# Assignment 3

Steven Schenk, Cees-Jan Nolen, Robin Bakker, William de Visser, Robert Kraaijeveld – Groep 1 INF2C

Inhoudsopgave

[Assignment 3 1](#_Toc432363775)

[A. 3](#_Toc432363776)

[B. 9](#_Toc432363777)

[C. 11](#_Toc432363778)

[Implementation plan 11](#_Toc432363779)

[Doel van het project 11](#_Toc432363780)

[Indeling 11](#_Toc432363781)

[Timeline (les)week 7-10 11](#_Toc432363782)

[Timeline (les)week 11-14 11](#_Toc432363783)

[Timeline (les)week 15-16 11](#_Toc432363784)

[Rolverdeling en verantwoordelijkheden 11](#_Toc432363785)

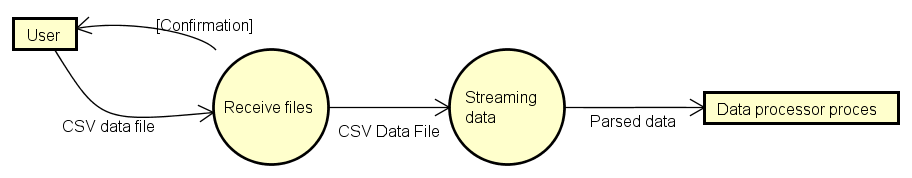
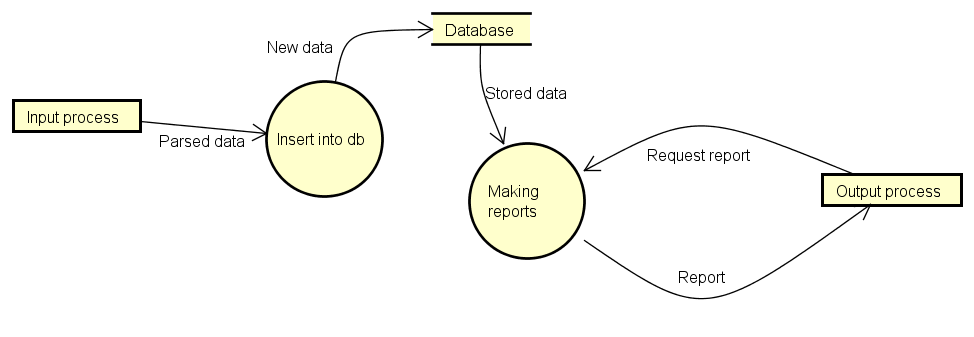
[Technieken 12](#_Toc432363786)

[Communicatie 12](#_Toc432363787)

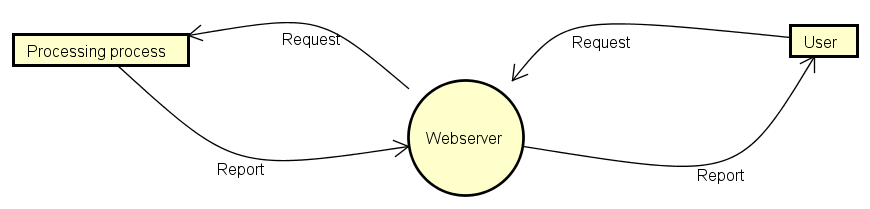
## A.

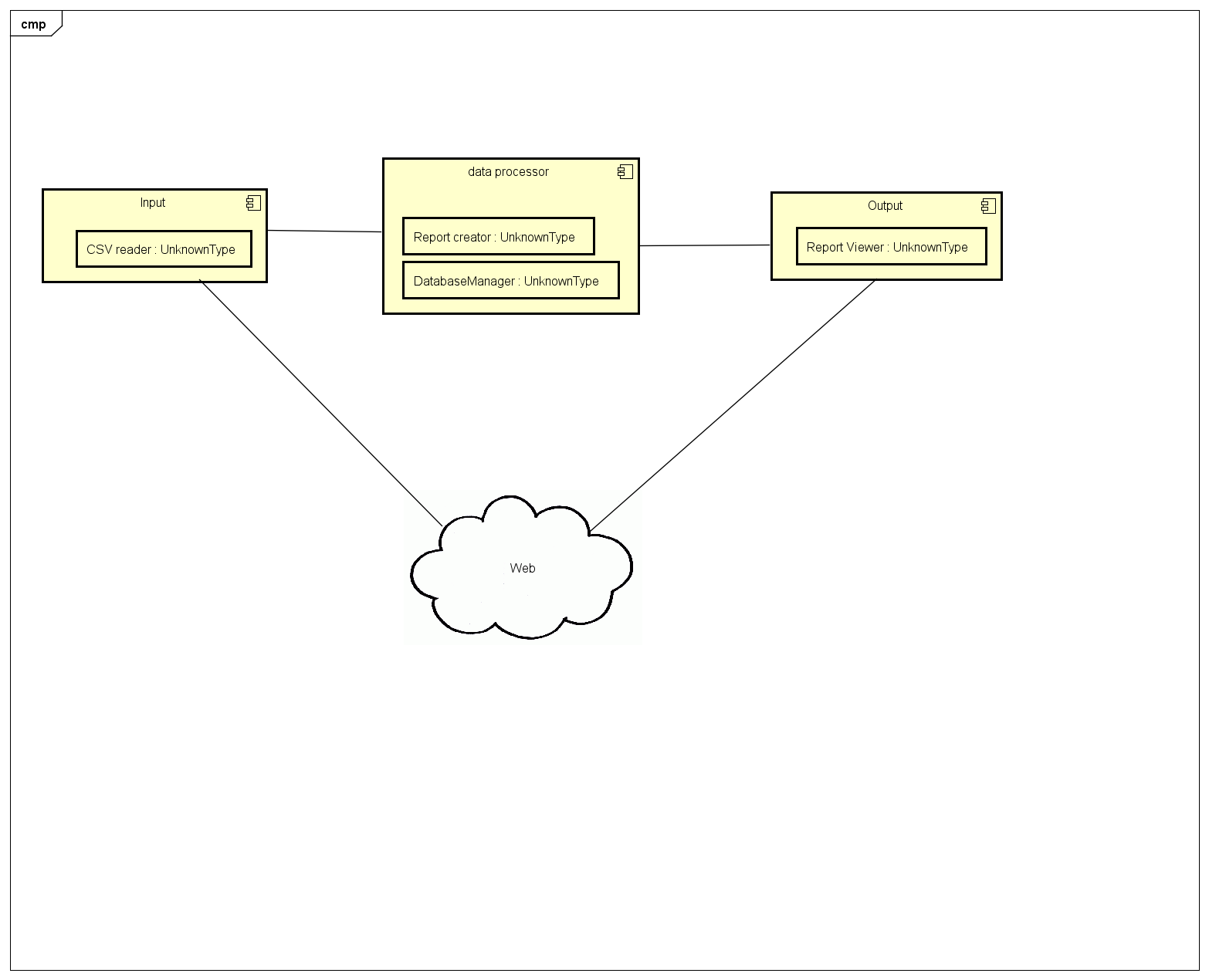
Wij hebben naast onze al bestaande dfd/contextdiagrammen die u hieronder kunt vinden sequence diagrammen gemaakt, die elk bij 1 van onze Use Cases horen. (Zie voor deze use cases Assignment .)

*Model 1:DFD van Input process*

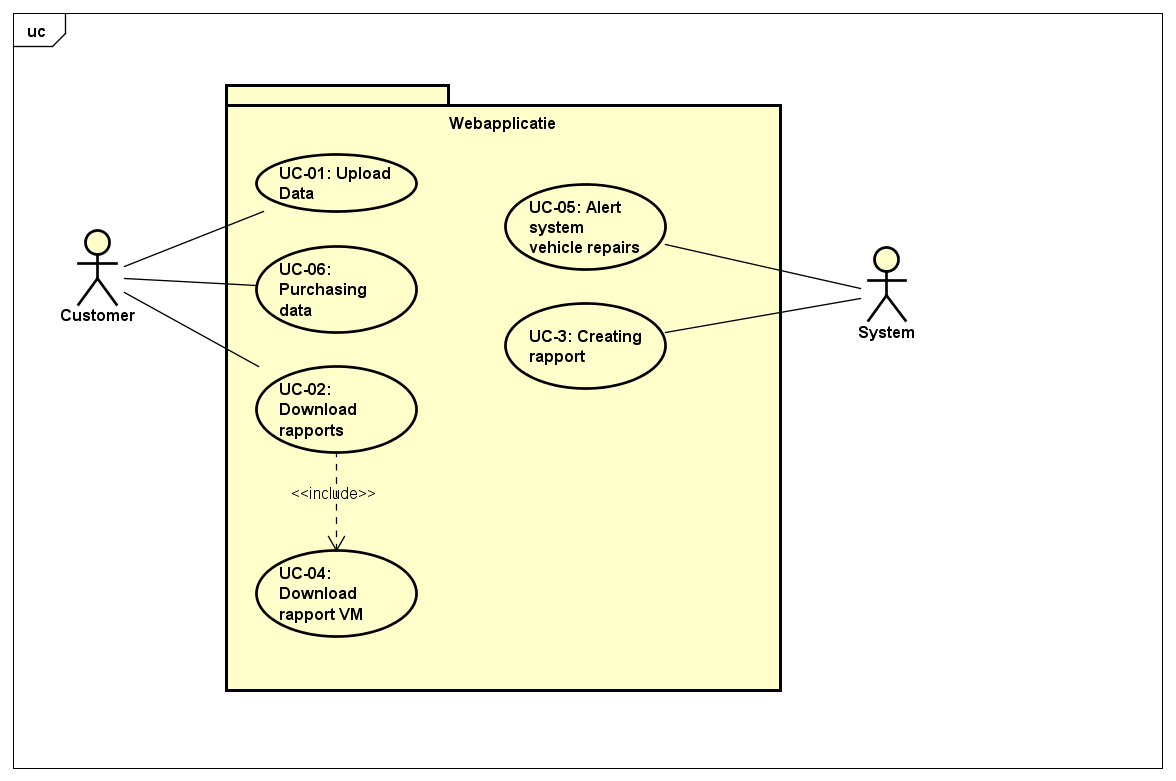
*Model 2: DFD processing process*

*Model 3: DFD Output process*

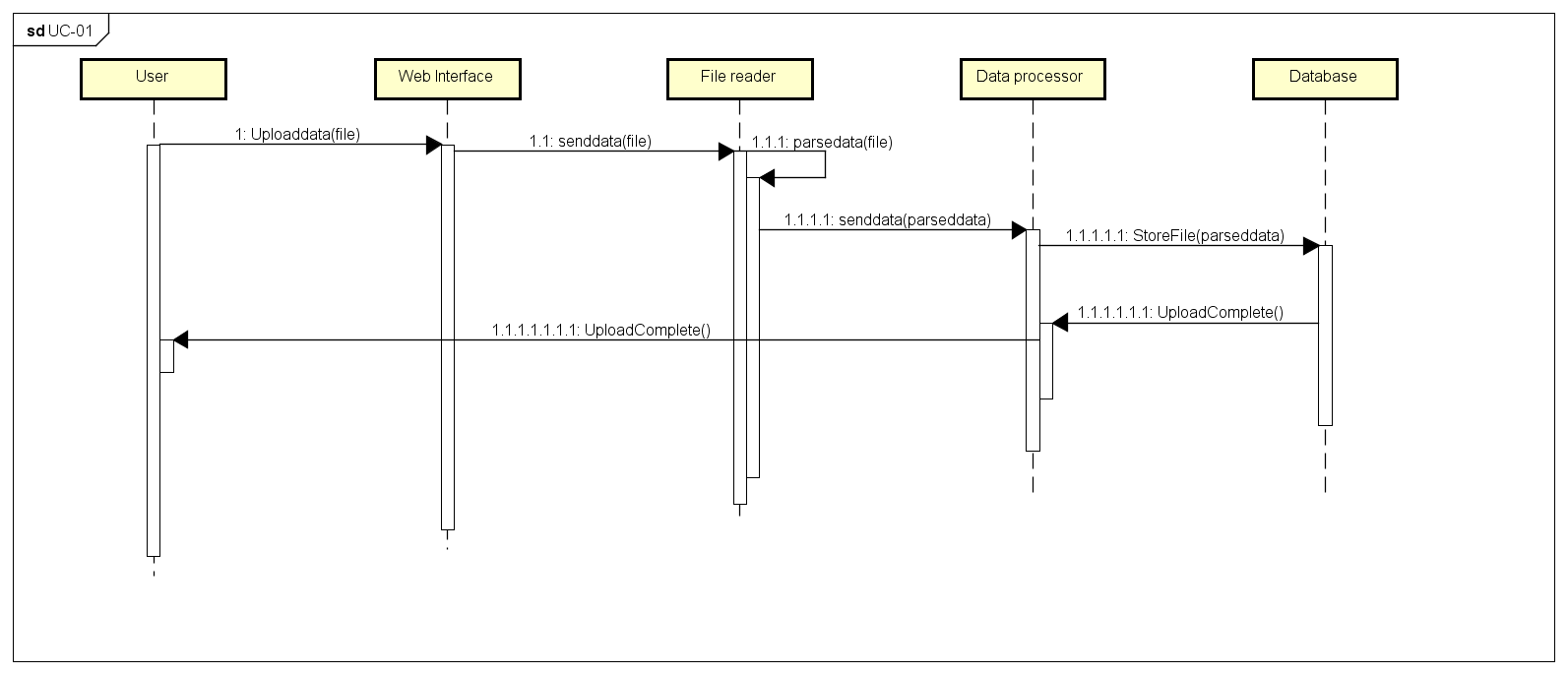
**

*Model 4: High level system architecture diagram*

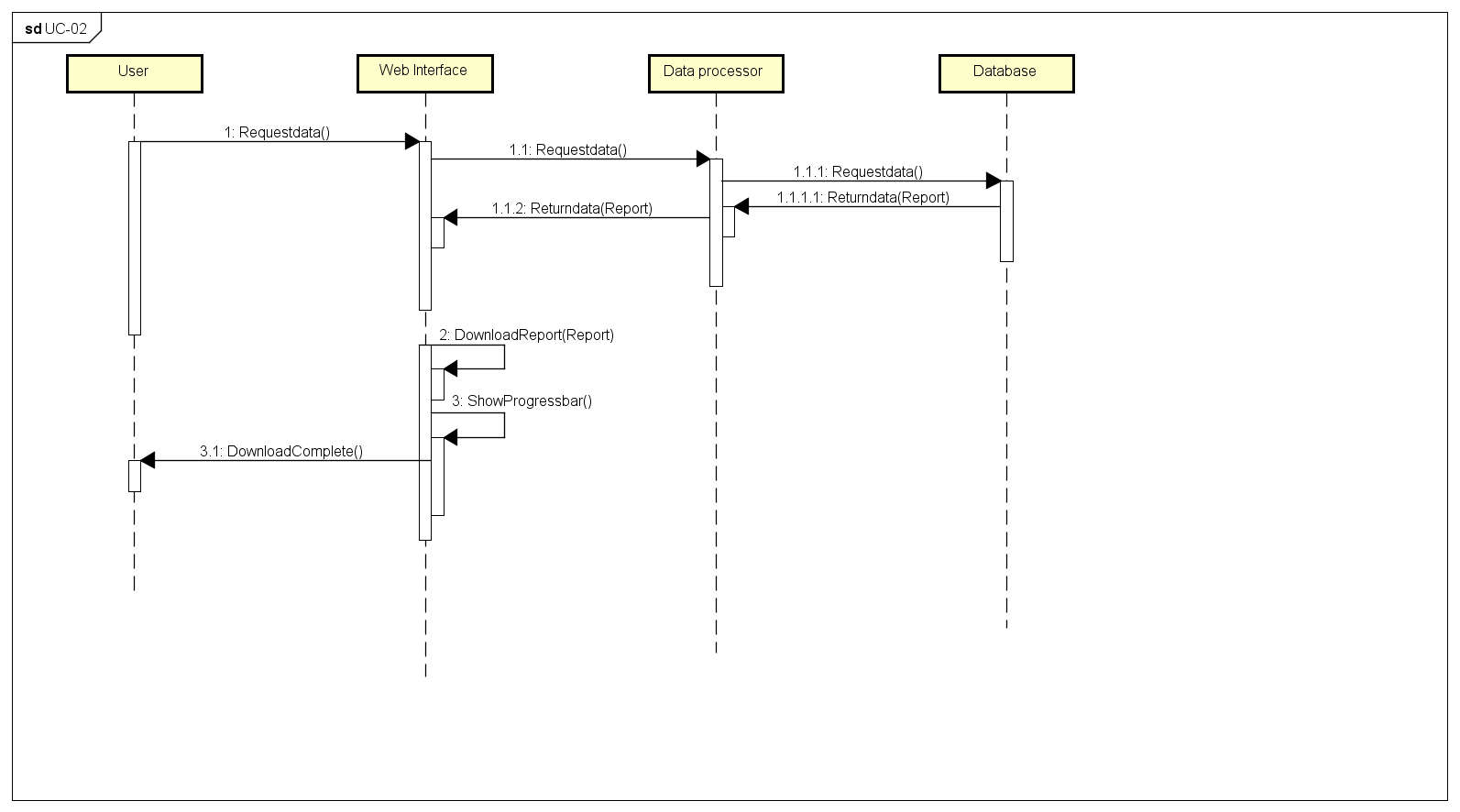
*Use cases diagram:*



*Sequence diagram 1: Use Case 1*

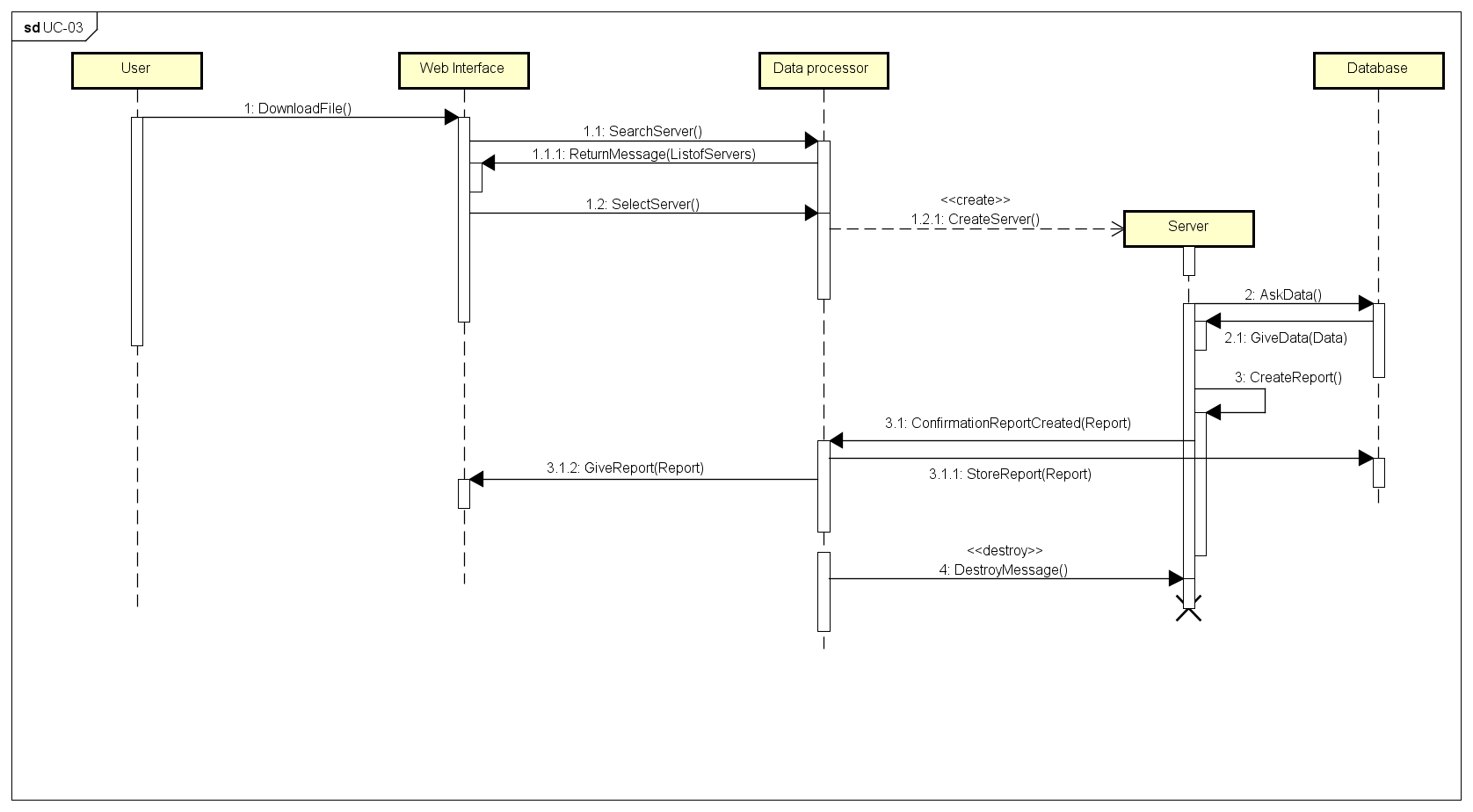


*Sequence diagram 2: Use Case 2*

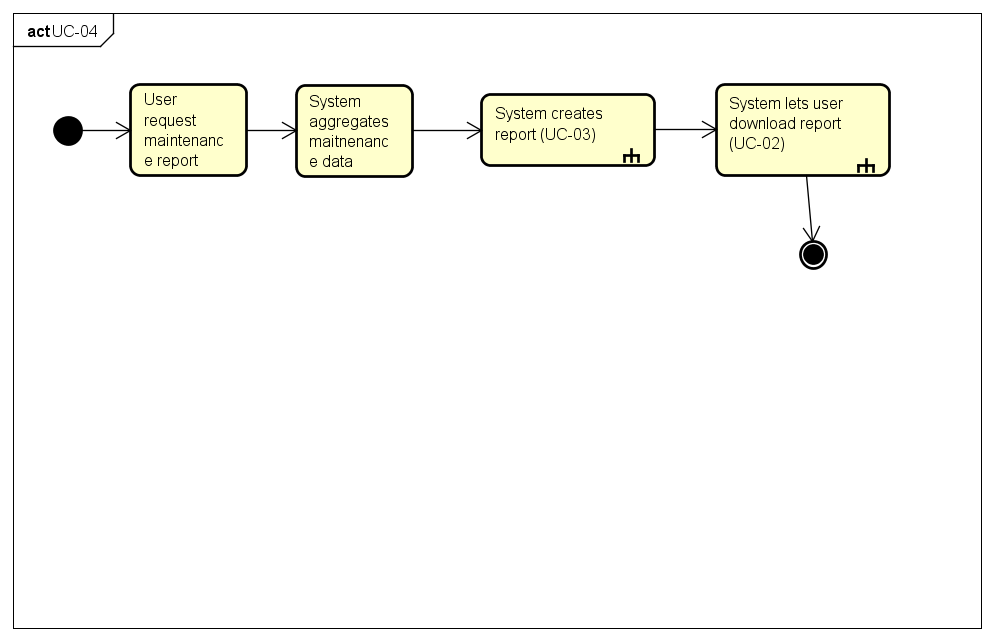


*Sequence diagram 2: Use Case 2 (hierboven)*

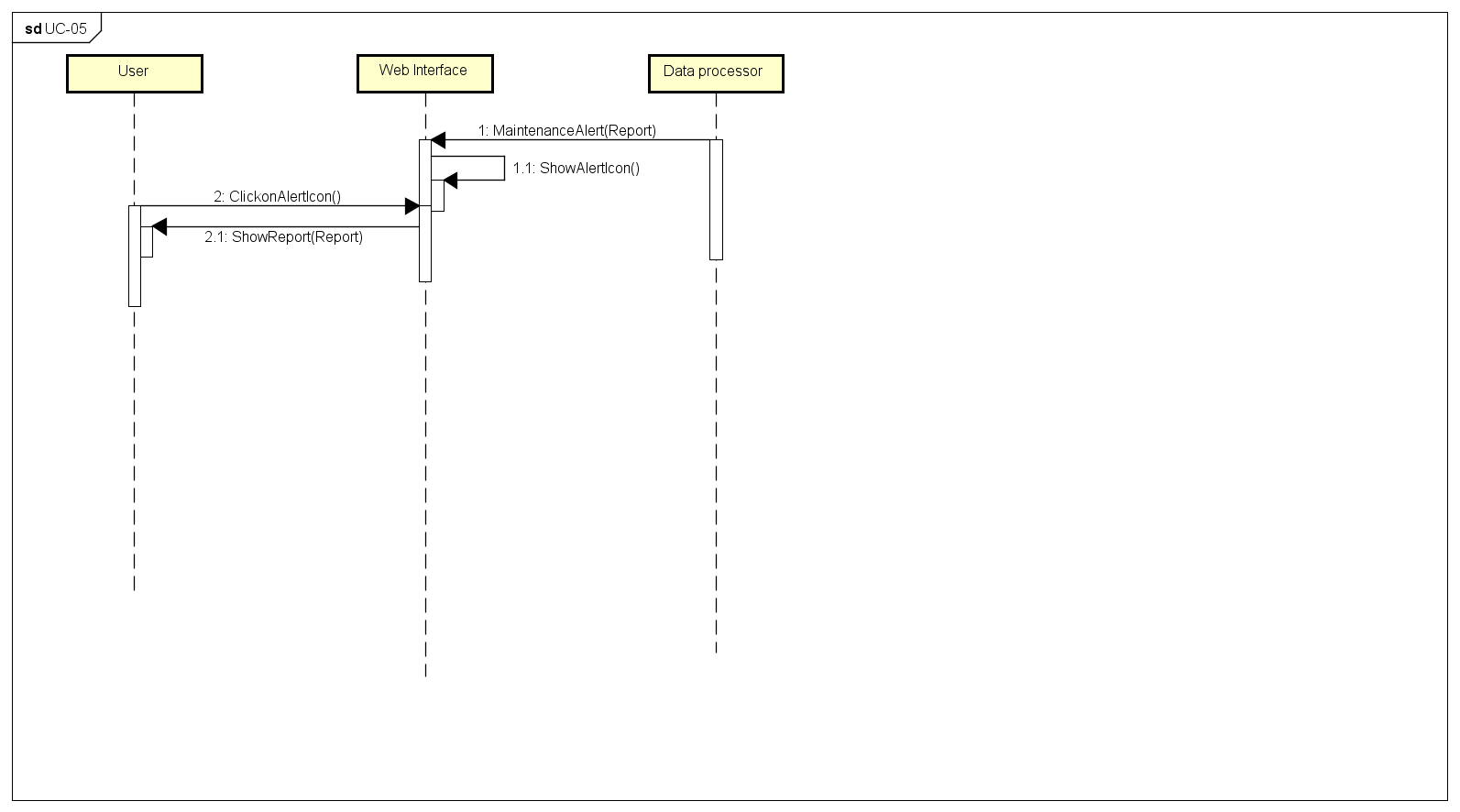
*Sequence diagram 3: Use Case 3*

**

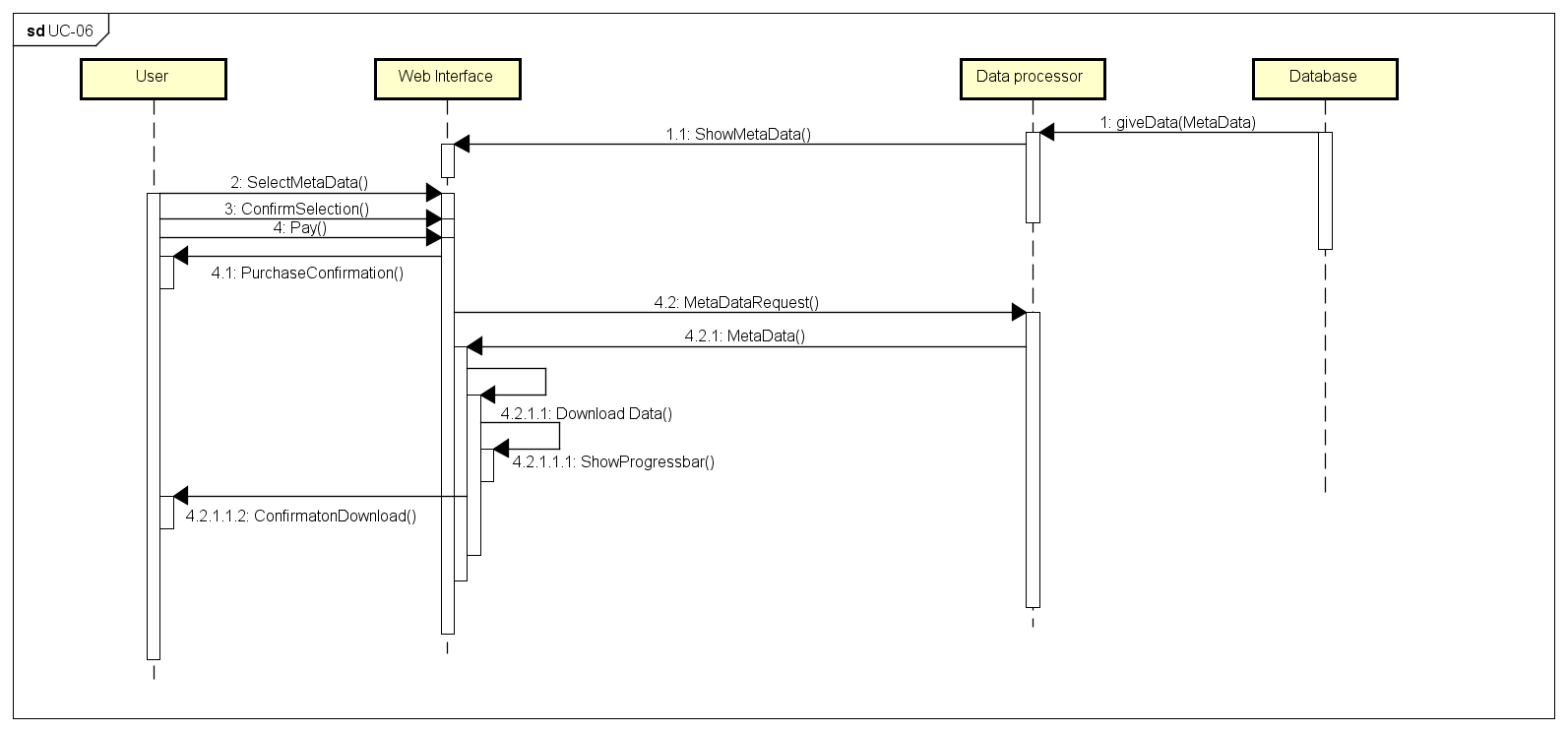
*Activity diagram: Use case 4*

**

*Sequence diagram 5: Use Case 5*

**

*Sequence diagram 6: Use Case 6*



## B.

Als eerste hebben wij enkele standaardvragen mbt. Tot de opzet van de architectuur van ons systeem beantwoord: Onze antwoorden op deze vragen scheppen duidelijkheid over enkele beslissingen die wij omtrent onze systeem-architectuur hebben gemaakt.

**Vraag 1: Hoe zullen de structurele componenten van het systemen onderverdeeld worden in sub-componenten?**

Het systeem zal in 3 componenten verdeeld worden: Input, verwerking en output. Deze 3 componenten zullen als losse, geïsoleerde services geimplementeerd worden zodat zij onafhankelijk

**Vraag 2: Welke fundamentele aanpak zal gebruikt worden om het systeem te structureren?**

Wij zullen SOA(Service Oriented Architecture) gebruiken. De sub-componenten zullen geisoleerd van elkaar gebouwd worden zodat zij makkelijk over verschillende servers verdeeld kunnen worden.

**Vraag 3: Is er een generieke applicatie architectuur die als sjabloon voor het design van het systeem gebruikt kan worden?**

Jazeker, namelijk het onderverdelen van het systeem in de Input, Verwerking en Output componenten; dit gebeurt vaak in veel verschillende software-projecten. Het output-component zal ook gebruik maken van de MVC (model-view-controller) architectuur.

**Vraag 4: Hoe zal het systeem verdeeld worden over hardware-cores of processors?**

De specifieke verdeling van het systeem over de beschikbare hardware-infrastructuur zal op een later moment tijdens de implementatie van het systeem besloten worden.

**Vraag 5: Wat voor architecturele stijlen of patronen zullen er (wellicht) gebruikt worden?**

- SOA (Service Oriented Architecture).

- Het Client-Server model.

- MVC (model-view-controller)

**Vraag 6: Hoe zal de architectuur van het systeem gedocumenteerd worden?**

Het zal in het SRS-document gedocumenteerd worden, vooral door middel van DFD diagrammen.

**Vraag 7: Wat voor strategie zal er gebruikt worden om de operaties van decomponenten van het systeem te besturen?**

Welke strategieën precies gebruikt zullen worden zal pas in de implementatie-fase beslist worden . Zeker is echter dat MVC gebruikt gaat worden, aangezien het PHP framework Laravel deze strategie gebruikt. Ook zullen wij waarschijnlijk pThreads voor C++ gaan gebruiken om de Input en Verwerkings-componenten onderling met elkaar te laten communiceren.

**Vraag 8: Wat voor architecturele organizatie is het beste om de non-functional requirements van het system te creeëren?**

We hebben onze nonfunctional requirements gemaakt volgens het SRS formaat.

# C.

# Implementation plan

## Doel van het project

Het project heeft als doel om een systeem te maken dat snel data kan verwerken. De data wordt geleverd in csv format. Deze moet zo snel mogelijk geparsed worden en in een database gestopt worden. Daarna moet het systeem deze data kunnen aggregeren tot verslagen, zodat de gebruiker deze verslagen kan downloaden.

## Indeling

### Timeline (les)week 7-10

In deze fase zullen we vooral onderzoeken. We gaan ons verdiepen in de platformen en programmeertalen. Hierna zullen we gebruik maken van UML om blauwdrukken van ons systeem te maken. Dit zullen we vooral doen met classen diagrammen. Er zal in deze periode ook al kleine stukjes code geprogrammeerd worden. Dit zal vooral in het output proces zijn. De design van de site zal helemaal ontworpen zijn en eventueel enkele functionaliteiten ingebouwd.

### Timeline (les)week 11-14

In deze fase zullen alle processen ontwikkeld worden tot bruikbare componenten. Ieder proces moet geïsoleerd kunnen werken en moet op meerde servers kunnen functioneren die voldoen aan de hardware eisen.

### Timeline (les)week 15-16

In deze periode zullen de onafhankelijke processen aan elkaar worden gekoppeld en het systeem zal gevormd worden tot een geheel. Daarnaast zal er ook grondig getest worden en zal de code refactored worden waar nodig is. In deze fase worden de puntjes op de i gezet.

## Rolverdeling en verantwoordelijkheden

|  |  |
| --- | --- |
| Persoon | Taken/verantwoordelijkheden |
| Steven Schenk | * Processing proces * Output proces (prototype) * UML diagrammen * Aanspreekpunt voor begeleider |
| Robert Kraaijeveld | * Processing proces * UML diagrammen |
| Cees-Jan Nolen | * Input proces * UML diagrammen |
| William de Visser | * Output proces * UML diagrammen |
| Robin Bakker | * Output proces * UML diagrammen |

## Technieken

|  |  |
| --- | --- |
| Wat | Technieken |
| Input proces | * C++ * pThread (onderzoek vereist) * Sockets |
| Processing proces | * C++ * ODB (ORM) * Sockets |
| Output proces | * PHP * Laravel framework * Bootstrap |
| Distributie | * Git * Github |

### Communicatie

Voor de communicatie tussen processen zal er gebruik gemaakt worden van sockets. Dit betekent dus dat er van te voren goed afgesproken moet worden over welke poorten we communiceren over het TCP protocol. Deze afspraken moeten nog worden gemaakt. Informatie over welke poorten we gebruiken en welke data type we gebruiken voor uitwisseling van bestanden/informatie is nog niet duidelijk en zal nader bekend gemaakt worden.