

FitnessManager

Sjors Clabbers Matjah Sonneveld Gerco van Heerdt

18 mei 2012

Inhoudsopgave

1	Inleiding	2
2	Requirements	3
2.1	Functional requirements	3
2.2	Non-functional requirements	3
2.3	Use case survey	4
2.4	Use case diagram	5
2.5	Use cases	6
2.5.1	Apparaten invoeren	6
2.5.2	Doelen stellen	8
2.5.3	Trainingssessie plannen	9
2.5.4	Trainingssessie afspelen	11
2.5.5	Vooruitgang opvragen	12
3	Ontwerp	13
3.1	Globaal Ontwerp	13
3.1.1	ApparaatBeheerder	13
3.1.2	SessiePlanner	13
3.1.3	SessieBegeleider	13
3.1.4	VooruitgangsBeheerder	13
3.2	Gegevensontwerp	14
3.3	Detail-ontwerp per component	14
3.4	Gebruikersinterface	16
3.4.1	MainView/VooruitgangView	16
3.4.2	ApparatenView	16
3.4.3	SessieView	16
3.4.4	KrachtApparaatToevoegen	17
3.4.5	SessieBegeleider	18
4	Planning	20

1 Inleiding

Ons doel is het maken van een app die gebruikers helpt in efficiënter en effectiever fitnessen door op een handige en vlotte manier bij te houden wat de gebruiker in vorige sessie's gedaan heeft. Deze app is dus bedoeld voor mensen die regelmatig fitnessen maar moeite hebben met het onthouden van de volgorde van apparaten en wat ze de vorige keer voor instellingen gebruikt hebben, bijvoorbeeld het gewicht. Als extra functionaliteit zal deze app de gebruiker helpen met het behalen van gezette doelen en het zo optimaal mogelijk plannen van fitness sessie's.

2 Requirements

2.1 Functional requirements

De functional requirements zijn

- Fitnessapparaten invoeren
- Fitnessapparaten tot een lijst samenvoegen
- Persoonlijke doelen stellen
- Trainingssessies plannen
- Trainingssessies afspelen
- Resultaten overzien

2.2 Non-functional requirements

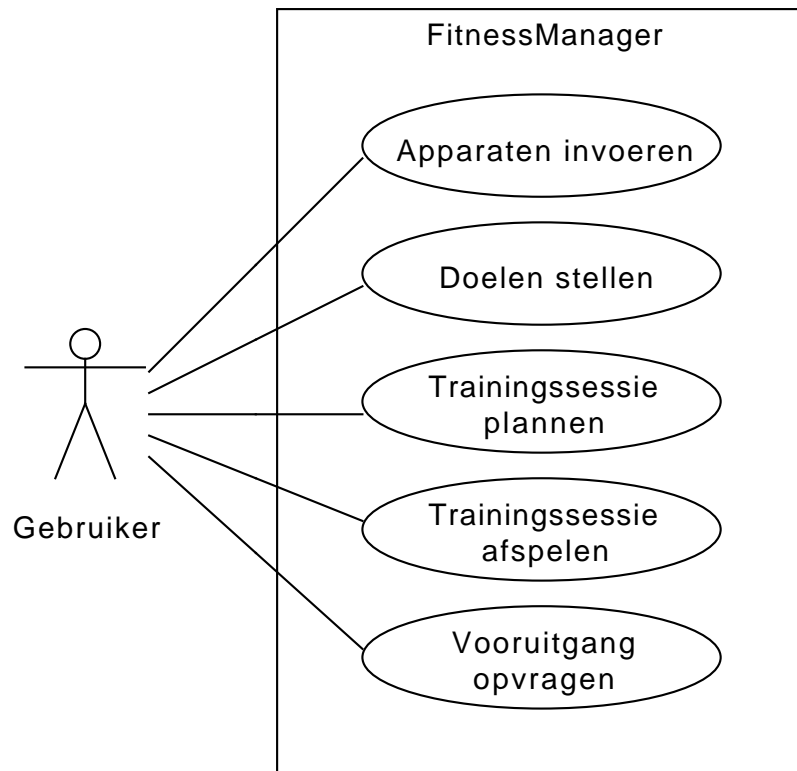
Onze app moet gaan voldoen aan de volgende non-functional requirements.

- Configurability
- Data integrity
- Maintainability
- Scalability
- Usability, achievability

2.3 Use case survey

Use case	Usecase beschrijving	Actoren
Apparaten invoeren	Deze use case stelt de gebruiker in staat om apparaten aan de app toe te voegen.	Gebruikers
Doelen stellen	Deze use case stelt de gebruiker in staat doelen te stellen en het systeem genereert een programma daar naartoe.	Gebruikers
Trainingsessie plannen	Deze use case stelt de gebruiker in staat een trainingssessie op te geven.	Gebruikers
Trainingsessie afspelen	Deze use case stelt de gebruiker in staat feedback te krijgen op zijn/haar acties tijdens een sessie.	Gebruikers
Vooruitgang opvragen	Deze use case stelt de gebruiker in staat zijn/haar vooruitgang te zien door middel van diagrammen.	Gebruikers

2.4 Use case diagram



Figuur 1: Use case diagram

2.5 Use cases

2.5.1 Apparaten invoeren

Use Case Name	Apparaten invoeren
Description	Deze use case stelt de gebruiker in staat om apparaten aan de app toe te voegen.
Actors	Gebruiker
Trigger	De gebruiker drukt op de knop 'Voeg apparaat toe'.
Pre conditions	<ul style="list-style-type: none"> • Er is een lijst aanwezig, die mag deels gevuld zijn. • Het apparaat is nog niet aan de lijst toegevoegd.
BCoE	<ol style="list-style-type: none"> 1. Systeem: Vraagt om een lijst aan te maken/veranderen 2. Gebruiker: Kiest lijst veranderen 3. Systeem: Vraagt om handmatig of automatisch invoegen 4. Gebruiker: kiest handmatig invoegen 5. Systeem: Vraagt om naam apparaat 6. Gebruiker: Voegt naam in en bevestigt 7. Systeem: vraagt om type apparaat cardio, krachttraining 8. Gebruiker: Kiest cardio 9. Systeem: vraagt om tijdsduur en snelheid 10. Gebruiker: kiest tijdsduur en snelheid 11. Systeem: Laat resultaten zien en vraagt om bevestiging 12. Gebruiker: Bevestigt

Alternate Paths	<p>Eerste Alternate path</p> <ol style="list-style-type: none"> 4. Gebruiker: Kiest automatisch invoeren 5. Gebruiker: Scant apperaat in 6. Systeem: Laat apparaat zien en vraagt om Gegevens van het apparaat 7. Gebruiker: Geeft gegevens en bevestiging <p>Tweede alternate path</p> <ol style="list-style-type: none"> 8. Gebruiker: Kiest krachttraining 9. Systeem: Vraagt om Gewicht en herhaling 10. Gebruiker: Voert gegevens in 11. Systeem: Vraagt om bevestiging 12. Gebruiker: Bevestigt <p>Derde alternate path</p> <ol style="list-style-type: none"> 2. Gebruiker: Kiest lijst maken 3. Systeem: Vraagt naam lijst 4. Gebruiker: Geeft naam lijst 5. Systeem: Vraagt om bevestiging 6. Gebruiker: Geeft bevestiging 7. Systeem: Gaat naar menu
Post conditions	Apparaat is toegevoegd aan lijst

2.5.2 Doelen stellen

Use Case Name	Doelen stellen
Description	Deze use case stelt de gebruiker in staat om persoonlijke doelen in te stellen.
Actors	Gebruiker
Trigger	De gebruiker drukt op de knop 'Doelen instellen'.
Pre conditions	<ul style="list-style-type: none"> • Het doel wat je wilt toevoegen is nog niet toegevoegd.
BCoE	<ol style="list-style-type: none"> 1. Systeem: Vraagt om doelnaam 2. Gebruiker: Voegt naam in en bevestigt 3. Systeem: Vraagt om apparaat waarop je je doel wilt instellen 4. Gebruiker: Kiest apparaat 5. Systeem: Vraagt gegevens die je wilt bereiken 6. Gebruiker: voert gegevens in 7. Systeem: Vraagt om periode waarin je ze wilt bereiken 8. Gebruiker: Voegt gegevens in en bevestigt 9. Systeem: Laat totaal plaatje zien
Alternate Paths	-
Post conditions	Doel is toegevoegd

2.5.3 Trainings sessie plannen

Use Case Name	Trainings sessie plannen
Description	Deze use case stelt de gebruiker in staat een trainings sessie op te geven.
Actors	Gebruiker
Trigger	De gebruiker drukt op de knop 'Trainings sessie plannen'.
Pre conditions	<ul style="list-style-type: none"> • De trainings sessie die je wilt toevoegen is nog niet toegevoegd. • De trainings sessie die je wilt aanpassen bestaat.
BCoE	<ol style="list-style-type: none"> 1. Systeem: Vraagt, of je een trainings sessie wilt aanpassen toevoegen 2. Gebruiker: Kiest toevoegen trainings sessie 3. Systeem: Laat lijst met ingevoerde apparaten zien 4. Gebruiker: Kiest apparaten die hij/zij tot lijst wilt toevoegen 5. Systeem: Laat lijst zien en vraagt om bevestiging 6. Gebruiker: Accepteert 7. Systeem: Slaat lijst op
Alternate Paths	<p>Eerste Alternate Path</p> <ol style="list-style-type: none"> 2. Gebruiker: Kiest trainings sessie aanpassen 3. Systeem: geeft lijst met huidige trainings sessies 4. Gebruiker: Kiest lijst die hij wilt aanpassen 5. Systeem: Geeft aanpasbare lijst 6. Gebruiker: Verandert lijst en slaat op <p>Tweede Alternate Path</p> <ol style="list-style-type: none"> 6. Gebruiker: Weigert bevestiging 7. Systeem: Slaat niet op en sluit venster af

Post conditions	<ul style="list-style-type: none">• Trainingssessie is aangepast
------------------------	--------------------------------------------------------------------------------

2.5.4 Trainingssessie afspelen

Use Case Name	Trainingssessie afspelen
Description	Deze use case stelt de gebruiker in staat zijn resultaten in te voeren in de app.
Actors	Gebruiker
Trigger	De gebruiker drukt op de knop 'Start sessie'.
Pre conditions	<ul style="list-style-type: none"> • Je bent nog geen sessie gestart.
BCoE	<ol style="list-style-type: none"> 1. Systeem: Laat apparaat zien 2. Gebruiker: Kiest apparaat 3. Systeem: Vraagt om resultaten zoals te zien bij apparaat invoeren 4. Gebruiker: Vult resultaten in 5. Systeem: Vraagt om bevestiging 6. Gebruiker: Geeft bevestiging 7. Systeem: Vraagt om volgende apparaat 8. Gebruiker: Kiest ja 9. Herstart use case
Alternate Paths	<ol style="list-style-type: none"> 7. Gebruiker: Kiest nee 8. Systeem: Sluit huidige venster af
Post conditions	<ul style="list-style-type: none"> • De resultaten van de oefening zijn ingevoerd.

2.5.5 Vooruitgang opvragen

Use Case Name	Vooruitgang opvragen
Description	Deze use case stelt de gebruiker in staat zijn vooruitgang te zien.
Actors	Gebruiker
Trigger	De gebruiker drukt op de knop 'Vooruitgang.'
Pre conditions	-
BCoE	<ol style="list-style-type: none">1. Systeem: Vraagt om apparaat waar je je resultaten van wilt zien2. Gebruiker: Kiest apparaat3. Systeem: Laat resultaten zien
Alternate Paths	-
Post conditions	<ul style="list-style-type: none">• Je bent op de hoogte van je resultaten

3 Ontwerp

3.1 Globaal Ontwerp

Onze app is onder te verdelen in de volgende componenten:

3.1.1 ApparaatBeheerder

Laat een gebruiker lijsten van apparaten instellen. Per apparaat kan ook een doel ingesteld worden.

3.1.2 SessiePlanner

Laat een gebruiker sessies instellen en kan deze optimaal ordenen. De SessiePlanner krijgt apparaten via de ApparaatBeheerder.

3.1.3 SessieBegeleider

Begeleidt de gebruiker tijdens een sessie en laat hem of haar resultaten aangeven. De SessieBegeleider krijgt een sessie via de SessiePlanner en apparaten van de ApparaatBeheerder.

3.1.4 VooruitgangsBeheerder

Toont de vooruitgang per apparaat. De VooruitgangsBeheerder slaat gegevens op die hij krijgt van de SessieBegeleider om die later te kunnen tonen en verwerken. Verder krijgt hij apparaten van de ApparatenBeheerder.

3.2 Gegevensontwerp

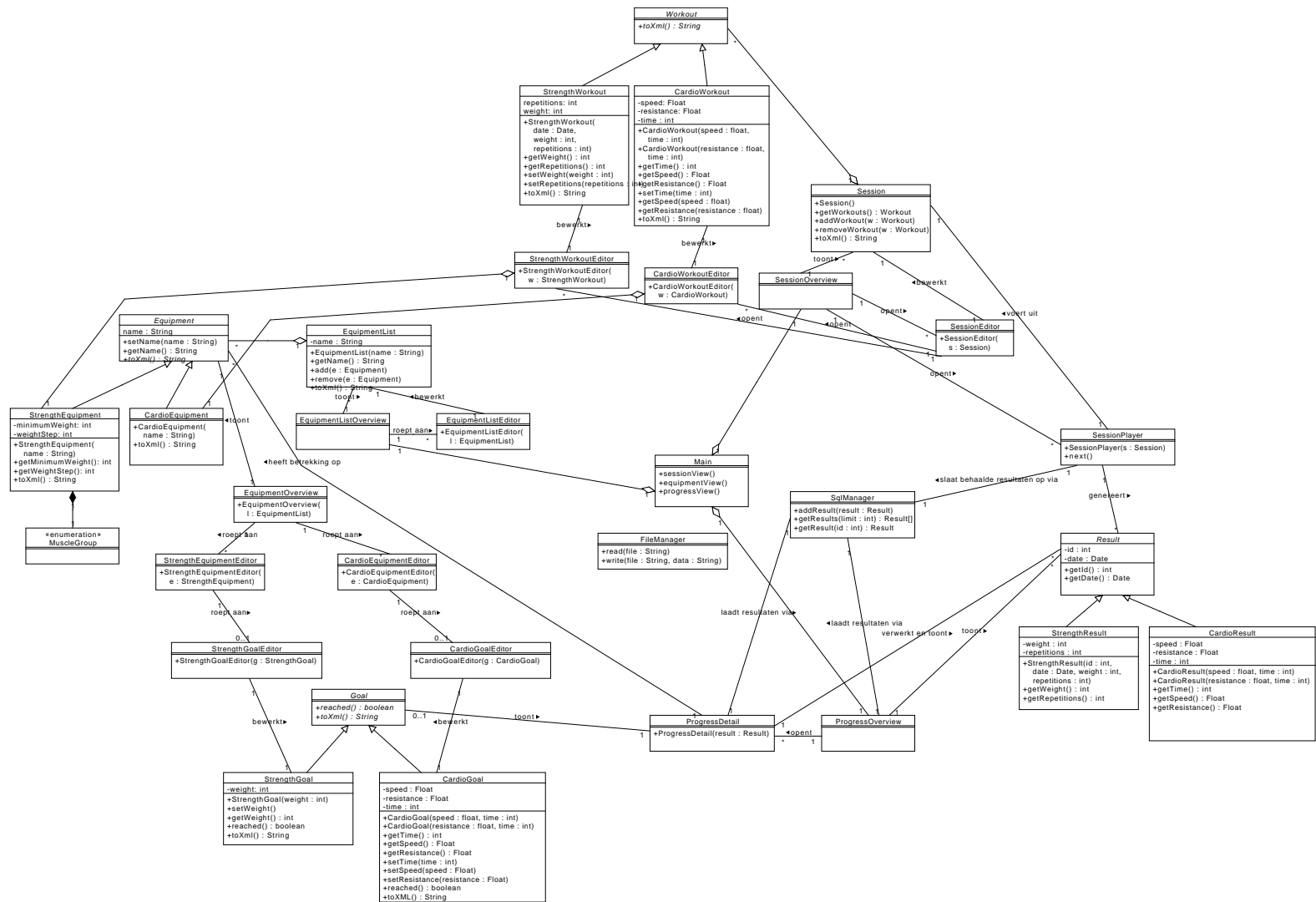
We slaan lijsten van apparaten, apparaten zelf, doelen en sessies op in XML-bestanden. Op deze manier kunnen deze gegevens makkelijker overgezet en dus ook gedownload worden. We zouden de app dan uit kunnen breiden met ondersteuning voor het inlezen van QR-codes die verwijzen naar webpagina's die een lijst van apparaten in XML-formaat aanbieden, zodat een sportschool zo'n code op kan hangen en gebruikers niet zelf alle apparaten in hoeven te stellen. Voor het lezen van de XML-bestanden willen we een hoofdklasse aanmaken die dit afhandelt en de resultaten cachet, om te voorkomen dat meerdere componenten bijvoorbeeld dezelfde apparaatbestanden steeds weer helemaal in moeten lezen.

Voor behaalde resultaten gebruiken we SQL zodat de app niet te traag wordt als er heel veel resultaten aanwezig zijn. Ook hiervoor kunnen we weer een klasse maken die de communicatie met de SQL-server afhandelt.

3.3 Detail-ontwerp per component

Naast de aangegeven componenten hebben we nog een aantal globale klassen, waarvan een deel misschien 'statisch' wordt, nodig die we steeds verder uitbreiden tijdens het ontwikkelen van de componenten. De relaties die klassen met de FileManager hebben hebben we niet uitgewerkt om het overzicht nog enigszins te behouden. Het idee is dat de klassen StrengthEquipment, CardioEquipment, StrengthGoal, CardioGoal en Session zichzelf via deze klasse op kunnen slaan en uit kunnen lezen. De klasse Main wordt een TabActivity en veel klassen breiden Activity uit, maar dit hebben we niet aangegeven omdat het uit hun naam duidelijk zou moeten worden.

Uiteindelijk willen we de applicatie in het Engels gaan maken. In de rest van dit document komt dat niet naar voren, maar we hebben ervoor gekozen om dit al wel bij de invulling van de klassen aan te houden.



3.4 Gebruikersinterface

3.4.1 MainView/VooruitgangView



Figuur 2: VooruitgangView

Dit is het scherm wat de gebruiker te zien krijgt als de app wordt opgestart. Aan de boven kant zijn 3 tabs te zien met alle “hoofd” schermen zodat het makkelijk navigeren is tussen de administratieve features.

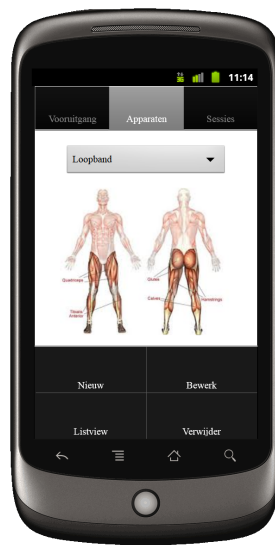
de VooruitgangView heeft een dropdown menu om een apparaat te selecteren en daaronder een lijst met wat er wanneer met het gekozen apparaat gedaan is. In het hier niet afgebeelde menu zal een optie zijn om deze geschiedenis te verwijderen en, als we tijd over hebben, een optie om de geschiedenis in een grafiek te tonen.

3.4.2 ApparatenView

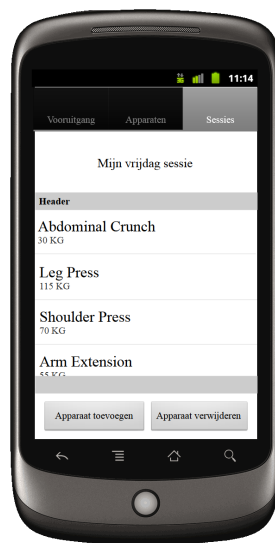
Dit scherm heeft weer een dropdown menu om een apparaat te kiezen. Van het geselecteerde apparaat zal alle info te zien zijn die de gebruiker ooit heeft ingevoerd bij het aanmaken van het apparaat. Dit zijn dus de spiergroepen, naam, etc. Als de gebruiker op de menu knopt drukt (hier wel afgebeeld) zal er een menu verschijnen met de opties om een nieuw apparaat toe te voegen, het huidige apparaat te bewerken, een andere lijst met apparaten te selecteren en om het huidige apparaat te verwijderen. Men kan kiezen uit een kracht apparaat (zie bijvoorbeeld KrachtApparaatToevoegen) en een cardio apparaat.

3.4.3 SessieView

In dit scherm kan de gebruiker sessies aanmaken. Bij het starten van het scherm zal de laatst gebruikte sessie getoond worden. Bovenaan zal een label met de naam van de sessie te zien zijn met daaronder een lijst van alle apparaten in die sessie met het gewicht dat de gebruiker moet verplaatsen. Ook zijn er hier



Figuur 3: ApparatView

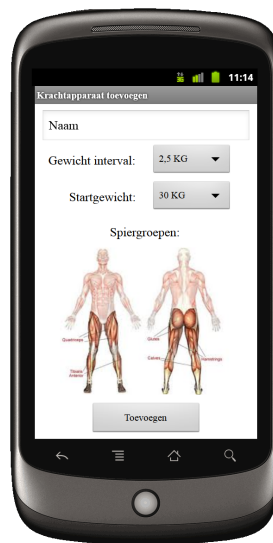


Figuur 4: SessieView

opties om een apparaat aan de sessie toe te voegen en om apparaten uit de sessie te verwijderen. In het hier niet getoonde menu zullen opties zijn om een nieuwe sessie aan te maken, om deze sessie te verwijderen, om deze sessie optimaal te ordenen en om de sessie te starten.

3.4.4 KrachtApparaatToevoegen

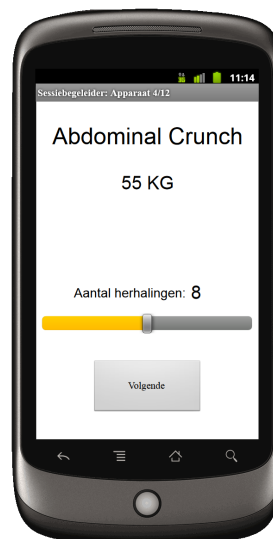
Hier kan de gebruiker een nieuw kracht apparaat toevoegen. De gebruiker moet een naam, het gewicht interval en het startgewicht kiezen, daarna moet op de



Figuur 5: KrachtApparaatToevoegen

spiergroepkaart worden aangegeven welke spieren het apparaat belast. Dit is nodig voor het automatisch optimaliseren van een sessie.

3.4.5 SessieBegeleider



Figuur 6: SessieBegeleider

Dit is wat de gebruiker te zien krijgt terwijl hij bezig is met een sessie in de sportschool. Dit scherm is zo simpel mogelijk gehouden zodat de gebruiker zich zoveel mogelijk kan concentreren op het sporten en niet hoeft te zoeken naar knopjes. Bovenaan staat groot met welk apparaat men nu bezig is en daaronder

staat het gewicht dat gebruikt zou moeten worden. Daaronder is een slider te vinden waar de gebruiker kan aangeven hoeveel herhalingen gelukt zijn en als laatste is er een knop om naar het volgende apparaat te gaan. In het hier niet getoonde menu is nog een optie om de sessie te stoppen.

4 Planning

21 mei – 22 mei	Klassendiagram omzetten naar Android project
22 mei – 24 mei	Interface opzetten
24 mei – 26 mei	ApparaatBeheerder
27 mei – 30 mei	SessiePlanner (handmatig)
29 mei – 31 mei	SessieBegeleider
1 juni	Prototype af
2 juni – 4 juni	Presentatie voorbereiden
5 of 6 juni	Presentatie fase 1
5 juni – 7 juni	SessiePlanner (automatische ordening)
5 juni – 9 juni	Usability test
8 juni – 10 juni	VooruitgangsBeheerder
11 juni – 14 juni	Feedback usability test verwerken
13 juni – 15 juni	Overig programmeerwerk
16 juni – 19 juni	Debuggen
20 juni	App af
21 of 22 juni	Demo
21 juni – 24 juni	Eindverslag