



Auteur(s):  
Thomas Dik  
Erik Frèrejean  
Ino Gottschal  
Frank Kehl  
Marc Pruntel

Versie:  
Status:  
Datum:

1.0  
Definitief  
24 oktober 2007

TIGAM is onderdeel van de  
Hogeschool van Amsterdam

# Inhoudsopgave

Definitief ontwerp .....	1
Definitief ontwerp .....	1
Definitief ontwerp .....	1
Definitief ontwerp .....	1
Inhoudsopgave.....	2
Inhoudsopgave.....	2
Inhoudsopgave.....	2
Inhoudsopgave.....	2
<a href="#">1. Business Model.....</a>	<a href="#">3</a>
<a href="#">1.1 Probleemstelling.....</a>	<a href="#">3</a>
<a href="#">1.2 Actor list.....</a>	<a href="#">3</a>
<a href="#">1.3 Use case list.....</a>	<a href="#">3</a>
<a href="#">2. System Requirements.....</a>	<a href="#">4</a>
<a href="#">2.1 System actors.....</a>	<a href="#">4</a>
<a href="#">Use cases.....</a>	<a href="#">4</a>
<a href="#">2.2 Supplementary requirements.....</a>	<a href="#">4</a>
<a href="#">2.3 Interface sketches.....</a>	<a href="#">4</a>
<a href="#">3. Static Analysis.....</a>	<a href="#">5</a>
<a href="#">4. Dynamic Analysis.....</a>	<a href="#">6</a>
<a href="#">4.1 Communication Diagram .....</a>	<a href="#">6</a>
<a href="#">4.2 State Diagrams.....</a>	<a href="#">7</a>
<a href="#">4.2.1 State Diagram Reiziger.....</a>	<a href="#">7</a>
<a href="#">4.2.2 State Diagram Trein.....</a>	<a href="#">7</a>
<a href="#">5. Bijlagen.....</a>	<a href="#">8</a>
<a href="#">5.1 Interface sketches.....</a>	<a href="#">8</a>
<a href="#">5.1.1 Reis interface.....</a>	<a href="#">8</a>
<a href="#">5.1.2 Betaling interface.....</a>	<a href="#">9</a>
<a href="#">5.1.3 Reisdetails interface.....</a>	<a href="#">9</a>
<a href="#">5.2 Class Diagrams.....</a>	<a href="#">10</a>
<a href="#">Zie externe bijlage in de mail.....</a>	<a href="#">10</a>
<a href="#">5.3 Usecase diagram.....</a>	<a href="#">11</a>
<a href="#">5.3.1 Usecase diagram.....</a>	<a href="#">11</a>
<a href="#">5.4 Usecase details.....</a>	<a href="#">12</a>

# 1. Business Model

In dit hoofdstuk wordt het business model uitgelegd, beginnend met de probleemstelling. Verder wordt er een eerste indicatie gegeven van de betrokken *Actors* en de gebruikte *use case list*.

## 1.1 Probleemstelling

Het huidige spoornet systeem is inefficiënt en onhandig. Personen moeten naar vaste stations toe waarna ze op vaste tijden vervoerd kunnen worden. Ook kun je op dit moment alleen maar reizen via vaste trajecten. Is het mogelijk om een systeem te ontwerpen waarbij de vertrektijden en trajecten variabel zijn?

## 1.2 Actor list

- Beheerder (controle van het systeem, onderhoud)
- Reiziger (de reiziger)

## 1.3 Use case list

{Reiziger}

- in/uit stappen (*Het in en uit stappen van de reiziger m.b.t. de trein*)
- Bestemming bepalen (*de reiziger kiest een eindbestemming*)
- Details bepalen (*het weergeven van gegevens over de reis*)
- Reis betalen (*het betalen van het "kaartje" voor de reis*)

{Beheerder}

- Overzicht (*een overzicht van de treinen en wissels*)
- Stop/reset uitvoeren (*het systeem stil zetten of reseten*)
- Wijzigen (*het wijzigen van het aantal treinen*)

## 2. System Requirements

In dit hoofdstuk wordt het uiteindelijke systeem beschreven. Dit wordt gedaan aan de hand van: het beschrijven van de gebruikers (2.1), de use cases (2.2 en 2.3), de niet functionele eisen (2.4), de interface sketches (2.5) en het ordenen van de use cases in volgorde van belangrijkheid (2.6).

### 2.1 System actors

- Beheerder.  
Controleert het hele systeem.
- Reiziger.  
De klant die railcab gebruikt om van A naar B te reizen.

### Use cases

**U1: Plan reis:** Een reiziger kiest zijn eindbestemming.

**U2: Bekijk reis details:** De reiziger vraag de details van de geplande reis op.

**U3: Betaal reis:** De reiziger betaalt de reis.

**U4: Voeg trein toe:** De beheerder kan een trein toevoegen aan het systeem.

**U5: Verwijder trein:** De beheerder kan een trein uit het systeem verwijderen.

### 2.2 Supplementary requirements

**S1:** De applicaties draaien op java JDK 6.0

**S2:**

### 2.3 Interface sketches

Om een eerste indruk over de interface van het systeem te kunnen geven zijn er interface schetsen gemaakt.

Voor het railcab project zijn er een tweetal van deze schetsen gemaakt, namelijk voor:

- De interface waar de reiziger een reis bepaald.
- De interface waar de reiziger een betaling doet.

U kunt deze interface sketches vinden in hoofdstuk 5.1.2

## 3. Static Analysis

In dit hoofdstuk wordt het systeem verder uitgelegd met behulp van een klassen diagram. Waarin de klassen, attributen en methodes zijn aangegeven. Het systeem bestaat uit 7 packages: de GUI, algorithm, controller, database, helpers, main en trains.

De GUI (Graphic User Interface) verzorgt de schermen die de klant en de operator te zien krijgen. Deze schermen zorgen dat de gebruikers van het systeem makkelijk en klantvriendelijk met het systeem overweg kunnen. Enkele voorbeelden van deze schermen is het scherm waar de bestemming wordt gekozen, het scherm voor het toevoegen van de treinen en het scherm voor het overzicht van het systeem.

In de package algorithm worden er verschillende dingen bepaald, zoals welke trein er moet gaan rijden om de passagiers op te halen. Of de treinen genoeg afstand houden en er wordt gekeken of de treinen veilig van hun station kunnen vertrekken.

De Controller is de draaiende kern van het systeem. Hier alle aansturingen geregeld, zoals de treinaansturing en het genereren van de coördinaten van de baan.

De package database zorgt voor een connectie met de database en slaat daar gegevens van de treinen en passagiers in op. Tevens worden uit de database gegevens gelezen zodat er een overzicht kan worden gemaakt. Uit dit overzicht kan de beheerder zien of alle treinen goed hebben gereden en hoe efficiënt ze hebben gereden.

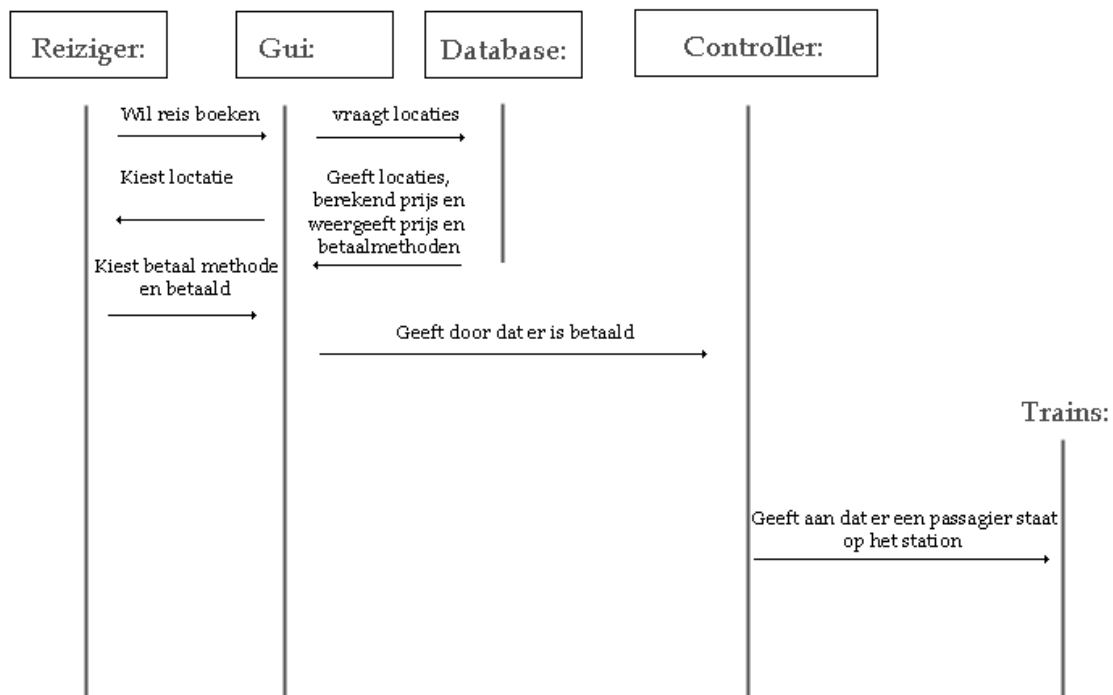
De package helpers is voor het berekenen van de tijd die de treinen er over doen om een opdracht te voltooien. Ook bevindt zich hier een klasse die er voor zorgt dat de informatie makkelijk te bereiken is.

Package main wordt gebruikt voor het starten van de applicatie.

In package trains worden de treinen aangemaakt die de passagiers moeten vervoeren. Hier worden normale en Railcab treinen aangemaakt. Dit is om het verschil te laten zien tussen het huidige en het nieuwe systeem.

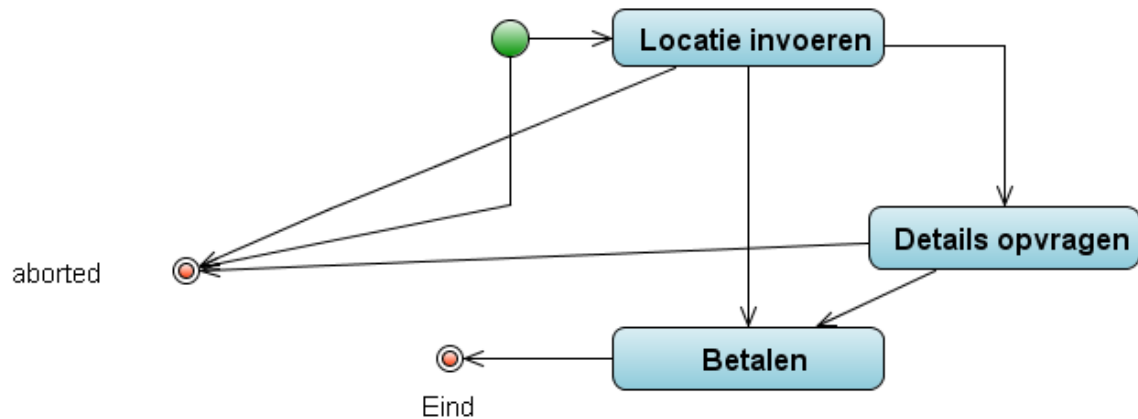
## 4. Dynamic Analysis

### 4.1 Communication Diagram

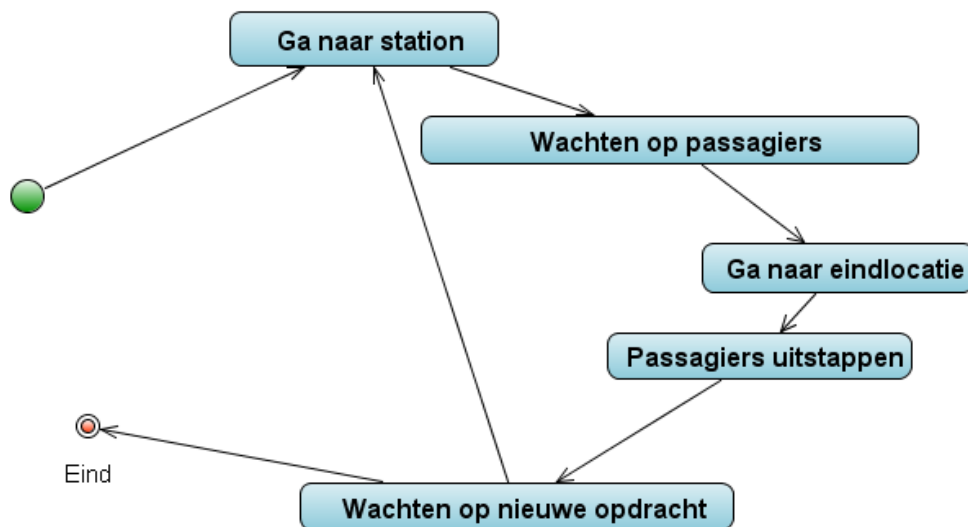


## 4.2 State Diagrams

### 4.2.1 State Diagram Reiziger



### 4.2.2 State Diagram Trein



## 5. Bijlagen

### 5.1 Interface sketches

De interface sketches zijn een eerste impressie over de interface. Deze schetsen zijn puur gericht op de functionele werking van de interface.

#### 5.1.1 Reis interface



Afbeelding

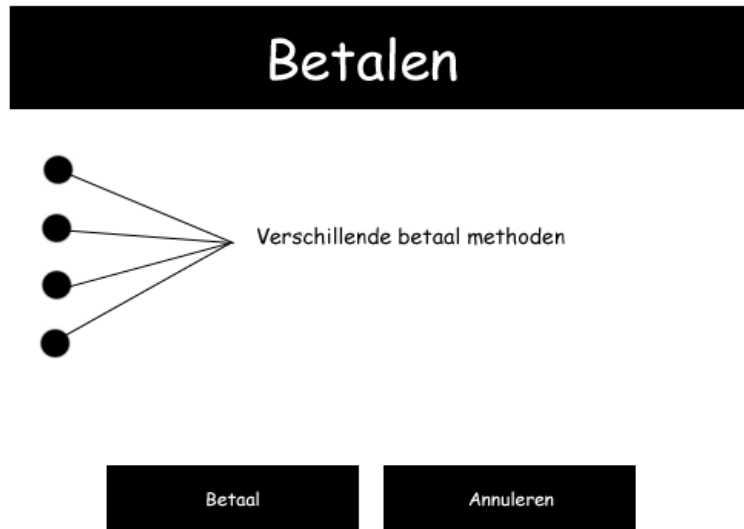
- locatie 1
- locatie 2
- locatie 3
- locatie 4
- locatie 5
- locatie 6
- locatie 7
- locatie 8

Verzend      Annuleren

Hier wordt de gewenste locatie ingevoerd.  
Door op verzend te drukken word het betaal scherm aangeroepen.  
Door op annuleren te drukken kan op elk gewenst moment gestopt worden.



### 5.1.2 Betaling interface



Als er een gekozen is kan er op betaal gedrukt worden om te gaan betalen  
Er kan natuurlijk ook op elk gewenst moment op annuleren  
gedrukt worden om te stoppen

### 5.1.3 Reisdetails interface



The diagram shows the travel details interface. It features a black header with the text "Reis Details" in white. Below the header, there are six labels on the left, each followed by a light gray rectangular input field: "Vertrekpunt:", "Aankomstpunt:", "Vertrektijd:", "Aankomst tijd:", "Totale reistijd:", and "Reiskosten:". At the bottom of the interface, there are two black buttons: "Annuleren" on the left and "Verder" on the right.

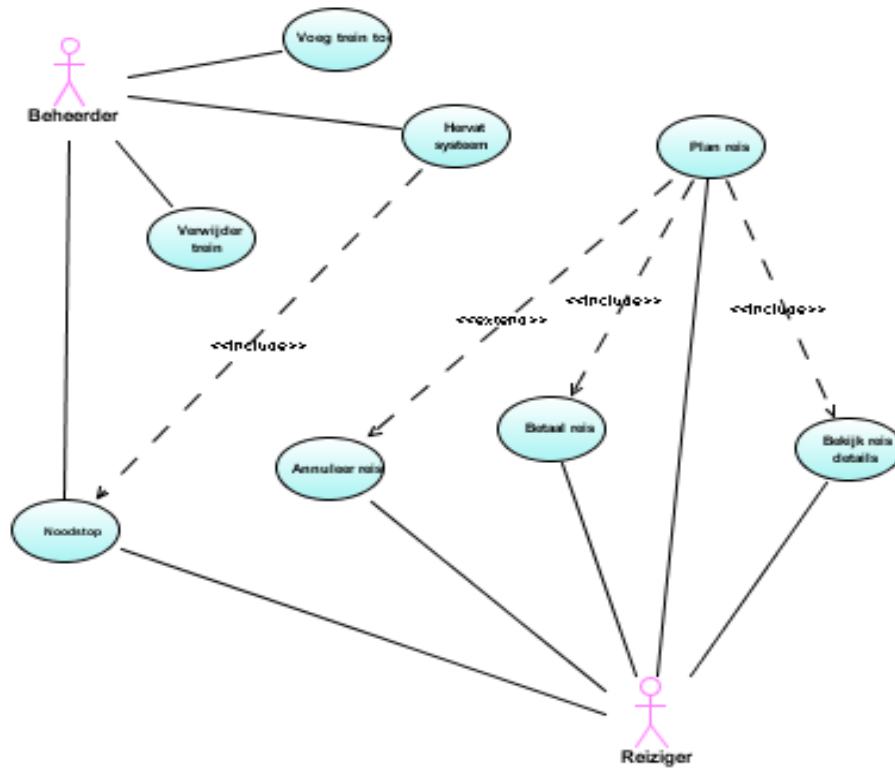
De te weergeven data worden in tekstvelden gezet.  
Door op annuleren te drukken word de complete reis geannuleerd.  
Door op verder te drukken wordt je doorverwezen naar het betaal scherm

## 5.2 Class Diagrams

Zie externe bijlage in de mail.

## 5.3 Usecase diagram

### 5.3.1 Usecase diagram



## 5.4 Usecase details

<p>Usecase nummer: 1 Titel:voeg trein toe</p> <p>Preconditions:</p> <ul style="list-style-type: none"><li>-vrije trein</li></ul> <p>Steps:</p> <ul style="list-style-type: none"><li>-Geef id nummer trein</li><li>-Geef Coördinaten</li></ul> <p>Postconditions:</p> <ul style="list-style-type: none"><li>-vrije plaats in geheugen</li></ul> <p>Abnormal paths:</p> <ul style="list-style-type: none"><li>-Dubbele identiteit</li></ul> <p>Nonfunctional Requirements:</p> <ul style="list-style-type: none"><li>-herbenoem trein identiteit</li></ul>	<p>Usecase nummer: 2 Titel:start systeem</p> <p>Preconditions:</p> <ul style="list-style-type: none"><li>-Systeem niet gestart</li></ul> <p>Steps:</p> <ul style="list-style-type: none"><li>-start treinen</li></ul> <p>Postconditions:</p> <ul style="list-style-type: none"><li>-geen</li></ul> <p>Abnormal paths:</p> <ul style="list-style-type: none"><li>-geen electriciteit</li></ul> <p>Nonfunctional Requirements:</p> <ul style="list-style-type: none"><li>-geen</li></ul>
---	--

Usecase nummer: 3  
Titel:Verwijder trein

Preconditions:

- trein aanwezig

Steps:

- verwijder trein

Postconditions:

- trein is verwijderd

Abnormal paths:

- trein bestaat niet

Nonfunctional Requirements:

- geen

Usecase nummer: 5  
Titel:Annuleer reis

Preconditions:

- locatie is opgegeven
- reis is nog niet betaald

Steps:

- zet alle variabelen op null
- ga naar hoofdscherm

Postconditions:

- applicatie staat weer op hoofdscherm

Abnormal paths:

- reis is al betaald

Nonfunctional Requirements:

- geen

Usecase nummer: 6

Titel:Betaal reis

Preconditions:

- locatie is opgegeven
- klant heeft geld /  
betaalkaart

Steps:

- Kies betaalmethode
- voer pas / code in of  
voer geld in

Postconditions:

- klant heeft kaartje

Abnormal paths:

- klant annuleert

Nonfunctional Requirements:

- geen

Usecase nummer: 7

Titel:bekijk reis details

Preconditions:

- locatie is opgegeven
- klant heeft betaald
- reis is geboekt

Steps:

- weergeef afstand
- weergeef reistijd
- weergeef kosten

Postconditions:

- klant heeft gegevens gezien

Abnormal paths:

- er is geen reis geboekt

Nonfunctional Requirements:

- geen

Usecase nummer: 8

Titel:Plan reis

Preconditions:

- geen

Steps:

- vraag locatie
- vraag betaalmethode
- weergeef reset knop

Postconditions:

- gevraagde gegevens zijn ingevoerd

Abnormal paths:

- klant annuleert

Nonfunctional Requirements:

- geen