

Modulewijzer GMT minor Web/Mobile:

MEDMWM02-1 (WEM01)

Mobile Applications

With Google Android

Aantal ECTS

4

Opleiding

Grafimatechnologie / CMI

Versie

1.0, start feb 2009

A4-Modulebeschrijving

Modulecode:	MEDMWM02-1 (WEM01)
Modulenaam:	Mobile Applications with Google Android
Belasting (aantal CP):	4 CP
Relatie met ander onderwijs	Deze module gaat in op technische realisatie en biedt basiskennis die in de verdere minor web/mobile noodzakelijk is.
Voorkennis:	Object georiënteerd programmeren (ActionScript 3.0 / Java (ME) / C#)
Programmasoort	Kennisgestuurd
Werkvormen:	Theorie, practicum/workshops 4 uur / week (± 8 uur zelfstudie)
Looptijd	10 weken
Toetsing:	Eindproduct (applicatie) en door studenten te verzorgen workshops/presentaties
Vrijstelling:	via intake assessment (op basis van aantoonbare ervaringen, product + kennis). Deze module is niet binnen een duaal bedrijf te volgen, (tenzij het duale traject ontwikkeling in Android betreft..)
Leermiddelen:	Eigen laptop
leerdoelen:	<p><u>Leerdoelen:</u></p> <p>De student kent:</p> <ul style="list-style-type: none"> - Basisbegrippen, termen en het principe van ontwikkelen van applicaties voor Google Android in Java - De mogelijkheden en beperkingen van het Android platform - Basisbegrippen en termen binnen object georiënteerd denken - Methoden om een applicatie te modeleren (o.a. UML) <p>De student kan:</p> <ul style="list-style-type: none"> - Een gestructureerde werkomgeving opzetten voor Java / Android - Klassen opstellen in Java en omgaan met objecten - Binnen Android applicaties ontwikkelen - Een concept uitwerken tot logische ontwerpen/documentatie en realiseren in Java voor Google Android
Inhoud:	<p>Het hoofddoel van de module is het kunnen ontwikkelen van een applicatie binnen het Google Android platform. Hierbij staan twee andere punten centraal, namelijk: Het in een kort tijdsbestek een nieuw platform (en mogelijk programmeertaal) kunnen doorgronden, op basis van een verscheidenheid aan bronnen. En het op een object georiënteerde manier kunnen analyseren en realiseren van een applicatie.</p> <p>Als eindproduct dient een (eigen) concept voor het Google Android platform in groepsverband (2 personen) te worden gerealiseerd.</p> <p>De module biedt hiertoe houvast op de volgende manieren:</p> <ol style="list-style-type: none"> 1. Algemene introductie Google Android & Java programmeren 2. Inbreng van een aantal globale richtingen binnen Android, waaruit een concept voor een applicatie kan worden afgeleid 3. Workshops (door docent) voor specifieke onderwerpen (naar behoefte van studenten) 4. Wekelijkse workshops door studenten op het gebied waar zij in bezig zijn (beginnend vanaf week 3 of 4) 5. Verplichte diversiteit aan projecten waar studenten aan werken (waardoor er een breder assortiment aan nieuwe kennis wordt ingebracht in de workshops) 6. Ruimte voor kennisoverdracht op basis van ingebrachte problemen
Opmerkingen:	<p>Tijdens de eerste lessen ligt de nadruk op het introduceren van nieuwe en algemeen noodzakelijke kennis. Daarna wordt een groot deel van de kennisinbreng door studenten verzorgd in de vorm van workshops. (Deze workshops tellen mee in de beoordeling van de module!)</p> <p>Let op: Aanwezigheid is in principe verplicht voor de gehele module.</p>
Auteur(s):	M. van Gooswilligen
Versiedatum:	Versie 1.0 29-01-2009, start minor feb. 2009
Module beheerder:	Opleiding GMT / CMI

Inhoudsopgave

1	ALGEMENE OMSCHRIJVING	4
1.1	Inleiding.....	4
	Inhoud.....	5
1.2	Relatie met andere onderwijseenheden.....	6
1.4	Werkvorm(en).....	6
1.6	Toetsing.....	6
2	PROGRAMMA	7
2.1	WELKE ONDERWERPEN KOMEN AANBOD?	7
2.1.1	Onderwerp 1: Programmeren /Java algemeen.....	7
2.1.2	Onderwerp 2: Google Android.....	7
2.1.4	Onderwerp 3: Object georiënteerd denken, ontwerpen en programmeren.....	7
2.1.3	Onderwerp 4: UML.....	7
2.1.5	Onderwerp 5: Threading.....	7
2.2	ONDERZOEKSGEBIEDEN / WORKSHOP ONDERWERPEN	8
2.3	TE BESTUDEREN LITERATUUR EN ANDERE BRONNEN	9
2.4	WEEKSCHEMA	10
3	TOETSING EN BEOORDELING	12
3.1	Procedure.....	12
3.2	Beoordelingscriteria.....	12
3.3	Weging.....	13
3.3	Herkansingen.....	13

1 Algemene omschrijving

1.1 Inleiding

Het mobiele platform heeft zich in korte tijd stevig in onze samenleving weten te verankeren. Net zoals eerder al de computer en het internet onmisbaar zijn geworden, kunnen tegenwoordig ook weinig mensen meer zonder een mobiel apparaat. Daarnaast blijft het toepassingsgebied alsnog toenemen. Zowel voor zakelijke doeleinden als privé zaken komen steeds meer mogelijkheden beschikbaar. Hierdoor is de mobiele markt een zeer interessante markt om jezelf in te bekwamen.

Tot op heden is en blijft Java ME één van het meest gebruikte en flexibele platform om mobiele applicaties in te ontwikkelen. Dit is primair toe te schrijven aan het feit dat een applicatie voor Java ME op vele verschillende telefoons (en andere apparaten) kan worden uitgevoerd zonder dat hiervoor aanpassingen moeten worden gemaakt (in theorie..).

Wel zijn er diverse ontwikkelingen gaande waardoor Java ME er veel concurrentie bij heeft gekregen. Het dichtst bij Java ME, in termen van werking en niet in marktaandeel, staat bijvoorbeeld Flash Lite. Ook hierbij is het de bedoeling dat een applicatie in Flash Lite op vele verschillende apparaten kan draaien zonder noodzakelijke aanpassingen door de ontwikkelaar.

Naast Flash Lite en Java ME zijn er ook mogelijkheden om voor een specifieke telefoon, en dan voornamelijk voor een specifiek besturingssysteem op die telefoon, te ontwikkelen. Hoewel dit al lange tijd mogelijk is geweest (denk aan Windows Mobile, of Symbian). Lijkt dit een nieuwe impuls te hebben gekregen met de Apple iPhone en natuurlijk het nieuwe Google Android.

Vooraf deze twee nieuwe spelers zijn voor een gemiddelde ontwikkelaar extra interessant. Beide partijen hebben namelijk dezelfde filosofie met betrekking tot applicaties op hun telefoon. Er wordt hier extra veel moeite gedaan om het voor zowel een gebruiker als een ontwikkelaar makkelijk te maken om applicaties te verkrijgen c.q. aan te bieden.

Dit wordt gedaan door o.a. een centraal platform aan te bieden waar applicaties vandaan kunnen worden gehaald (Iphone App store en Google's Android market). Ontwikkelaars kunnen hier hun applicatie aanbieden (gratis of betaald) en gebruikers kunnen deze hier kopen/downloaden.

Hierdoor is een enorme drempel weggenomen om enerzijds een applicatie te 'kopen' (d.m.v. het extra vertrouwen wat een dergelijk centraal punt uitstraalt) en anderzijds om een applicatie aan te bieden (geen ingewikkelde trajecten, licenties, marketing etc). Dit zorgt vervolgens voor een zelfversterkend effect waardoor (gehoopt wordt dat) er meer wordt gekocht en meer zal worden aangeboden.

Voor deze module is er gekozen om het Google Android platform verder te onderzoeken. Android is een opensource (Linux based) besturingssysteem voor mobiele apparaten. Het doel van Google is om het OS geschikt te maken voor zowel smartphone's als meer beperkte apparaten. Voor fabrikanten kan dit zeer interessant zijn omdat ze op deze manier niet zelf ofwel opnieuw het wiel hoeven uit te vinden, ofwel een enorm ontwikkelbudget hoeven aan te houden om de 'concurrentie' bij te blijven. Op elke telefoon in de toekomst zou bijvoorbeeld Android kunnen draaien. Waarbij bepaalde mogelijkheden aan of uit staan afhankelijk van of de telefoon het ondersteunt.

Het potentiële bereik van Android is dus bijzonder groot. De toekomst zal moeten uitwijzen of dit ook daadwerkelijk zijn doorgang vindt. Zo niet, dan zal de kennis van Java en het programmeren in Android zeker bijdragen bij de overstap naar het op dat moment relevante mobiele platform!

1.2 Inhoud

Het hoofddoel van de module is het kunnen ontwikkelen van een applicatie binnen het Google Android platform. Hierbij staan twee andere punten centraal, namelijk: Het binnen een kort tijdbestek een nieuw platform (en eventueel ook een programmeertaal) kunnen doorgronden, op basis van een verscheidenheid aan bronnen. En het op een object georiënteerde manier kunnen analyseren en realiseren van een applicatie.

De ontwikkeling van applicaties voor Android gebeurt met de Object Georiënteerde programmeertaal Java. Hierbij biedt Android een groot deel van de standaard Java API (denk aan XML verwerking, Netwerkcommunicatie). En een eigen set aan API's om specifiek tegen Android (en de telefoon) te kunnen praten. Hoewel de structuur van een Android applicatie in vele opzichten lijkt op de structuur van een Java ME applicatie zijn zowel de code als de API's niet gelijk. Een Java ME applicatie zal dus niet draaien onder Android. Maar met Java ME kennis is Android sneller te begrijpen (en andersom is de overstap naar Java ME ook weer makkelijker met Android kennis).

Verder heeft de taal Java vele overeenkomsten met o.a. `c#` (Microsoft .NET platform) en Actionscript 3.0 (Flash platform) waardoor het in de toekomst gemakkelijker wordt om eventueel te schakelen naar een ander platform indien de verhoudingen in de markt dusdanig verschuiven (Apple's iPhone gebruikt helaas een taal (Objective-C) die minder lijkt op Java, dus die overstap is iets lastiger).

Uiteraard doe je ook (basis) programmeerkennis op die altijd van pas zal blijven komen, en is het goed en intuïtief object georiënteerd kunnen denken van onschatbare waarde gedurende je verdere (programmeer) carrière. Naast de focus op de technische realisatie Zullen ook diverse usability aspecten aan bod komen, zoals het werken met een beperkte interface en/of beperkt schermoppervlakte en beperkte rekenkracht.

Als eindproduct dient een (eigen) concept voor het Google Android platform in groepsverband (2 personen) te worden gerealiseerd.

De module biedt hiertoe houvast op de volgende manieren:

1. Algemene introductie Google Android & Java programmeren,
2. Inbreng van een aantal globale richtingen binnen Android, waaruit een concept voor een applicatie kan worden afgeleid,
3. Workshops (door docent) voor specifieke onderwerpen (naar behoefte van studenten),
4. Wekelijkse workshops door studenten op het gebied waar zij in bezig zijn (beginnend vanaf week 3 of 4),
5. Verplichte diversiteit aan projecten waar studenten aan werken (waardoor er een breder assortiment aan nieuwe kennis wordt ingebracht in de workshops),
6. Ruimte voor kennisoverdracht op basis van ingebrachte problemen.

Tijdens het ontwikkelen van de applicatie zal je door middel van een workshop medestudenten uitleg geven over het specifieke Android deel waar jij je op focust. Andersom krijg je door diverse workshops van deze medestudenten weer extra inzicht in delen van Android die je mogelijk weer kan gebruiken in het eigen product.

1.3 Relatie met andere onderwijseenheden

Deze module gaat in op technische realisatie en biedt (basis)kennis die in de verdere minor web/mobile noodzakelijk is.

Leerdoelen:

De student kent:

- Basisbegrippen, termen en het principe van ontwikkelen van applicaties voor Google Android in Java
- De mogelijkheden en beperkingen van het Android platform
- Basisbegrippen en termen binnen object georiënteerd denken
- Methoden om een applicatie / game te modeleren (o.a. UML)

De student kan:

- Een gestructureerde werkomgeving opzetten voor Java / Android
- Klassen opstellen in Java en omgaan met objecten
- Gebruik maken van API documentatie
- Binnen Android applicaties / games ontwikkelen
- Een concept uitwerken tot logische ontwerpen/documentatie en realiseren in Java voor Google Android

1.4 Werkvorm(en)

De eerste drie á vier lessen zullen grotendeels een klassieke invulling kennen. De nadruk ligt hierbij op het overbrengen van nieuwe kennis (of het herhalen hiervan) om een begin te kunnen maken met een Android applicatie. Van jou als student wordt verwacht dat je actief mee doet tijdens deze lessen (geen hoorcolleges). De theorie wordt daarom primair verpakt in praktische opdrachten die individueel, in groepsverband, of klassikaal worden gemaakt/bediscussieerd.

Elk van deze eerste lessen zal in grote mate bijdragen aan de uiteindelijke op te leveren applicatie.

Vanaf les vier wordt verwacht dat elke week twee groepen studenten een korte workshop houden van maximaal 1 (klok)uur waarin een specifiek Android onderdeel wordt uitgelegd aan de overige studenten. Deze materialen kunnen vervolgens, indien relevant, gebruikt worden in hun eigen applicatie. Iedere student dient minimaal 1x een workshop te geven.

De workshops door studenten kunnen, indien gewenst, worden afgewisseld met workshops door de docent over specifieke onderdelen (op basis van vragen uit de groep). Hierbij kan worden gedacht aan een diepere uitleg over Object Georiënteerd programmeren, hoe om te gaan met Threads, etc.

1.5 Keuzeruimte

De module richt zich op het Android platform. Daarbinnen is het soort applicatie in principe vrij. Wel wordt er vooraf afgesproken wie wat gaat doen om voor voldoende diversiteit te zorgen. Tevens zal er een lijst (zie verderop) zijn met Android onderzoeksgebieden waaruit de keuze voor een applicatie (concept) kan worden gemaakt. Het concept is hierbij vrij en kan bijvoorbeeld variëren van een simpele Gadget, een database gestuurde applicatie, tot een client-server Game (MMO?). Wel wordt het concept beoordeeld op haalbaarheid (mede op basis van kennis niveau studenten). En wordt er van iedere groep verwacht dat ze werken met deelproducten en tussentijdse versies.

1.6 Toetsing

De toetsing wordt gedaan op basis van het opgeleverde eindproduct (zowel werkende applicatie, de bijbehorende code, ontwerpdocumentatie, en verantwoording van wie wat heeft gedaan). En de gegeven workshop. Daarnaast is er een deel van het cijfer wat wordt bepaald door de betrokkenheid bij de lessen en overige workshops.

De opgeleverde applicatie wordt in de laatste week gepresenteerd aan de groep en docent (max 15 minuten) waarbij alle groepsleden een deel voor hun rekening nemen.

2 Programma

2.1 welke onderwerpen komen aanbod?

Let op dat de onderwerpen verspreid kunnen zijn over meerdere lessen, of dat onderwerpen op basis van vragen vanuit de groep pas worden geïntroduceerd. Alleen onderwerp één en twee zullen minimaal aanbod komen (klassikaal in lesvorm).

2.1.1 Onderwerp 1: Programmeren /Java algemeen

- Java opbouw: Virtual machine, API
- Programmeer/java basisbegrippen:
 - o Variabelen
 - o Methoden/Functies
 - o Control statements (conditioneel, lus, vertakking)
 - o Operatoren
 - o Structuur (class, package, access modifiers)
 - o Arrays & Collections

2.1.2 Onderwerp 2: Google Android

- Wat is Google Android?
- Ontwikkelomgeving (emulator) voor Android
 - o Noodzakelijke Security settings (bijv. internet toegang)
 - o Uitbreidingen op emulator (bijv. voor motion detection)
- Opbouw van een applicatie in Google Android
 - o Activities
 - o Views
 - o Intents
 - o Services
 - o Content Provider
- Simpele applicaties in Android
- Simpele 2d games/gadgets in Android

2.1.4 Onderwerp 3: Object georiënteerd denken, ontwerpen en programmeren

- Wat is Object Oriëntatie?
- Objecten, relaties, members
- Interfaces
- Inheritance
- Java & OO

2.1.3 Onderwerp 4: UML

- Wat is UML?
- Class diagram
- Activity diagram

2.1.5 Onderwerp 5: Threading

- Wat zijn Threads?
- Noodzaak & gebruik
- Threads (standaard Java) en Handler's (Android)

2.2 Onderzoeksgebieden / Workshop onderwerpen

De in deze paragraaf genoemde onderwerpen zijn bedoeld als voorzet voor de te ontwikkelen concepten. Het is de bedoeling dat elke groep een ander (deel)gebied onderzoekt. De uiteindelijke applicatie mag elke vorm hebben. Bijvoorbeeld een spel, een gadget of een 'bedrijfsapplicatie'.

De inhoud van de workshop dient te gaan over een klein en specifiek onderwerp, en zal verder moeten gaan dan standaard functionaliteit die mogelijk iedereen zal onderzoeken. Onderstaande onderwerpen kunnen tevens dienen als workshop onderwerp.

1. Device specifieke mogelijkheden zoals: **Accelerometer, kompas en GPS**
 - a. Aanhaken op accelerometer functionaliteit
 - b. Locatieinformatie met kompas, GPS en GoogleMaps combineren met eigen concept
2. **Multimedia** mogelijkheden (**streaming media, media player, MMS**)
3. **Camera integratie**
 - a. Beeldherkenning (barcodes?)
 - b. Afbeeldingen/filmpjes uploaden naar community site
4. **Client – server** communicatie via HTTP
 - a. PHP/ASP.NET backend met applicatie laten communiceren voor bijvoorbeeld Info opslaan/ophalen op server ipv lokaal
5. **Client – server – Client** communicatie via HTTP of directe netwerk communicatie.
 - a. Via een server (eventueel webserver) communicatie tussen verschillende devices bewerkstelligen.
6. ~~Bluetooth device – device communicatie~~
 - a. **Bluetooth support zit nog niet in SDK versie 1.0**
7. **OpenGL ES (3d)**
 - a. Game of applicatie met 3d visualisatie
8. **SQLite**
 - a. Applicatie met database
9. **Combinatie** van bovenstaande onderdelen (evt met meerdere groepen)

2.3 Te bestuderen literatuur en andere bronnen

De meest uitgebreide bron voor Java en Android is het internet, en specifiek de site van de makers van Java, Sun Microsystems, en uiteraard Google Android. Voor beide onderwerpen zijn uitgebreide fora te vinden waar de meeste vragen wel een keer gesteld zijn (zoek ze dus eerst door voordat je zelf hier iets gaan plaatsen!).

Hieronder volgt een lijst met url's naar specifieke onderdelen of tutorials. Voor het gebruik van OO en UML kan je daarnaast het boek Praktisch UML gebruiken wat op je boekenlijst vermeld staat.

Wellicht ten overvloede: Er wordt verwacht dat je zelfstandig voorbeelden en uitleg kan verwerken tot hetgeen je nodig hebt. Gebruik deze links dan ook meer in het kader van "Hoe lees ik een Jpg in en plaats ik deze in mijn app" dan dat je zoekt naar "Hoe maak ik <insert jouw concept> voor Android".

Boek:

Praktisch UML 3^e editie – Warmer en Kleppe - 9043008125

Internet:

- Syntax van Java en algemene zaken zoals if/else while etc
<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/index.html>
- Basisprincipes van Object Georiënteerd denken/werken.
<http://java.sun.com/docs/books/tutorial/java/concepts/index.html>
- Basis implementatie van de objectengedachte in Java, code voorbeelden, etc.
<http://java.sun.com/docs/books/tutorial/java/javaOO/index.html>
- Startpunt Android tutorials en meer:
<http://code.google.com/intl/nl/android/documentation.html>
- Eclipse (ontwikkelomgeving, kies Java Developers, 2e item)
<http://www.eclipse.org/downloads/>
- Google Books voorbeeld: The busy coder's guide to Android Development
<http://books.google.nl/books?id=Irf4xxsgmd8C>
- Emulator uitbreiding voor beweging:
<http://code.google.com/p/openintents/wiki/SensorSimulator>

2.4 Weekschema

Week	Primaire theorie	Praktijk & Producten
1 (7)	<p>Introductie in de les en het einddoel (aantal voorbeelden van Android).</p> <ul style="list-style-type: none"> - Opzetten ontwikkelomgeving - Korte intro in opbouw android - Gezamenlijk bouwen van een aantal simpele apps. 	<p>Diverse kleine opdrachten ter ondersteuning van de theorie.</p> <p>Huiswerk:</p> <ul style="list-style-type: none"> - opdrachten van les verder uitwerken - lezen literatuur over Android (zie presentatie) - Samenstellen groepen - Bedenken en uitwerken concept voor eindopdracht
2 (8)	<ul style="list-style-type: none"> - Verder ingaan op android principes vanuit huiswerk ed. - Voorbeeld apps: app met verschillende schermen, menu en formulier elementen (buttons textfields etc). - Bespreken opdrachten (code) - Bespreken van concepten 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - Bepalen welke kennis in welke volgorde moet worden opgedaan - Taakverdeling en planning opstellen voor tussentijdse versies - Inleveren definitief concept en planning via mail - Beginnen met prototypes in Android (eerste versie)
-	<i>Voorjaarsvakantie</i>	
3 (10)	<ul style="list-style-type: none"> - 2D android principes (game?) - interactie handling - Drawing - Resources (afbeeldingen) inladen 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - workshop voorbereiden / - uitwerken eindcase versie 1 - wensenlijst opstellen voor extra theorie en mailen
4 (11)	<ul style="list-style-type: none"> - Workshop 2 groepen - Behoeftte?: Workshop docent 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - workshop voorbereiden / - uitwerken eindcase versie 1 - wensenlijst opstellen voor extra theorie en mailen
5 (12)	<ul style="list-style-type: none"> - Workshop 2 groepen - Tonen tussentijdse voortgang (versie 1?) - Behoeftte?: Workshop docent 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - workshop voorbereiden / - uitwerken eindcase versie 2 - wensenlijst opstellen voor extra theorie en mailen
6 (13)	<ul style="list-style-type: none"> - Workshop 2 groepen - Behoeftte?: Workshop docent 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - workshop voorbereiden / - uitwerken eindcase versie 2 - wensenlijst opstellen voor extra theorie en mailen
7 (14)	<ul style="list-style-type: none"> - Workshop 2 groepen - Tonen tussentijdse voortgang (versie 2?) - Behoeftte?: Workshop docent 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - workshop voorbereiden / - uitwerken eindcase versie 3 - wensenlijst opstellen voor extra theorie en mailen
8 (15)	<ul style="list-style-type: none"> - Workshop 2 groepen - Behoeftte?: Workshop docent 	<p>Huiswerk:</p> <ul style="list-style-type: none"> - Uitwerken eindcase
9 (16)	Pasen etc. GEEN LES	<p>Huiswerk:</p> <ul style="list-style-type: none"> - Uitwerken eindcase
10 (17)	Presentatie eindproduct, afsluitend de oplevering van applicatie en documentatie.	

Over het huiswerk

Belangrijke data

- **Week 2 (8):** maandagochtend (**16-2**) eerste concept dient af te zijn voor aanvang van les. En einde van les is indeling workshops vastgesteld.
- **Week 5 (12):** maandagochtend (**16-3**) 9 uur: versie 1 van applicatie dient af te zijn voor aanvang van les.
- **Week 7 (14):** maandagochtend (**31-3**) 9 uur: versie 1 van applicatie dient af te zijn voor aanvang van les.
- **Week 10 (17):** maandagochtend (**20-4**) 9 uur: Applicatie dient af te zijn en presentatie van max 15 minuten dient voorbereid te zijn (en te worden gegeven).

3 Toetsing en beoordeling

3.1 Procedure

Om voor een beoordeling van de module in aanmerking te komen moet je aan een aantal criteria hebben voldaan. Hieronder wordt beschreven wat deze criteria zijn:

- *Inleveren producten: Definitieve concept, eindapplicatie, ontwerpdocumentatie*
Er geldt een verplichting aan op tijd inleveren van materialen. In het weekschema staat beschreven wat deze data zijn. Hierop kan in de regel geen uitzondering gemaakt worden.
- *Aanwezigheid,*
Om voor een beoordeling in aanmerking te komen moet je minimaal bij 90% van de lessen aanwezig zijn geweest. Uitzonderingen hierop worden alleen gemaakt als dit vooraf aan de lessen is gemeld, en uiteraard met een geldige reden!
- *Workshop:*
Iedere student is verplicht om een workshop te geven (samen met zijn of haar groep).

3.2 Beoordelingscriteria

Voor de beoordeling van de diverse producten gelden o.a. de volgende criteria:

Concept:

Het concept is je eerste opzet tot het uitwerken van je ontwerpdocumentatie en dus uiteindelijk voor het bouwen van het applicatie. Hierdoor is het alleen noemen van een bestaand of zeer abstract concept zeker niet voldoende. Je zult het concept moeten ontleden in diverse aspecten zoals welke interactie vind er plaats? Hoe gebruik ik de invoer en het scherm? Welke technieken komen er bij kijken? Etc etc.

Workshop:

In de workshop probeer je in maximaal een uur (klokuur) een klein onderwerp binnen Android over te dragen aan de overige studenten. Hoewel het onderwerp misschien omvangrijk kan zijn is het jouw/jullie taak om met simpele voorbeelden (zonder overbodige of onduidelijke code) het geheel uit te leggen.

Een goede workshop bevat in ieder geval:

- Introductie in het onderwerp (wat maakt het spannend/ moeilijk/ interessant?)
- Abstract overzicht van hoe het werkt
- Meest simpele code die werkt (en uitleggen hoe deze werkt)
- Opdracht(en) om het zelf te kunnen (hiermee zorg je ervoor dat de kennis blijft hangen, en iedereen een voorbeeld implementatie heeft indien ze het later opnieuw willen inzetten).

Let op, de workshops die in week 4 worden gegeven, zullen anders worden beoordeeld dan de workshops die aan het einde worden gegeven. Immers aan het begin is er minder voorbereidingstijd en zijn er nog geen voorbeelden van medestudenten om inspiratie uit op te doen.

Ontwerpdocumentatie:

De documentatie zal om te beginnen de eisen en interactie beschrijven. Deze moeten vervolgens zijn vertaald naar specificaties voor je applicatie. Het is aan te raden om in dit stadium ook Use-cases te gebruiken.

Middels bijvoorbeeld activity diagrammen of eventueel state-transition diagrammen en klassediagrammen geef je verder vorm aan de technische werking van je systeem.

Let op dat de documentatie relevant blijft. Uiteindelijk blijft het documenteren, en de UML diagrammen een hulpmiddel om dat wat onduidelijk is (of kan zijn voor anderen) inzichtelijk te maken. Het is dus niet zo dat

er minimaal xx diagrammen van type I aanwezig moeten zijn en xx diagrammen van type II. Uiteindelijk maak jij als ontwikkelaar de keuze over wat wel en wat niet moet worden uitgewerkt.

Het product

Het project moet in compileerbare versie worden opgeleverd (bij voorkeur een Eclipse project). Indien het geheel niet compileert, wordt het als 'niet ingeleverd' beschouwd. Daarnaast dient de code een goede reflectie te zijn van je uiteindelijke ontwerp (dus klassen en architectuur komen overeen). Uiteraard heb je nette en goed leesbare code geschreven die waar nodig is voorzien van commentaar. Je hebt je (grotendeels) gehouden aan Java programmeerstandaarden en geen onlogische of niet object georiënteerde code geschreven (of indien wel heb je gedocumenteerd waarom).

Als je al code hebt 'geleend' van anderen heb je deze goed geïntegreerd door naamgeving te kiezen die in je applicatie kloppen en geen ongebruikte variabelen over te laten.

Je hebt ook zoveel mogelijk logic die meerdere keren plaats vond verplaatst naar een centrale method of class om de onderhoudbaarheid te vergroten.

3.3 Weging

De weging tussen de op te leveren materialen is als volgt:

- 40% Workshop
- 50% Eindproduct en documentatie
- 10% Houding in klas en tijdens workshops

Let op: Indien geen Workshop gegeven, of geen eindproduct opgeleverd wordt het gehele vak als onvoldoende beschouwd.

Alle producten worden gezien als het resultaat van een groepsinspanning. Zodoende zullen alle groepsleden hetzelfde cijfer krijgen. Indien de werkzaamheden onevenredig zijn verdeeld dient dit ruim op tijd te worden aangegeven. Nadat een beoordeling is vastgesteld kan er geen rekening meer worden gehouden met groepsleden die mogelijk minder hun best hebben gedaan.

3.3 Herkansingen

De **herkansingen** zijn als volgt geregeld:

- bij een onvoldoende resultaat aan het eind van de lesperiode, nadat alle deelcijfers zijn verwerkt tot een eindcijfer, volgt een herkansing in de direct daar opvolgende lesperiode. Dit houdt in dat, nadat het eindcijfer bekend is, de student individuele afspraken maakt met de begeleidende docenten over het te volgen herkansingstraject.
- **Indien 1 van de deelproducten (workshop of eindproduct/documentatie) niet is uitgevoerd/opgeleverd kan er geen herkansing plaatsvinden. Lever dus altijd iets in!**
- Voor herkansing van de Workshop geldt dat dit alleen tijdens de lesperiode kan worden gedaan (of tijdens een les in een volgend schooljaar). Dit is de verantwoordelijkheid van de student.
- bij onvoldoend resultaat van het eindproduct worden er individuele afspraken gemaakt met de studenten in de groep over zogenaamde reparatiewerkzaamheden, om dit resultaat om te vormen tot een voldoende resultaat. Dit gebeurt direct nadat beoordeling heeft plaatsgevonden. De reparatie werkzaamheden dienen uiterlijk 5 weken na het vastleggen van de reparatieafspraken te zijn ingeleverd. Let op dat er extra functionaliteit kan worden verwacht als onderdeel van de reparatieslag.

Indien het totale eindresultaat weer onvoldoende mocht zijn, volgt het cursusjaar daarop een nieuwe kans. Maak je geen gebruik van de herkansingsmogelijkheid, dan vervalt het recht op herkansing. In uitzonderlijke gevallen kan hiervan worden afgeweken, echter uitsluitend in overleg met SLB-coach en begeleidende docenten.