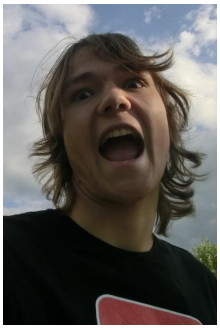
Contextproject Programming Life
Group 2 - GEVATT
Final Report

TU Delft



Ruben Bes
rbes
4227492

Mathijs Hoogland
mhhoogland
4237676

Jasper Denkers
jdenkers
4212584

Robbert van Staveren
rhvanstaveren
1527118

Willem Jan Glerum
wglerum
4141040

June 19, 2014

Abstract

This is the final report for the Programming Life Contextproject, a second year course from Computer Science at TU Delft. The Contextproject course is about applying all learned skills in a particular context at developing a piece of software. In this case the context was bioinformatics and we worked 10 weeks with a team of five people.

This document contains the main information about development, implementation and validation of the product. Main features of the product will be presented and it will be discussed how they satisfy the user needs. Furthermore, this document will contain an HCI module about the interaction of users with the product. Finally, an outlook will be given to show what possible improvements could be implemented if this project will continue in the future.

Besides this document several other documents are made covering other parts of the project. This is the final document consisting of the most information about the project.

Contents

1 Introduction

The application developed is called GEVATT, which stands for GEnetic Variations Analyzer Through Triodata, which is an application used by doctors to browse genetic information of patients. Such applications are also called genome browsers. It is a secured web application based on the Play Framework. Doctors need to login from any device with a browser in order to use the application.

The application makes it possible for a doctor to upload a VCF file and let the application analyse it. A VCF (Variant Call Format) file is a file used in bioinformatics for storing gene sequence variations. This storing of genetic information of patients is based on variations between de patient and a reference genome. This is done because storing al information would be taking too much space.

After uploading the file the user of the application (a doctor) waits until the file is processed. In the meanwhile the doctor could browse other patients he uploaded information about earlier. After uploading, the first important part of the applications is executed: analysing the data. The outcome of the analysation consists of mutations found in the genome of the patient and the relations between these mutations.

Secondly, the main focus of the application is about visualising the found mutations. This is firstly done by giving a main overview of the whole patient. An overview of all chromosomes is given and per chromosome is indicated if and how many mutations it contains. Besides a visual overview of the chromosomes there's a tabular overview given with all mutations and some extra information per mutation. This makes it easier to estimate which mutations are more harmful than others.

Most information is shown on the overview pages per mutation. The pages with these visualisations have two visualisations which both give another insight in the mutation. The first part shows the position of the mutation relatively to a gene. The second visualisation is a graph based interactive visualisation showing proteins related to the mutation and the connections between these proteins. Here we also have a tabular overview per protein with per protein information about which deceases they could cause and to which other mutations of the patient they are related.

2 Overview

The developed product, a secured web application, is built on de Play Framework. It contains several parts that work together to deliver a user friendly environment for the users to explore genetic information.

2.1 Authentication

Starting with the basis, it's a secured web application where a user needs to log in. After logging in some secured pages become accessible. Some other pages already are accessible before logging in, e.g. the documentation and about pages. We distinguish the following parts:

- A login page

- Securing pages that aren't publicly accessible
- Redirect if secured page is unauthenticated requested
- Prevent doctors from accessing patients data of other doctors

2.2 Context specific

The web framework and it's builtin securing module isn't developed by ourselves so we could focus on developing the context specific parts. This was mainly separable in three parts: analysing data, retrieval of relevant data of databases and visualising data.

2.2.1 Data analysation

Analysing the data is done by processing the VCF file and detecting mutations. The outcomes of the processing is used in the visualisation. At both of these parts there's information used given by some databases. The information was used the get more information about mutations and find relations between several mutations. The application does the following:

- Read the VCF file and find mutations of two types: de novo's and recessive homozygous
- Save metadata about the uploaded file
- Find relations between found mutations

2.2.2 Data retrieval

Information from multiple databases is used for both visualisation and finding relations between mutations. This includes:

- Manage connections to multiple databases
- Querying databases to get relevant information from mutations and proteins

2.2.3 Visualisation

The most important part of the application: visualisation. Multiple views at the same found mutations are taken to give as most insights as possible. This includes the following visualisations:

- An overview of all found mutations in a patients VCF file with a view per chromosome
- Per separate mutation a distinct page with:
-
  - The position of the mutations relative to nearby genes
  - A graph with a protein related to the mutations and related proteins to that protein

3 Description

This chapter covers a detailed description of the developed functionalities in the product. The description is based on the overview of functionalities given in the previous chapter and therefore has the same structure.

3.1 Webapplication

The application is a web application based on the Play Framework. This means the application is accessible via devices with an internet connection and a browser. The layout is optimised for devices with a screen with a minimum width of 1024 pixels. The page structure is as follows:

- A login page (see authentication below)
- A dashboard page with an introduction of the application and links to the context specific parts
-
  - The patient overview with a sortable list of all patients with some meta data. By clicking on a patient the user gets on a patient specific page
  - On the patient page an overview of the found mutations is given. It contains a visual overview of the chromosomes and the mutations found per chromosome
  - Per mutation there's a separate page with visualisations and information (see visualisation below)
  - There's a separate page for adding new patients to the database

3.2 Authentication

The authentication part takes care of securing the pages and data of the application from being accessible to everyone. It does the following:

- If a user isn't authenticated and tries to open a secured page, he is redirect to the login page
- At the login page, a user needs to fill in his username and password to sign in. When these credentials aren't recognised, the user receives an error message
- When logged in, each page contains some user specific information, like his name. Furthermore, the links to application specific pages become visible and a logout button appears

3.3 Context specific

3.3.1 Data analysation

The analysing of data is done after a VCF file is uploaded.

- When a user has added a patient by filling the associated information and uploading a VCF file, the user gets redirect to the patient overview. In the background the file gets processed. After processing the patient overview gets automatically updated and the patient becomes available.
- The mutations that are found are stored in a databases dedicated to the application
- There's searched to relations between the found mutations and those relations are also stored in the database

3.3.2 Data retrieval

We use the following databases to retrieve relevant information:

- CADD for retrieving scores of danger of mutations
- dbSNP for retrieving all kind of information related to SNPs
- STRING for retrieving everything related to proteins

3.3.3 Visualisation

Several visualisations are used the bring the retrieved information in a clear way to the user.

- The first visualisation is about giving an overview of all mutations found and showing them per chromosome. This overview is given on the patient overview page. It contains a graphical representation of all chromosome pairs and each pair is colored red or black if it respectively contains a mutation or not. While hovering over a chromosome pair, a list appears with links to the mutations found in that chromosome pair.
- On the pages for individual mutations there are two extra visualisations
-
  - The first mutation specific visualisation is the top one found on the mutation page. It shows a part of the chromosome the mutation is on and displays the position of genes relative to base pairs in the chromosome indicated by a number.
  - There's also an overview in a graph form of proteins related to the mutation. The protein directly related to the mutation is marked and in a graph structure related proteins are shown. This graph is interactive so proteins can be dragged around to get a clearer view. Connections between protein are shown darker and thicker if the connectivity between two proteins is high relatively to the other connections in the graph.

4 Human Computer Interaction (HCI)

Blaat

5 Evaluation

Blaat

6 Outlook

GEVATT is developed in quite a short time. The course Contextproject lasts about ten weeks and in this ten weeks the team had to become skilled in the context, make accompanying documents and present the final product. So only a limited part of the time is actually used to develop the application. This means there's focussed on the main parts of the application and there are more functionalities that could be added later if development would continue. This chapter is about what functionalities could be added.

One thing is exporting; this is currently not possible in the application. The found mutations and information from the VCF files are only visible in the application itself and require logging in. For a doctor, it might be handy to export some data and visualisations to make it easier to present them. E.g. it's easier for a doctor to show a print of visualisations to his patients than showing a screenshot of the application. It also might be handy to have the data exported to a spreadsheet format.

The application is web based and therefor in principal platform independent: each device with a modern browser and an internet connection could open the application. Something that is limited to the use of the application now is the minimum screen width. When a device with a resolution of lower than 1024px is used, the user need to scroll to see all information. The application could be modified so that the layout adopts to smaller screens and become thereby usable on tablets and phones. A nice feature, but is questionable if a doctor would use the application on such devices.

Currently the application is mainly focused on individual mutations found in the uploaded data, and less on the relations between these mutations. When development of this application would continue, it will probably make sense to focus on implementing/extending visualisations that show these relations.

Testing is only done on a low scale. If GEVATT would be used in production, there should be some testing be done based on heavy use. Furthermore, if the application would be used in producteion, some managing features should be added. For example, creating accounts en revoking access to specific doctors. Due to time considerations this is left out in this project so far.