Written by: Oliver Zellman Waldenström & Fabian Noack

|  |
| --- |
|  |
| Health Analyzer |
|  |
| Detailed Project Specification |

Health Analyzer

Detailed Project Specification

General Idea

The innovative driven idea of this project is built upon the interaction between medicinal substances, called the “Cocktail-effect”.

The idea is, that the user or the patient, has the possibility to add their prescribed medications to his profile, and gets feedback on potential drug interactions.

This feedback can consist of warnings, concerning the interactions severity, as well as a description of the potential interaction.

There are already existing platforms where you can check potential interactions for a medicine. What makes our platform unique is the feature to add prescribed medicines to a persistent profile. When the user has added his medicines to his account, our application will analyze potential drug interactions and return feedback to him. The parameters that we return are the number of interactions, the severity and a short description of the interacting substances.

As additional functionality, we offer a forum, where users can share their experiences, questions and maybe advices, concerning specific drug interactions.

Technical Specification:

For this project, we have decided to follow the guidelines of the MVC-Design pattern, using different frameworks, libraries, tools and services.

* J2EE – provides basic libraries and web application functionalities, such as JavaBeans and Facelets
* JSF – used to generate Frond-End
* The RxNorm API – a web service API, where we get medicinal proposals related to the users input
* The RxNav interaction API - provides information regarding drugs and their interactions
* MySql – database, used to store profile information, drug interactions and forum tables
* PhpMyAdmin – the tool, used to simplify the process of creating MySql-tables
* Hibernate – used to simplify the connection to MySql
* GitHub – used as common repository to be able to work synchronized separately
* DoomParser – used to parse XML from the RxNav webservice

The main application – Health Analyzer

MedicineAnalyserUML

Figure 1 : Project component – the health analyzer

For the Health Analyzer we use JSF to develop the Front-End Client interface, which represents the view in MVC and connected it to the controller, which is in this case represented by ManagedBeans, that will apply the business logic.

Internally the controllerBean handles requests of user requests and forwards queries to the database, as well as it sends requests to retrieve data from the Rxnav RESTful web services.

The Rxnav API offers web services to retrieve information regarding drugs and their attributes, as well as their relations to other drugs, which have concurrent interactions(see “Cocktail-effect”).

The response will be send using XML or JSON, so we implemented a helper class “XMLParser”, that will parse the information into Simple Java Objects, which can then be used by the controllerBean.

The retrieved information will then be displayed to the Front-End or saved into the user’s profile data in the database.

(use of mysql)

An additional functionality of our web application is a forum, where users can create and comment on medicinal threads, concerning questions, answers and experiences.

As to support a forum and user profiles, there will also be the functionality of registration, login, logout and removal of users.

Components:

JEE

JSF

JavaBean

Our goal is to create a web application that analyses our users medical state and provide them with related information topics. We will try to find a web service that provide data about medical states so our application can analyze and send information back to the user/client suggesting for example pharmaceutics or advise them to seek medical assistance. Each user will be given certain input parameters for example blood pressure, sugar levels etc. This information will be used to display diagrams on the user’s profile so he can have an overview of his current medical state.

We have briefly researched multiple web services that can provide reasonable information to our database. But we have yet to decide what services to use. One example is <http://www.programmableweb.com/category/medical/apis?category=19994> which is listing 242 web service API's concerning medical services.

We will continue our research along the development of the project.

Concerning network references, we will use the lectures and the course literature as well as oracles API.

As consequence of using web services we will implement a n-tier application structure. A detailed diagram of the structure will be provided along with the final project proposal (second meeting). We will implement a simple mail or contact form so the user can contact the manufacturer if necessary.

References:

<https://rxnav.nlm.nih.gov/RxNormAPIs.html#>

<http://docs.oracle.com/cd/E17802_01/j2ee/javaee/javaserverfaces/2.0/docs/api/>

Java Network Programming – Elliotte Rusty Harold (Third Edition).