Renal Transplant Health Analysis

The analysis of data concerning the health of renal transplant patients

Group C

Faculty Electrical Engineering Mathematics and Computer Science

TU Delft Delft University of Technology

Challenge the future

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/.

TU Delft Delft University of Technology

ABSTRACT

abstract here

Contents

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

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

3

Engineering Reflection

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

3.1. GitHub Pulls

It was not allowed to push changes directly to the master. When a person wanted to merge a change into the master he had to open a pull request. The pull request had to be approved by at least two team members before it could be merged with the master. The code for the pull request should contain comments and javadoc. Furthermore it should be tested and the code should have a good quality. And at last, the tools we use to evaluate the quality of the code, must say that the code was good. We look very critical to pull request. So often the one who had requested the pull requested had to improve some parts of his code. This led to good code quality. However most comments were related to code style and not directly to the implementation of the code. The size of the pull requests varies a lot. In the beginning of the project they were very large. But since they were large, reviewing was difficult. And since they were large it was not possible to do the review quick, so people postponed the reviews. Therefore we tried to create smaller pull requests. This result of this was that pull requests were handled quicker and that we could review them better. However there were still some very large pull requests. The time that a pull request was open varies a lot. Small reviews were most of the time handled within an hour. Larger pull request

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

Describe the product here in its entirety, its functional modules and the failure analysis. 2 PAGES

6.1. PRODUCT

Evaluate the product: how we thought it would turn out, how it has turned out.

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

A

Sprint Plans

B

Sprint Reflections