

AppArt - The art of application design

# Virtual Student Application

Software Design Document



*R&D1*  
*22-04*

Elmar Dongelmans  
Ramon Janssen  
Wesley Janssen

## **Introduction**

### **Purpose**

Het doel van dit Software Design Document is het inzicht geven in ons R&D project (ViStA) door architectonische beschrijvingen te leveren. Het is vooral bedoeld voor onze opdrachtgevers (Sjaak & Erik).

### **Scope**

Waar we uiteindelijk naar toe willen is het maken van een gemakkelijke application, zodat aankomende studenten overwegen het eens te proberen op diezelfde universiteit als waarop ze die leuke android-app waren tegengekomen. Natuurlijk proberen we hiermee ook een leuk cijfertje voor R&D1 te verdienen.

### **Overzicht**

Introduction .....	2
Purpose.....	2
Scope .....	2
Overzicht: .....	2
System Overview .....	3
Architectural Design.....	3
Decomposition Description.....	3
Data Design .....	4
Data Description.....	4
Human Interface Design .....	4
Overview of User Interface .....	4
Screen Images: .....	5
Screen Objects and Actions .....	6

## ***System Overview***

### **Architectural Design**

Belangrijkste opdeling die hier gemaakt moet worden is de scheiding tussen de Activity en de Service. Voor de rest bevatten deze twee delen nog enkele klassen omdat dit structureel wel logisch is, maar deze klassen zullen niet meer zijn dan ofwel een afscheiding van enkele functionaliteiten, ofwel simpele subklassen van klassen die nodig zijn om het OS aan te sturen, zoals handlers en listeners voor communicatie tussen de gebruiker en de app of tussen delen van de app onderling. De service zal één klasse bevatten voor het opslaan van de database, en de activity zal een klasse bevatten voor de GUI. Voor de rest kunnen er nog klassen toegevoegd worden als dat nodig blijkt. We zullen de gekozen scheiding hieronder in meer detail beschrijven:

### **Decomposition Description**

#### Activity

De activity kan hier gezien worden als een soort *view*.

Hierin wordt alle informatie (die van de service afkomt) weergegeven in een GUI, die in een klasse binnen deze activity afgehandeld wordt. Ook zullen alle *eventHandlers* zich hierin bevinden, om deze door te sturen naar de Service.

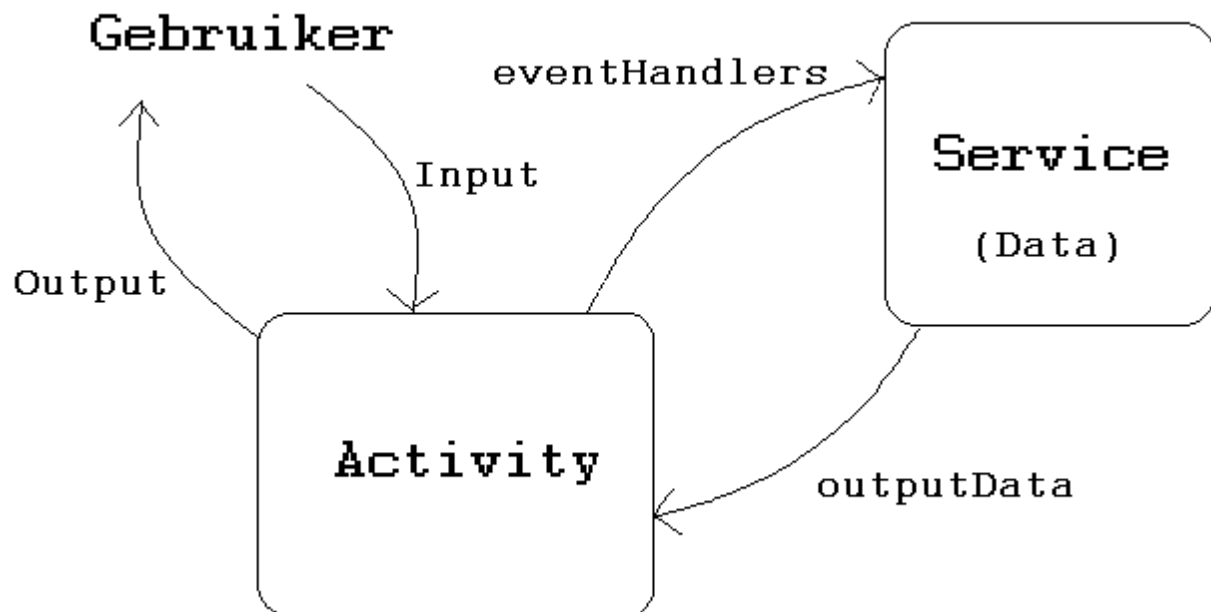
#### Service

De service, is voornamelijk verantwoordelijk voor alle dataopslag/-handling. Alle informatie voor de metertjes zullen hier bijgehouden worden, alsmede de (door de gebruiker ingevoerde) instellingen. Ook zal de service het gedrag van de student al naar behoren aanpassen. Alle data van de ViSTA kan door de activity worden opgevraagd en de service zal dit zelf bij elke update ook naar de activity doorsturen. Buiten de activity om moet de service ook buzzers en toast-berichten kunnen gebruiken om de aandacht van de gebruiker te krijgen. Deze service is nodig omdat deze op de achtergrond kan draaien, terwijl de activity alleen actief is als de gebruiker de activity op dat moment open heeft staan. Met deze scheiding kan er dus voor gezorgd worden dat de ViSTA op de achtergrond actief blijft.

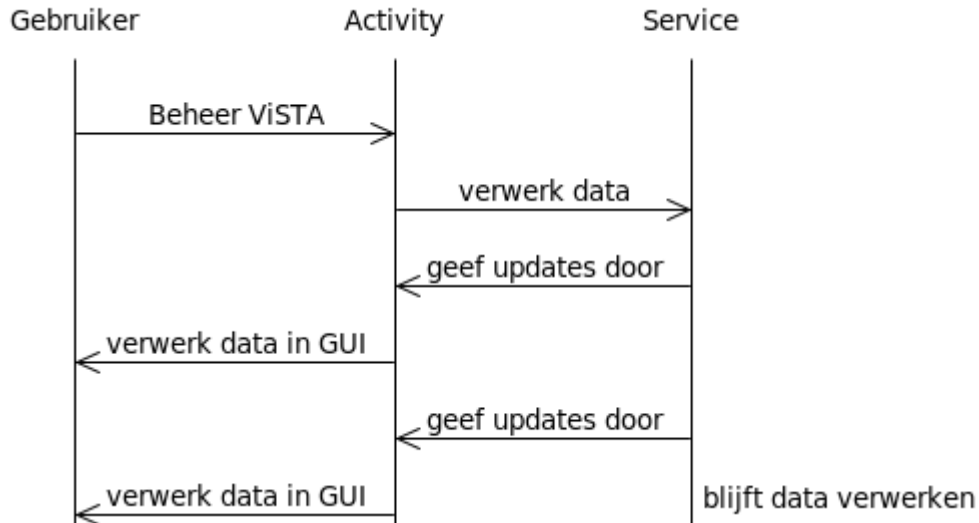
Verder zal de service ook de data op de SD-kaart moeten opslaan als de service of telefoon wordt afgesloten, zodat de gebruiker de voortgang van de ViSTA niet verloren gaat.

# AppArt - The art of application design

Ter verduidelijking de volgende afbeelding:

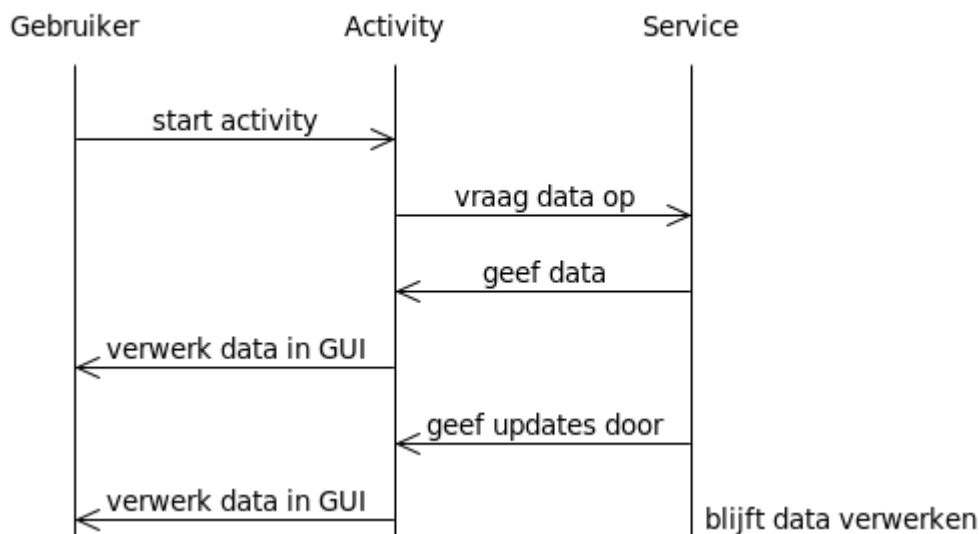


De volgende sequence diagrams geven de volgorde van communicatie tussen onderdelen aan. Voor het beheren van een ViSTA ziet dit diagram er zo uit:



# AppArt - The art of application design

voor het opstarten van de activity als er al een ViSTA aangemaakt is ziet de volgorde er zo uit:



Hierin is te zien dat de service een vrij eenvoudige taak heeft: het beheert de data, en geeft die aan de activity als deze data wordt opgevraagd of als de service deze data update (en de activity actief is).

## Design Rationale

De opbouw van deze app is vrij eenvoudig en werd dan ook snel gevonden. De service is sowieso nodig om de ViSTA actief te kunnen laten zijn op de achtergrond, en de activity is nodig om de GUI te kunnen verwerken, dus alternatieve vormen van opbouw zijn er voor deze hoofdindeling niet. De enige opdeling die verder gemaakt diende te worden is die in klassen die binnen deze service en activity gebruikt worden.

## Data Design

### Data Description

Voor het bijhouden van de gegevens van de ViSTA zullen we een simpele database-klasse gebruiken. Deze wordt door de Service beheerd, en deze database-klasse zal de data in een aantal variabelen of in array-achtige structuren opslaan. Eigenlijk wordt al het relevante (meters, ViStA uiterlijk, instellingen) opgeslagen. Omdat onze app niet erg veel informatie omvat, leek dit ons de beste keuze.

### Data dictionary

De volgende data over de ViSTA zal worden opgeslagen:

## AppArt - The art of application design

- een boolean om aan te geven of deze aangemaakt is
- de naam die de gebruiker aan de ViSTA heeft gegeven
- en getalswaardes voor de verschillende attributen van de Vista (honger, energie, etc.)

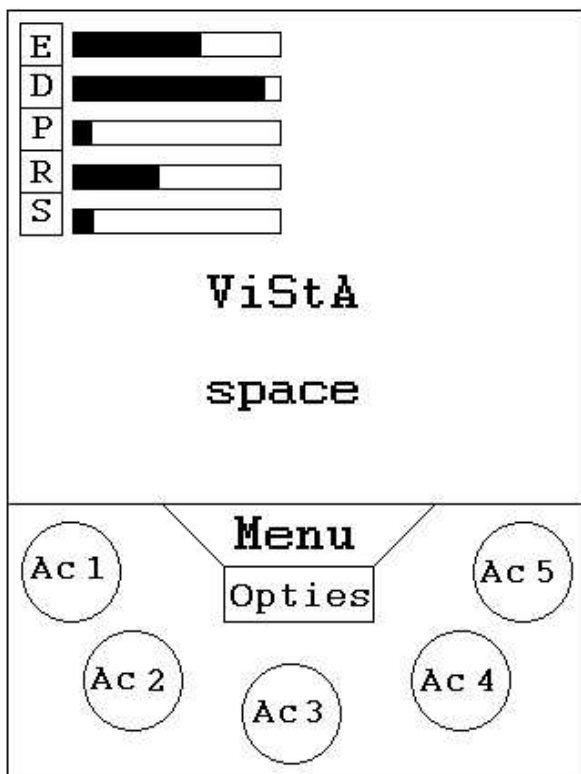
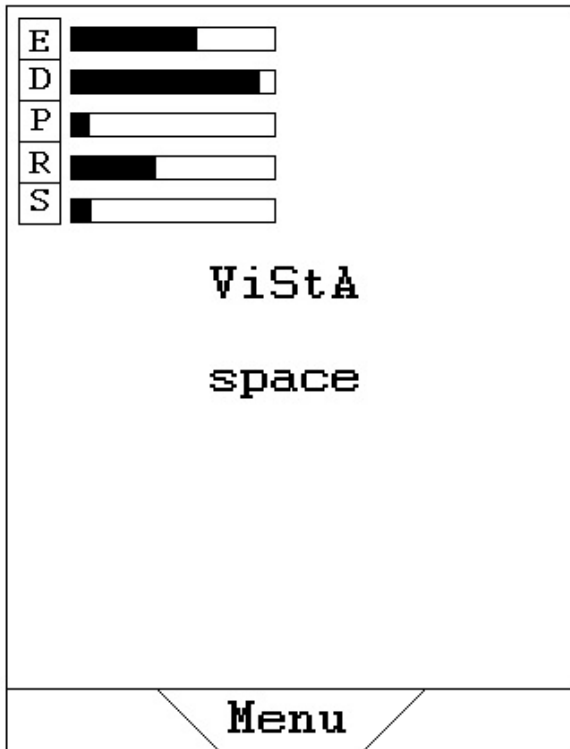
### ***Human Interface Design***

#### **Overview of User Interface**

Er zullen een aantal verschillende manieren zijn om het systeem met de gebruiker te laten communiceren. Allereerst is er als output natuurlijk het scherm van de smartphone, waarop alle informatie op zal worden doorgegeven. Er zal gebruik gemaakt gaan worden van geluidseffectjes en buzzers. Ook zullen er berichtjes doorgestuurd worden naar de gebruiker, om te laten weten dat ViStA aandacht nodig heeft.

Als input gaan we gebruik maken van een aantal traditionele en minder vanzelfsprekende methoden. Natuurlijk gaan we gebruik maken van de touch-functionaliteiten van de telefoon, dit om bijvoorbeeld het menu omhoog te scrollen, de verschillende activiteiten te starten of om de opties te veranderen. Daarnaast gaan we binnen de activiteiten de acceleratiemeter van de telefoon benutten. Dit om bijvoorbeeld het 'leegdrinken van een glas' te simuleren.

**Screen Images:**



# AppArt - The art of application design

## Screen Objects and Actions

Linksbovenin: De meters om de status van de student te laten zien aan de gebruiker, hier aangeduidt met letters (**E**ten, **D**rinken, **P**lezier, **R**ust, **S**tudie). In de uiteindelijke versie zullen we waarschijnlijk gaan werken met icoontjes. (bv een boterham voor eten, boekje voor studie enz.)

Midden: Dit is waar de student komt, momenteel nog ongetekend. We zullen dit in de loop van dit project aanpassen/-vullen.

Onderin: Menu is een menuutje dat omhoog gescrold kan worden (1) om de inhoud te laten zien (2). Deze inhoud bestaat uit opties (een pop-up menu zal hier verschijnen om bv informatie/help te tonen en verschillende instellingen te kunnen veranderen), en een aantal **A**ctiviteiten. Dit zijn enkele 'minigames' om de status (en dus de metertjes) te verhogen. Het menu kan weer omlaag worden gescrold met de touchscreen.