.4 Qualität

Das Programm soll einwandfrei funktionieren und auch für den nicht-computerfreundlichen Nutzer verständlich sein. Fehler sollten nie auftreten.

7 Liefer- und Abnahmebedingungen

Das Programm wird den Technologie- und Planung-Lehrern der Klasse 4BT abgegeben. Dabei passiert das ganze unentgeltlich. Nach der Abgabe beginnt die Wettbewerbsphase, wo alle Gruppen ihr Programm gegeneinander antreten lassen. Das Programm, welches funktioniert und die wenigste Zeit braucht gewinnt.

8 Anhang

8.1 Quellen

Stand 13.01.2021

- https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden
- [https://de.wikipedia.org/wiki/NP_\(Komplexit%C3%A4tsklasse\)](https://de.wikipedia.org/wiki/NP_(Komplexit%C3%A4tsklasse))
- <https://de.wikipedia.org/wiki/Adjazenzmatrix>
- https://de.wikipedia.org/wiki/Kombinatorische_Optimierung

- <https://de.wikipedia.org/wiki/Hamiltonkreisproblem>
- <https://de.wikipedia.org/wiki/Ameisenalgorithmus>
- <https://tspvis.com>
- <https://www.ionos.de/digitalguide/websites/web-entwicklung/pflichtenheft/>
- <https://www.youtube.com/watch?v=pA7yeEV1CEo>
- <https://hobbyblogging.de/wp-content/uploads/2016/10/Wasserfalldiagramm.jpg>