

Auteur(s): Team 7

Raymond van der Rots
Fabian van den IJssel
Maarten Bonsema
Martijn de Jong
Erik Hoogland
Yorick Groot

Datum: 20 november 2007

TIGAM is onderdeel van de
Hogeschool van Amsterdam

Inhoudsopgave

Glossary.....	2
Genesis.....	4
Doel.....	4
Testplan.....	4
Business case.....	5
Actor lijst.....	5
Use cases.....	5
Use case details.....	6
System Requirements.....	7
Systeem actoren.....	7
System use cases.....	7
Use case details.....	8
User Interface Sketch.....	10
Requirements.....	11
Static Analysis.....	12
Communication diagram.....	14



1. Glossary

Hier kunt u de betekenissen en definitie van veel gebruikte termen vinden.

Wachttijd

De verstreken tijd tussen het binnenkomen van het systeem en het instappen in een trein.

Beheerder

Persoon die het systeem in de gaten houdt.

Controle paneel

De GUI van de simulatie waarin de controle over het systeem uitgeoefend word.

GUIMain (klasse)

Het hoofdvenster van de gebruikersinterface. Hier wordt ook de benodigde informatie van de gui in opgeslagen.

SimulatiePaneel (klasse)

Dit paneel beslaat de schets van de baan en is verantwoordelijk voor het updaten hiervan.

StationPaneel (klasse)

Dit paneel is een van de tabbladen onderin de gebruikersinterface en geeft een lijst weer met de stations en hun reizigers.

TreinPaneel (klasse)

Dit paneel is een van de tabbladen onderin de gebruikersinterface en geeft een lijst weer met de treinen en hun attributen.

ReizigerPaneel (klasse)

Dit paneel is een van de tabbladen onderin de gebruikersinterface en geeft een lijst weer met de alle reizigers in het systeem. Er is ook de optie om de informatie in het logbestand te betrekken.

OverzichtPaneel (klasse)

Dit is het 'hoofdtabblad' en geeft algemene informatie over het systeem (zoals gemiddelde wachttijden)

Main (Klasse)

Algemene klasse waarin de systeemtaken worden verdeeld onder de verschillende treinen. Het houdt ook alle treinen en de reizigerslijst van het systeem bij.

LinkedList (klasse)

Dit is een hulpklasse om alle reizigers op te slaan in een aaneengesloten lijst.

Trein(Klasse)

Klasse waarin de eigenschappen van de trein worden bijgehouden. Elke trein rijdt zelfstandig.



Reiziger(Klasse)

Bevat de informatie en status van een reiziger in de klasse Reiziger.

Wissel(Klasse)

Klasse waarin de status van de wissel staat en waarmee de wissel omgezet kan worden.

BaanBehavior (Interface)

Deze interface geeft het gedrag van de baan aan zodat de simulatie en de baan altijd dezelfde opdrachten kunnen uitvoeren. Dit is om het systeem robuuster te maken. Er worden verschillende methodes meegegeven aan de Baan en de Simulatie om de treinen en wissels aan te kunnen sporen.

Baan (klasse)

De interface naar de baan toe. Erft al het gedrag van BaanBehavior.

Simulatie (klasse)

Zorgt voor een simulatie als de baan niet gekoppeld is. Imiteert al het gedrag van de baan.

Main(Package)

Bevat alle klassen die betrekking hebben op 'taakverdeling' van het systeem.

GUI (Package)

Dit package bevat de opmaak van de gebruikersinterface.

Data (Package)

Dit package beslaat de onderste laag van het systeem en hier worden alle eindklassen in bijgehouden (reiziger, trein en wissel).

Baan(Package)

In dit package zitten alle klassen die de communicatie voorzien met de baan of met de simulatie van de baan.

2. Genisis

2.1 Doel

Het doel van het project is het maken van een systeem om treinen automatisch in een taxi systeem te laten rijden. Als een klant een trein aanvraagt zal hij zo snel mogelijk naar zijn eindbestemming gereden worden. Er wordt dus geen gebruik meer gemaakt van een dienstregeling.

De baan telt acht stations en een nader te bepalen aantal treinen. Alle treinen tellen maximaal vijf personen en zullen in één richting rijden. Ook kunnen de treinen gekoppeld worden. Er wordt aangenomen dat de begin- en eindbestemming bekend zijn alsmede de vertrektijd. Er zal vanuit worden gegaan dat de reiziger zich ook op dat tijdstip op het station bevindt.

2.2 Testplan

De gemaakte simulaties en hardware aansturing worden getest via het white box testing principe. Dit betekent dat ieder onderdeel van de hard- en software onafhankelijk op zijn gedrag getest wordt.

Er worden zoveel mogelijk scenario's gesimuleerd om een goed beeld van de stabiliteit van het onderdeel te verkrijgen. Als het onderdeel niet aan de wensen voldoet zal het worden aangepast en opnieuw getest. Deze cyclus zal net zo lang herhaald worden totdat het onderdeel aan de eisen die daaraan gesteld worden voldoet.

Het testen zal gebeuren door het team en de medewerkers van TIGAM.

Het testen zal zowel tijdens als na het ontwikkelen van het systeem gebeuren. Als het systeem voltooid is zal er ook een black box test gedaan worden waarbij het systeem als geheel getest zal worden.

3. Business case

In dit hoofdstuk wordt het project vanuit een bedrijfsstandpunt geanalyseerd. Er zal eerst een lijst met betrokkenen (actoren) gegeven worden, hierop volgen de acties die zijn kunnen verrichten binnen het systeem (use cases) met tenslotte een uitwerking hiervan.

3.1 Actor lijst

Beheerder

De beheerder is degene die het systeem bediend. Het systeem kan slechts door een persoon tegelijk bediend worden en deze heeft hier volledige toegang toe.

3.2 Use cases

B 1: Voeg reiziger toe Beheerder

Met deze use case wordt er een reiziger toegevoegd aan het systeem. Bij het toevoegen van een reiziger moet een begin- en eindpunt opgegeven worden. Ook moet er opgegeven worden op welke tijd de reiziger vertrekt. Zodra de reiziger op zijn bestemming is aangekomen word deze verwijderd uit het systeem.

B 2: Verwijder reiziger Beheerder

De beheerder kan een reiziger verwijderen uit het systeem.

Use case details

B 1	Voeg reiziger toe
Actor	Beheerder
Beschrijving	Met deze use case wordt er een reiziger toegevoegd aan het systeem. Bij het toevoegen van een reiziger moet een begin- en eindpunt opgegeven worden. Ook moet er opgegeven worden op welke tijd de reiziger vertrekt. Zodra de reiziger op zijn bestemming is aangekomen word deze verwijderd uit het systeem.
Stappen	<ol style="list-style-type: none"> 1. De beheerder geeft het begin-, eindbestemming en vertrektijd op. De reiziger wordt toegevoegd aan het systeem. 2. De trein komt zo snel mogelijk voorrijden, de reiziger neemt plaats 3. De reiziger stapt uit op zijn plaats van bestemming. 4. De reiziger wordt verwijderd uit het systeem.

B 2	Verwijder reiziger
Actor	Beheerder
Beschrijving	Het verwijderen van een reiziger.
Preconditie	De reiziger is bekend in het systeem en staat nog bij het beginstation.
Stappen	<ol style="list-style-type: none"> 1. De beheerder geeft aan een specifieke reiziger te willen verwijderen. 2. De evt. toegewezen trein wordt vrijgemaakt 3. De reiziger wordt uit het systeem verwijderd.
Postconditie	De reiziger bevindt zich niet meer in het systeem.
Abnormale paden	<ol style="list-style-type: none"> 1. De reiziger bestaat niet: Er wordt een foutmelding gegeven en doorgegaan met uitvoering van het systeem.

4. System Requirements

In dit hoofdstuk wordt het systeem technisch verder uitgediept en worden alle actoren en hun use cases vanuit een ontwikkeloogpunt gegeven. Ook wordt er een schets gegeven van de gebruikersinterface van het systeem en de eisen waar het systeem aan moet voldoen.

4.1 Systeem actoren

Beheerder Operator die het systeem controleert en zo nodig kan wijzigen.

4.2 System use cases

U 1: Toevoegen reiziger

Beheerder

Met deze use case wordt er een reiziger toegevoegd aan het systeem. Hij voert het begin- en eindpunt in alsmede de vertrektijd. Het systeem zal om die tijd een trein ter beschikking stellen om de reiziger naar de eindbestemming te brengen.

U 2: Verwijder reiziger

Beheerder

De beheerder kan een reiziger verwijderen uit het systeem.

U 3: Toevoegen trein

Beheerder

De beheerder kan een extra trein inzetten.

U 4: Verwijder trein

Beheerder

De beheerder kan een trein uit het systeem verwijderen.

U 5: Simulatiesnelheid aanpassen

Beheerder

De beheerder kan de simulatiesnelheid van het systeem aanpassen mits de baan niet gekoppeld is.

4.3 Use case details

U 1	Toevoegen reiziger
Actor	Beheerder
Beschrijving	Het handmatig toevoegen van een reiziger.
Preconditie	Geen
Stappen	<ol style="list-style-type: none"> 1. De beheerder geeft aan een reiziger toe te willen voegen 2. De begin-, eindbestemming en vertrektijd worden opgegeven. 3. Er zal een trein door het systeem beschikbaar worden gesteld en klaargezet worden op het station. 4. De trein rijdt direct naar de eindbestemming 5. De reiziger word uit het systeem verwijderd en de trein is weer beschikbaar.
Postconditie	De trein is van het begin- naar het eindpunt gereden.
Abnormaal pad	De reiziger is al bekend in het systeem en kan niet worden toegevoegd. Er zal geen actie ondernomen worden.

U 2	Verwijder reiziger
Actor	Beheerder
Beschrijving	Het verwijderen van een reiziger.
Preconditie	De reiziger is bekend in het systeem en staat nog bij het beginstation.
Stappen	<ol style="list-style-type: none"> 1. De beheerder geeft aan een specifieke reiziger te willen verwijderen. 2. De evt. toegewezen trein wordt vrijgemaakt 3. De reiziger wordt uit het systeem verwijderd.
Postconditie	De reiziger bevindt zich niet meer in het systeem.
Abnormale paden	<ol style="list-style-type: none"> 1. De reiziger bestaat niet: Er wordt een foutmelding gegeven en doorgedaan met uitvoering van het systeem.



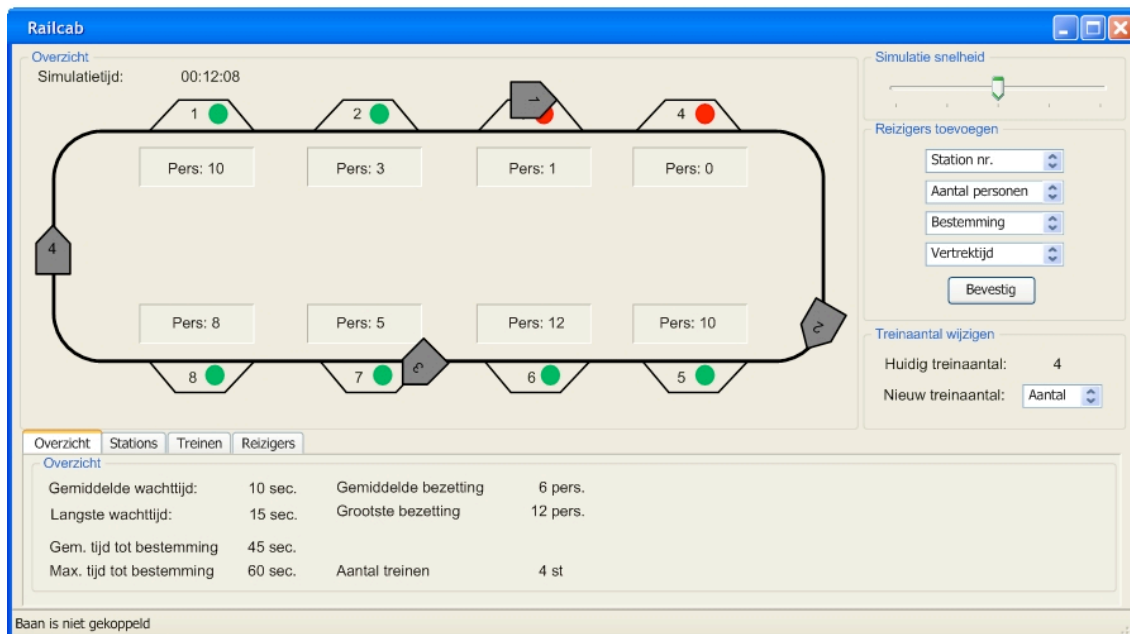
U 3	Toevoegen trein
Actor	Beheerder
Beschrijving	Een extra trein in het systeem brengen
Preconditie	Als de baan gekoppeld is: De trein bevind zich op de juiste plek op de baan.
Stappen	<ol style="list-style-type: none"> 1. De beheerder geeft de trein op die toegevoegd moet worden. 2. De trein wordt opgenomen in het systeem en beschikbaar gesteld.
Postconditie	De trein is aan het systeem toegevoegd

U 4	Verwijder trein
Actor	Beheerder
Beschrijving	Het verwijderen van een trein uit het systeem
Preconditie	Er is minimaal 1 trein in het systeem. De te verwijderen trein bevat geen reizigers
Stappen	<ol style="list-style-type: none"> 1. De beheerder geeft de te verwijderen trein op 2. De trein wordt uit het systeem gehaald
Postconditie	De trein bestaat niet meer in het systeem

U 5	Simulatiesnelheid aanpassen
Actor	Beheerder
Beschrijving	Het wijzigen van de snelheid waarop het systeem zijn acties uitvoert. (b.v. de snelheid van de treinen)
Preconditie	De baan is niet gekoppeld
Stappen	De beheerder voert een nieuwe simulatiesnelheid in.
Postconditie	De snelheid waarop het systeem zijn acties uitvoert is gewijzigd.

4.4 User Interface Sketch

In deze paragraaf word de voorlopige interface van het programma besproken. Wij hebben gekozen voor een gezamenlijke interface van zowel de simulatie zonder als de simulatie met baan.



Het gehele programma wordt in een enkel scherm bevat. Dit bevordert het overzicht en voorkomt het moeten schakelen tussen verschillende vensters.

Het grootste gedeelte van het scherm wordt gedomineerd door een overzichtstekening van de baan. Dit maakt het mogelijk in een oogopslag de status het geheel te bekijken. Hier wordt links bovenin de looptijd van de simulatie weergegeven.

Rechts van dit overzicht bevind zich een schuifknop om de snelheid van de simulatie aan te passen. Deze kan ook op 0 gezet worden zodat het systeem gepauzeerd wordt en de huidige status uitvoerig bekeken kan worden. Onder deze schuifbalk bevinden zich knoppen voor het toevoegen van reizigers en het wijzigen van het aantal treinstellen.

Onderaan bevinden zich verschillende tabbladen waarmee de systeemstatus bekeken kan worden. In het tabblad algemeen worden de belangrijkste gegevens concreet samengevat. Er wordt informatie verschaft over het aantal aanwezige personen in het systeem en de verschillende reistijden.

In het tabblad stations kunnen alle gegevens van de stations opgezocht worden. Hier worden gegevens als het aantal aanwezige reizigers, de status van de wissel en het aantal verwerkte reizigers opgevraagd worden.

Het tabblad treinen geeft de informatie weer van de verschillende treinen. Dit houdt de locatie van de treinen, de route en het aantal passagiers in.



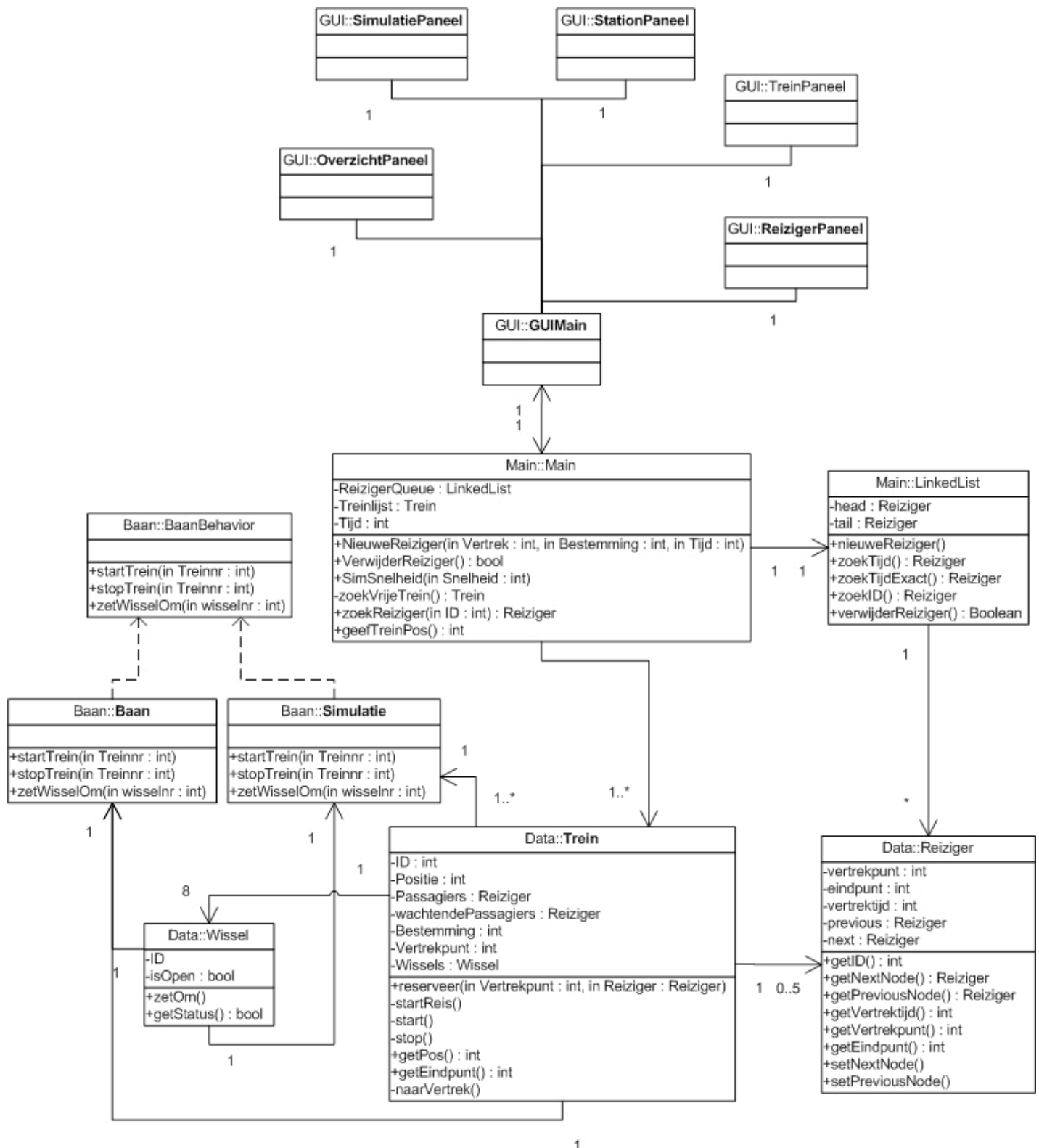
Het laatste tabblad is het reizigers tabblad. Hier staat een enkele tabel in die alle reizigers in het systeem weergeeft. In deze tabel staat het nummer van de reiziger, de vertrek- en aankomsttijden, zijn beginpunt en tenslotte ook het eindpunt vermeld.

4.5 Requirements

1. De gemiddelde wachttijd mag niet meer dan 3 minuten bedragen.
2. De wachttijd mag in geen enkel geval meer dan 7 minuten bedragen
3. De simulatie moet afzonderlijk van de baan kunnen werken.
4. De wissels moeten automatisch bediend worden.
5. Als meerdere treintjes op dezelfde tijd over hetzelfde baangedeelte rijden dienen deze gekoppeld te worden.
6. De output van het systeem moet de volgende gegevens tekstueel weergeven:
 - Gemiddelde wachttijd
 - Maximale wachttijd
 - Aantal reizigers
 - Aantal reizigers op station
7. De snelheid in de simulator moet variabel zijn
8. Het maximale aantal reizigers dat het systeem moet kunnen verwerken is 100
9. Reizigers moeten kunnen worden ingepland met een vertrektijd
10. Treintjes die geen doel hebben wachten op een station

5. Static Analysis

Dit hoofdstuk behandelt het klassendiagram van het RailCab project. Het project gaat uit vier verschillende packages bestaan; GUI, main, data en baan.



De GUI package is verantwoordelijk voor de grafische interface naar de gebruiker toe. Het bestaat uit 5 verschillende panelen gekoppeld aan een hoofdpaneel.

- Het simulatiepaneel is verantwoordelijk voor het tekenen en bijhouden van de baan en de objecten op de baan.
- Het overzichtpaneel geeft een statistiek overzicht van het systeem.
- Het stationpaneel geeft een overzicht van de stations en hun reizigers.
- Het treinpaneel geeft een overzicht van de verschillende treinen met hun positie, de passagiers en zijn eindbestemming.
- Het reizigerspaneel geeft een overzicht van alle reizigers in het systeem. Hierbij komt het vertrekpunt, eindbestemming, huidige vertrek- en wachttijd.

De core package zorgt voor de allocatie van de verschillende treinen en beheert de verschillende objecten in het systeem.

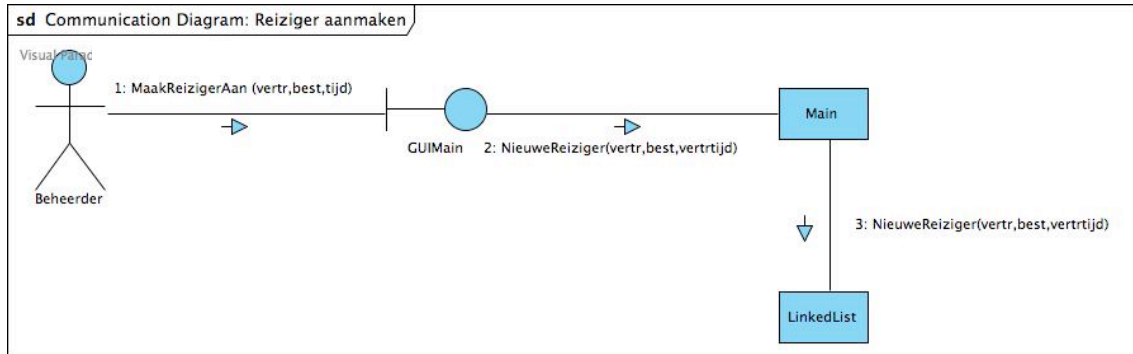
De data package bevat de verschillende objecten in het systeem. Hier worden de wissels, treinen en reizigers onder verstaan.

De treinen zorgen dat de aangewezen reizigers op het aangewezen eindpunt komen. Elke reiziger heeft zo zijn eigen begin- en eindpunt en een vertrektijd.

De baan package zorgt voor een interface naar de baan toe. Als deze niet gekoppeld is neemt de Simulatie klasse zijn taak over door al het gedrag van de baan te simuleren.

5.1 Communication diagram

Hieronder wordt het communicatie diagram weergegeven voor het aanmaken van een reiziger.



Vertr = vertrekstation, best = bestemmingsstation, tijd = vertrektijd. Dit zijn allemaal integers.

De knop voor het aanmaken van een reiziger zit in het hoofdpaneel van de GUI en dit is dan ook waar de methode wordt aangeroepen. Deze wordt dan doorgespeeld aan de Main klasse die deze nieuwe reiziger toevoegt aan zijn ReizigersQueue (van het type LinkedList). Deze is op tijd gesorteerd en bij het toevoegen wordt dit in stand gehouden.

Bij het verwijderen van een reiziger wordt deze opgezocht in de linkedlist en hieruit verwijderd. Als de reiziger niet verwijderd kan worden omdat deze niet gevonden is dan zal er een melding op het scherm gegeven worden.

Bij het toevoegen of verwijderen van treinen worden er treinobjecten aangemaakt en toegevoegd aan de treinlijst in de klasse main.

Voor meer informatie over de interne werking van het systeem verwijs ik u naar het design model.