KIT

Karlsruher Institut für Technologie

Functional Specification

Hier koennte unser Logo oder Bild hinkommen. Die Sectionsueberschriften und Subsections bekommen die Tage noch n schoeneren Look.

November 13, 2012

Functional Specification	**Alexander Noe**
Design	**Jonathan Klawitter**
Implementation	**Anas Saber**
QA / Testing	**Niko Moraitakis**
Final	**Lukas Ehnle**

Introduction

This is the functional specification for Group 14 in the PSE assignment 10 in winter semester 2012/13. In this project we have to analyze and visualize the serverlog of the Skyserver, which is a database for astronomical data. This assignment is one of two PSE assignments which will be handled in English. Because none of us speaks English as his first language, we have to apologize that our documentation may contain simple or maybe in some parts incorrectly used language. We focus more on having easy-to-understand and correct documentation than on perfect English, which sounds like written by a native speaker.

We are going to write a web-frontend in javascript which allows users to access the finished application from everywhere with recent browsers. The actual application can be spliited into two different parts. The first part get the internal name Server Log Parser (SLP) and is going to convert the CSV-formatted serverlog from Skyserver into a data-warehouse. It will be accessible via admin-login to the webpage where we can enter a logfile which will be parsed into our warehouse. The second part - which got the internal name Warehouse Analyzing Tool (WHAT) works on this data-warehouse and allows every user of our webpage to create various charts, for example scatterplots and histograms. Due to the fact that we split our project in two smaller parts, many parts of this specification are going to be splitted in two.

We have to state that no one of us has much experience in working with databases and javascript. Therefore we can't guarantee for the correctness of all information stated in this specification. It could happen that we will alter some of the specifications when designing or implementing the actual program.

Contents

1 Goals

1.1 Server Log Parser (SLP)

1.1.1 Obligatory criteria

- The SLP can read a CSV-formatted log file from sdss.skyserver.org and fill a data-warehouse with multi-dimensional user information. The dimensions of this data are location of the user, location of the database and time of the server-access.

- SLP will recognise invalid logs and won't add them to the data-warehouse. Every log with a mistake won't be accepted, because an error-message is not as bad as a corrupted data-warehouse.

- SLP will be accessible with admin-login to the webpage.

1.1.2 Optional criteria

- SLP will contain a small manual which is designed for experienced users.

1.1.3 Exclusion criteria

- The SLP can only use CSV-formatted logs from sdss.skyserver.org. It can neither read logs in another format nor logs from another source.

- There is no way to avoid using SLP when adding data to the warehouse. This can stop corrupting the warehouse to guarantee correct data in the warehouse.

- SLP doesn't correct mistakes in the log file.

- Due to being only a tool for the persons who wrote it, SLP doesn't need to be easy to use for first-time users. It will be a console program or maybe contain a simple GUI.

1.2 Warehouse Analyzing Tool (WHAT)

1.2.1 Obligatory criteria

- WHAT can use the data-warehouse which is filled with user data from sdss.skyverser.org to create various charts. Those charts are created by user command and are visible on a javascript-webpage.
    - scatterplots
    - histograms

- The website contains small guides for experienced users on how to create a specific chart. (similar to the guides at skyserver.sdss.org)

1.2.2 Optional criteria

- The language of the website can be changed into different languages. (e.g. German)

- WHAT will support different chart-types.

1.2.3 Exclusion criteria

- WHAT takes correct data in the warehouse for granted. It doesn't check the data because it already checked by SLP.

2 Usage

2.1 Applications

Wat.

2.2 Target groups

The target group of this application is people that want to analyze and visualize data collected from the skyserver. This implies 1) People that know about the skyserver. Since the skyserver only allows queries via sql and has an arcane website this greatly reduces the prospective audience. We do not expect the webserver to run into scaling issues. 2) People that are interested in what other people are using the web server for. Our project mainly visualizes sql queries from other people. Knowing sql, while not a prerequisite, would allow the user to fully utilize the software.

This is a technical audience.

That said, this does not prevent the (web) user interface from being functional, usable and prettier than what you would expect a group of computer science students to design.

2.3 Operating conditions

The program is mainly used as a website, with the primary difference being that the server has to be started if the capacity for it to run all the time on a dedicated machine doesn't exist. //this is formal english for: we do not have a server. Gibe server plos The program needs a server to run.

If a dedicated server exists, the program can be used from anywhere with a decent network connection with the server.

If not, the program can still be run on the same computer as the server (on localhost), but the server will have to be started first.

3 Operating environment

3.1 Software

The client requires the latest version of a modern browser, like Chromium or Firefox. Other browsers might work, but might not been tested as extensively. Javascript needs to be enabled.

The server requires a recent version of the Java Runtime Enviroment.

3.2 Hardware

The server has to be fast enough to support all clients. This depends on the expected number of clients. Most computations will be done on the server.

The client needs to be fast enough to visualize the data received from the server. The required hardware thus depends greatly on the amount of data that needs to be visualized. That said, any recent computer, with for example, an Intel® Core™2 Duo CPU E8400 @ 3.00GHz × 2 processor, 4GB(2x2GB) of 667Mhz DDR2 SDRAM, running Ubuntu Linux 12.10 Quantal Quetzal, //this was, like, a joke, because it's, like, my computer as a point of reference, should be able to visualize about 50.000 data points on a scatterplot with ease.

3.3 Orgware

The server needs to be able to connect to the client with a reasonable latency. The server has to be set up before the client can connect to it.

3.4 Product interfaces

4 Functional requirements

4.1 Main functions

This functions are required to fulfill the obligatory criteria.

General

/F10/ Provide access via web page

/F20/ Provide options for chart types

/F30/ Show diagrams

/F40/ Show histograms

/F50/ Provide option for the two variables of the histrogram

/F60/ Provide option for the interval or selection of the x-axis variable

/F70/ Show scattered plots

/F80/ Provide option for the two variables of the scattered plot

/F90/ Provide option for the intervals or selections of the two variables

/F100/ Show bubble charts

/F110/ Provide option for the three variables of the scattered plot

/F120/ Provide option for the intervals or selections of the two variables

/F130/ Show information about chart types

/F140/ Show information about selectable variables

Adminstrator specific

This functions are required for the adminstrative business.

/F150/ Provide access via web page with administrator rights

/F160/ Provide the oppertunity to initialize the data warehouse *

/F170/ Provide the oppertunity to pass log-files to the parser

/F180/ Provide the oppertunity to clean the data warehouse

Parser specific

The following functions specify the parses functionality. Thereby /FXU/* to /FXZ/* specify the function /FXT/.

/F190/ Extract specific data from the log-files

/F200/ Extract access database, access time and user information (dimensions)

/F210/ Extract number of rows, elapsed time, busy time (measures)

/F220/ Extract type of data requested from the where part

/F230/ Transform the data to fit into the data warehouse schema

/F240/ Load the data into the data warehouse

Analyzer specific

/F250/ Run specific queries against data warehouse

/F260/ Build up diagrams with received data

/F270/ Build up histrograms

/F280/ Build up scattered plots

/F290/ Build up bubble charts

4.2 Optional functions

To fulfill the optional goals the following functions are required.

/F300/ Select language on web page

/F310/ Switch language to German or other

/F320/ Build up and show bubble map

/F330/ Build up and show other chart types

/F340/ . . .

5 Data

5.1 First Whatever

/D10/ blabla

/D20/ tadaa

5.2 Next Whatever

/D30/ 30!

/D40/ 40!

6 Nonfunctional requirements

6.1 First Whatever

/NF10/ blabla

/NF20/ tadaa

6.2 Next Whatever

/NF30/ 30!

/NF40/ 40!

7 Global test cases

13

8 Models

8.1 User stories

8.2 Object models

8.3 Dynamic models

8.4 Web interfaces

9 Glossary