**Rooster App**

**Technisch ontwerp**

Leerlingen: Teun Aarts, Jorrit Meeuwissen

Klas: RIO4-MED3A

SLB: Piet van Steen

School: Radiuscollege

Jaar: 2016-2017

Datum: 23-2-2017

Inhoudsopgave

[Titelvervolgblad 2](#_Toc475612879)

[Modeldictionary 3](#_Toc475612880)

[Klassendiagram 3](#_Toc475612881)

[Datadictionary 3](#_Toc475612882)

[Sequentiediagrammen 3](#_Toc475612883)

# Titelvervolgblad

**Titel verslag** : Technisch ontwerp

**Project naam** : Rooster App

**Naam** **leerlingen** : Jorrit Meeuwissen, Teun Aarts

**E-mail** **adressen**  : [D206218@edu.rocwb.nl](mailto:D206218@edu.rocwb.nl) , D195250@edu.rocwb.nl

**Mobiele** **nummer** **leerlingen** : 06-39 11 06 62, +31641774751

**Groep** : RIO4-MED3A

**OV-nummers** : D206218

**Schooljaar** : 2016-2017

**Datum** : 23-02-2017



**Handtekeningen groepsleden** : ……………………… ………………………..

**Naam** **opdrachtgever-bedrijf** : WallWei

**Adres opdrachtgever -bedrijf** : Terheijdenseweg 350

**Postcode opdrachtgever -bedrijf** : 4826 AA Breda

**Naam opdrachtgever** : Jia Yun Kuo

**Handtekening opdrachtgever** : ………………………………………………..

**Beoordeling opdrachtgever** : ………………………………………………..

**Opmerking opdrachtgever** : ……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

# Modeldictionary

Eerst hebben we gekeken naar de hernieuwde opdracht zodat we wisten wat erin moest komen te staan. Daarvan hebben een lijst gemaakt met de klassen die we nodig hadden in het systeem. Hier hebben we aangegeven waarom we die nodig hadden en wat voor doel de klassen hadden.

# Klassendiagram

Hier staan de klassen die we in de modeldictionary hebben beschreven in vakjes met de functies die de klassen hebben. Hier staan ook wat de klassen nodig hebben voor velden. Dus per klas kan je zien welke velden ze nodig hebben en daarnaast ook nog welke functies ze nodig hebben. De relaties met elkaar staan ook beschreven over het systeem met lijntjes. Zoals een les die kan maar 1 docent hebben omdat wij dat zo hebben gedaan. En zo staat dat ook in de klassendiagram zodat iedereen ook gelijk de relaties weet tussen de klassen.

# Datadictionary

Hier staan precies in beschreven hoe de database er zal gaan uit zien. Dus dat is per tabel welke kolommen erin komen met welk eenheid het is: dus bijvoorbeeld string of int. Daarnaast staat er ook nog bij hoeveel ruimte er moet gereserveerd worden voor een kolom. Daarnaast staat ook of dat het veld verplicht is. Dan ook nog een omschrijving van de kolom en ook nog of het een primary key of een foreign key of helemaal niks.

# Sequentiediagrammen

Hier staat precies beschreven hoe 1 onderdeel van het systeem eruit zal gaan zien. Wij hebben gekozen voor absentie toevoegen. Dus hier hebben we beschreven wat de gebruiker eerst doet om een absentie toe te voegen. En dan laat de diagram goed zien wat het systeem allemaal doet om dat af te handelen. Het laat ook zien welke checks er allemaal gebeuren voordat de data in de database komt. En dat is wel fijn dat iedereen dan goed kan zien hoe dat in systeem gaat.