

3e Bachelor Informatica  
Academiejaar 2010-2011

Departement Toegepaste Ingenieurswetenschappen  
Schoonmeersstraat 52 - 9000 Gent

# Richtlijnen voor projectverslagen

Visualisatie van xkcd

Groep 6   Jan Acke  
              Davy Loose  
              Simon Gaeremynck  
              Christian Vuerings

# Inhoudsopgave

Inhoudsopgave	1
<b>1 Overzicht</b>	<b>2</b>
<b>2 Model</b>	<b>3</b>
2.1 Statisch model . . . . .	3
2.1.1 Klassenmodel . . . . .	3
2.2 Dynamisch model . . . . .	4
2.2.1 Start simulatie . . . . .	5
2.2.2 Maak computer aan . . . . .	6
2.2.3 Verwijder computer . . . . .	8
2.2.4 Kuis computer . . . . .	9
2.2.5 Stuur bericht . . . . .	10
2.3 Woordenboek . . . . .	11
2.3.1 Beheersysteem . . . . .	11
2.3.2 Computer . . . . .	11
2.3.3 Bestand . . . . .	12
2.3.4 VisualisatieBibliotheek . . . . .	12
2.4 Aanvullingen . . . . .	13
<b>3 Realisatie</b>	<b>14</b>
<b>4 Gebruik</b>	<b>15</b>
4.1 L <sup>A</sup> T <sub>E</sub> X-gebruikers . . . . .	15
4.2 installatie . . . . .	15
4.3 Opmerkingen . . . . .	15
4.4 Niet-L <sup>A</sup> T <sub>E</sub> X-gebruikers . . . . .	16
<b>5 Logboek</b>	<b>17</b>

# Hoofdstuk 1

## Overzicht

Bedoeling van dit project is om studenten een standaardsjabloon aan te bieden bij het maken van projectverslagen. Dit sjabloon kan bijvoorbeeld gebruikt worden bij het vakoverschrijdend project, of bij het labo systeemanalyse.

Er zijn twee mogelijke groepen van gebruikers:

1.  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -gebruikers. Deze kunnen een ziparchief met de nodige bestanden downloaden en installeren. Daarna kunnen ze het `project.tex`-bestand gebruiken als sjabloon.
2. Niet- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -gebruikers. Deze dienen op eigen kracht de opmaak en de indeling van het voorbeeld naar hun tekstverwerker over te brengen. Hiertoe dienen ze het bestand `project.pdf` uit het ziparchief te raadplegen.

# Hoofdstuk 2

## Model

### 2.1 Statisch model

#### 2.1.1 Klassenmodel

We beschrijven voor iedere klasse wat hij precies doet.

**Bestand::** Dit is het object die wordt doorgestuurd tussen de verschillende computers. Het bestand kan opgesplitst worden in 2 sub onderverdelingen.

**GoedBestand::** Dit is een bestand die geef nare gevolgen heeft voor de computer die het ontvangt, men kan dit vergelijken met een email met alleen tekst.

**SlechtBestand::** Dit is een bestand die een virus, trojan of worm bevat. Via de gegevens van het bestand kunnen we zo het risico berekenen dat de computer besmet wordt.

**Computer::** Zoals men kan verwachten is dit een object die bestanden kan ontvangen en versturen. Het is altijd verbonden met tenminste n andere computer. Aan de hand van een lijst van virussen dat de computer bevat kan hij virussen doorsturen naar buurcomputers.

**Beheersysteem::** Dit is het object die het scherm beïnvloed. Deze zorgt voor de demo modus van het systeem. Het beheersysteem zal de keuze maken welke computers er nieuwe bureaus krijgen, welke computers bestanden versturen en gekuist worden van virussen, en welke computers verwijderd zullen worden.

**VisualisatieBibliotheek::** Dit is een bibliotheek dat we gebruiken voor het tekenen van de graaf. Via deze bibliotheek kunnen we gemakkelijk zien welke computers er met elkaar verbonden zijn en kunnen we dit ook tekenen.

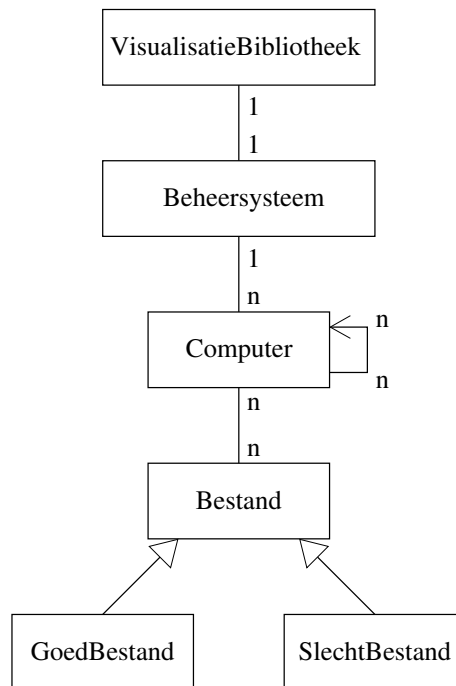


Diagram 2.1: Het klassendiagram

## 2.2 Dynamisch model

We geven hieronder een lijst van alle taken.

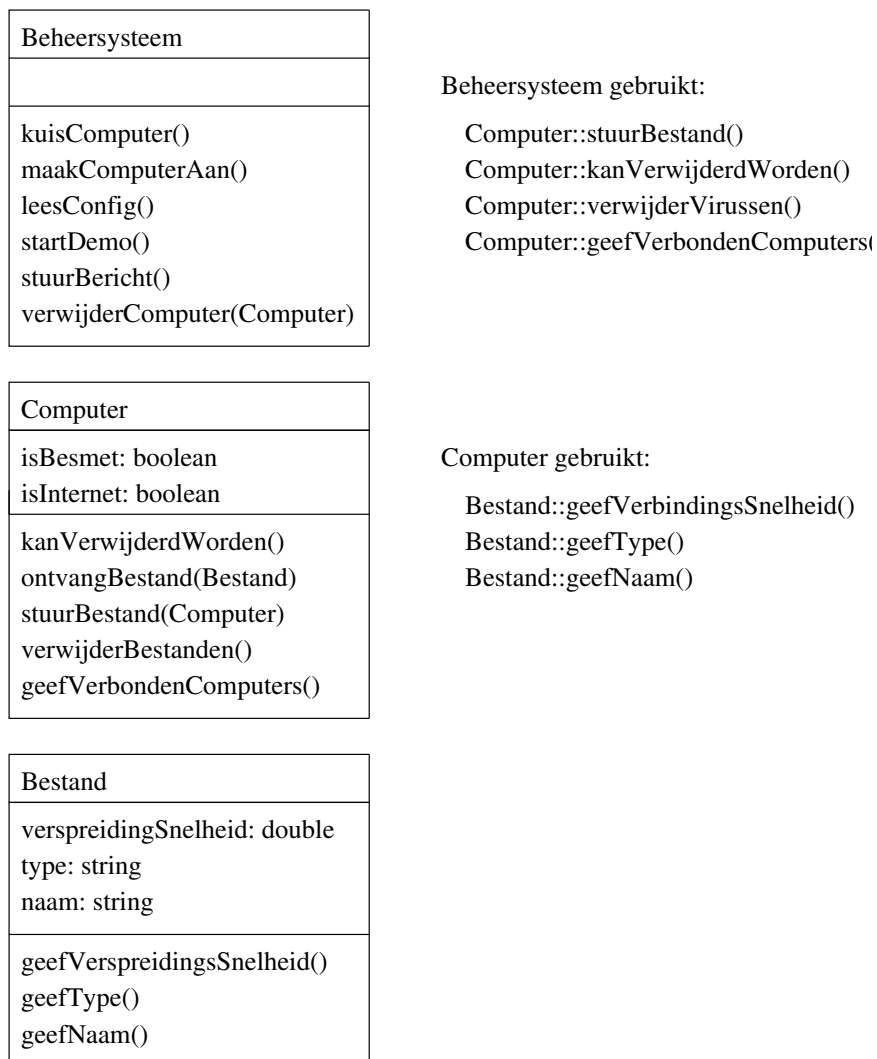


Diagram 2.2: De CRC kaarten

### 2.2.1 Start simulatie

**Actor:** deGebruiker.

**Beschrijving:** Via deze taak kan de gebruiker de demo mode activeren en zal het beheersysteem op basis van de klok de verschillende taken automatisch uitvoeren.

**Sequentiediagram:**

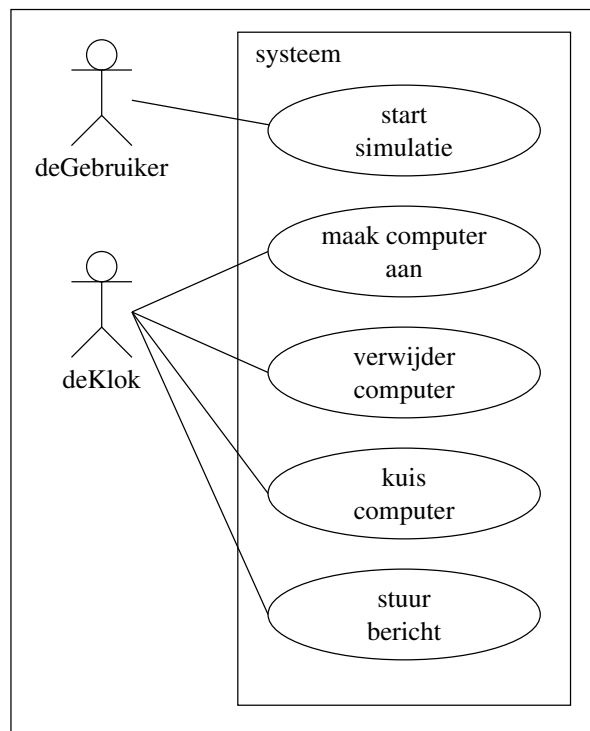


Diagram 2.3: Takendiagram

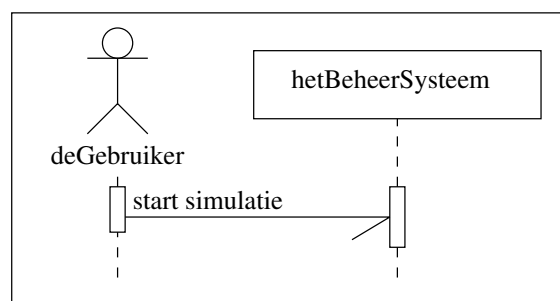


Diagram 2.4: Start simulatie

### 2.2.2 Maak computer aan

**Actor:** deKlok.

**Beschrijving:** De klok stuurt een bericht naar het BeheerSysteem dat er een computer mag aangemaakt worden. Er moet niet gewacht worden op een antwoord van de computer om verder te doen met het systeem. De computer zal

wel een asynchroon antwoord sturen indien hij aangemaakt is.

**Sequentiediagram:**

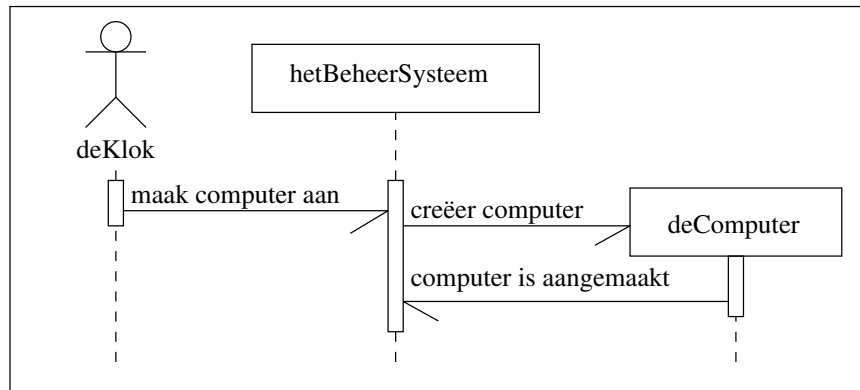


Diagram 2.5: Maak computer aan



### 2.2.3 Verwijder computer

**Actor:** deKlok.

**Beschrijving:** De klok stuurt een signaal naar het BeheerSysteem dat er een computer verwijderd mag worden. Het beheersysteem maakt dan een keuze uit alle computers en beslist welke computer er verwijderd mag worden. Hij stuurt dan een bericht naar deze computer, deze computer zal dan een antwoord terugsturen naar het beheersysteem dat deze verwijderd zal worden, zodat we deze ook uit de lijst van computers kunnen halen.

**Sequentiediagram:**

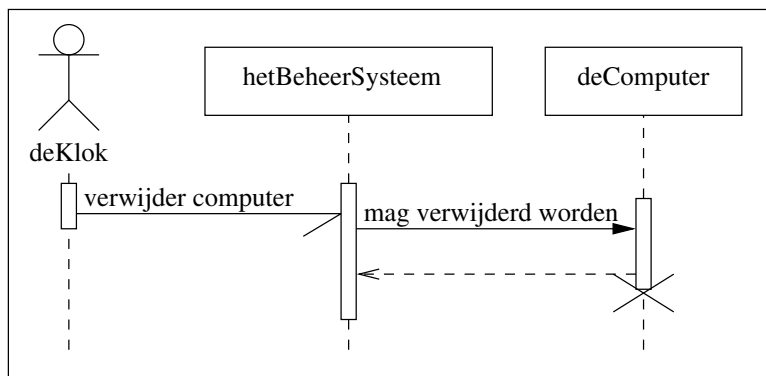


Diagram 2.6: Verwijder computer

## 2.2.4 Kuis computer

**Actor:** deKlok.

**Beschrijving:** De klok stuurt een bericht naar het BeheerSysteem dat er computer mag gekuist worden, dit wil zeggen alle virussen verwijderen. Het beheersysteem zal dan willekeurig een computer kiezen, die hij kan kiezen uit de visualisatiebibliotheek, en een bericht sturen naar deze computer dat de virussen mogen verwijderd worden en zijn status van IsBesmet op false mag zetten.

**Sequentiediagram:**

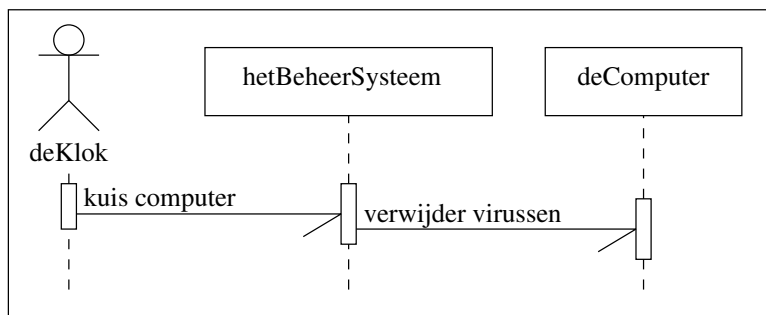


Diagram 2.7: Kuis computer

### 2.2.5 Stuur bericht

**Actor:** deKlok.

**Beschrijving:** De klok stuurt een bericht naar het beheersysteem dat er bestanden mogen doorgestuurd worden van een computer naar andere computers. Het beheersysteem roept de methode `stuurBestand` op van een willekeurig geselecteerde computer, deze zal dan een keuze maken welk soort bestand er verstuurd zal worden, dit kan een goed bestand zijn of een slecht bestand. Hierna roept de computer van al zijn burens de methode `ontvangbestand` op en stuurt dit bestand dan door. Dit alles gebeurt asynchroon zodat de computer niet hoeft te wachten op een antwoord van de te ontvangen computer.

**Sequentiediagram:**

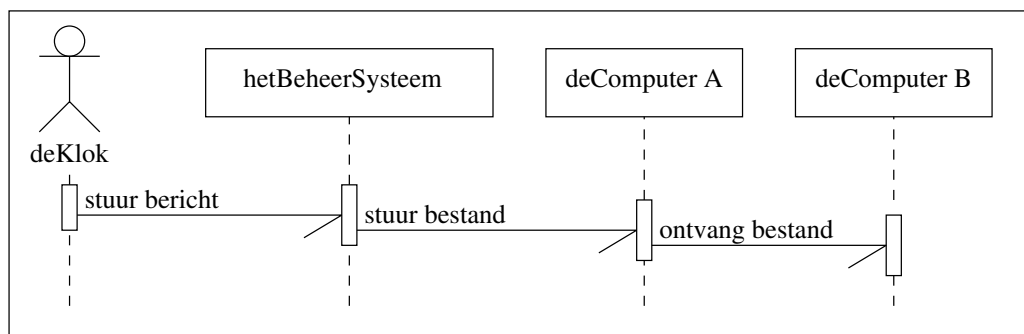


Diagram 2.8: Stuur bericht

## 2.3 Woordenboek

### 2.3.1 Beheersysteem

**leesConfig():** Leest het config bestand, config.js in en construeert de nodige objecten.

**startDemo():** De demo mode van het programma wordt nu gestart. Dit wil zeggen dat vanaf nu er automatisch computers aangemaakt zullen worden, virussen verspreid zullen worden, computers schoongemaakt kunnen worden en ook computers kunnen verwijderd worden.

**kuisComputer():** Het beheersysteem zal willekeurig een computer kiezen uit de lijst van aangemaakte computers en zal er 1 van verwijderen. Bij het verwijderen legt het beheersysteem de verbindingen tussen de buurcomputers. Om dit te realiseren zal het beheersysteem gebruik maken van de volgende lidfuncties:

- Computer:verwijderVirussen()

**maakComputerAan():** Zoals gezegd zal deze functie een nieuwe computer aanmaken en zelf beslissen met wie hij verbonden zal worden.

**stuurBericht():** Het beheersysteem zal beslissen uit de lijst van computers welke computers er een bestand mogen sturen naar hun bureaus. Deze functie maakt gebruik van de volgende lidfuncties:

- Computer:stuurBestand()

**verwijderComputer():** Het beheersysteem zal willekeurig een computer kiezen uit de lijst welke computer er verwijderd zal worden. Hiervoor maakt hij gebruik van de volgende lidfuncties:

- Computer:geefVerbondenComputers()
- Computer:kanVerwijderdworden()

### 2.3.2 Computer

**kanVerwijderdWorden():** Deze methode geeft terug of de computer kan verwijderd worden. Dit wordt bepaald door zijn staat isInternet. Dit wil zeggen dat deze computer een lijst bevat met alle virussen en zodoende niet kan verwijderd worden indien we willen dat alle virussen nog doorgegeven kunnen worden.

**ontvangBestand(Bestand):** Het beheersysteem zal willekeurig een computer kiezen uit de lijst welke computer er verwijderd zal worden. Hiervoor maakt hij gebruik van de volgende lidfuncties:

- Bestand:geefVerspreidingssnelheid()
- Bestand:geefType()
- Bestand:geefNaam()

**stuurBestand(Computer):** Via deze methode wordt een buur gekozen uit de lijst van verbonden computers. Naar deze computer zullen we dan een bestand sturen die bijvoorbeeld een virus kan bevatten.

**verwijderBestanden():** Hierin wordt de computer gekuist van alle virussen die hij heeft en wordt de staat isBesmet op false gezet.

**geefVerbondenComputers():** Via deze methode kunnen we een lijst verbonden burens van de computer terugkrijgen.

### 2.3.3 Bestand

**geefVerspreidingsSnelheid():** Hierin kunnen we de verspreidingssnelheid van een virus terugkrijgen.

**geefType():** Via deze methode krijgen we het type bestand terug.

**geefNaam():** Zoals de methode al zegt krijgen we hier de naam van het bestand terug.

### 2.3.4 VisualisatieBibliotheek

**forceDirected():** De Visualisatie bibliotheek ondersteunt meerdere types van grafieken. Degene die hier gebruikt moet worden is het ForceDirected type.

**loadJSON():** Laad de config in met de data van het netwerk. Dit moet in het correcte formaat zijn.

**compute():** Zal de posities van de computers berekenen en de verbindingen ertussen. Dit levert telkens andere posities op. Dit mag enkel in het begin van de simulatie gebruikt worden.

**computeIncremental():** Voert de bovenstaande functie uit, maar in zo een manier dat de resources optimaal gebruikt worden.

**plot():** Tekent de berekende posities van de computers. Als er iets hertekent moet worden, zal deze functie gebruikt moeten worden.

## 2.4 Aanvullingen

Deze paragraaf is optioneel, en kan gebruikt worden om elementen van het model in onder te brengen die niet bij elk project voorkomen, zoals

- Functioneel model, met eventuele gegevensstroomdiagrammen en formules/algoritmes.
- Statendiagrammen.
- Databankontwerp in de vorm van EPR-diagrammen, tabeldefinities en dergelijke.

In deze paragraaf worden ook bijkomende technische elementen vermeld. Zo dient bijvoorbeeld, bij het gebruik van het MVC-paradigma, hier vermeld worden welke elementen de view en de controller uitmaken.

## Hoofdstuk 3

# Realisatie

Dit hoofdstuk is optioneel. Het wordt gebruikt om een aantal ontwerpbeslissingen aan te geven die bij realisatie belangrijk zijn. Voorbeelden van dergelijke beslissingen zijn de keuze van de programmeertaal en/of de ontwikkelingsomgeving en het gebruik van bepaalde pakketten en technieken.

## Hoofdstuk 4

# Gebruik

In dit hoofdstuk wordt het gebruik en de installatie beschreven.

### 4.1 L<sup>A</sup>T<sub>E</sub>X-gebruikers

Het is zeker niet de bedoeling dat deze tekst uitgroeit tot een handleiding LaTeX. Enkel een paar punten specifiek voor de Hogeschool of voor informaticastudenten worden hier vermeld. Interessante info over LaTeX zoals gebruikt door Gentse scriptiestudenten, inclusief een handleiding: `\http://latex.ugent.be`.

### 4.2 installatie

Eenvoudige installatie: dump volgende bestanden in de directory waar je wil werken:

- `logohogent.eps`
- `logohogent.pdf`
- `iiiscriptie.sty`
- `project.tex`

Gebruik `project.tex` als sjabloon om te beginnen. Voor een beschrijving van de macro's verwijzen we naar de lijst in het woordenboek. `project.tex` bevat voorbeelden van het gebruik van alle macro's.

Nette installatie: je kan ook de elementen van het pakket op de geëigende plaatsen van je T<sub>E</sub>X-installatie onderbrengen. Raadpleeg hiervoor de handleiding.

### 4.3 Opmerkingen

1. Het pakket gaat ervan uit dat de afmetingen van marges niet veranderd werden.



2. Het pakket veronderstelt dat het **graphicx**pakket gebruikt wordt, met ofwel de optie **pdftex**, ofwel de optie **dvips**, of zonder optie. Dezelfde optie moet doorgegeven worden aan het **iiiscriptie**pakket.
3. Verwijzingen. De gemakkelijkste manier is om de standaardstijl van Latex te gebruiken. Dit geeft verwijzingen van de vorm [?]. Het gebruik van **BIBTEX** vergroot natuurlijk de mo(g)e(i)lijkheden.
4. URL's worden, zoals op <http://www.math.uiuc.edu/~hildebr/tex/bibliographies.html> wordt uitgelegd, gezet met het `\url{}`-commando.

Succes ermee.

## 4.4 Niet-L<sup>A</sup>T<sub>E</sub>X-gebruikers

Niet-L<sup>A</sup>T<sub>E</sub>X-gebruikers volgen raadplegen het `project.pdf`-bestand. Ze dienen nauwkeurig de indeling en de lay-out van dit bestand te volgen.

## Hoofdstuk 5

# Logboek

Hieronder vind je een lijst met wie wat juist gedaan heeft op welk moment.

Log van <b>Jan Acke</b>		
26/11/10	16u45-17u45	Eerste labo
01/12/10	08u15-10u15	Tweede bijeenkomst - sequentiediagrammen aanmaken, takendiagram aanpassen en overlopen van geschikte software
03/12/10	15u45-17u15	Tweede labo
08/12/10	08u15-10u15	Derde bijeenkomst - CRC kaarten, verdeling van het werk
10/12/10	15u45-17u45	Vierde bijeenkomst
17/12/10	15u45-17u45	Vijfde bijeenkomst
22/12/10	15u45-17u45	Finale bijeenkomst
<b>totaal</b>		12 uur 30 minuten

Log van <b>Davy Loose</b>		
26/11/10	16u45-17u45	Eerste labo
01/12/10	08u15-10u15	Tweede bijeenkomst - sequentiediagrammen aanmaken, takendiagram aanpassen en overlopen van geschikte software
03/12/10	15u45-17u15	Tweede labo
08/12/10	08u15-10u15	Derde bijeenkomst - CRC kaarten, verdeling van het werk
22/12/10	15u45-17u45	Finale bijeenkomst

(voortgezet op volgende pagina)

---

Voortzetting log van **Davy Loose**

---

<b>totaal</b>	8 uur 30 minuten
---------------	------------------

---

---

Log van **Simon Gaeremynck**

---

26/11/10	16u45-17u45	Eerste labo
27/11/10	10u00-12u30	Zoeken naar mogelijkheden voor hergebruik & prototype in Javascript
01/12/10	08u15-10u15	Tweede bijeenkomst - sequentiediagrammen aanmaken, takendiagram aanpassen en overlopen van geschikte software
01/12/10	20u00-21u00	Prototype: Feature checking
03/12/10	15u45-17u15	Tweede labo
06/12/10	20u00-20u45	Prototype: tests + simulatie van bestanden verzenden
08/12/10	08u15-10u15	Derde bijeenkomst - CRC kaarten, verdeling van het werk
10/12/10	11u00-11u45	Berichten versturen
10/12/10	15u45-17u45	Vierde bijeenkomst
17/12/10	15u45-17u45	Vijfde bijeenkomst
22/12/10	15u45-17u45	Finale bijeenkomst
<b>totaal</b>	17 uur 30 minuten	

---

---

Log van **Christian Vuerings**

---

26/11/10	16u45-17u45	Eerste labo
27/11/10	13u00-13u45	De code van het prototype bestuderen en aanpassen doorvoeren
28/11/10	18u00-18u45	Eerste versies maken van het taken -en klassendiagram
28/11/10	18u45-19u15	Initiele versie aanmaken van het latex document
01/12/10	08u15-10u15	Tweede bijeenkomst - sequentiediagrammen aanmaken, takendiagram aanpassen en overlopen van geschikte software
02/12/10	18u00-19u00	Sequentiediagrammen ontwerpen in xfig
03/12/10	15u45-17u15	Tweede labo

(voortgezet op volgende pagina)

---

---

Voortzetting log van **Christian Vuerings**

---

03/12/10	17u30-18u00	Sequentie, klassen - en takendiagrammen aanpassen en beschrijving updaten waar nodig
03/12/10	18u00-18u45	Aanmaken van de initiale versie van de CRC-kaarten
08/12/10	08u15-10u15	Derde bijeenkomst - CRC kaarten, verdeling van het werk
08/12/10	18u30-19u15	Tonen van informatie over een computer op random tijdstippen
10/12/10	15u45-17u45	Vierde bijeenkomst
17/12/10	15u45-17u45	Vijfde bijeenkomst
17/12/10	17u15-17u30	Update van bepaalde sequentiediagrammen en van het klassendiagram
22/12/10	15u45-17u45	Finale bijeenkomst
<b>totaal</b>		17 uur 45 minuten

---