

Metro Simulatie

Documentsoort:	Behoeftespecificatie
Versie:	2.0
Datum:	20 maart 2017
Auteurs:	Brent van Bladel
Status:	Opgeleverd

1 Samenvatting

Dit document bevat de specificaties voor een informaticasysteem ter ondersteuning van een metrosimulatie. Het is geschreven in het kader van het vak “Project Software Engineering” (1ste bachelor informatica - Universiteit Antwerpen).

2 Context

Sinds 1 februari 2017 is de stad Antwerpen in de kernstad en op linkeroever een lage-emissiezone. In deze lage-emissiezone mogen enkel voertuigen rijden die aan bepaalde milieucriteria voldoen. Aangezien een heleboel voertuigen daarom niet langer de stad in mogen, verwacht de gewestelijke vervoersmaatschappij De Lijn een toename in het aantal passagiers. Het is voor De Lijn van groot belang op voorhand een duidelijk beeld te hebben van de situatie (qua bezetting en piekmomenten) van het metro net. Daarom heeft De Lijn geopteerd een simulatie model te laten ontwikkelen dat het tramverkeer kan simuleren.

De Universiteit Antwerpen is gevraagd dit systeem te ontwikkelen. In de eerste bachelor informatica zal onder de vakken “Computer Graphics” en “Project Software Engineering” gewerkt worden aan dit project. Tijdens de practica Computer Graphics zal de visualisatie van de simulatie ontwikkeld worden, tijdens de practica Project Software Engineering zal gewerkt worden aan de simulatie applicatie zelf.

3 Legende

De behoeftespecificatie is opgesteld aan de hand van zogenaamde use-cases. Elke use-case beschrijft een klein gedeelte van de gewenste functionaliteit. Het is de bedoeling dat tijdens elke fase van het project verschillende van die use cases geïmplementeerd worden. Een typische use-case bevat de volgende onderdelen:

- **Refertenummer & titel:**

Wordt gebruikt om naar een bepaalde use-case te verwijzen.

- **Prioriteit:**

De specificatie van een systeem vraagt meer dan wat binnen de voorziene tijd op te leveren is. Vandaar dat we per use-case aangeven in hoeverre die functionaliteit belangrijk is. In volgorde van belangrijkheid kan hier staan: VERPLICHT (deze use-case moet opgeleverd worden), BELANGRIJK (niet essentieel maar bij voorkeur toch opleveren), NUTTIG (interessant maar kan weggelaten worden).

- **Doel:**

Summiere beschrijving van het waarom van de use-case, t.t.z. wat de use-case bijdraagt tot de gehele functionaliteit.

- **Preconditie:**

Summiere beschrijving van de uitgangspunten bij aanvang van de use-case.

- **Succesvol einde:**

Summiere beschrijving van wat opgeleverd zal worden als er niks fout is gegaan.

- **Stappen:**

Een sequentiële beschrijving van hoe de use-case precies zal verlopen als alles goed gaat (het zogenaamde "happy day scenario"). De stappen zijn genummerd en kunnen controle instructies (WHILE, IF, ...) bevatten.

- **Uitzonderingen:**

Een lijst van mogelijke probleemgevallen en hoe die behandeld zullen worden. Een probleem geval (a) verwijst naar het nummer van de stap waar het probleem kan optreden, (b) bevat een conditie die aangeeft wanneer het probleemgeval optreedt, (c) omschrijft heel kort (een lijn) hoe het probleem behandeld zal worden.

- **Voorbeeld:**

Een voorbeeld van wat in- of uitgevoerd kan worden.

Soms is een use-case een uitbreiding van een andere use-case, en dan zijn volgende onderdelen relevant:

- **Uitbreiding:**

Een referte naar de use-case waarvan deze een uitbreiding is.

- **Stappen:**

Een lijst van extra en/of aangepaste stappen t.o.v de use-case waarvan deze een uitbreiding is.

Een uitbreiding (a) verwijst naar het nummer van de stap die uitgebreid wordt, (b) zegt of de uitbreiding voor, na of tijdens de normale stap zal gebeuren, (c) omschrijft wat precies in de uitbreiding zal gebeuren.

4 Overzicht

Use-Case	Prioriteit
<i>1: Invoer</i>	
1.1 Trams en stations inlezen	VERPLICHT
1.2 Passagiers inlezen	VERPLICHT
1.3 Trams en station met type inlezen	VERPLICHT
1.4 Trams met voertuignummer inlezen	BELANGRIJK
1.5 Stations met meerdere sporen inlezen	NUTTIG
<i>2: Uitvoer</i>	
2.1 Simpele uitvoer	VERPLICHT
2.2 Simpele uitvoer (herzien)	VERPLICHT
2.3 Grafische impressie	BELANGRIJK
2.4 Integratie met Graphics	BELANGRIJK
<i>3: Simulatie</i>	
3.1 Rijden van trams	VERPLICHT
3.2 Op- en afstappen van passagiers	BELANGRIJK
3.3 Op- en afstappen van passagiers (herzien)	VERPLICHT
3.4 Automatische simulatie	VERPLICHT
3.5 Rijden van meerdere trams	BELANGRIJK
3.6 Botspreventie	BELANGRIJK
3.7 Routeplanner passagiers	NUTTIG
3.8 Omzet per tram	NUTTIG
3.9 Rijden met reële snelheid	NUTTIG
3.10 Statistische verwerking simulatie	NUTTIG
<i>4: Gebruikersinterface</i>	
4.1 GUI voor rijden van trams	NUTTIG
4.2 GUI voor beheren van passagiers	NUTTIG
4.3 GUI voor statistische gegevens	NUTTIG

1.1. Trams en stations inlezen

Prioriteit:

VERPLICHT

Doel:

Inlezen van het schema van het metro net. De verschillende stations, hoe die met elkaar verbonden zijn en de verschillende trams.

Preconditie:

Een ASCII bestand met daarop een beschrijving van de stations en trams. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:

Het systeem bevat een spoorschema met de verschillende stations, en informatie over alle trams.

Stappen:

1. Open invoerbestand
2. WHILE Bestand niet ingelezen
 - 2.1. Herken het soort element (STATION, TRAM)
 - 2.2. Lees verdere informatie voor het element
 - 2.3. IF Verifieer geldige informatie
 - 2.3.1. THEN Voeg element toe aan het virtuele metronet
 - 2.3.1. ELSE Foutboodschap + positioneer op volgende element in het bestand
3. Verifieer consistentie van het metronet
4. Sluit invoerbestand

Uitzonderingen:

- 2.1. [Onherkenbaar element] Foutboodschap + positioneer op volgende element in het bestand \Rightarrow verdergaan vanaf stap 2
- 2.2. [Ongeldige informatie] Foutboodschap + positioneer op volgende element in het bestand \Rightarrow verdergaan vanaf stap 2
3. [Inconsistent metronet] Foutboodschap \Rightarrow verdergaan vanaf stap 4

Voorbeeld:

Een metronet met drie stations (A,B,C), een spoor (12) en een tram (12).

```
<STATION>
  <naam>A</naam>
  <volgende>B</volgende>
  <vorige>C</vorige>
  <spoor>12</spoor>
</STATION>
<STATION>
  <naam>B</naam>
  <volgende>C</volgende>
  <vorige>A</vorige>
  <spoor>12</spoor>
</STATION>
<STATION>
  <naam>C</naam>
  <volgende>A</volgende>
  <vorige>B</vorige>
  <spoor>12</spoor>
</STATION>
<TRAM>
  <lijnNr>12</lijnNr>
  <zitplaatsen>32</zitplaatsen>
  <snelheid>60</snelheid>
  <beginStation>A</beginStation>
</TRAM>
```

1.2. Passagiers inlezen

Prioriteit:

VERPLICHT

Doel:

Inlezen van passagiers met eigenschappen zoals een naam, hoeveelheid, begin- en eindstation.

Preconditie:

Een ASCII bestand met daarop een beschrijving van de passagiers. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:

De passagiers zijn ingelezen en bevinden zich in hun opstap station.

Stappen:

1. Open invoerbestand
2. WHILE Bestand niet ingelezen
 - 2.1. Herken het PASSAGIER element
 - 2.2. Lees verdere informatie voor het element
 - 2.3. Voeg element toe aan het virtuele metronet
3. Verifieer consistentie van het metronet
4. Sluit invoerbestand

Uitzonderingen:

- 2.1. [Onherkenbaar element] Foutboodschap + positioneer op volgende element in het bestand \Rightarrow verdergaan vanaf stap 2
- 2.2. [Ongeldige informatie] Foutboodschap + positioneer op volgende element in het bestand \Rightarrow verdergaan vanaf stap 2
3. [Inconsistent metronet] Foutboodschap \Rightarrow verdergaan vanaf stap 4

Voorbeeld:

bestand: passagiers.xml

```
<PASSAGIER>
  <naam>Groep1</naam>
  <beginStation>A</beginStation>
  <eindStation>C</eindStation>
  <hoeveelheid>2</hoeveelheid>
</PASSAGIER>
<PASSAGIER>
  <naam>Groep2</naam>
  <beginstation>C</beginstation>
  <eindstation>A</eindstation>
  <hoeveelheid>7</hoeveelheid>
</PASSAGIER>
```


1.3. Trams en station met type inlezen

Prioriteit:VERPLICHT

Doel:

Sinds mei 2015 heeft DeLijn een nieuw model van trams aangekocht: de Albatros. Het doel is om tegen 2018 alle oude PCC-trams te vervangen. Momenteel zitten we echter in een overgangsfase waar beide types tram tegelijk in gebruik zijn. Om tot een meer realistische simulatie te komen moet het mogelijk zijn onderscheid te maken tussen de twee types tram. De albatros tram heeft een lengte van 42 meter, wat aanzienlijk langer is dan de oude PCC-trams. Hierdoor kunnen ze niet stoppen aan bovengrondse haltes, aangezien het autoverkeer dan gehinderd wordt. Daarom moet het ook mogelijk zijn onderscheid te maken tussen twee types stations: het ondergrondse 'metrostation' en een bovengrondse 'halte'.

Referenties:

[https://nl.wikipedia.org/wiki/Albatros_\(tram\)](https://nl.wikipedia.org/wiki/Albatros_(tram))

https://nl.wikipedia.org/wiki/Presidents%27_Conference_Committee-tram

Uitbreiding:

Use Case 1.1

Stappen:

[2, tijdens] Hou rekening met extra attribuut 'type'

Uitzonderingen:

Geen

Voorbeeld:

Voorbeeld van verschillende types. Opgepast: om het kort te houden is dit voorbeeld niet volledig: de stations van lijn 9 ontbreken.

```
<STATION>
  <naam>A</naam>
  <type>Metrostation</type>
  <volgende>B</volgende>
  <vorige>C</vorige>
  <spoor>15</spoor>
</STATION>
<STATION>
  <naam>B</naam>
  <type>Halte</type>
  <volgende>C</volgende>
  <vorige>A</vorige>
  <spoor>15</spoor>
</STATION>
<STATION>
  <naam>C</naam>
  <type>Metrostation</type>
  <volgende>A</volgende>
  <vorige>B</vorige>
  <spoor>15</spoor>
</STATION>
<TRAM>
  <lijnNr>9</lijnNr>
  <type>PCC</type>
  <zitplaatsen>16</zitplaatsen>
  <snelheid>40</snelheid>
  <beginStation>X</beginStation>
</TRAM>
<TRAM>
  <lijnNr>15</lijnNr>
  <type>Albatros</type>
  <zitplaatsen>72</zitplaatsen>
  <snelheid>70</snelheid>
  <beginStation>A</beginStation>
</TRAM>
```

1.4. Trams met voertuignummer inlezen

Prioriteit:

BELANGRIJK

Doel:

Om tot een meer realistische simulatie te komen moet het mogelijk zijn dat meerdere trams op een zelfde spoor voorkomen. Daarvoor moet elke tram voorzien zijn van een voertuignummer.

Uitbreiding:

Use Case 1.1

Stappen:

[2, tijdens] Hou rekening met extra attribuut voertuignummer

Uitzonderingen:

Geen

Voorbeeld:

Twee trams op lijn 12, gegeven de input van 1.1

```
<TRAM>
  <lijnNr>12</lijnNr>
  <voertuigNr>1</voertuigNr>
  <zitplaatsen>32</zitplaatsen>
  <snelheid>60</snelheid>
  <beginStation>A</beginStation>
</TRAM>
<TRAM>
  <lijnNr>12</lijnNr>
  <voertuigNr>2</voertuigNr>
  <zitplaatsen>16</zitplaatsen>
  <snelheid>80</snelheid>
  <beginStation>B</beginStation>
</TRAM>
```

1.5 Stations met meerdere sporen inlezen

Prioriteit:

NUTTIG

Doel:

Om tot een meer realistische simulatie te komen moet het mogelijk zijn dat een station meerdere sporen kan bevatten. Opgepast: enkel een metrostation kan meerdere sporen bevatten; een halte heeft maximaal één spoor (zie Use Case 1.3).

Uitbreiding:

Use Case 1.1

Stappen:

[2, tijdens] Hou rekening met andere structuur

Uitzonderingen:

Geen

Voorbeeld:

Drie stations met elk twee sporen, waarbij spoor 21 dezelfde route doet als spoor 12 in de omgekeerde richting. Merk op dat met deze uitbreiding van use case 1.1 het invoerformaat voor stations wijzigt.

```
<STATION>
  <naam>A</naam>
  <type>Metrostation</type>
  <SPOOR>
    <spoor>12</spoor>
    <volgende>B</volgende>
    <vorige>C</vorige>
  </SPOOR>
  <SPOOR>
    <spoor>21</spoor>
    <volgende>C</volgende>
    <vorige>B</vorige>
  </SPOOR>
</STATION>
<STATION>
  <naam>B</naam>
  <type>Metrostation</type>
  <SPOOR>
    <spoor>12</spoor>
    <volgende>C</volgende>
    <vorige>A</vorige>
  </SPOOR>
  <SPOOR>
    <spoor>21</spoor>
    <volgende>A</volgende>
    <vorige>C</vorige>
  </SPOOR>
</STATION>
<STATION>
  <naam>C</naam>
  <type>Metrostation</type>
  <SPOOR>
    <spoor>12</spoor>
    <volgende>A</volgende>
    <vorige>B</vorige>
  </SPOOR>
  <SPOOR>
    <spoor>21</spoor>
    <volgende>B</volgende>
    <vorige>A</vorige>
  </SPOOR>
</STATION>
```

2.2. Simpele uitvoer (herzien)

Prioriteit:

VERPLICHT

Doel:

Om een beter overzicht te krijgen van het metronet wil de klant een overzichtelijker soort uitvoer van alle informatie. Zo moet bvb de volgorde van stations op n spoorlijn duidelijker naar voor komen.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Het systeem heeft een tekstbestand (ASCII) uitgevoerd, waarin de informatie over het virtuele metronet netjes is uitgeschreven.

Uitbreiding:

Dit is een aanpassing van use case 2.1

Stappen:

1. Open uitvoerbestand
2. WHILE Nog stations beschikbaar
 - 2.1. Schrijf stationgegevens uit
3. WHILE nog trams op spoorlijn
 - 3.1. Schrijf tramgegevens uit
4. Sluit uitvoerbestand

Uitzonderingen:

Geen

Voorbeeld:

Het al dan niet afdrukken van sommige eigenschappen van trams en/of passagiers (bvb voertuignummers van trams) is afhankelijk van welke use cases jullie wensen te implementeren.

---= STATIONS =---

-> Station A

wachtende passagiers:

-> Station B

wachtende passagiers:

-> Station C

wachtende passagiers:

* Groep2, 7 mensen, reist naar station A

---= TRAMS =---

Tram 12 nr 1

zitplaatsen: 32

snelheid: 60

huidig station: A

reizende passagiers:

* Groep1, 2 mensen, reist naar station C

Tram 12 nr 2

zitplaatsen: 16

snelheid: 80

huidig station: B

reizende passagiers:

2.3. Grafische impressie

Prioriteit:

BELANGRIJK

Doel:

De toestand van het metronet wordt grafisch weergegeven.

Preconditie:

Het systeem is correct geïnitialiseerd.

Succesvol einde:

Het systeem heeft een tekstbestand (ASCII) uitgevoerd, waarin de toestand van het metronet staat beschreven.

Stappen:

1. Open uitvoerbestand
2. Teken gegevens uit voor de toestand van het metronet
3. Sluit uitvoerbestand

Uitzonderingen:

Geen

Voorbeeld:

Indien drie STATIONS (A, B, C) en twee trams (T):

```
=A===B===C=  
T    T
```


2.4 Integratie met Graphics

Prioriteit:

BELANGRIJK

Doel:

Om het stadsbestuur van de stad Antwerpen te overtuigen had onze klant graag een 3D visualisatie gehad van het rondrijden van de trams. Hiervoor kan je een standaard interface voor je graphics engine gebruiken.

Preconditie:

Het systeem is correct geïnitieerd.

Succesvol einde:

Elke beweging van trams wordt weergegeven in een 3D omgeving.

3.1. Rijden van trams

Prioriteit:

VERPLICHT

Doel:

Simuleren van rondrijdende trams op het metronet.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Een tram bevindt zich op een nieuwe locatie in het metronet. Het systeem heeft een boodschap afgedrukt met de details van de verplaatsing.

Stappen:

1. Voer verplaatsing uit voor tram op gegeven spoor in gegeven station
2. Schrijf overzicht uit

Uitzonderingen:

Geen

Voorbeeld:

Gegeven de input van 1.1

Tram 12 reed van station A naar station B.

3.3. Op- en afstappen van passagiers(herzien)

Prioriteit:

VERPLICHT

Doel:

Om een meer nauwkeurige simulatie te verkrijgen willen we passagiers laten op- en afstappen volgens hun opgegeven begin- en eindstation. Opgepast: de albatros trams stoppen niet aan een bovengrondse haltes; passagiers van dit soort tram kunnen dus niet afstappen aan stations van het type halte (zie Use Case 1.3).

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Passagiers in hun beginstation zijn op een tram gestapt die in het station passeert indien er nog plaats is. Passagiers op een tram die arriveren in hun eindstation zijn afgestapt.

Uitbreiding:

Dit is een aanpassing van use case 3.2

Stappen:

1. IF Tram in Station
 - 1.1. FOR Passagiers op Tram
 - 1.1.1. IF Einstation = Station
 - 1.1.1.1. Passagier stapt af
 - 1.2. FOR Passagiers in Station
 - 1.2.1. IF Beginstation = Station
 - 1.2.1.1. IF plaats
 - 1.2.1.1.1. Passagier stapt op

Uitzonderingen:

- 1.2.1.1. [er is geen plaats meer] Print een foutboodschap met vermelding van de naam van de groep van de passagier. Ga verder met stap 1.2. Passagiers die niet meer kunnen opstappen wachten op de volgende tram.

3.4. Automatische simulatie

Prioriteit:

VERPLICHT

Doel:

Simulatie automatisch laten lopen tot alle passagiers op eindbestemming zijn.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Alle passagiers zijn op hun bestemming geraakt en de simulatie stopt.

Stappen:

1. WHILE passagiers nog niet op eindbestemming
 - 1.1 voer use case 3.1 uit
 - 1.2 voer use case 3.3 uit

3.5. Rijden van meerdere trams

Prioriteit:

BELANGRIJK

Doel:

Om tot een meer realistische simulatie te komen moeten meerdere trams tegelijkertijd kunnen rondrijden op hetzelfde spoor. Hiervoor moet je ook use case 1.4 implementeren.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Use case 1.4 is geïmplementeerd.

Succesvol einde:

Alle trams bevinden zich op een nieuwe locatie in het metronet, het systeem heeft een boodschap afgedrukt met de details over de verplaatsingen.

Uitbreiding:

Use case 3.1

Stappen:

1. FOR EACH tram

1.1 voer use case 3.1 uit

3.6. Botspreventie

Prioriteit:

BELANGRIJK

Doel:

Trams op eenzelfde spoor kunnen elkaar niet inhalen, maar moeten wachten tot de tram voor hen in een bepaald station vertrekt.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Een tram die naar station X moet, moet wachten in station X-1 als er nog een tram in station X staat.

Uitbreiding:

Use case 3.5

Stappen:

[1.1, voor]

1.1 IF volgend station bezet

1.1.1 wacht in vorig station

3.7. Routeplanner passagiers

Prioriteit:

NUTTIG

Doel:

Om tot een meer realistische simulatie te komen moet het mogelijk zijn dat passagiers de snelste weg kiezen naar hun eindbestemming. Bovendien moeten passagiers ook kunnen overstappen in een station met meerdere sporen. Hiervoor moet je ook use case 1.5 implementeren. Opgepast: de albatros trams stoppen niet aan een bovengrondse halte; passagiers met een bestemming van type halte kunnen beter niet de albatros trams nemen (zie Use Case 1.3).

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

De passagiers nemen altijd de snelste route naar hun bestemming. Passagiers kunnen tussen overstappen op een andere tram in een station met meerdere sporen.

Uitbreiding:

Use case 3.3

Voorbeeld:

Gegeven de stations in het voorbeeld van use case 1.5, kunnen passagiers zowel tram 12 als tram 21 nemen om tussen de stations te reizen. Passagiers zullen altijd voorkeur geven aan de snelste reisweg. Indien use case 3.9 geïmplementeerd is kan dit berekend worden met de snelheid van de trams, anders zal het aantal tussenliggende stations gebruikt worden als eenheid.

3.8. Omzet per tram

Prioriteit:

NUTTIG

Doel:

Om de mogelijke opbrengst te simuleren houdt elke tram zijn omzet bij (alle passagiers betalen op de tram).

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

De tram kan op elk moment zijn omzet van de volledige simulatie tot nog toe opvragen.

Uitbreiding:

Use case 3.3

Stappen:

[1.2.1.1.1., na]

1.2.1.1.2. Omzet tram gaat omhoog met 2 euro per passagier

Voorbeeld:

Elke tram begint de simulatie met een omzet van 0 euro. Als de passagiers uit de file hieronder opstappen, wordt de omzet van de tram verhoogt naar 4 euro (2 euro per passagier). Deze omzet moet ook worden weergegeven voor elke tram in de simpele uitvoer (use case 2.2). Ook moet de totale omzet voor alle trams kunnen worden weergegeven.

```
<PASSAGIER>
  <naam>Groep1</naam>
  <beginStation>A</beginStation>
  <eindStation>C</eindStation>
  <hoeveelheid>2</hoeveelheid>
</PASSAGIER>
```


3.9. Rijden met reële snelheid

Prioriteit:

NUTTIG

Doel:

Meer realistische simulatie van trams, zodat bij de grafische simulatie de trams tegen verschillende snelheden kunnen rondrijden.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

De tram kan zich tegen de gegeven snelheid van een station naar een ander station begeven.

Uitbreiding:

Use case 2.4

3.10 Statistische verwerking simulatie

Prioriteit:

NUTTIG

Doel:

Gedurende de uitvoering worden er relatieve gegevens verzamelt ivm bezettingsgraad en totale omzet. Alsook absolute gegevens over het gebruik van bepaalde tramlijnen.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Rapport afgedrukt met statistische gegevens.

Uitbreiding:

Use case 3.4

Stappen:

1. WHILE simulatie actief
 - 1.1 Verzamel gegevens
2. Bereken waarden
3. Druk rapport af

4.1. GUI voor het rijden van trams

Prioriteit:

NUTTIG

Doel:

Een gebruiksvriendelijke userinterface hebben voor het individueel besturen van trams.

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Trams kunnen manueel bestuurd worden aan de hand van een grafische userinterface.

Stappen:

1. Bepaal tram
2. Laat tram 1 stap verder rijden

4.2 GUI voor beheren van passagiers

Prioriteit:

NUTTIG

Doel:

Een gebruiksvriendelijke userinterface hebben voor het beheren van passagiers: tussen simulatiestappen kan je passagiers toevoegen aan stations, passagiers verwijderen uit stations, eigenschappen van wachtende passagiers aanpassen (bvb bestemming, hoeveelheid, ...).

Preconditie:

Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:

Passagiers kunnen manueel aangepast worden aan de hand van een grafische userinterface.

4.3 GUI voor statistische gegevens

Prioriteit:

NUTTIG

Doel:

Visualizatie van de statistische gegevens verzamelt in use case 3.10.

Preconditie:

Het systeem heeft statistische gegevens verzamelt.

Succesvol einde:

De statistische gegevens worden weergegeven in een grafiek door middel van een grafische userinterface.

Uitbreiding:

Use case 3.10

A Invoer formaat

Het invoerformaat voor het virtueel metronet is zodanig gekozen dat nieuwe attributen en elementen makkelijk kunnen worden toegevoegd.

```
MetroNet = { Element }
Element = "<" ElementType ">" AttribuutLijst "</" ElementType ">"
ElementType = "STATION" | "TRAM" | "PASSAGIER"
AttribuutLijst = Attribuut { Attribuut }
Attribuut = "<" AttribuutType ">" AttribuutWaarde "</" AttribuutType ">"
AttribuutType = "naam" | "vorige" | "volgende" | "spoor" | "lijnNr"
               | "zitplaatsen" | "snelheid" | "beginStation"
               | "hoeveelheid" | "voertuigNr" | "type"
AttribuutWaarde = Primitief
Primitief = Integer | String
Integer = Digit { Digit }
Digit = "0" ... "9"
String = Letter { Letter }
Letter = "a" ... "z" | "A" ... "Z"
```

Merk op dat de AttribuutLijst een relatief vrij formaat heeft wat sterk zal afhangen van het soort element dat gedefinieerd wordt. De volgende tabel toont de verplichte en optionele attributen voor elk element:

ElementType	Attribuut (verplicht)	Attribuut (optioneel)
STATION	naam, vorige, volgende, spoor, type	-/-
TRAM	lijnNr, zitplaatsen, beginStation, snelheid, type	voertuigNr
PASSAGIER	naam, beginStation, eindStation, hoeveelheid	-/-

Bovendien zal afhankelijk van het AttribuutType slechts een bepaalde AttribuutWaarde toegelaten zijn:

AttribuutType	AttribuutWaarde
naam, vorige, volgende, beginStation, type	String
spoor, lijnNr, zitplaatsen, snelheid, voertuigNr, hoeveelheid	Integer

Bovendien moet de openings tag steeds overeenkomen met de sluitingstag. Vandaar dat tijdens de invoer moet gecontroleerd worden of de invoer al dan niet geldig is.

Het bestand met het in te lezen metro net wordt met de hand geschreven. Om het ingelezen metronet te kunnen simuleren moet de informatie consistent zijn.

Een metronet is consistent als:

- elk station is verbonden met een voorgaand en een volgend station voor elk spoor
- er bestaan geen trams met een lijn nummer dat niet overeenkomt met een spoor in een station
- er geen sporen zijn waarvoor geen tram bestaat
- het beginstation van een tram een geldig station in het metronet is
- elk spoor maximaal één keer door elk station komt
- een passagier in een ander station afstapt dan het station waar hij/zij opstapt
- elke passagier zijn bestemming kan bereiken