

# HO GENT

Software Development Project I  
(academiejaar 24-25)

# Table of Contents

1. Voorwoord .....	1
2. Deelname aan het project .....	2
3. Planning gedurende het semester .....	3
4. Contactmomenten .....	3
5. Puntenverdeling .....	4
5.1. Product (eindresultaat) .....	4
5.2. Productontwikkeling .....	5
5.3. Proces .....	6
6. Eindproducten .....	7
7. Opdracht .....	8
8. Eindpresentatie .....	9
9. Een aantal extra's .....	10
9.1. Projectopgave Alhambra: The Dice Game .....	10
9.2. Overzicht criteria met beschrijving levels .....	11
9.3. Planning coachingsmomenten / werkbelasting .....	12
9.4. Logboek .....	13
9.5. Opvolgingsformulier .....	13

# 1. Voorwoord

Beste student

Welkom in semester 2!

Tijdens dit semester werk je in team aan een opdracht waarin heel wat vaardigheden van een IT'er aan bod komen. Je leert door middel van een **vakoverschrijdende opdracht** in team werken maar ook zelfstandig informatie opzoeken. Op die manier verwerf je kennis en zet je die ineens om in de praktijk.

In deze brochure vertellen we je alles over dit opleidingsonderdeel Software Development Project I. We proberen klaarheid te scheppen in de chaos die vaak – ook in het werkveld – gepaard gaat met het opstarten van een project.

Op het einde van het project beoordelen wij niet enkel het eindproduct. We houden ook rekening met je individuele inbreng, je presentatie, je werkhouding, je rol en bijdrage in de groep en je vorderingen tijdens het verloop van het project. Voor dit opleidingsonderdeel voorzien we 4 studiepunten.

Projectwerking sluit goed aan bij wat de Amerikaanse pedagoog John Dewey ooit zei: ‘een school moet geen voorbereiding zijn op het leven van later, ze moet het leven zelf zijn’. Door te werken aan dit project kan je je voorbereiden op je professionele leven van later (lees: binnenkort): competent individueel en als teamplayer handelen in een voortdurend veranderende beroepspraktijk.

Ik wens je, namens alle lectoren die instaan voor de begeleiding en coaching, alvast veel succes!

Fien Spriet

Coördinator Software Development Project I

## 2. Deelname aan het project

In deze brochure vind je de globale projectopdracht en uitleg over de projectwerking.

Dit project is vakoverschrijdend. De leerstof gezien in de opleidingsonderdelen **Databases**, **Software Analysis** en **Object-oriented Software Development I** moeten hier toegepast worden. Ook hetgeen je dit semester leert in **Object-oriented Software Development II** moet onmiddellijk in praktijk omgezet worden.

Een projectgroep bestaat uit vijf studenten, niet noodzakelijk uit dezelfde klas. Iedereen in de projectgroep werkt mee aan elk onderdeel (analyse, ontwerpen, testen, programmeren, databank).

**Inschrijven** doe je in de Chamilo-cursus Software Development Project I **vanaf dinsdag 11 februari rond 19u00**. Direct na de online infosessie starten de inschrijvingen. Kies een groep en schrijf jezelf in. Je teamleden moeten zich in dezelfde groep inschrijven. Iedereen schrijft zich in **ten laatste op donderdag 13 februari 23u59**.

Als je inschrijft voor een bepaalde groep, dan leg je ook vast wanneer je wekelijkse afspraak zal ingepland worden. Je legt de halve dag vast, je afspraak wordt tijdens die halve dag gepland. Dat hoeft niet elke week op hetzelfde uur te zijn.

### Contactmomenten groepen Software Development Project I (SDP I) / System Engineering Lab (SEL)

In principe kies je hetzelfde groepsnummer voor zowel SDP als SEL. Deze lijst geeft een overzicht van de ingeplande niet overlappende contactmomenten. Een specifiek tijdslot krijg je nog eenmaal iedereen is ingeschreven in een groep.

Groepen Gent  
Groepen Aalst  
Groepen VC  
Groepen TIAO

Groep	SDP	SEL	Groep	SDP	SEL	Groep	SDP	SEL	Groep	SDP	SEL	Groep	SDP	SEL
G01	DI VM	DO NM	G26	DI VM	VRIJ VM	G51	VRIJ VM	DI VM	G68	Do NM	Vr VM	G82	Do NM	Vr VM
G02	DI VM	DO NM	G27	DI VM	VRIJ VM	G52	VRIJ VM	DI VM	G69	Do NM	Vr VM	G83	Do NM	Vr VM
G03	DI VM	DO NM	G28	DI VM	/	G53	VRIJ VM	DI VM	G70	Do NM	Vr VM	G84	Do NM	Vr VM
G04	DI VM	DO NM	G29	DI VM	/	G54	VRIJ VM	DI VM	G71	Do NM	Vr VM	G85	Do NM	Vr VM
G05	DI VM	DO NM	G30	DI VM	/	G55	VRIJ VM	DI VM	G72	Do NM	Vr VM			
G06	DI VM	DO NM	G31	DO NM	DI VM	G56	VRIJ VM	DO NM	G73	Do NM	Vr VM			
G07	DI VM	DO NM	G32	DO NM	DI VM	G57	VRIJ VM	DO NM	G74	Do NM	Vr VM	G91		TIAO
G08	DI VM	DO NM	G33	DO NM	DI VM	G58	VRIJ VM	DO NM	G75	Do NM	Vr VM	G92		TIAO
G09	DI VM	DO NM	G34	DO NM	DI VM	G59	VRIJ VM	DO NM	G76	Do NM	Vr VM	G93		TIAO
G10	DI VM	DO NM	G35	DO NM	DI VM	G60	VRIJ VM	DO NM	G77	Do NM	Vr VM	G94		TIAO
G11	DI VM	DO NM	G36	DO NM	DI VM	G61	VRIJ VM	DO NM	G78	Do NM	Vr VM	G95		TIAO
G12	DI VM	DO NM	G37	DO NM	DI VM	G62	VRIJ VM	DO NM	G79	Do NM	Vr VM	G96		TIAO
G13	DI VM	DO NM	G38	DO NM	DI VM	G63	VRIJ VM	DO NM	G80	Do NM	/	G97		TIAO
G14	DI VM	DO NM	G39	DO NM	VRIJ VM	G64	VRIJ VM	/				G98	/	TIAO
G15	DI VM	DO NM	G40	DO NM	VRIJ VM	G65	VRIJ VM	/						
G16	DI VM	DO NM	G41	DO NM	VRIJ VM									
G17	DI VM	VRIJ VM	G42	DO NM	VRIJ VM									
G18	DI VM	VRIJ VM	G43	DO NM	VRIJ VM									
G19	DI VM	VRIJ VM	G44	DO NM	VRIJ VM									
G20	DI VM	VRIJ VM	G45	DO NM	/									
G21	DI VM	VRIJ VM	G46	VRIJ VM	DI VM									
G22	DI VM	VRIJ VM	G47	VRIJ VM	DI VM									
G23	DI VM	VRIJ VM	G48	VRIJ VM	DI VM									
G24	DI VM	VRIJ VM	G49	VRIJ VM	DI VM									
G25	DI VM	VRIJ VM	G50	VRIJ VM	DI VM									

Deelnemen aan het project betekent dat je:

- wekelijks (vanaf week 4 t.e.m. week 11) aanwezig bent op de afspraak. Dit is telkens een **verplichte** aanwezigheid!
- wekelijks ongeveer 10u werkt aan dit project (week 3 - week 12)

### 3. Planning gedurende het semester

Wee k	TODO
1	online infosessie bijwonen + inschrijven in een groep
2	werkomgeving opzetten (uitleg documenten + instructiefilmpjes)
3	starten (UC1 console, UC1 databank, UC1 testen, UC2 analyse)
4 - 11	wekelijks contactmoment waarbij vooruitgang opgevolgd kan worden, vragen besproken kunnen worden
12	indienen eindresultaat van het project
13	technische eindpresentatie

### 4. Contactmomenten

De beoordeling van het opleidingsonderdeel gebeurt op basis van **permanente evaluatie**. Dit betekent dat je dus voortdurend geëvalueerd zal worden tijdens de contactmomenten. Elke afwezigheid moet je dan ook wettigen. Indien je onmogelijk een bepaalde afspraak kan nakomen, verwittig dan de begeleidende lector én je groepsleden. De afwezigheid moet officieel gemeld worden (ten laatste op de dag van de afspraak zelf) via iBamaFlex, binnen de 48u na de melding moet de correcte wettiging opgeladen worden.

Studenten die wel aanwezig zijn op alle afspraken, maar omwille van dwingende redenen afwezig zijn op de eindpresentatie, krijgen een AFW (afwezig) voor de evaluatie van de eerste zitting. Deze studenten worden verwezen naar de tweede zitting en halen tijdens de feedback de opgave voor de tweede zitting op.

Wat gebeurt er heel concreet bij **ongewettigde afwezigheden**?

Als je **2 of meer ongewettigde afwezigheden** laat optekenen, dan krijg je als **examencijfer** voor het opleidingsonderdeel de vermelding **afwezig**. Deze afwezigheid neem je mee naar de 2de zitting.

De punten worden negatief beïnvloed door elke ongewettigde afwezigheid:

- je progressie voor die week kan niet beoordeeld worden, hierdoor daalt je eindcijfer.
- punten groepswork dalen (pas op: dit cijfer neem je mee naar 2de zit, zie puntenverdeling)

Wat gebeurt er als je aanwezig bent maar **onvoldoende betrokken** bent bij het project?

Tijdens een contactmoment (én thuis) moet er hard gewerkt worden aan dit project. Elke student

moet voldoende bijdragen. Een gebrek aan betrokkenheid kan ook resulteren in een vermelding afwezig voor dit opleidingsonderdeel.

Bij een gebrek aan betrokkenheid krijg je als student een uitnodiging tot een persoonlijk gesprek met je begeleidende lector. De andere groepsleden kunnen ook uitgenodigd worden op dit gesprek. Indien je afwezig bent op dit gesprek of het leidt niet tot een terug correcte betrokkenheid in het project, krijg je die vermelding afwezig. Elk teamlid wordt hier via mail van op de hoogte gebracht. Om de mails te versturen, gebruiken de lectoren het standaard communicatiekanaal, nl. Chamilo.

Deze informatie kan je nalezen in het [DOER](#):

- Artikel 4: Deelname aan onderwijs- en evaluatieactiviteiten (paragraaf 1,2,3 en 4)
- Artikel 6: Gevolgen bij afwezigheid bij evaluatie

Het DOER (zie hierboven) is aanvullend op het [OER](#).

## 5. Puntenverdeling

De uiteindelijke verdeling van het totaalcijfer:



Hierbij wordt rekening gehouden met volgende criteria:

### 5.1. Product (eindresultaat)

#### criterium 1: Volledigheid product qua vorm

1. einddossier met
  - inhoudstafel
  - genummerde blz.
  - per UC analyse, ontwerp, databank
  - reflectie ivm groepswork van elk groepslid
2. Eclipse project met code en VP UML-diagrammen
3. gegenereerde javadoc API van het domein

### **Criterium 2: Volledigheid product qua functionaliteit**

Alle functionaliteiten zijn beschreven in de use cases.

### **Criterium 3: Volledigheid product qua niet functionele requirements**

- user interface (console / grafisch)
- meertalige applicatie

## **5.2. Productontwikkeling**

### **Criterium 1: Analyseren**

Producten zijn:

- activity diagram
- domeinmodel
- SSD, bijhorende OC's

conform de regels toegelicht in Software Analysis en conform de opgave

### **Criterium 2: Ontwerpen**

Opstellen van het DCD:

- DCD sluit aan op SSD's
- DCD bevat wat beschreven werd in OC

conform de regels toegelicht in OOSDI en conform de opgestelde analyse

### **Criterium 3: Testen**

Elke "pure" domeinklasse, d.w.z. NIET DomeinController en repositoryklassen, wordt via JUNIT testen voldoende getest.

### **Criterium 4: Programmeren**

Kwaliteitsvol implementeren betekent:

- respect voor OO
- respect voor de 3 lagen
- correct documenteren doorheen volledige proces
- robuuste applicatie
- in de code vind je de juiste methodes vanop het DCD terug

conform de regels toegelicht in OOSDI, OOSDII én conform het gemaakte ontwerp

## **Criterium 5: Databank**

Eigen opgestelde databank voldoet aan:

- conceptueel model ERD
- overeenkomstig relationeel model
- overeenkomstige implementatie in relationele databank

Beheren betekent:

- lezen van data
- schrijven van data

conform de regels toegelicht in Databases

## **5.3. Proces**

### **Criterium 1: Registreren**

Logboek en opvolgingsformulier aanvullen.

### **Criterium 2: Plannen**

Wekelijkse planning opstellen, werken volgens die planning en controleren of je op schema zit.

### **Criterium 3: Teamwork**

Goed samenwerken en dit ook stimuleren.

### **Criterium 4: Voorbereiding wekelijks contactmoment**

Vorbereiding betekent:

- Voorleggen en op de hoogte zijn van het werk van de voorbije week
- Vragen opsommen en meebrengen

### **Criterium 5: Omgaan met feedback / adviezen volgen**

Dit gaat om zowel inhoudelijke feedback als procesmatige feedback.

### **Criterium 6: Stiptheid**

Wees elke afspraak (online of op school) stipt op tijd.

Per criterium worden 5 levels beschreven (zie extra's). Twee maal (halverwege het semester en op het eind) zal je voor jezelf en de anderen moeten bepalen op welk level je je bevindt per criterium en dat ook wat duiden. Zelfkennis is een belangrijke schakel in het succesvol werken in team.



Opgelet: je hoeft niet op elk criteria level 4 te halen om te kunnen slagen. De beschrijving van deze levels dient vooral om beter zicht te krijgen op jouw kunnen. Anderzijds, als je overal op level 0 zit, kan je wel onmogelijk slagen.

Wat gebeurt er als je in juni niet slaagt voor dit opleidingsonderdeel?

Wie niet slaagt krijgt een **gepersonaliseerde taak** voor EP3. Deze taak werk je **individueel** uit en je vertrekt altijd vanuit het project dat gemaakt werd met jouw groep in semester2. Tijdens de tweede examenkans kom je dan **mondeling** je uitgewerkte taak bespreken.

Voor het onderdeel groepswork kan je geen nieuwe taak krijgen. **Dit cijfer (/30) neem je mee uit de eerste zitting.** Het is dus van cruciaal belang dat je je inzet voor het groepswork tijdens het semester! Als je een afwezig hebt op dit onderdeel (bv. te veel ongewettigd afwezig, onvoldoende betrokkenheid) neem je dit ook mee naar de 2de zitting.

## 6. Eindproducten

Op het einde van de projectwerking geef je een **verzorgd dossier** af met daarin:

- Voorblad (zie Chamilo);
- ERD, relationeel model, script met DDL-instructies (per use case indien van toepassing);
- Analyse en ontwerp (domeinmodel, activiteitendiagram, SSD, OC en DCD) per use case (of scenario indien van toepassing);
- 1 A4 blad tekst per groepslid met je ervaringen ivm groepswork: wat heb je geleerd, wat waren de moeilijkheden, wat zijn jouw sterke punten, ...

Via de opdrachtenmodule van Chamilo dien je je project in:

Maak **1 zip-bestand** met daarin:

- Verzorgd dossier
- Eclipse project met de javacode
- De gegenereerde Java documentatie van de code (Javadoc)
- De volledige analyse en ontwerp (vpp-bestand van Visual Paradigm)
- Script met DDL-instructies van de MySQL database
- Exportbestand van de MySQL database (alle data).

**Dit afgewerkte dossier laad je op via de opdrachtenmodule in Chamilo vóór maandag 12 mei 23u59**

Je maakt via **Teams** (opname van een vergadering met het volledige team) een **filmpje** waarin je jullie eindresultaat demonstreert. Het eindresultaat is een applicatie met een grafische user interface waarbij je alle gevraagde functionaliteiten kan tonen. Ook dit filmpje is groepswork, elk teamlid komt aan het woord. **Dit filmpje is via teams beschikbaar ten laatste maandag 12 mei 23u59.** Dit filmpje is geen montage, er mag niets geknipt worden, het is dus de opname van een live demo.

Tijdens het semester bouw je ook een **digitaal dossier** op. Elke groep krijgt zijn eigen documentenfolder op OneDrive. Deze folder is enkel toegankelijk voor de groepsleden én alle begeleidende lectoren.

Per week wordt hier het **logboek** geplaatst van de voorbije week. Ook het **opvolgingsformulier** krijgt hier een plaats.

Verder kunnen alle documenten waar je als groep samen aan werkt, hier gestockeerd worden (bv werkdossier, planningsdocumenten, ...).

Bewaar niets enkel lokaal op jouw pc. Zorg ervoor dat alles op elk moment online toegankelijk is!

De recentste code staat op github, de recentste analyse + ontwerp vinden we op VP online, andere documenten deel je via OneDrive voor alle teamleden én de begeleidende lector.

## 7. Opdracht

1. Maak per use case (bevat alle vereiste functionaliteit):
  - a. Het **ontwerp van de database**. Doorloop hiervoor de stappen gezien tijdens de lessen Databases: opstellen van een ERD, daarna het relationeel model en tot slot het script met de DDL-instructies om de databank te implementeren in MySQL.
  - b. De **analyse** zoals aangeleerd in de lessen Software Analysis:
    - domeinmodel
    - alle zinvolle systeem sequentiediagrammen met bijhorende OC's
    - activiteitendiagrammen
  - c. Het **ontwerp** zoals aangeleerd in de lessen OOSDI : DCD
  - d. Een **console-applicatie** die ons toelaat om via deze interface de functionaliteit van de domein- en persistentielaag te testen. De implementatie van een drielagenproject leerde je in de lessen OOSD I (én OOSD II). Op een bepaald moment zal je mogen overschakelen naar een grafische user interface.
2. Ontwikkel een **grafische user interface**. Je bent vrij de GUI naar eigen inzicht vorm te geven. Wees creatief. Zorg ervoor dat de gevraagde functionaliteit gerealiseerd wordt. Begin niet aan de GUI voordat je begeleidende lector daarvoor **toestemming** geeft. Let op: ook **de console-applicatie blijft werken!**
3. De applicatie moet **meertalig** zijn. Gebruik hiervoor resourcebundles. Hoe deze precies werken, moet je zelf opzoeken! Je hoeft niet alle teksten te vertalen, maar technisch moet het eenvoudig te vertalen zijn!
4. Per use case wordt een **werkdossier** aangevuld met het gerealiseerde werk:
  - Het ERD, RM, script met DDL-instructies
  - De analyse (domeinmodel, activiteitendiagrammen en SSD's en OC's)

- Het ontwerp (DCD)

Dit werkdoosier vormt een belangrijke basis voor je **einddoosier**. Houd het dus heel nauwkeurig bij.

5. Per groep wordt er een **logboek** bijgehouden. Dit logboek bevat een chronologisch overzicht van de taken die de verschillende groepsleden op zich nemen. Telkens een groepslid aan het project werkt noteert die de details van het gedane werk (omschrijving en tijdsregistratie) meteen ook in het logboek.
6. Als voorbereiding op de wekelijkse afspraak met je begeleidende lector vul je het **opvolgingsformulier** in (wat is er gebeurd de voorbije week en wat zullen we doen de komende week, groepsafspraken, ....). Noteer zeker ook al jullie vragen zodat die beantwoord kunnen worden tijdens het coachingsmoment.

## 8. Eindpresentatie

Van 19 tot en met 23 mei volgt de technische eindpresentatie. Deze presentatie wordt bijgewoond door je begeleidende lector.

Elke student krijgt specifieke technische vragen (op basis van ingediend project, dossier en observaties tijdens het semester).

Het al dan niet goed beantwoorden van deze vragen beïnvloedt **héél sterk** je persoonlijk cijfer voor het project (deel informatica 70%).

**Voorbeeldvragen** zijn:

1. Hoe werken resourcebundels? Wat is het nut ervan? Toon in de code hoe je ermee werkt. Leg uit.
2. Wat is het nut/voordeel van de javadoc-comment? Toon aan.
3. Toon aan dat je applicatie voor UC.... een robuuste applicatie is.
4. Wat is de taak/zijn de verantwoordelijkheden van een DomeinController? Staaf telkens met een voorbeeld in jullie project.
5. Wat zijn de verantwoordelijkheden van een repositoryklasse? Staaf telkens met een voorbeeld in jullie project.
6. Wat doet een mapperklasse? Bespreek de code van een methode in zo'n mapperklasse.
7. Illustreer hoe en tot welke producten je tijdens de analyse van UC x bent gekomen (domeinmodel/SSD/OC/AD).
8. Illustreer hoe doing en knowing verantwoordelijkheden gescheiden worden gehouden in het project.
9. Illustreer op welke manier je GRASP patronen (controlller/expert/creator) hebt toegepast bij het maken van het ontwerp.
10. Leg bondig het 3-lagen model uit en illustreer de toepassing ervan adhv een voorbeeld uit je code.

11. Het ontwerp van de grafische laag is heel programmeertaalafhankelijk. Leg uit hoe je deze laag opgebouwd hebt in jullie project.
12. Eventafhandeling en/of wisselen van scherm: toon hiervan een voorbeeld uit jullie project en leg de code uit.
13. Illustreer hoe je jouw applicatie getest heb aan de hand van analyse schema's voor UCx.
14. Bespreek hoe en waarom jouw domeinmodel, EERD en DCD verschillend zijn.
15. Toon aan en leg uit hoe je zeker bent dat domeinklasse X goed werkt.

## 9. Een aantal extra's

### 9.1. Projectopgave Alhambra: The Dice Game



In **Alhambra: The Dice Game** probeert elke speler met zo gunstig mogelijke dobbelresultaten te behalen meerderheden te behalen in de 6 gebouwensoorten om daarmee overwinningspunten te verdienen.

Het spel bestaat uit 3 rondes en afhankelijk van het aantal spelers krijgt elke speler 3, 4 of 5 beurten per ronde. Per beurt mag de speler maximaal 3 keer de dobbelstenen werpen en vervolgens 1 van de geworpen gebouwensoorten selecteren om zijn zetsteen te plaatsen. Aan het einde van de ronde worden zetstenen met de beste positie beloond met gebouwenpunten. Op basis van deze gebouwenpunten worden vervolgens overwinningspunten uitgedeeld.

De opgave van het spel werd uitgeschreven in een aantal use cases. De use cases zijn het contract met de klant, hierin staat beschreven hoe het precies allemaal moet werken. Begin dus niet in het wilde weg te programmeren. Volg exact wat er beschreven staat in de use case. Doorloop alle fasen van een iteratie (analyse, ontwerp, testen, implementatie). Als de ene iteratie afgewerkt is kan je met de volgende starten.

1. UC1 - Registreer nieuwe speler
2. UC2 - Start nieuw spel
3. UC3 - Speel spel
4. UC4 - Speel ronde
5. UC5 - Speel beurt

Dit filmpje toont hoe het spel gespeeld kan worden: <https://www.youtube.com/watch?v=OpsmVcwwa0w>



Om het spel te ontwikkelen, houd je je wel strikt aan de uitgeschreven use cases. Sommige spelregels zijn namelijk aangepast ten opzichte van het filmpje!

## 9.2. Overzicht criteria met beschrijving levels

Categorieën	Criteria	Level 0	Level 1	Level 2	Level 3	Level 4
Product	Volledigheid product qua vorm	Het product is <b>onvolledig</b> : Er ontbreken essentiële onderdelen.	Het product is <b>volledig</b> met alle onderdelen, maar <b>sommige onderdelen</b> zijn <b>onvolledig</b> .		Het product is <b>volledig</b> en conform de eisen.	Het product is <b>volledig</b> en conform de eisen op een <b>professionele wijze</b> .
	Volledigheid product qua functionaliteit	Het product bevat <b>minder dan de helft</b> van de gevraagde functionaliteiten.	Het product bevat <b>minstens de helft</b> van de gevraagde functionaliteiten.	Het product bevat <b>alle</b> gevraagde functionaliteiten.	Het product bevat <b>alle</b> gevraagde functionaliteiten.	Het product bevat <b>alle</b> gevraagde functionaliteiten.
		De opgeleverde applicatie is <b>niet stabiel</b> , vormt <b>geen geheel</b> , moet vaak herstart worden.	De opgeleverde applicatie is <b>niet stabiel</b> maar vormt <b>één geheel</b> .	De opgeleverde applicatie is <b>bijna volledig stabiel</b> , maar af en toe ontbreekt een nodige validatie.	De opgeleverde applicatie is <b>volledig stabiel</b> . Alle gevraagde validaties zijn aanwezig.	De opgeleverde applicatie is <b>volledig stabiel</b> . Alle gevraagde validaties zijn aanwezig. Extra validaties zijn toegevoegd.
	Volledigheid product qua niet functionele requirements	De student kan de vooropgestelde <b>niet functionele requirements niet</b> uitwerken.	De student kan de vooropgestelde <b>niet functionele requirements</b> voor een beperkt deel van de functionaliteiten uitwerken.	De student kan de vooropgestelde <b>niet functionele requirements</b> uitwerken, sturing nodig.	De student kan de vooropgestelde <b>niet functionele requirements</b> autonoom uitwerken.	De student kan de vooropgestelde <b>niet functionele requirements</b> autonoom uitwerken (bewust en met nodige toelichting).

Categorieën	Criteria	Level 0	Level 1	Level 2	Level 3	Level 4
P r o d u c t o n t w i k k e l i n g	Analyseren	De student kan de producten uit de analysefase <b>niet</b> opstellen en toelichten.	De student kan de producten uit de analysefase <b>opstellen en toelichten</b> , maar weinig kwaliteitsvol. Er is <b>frequent</b> intense bijsturing nodig.	De student kan de producten uit de analysefase kwaliteitsvol opstellen en toelichten. Er is soms bijsturing nodig (ganse periode).	De student kan de producten uit de analysefase kwaliteitsvol opstellen en toelichten. De bijsturing kan geleidelijk aan verminderen.	De student kan de producten uit de analysefase kwaliteitsvol opstellen en toelichten. Er is <b>amper</b> bijsturing nodig.
	Ontwerpen	De student kan de producten uit de ontwerpfase <b>niet</b> opstellen en toelichten.	De student kan de producten uit de ontwerpfase <b>opstellen en toelichten</b> , maar weinig kwaliteitsvol. Er is <b>frequent</b> intense bijsturing nodig.	De student kan de producten uit de ontwerpfase kwaliteitsvol opstellen en toelichten. Er is soms bijsturing nodig (ganse periode).	De student kan de producten uit de ontwerpfase kwaliteitsvol opstellen en toelichten. De bijsturing kan geleidelijk aan verminderen.	De student kan de producten uit de ontwerpfase kwaliteitsvol opstellen en toelichten. Er is <b>amper</b> bijsturing nodig.
	Testen	De student kan <b>geen</b> correcte unittesten schrijven voor de verschillende methodes uit de domeinklassen.	De student kan <b>beperkt</b> unittesten schrijven voor de verschillende methodes uit de domeinklassen.	De student kan <b>unittesten</b> schrijven voor de verschillende methodes uit de domeinklassen maar de testen zijn <b>niet</b> volledig.	De student kan <b>unittesten</b> schrijven voor de verschillende methodes uit de domeinklassen en de testen zijn <b>volledig</b> .	De student kan <b>unittesten</b> schrijven voor de verschillende methodes uit de domeinklassen, de testen zijn <b>volledig</b> . De student doet dit volledig op een <b>TDD</b> wijze en moet hiertoe niet verplicht worden.
	Programmeren	De student kan functionaliteiten <b>niet</b> programmeren.	De student kan functionaliteiten <b>weinig</b> kwaliteitsvol programmeren. Er is <b>frequent</b> intense bijsturing nodig.	De student kan functionaliteiten kwaliteitsvol programmeren. Er is soms bijsturing nodig (ganse periode).	De student kan functionaliteiten kwaliteitsvol programmeren. De bijsturing kan geleidelijk aan verminderen en de student <b>vraagt gericht hulp</b> waar nodig.	De student kan functionaliteiten kwaliteitsvol programmeren. Er is <b>amper</b> bijsturing nodig.
	Databank	De student kan de databank <b>niet</b> opstellen conform de eisen.	De student kan de databank <b>opstellen</b> conform de eisen. Er is <b>frequent</b> intense bijsturing nodig.	De student kan de databank <b>opstellen</b> conform de eisen. Er is soms bijsturing nodig (ganse periode).	De student kan de databank <b>opstellen</b> conform de eisen met <b>beperkte</b> bijsturing.	De student kan de databank <b>opstellen</b> conform de eisen. Er is <b>amper</b> bijsturing nodig.
		De student kan de eigen opgestelde databank <b>niet</b> beheren vanuit de applicatie.	De student kan de opgestelde databank <b>beheren</b> vanuit de applicatie. Er is <b>frequent</b> intense bijsturing nodig.	De student kan de opgestelde databank <b>beheren</b> vanuit de applicatie. Er is <b>soms</b> bijsturing nodig (ganse periode).	De student kan de opgestelde databank <b>beheren</b> vanuit de applicatie en ook <b>toelichten</b> (niet enkel nabootsen van voorbeeldapplicatie). De bijsturing kan geleidelijk aan verminderen.	De student kan de opgestelde databank <b>beheren</b> vanuit de applicatie en ook <b>toelichten</b> (niet enkel nabootsen van voorbeeldapplicatie). Er is <b>amper</b> bijsturing nodig.

Categorieën	Criteria	Level 0	Level 1	Level 2	Level 3	Level 4
P R O C E S	Registreren	De student registreert <b>niet of onvoldoende</b> (aantal keer + correct).	De student registreert op expliciete vraag maar <b>niet altijd correct of niet consequent</b> . Er is <b>frequent</b> intense bijsturing nodig.	De student registreert <b>consequent</b> . Er is soms <b>inhoudelijke</b> bijsturing nodig (ganse periode).	De student registreert <b>spontaan en inhoudelijk</b> meestal goed. De inhoudelijke bijsturing kan geleidelijk aan verminderen.	De student registreert van bij de start <b>spontaan en inhoudelijk correct</b> . Er is <b>amper</b> bijsturing nodig.
	Plannen	De student plant niet: neemt <b>ad hoc</b> taken op, volgt afspraken <b>niet</b> en controleert <b>niet</b> of hij/zij op schema zit.	De student plant zelden: kijkt <b>dag per dag</b> naar de taken, volgt afspraken op als hij/zij <b>erop</b> gewezen wordt, <b>controleert</b> zelden de volledige planning.	De student plant <b>zelfstandig</b> op korte termijn, <b>respecteert</b> de gemaakte <b>weekplanning</b> en controleert af en toe of het hij/zij nog op schema zit.	De student plant <b>zelfstandig</b> op korte termijn, rekening houdend met vorige werkervaringen, <b>respecteert</b> weekplanning, <b>controleert</b> planning en <b>stuurt</b> bij waar nodig.	De student plant <b>zelfstandig</b> op korte termijn, rekening houdend met vorige werkervaringen, <b>respecteert</b> weekplanning, <b>controleert</b> planning en <b>stuurt</b> bij waar nodig. <b>Geen</b> bijsturing nodig, van bij de start op hoog niveau!
	Teamwork	De student werkt <b>niet samen</b> in groep en werkt voor zichzelf binnen de groepsoopdracht.	De student werkt <b>soms</b> samen, maar houdt zich vooral bezig met eigen taken en opdrachten.	De student werkt <b>samen</b> in groep maar stelt <b>af en toe</b> het <b>eigenbelang</b> boven het groepsbelang.	De student werkt <b>samen</b> in groep en <b> motiveert</b> anderen. De student stelt het groepsbelang meestal boven het eigenbelang.	De student werkt <b>samen</b> in groep en <b>stimuleert</b> samenwerken. De student stelt het groepsbelang prioriteer.
	Vorbereiding wekelijks gesprek	De student heeft <b>niets</b> voorbereid.	De student heeft <b>gedeeltelijke</b> of onsamenvattende voorbereiding gemaakt.	De student heeft een <b>relevante</b> voorbereiding gemaakt.	De student heeft een <b>grondige</b> voorbereiding inclusief vragen gemaakt.	De student heeft een <b>grondige</b> voorbereiding gemaakt in een ruimere context waardoor hij <b>anticipeert</b> op mogelijke antwoorden.
	Omgaan met feedback / Adviezen opvolgen	De student houdt <b>geen</b> of <b>amper</b> rekening met feedback van groepsgenoten en lector. De student reageert onverschillig of aangevallen.	De student houdt rekening met feedback van groepsgenoten en lector <b>na</b> <b>meermalen</b> aandringen. De student reageert soms onverschillig of aangevallen.	De student houdt <b>meestal</b> rekening met feedback van groepsgenoten en lector.	De student houdt rekening met feedback van groepsgenoten en lector. <b>Vraagt actief</b> en spontaan feedback.	De student houdt rekening met feedback van groepsgenoten en lector. <b>Vraagt proactief</b> spontaan feedback en geeft zelf suggesties.
	Stiptheid	Is altijd te laat op sessies.	Is sporadisch op tijd op sessies.	Is meestal op tijd op sessies.	Is altijd op tijd op sessies	

## 9.3. Planning coachingsmomenten / werkbelasting

Tijdens de week is sowieso een halve dag voorzien om samen te werken aan dit project. De groep waarin je je inschrijft bepaalt de halve dag (zie rooster bij "Deelname aan het project"). Je bent 2u aanwezig in de klas en tijdens die twee uur heb je een half uur een afspraak met je begeleider.

Het is belangrijk om zoveel mogelijk te halen uit de 30 minuten die je krijgt samen met je begeleidende lector. Zijn er vragen? Verzamel ze op het opvolgingsformulier zodat je ze tijdens de afspraak samen kan overlopen. Zo vergeet je niets en kan je als groep weer een weekje verder werken!

Vergeet niet de andere 2u een plaats te zoeken op school waar je samen kunt verder werken! De

cafetaria, de bibliotheek, werklokaaltjes die je kan reserveren... er zijn heel veel mogelijkheden om af te spreken en te werken!

Naast die halve dag is het aan te raden zeker nog elke week een 6tal uur extra per persoon te werken om dit project te laten slagen.

Totale studietijd: 100u

Week	Werklast per student
1	1,5u
2	6u
3	6u
4 - 11	4u (halve dag) + 6u extra
12	4u
13	1u

Studiefiche:

- Gent: <https://bamaflexweb.hogent.be/BMFUIDetailxOLOD.aspx?a=180984&b=5&c=1>
- Aalst: <https://bamaflexweb.hogent.be/BMFUIDetailxOLOD.aspx?a=181243&b=5&c=1>
- VC: <https://bamaflexweb.hogent.be/BMFUIDetailxOLOD.aspx?a=181115&b=5&c=1>

## 9.4. Logboek

Noteer alle uren die je besteedt aan dit project. Je krijgt geen punten voor je uren, maar op die manier krijgen we een zicht op de tijdsbesteding in een project. Welk luik vraagt meest tijd?

ZATRE - groep XX									
Datum	Omschrijving taak	Student	Tijd in min.	DB	OA	OO	OP	?	commit/revisie #
			↕						
			↕						
			↕						
			↕						
			↕						
			↕						
			↕						
			↕						
			↕						
			↕						

Totaal in uren						
	Globaal	DB	OA	OO	OP	Andere
st1	0	0	0	0	0	0
st2	0	0	0	0	0	0
st3	0	0	0	0	0	0
st4	0	0	0	0	0	0
Groep	0	0	0	0	0	0
Gemiddeld per persoon	0	0	0	0	0	0

## 9.5. Opvolgingsformulier

Elke week vullen we het opvolgingsformulier aan.

- **DONE:** Dit vul je aan vóór de afspraak! Wat heb je de voorbije week gedaan?
- **TODO:** Dit vul je aan vóór de afspraak! Wat ben je van plan te doen de komende week?
- **Hoe verloopt het groepswerk?:** Noteer hier alle probleempjes, vragen, sterke punten, ...

Deel 1 - In te vullen door de studenten

	DONE - Wat werd voorbije week gerealiseerd?
Algemeen	
Student1	
<u>Student2</u>	
<u>Student3</u>	
Student4	

	TODO - Wat plannen we volgende week te realiseren?
Algemeen	
Student1	
<u>Student2</u>	
<u>Student3</u>	
Student4	

Hoe verloopt het groepswerk?