

# Renal Transplant Health Analysis

The analysis of data concerning the health of renal transplant patients

Group C

Faculty Electrical Engineering Mathematics and Computer Science



# RENAL TRANSPLANT HEALTH ANALYSIS

## THE ANALYSIS OF DATA CONCERNING THE HEALTH OF RENAL TRANSPLANT PATIENTS

by

### **Group C**

Louis Gosschalk, Boudewijn van Groos, Jens Langerak, Chris Langhout & Paul van Wijk  
lgosschalk (4214528), bvangroos (4229843), jlangerak (4317327), clanghout (4281705), pvanwijk (4285034)

in partial fulfillment of the requirements for the course of

### **Context Project** in Computer Science

at the Delft University of Technology,  
to be presented publicly on Friday June 26th, 2015.

Project Coordinator:	Prof. Dr. A. Hanjalic Dr. A. Bacchelli
Context Coordinator:	Dr. ir. W. P. Brinkman
Software Engineering TA:	T. M. Hegeman

An electronic version of this report is available at  
<https://github.com/clanghout/Health-Informatics-3/>.



# ABSTRACT

abstract here



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Software Overview</b>	<b>3</b>
2.1	Must Have . . . . .	3
2.2	Should Have . . . . .	3
2.3	Additional Functionalities . . . . .	3
<b>3</b>	<b>Engineering Reflection</b>	<b>5</b>
3.1	GitHub Pulls . . . . .	5
3.2	Code Quality . . . . .	5
3.3	SIG Feedback . . . . .	5
<b>4</b>	<b>Feature Description</b>	<b>7</b>
4.1	Import . . . . .	7
4.1.1	XML . . . . .	7
4.2	Analysis . . . . .	7
4.2.1	The C's . . . . .	7
4.2.2	Additional Analysis . . . . .	7
4.3	Visualize . . . . .	7
4.4	Export. . . . .	7
<b>5</b>	<b>Interaction Design</b>	<b>9</b>
5.1	Methods . . . . .	9
5.2	General Persona . . . . .	9
5.3	Evaluations . . . . .	9
<b>6</b>	<b>Product Evaluation</b>	<b>11</b>
6.1	Product . . . . .	11
6.2	Modules. . . . .	11
6.3	Failure Analysis . . . . .	11
<b>7</b>	<b>Outlook</b>	<b>13</b>
7.1	Improvements . . . . .	13
7.2	Future Development . . . . .	13
<b>A</b>	<b>Sprint Plans</b>	<b>15</b>
<b>B</b>	<b>Sprint Reflections</b>	<b>17</b>





# 1

## INTRODUCTION

Lezer, Verslag. Verslag, Lezer. Aangenaam. 1 PAGE



# 2

## SOFTWARE OVERVIEW

Introduction to software overview. (what has been made, all must haves implemented, etc). 1 PAGE

### 2.1. MUST HAVE

Describe all must haves that have been implemented.

### 2.2. SHOULD HAVE

Describe the should haves which have been implemented.

### 2.3. ADDITIONAL FUNCTIONALITIES

Describe any additional functionalities we have implemented (could & would haves)



# 3

## ENGINEERING REFLECTION

Introduce the chapter, explain what will be described. 2 PAGES

### **3.1. GITHUB PULLS**

Talk about the way we handled GitHub pulls

### **3.2. CODE QUALITY**

Talk about the way we maintained quality in the code

### **3.3. SIG FEEDBACK**

Talk about how we made changes after SIG feedback



# 4

## FEATURE DESCRIPTION

Write introduction to feature description. 2 PAGES

### 4.1. [IMPORT](#)

#### 4.1.1. [XML](#)

### 4.2. [ANALYSIS](#)

#### 4.2.1. [THE C's](#)

#### 4.2.2. [ADDITIONAL ANALYSIS](#)

### 4.3. [VISUALIZE](#)

### 4.4. [EXPORT](#)





# 5

## INTERACTION DESIGN

Introduction to this chapter, describe what we are going to show here. 2 PAGES

### 5.1. METHODS

Here we will list and explain the interaction design methods we have used. (emotions and social aspects are irrelevant, etc)

### 5.2. GENERAL PERSONA

Here we will describe John Doe, an abstraction of our typical user.

### 5.3. EVALUATIONS

Describe our method of evaluation here (friday evaluations, high fidelity prototype etc)



# 6

## PRODUCT EVALUATION

### 6.1. PRODUCT

Even though our product has many features, it can be reduced to 4 simple modules:

- The Data Importation Module
- The Data Viewing Module
- The Analysis Module
- The Visualization Module

These are the same modules that'd been mentioned in the first context specific lecture, so in general we got the basics down. For our features we focussed mostly of the must haves in the requirements. These were roughly specified in the first lectures and after a couple of weeks we got the actual product requirements.

As mentioned in the engineering reflection we'd hoped to finish the must haves early and perhaps tackle a couple of could haves. However due to some issues we encountered we had to adjust our planning to finish all the must haves. This meant that some features we were hoping to get in didn't make the cut.

Overall we must say that it is really hard to anticipate exactly how the final product will turn out. You have an idea of which features you should provide, but the way all those features work together is something else entirely. In the end were satisfied with the result and how everything turned out. We missed the time to really polish it up and get all the bugs out of there, but it is a program which does what it is supposed to and it can do a whole lot more.

### 6.2. MODULES

Explain the functional modules (importing analyzing visualizing and exporting (short recap of chapter 4))

### 6.3. FAILURE ANALYSIS

Talk about the failures in our system. (problems with types (defining abstract methods for float and int) problems with visualisations and importing)



# 7

## OUTLOOK

1 PAGE

This software is part of an educational course. In this course multiple groups create software based on the same customer and requirements. The software most usable to the customer will be used by the customer in his research. If there are any improvements to be made on the software, the improvements will only have to be made if the software is elected amongst the others to be used by the customer.

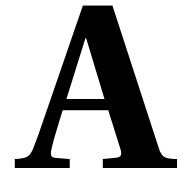
### 7.1. IMPROVEMENTS

Describe the possible improvements in case the software will be used

### 7.2. FUTURE DEVELOPMENT

In case the software will be used, development will probably be continued in the same way, etc etc etc





## SPRINT PLANS





# B

## SPRINT REFLECTIONS