

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

| | | |
|----------|--------------------------------------|-----------|
| 1 | Introduction | 1 |
| 2 | Software Overview | 3 |
| 2.1 | Must Have | 3 |
| 2.2 | Should Have | 3 |
| 2.3 | Additional Functionalities | 3 |
| 3 | Engineering Reflection | 5 |
| 3.1 | GitHub Pulls | 5 |
| 3.2 | Code Quality | 5 |
| 3.3 | SIG Feedback | 5 |
| 4 | Feature Description | 7 |
| 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 |
| 5 | Interaction Design | 9 |
| 5.1 | Methods | 9 |
| 5.2 | General Persona | 9 |
| 5.3 | Evaluations | 9 |
| 6 | Product Evaluation | 11 |
| 6.1 | Product | 11 |
| 6.2 | Modules. | 11 |
| 6.2.1 | Data Importation | 11 |
| 6.2.2 | Data Viewing. | 11 |
| 6.2.3 | Data Exportation. | 12 |
| 6.2.4 | Analysis | 12 |
| 6.2.5 | Visualization | 12 |
| 6.3 | Failure Analysis | 12 |
| 7 | Outlook | 13 |
| 7.1 | Improvements | 13 |
| 7.2 | Future Development | 13 |
| A | Sprint Plans | 15 |
| B | Sprint Reflections | 17 |

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 five simple modules:

- The Data Importation Module
- The Data Viewing Module
- The Data Exportation 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

In this section we'll discuss the precise operation of the various modules.

6.2.1. DATA IMPORTATION

The data importation module consists of two parts: the data reading part and the data describing part. The data reading part takes an XML file which can be generated using the data describing part. We've noticed the describing process isn't completely intuitive and it is something we should have spend more time on. That being said, the process in itself is rather complex and a UI can only get you so far.

6.2.2. DATA VIEWING

The data viewing module is rather simple. You have a list of the left of the various tables present in the program and a table on the right showing the contents of the table. There is also a button to save a table which opens a wizard, more on this in the next section. This module is rather simple and we're satisfied with its simplicity.

6.2.3. DATA EXPORTATION

This module consists of a dialog which can be used to save a table to a file. It's got a couple of simple options to select how exactly the file is to be saved and it's all very user-friendly.

6.2.4. ANALYSIS

The analysis module consists of a textbox and a couple of buttons to run, save and load scripts. This is by far the largest module as the entire point of our program was performing those analyses. We support a great number of analyses and some other processes which can do some rather complicated stuff with the input data. We wish we'd implemented a couple more analyses, but there was only so much we could do in the given time. In the end we're content with the rather complicated, yet subtle workings of our language. We do realize that the complex workings of our language present the user with a bit of a challenge to learn our language.

6.2.5. VISUALIZATION

In the visualization module we support generating a box plot, frequency graph and state transition matrix. All of this is done through a simple dialog, which is simple and easily understandable. We do think that a little more time could have polished it up a bit and removed a couple of bugs. However the graphs created are pretty and useful. Also we don't think that the visualization is the core part of our program, there exist other better tools to do such things.

6.3. FAILURE ANALYSIS

In this section we'll discuss the various issues that are still present in the program.

One of the major issues we encountered in the realisation process was that Java generics aren't available in runtime. This is something we encountered along the way and didn't anticipate beforehand. Currently it means that we're not able to do type checking in our language. So multiplying a date by a boolean is okay for the parser and it will throw a non descriptive runtime exception. In the future I think it is something we'll take into account when dealing with generics.

One of the other major issues is that our program isn't very user friendly. This isn't a major concern as the program is very specialistic and main target audience is very limited. However it is something we should have paid more attention to.

Another issue is one of performance. The data that is entered in our program is quite large and it hasn't been optimized one bit. This means the user has to wait a while for the larger analyses. However it is perfectly possible to reduce the data by performing a simple analysis. This should speed things up for the user.

In the visualization module there is an error message which doesn't go away when it should, this could have been solved by testing more thoroughly.

7

OUTLOOK

1 PAGE

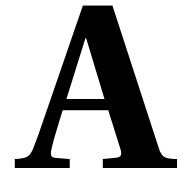
This software is part of an educational course. In this course multiple groups create software based on the same customer and requirements. The software most usable to the customer will be used by the customer in his research. If there are any improvements to be made on the software, the improvements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