Linnæus University
School of Computer Science, Physics and Mathematics

Degree project

Visual Analysis of Swedish Research Council's Project Database

Author: Thorsten Haga
Date: 2012-05-12
Subject: Computer Science
Level: Bachelor Degree
Course code: 2DV00E

Abstract

A human can understand data visualizations easier than reading the source . The goal of this thesis is to support the user with an application to fulfill this problem, so he is able to cope with the data and also filter it for his interests. This thesis aims to visualize projects of the Swedish Research Council (Vetenskapsrådet) supported by the latest web technologies. The visualizations, which will be created from the projects, are interactive, so the user is able to select a single university and their faculties by years and other categories. The application is quite transparent, so it is conceivable that it also fits in most organisations or firms who want to analyse their departments project budgets.

The web application is built with the newest Hypertext Markup Language Standard (HTML5) and Cascading Style Sheets (CSS3). A large part of the application was programmed with the help of the new Toolkitfrom the Google family which is called Google Web Toolkit.

Keywords: Vetenskapsrådet, Swedish Research Council, visualization, GWT, Google Web Toolkit, HTML 5, CSS 3

Contents

List of Figures

List of Tables

1 Introduction

In this chapter the problem of the thesis and the motivation to choose this particular topic will be described. Another part defines the goals and their criteria.

1.1 Problem and Motivation

Information is everywhere.We live in a society were the amount of data increases steadily. This arises from different sources for different purposes.Nevertheless, it can be assigned in two main objectives: communication and exploration. This means that information will be used to communicate an idea or by using information to create new ideas.Compared to earlier time periods, the production of information has dramatically rose and, due to the Internet, information is available at any time which has the unsightly side effect of an information overload in our society.The Internet offers a wide range of information which arrives unfiltered to the user and makes it hard to separate important information from that which is unimportant.[13] Therefore it is necessary to visualize data in a proper way, so that it is easy for the human brain to understand. The data, that is used in the web application, will be extracted from the Vetenskapsrådet's database which was established in 2001. The Vetenskapsrådet,the Swedish Research Council (SRC), which is a part of the government. They are responsible for the support and the development of basic scientific research projects. Therefore they are operating a database which contains all former projects (beginning from 2001), the currently active and the upcoming projects in Swedish universities which have applied for money support from the SRC. The main interest in these projects is which university has applied for the funds of Science Research Council, for which period of time and which institute reached the highest numbers of projects.

1.2 Goals and Criteria

This section describes the goals and the criteria which will be used to solve the problems of this thesis. **The declared goal is to build an interactive web application from the data of Vetenskapsrådet to give a overview of the invested money on Swedish projects.** Unfortunately, due to the lack of time ,this thesis can not cover all potentials of the data and the resulting outcomes (visualizations). The amount of possibilities leave space for a further thesis , maybe even a master.
The declared goals are as follows:

- The first goal is to create an web application which supports the new web standard HTML5 and CSS3, including the visualized data and a detailed overview of all the projects depended on their universities.

- The second goal will involve the user in the application. Analysing, filtering data and interaction between the user and the application is mandatory. This means for instance that the user can define a time range to filter the data to see only the projects of a specific university in this time period.

- The last goal aims to update the project database of the Swedish Research Council without duplication (by the first run, the application should be capable of downloading the entire database).

The application needs to be separated into logical parts (modules) which should, as far as possible, operate independent from each other to make the system more flexible. This procedure will avoid reprogramming the whole application by switching to alternatives.

1.3 Report Structure

The report contains eight Chapters. The first three are about theoretical work and leads to the practical part which starts with the fourth chapter. The theoretical part is beginning with the Introduction which contains the explanation of the problem and the motivation which stands behind it. The second section implies the desired target that this thesis aims to reach. Thereafter follows a short snapshot of the visualization history and a few examples of visualizations of today. The third chapter contains the theoretical approach of visualizations, especially how to create structured data and how to measure information content. By the end of the third chapter the thesis passes over to the practical part which contains the interactions that the application offers to the user. Furthermore the tools will be introduced which have contributed to create the application. The fifth chapter shows the non-functional and functional requirements that the application have to fulfill. Thereafter, the sixth chapter will illustrate the structure of the architecture. Throughout the seventh section, problems which came up during the development will be discussed. The last section contains the conclusion and proposals for the future work and which parts of the program could be enhanced.

2 Related Work

This chapter introduces a small summary of the history of visualization and some selected examples of how effectiv and impressive visualizations can be.

2.1 History

In this section it will be discussed, what background visualization has and how far it influences our society. Data visualization is simply the presentation of quantitative or qualitative data which makes it easier for the reader to understand and process the information. One of the first graphics which has been found was in the Chauvet-Pont-d'Arc Cave about 30000 years ago.It contains 250 paintings. One of the first writing systems used *logograms*, which is a system that uses pictures to encode symbols and even whole words. The first early visualizations were mostly maps for traveling or used for religion and communication purposes.

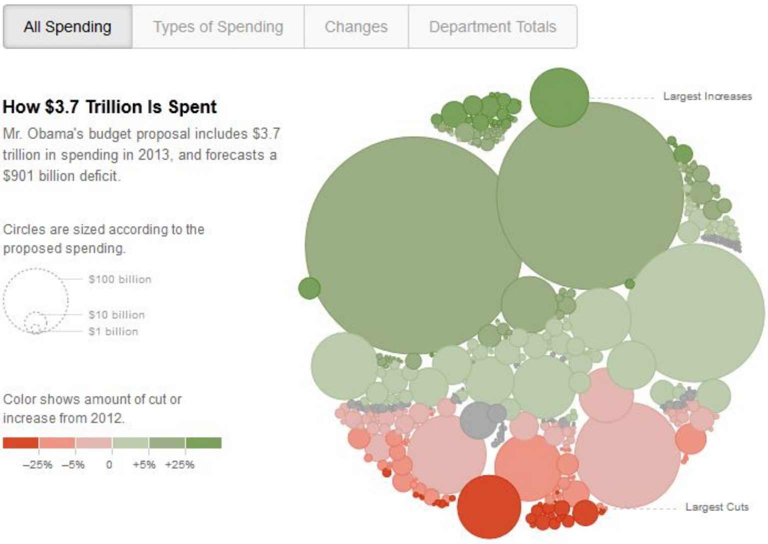
Additional to the maps which were created in the early phases of visualization, Leonardo Da Vinci created, among other things, very famous medical visualizations which were used to train new medical doctors.(Please see: [16, S.15]) A Pioneer in the field of visualization was a Scottish engineer and political economist, William Playfair. He is credited as the inventor of the modern chart and graphical methods of statistics. especially known for these four diagrams : the line graph, the bar chart, the pie chart and the circle graph and published the first charts in his Atlas Commercial and Political Atlas and Statistical Breviary from 1786 [5] which was in the same year.

Figure 2.1: Peutinger Road Map shows a 70000 miles long section of the roman world [16, S. 8]

2.2 Related Visualization Work

Obama's Budget Proposal 2013

The New York Times has produced a visualization of the budget proposal from the President of the United States Barack Obama for the year 2013. The visualization points out how much the various departments for this legislature period become ,in proportion with the last accepted budget. The difference will be shown in percent. Furthermore it illustrates precisely how much money the various departments receive by the volume of the various circles.



How $3.7 Trillion Is Spent

Mr. Obama's budget proposal includes $3.7 trillion in spending in 2013, and forecasts a $901 billion deficit.

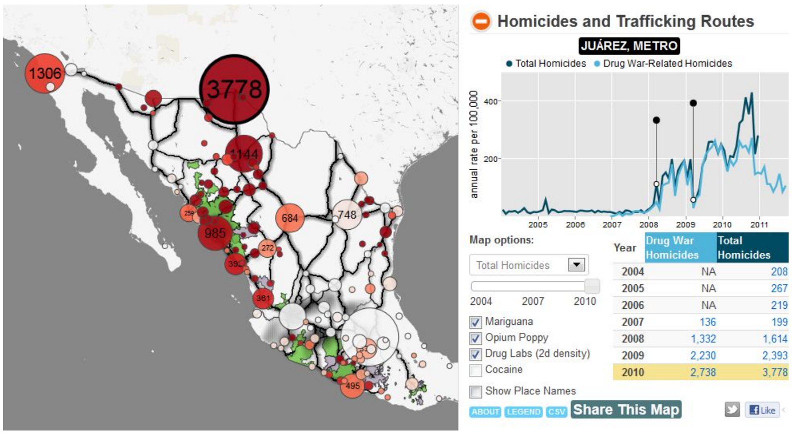
Circles are sized according to the proposed spending.

Color shows amount of cut or increase from 2012.

Figure 2.2: Obama's Budget Proposal [3]

Map of the Drug War in Mexico

The visualization present a google maps which compare homicides and drug related homicides with the option to examine marijuana,opium, and drug-lab-related homicides.By clicking on one of the circles it will highlight the number of murders over the time period.Important events are highlighted in the graph too.



☑ Mariguana
☑ Opium Poppy
☑ Drug Labs (2d density)
☐ Cocaine
☐ Show Place Names

Year	Drug War Homicides	Total Homicides
2004	NA	208
2005	NA	267
2006	NA	219
2007	136	199
2008	1,332	1,614
2009	2,230	2,393
2010	2,738	3,778

Figure 2.3: Map of the drug war in Mexico [14]

The ChronoZoom Project

This opensource project ChronoZoom, which was initialized from the Microsoft Research Team, takes advantage of the full power of HTML5 to create a astonishing new timeline tool that aims to deliver an online visualization of Big History. It combines studies from many different disciplines (including biology, astronomy, geology, climatology, archeology, economics, anthropology and environmental studies) to visualize history from the beginning of time. It encloses 13.7 billion years and includes an enormous amount of data. ChronoZoom combines this data from the cosmos and earth history, to make them search able and displayed in a comprehensible way.

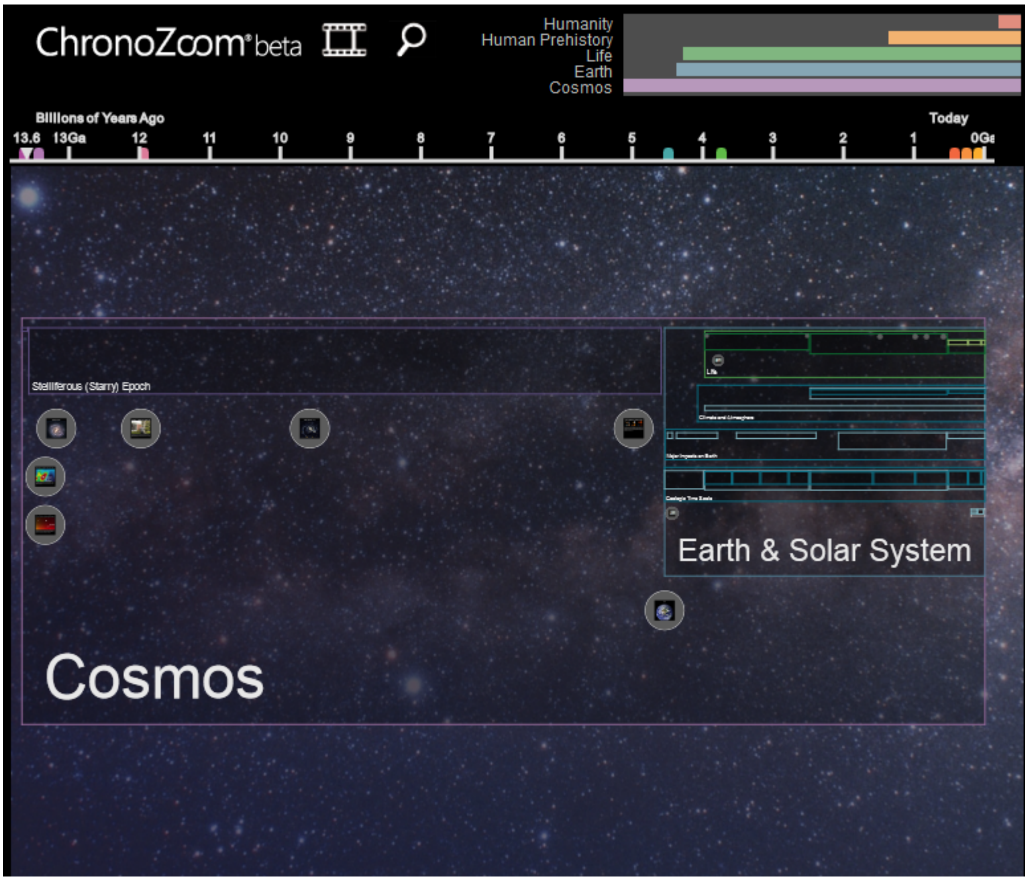


Figure 2.4: A astonishing way to visualize the Cosmos and Earth's history [11]

3 Visualization

This chapter provides an introduction of how data could be visualized and a general introducing about visualization.It presents reasons why the following charts were chosen and discusses the process.

Which steps are required to get a visualization? This was the first question ,which came up, after the data were explored. The visualization pipeline in figure 3.1 abstract the problem to map almost all existing visualizations.

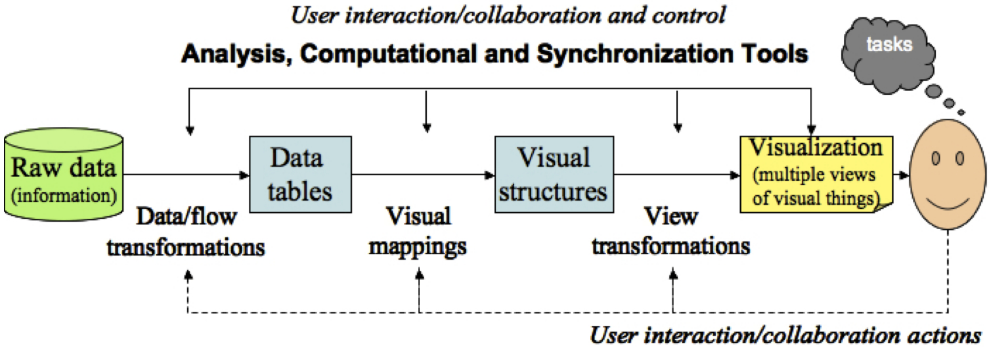

User interaction/collaboration and control
Analysis, Computational and Synchronization Tools
tasks
Raw data (information)
Data tables
Visual structures
Visualization (multiple views of visual things)
Data/flow transformations
Visual mappings
View transformations
User interaction/collaboration actions

Figure 3.1: Visualization pipeline [16, S28/29]

A further step towards the goal is to find common grounds in the sequence, so for example, occurred ideally user interactions at all points in the pipeline or that each link has a many-to-many mapping. So lets focus on the data. The first step should be to transform or filter the data into a usable form for the visualizations application. Furthermore it is important to see if the data is mapped to the visualizations. The next step should be to deal with missing values, errors in input and data which are too large to handle. Missing data may require interpolation. Large data may require sampling, filtering, aggregation or partitioning. After eliminating of all data issues the data is ready to be used. The next step will be to decide what we need them for. An important influence in this question is effectiveness and expressiveness which will be explained below in further detail. The last task in order to reach the goal is to map the data to a image most of the time also interact with a Application Programmers Interface (API). Selecting a viewing parameters, the shading technique (if 3D is chosen) and the device transformations (like displays or printers . . .). This part is dependent on the library which will be used. There are two mathematic measures whose importance rises steadily. Expressiveness and effectiveness can be found in all stages of the pipeline that is given in figure 3.1.

Expressiveness:

An expressive visualization exhibit only the information. A good approach is to define one measure of expressiveness as the ratio M_{exp} of that information, divided by the information that will be presented to the user. The scale for ideal expressiveness is between $0 \leq M_{exp} \leq 1$. If the expressiveness is under 1 ($M_{exp} < 1$) then the content of information is too small and should be filled with more data. The opposite is also possible when the expressiveness is over 1 ($M_{exp} < 1$) this means that we present too much information to the user which can not be handled. Supporting with too much information is not necessary for the user to get the essential information and could lead to the interpretation of this information possibly not being correct. This measure of expressiveness may be extended to fit on other various sets of information [16, S. 131]

Effectiveness:

A visualization is powerful if the information content are accurate , quick to understand and the creation of visualization takes place in a short term (cost-effective manner). As well as for expressiveness it can be defined as a measure of effectiveness M_{eff} as some ratio. The formula is a little more complex in relation to the previous. The measure should include the interpretation as well as the rendering time.Please note that rendering time is irrelevant for smaller data, due to the fact that it is usually very fast. If the time increases, because the complexity or the size of data increases, M_{eff} decreases.The following function is mirroring this:$M_{eff} = \frac{1}{(1+interpret+render)}$ The scale will be the same like in expressiveness : $0 < M_{eff} \leq 1$ The higher M_{eff} is, the greater the visualization's effectiveness will be. If M_{eff} is less it could be for various reasons. Either the interpretation time or the rendering time is higher. If M_{eff} is near 1 both, interpretation as well as rendering time, are very small.[16, S. 132]

4 Visualization Tools and Techniques

This section introduces the technologies which will be used in the application. The most of the selected technologies are mandatory to solve the problem. The possible alternative technologies will be explained at chapter 7.1.

4.1 HTML5 AND CSS3

Hypertext Markup Language 5 (HTML 5) defines the fifth major revision of the principles offer guidance for the design of HTML in the areas of compatibility, utility and interoperability.

HTML5 has some new features like the tag <video> or <audio> which is used for multimedia content. Both tags support the author of an application with an API to write his own script user interface. Among other things, a tag which is called <canvas> exists. It will be used to render dynamic bitmap graphics on the fly, such as graphs or games. Furthermore, it contains a callback system which will inform the user if the browser does not support the <canvas> tag. Cascading Style Sheets 3 (CSS3) is still under construction.Unlike CSS2, CSS3 is divided in modules.These modules have different statuses and stability levels.A selection has already reached the status of W3C recommendations. The world wide web community (W3C) is a international association to expand and develop new web standards.The declared mission of the world wide web community is to lead the web to its full potential.It consists of more than 350 members who work together to define a common web standard. [15]

4.2 AJAX

Asynchronous JavaScript And XML (AJAX) describes a method to avoid the reload of whole websites and the associated waste of time, if only small parts of the websites have changed.In order that the user can continue to work while refreshing the data, the data fetch must be asynchronous. AJAX doesn't change the behaviour of how the browser communicates with the server, but it increases the response time and decrease the data exchange. Another advantage of this method is that the state of the application can be saved on the client side (browser) which saves resource on the server [10]

4.3 Data-Driven Documents (D3)

Data Driven Documents (D3) is a JavaScript library written by Mike Bostock. It allows to bind arbitrary data to a Document Object Model (DOM) and apply data-driven transformations to the document which could also include interactive components. D3 is based on ProtoVis, a graphic toolkit developed by Michael Bostock and Jeff Heer that uses JavaScript and SVG for webnative visualizations. ProtoVis is no longer under active development. [8]

4.4 JavaScript

JavaScript is a script language and contains internal instruction sets, a syntax and a defined structure that is capable to control and manipulate various objects of a web browser.It is a object-based, but not a object-oriented, language which contains a small amount of own objects but mainly created to access numerous objects in Document Object Model (DOM) of the browser. [12]

4.5 JQuery and JQuery UI

JQuery is a powerful, yet easy-to-use JavaScript library that helps web developers and designers add dynamic and interactive elements to their sites, smoothing out browser inconsistencies and greatly reduction of development time. The plugin JQuery UI supports the library with low-level interaction and animation, advanced effects and high-level,themeable widgets to build interactive web applications. [4]

4.6 GWT

Google WebTool Kit (GWT) is a toolkit from the Google family which includes a Java-to-JavaScript Compiler. It offers the possibility to develop efficient and stable crossbrowser applications, which are nearly free from native JavaScript code during the development process. One convenience which GWT offers is the excellent debugging mode. Using one program language, instead of two or more, leads to the huge advantage of using only one debugger. During the development the client side code is indeed Java Code, but during debugging process it behaves like JavaScript in order to prevent later problems by compiling from Java to JavaScript. GWT highly optimizes JavaScript code during the compiling procedure. It removes automatically unused APIs and provides several Javascript versions of the application to make sure that every browser will be supported by the application. This,and other reasons explains why GWT applications are developed 5 times faster then Java2EE applications. Scalability is very important when it comes to question of how many users are expected on the website. Therefore GWT could support stateless server if required. Another big question is how to get the data from the server to the client side. GWT transferring data by taking the advantages of RPC (Remote Procedure Calls). For a detailed description of it, please visit the section Remote Procedure Call. [6]

5 Requirements

This section includes the program requirements. It presents the functional and the non-functional requirements which are necessary for a smooth running of the system. The requirements were defined by the writer of this thesis

5.1 Non-Functional Requirements

N-FR1	System should react in reasonable time
Importance	Essential
Fit Criterion	5 seconds
Description	The application should response in reasonable time. If a calculation needs longer the user should be informed.
N-FR2	Handle steadily growing data
Importance	Essential
Fit Criterion	It should be more then 600 MB free
Description	For the programme to run smoothly ensure that there is sufficient space on the hard drive
N-FR3	Updates may be made only by authorized persons
Importance	Essential
Description	The possibility to update the application from the Vetenskapsrådet should only be accessible to authorized persons
Fit Criterion	Access restriction
N-FR4	Processing times - functions and calculations
Importance	Essential
Fit Criterion	The waiting time depends on the amount of data to calculate
Description	The calculations of the system must be as optimized as much as possible, so that a rapid response can be forwarded to the application
N-FR5	Hours of operation
Importance	Essential
Fit Criterion	Minimum 6 month without any interruptions
Description	The application should be designed so that it can work easily for several months without assistance

Table 5.1: Non - Functional Requirements

5.2 Functional Requirements

FR1	Google maps marker must be clickable
Importance	Essential
Description	The marker of google maps should fire an event to refresh the table and other involved visualizations with an update .
FR2	Mouseover effect for the google maps marker
Importance	Essential
Description	By hoovering with the mouse over a marker, the name of the university must be displayed.
FR3	Legend should change after event fire
Importance	Essential
Description	The legend of google maps should change the range of the accepted money from Vetenskapsrådet if the slider was moved .
FR4	Legend values must change after event
Importance	Essential
Description	The value of the legend should be changing after firing up a slider event.
FR5	The slider should be moveable in his selected range
Importance	Desirable
Description	The slider should move to the left or the right and should hold the distance between the given parameters at the same time.
FR6	The word cloud must react after selecting a word
Importance	Essential
Description	The word cloud must search for projects in the selection of universities and should be able to display the result in the table. Furthermore it should automatically show the table view after the table is ready.
FR7	DataTable must show details in a separate window
Importance	Essential
Description	The table should fire an event after it was selected by the user. This should lead to another pop-up window which contains the details of the selected project.

Table 5.2: Functional Requirements (1)

FR9	Parallel coordinates must have a filter function
Importance	Essential
Description	The parallel coordinates should have a filter function which allow the user to see only a selection of projects. These filter functions should be working on every dimension.
FR10	Bar chart must react after clicking on a bar
Importance	Essential
Description	The bar chart fires an event after the user selects a bar. The Table should display the projects which contains the selected year. After the table is ready the bar chart should change to the table view
FR11	Pie chart must react after a pie part was selected
Importance	Essential
Description	The pie chart should fire an event after the user selects the a section of the pie chart. The Table should display the projects which contains the selected institution. After the table is ready the pie chart should change to the table view
FR12	Administrator menu should be password protected
Importance	Essential
Description	The application should own a administrator menu which is protected with a password and a username.This will ensure that only authorized personnel have access to the update function.
FR12	Create new Administrator accounts
Importance	Essential
Description	The menu should provide a button to create new administrator accounts

Table 5.3: Functional Requirements (2)

5.3 Technical Requirements

TR1	HTTP Server
Importance	Essential
Description	A HTTP Server is needed to show the program. For example apache or lighttpd .
TR2	Application plattform
Importance	Essential
Description	A application plattform is need for a java program. For example glassfish or tomcat .
TR4	Persistent storage
Importance	Essential
Description	Persistent storage of the data. A database have to be running in the background to save the data. For example MySQL or Apache Derby
TR4	JSON Objects
Importance	Desirable
Description	The data objects should be transfered in JSON packages

Table 5.4: Technical Requirements

5.4 Interaction

This section contains the activity diagrams which shows the different processes of the application.

General behaviour

This diagram demonstrates the general behavior of the application. It begins with the slider module which is located on the bottom of the application and the range of years between 2000 to 2020. The user now has the possibility to select the maximum and the minimum range of years which is shown in the application. Note that the more years are selected, the longer the calculation will take. The application offers two possibilities on how to select a single university: The first is to use google maps, which demonstrates all universities of Sweden. The other opportunity is to use the gradient bar chart next to google maps. After the user selected the university of interest, all visualizations which contains the illustrated data, will be updated. During the refresh stage the user will be instructed to wait until the data update is finished. The following visualizations depend on the selection of the slider range and the university. The user can now choose from several visualizations which the application offers: A bar chart, which displays the amount of money of every year. A pie chart which shows the different institutes of the selected university. And also a cumulus cloud which contains the 15 most frequently used words of all project descriptions in the selection range. Another visualization is called parallel coordinate which displays on several dimensions which are connected with a line. Each line represents one project. Excluding the parallel coordinates, all visualizations have interaction buttons which lead to a further selection of the data and this will show in the projects overview. Every single project can be shown in details. If the user wants to have the original data from the beginning of his filtering he has the opportunity to click on the reset button on the top of the web application to get all projects displayed. An overview of the interactive part is given in figure 5.1

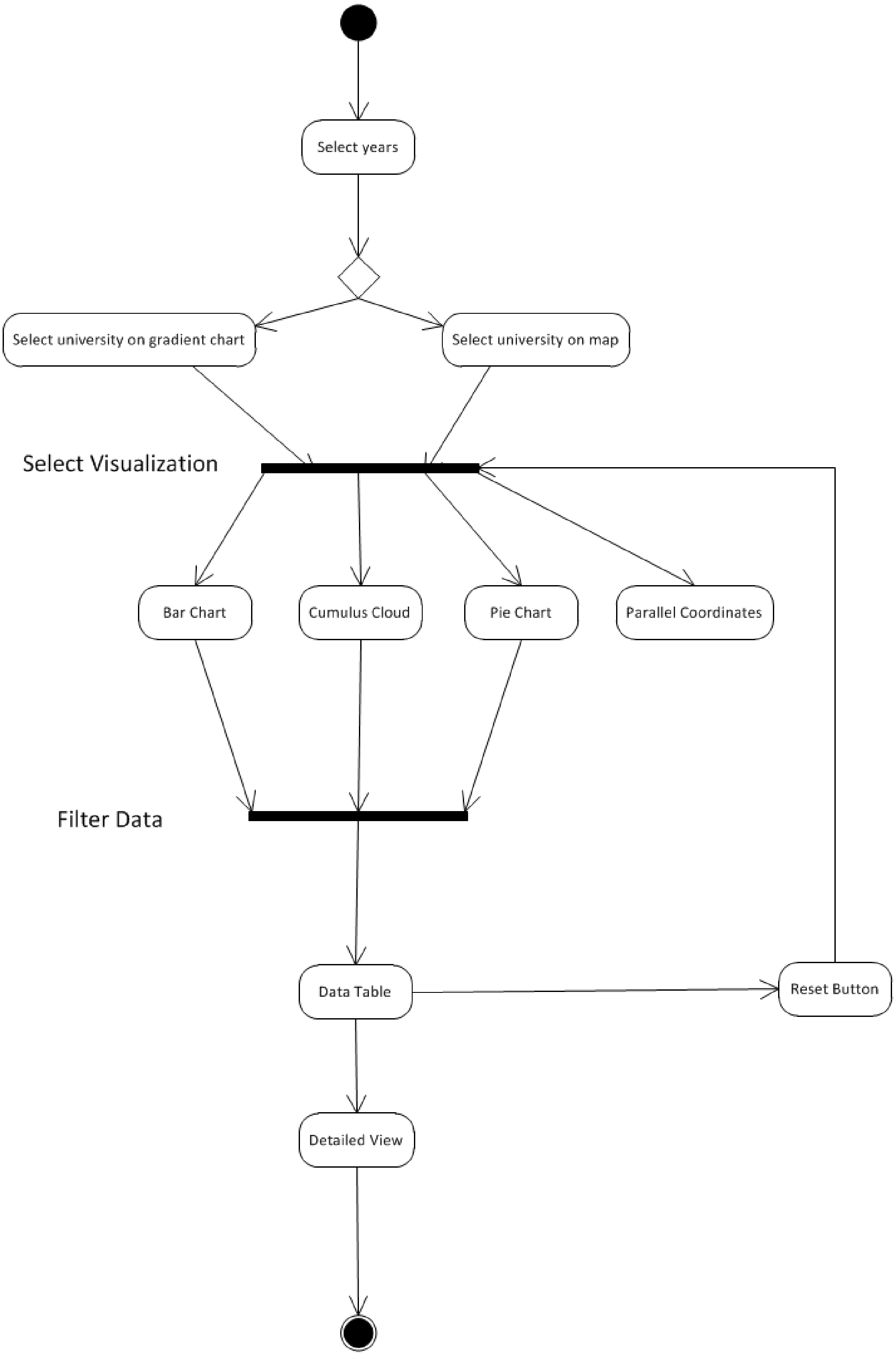


Figure 5.1: Activity diagram with the general functions

Update

This section addresses the process of how to update the introduced web application. When running it the first time, the user has to set a password and a username for security reasons before the procedure can be continued. Only administrators are allowed to enter the update procedure. If this is done the update can be started. The update process will be visualized in figure 5.2

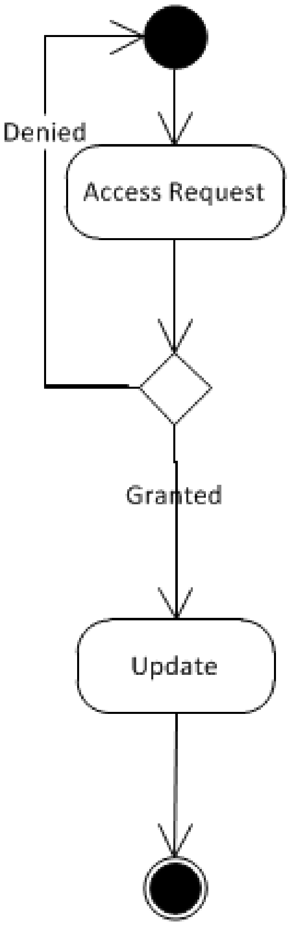


Figure 5.2: Activity diagram for updating the database

User Account

This section contains an activity diagram of how to handle new user account production. With the first run of the application, it is recommended to create a user account, otherwise the access to the update section will be denied.This is justified due to security considerations.After entering the username and the password, the new account will be stored in the database. If at least one user already exists, the application will ask for the user name and password to create a new account. If not, the application gives the opportunity to create a new account. The process of the update sequence will be shown in figure 5.3

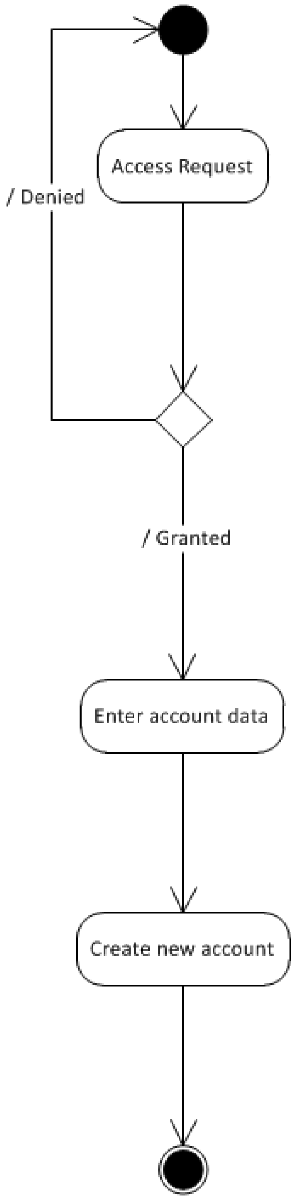


Figure 5.3: Activity diagram of the update procedure

6 Architecture

This chapter describes the structure of the database and gives an introduction to which method was used to structure the program files.

6.1 Database Structure

Since there was no direct access to the Vetenskapsrådet database, it was necessary to create a new database. The database structure is constructed very simply, because no special requirements are needed to store the data. There is one exception regarding the year/money relationship for every project. A project could run several years, the required money will be calculated up to the end of the project. This calculation will take place at the beginning of the project and can be changed after the information changes. The cardinality of that relationship is called 1 : n. This means that one project (1) can have more than one year/money-entry (n).

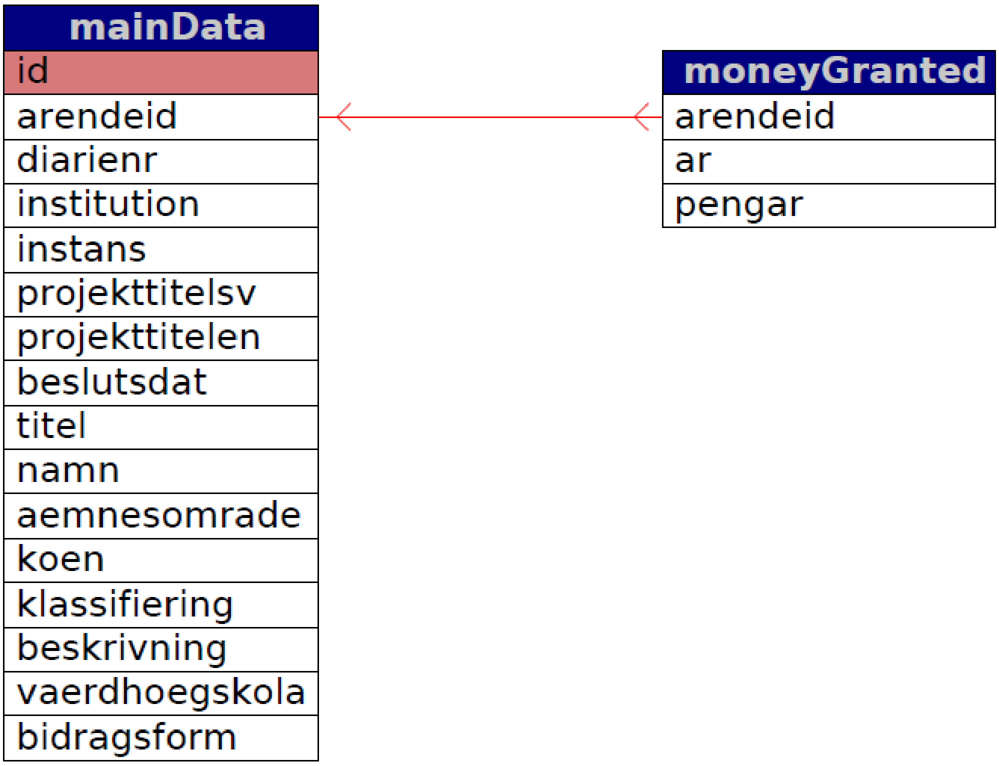


Figure 6.1: The database structure

6.2 Model-View-Controller

This section will discuss the advantage of the applied architectural style which is used for the design of the application. The aim of this architectural style, which is called Model-View-Controller (MVC), is to have a flexible structure and the option for a reusability of code parts. A further reason is the high maintainability which decreases the reaction time for spontaneous changes in the program. Expanding programmes are often getting quite hard to understand, therefore it is advisable to separate the code into different layers. Isolating the functional units leads to a huge advantage due to the fact that members of the application could work separately without the need to know the program code from other members. This simplifies the code and reduces the time for the completion of programs which use this structure. [9]

The MVC Model is implemented on the client side as well as on the server side.The structure part view exists only one time due to the fact that the server side has no visual surface. Note that every MVC structure (Model, View and Controller) has his own subfolders to refine the separation of the subgroup topics.

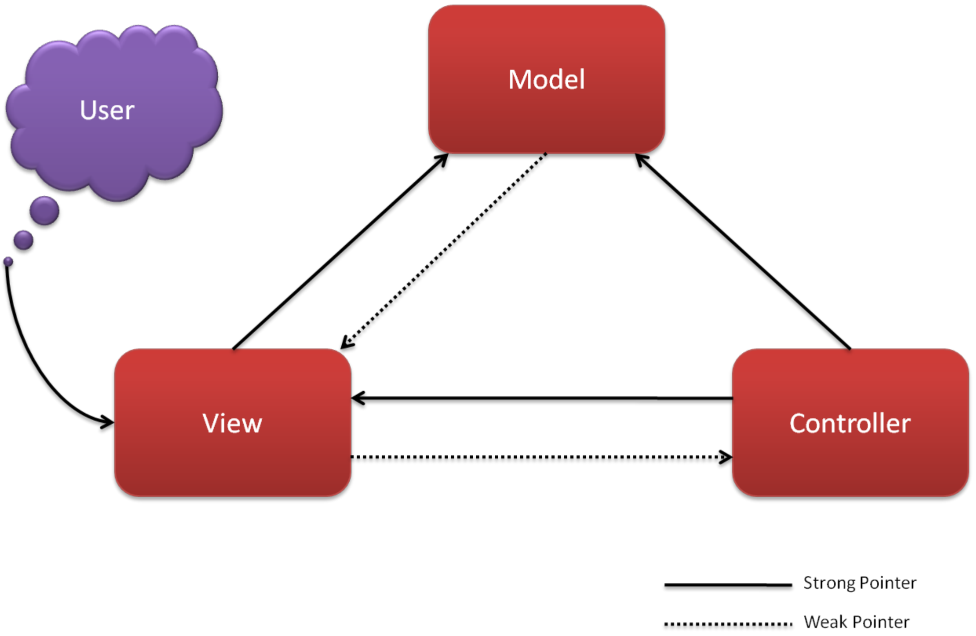


Figure 6.2: Model View Controller [1]

6.2.1 Model

The Model reflects the objects which were needed in the program.

Server Side

The server side contains the configuration class, which is a central place for holding the access data of the used database. The class "SQLRequests" holds the needed SQL queries of this application. The reason to have all database queries at one place, was to have a rapid overview and to avoid query redundancies.

Client Side

Note that the viewport on the client site is realized with serializations which are coming from the GWT library.Serialization converts an object state into a format that can be stored in a file or a memory buffer. The main component of the model classes is called "DataEntry", which holds all the valuable data of the database (The file "DataEntry" is located in the attachment on page 24). The most of the datatypes consists of Strings or Integers except a ArrayList holding the objects of the the model class "MoneyGrantedEntry" (located at the JavaDoc attachment page 34) which is called "beviljat".

6.2.2 View

This section contains the view point of the MVC Model. It is in charge of how the data will be presented to the user. This part is represented only one time in the application due to the fact that the server side does not illustrate data to the user. It contains a PopUp-Window with a waiting symbol, a PopUp-Window with project details, a pie chart which shows institutes of the selected university, a line chart which shows the received money per year of an university, google maps which contains markers with the coordinates of the universities,a slider which adjust the range of selected years and the parallel coordinates, which shows selected project objects and connects it with lines. Note that cumulus cloud, slider and parallel coordinates are implemented with JSNI, this means that the actual work will be done on these external JavaScripts.

6.2.3 Controller

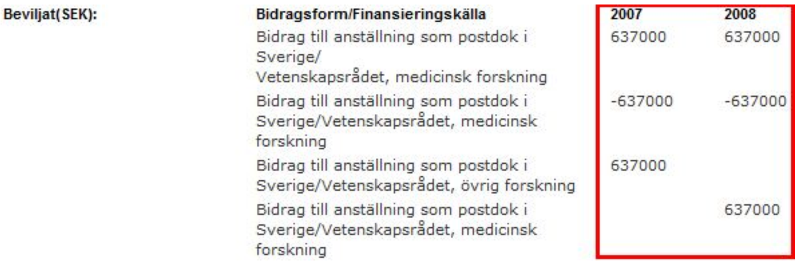
The Controller is a very important link between the view and the model. All the calculations will take place here at the server side as well as on the client side.

Server Side

The Server Side has various subfolders which are divided as followed: charts, database and parse.

Charts The folder charts contains all calculations which are necessary to be executed on the server side.Note that the university objects, which includes the coordinates (latitude and longitude) of each university and the exact search name to find the data, will be created here as an ArrayList of university objects. Another class is the "CountFrequentlyWords" (JavaDoc page 92) which counts the words in every description of each projects which is filtered by the Slider range and the selected university on the map.The class "ParallelCoordinates" (JavaDoc page 95)contains the code to create a comma separated values (csv) file, which is the most practical way to send data to the external javascript of D3's parallel coordinates (for more information of D3 visit the section 4.3)

Parse and Database This folder contains the whole MYSQL communication. The class "MySQL" uses the "DB_Model"(JavaDoc page 112) class in the model to establish a connection to the database which will be done with the class "DB_Access" (JavaDoc page 99). It is a requirement for all further communications between the database and the application. The class "GWT_Update"(JavaDoc page 34) gets the latest data from the database of Vetenskapsrådet and will update the database on the application. This is realized by parsing the HTML twice. At the first time the application will extract all identification numbers (arendeID) of all projects in the database and compares it with the application database. This process will prevent the website as well as the application from unnecessary work to download and erase duplications on the server. On the second step it demands all projects which are not registered on the application database and gets every single project as a HTML content.This will be parsed in the class "Parse_HTML" (JavaDoc page 110)to extract the raw project data. Note that there are some complications with the approved money from Vetenskapsrådet.Due to the fact that a HTML parser receives the data in line shapes, it is not always traceable which year the money belongs to. The picture 6.3 shows the approved money from the swedish council. This procedure will take place in the class "Parse_DB" as well as in "Parse_HTML" (JavaDoc page 108).



Beviljat(SEK):	Bidragsform/Finansieringskälla	2007	2008
	Bidrag till anställning som postdok i Sverige/ Vetenskapsrådet, medicinsk forskning	637000	637000
	Bidrag till anställning som postdok i Sverige/Vetenskapsrådet, medicinsk forskning	-637000	-637000
	Bidrag till anställning som postdok i Sverige/Vetenskapsrådet, övrig forskning	637000	
	Bidrag till anställning som postdok i Sverige/Vetenskapsrådet, medicinsk forskning		637000

Figure 6.3: approved money will be parsed per row [2]

Client Side

The Client side contains a folder "charts" as well as a folder called "database". The folder "charts" contains the class "GWT_Update"(JavaDoc page 34). This updates the various components and refreshes particular parts or the entire content in the application, depending on which requests the user want to have from the application. Furthermore, another important class is called "GWT_Table" (JavaDoc page 7). It contains the structure (AbstractDataModel) of the DataTable visualization which presents the projects depending on which university is currently selected and which range of years are selected.

6.3 Remote Procedure Calls

Remote Procedure Call realizes an interprocess communication between the server and the client side. The technique allows to exchange data. The server side, that gets invoked from the client, is referred to a service which uses a automatically created proxy class to exchange the data. GWT uses serializations of Java Objects to pass the information through RPC. The data will be internally packed in a JSON format. The structure of a RPC will be shown in figure 6.4

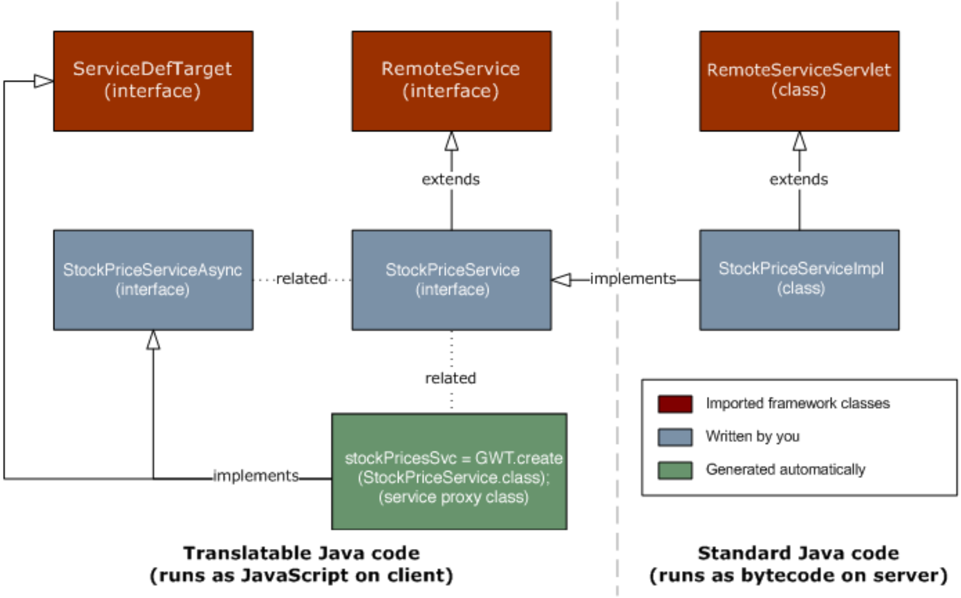

ServiceDefTarget (interface)
RemoteService (interface)
RemoteServiceServlet (class)
extends
extends
StockPriceServiceAsync (interface)
related
StockPriceService (interface)
implements
StockPriceServiceImpl (class)
related
Imported framework classes
Written by you
Generated automatically
implements
stockPricesSvc = GWT.create (StockPriceService.class); (service proxy class)
Translatable Java code (runs as JavaScript on client)
Standard Java code (runs as bytecode on server)

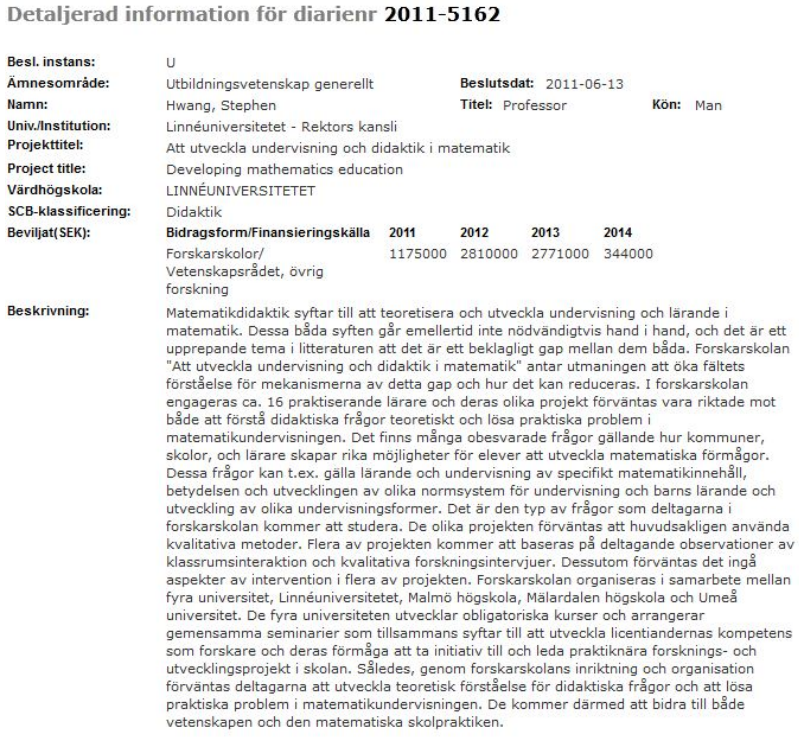
Figure 6.4: Remote Procedure Calls Patterns [7]

7 Development Process

This chapter shows the difficulties and their solutions which came up during the process of the application developing

7.1 Problems and Solutions

A precondition to reach any goal of this thesis was to collect data sets from the Science Research Council of Sweden and add them, filtered and structured, into a new database. It turned out that extracting data was a little more challenging than expected due to the fact that there was no direct access to the database. However, Vetenskapsrådet provides a website form which is written in JavaScript that has access to the database.It was thus possible to query the database using the detour through the form.The form offered the possibility to extract the project data in two steps. At First it needed to get the unique ID of each project (Vetenskapsrådet calls it arendeID) and use this list of IDs to call every project HTML website and extract the data. After this is done, it is necessary, as the next step, to delete the HTML tags to get a line separated file which had the titles of their topics and their contents. The last problem to solve was the conversion of the data.Vetenskapsrådet uses ISO-8859-1 to represent the projects, but the new database is created in UTF-8. Therefore, it was necessary to convert the data during the extraction. After this was done the new database was ready to use. A detailed view of a single project will be shown in figure 7.1



Detaljerad information för diarienr 2011-5162

Besl. instans:	U				
Ämnesområde:	Utbildningsvetenskap generellt	**Beslutsdat:** 2011-06-13			
Namn:	Hwang, Stephen	**Titel:** Professor		**Kön:** Man	
Univ./Institution:	Linnéuniversitetet - Rektors kansli				
Projekttitel:	Att utveckla undervisning och didaktik i matematik				
Project title:	Developing mathematics education				
Värdhögskola:	LINNÉUNIVERSITETET				
SCB-klassificering:	Didaktik				
Beviljat(SEK):	**Bidragsform/Finansieringskälla**	**2011**	**2012**	**2013**	**2014**
	Forskarskolor/ Vetenskapsrådet, övrig forskning	1175000	2810000	2771000	344000

Beskrivning: Matematikdidaktik syftar till att teoretisera och utveckla undervisning och lärande i matematik. Dessa båda syften går emellertid inte nödvändigtvis hand i hand, och det är ett upprepande tema i litteraturen att det är ett beklagligt gap mellan dem båda. Forskarskolan "Att utveckla undervisning och didaktik i matematik" antar utmaningen att öka fältets förståelse för mekanismerna av detta gap och hur det kan reduceras. I forskarskolan engageras ca. 16 praktiserande lärare och deras olika projekt förväntas vara riktade mot både att förstå didaktiska frågor teoretiskt och lösa praktiska problem i matematikundervisningen. Det finns många obesvarade frågor gällande hur kommuner, skolor, och lärare skapar rika möjligheter för elever att utveckla matematiska förmågor. Dessa frågor kan t.ex. gälla lärande och undervisning av specifikt matematikinnehåll, betydelsen och utvecklingen av olika normsystem för undervisning och barns lärande och utveckling av olika undervisningsformer. Det är den typ av frågor som deltagarna i forskarskolan kommer att studera. De olika projekten förväntas att huvudsakligen använda kvalitativa metoder. Flera av projekten kommer att baseras på deltagande observationer av klassrumsinteraktion och kvalitativa forskningsintervjuer. Dessutom förväntas det ingå aspekter av intervention i flera av projekten. Forskarskolan organiseras i samarbete mellan fyra universitet, Linnéuniversitetet, Malmö högskola, Mälardalen högskola och Umeå universitet. De fyra universiteten utvecklar obligatoriska kurser och arrangerar gemensamma seminarier som tillsammans syftar till att utveckla licentiandernas kompetens som forskare och deras förmåga att ta initiativ till och leda praktiknära forsknings- och utvecklingsprojekt i skolan. Således, genom forskarskolans inriktning och organisation förväntas deltagarna att utveckla teoretisk förståelse för didaktiska frågor och att lösa praktiska problem i matematikundervisningen. De kommer därmed att bidra till både vetenskapen och den matematiska skolpraktiken.

Figure 7.1: Details formular of the Swedish Research Council of Sweden [2]

One of the hardest decisions that had to be made in the beginning of a every project was to chose the right programming language. There is a old saying which asserts that programming languages have a lot in common with shoes: some shoes are good looking, but for others they are simply ugly, some shoes fit, others not. The key element is that when it comes to a web application, programming languages operates in similar ways. They all allow you to work with relational databases, all of these offer efficient handling of files and the interaction with the web server. The question of which language should be chosen depends seldom on what a language can do or not, but almost always on the question of which language you can perform a task as easy as possible. The first try of having a possible match of such an application was PHP which is well proven for the server side application work. Out of question was the choice of the programming language on the client side because HTML5 has been using JavaScript to handle the client side programming.Unfortunatly it turns out that PHP isn't that easy to handle and after 2 weeks of trying and using it, it became clear that it needs more than a few days to get into PHP. The next experiment was to use the language phyton for the server side which was easy to handle but in the end it doesn't fit into the solution, that the goal should be to use one language for both sides. Furthermore the question has arisen that the code, which updates the database and is written in Java, should be rewritten into another language or to stay with java which runs on a Java Virtual Machine (JVM). JVM needs a lot of memory on the server side.Due to the lack of time and since Java is faster in application execution time as python, it seemed logical to stay with Java. After the final decision to use Java was made, the second step was to search for a proper way to use it. It appeared ineffective to work with multiple programming languages, so this was the trigger to search the Internet for the best practical way to use it. It turned out that Google offers a Java-to-JavaScript-Compiler which opened up the possibility to write Java Code which is compiled to JavaScript during the deploying.The tool is called Google Web Toolkit (GWT) and managed the balance act to reach the goal to use one language for both sides.GWT interprets the final Java code, during building the war file, into JavaScript (on the client side).GWT has a so called hosted mode which interact with the application without translating the client side code into JavaScript. This means during the development status the application will always work in this mode. The advantage of it is that both sides, client and server side, can be debugged from the Java integrated development environment (IDE). To do this in the traditional "code-test-debug" cycle, hosted mode is by far the most productive way to develop an application quickly.Unfortunately GWT didn't cover all possible cases of JavaScript and also not the supported there libraries. Therefore the Google Developer programmed a lower level script to include JavaScript code into Java code. JavaScript Native Interface (JSNI) is practical if a handwritten JavaScript already exists, which should be included into the application. So this was done with the TagCanvas Cloud (word cloud) and the D3's parallel coordinates. JSNI has strict rules on how handle the script and to call JavaScript functions. One of the restrictions are that a called JSNI function have to be $wnd. on front of all functions to operate.Another limitation is that a JSNI function has to be called in a static way. Note that JSNI has some serious issues, therefore google recommends to use it wisely and moderatly:

> JSNI code is potentially less portable across browsers, more likely to leak memory, less amenable to Java tools, and harder for the compiler to optimize.[17]

7.2 UserInterface

This chapter describes the visualization tools and also the administrator menu which will be found in the application.

7.2.1 Overview

This application is equipped with two main panels.The first is the google maps that displays all universities in Sweden, in which the second includes a Tab Panel, that in turn, contains a representation of parallel coordinates, a tag cloud with the most frequently words, a bar chart with the accepted money per university and year, a pie chart with the institutes of the selected university and also a data table which has attached on every project a detailed view of the selected project.

7.2.2 Map

The google maps is an important element of the introduced application. It displays the universities at there locations. The markers are coloured depending on how much money each university has received from the Swedish Research Council, during the selected time period. The markers have also a mouse over function which displays, on the upper left corner, the name of the university from the current location of the mouse cursor on the google maps.How a google maps with a customized marker looks like will be shown in figure 7.2

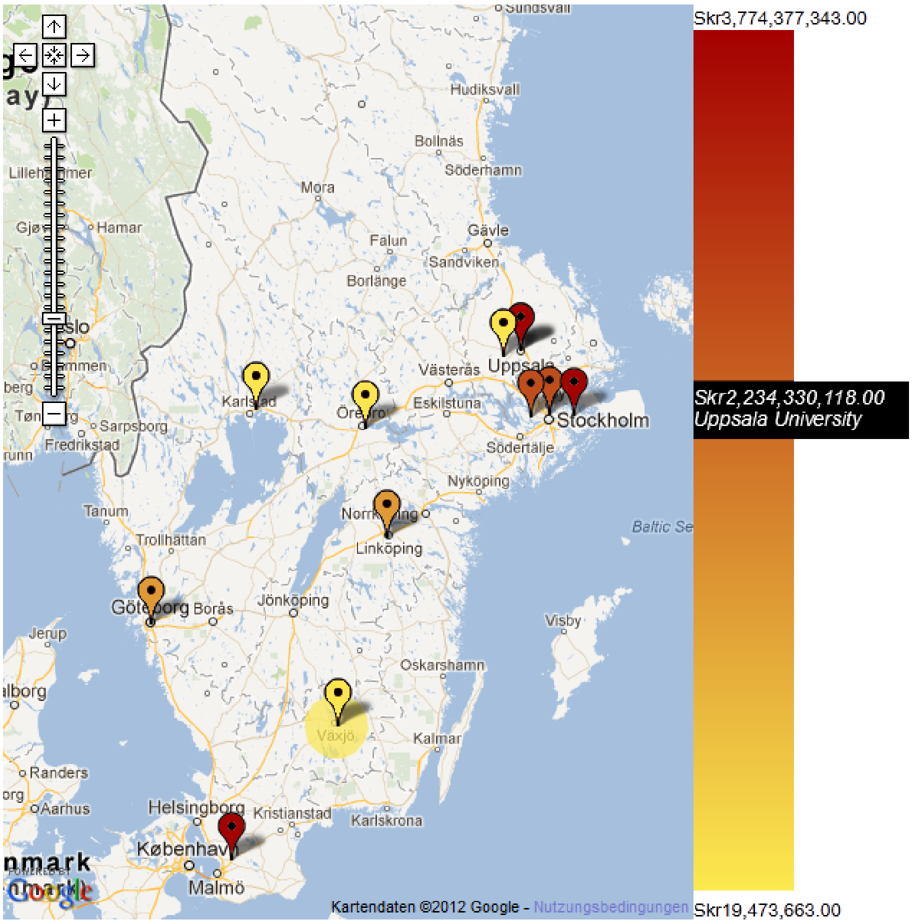
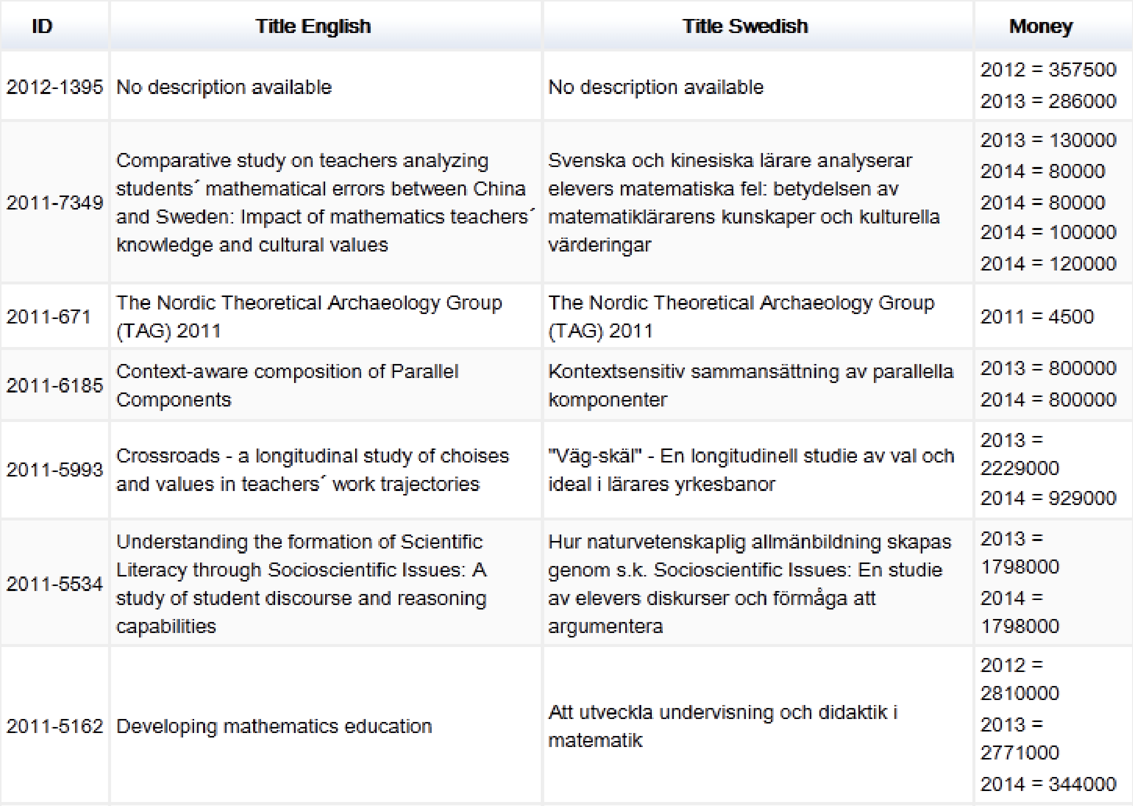


Figure 7.2: Google maps with customized markers

7.2.3 Data Table

The class which is called "GWT_Table"(located at the JavaDoc attachment page 7) will be used as a introduction for the filtered projects.They are filtered in dependencies of the arranged Slider and the current chosen university. Every project is provided with interaction handler. By clicking on a project the listener fires an event that opens a pop-up window which contains a detailed view of the selected project. Note that, due the lack of space, it was not possible to show the full details in the table view. The minimized representation intend to give a clear overview. It contains the identification number, the English and the Swedish title and the authorized money, separated in years. The data will be manipulated from the various filters.An overview of a GWT_Table will be shown in figure 7.3



ID	Title English	Title Swedish	Money
2012-1395	No description available	No description available	2012 = 357500 2013 = 286000
2011-7349	Comparative study on teachers analyzing students´ mathematical errors between China and Sweden: Impact of mathematics teachers´ knowledge and cultural values	Svenska och kinesiska lärare analyserar elevers matematiska fel: betydelsen av matematiklärarens kunskaper och kulturella värderingar	2013 = 130000 2014 = 80000 2014 = 80000 2014 = 100000 2014 = 120000
2011-671	The Nordic Theoretical Archaeology Group (TAG) 2011	The Nordic Theoretical Archaeology Group (TAG) 2011	2011 = 4500
2011-6185	Context-aware composition of Parallel Components	Kontextsensitiv sammansättning av parallella komponenter	2013 = 800000 2014 = 800000
2011-5993	Crossroads - a longitudinal study of choises and values in teachers´ work trajectories	"Väg-skäl" - En longitudinell studie av val och ideal i lärares yrkesbanor	2013 = 2229000 2014 = 929000
2011-5534	Understanding the formation of Scientific Literacy through Socioscientific Issues: A study of student discourse and reasoning capabilities	Hur naturvetenskaplig allmänbildning skapas genom s.k. Socioscientific Issues: En studie av elevers diskurser och förmåga att argumentera	2013 = 1798000 2014 = 1798000
2011-5162	Developing mathematics education	Att utveckla undervisning och didaktik i matematik	2012 = 2810000 2013 = 2771000 2014 = 344000

Figure 7.3: DataTable Overview

7.2.4 Pie Chart

This application contains a pie chart which illustrates every Institution of the selected university which has applied for money from Vetenskapsrådet. The Visualization owns a interactive component, therefore it reacts on mouse clicks on every section of the pie chart and fires an event, which leads to the data table view that, in turn, contains those projects which were initiated from the selected institution. Each pie chart section includes also a tooltip that will be displayed as a mouse over effect, which shows how much projects the institute has applied for money and how much money in general the institute became from the Vetenskapsrådet.In figure 7.4 will be illustrated how a PieChart looks in the application.

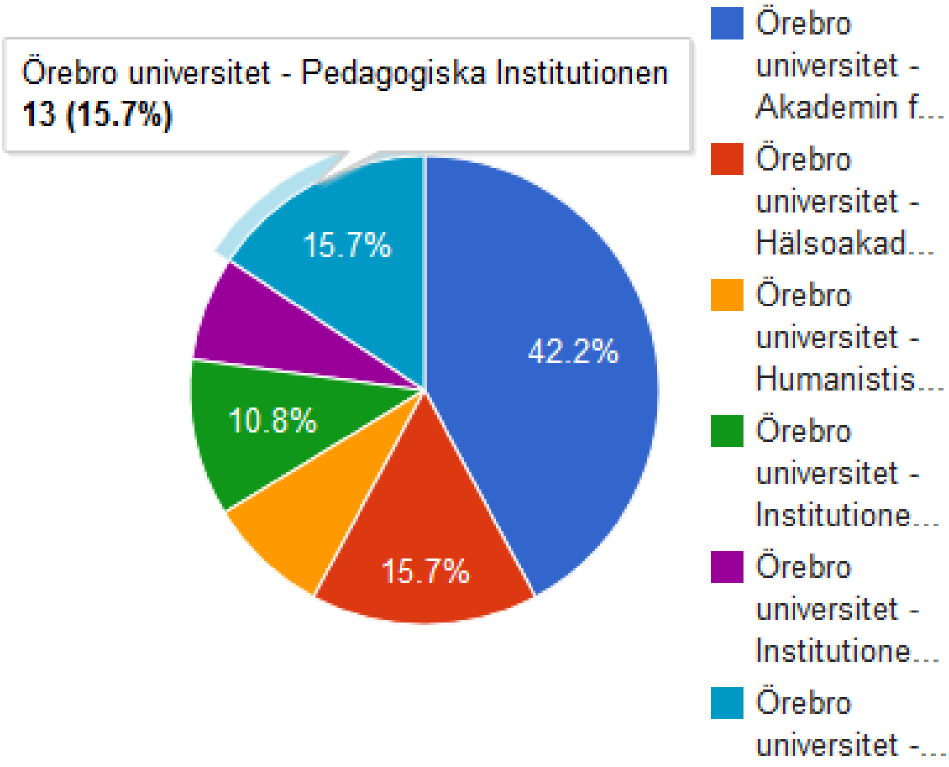


Figure 7.4: PieChart from Institutes of a single University

7.2.5 Parallel Coordinates

The various dimensions are associated with lines which shows the project connection. Every dimension is also interactive, which means, it is possible to emphasize projects by selecting them on each dimension.Until now, there exists 4 dimensions. Each of them represent one attribute of every selected project.The following attributes are illustrated: the amount of money, the duration of the project, the Granted Years for the money and the Gender of the applicant.Please note that it was not possible to use letters instead of numbers. Every full number represent one gender, 1 stands for male, 2 for female and 3 for others.The parallel coordinates will be shown in figure 7.5

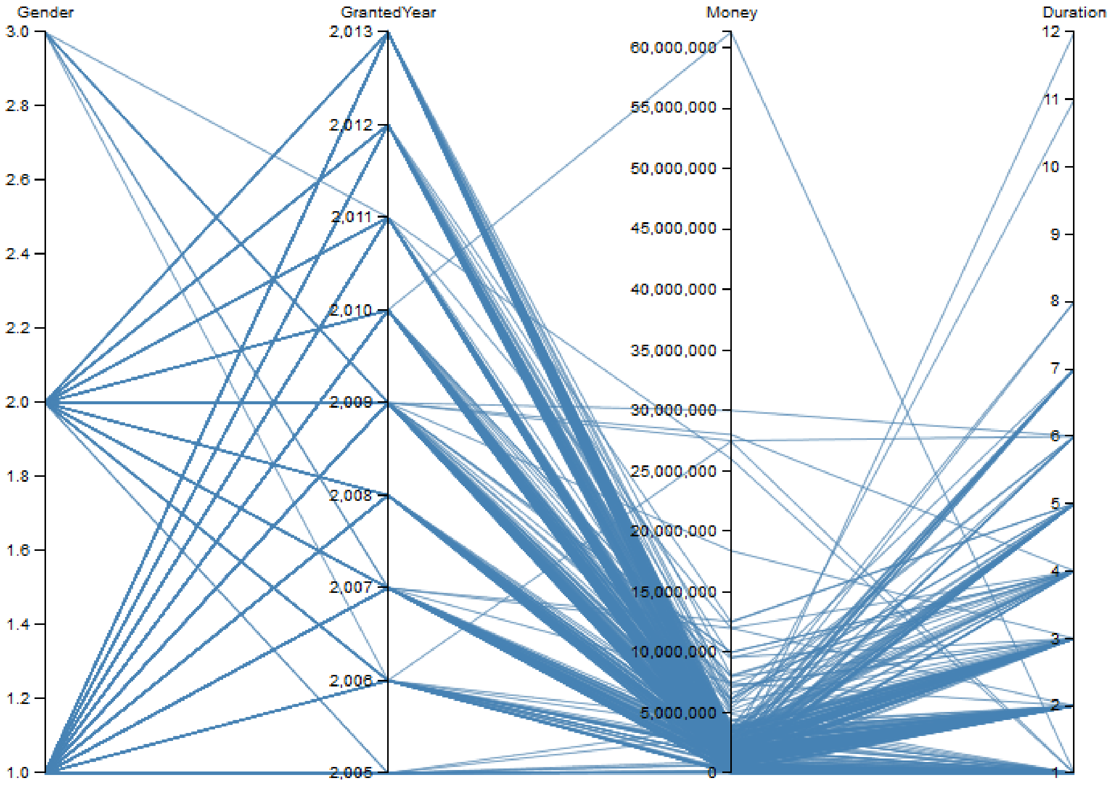


Figure 7.5: Parallel Coordinates of all Projects with there attributes

7.2.6 Bar Chart

The bar chart is used to illustrate to the user how much money each university has been granted, listed in years . By selecting one of the universities at google maps, it will be calculated how much the university has received and those results will be presented in the bar chart visualization. Each of the selectable bars have a click handler implemented, which leads to the table view. The table view will be automatically refreshed with data of the projects from the selected bar. A image of a Bar Chart in the application will be shown in figure 7.6

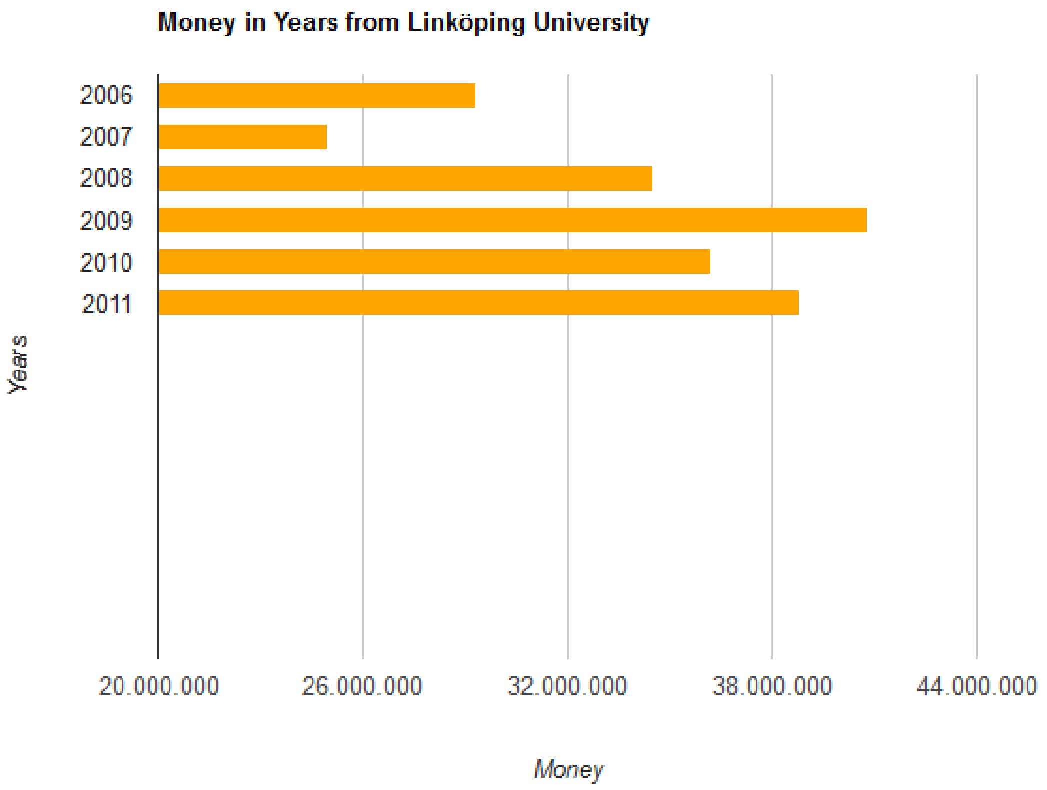


Figure 7.6: Bar chart which shows the agreed money of the selected university divided in years

7.2.7 Cumulus Cloud

The 3D cloud presents the last 15 frequently most used words that were found in the selected project descriptions. Note that without providing a proper list of stop words, the result would contain only : det, den, är, hos,... etc. The selection of projects depends on the chosen university and the range of years on the slider. By clicking on one of the presented words, an event will be fired which calls the project database for common words. After the search is finished, the result will be presented in the table view. An example of the cumulus cloud will be shown in figure 7.7

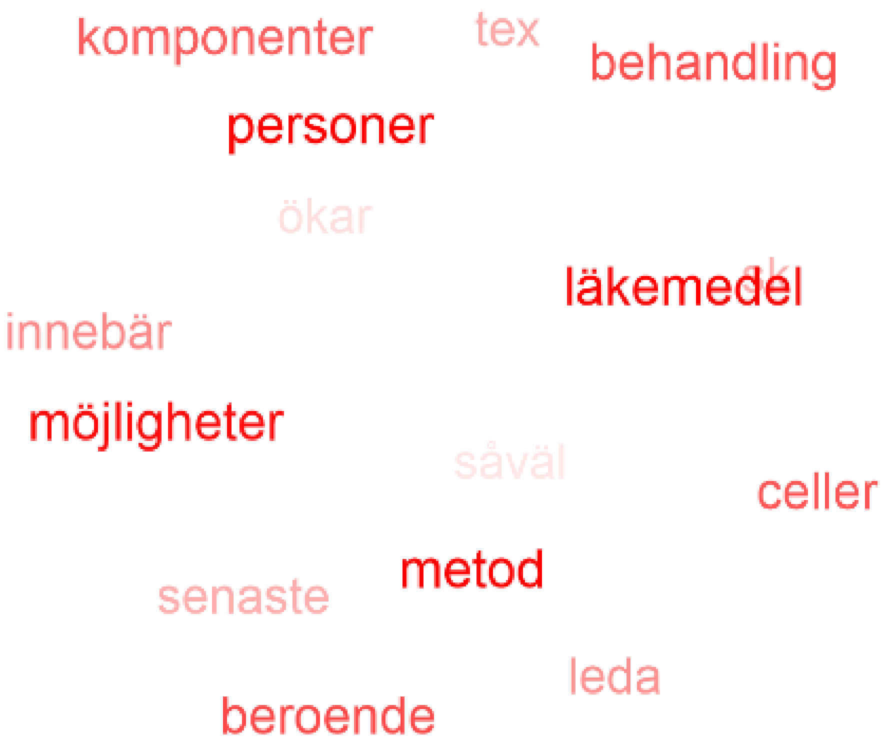


Figure 7.7: The CumulusCloud contains the frequent words of all project descriptions from the selected university

7.2.8 Administrator menu

The menu is located on the left-top of the application. It is named "Administrator" and has two main functions: the first one is to update or create a new table and the second is to create more administrator accounts.Before it is possible to update the application it has to have first a user. This can be done under "user accounts" in the menu. It is only possible to start an update if a user already exists.

8 Conclusion and Future Work

This section contains the impressions and the learning processes as well a solution for the problem which was set at the beginning of this thesis. To create applications, which includes interactions from the user, are always difficult to handle because nobody can foresee how a user will react in a new program environment. It should be simple and comprehensible to the user.

8.1 Conclusion

First goal

The first goal is to create an web application which supports the new web standard HTML5 and CSS3, including the visualized data and a detailed overview of all the projects depended on the universities.

The first big challenge was to write a program which downloads and parses the project data from the Swedish Research Council which was the prerequisite to achieve the first goal of having an application. During the extraction, it turned out that the parsing could lead to problems, due to the inconsistency of some data sets. These are reflected in a small number of projects which sometimes have the same money few times in one year. Since the Money belongs to the projects, there is no need to skip these projects

Second goal

Analysing, filtering data and interaction between the user and the application is mandatory. This means for instance that the user can define a time range to filter the data to see only the projects of a specific university in this time period.

This has been illustrated at section 7.2, due to the fact that every visualization is able to filter the source to the user purpose and reduces the data. This means for instance that the user can define a time range to filter the data to see only the projects of a specific university in this time period and then continue to reduce the data again with clicking by one of the visualizations. This leads to a costumized analysis of data. The user can choose between various visualizations and their filter functions.The user can also reset to the default settings of the selected university by clicking on the reset button which is located on the middle-top of the browser.

Third goal

The last goal aims to update the project database of the Swedish Research Council without duplication (by the first run, the application should be capable of downloading the entire database) This goal is reached. A few problems came up, in how to handle HTML data(agreed money of the projects from Vetenskapsrådet). Due to the fact that a parser parses a document row (read the document horizontal) instead of column (read the document vertical) caused a problem on how to read a HTML table probably. This problem could be solved by assigning the last year of the project duration time to the money that has no relation to a year.

8.2 Future Work

This section contains proposals on how to enhance parts of the application.

8.2.1 Improve Coordinates

D3 from Michael Bostock is a very powerful library which conjures up wonderful visualizations. Unfortunately, to program such a visualization is a very complex and confusing venture due to the nested programming in the library.I tried for more than 1 month to understand how to attach a click listener on the filtered projects but it was impossible. I would recommend to implement such a click listener and then pass the filtered projects to the project tables. That would be a huge improvement for the application.Further improvements would require the transfer of data on the fly to the external JavaScript for creating a parallel coordinates visualization. Until now, the transfer of data to the external Script is realized through a csv file.

8.2.2 Create Database Access Configuration Menu Panel

Until now the access data of a server has to be written into the source code of the application. It would a great improvement if the administrator could have the possibility to change the configurations of the database access dynamically during the run of the application.

8.2.3 Expand the Account Manager

A further enhancement would be a development of a real account manager.Until now there are only the necessary functions implemented.The administrator can create a new account. The next step would be to create functions to delete an account or edit this account. Furthermore it would be nice to restrict access into various levels (role management system)

8.2.4 Save and Load User Settings

The user should have the possibility to save his filter settings and load them again. For this purpose, a user account system and session management should be developed.

Bibliography

[1] Figure: Model-view-controller architecture. *http://techknowfreak.com/wp-content/uploads/Model-View-Controller-Architecture.png*.

[2] Swedish research council of sweden - project details. *http://vrproj.vr.se/detail.asp?arendeid=86593*.

[3] Shan Carter. Figure: Obamas budget proposal 2013. *http://www.nytimes.com/interactive/2012/02/13/us/politics/2013-budget-proposal-graphic.html*, Last Updated February 12, 2012.

[4] Jonathan Chaffer and Karl Swedberg. *jQuery Reference Guide - A Comprehensive Exploration of the popular JavaScript Library*. Packt Publishing, 2007.

[5] Patricia Costigan-Eaves and Michael Macdonald-Ross. Statistical science. *http://projecteuclid.org/euclid.ss/1177012100*, pages 318–326, 1990.

[6] Google Inc. Google web toolkit: Introduction. *https://developers.google.com/web-toolkit/overview*, 2012.

[7] Google Inc. Google web toolkit: Making remote procedure calls. *https://developers.google.com/web-toolkit/doc/latest/tutorial/RPC/*, 2012.

[8] Noah Iliinsky Julie Steele. *Designing Data Visualizations: Representing Informational Relationships*. O'Reilly, 2011.

[9] Glenn E. Krasner and Stephan T. Pope. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. page 34, 1998.

[10] Brett McLaughlin. *Head rush Ajax*. O'Reilly, 2006.

[11] Microsoft Research. Figure: Chronozoom project. *http://www.chronozoomproject.org*, Accessed May,2012.

[12] Ralf Steyer. *Das JavaScript-Handbuch : Einführung, Praxis und Referenz*. Addison-Wesley, 2010.

[13] E.R. Tufte. *The Visual Display of Quanititive Information*. Graphics Press, 1983.

[14] Diego Valle-Jones. Figure: Toll of mexico's drug war. *http://www.diegovalle.net/drug-war-map.html*, Last Updated February 05, 2012.

[15] Anne van Kesteren and Maciej Stachowiak. Html design principles. *http://www.w3.org/TR/html-design-principles/*, Accessed November 26, 2007.

[16] Matthew Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization*. A K Peters Ltd., 2010.

[17] Zundel. Javascript native interface (jsni). *http://code.google.com/p/google-web-toolkit-doc-1-4/wiki/DevGuideJavaScriptNativeInterface*, Accessed February 4, 2010.

Java Documentation

This section contains the java documentation which is generated with the software development environment Eclipse.

Package
se.lnu.client

se.lnu.client
Class Vetenskapenradet

```
java.lang.Object
   |
   +-se.lnu.client.Vetenskapenradet
```

All Implemented Interfaces:
com.google.gwt.core.client.EntryPoint

public class **Vetenskapenradet**
extends java.lang.Object
implements com.google.gwt.core.client.EntryPoint

Constructor Summary

public	Vetenskapenradet()

Method Summary

void	onModuleLoad() This method will load the menu and the visualisations

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.google.gwt.core.client.EntryPoint`

`onModuleLoad`

Constructors

Vetenskapenradet

`public Vetenskapenradet()`

Methods

onModuleLoad

`public void onModuleLoad()`

This method will load the menu and the visualisations

Package se.lnu.client.controller.charts

se.lnu.client.controller.charts
Class GWT_CloudDescription

```
java.lang.Object
  |
  +-se.lnu.client.controller.charts.GWT_CloudDescription
```

public class **GWT_CloudDescription**
extends java.lang.Object

Constructor Summary

public	GWT_CloudDescription()

Method Summary

java.lang.String	getDescription() This method will collect all descriptions from the selected/filtered projects and return everything in a String to calculate the 15 frequently words on the server

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GWT_CloudDescription

public **GWT_CloudDescription**()

Methods

getDescription

public java.lang.String **getDescription**()

This method will collect all descriptions from the selected/filtered projects and return everything in a String to calculate the 15 frequently words on the server

Returns:
String

se.lnu.client.controller.charts
Class GWT_PieChart

```
java.lang.Object
  |
  +-se.lnu.client.controller.charts.GWT_PieChart
```

public class **GWT_PieChart**
extends java.lang.Object

Constructor Summary

public	GWT_PieChart()

Method Summary

java.util.ArrayList	getInstitutionsEntriesFromSelectedUniversity() This method counts the institutions and count the number of projects for each institution

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GWT_PieChart

public **GWT_PieChart**()

Methods

getInstitutionsEntriesFromSelectedUniversity

public java.util.ArrayList **getInstitutionsEntriesFromSelectedUniversity**()

This method counts the institutions and count the number of projects for each institution

Returns:

ArrayList

se.lnu.client.controller.charts
Class GWT_setSize

```
java.lang.Object
  |
  +-se.lnu.client.controller.charts.GWT_setSize
```

public class **GWT_setSize**
extends java.lang.Object

Constructor Summary

public	`GWT_setSize()` This is the constructor of the class

Method Summary

void	`setLeftPanelSize()` This class sets the height and the width for the left panel in depending of the browser window
void	`setRightPanelSize()` This class sets the height and the width for the right panel in depending of the browser window

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GWT_setSize

`public GWT_setSize()`

This is the constructor of the class

Methods

setLeftPanelSize

`public void setLeftPanelSize()`

This class sets the height and the width for the left panel in depending of the browser window

setRightPanelSize

`public void setRightPanelSize()`

This class sets the height and the width for the right panel in depending of the browser window

se.lnu.client.controller.charts
Class GWT_Table

```
java.lang.Object
  |
  +-se.lnu.client.controller.charts.GWT_Table
```

public class **GWT_Table**
extends java.lang.Object

Constructor Summary

public	GWT_Table()

Method Summary

static java.lang.String	beviljatToString(java.util.ArrayList bEntry) This class prepares a String which shows in the table the related money to a year in a single project
com.google.gwt.visualization.client.visualizations.Table	createGWTTable(java.util.ArrayList source) This method creates a GWT table with the options and the abstracted model which contains the data

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

GWT_Table

public **GWT_Table**()

Methods

beviljatToString

public static java.lang.String **beviljatToString**(java.util.ArrayList bEntry)

This class prepares a String which shows in the table the related money to a year in a single project

Parameters:
ArrayList

Returns:
String

createGWTTable

```
public com.google.gwt.visualization.client.visualizations.Table
createGWTTable(java.util.ArrayList source)
```

This method creates a GWT table with the options and the abstracted model which contains the data

Parameters:

`ArrayList`

Returns:

GWT Table

se.lnu.client.controller.charts
Class GWT_Update

```
java.lang.Object
  |
  +-se.lnu.client.controller.charts.GWT_Update
```

public class **GWT_Update**
extends java.lang.Object

Constructor Summary

`public`	`GWT_Update()` This is the constructor of the class.

Method Summary

`static void`	`addPanel(com.google.gwt.user.client.ui.Panel panel, java.lang.String name)` This method adds another panel to the tabpanel where every visualisation is placed
`static void`	`getWordsForTagCloud()` This method gets the 15 frequently words if the return from the server was successful
`static void`	`removePanel(com.google.gwt.user.client.ui.Panel panel)` This method will remove a added panel from the tabpanel
`static void`	`setFocus(com.google.gwt.user.client.ui.Widget focusWidget)` This method selects the tab which contains the given widget
`void`	`start()` This method starts the update and also a popup window which informs the user of the update

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GWT_Update

`public GWT_Update()`

This is the constructor of the class. It attaches a empty tabpanel,calls the range slider and also the google maps card

Methods

(continued from last page)

start

```
public void start()
```

This method starts the update and also a popup window which informs the user of the update

addPanel

```
public static void addPanel(com.google.gwt.user.client.ui.Panel panel,
         java.lang.String name)
```

This method adds another panel to the tabpanel where every visualisation is placed

Parameters:

`panel`

`name`

removePanel

```
public static void removePanel(com.google.gwt.user.client.ui.Panel panel)
```

This method will remove a added panel from the tabpanel

Parameters:

`panel`

setFocus

```
public static void setFocus(com.google.gwt.user.client.ui.Widget focusWidget)
```

This method selects the tab which contains the given widget

Parameters:

`focusWidget`

getWordsForTagCloud

```
public static void getWordsForTagCloud()
```

This method gets the 15 frequently words if the return from the server was successful

Package se.lnu.client.controller.database

se.lnu.client.controller.database
Interface Database_Service

All Known Implementing Classes:

Database_ServiceImpl

public interface **Database_Service**
extends com.google.gwt.user.client.rpc.RemoteService

Method Summary

`java.util.ArrayList`	`getAccessList()`
`java.util.ArrayList`	`getAllUniversitys(java.lang.String currentMinValue, java.lang.String currentMaxValue)`
`java.util.ArrayList`	`getData(java.lang.String uni, java.lang.String minValue, java.lang.String maxValue)`
`java.util.ArrayList`	`getFrequentlyWords(java.lang.String uni)`
`void`	`getMultiCoordinatesData()`
`void`	`setNewUser(GWT_Access newAccess)`
`java.lang.String`	`updateDB()`

Methods

getAllUniversitys

```
public java.util.ArrayList getAllUniversitys(java.lang.String currentMinValue,
          java.lang.String currentMaxValue)
```

getData

```
public java.util.ArrayList getData(java.lang.String uni,
          java.lang.String minValue,
          java.lang.String maxValue)
```

getFrequentlyWords

```
public java.util.ArrayList getFrequentlyWords(java.lang.String uni)
```

getMultiCoordinatesData

```
public void getMultiCoordinatesData()
```

updateDB

```
public java.lang.String updateDB()
```

getAccessList

```
public java.util.ArrayList getAccessList()
```

setNewUser

```
public void setNewUser(GWT_Access newAccess)
```

se.lnu.client.controller.database
Interface Database_ServiceAsync

public interface **Database_ServiceAsync**
extends

Method Summary

void	getAccessList(com.google.gwt.user.client.rpc.AsyncCallback callback)
void	getAllUniversitys(java.lang.String currentMinValue, java.lang.String currentMaxValue, com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)
void	getData(java.lang.String uni, java.lang.String minValue, java.lang.String maxValue, com.google.gwt.user.client.rpc.AsyncCallback async)
void	getFrequentlyWords(java.lang.String uni, com.google.gwt.user.client.rpc.AsyncCallback callback)
void	getMultiCoordinatesData(com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)
void	setNewUser(GWT_Access newAccess, com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)
void	updateDB(com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)

Methods

getAllUniversitys

```
public void getAllUniversitys(java.lang.String currentMinValue,
          java.lang.String currentMaxValue,
          com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)
```

getData

```
public void getData(java.lang.String uni,
          java.lang.String minValue,
          java.lang.String maxValue,
          com.google.gwt.user.client.rpc.AsyncCallback async)
```

(continued from last page)

getFrequentlyWords

```
public void getFrequentlyWords(java.lang.String uni,
         com.google.gwt.user.client.rpc.AsyncCallback callback)
```

getMultiCoordinatesData

```
public void getMultiCoordinatesData(com.google.gwt.user.client.rpc.AsyncCallback
asyncCallback)
```

updateDB

```
public void updateDB(com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)
```

setNewUser

```
public void setNewUser(GWT_Access newAccess,
         com.google.gwt.user.client.rpc.AsyncCallback asyncCallback)
```

getAccessList

```
public void getAccessList(com.google.gwt.user.client.rpc.AsyncCallback callback)
```

Package se.lnu.client.model

se.lnu.client.model
Class GWT_Access

```
java.lang.Object
  |
  +-se.lnu.client.model.GWT_Access
```

All Implemented Interfaces:

com.google.gwt.user.client.rpc.IsSerializable

public class **GWT_Access**
extends java.lang.Object
implements com.google.gwt.user.client.rpc.IsSerializable

Constructor Summary

public	GWT_Access()

Method Summary

java.lang.String	getPassword()
java.lang.String	getUsername()
boolean	isEmpty()
void	setPassword(java.lang.String password)
void	setUsername(java.lang.String username)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GWT_Access

`public GWT_Access()`

Methods

isEmpty

`public boolean isEmpty()`

(continued from last page)

getUsername

```
public java.lang.String getUsername()
```

Returns:

the username

setUsername

```
public void setUsername(java.lang.String username)
```

Parameters:

`username` - the username to set

getPassword

```
public java.lang.String getPassword()
```

Returns:

the password

setPassword

```
public void setPassword(java.lang.String password)
```

Parameters:

`password` - the password to set

se.lnu.client.model
Class GWT_DataStorage

```
java.lang.Object
  |
  +-se.lnu.client.model.GWT_DataStorage
```

public class **GWT_DataStorage**
extends java.lang.Object

Constructor Summary

public	GWT_DataStorage()

Method Summary

static java.util.ArrayList	getCurrentDataEntries()
static double	getCurrentMarkerLat()
static double	getCurrentMarkerLong()
static java.lang.String	getCurrentUniName()
static java.lang.String	getCurrentUniSelected()
static java.util.ArrayList	getDataEntry()
static java.util.ArrayList	getUniversityEntries()
static void	setCurrentDataEntries(java.util.ArrayList currentDataEntries)
static void	setCurrentMarkerLat(double currentMarkerLat)
static void	setCurrentMarkerLong(double currentMarkerLong)
static void	setCurrentUniName(java.lang.String currentUniName)
static void	setCurrentUniSelected(java.lang.String currentUniSelected)
static void	setDataEntry(java.util.ArrayList dataEntry)
static void	setUniversityEntries(java.util.ArrayList universityEntries)

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructors

GWT_DataStorage

`public GWT_DataStorage()`

Methods

getCurrentMarkerLat

`public static double getCurrentMarkerLat()`

Returns:
the currentMarkerLat

getCurrentMarkerLong

`public static double getCurrentMarkerLong()`

Returns:
the currentMarkerLong

setCurrentMarkerLong

`public static void setCurrentMarkerLong(double currentMarkerLong)`

Parameters:
`currentMarkerLong` - the currentMarkerLong to set

setCurrentMarkerLat

`public static void setCurrentMarkerLat(double currentMarkerLat)`

Parameters:
`currentMarkerLat` - the currentMarkerLat to set

getCurrentDataEntries

`public static java.util.ArrayList getCurrentDataEntries()`

(continued from last page)

Returns:
the currentDataEntries

setCurrentDataEntries

```
public static void setCurrentDataEntries(java.util.ArrayList currentDataEntries)
```

Parameters:
`currentDataEntries` - the currentDataEntries to set

getCurrentUniName

```
public static java.lang.String getCurrentUniName()
```

Returns:
the currentUniName

getCurrentUniSelected

```
public static java.lang.String getCurrentUniSelected()
```

Returns:
the currentUniSelected

getDataEntry

```
public static java.util.ArrayList getDataEntry()
```

Returns:
the resultt

getUniversityEntries

```
public static java.util.ArrayList getUniversityEntries()
```

Returns:
the universityLatLng

setCurrentUniName

```
public static void setCurrentUniName(java.lang.String currentUniName)
```

Parameters:
`currentUniName` - the currentUniName to set

(continued from last page)

setCurrentUniSelected

public static void **setCurrentUniSelected**(java.lang.String currentUniSelected)

Parameters:

`currentUniSelected` - the currentUniSelected to set

setDataEntry

public static void **setDataEntry**(java.util.ArrayList dataEntry)

Parameters:

`resultt` - the resultt to set

setUniversityEntries

public static void **setUniversityEntries**(java.util.ArrayList universityEntries)

Parameters:

`universityLatLng` - the universityLatLng to set

Package se.lnu.client.model.charts

se.lnu.client.model.charts
Class DataEntry

```
java.lang.Object
  |
  +-se.lnu.client.model.charts.DataEntry
```

All Implemented Interfaces:
com.google.gwt.user.client.rpc.IsSerializable

public class **DataEntry**
extends java.lang.Object
implements com.google.gwt.user.client.rpc.IsSerializable

Constructor Summary

public	DataEntry()

Method Summary

static java.lang.String	beviljatToString(java.util.ArrayList bEntry) this method is only a helper for the method toString() which is used for console outputs
java.lang.String	getAemnesomrade()
java.lang.String	getArendeid()
java.lang.String	getBeskrivning()
java.lang.String	getBeslutsdat()
java.util.ArrayList	getBeviljat()
java.lang.String	getBidragsform()
java.lang.String	getDiarienr()
java.lang.String	getId()
java.lang.String	getInstans()
java.lang.String	getInstitution()
java.lang.String	getKlassifiering()
java.lang.String	getKoen()

`java.lang.String`	`getNamn()`
`java.lang.String`	`getProjekttitelen()`
`java.lang.String`	`getProjekttitelsv()`
`java.lang.String`	`getTitel()`
`java.lang.String`	`getVaerdhoegskola()`
`void`	`setAemnesomrade(java.lang.String aemnesomrade)`
`void`	`setArendeid(java.lang.String arendeid)`
`void`	`setBeskrivning(java.lang.String beskrivning)`
`void`	`setBeslutsdat(java.lang.String beslutsdat)`
`void`	`setBeviljat(java.util.ArrayList beviljat)`
`void`	`setBidragsform(java.lang.String bidragsform)`
`void`	`setDiarienr(java.lang.String diarienr)`
`void`	`setId(java.lang.String id)`
`void`	`setInstans(java.lang.String instans)`
`void`	`setInstitution(java.lang.String institution)`
`void`	`setKlassifiering(java.lang.String klassifiering)`
`void`	`setKoen(java.lang.String koen)`
`void`	`setNamn(java.lang.String namn)`
`void`	`setProjekttitelen(java.lang.String projekttitelen)`
`void`	`setProjekttitelsv(java.lang.String projekttitelsv)`
`void`	`setTitel(java.lang.String titel)`
`void`	`setVaerdhoegskola(java.lang.String vaerdhoegskola)`
`java.lang.String`	`toString()` this method is only for console output

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructors

DataEntry

```
public DataEntry()
```

Methods

toString

```
public java.lang.String toString()
```

this method is only for console output

beviljatToString

```
public static java.lang.String beviljatToString(java.util.ArrayList bEntry)
```

this method is only a helper for the method toString() which is used for console outputs

getId

```
public java.lang.String getId()
```

Returns:
the id

setId

```
public void setId(java.lang.String id)
```

Parameters:
`id` - the id to set

getArendeid

```
public java.lang.String getArendeid()
```

Returns:
the arendeid

(continued from last page)

setArendeid

```
public void setArendeid(java.lang.String arendeid)
```

Parameters:

`arendeid` - the arendeid to set

getDiarienr

```
public java.lang.String getDiarienr()
```

Returns:

the diarienr

setDiarienr

```
public void setDiarienr(java.lang.String diarienr)
```

Parameters:

`diarienr` - the diarienr to set

getInstitution

```
public java.lang.String getInstitution()
```

Returns:

the institution

setInstitution

```
public void setInstitution(java.lang.String institution)
```

Parameters:

`institution` - the institution to set

getInstans

```
public java.lang.String getInstans()
```

Returns:

the instans

setInstans

```
public void setInstans(java.lang.String instans)
```

(continued from last page)

Parameters:

`instans` - the instans to set

getProjekttitelsv

```
public java.lang.String getProjekttitelsv()
```

Returns:

the projekttitelsv

setProjekttitelsv

```
public void setProjekttitelsv(java.lang.String projekttitelsv)
```

Parameters:

`projekttitelsv` - the projekttitelsv to set

getProjekttitelen

```
public java.lang.String getProjekttitelen()
```

Returns:

the projekttitelen

setProjekttitelen

```
public void setProjekttitelen(java.lang.String projekttitelen)
```

Parameters:

`projekttitelen` - the projekttitelen to set

getBeslutsdat

```
public java.lang.String getBeslutsdat()
```

Returns:

the beslutsdat

setBeslutsdat

```
public void setBeslutsdat(java.lang.String beslutsdat)
```

Parameters:

`beslutsdat` - the beslutsdat to set

getTitel

```
public java.lang.String getTitel()
```

Returns:
the titel

setTitel

```
public void setTitel(java.lang.String titel)
```

Parameters:
`titel` - the titel to set

getNamn

```
public java.lang.String getNamn()
```

Returns:
the namn

setNamn

```
public void setNamn(java.lang.String namn)
```

Parameters:
`namn` - the namn to set

getAemnesomrade

```
public java.lang.String getAemnesomrade()
```

Returns:
the aemnesomrade

setAemnesomrade

```
public void setAemnesomrade(java.lang.String aemnesomrade)
```

Parameters:
`aemnesomrade` - the aemnesomrade to set

getKoen

```
public java.lang.String getKoen()
```

(continued from last page)

Returns:
the koen

setKoen

```
public void setKoen(java.lang.String koen)
```

Parameters:
`koen` - the koen to set

getKlassifiering

```
public java.lang.String getKlassifiering()
```

Returns:
the klassifiering

setKlassifiering

```
public void setKlassifiering(java.lang.String klassifiering)
```

Parameters:
`klassifiering` - the klassifiering to set

getBeskrivning

```
public java.lang.String getBeskrivning()
```

Returns:
the beskrivning

setBeskrivning

```
public void setBeskrivning(java.lang.String beskrivning)
```

Parameters:
`beskrivning` - the beskrivning to set

getVaerdhoegskola

```
public java.lang.String getVaerdhoegskola()
```

Returns:
the vaerdhoegskola

setVaerdhoegskola

```
public void setVaerdhoegskola(java.lang.String vaerdhoegskola)
```

Parameters:

`vaerdhoegskola` - the vaerdhoegskola to set

getBidragsform

```
public java.lang.String getBidragsform()
```

Returns:

the bidragsform

setBidragsform

```
public void setBidragsform(java.lang.String bidragsform)
```

Parameters:

`bidragsform` - the bidragsform to set

getBeviljat

```
public java.util.ArrayList getBeviljat()
```

Returns:

the beviljat

setBeviljat

```
public void setBeviljat(java.util.ArrayList beviljat)
```

Parameters:

`beviljat` - the beviljat to set

se.lnu.client.model.charts
Class InstitutionEntry

```
java.lang.Object
   |
   +-se.lnu.client.model.charts.InstitutionEntry
```

All Implemented Interfaces:

com.google.gwt.user.client.rpc.IsSerializable

---

public class **InstitutionEntry**
extends java.lang.Object
implements com.google.gwt.user.client.rpc.IsSerializable

Constructor Summary

public	InstitutionEntry()

Method Summary

java.lang.String	getInstitution()
int	getNumber()
void	setInstitution(java.lang.String institution)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

Constructors

InstitutionEntry

`public InstitutionEntry()`

Methods

getInstitution

`public java.lang.String getInstitution()`

Returns:

the institution

getNumber

```
public int getNumber()
```

Returns:

the number

setInstitution

```
public void setInstitution(java.lang.String institution)
```

Parameters:

`institution` - to set

se.lnu.client.model.charts
Class MoneyGrantedEntry

```
java.lang.Object
  |
  +-se.lnu.client.model.charts.MoneyGrantedEntry
```

All Implemented Interfaces:
com.google.gwt.user.client.rpc.IsSerializable

public class **MoneyGrantedEntry**
extends java.lang.Object
implements com.google.gwt.user.client.rpc.IsSerializable

Constructor Summary

public	MoneyGrantedEntry()

Method Summary

int	getAr()
long	getPengar()
java.lang.String	getUniName()
boolean	isEmpty()
void	setAr(int ar)
void	setPengar(long pengar)
void	setUniName(java.lang.String uniName)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MoneyGrantedEntry

public **MoneyGrantedEntry**()

(continued from last page)

Methods

getAr

`public int **getAr**()`

Returns:
the ar

getPengar

`public long **getPengar**()`

Returns:
the pengar

getUniName

`public java.lang.String **getUniName**()`

Returns:
the arendeid

isEmpty

`public boolean **isEmpty**()`

setAr

`public void **setAr**(int ar)`

Parameters:
`ar` - the ar to set

setPengar

`public void **setPengar**(long pengar)`

Parameters:
`pengar` - the pengar to set

setUniName

`public void **setUniName**(java.lang.String uniName)`

(continued from last page)

Parameters:

`arendeid` - the arendeid to set

se.lnu.client.model.charts
Class UniversityEntries

```
java.lang.Object
  |
  +-se.lnu.client.model.charts.UniversityEntries
```

All Implemented Interfaces:
com.google.gwt.user.client.rpc.IsSerializable

public class **UniversityEntries**
extends java.lang.Object
implements com.google.gwt.user.client.rpc.IsSerializable

Constructor Summary

public	UniversityEntries()

Method Summary

java.lang.String	getDbSearchName()
double	getLat()
double	getLng()
java.util.ArrayList	getMoneyPerYear()
long	getSumPengar()
java.lang.String	getUniName()
void	setDbSearchName(java.lang.String dbSearchName)
void	setLat(double lat)
void	setLng(double lng)
void	setMoneyPerYear(java.util.ArrayList moneyPerYear)
void	setSumPengar(long l)
void	setUniName(java.lang.String uniName)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

UniversityEntries

```
public UniversityEntries()
```

Methods

getDbSearchName

```
public java.lang.String getDbSearchName()
```

Returns:
the dbSearchName

getLat

```
public double getLat()
```

Returns:
the lat

getLng

```
public double getLng()
```

Returns:
the lng

getMoneyPerYear

```
public java.util.ArrayList getMoneyPerYear()
```

Returns:
the moneyPerYear

getSumPengar

```
public long getSumPengar()
```

Returns:
the sumPengar

getUniName

```
public java.lang.String getUniName()
```

setDbSearchName

```
public void setDbSearchName(java.lang.String dbSearchName)
```

Parameters:
`dbSearchName` - the dbSearchName to set

setLat

```
public void setLat(double lat)
```

Parameters:
`lat` - the lat to set

setLng

```
public void setLng(double lng)
```

Parameters:
`lng` - the lng to set

setMoneyPerYear

```
public void setMoneyPerYear(java.util.ArrayList moneyPerYear)
```

Parameters:
`moneyPerYear` - the moneyPerYear to set

setSumPengar

```
public void setSumPengar(long l)
```

Parameters:
`l` - the sumPengar to set

setUniName

```
public void setUniName(java.lang.String uniName)
```

Package se.lnu.client.model.window

se.lnu.client.model.window
Class GWT_WindowConfig

```
java.lang.Object
   |
   +-se.lnu.client.model.window.GWT_WindowConfig
```

public class **GWT_WindowConfig**
extends java.lang.Object

Constructor Summary

public	GWT_WindowConfig()

Method Summary

static int	getCurrentWindowsHeight()
static int	getCurrentWindowsWidth()
static java.lang.String	getDarkOrangeMarker()
static java.lang.String	getDivIdLeftPanel()
static java.lang.String	getDivIdMultiCoordinates()
static java.lang.String	getDivIdPieChart()
static java.lang.String	getDivIdRightPanel()
static java.lang.String	getDivIdSliderPanel()
static int	getGoogleMapsHeight()
static int	getGoogleMapsWidth()
static int	getLeftPanelHeight()
static int	getLeftPanelWidth()
static java.lang.String	getOrangeMarker()
static java.lang.String	getRedMarker()
static int	getRightPanelHeight()

static int	getRightPanelWidth()
static int	getVisualizationHeight()
static int	getVisualizationWidth()
static java.lang.String	getYellowMarker()
static void	setCurrentWindowsHeight(int currentWindowsHeight)
static void	setCurrentWindowsWidth(int currentWindowsWidth)
static void	setDivIdLeftPanel(java.lang.String divIdLeftPanel)
static void	setDivIdMultiCoordinates(java.lang.String divIdMultiCoordinates)
static void	setDivIdPieChart(java.lang.String divIdPieChart)
static void	setDivIdRightPanel(java.lang.String divIdRightPanel)
static void	setDivIdSliderPanel(java.lang.String divIdSliderPanel)
static void	setGoogleMapsHeight(int googleMapsHeight)
static void	setGoogleMapsWidth(int googleMapsWidth)
static void	setLeftPanelHeight(int leftPanelHeight)
static void	setLeftPanelWidth(int leftPanelWidth)
static void	setRightPanelHeight(int rightPanelHeight)
static void	setRightPanelWidth(int rightPanelWidth)
static void	setVisualizationHeight(int visualizationHeight)
static void	setVisualizationWidth(int visualizationWidth)

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

GWT_WindowConfig

```
public GWT_WindowConfig()
```

Methods

getYellowMarker

```
public static java.lang.String getYellowMarker()
```

Returns:
the yellowMarker

getOrangeMarker

```
public static java.lang.String getOrangeMarker()
```

Returns:
the orangeMarker

getDarkOrangeMarker

```
public static java.lang.String getDarkOrangeMarker()
```

Returns:
the darkOrangeMarker

getRedMarker

```
public static java.lang.String getRedMarker()
```

Returns:
the redMarker

getCurrentWindowsHeight

```
public static int getCurrentWindowsHeight()
```

Returns:
the currentWindowsHeight

setCurrentWindowsHeight

```
public static void setCurrentWindowsHeight(int currentWindowsHeight)
```

(continued from last page)

Parameters:

`currentWindowsHeight` - the currentWindowsHeight to set

getCurrentWindowsWidth

```
public static int getCurrentWindowsWidth()
```

Returns:

the currentWindowsWidth

setCurrentWindowsWidth

```
public static void setCurrentWindowsWidth(int currentWindowsWidth)
```

Parameters:

`currentWindowsWidth` - the currentWindowsWidth to set

getDivIdSliderPanel

```
public static java.lang.String getDivIdSliderPanel()
```

Returns:

the divIdSliderPanel

setDivIdSliderPanel

```
public static void setDivIdSliderPanel(java.lang.String divIdSliderPanel)
```

Parameters:

`divIdSliderPanel` - the divIdSliderPanel to set

getLeftPanelHeight

```
public static int getLeftPanelHeight()
```

Returns:

the leftPanelHeight

setLeftPanelHeight

```
public static void setLeftPanelHeight(int leftPanelHeight)
```

Parameters:

`leftPanelHeight` - the leftPanelHeight to set

getLeftPanelWidth

```
public static int getLeftPanelWidth()
```

Returns:

the leftPanelWidth

setLeftPanelWidth

```
public static void setLeftPanelWidth(int leftPanelWidth)
```

Parameters:

`leftPanelWidth` - the leftPanelWidth to set

getDivIdLeftPanel

```
public static java.lang.String getDivIdLeftPanel()
```

Returns:

the divIdLeftPanel

setDivIdLeftPanel

```
public static void setDivIdLeftPanel(java.lang.String divIdLeftPanel)
```

Parameters:

`divIdLeftPanel` - the divIdLeftPanel to set

getGoogleMapsHeight

```
public static int getGoogleMapsHeight()
```

Returns:

the googleMapsHeight

setGoogleMapsHeight

```
public static void setGoogleMapsHeight(int googleMapsHeight)
```

Parameters:

`googleMapsHeight` - the googleMapsHeight to set

getGoogleMapsWidth

```
public static int getGoogleMapsWidth()
```

(continued from last page)

Returns:
the googleMapsWidth

setGoogleMapsWidth

```
public static void setGoogleMapsWidth(int googleMapsWidth)
```

Parameters:
`googleMapsWidth` - the googleMapsWidth to set

getRightPanelHeight

```
public static int getRightPanelHeight()
```

Returns:
the rightPanelHeight

setRightPanelHeight

```
public static void setRightPanelHeight(int rightPanelHeight)
```

Parameters:
`rightPanelHeight` - the rightPanelHeight to set

getRightPanelWidth

```
public static int getRightPanelWidth()
```

Returns:
the rightPanelWidth

setRightPanelWidth

```
public static void setRightPanelWidth(int rightPanelWidth)
```

Parameters:
`rightPanelWidth` - the rightPanelWidth to set

getDivIdRightPanel

```
public static java.lang.String getDivIdRightPanel()
```

Returns:
the divIdRightPanel

setDivIdRightPanel

```
public static void setDivIdRightPanel(java.lang.String divIdRightPanel)
```

Parameters:

`divIdRightPanel` - the divIdRightPanel to set

getVisualizationWidth

```
public static int getVisualizationWidth()
```

Returns:

the visualizationWidth

setVisualizationWidth

```
public static void setVisualizationWidth(int visualizationWidth)
```

Parameters:

`visualizationWidth` - the visualizationWidth to set

getVisualizationHeight

```
public static int getVisualizationHeight()
```

Returns:

the visualizationHeight

setVisualizationHeight

```
public static void setVisualizationHeight(int visualizationHeight)
```

Parameters:

`visualizationHeight` - the visualizationHeight to set

getDivIdMultiCoordinates

```
public static java.lang.String getDivIdMultiCoordinates()
```

Returns:

the divIdMultiCoordinates

setDivIdMultiCoordinates

```
public static void setDivIdMultiCoordinates(java.lang.String divIdMultiCoordinates)
```

(continued from last page)

Parameters:

`divIdMultiCoordinates` - the divIdMultiCoordinates to set

getDivIdPieChart

```
public static java.lang.String getDivIdPieChart()
```

Returns:

the divIdPieChart

setDivIdPieChart

```
public static void setDivIdPieChart(java.lang.String divIdPieChart)
```

Parameters:

`divIdPieChart` - the divIdPieChart to set

Package
se.lnu.client.view

se.lnu.client.view
Class VR_SliderRange

```
java.lang.Object
  |
  +-se.lnu.client.view.VR_SliderRange
```

public class **VR_SliderRange**
extends java.lang.Object

Constructor Summary

public	VR_SliderRange()

Method Summary

static java.lang.String	getCurrentMaxValue() This method returns the current maximum value of the range slider
static java.lang.String	getCurrentMinValue() This method returns the current minimum value of the range slider
static native void	slider() This method is a JSNI method , it calls the slider from the JQuery-UI library
static void	updateData(java.lang.String nothing) This method call the Update function and refreshs the visualization after the ranges of the years on the slider have changed

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

VR_SliderRange

```
public VR_SliderRange()
```

Methods

getCurrentMaxValue

```
public static java.lang.String getCurrentMaxValue()
```

This method returns the current maximum value of the range slider

Returns:

(continued from last page)

String max Value Slider

getCurrentMinValue

```
public static java.lang.String getCurrentMinValue()
```

This method returns the current minimum value of the range slider

Returns:
String min Value Slider

slider

```
public static native void slider()
```

This method is a JSNI method , it calls the slider from the JQuery-UI library

updateData

```
public static void updateData(java.lang.String nothing)
```

This method call the Update function and refreshs the visualization after the ranges of the years on the slider have changed

se.lnu.client.view
Class VR_WaitPopUp

```
java.lang.Object
   |
   +-com.google.gwt.user.client.ui.UIObject
      |
      +-com.google.gwt.user.client.ui.Widget
         |
         +-com.google.gwt.user.client.ui.Panel
            |
            +-com.google.gwt.user.client.ui.SimplePanel
               |
               +-com.google.gwt.user.client.ui.PopupPanel
                  |
                  +-com.google.gwt.user.client.ui.DecoratedPopupPanel
                     |
                     +-se.lnu.client.view.VR_WaitPopUp
```

All Implemented Interfaces:

com.google.gwt.user.client.ui.HasVisibility, com.google.gwt.user.client.ui.IsWidget, com.google.gwt.event.logical.shared.HasAttachHandlers, com.google.gwt.user.client.EventListener, com.google.gwt.user.client.ui.HasWidgets.ForIsWidget, com.google.gwt.user.client.ui.HasOneWidget, com.google.gwt.event.logical.shared.HasCloseHandlers, com.google.gwt.user.client.ui.HasAnimation, com.google.gwt.user.client.EventPreview, com.google.gwt.user.client.ui.SourcesPopupEvents

public class **VR_WaitPopUp**
extends com.google.gwt.user.client.ui.DecoratedPopupPanel

Fields inherited from class `com.google.gwt.user.client.ui.UIObject`
`DEBUG_ID_PREFIX`

Constructor Summary

`public`	`VR_WaitPopUp()`

Methods inherited from class `com.google.gwt.user.client.ui.DecoratedPopupPanel`
`clear, doAttachChildren, doDetachChildren, getCellElement, getWidget, iterator, remove, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.PopupPanel`
`addAutoHidePartner, addCloseHandler, addPopupListener, center, getContainerElement, getGlassElement, getGlassStyleName, getOffsetHeight, getOffsetWidth, getPopupLeft, getPopupTop, getStyleElement, getTitle, hide, hide, isAnimationEnabled, isAutoHideEnabled, isAutoHideOnHistoryEventsEnabled, isGlassEnabled, isModal, isPreviewingAllNativeEvents, isShowing, isVisible, onEventPreview, onKeyDownPreview, onKeyPressPreview, onKeyUpPreview, onPreviewNativeEvent, onUnload, removeAutoHidePartner, removePopupListener, setAnimationEnabled, setAutoHideEnabled, setAutoHideOnHistoryEventsEnabled, setGlassEnabled, setGlassStyleName, setHeight, setModal, setPopupPosition, setPopupPositionAndShow, setPreviewingAllNativeEvents, setTitle, setWidget, setWidth, setVisible, show, showRelativeTo`

Methods inherited from class `com.google.gwt.user.client.ui.SimplePanel`

`add, getContainerElement, getWidget, iterator, remove, setWidget, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.Panel`
`add, add, adopt, adopt, clear, disown, doAttachChildren, doDetachChildren, orphan, remove, remove`

Methods inherited from class `com.google.gwt.user.client.ui.Widget`
`addAttachHandler, addBitlessDomHandler, addDomHandler, addHandler, asWidget, asWidgetOrNull, createHandlerManager, delegateEvent, doAttachChildren, doDetachChildren, fireEvent, getHandlerCount, getLayoutData, getParent, isAttached, isOrWasAttached, onAttach, onBrowserEvent, onDetach, onLoad, onUnload, removeFromParent, setLayoutData, sinkEvents`

Methods inherited from class `com.google.gwt.user.client.ui.UIObject`
`addStyleDependentName, addStyleName, ensureDebugId, ensureDebugId, ensureDebugId, getAbsoluteLeft, getAbsoluteTop, getElement, getOffsetHeight, getOffsetWidth, getStyleElement, getStyleName, getStyleName, getStylePrimaryName, getStylePrimaryName, getTitle, isVisible, isVisible, onEnsureDebugId, removeStyleDependentName, removeStyleName, resolvePotentialElement, setElement, setElement, setHeight, setPixelSize, setSize, setStyleDependentName, setStyleName, setStyleName, setStyleName, setStyleName, setStylePrimaryName, setStylePrimaryName, setTitle, setWidth, setVisible, setVisible, sinkBitlessEvent, sinkEvents, toString, unsinkEvents`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.google.gwt.user.client.ui.HasVisibility`
`isVisible, setVisible`

Methods inherited from interface `com.google.gwt.user.client.EventListener`
`onBrowserEvent`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasAttachHandlers`
`addAttachHandler, isAttached`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Methods inherited from interface `com.google.gwt.user.client.ui.IsWidget`
`asWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets.ForIsWidget`
`add, remove`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets`

`add, clear, iterator, remove`

Methods inherited from interface `java.lang.Iterable`
`iterator`

Methods inherited from interface `com.google.gwt.user.client.ui.HasOneWidget`
`getWidget, setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.AcceptsOneWidget`
`setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.SourcesPopupEvents`
`addPopupListener, removePopupListener`

Methods inherited from interface `com.google.gwt.user.client.EventPreview`
`onEventPreview`

Methods inherited from interface `com.google.gwt.user.client.ui.HasAnimation`
`isAnimationEnabled, setAnimationEnabled`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasCloseHandlers`
`addCloseHandler`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Constructors

VR_WaitPopUp

public **VR_WaitPopUp**()

Package se.lnu.client.view.charts

se.lnu.client.view.charts
Class VR_BarChart

```
java.lang.Object
  |
  +-se.lnu.client.view.charts.VR_BarChart
```

All Implemented Interfaces:
java.lang.Runnable

public class **VR_BarChart**
extends java.lang.Object
implements java.lang.Runnable

Constructor Summary

public	VR_BarChart()

Method Summary

void	run() This method is automatically created from the implementation of Runnable to the class.

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `java.lang.Runnable`

run

Constructors

VR_BarChart

`public VR_BarChart()`

Methods

run

`public void run()`

This method is automatically created from the implementation of Runnable to the class. It will be used if a new thread will be started. It will create a Bar Chart with help of the two helper methods createOptions() and createTable()

se.lnu.client.view.charts
Class VR_Cumulus

```
java.lang.Object
  |
  +-se.lnu.client.view.charts.VR_Cumulus
```

public class **VR_Cumulus**
extends java.lang.Object

This method creates a TagCloud which holds the 10 frequently words in a document and present it in a 3D Cloud

Constructor Summary

`public`	`VR_Cumulus(java.util.ArrayList source)` Constructor for the class VR_Cumulus

Method Summary

`void`	`createList()` This method creates a list of values which the TagCanvas module use for his TagCloud
`static void`	`getDataFromSelectedWord(java.lang.String word)`
`native void`	`sendSelectedWord()`
`native void`	`wordcloud()` starts the TagCloud and his listeners

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

VR_Cumulus

`public **VR_Cumulus**(java.util.ArrayList source)`

Constructor for the class VR_Cumulus

Parameters:
`vet`
`source`

Methods

getDataFromSelectedWord

`public static void **getDataFromSelectedWord**(java.lang.String word)`

(continued from last page)

createList

`public void` **`createList`**`()`

This method creates a list of values which the TagCanvas module use for his TagCloud

sendSelectedWord

`public native void` **`sendSelectedWord`**`()`

wordcloud

`public native void` **`wordcloud`**`()`

starts the TagCloud and his listeners

se.lnu.client.view.charts
Class VR_ParallelCoordinates

```
java.lang.Object
   |
   +-com.google.gwt.core.client.JavaScriptObject
      |
      +-com.google.gwt.dom.client.Node
         |
         +-com.google.gwt.dom.client.Element
            |
            +-se.lnu.client.view.charts.VR_ParallelCoordinates
```

public class **VR_ParallelCoordinates**
extends com.google.gwt.dom.client.Element

Fields inherited from class `com.google.gwt.dom.client.Element`
DRAGGABLE_AUTO, DRAGGABLE_FALSE, DRAGGABLE_TRUE

Fields inherited from class `com.google.gwt.dom.client.Node`
DOCUMENT_NODE, ELEMENT_NODE, TEXT_NODE

Constructor Summary

`protected`	`VR_ParallelCoordinates()` The class have to have a protected constructor otherwhise the JSNI would not work

Method Summary

`static native com.google.gwt.dom.client.Element`	`parallelCoordinates(int width, int height)` This method starts the Parallel Coordinates using JSNI
`static void`	`start()` This method creates the container for the Parallel Coordinates

Methods inherited from class `com.google.gwt.dom.client.Element`
addClassName, as, as, blur, dispatchEvent, focus, getAbsoluteBottom, getAbsoluteLeft, getAbsoluteRight, getAbsoluteTop, getAttribute, getClassName, getClientHeight, getClientWidth, getDir, getDraggable, getElementsByTagName, getFirstChildElement, getId, getInnerHTML, getInnerText, getLang, getNextSiblingElement, getOffsetHeight, getOffsetLeft, getOffsetParent, getOffsetTop, getOffsetWidth, getPropertyBoolean, getPropertyDouble, getPropertyInt, getPropertyJSO, getPropertyObject, getPropertyString, getScrollHeight, getScrollLeft, getScrollTop, getScrollWidth, getString, getStyle, getTabIndex, getTagName, getTitle, hasAttribute, hasTagName, is, is, removeAttribute, removeClassName, replaceClassName, scrollIntoView, setAttribute, setClassName, setDir, setDraggable, setId, setInnerHTML, setInnerText, setLang, setPropertyBoolean, setPropertyDouble, setPropertyInt, setPropertyJSO, setPropertyObject, setPropertyString, setScrollLeft, setScrollTop, setTabIndex, setTitle

Methods inherited from class `com.google.gwt.dom.client.Node`

`appendChild, as, cloneNode, getChild, getChildCount, getChildNodes, getFirstChild, getLastChild, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentElement, getParentNode, getPreviousSibling, hasChildNodes, hasParentElement, insertAfter, insertBefore, insertFirst, is, isOrHasChild, removeChild, removeFromParent, replaceChild, setNodeValue`

Methods inherited from class `com.google.gwt.core.client.JavaScriptObject`

`cast, createArray, createFunction, createObject, equals, hashCode, toSource, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

VR_ParallelCoordinates

```
protected VR_ParallelCoordinates()
```

The class have to have a protected constructor otherwhise the JSNI would not work

Methods

parallelCoordinates

```
public static native com.google.gwt.dom.client.Element parallelCoordinates(int width,
        int height)
```

This method starts the Parallel Coordinates using JSNI

Parameters:

`width`

`height`

Returns:

Element Parallel Coordinates

start

```
public static void start()
```

This method creates the container for the Parallel Coordinates

se.lnu.client.view.charts
Class VR_PieChart

```
java.lang.Object
  |
  +-se.lnu.client.view.charts.VR_PieChart
```

All Implemented Interfaces:
java.lang.Runnable

public class **VR_PieChart**
extends java.lang.Object
implements java.lang.Runnable

Constructor Summary

public	`VR_PieChart()` This method is a constructor and loads an ArrayList with the current university and their institutes

Method Summary

void	`run()` Due to the fact that this class has an implementation of Runnable it automatically creates this class.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `java.lang.Runnable`

`run`

Constructors

VR_PieChart

```
public VR_PieChart()
```

This method is a constructor and loads an ArrayList with the current university and their institutes

Methods

run

```
public void run()
```

Due to the fact that this class has an implementation of Runnable it automatically creates this class. It will be activated if a new thread is starting This method creates a container and the Pie Chart itself.

se.lnu.client.view.charts
Class VR_ShowDetails

```
java.lang.Object
  |
  +-com.google.gwt.user.client.ui.UIObject
      |
      +-com.google.gwt.user.client.ui.Widget
          |
          +-com.google.gwt.user.client.ui.Panel
              |
              +-com.google.gwt.user.client.ui.SimplePanel
                  |
                  +-com.google.gwt.user.client.ui.PopupPanel
                      |
                      +-com.google.gwt.user.client.ui.DecoratedPopupPanel
                          |
                          +-se.lnu.client.view.charts.VR_ShowDetails
```

All Implemented Interfaces:

com.google.gwt.user.client.ui.HasVisibility, com.google.gwt.user.client.ui.IsWidget, com.google.gwt.event.logical.shared.HasAttachHandlers, com.google.gwt.user.client.EventListener, com.google.gwt.user.client.ui.HasWidgets.ForIsWidget, com.google.gwt.user.client.ui.HasOneWidget, com.google.gwt.event.logical.shared.HasCloseHandlers, com.google.gwt.user.client.ui.HasAnimation, com.google.gwt.user.client.EventPreview, com.google.gwt.user.client.ui.SourcesPopupEvents

---

public class **VR_ShowDetails**
extends com.google.gwt.user.client.ui.DecoratedPopupPanel

Fields inherited from class `com.google.gwt.user.client.ui.UIObject`
`DEBUG_ID_PREFIX`

Constructor Summary

`public`	`VR_ShowDetails(int row)`

Method Summary

`static java.lang.String`	`beviljatToString(java.util.ArrayList bEntry)` This method helps to make the money/year relationship more readable in the detailed view

Methods inherited from class `com.google.gwt.user.client.ui.DecoratedPopupPanel`
`clear, doAttachChildren, doDetachChildren, getCellElement, getWidget, iterator, remove, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.PopupPanel`

addAutoHidePartner, addCloseHandler, addPopupListener, center, getContainerElement, getGlassElement, getGlassStyleName, getOffsetHeight, getOffsetWidth, getPopupLeft, getPopupTop, getStyleElement, getTitle, hide, hide, isAnimationEnabled, isAutoHideEnabled, isAutoHideOnHistoryEventsEnabled, isGlassEnabled, isModal, isPreviewingAllNativeEvents, isShowing, isVisible, onEventPreview, onKeyDownPreview, onKeyPressPreview, onKeyUpPreview, onPreviewNativeEvent, onUnload, removeAutoHidePartner, removePopupListener, setAnimationEnabled, setAutoHideEnabled, setAutoHideOnHistoryEventsEnabled, setGlassEnabled, setGlassStyleName, setHeight, setModal, setPopupPosition, setPopupPositionAndShow, setPreviewingAllNativeEvents, setTitle, setWidget, setWidth, setVisible, show, showRelativeTo

Methods inherited from class `com.google.gwt.user.client.ui.SimplePanel`
add, getContainerElement, getWidget, iterator, remove, setWidget, setWidget

Methods inherited from class `com.google.gwt.user.client.ui.Panel`
add, add, adopt, adopt, clear, disown, doAttachChildren, doDetachChildren, orphan, remove, remove

Methods inherited from class `com.google.gwt.user.client.ui.Widget`
addAttachHandler, addBitlessDomHandler, addDomHandler, addHandler, asWidget, asWidgetOrNull, createHandlerManager, delegateEvent, doAttachChildren, doDetachChildren, fireEvent, getHandlerCount, getLayoutData, getParent, isAttached, isOrWasAttached, onAttach, onBrowserEvent, onDetach, onLoad, onUnload, removeFromParent, setLayoutData, sinkEvents

Methods inherited from class `com.google.gwt.user.client.ui.UIObject`
addStyleDependentName, addStyleName, ensureDebugId, ensureDebugId, ensureDebugId, getAbsoluteLeft, getAbsoluteTop, getElement, getOffsetHeight, getOffsetWidth, getStyleElement, getStyleName, getStyleName, getStylePrimaryName, getStylePrimaryName, getTitle, isVisible, isVisible, onEnsureDebugId, removeStyleDependentName, removeStyleName, resolvePotentialElement, setElement, setElement, setHeight, setPixelSize, setSize, setStyleDependentName, setStyleName, setStyleName, setStyleName, setStyleName, setStylePrimaryName, setStylePrimaryName, setTitle, setWidth, setVisible, setVisible, sinkBitlessEvent, sinkEvents, toString, unsinkEvents

Methods inherited from class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `com.google.gwt.user.client.ui.HasVisibility`
isVisible, setVisible

Methods inherited from interface `com.google.gwt.user.client.EventListener`
onBrowserEvent

Methods inherited from interface `com.google.gwt.event.logical.shared.HasAttachHandlers`
addAttachHandler, isAttached

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Methods inherited from interface `com.google.gwt.user.client.ui.IsWidget`
`asWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets.ForIsWidget`
`add, remove`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets`
`add, clear, iterator, remove`

Methods inherited from interface `java.lang.Iterable`
`iterator`

Methods inherited from interface `com.google.gwt.user.client.ui.HasOneWidget`
`getWidget, setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.AcceptsOneWidget`
`setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.SourcesPopupEvents`
`addPopupListener, removePopupListener`

Methods inherited from interface `com.google.gwt.user.client.EventPreview`
`onEventPreview`

Methods inherited from interface `com.google.gwt.user.client.ui.HasAnimation`
`isAnimationEnabled, setAnimationEnabled`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasCloseHandlers`
`addCloseHandler`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Constructors

VR_ShowDetails

```
public VR_ShowDetails(int row)
```

Methods

beviljatToString

```
public static java.lang.String beviljatToString(java.util.ArrayList bEntry)
```

This method helps to make the money/year relationship more readable in the detailed view

Parameters:

`(ArrayList`

Returns:

String

se.lnu.client.view.charts
Class VR_Table

```
java.lang.Object
  |
  +-se.lnu.client.view.charts.VR_Table
```

All Implemented Interfaces:
java.lang.Runnable

public class **VR_Table**
extends java.lang.Object
implements java.lang.Runnable

Constructor Summary

`public`	`VR_Table()` This is the default constructor
`public`	`VR_Table(java.util.ArrayList source, boolean show)` This method is a constructor which only will be fired if a ArrayList of DataEntries and additional to that a boolean with the instruction to focus on the table.

Method Summary

`static java.util.ArrayList`	`getCurrentTableSource()` This method returns a ArrayList of the DataEntries which are currently loaded
`void`	`run()` This method is automatically created from the implementation of Runnable It creates the Table in the first run of this class

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `java.lang.Runnable`

`run`

Constructors

VR_Table

`public VR_Table()`

This is the default constructor

(continued from last page)

VR_Table

```
public VR_Table(java.util.ArrayList source,
                boolean show)
```

This method is a constructor which only will be fired if a ArrayList of DataEntries and additional to that a boolean with the instruction to focus on the table. Its a helper method to refresh the table after a visualisation has been clicked

Parameters:

`source`

`show`

Methods

run

```
public void run()
```

This method is automatically created from the implementation of Runnable It creates the Table in the first run of this class

getCurrentTableSource

```
public static java.util.ArrayList getCurrentTableSource()
```

This method returns a ArrayList of the DataEntries which are currently loaded

Returns:

ArrayList

Package se.lnu.client.view.googleMaps

se.lnu.client.view.googleMaps
Class GoogleMapsCircle

```
java.lang.Object
  |
  +-se.lnu.client.view.googleMaps.GoogleMapsCircle
```

public class **GoogleMapsCircle**
extends java.lang.Object

Constructor Summary

public	`GoogleMapsCircle()`

Method Summary

`com.google.gwt.maps.client.overlay.Polygon`	`drawCircleFromRadius(com.google.gwt.maps.client.geom.LatLng center, double radius, int nbOfPoints, java.lang.String color)` This

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GoogleMapsCircle

```
public GoogleMapsCircle()
```

Methods

drawCircleFromRadius

```
public com.google.gwt.maps.client.overlay.Polygon
drawCircleFromRadius(com.google.gwt.maps.client.geom.LatLng center,
         double radius,
         int nbOfPoints,
         java.lang.String color)
```

This

Parameters:

`center`
`radius`
`nbOfPoints`
`color`

Returns:

(continued from last page)

Polygon coloured Circle image

se.lnu.client.view.googleMaps
Class VR_GoogleMaps

```
java.lang.Object
   |
   +-se.lnu.client.view.googleMaps.VR_GoogleMaps
```

All Implemented Interfaces:
java.lang.Runnable

public class **VR_GoogleMaps**
extends java.lang.Object
implements java.lang.Runnable

Constructor Summary

`public`	`VR_GoogleMaps()` This constructor will start the Google Maps

Method Summary

`void`	`buildUi()` This method builds the Google Maps
`void`	`renewMarker()` This method will renew the markers with new data from the database if the selected range of time has changed
`void`	`run()` The run method of the thread

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `java.lang.Runnable`

`run`

Constructors

VR_GoogleMaps

`public VR_GoogleMaps()`

This constructor will start the Google Maps

Methods

(continued from last page)

run

```
public void run()
```

The run method of the thread

buildUi

```
protected void buildUi()
```

This method builds the Google Maps

renewMarker

```
public void renewMarker()
```

This method will renew the markers with new data from the database if the selected range of time has changed

se.lnu.client.view.googleMaps
Class VR_GradientChart

```
java.lang.Object
  |
  +-se.lnu.client.view.googleMaps.VR_GradientChart
```

public class **VR_GradientChart**
extends java.lang.Object

Constructor Summary

public	VR_GradientChart()

Method Summary

com.google.gwt.canvas.client.Canvas	initiate()

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

VR_GradientChart

`public VR_GradientChart()`

Methods

initiate

`public com.google.gwt.canvas.client.Canvas initiate()`

Package
se.lnu.client.view.menu

se.lnu.client.view.menu
Class VR_Access

```
java.lang.Object
   |
   +-com.google.gwt.user.client.ui.UIObject
      |
      +-com.google.gwt.user.client.ui.Widget
         |
         +-com.google.gwt.user.client.ui.Panel
            |
            +-com.google.gwt.user.client.ui.SimplePanel
               |
               +-com.google.gwt.user.client.ui.PopupPanel
                  |
                  +-com.google.gwt.user.client.ui.DecoratedPopupPanel
                     |
                     +-se.lnu.client.view.menu.VR_Access
```

All Implemented Interfaces:

com.google.gwt.user.client.ui.HasVisibility, com.google.gwt.user.client.ui.IsWidget, com.google.gwt.event.logical.shared.HasAttachHandlers, com.google.gwt.user.client.EventListener, com.google.gwt.user.client.ui.HasWidgets.ForIsWidget, com.google.gwt.user.client.ui.HasOneWidget, com.google.gwt.event.logical.shared.HasCloseHandlers, com.google.gwt.user.client.ui.HasAnimation, com.google.gwt.user.client.EventPreview, com.google.gwt.user.client.ui.SourcesPopupEvents

---

public class **VR_Access**
extends com.google.gwt.user.client.ui.DecoratedPopupPanel

Fields inherited from class `com.google.gwt.user.client.ui.UIObject`
`DEBUG_ID_PREFIX`

Constructor Summary

`public`	`VR_Access(java.lang.String windowName, java.util.ArrayList accessList)` This method is the constructor of the class and needs the name of the module (until now download or newAccount) and the accesslist which is received from the database

Methods inherited from class `com.google.gwt.user.client.ui.DecoratedPopupPanel`
`clear, doAttachChildren, doDetachChildren, getCellElement, getWidget, iterator, remove, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.PopupPanel`
`addAutoHidePartner, addCloseHandler, addPopupListener, center, getContainerElement, getGlassElement, getGlassStyleName, getOffsetHeight, getOffsetWidth, getPopupLeft, getPopupTop, getStyleElement, getTitle, hide, hide, isAnimationEnabled, isAutoHideEnabled, isAutoHideOnHistoryEventsEnabled, isGlassEnabled, isModal, isPreviewingAllNativeEvents, isShowing, isVisible, onEventPreview, onKeyDownPreview, onKeyPressPreview, onKeyUpPreview, onPreviewNativeEvent, onUnload, removeAutoHidePartner, removePopupListener, setAnimationEnabled, setAutoHideEnabled, setAutoHideOnHistoryEventsEnabled, setGlassEnabled, setGlassStyleName, setHeight, setModal, setPopupPosition, setPopupPositionAndShow, setPreviewingAllNativeEvents, setTitle, setWidget, setWidth, setVisible, show, showRelativeTo`

Methods inherited from class `com.google.gwt.user.client.ui.SimplePanel`

`add, getContainerElement, getWidget, iterator, remove, setWidget, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.Panel`
`add, add, adopt, adopt, clear, disown, doAttachChildren, doDetachChildren, orphan, remove, remove`

Methods inherited from class `com.google.gwt.user.client.ui.Widget`
`addAttachHandler, addBitlessDomHandler, addDomHandler, addHandler, asWidget, asWidgetOrNull, createHandlerManager, delegateEvent, doAttachChildren, doDetachChildren, fireEvent, getHandlerCount, getLayoutData, getParent, isAttached, isOrWasAttached, onAttach, onBrowserEvent, onDetach, onLoad, onUnload, removeFromParent, setLayoutData, sinkEvents`

Methods inherited from class `com.google.gwt.user.client.ui.UIObject`
`addStyleDependentName, addStyleName, ensureDebugId, ensureDebugId, ensureDebugId, getAbsoluteLeft, getAbsoluteTop, getElement, getOffsetHeight, getOffsetWidth, getStyleElement, getStyleName, getStyleName, getStylePrimaryName, getStylePrimaryName, getTitle, isVisible, isVisible, onEnsureDebugId, removeStyleDependentName, removeStyleName, resolvePotentialElement, setElement, setElement, setHeight, setPixelSize, setSize, setStyleDependentName, setStyleName, setStyleName, setStyleName, setStyleName, setStylePrimaryName, setStylePrimaryName, setTitle, setWidth, setVisible, setVisible, sinkBitlessEvent, sinkEvents, toString, unsinkEvents`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.google.gwt.user.client.ui.HasVisibility`
`isVisible, setVisible`

Methods inherited from interface `com.google.gwt.user.client.EventListener`
`onBrowserEvent`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasAttachHandlers`
`addAttachHandler, isAttached`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Methods inherited from interface `com.google.gwt.user.client.ui.IsWidget`
`asWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets.ForIsWidget`
`add, remove`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets`

`add, clear, iterator, remove`

Methods inherited from interface `java.lang.Iterable`
`iterator`

Methods inherited from interface `com.google.gwt.user.client.ui.HasOneWidget`
`getWidget, setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.AcceptsOneWidget`
`setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.SourcesPopupEvents`
`addPopupListener, removePopupListener`

Methods inherited from interface `com.google.gwt.user.client.EventPreview`
`onEventPreview`

Methods inherited from interface `com.google.gwt.user.client.ui.HasAnimation`
`isAnimationEnabled, setAnimationEnabled`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasCloseHandlers`
`addCloseHandler`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Constructors

VR_Access

```
public VR_Access(java.lang.String windowName,
                 java.util.ArrayList accessList)
```

This method is the constructor of the class and needs the name of the module (until now download or newAccount) and the accesslist which is received from the database

Parameters:

`String`

`ArrayList`

se.lnu.client.view.menu
Class VR_Download

```
java.lang.Object
  |
  +-com.google.gwt.user.client.ui.UIObject
      |
      +-com.google.gwt.user.client.ui.Widget
          |
          +-com.google.gwt.user.client.ui.Panel
              |
              +-com.google.gwt.user.client.ui.SimplePanel
                  |
                  +-com.google.gwt.user.client.ui.PopupPanel
                      |
                      +-com.google.gwt.user.client.ui.DecoratedPopupPanel
                          |
                          +-se.lnu.client.view.menu.VR_Download
```

All Implemented Interfaces:

com.google.gwt.user.client.ui.HasVisibility, com.google.gwt.user.client.ui.IsWidget, com.google.gwt.event.logical.shared.HasAttachHandlers, com.google.gwt.user.client.EventListener, com.google.gwt.user.client.ui.HasWidgets.ForIsWidget, com.google.gwt.user.client.ui.HasOneWidget, com.google.gwt.event.logical.shared.HasCloseHandlers, com.google.gwt.user.client.ui.HasAnimation, com.google.gwt.user.client.EventPreview, com.google.gwt.user.client.ui.SourcesPopupEvents

---

public class **VR_Download**
extends com.google.gwt.user.client.ui.DecoratedPopupPanel

Fields inherited from class `com.google.gwt.user.client.ui.UIObject`
`DEBUG_ID_PREFIX`

Constructor Summary

`public`	`VR_Download()` This constructor is creating the PopUp - Panel and starts the update

Method Summary

`void`	`updateDatabase()` This method is calling a method on the server which updates the database

Methods inherited from class `com.google.gwt.user.client.ui.DecoratedPopupPanel`
`clear, doAttachChildren, doDetachChildren, getCellElement, getWidget, iterator, remove, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.PopupPanel`

addAutoHidePartner, addCloseHandler, addPopupListener, center, getContainerElement, getGlassElement, getGlassStyleName, getOffsetHeight, getOffsetWidth, getPopupLeft, getPopupTop, getStyleElement, getTitle, hide, hide, isAnimationEnabled, isAutoHideEnabled, isAutoHideOnHistoryEventsEnabled, isGlassEnabled, isModal, isPreviewingAllNativeEvents, isShowing, isVisible, onEventPreview, onKeyDownPreview, onKeyPressPreview, onKeyUpPreview, onPreviewNativeEvent, onUnload, removeAutoHidePartner, removePopupListener, setAnimationEnabled, setAutoHideEnabled, setAutoHideOnHistoryEventsEnabled, setGlassEnabled, setGlassStyleName, setHeight, setModal, setPopupPosition, setPopupPositionAndShow, setPreviewingAllNativeEvents, setTitle, setWidget, setWidth, setVisible, show, showRelativeTo

Methods inherited from class com.google.gwt.user.client.ui.SimplePanel
add, getContainerElement, getWidget, iterator, remove, setWidget, setWidget

Methods inherited from class com.google.gwt.user.client.ui.Panel
add, add, adopt, adopt, clear, disown, doAttachChildren, doDetachChildren, orphan, remove, remove

Methods inherited from class com.google.gwt.user.client.ui.Widget
addAttachHandler, addBitlessDomHandler, addDomHandler, addHandler, asWidget, asWidgetOrNull, createHandlerManager, delegateEvent, doAttachChildren, doDetachChildren, fireEvent, getHandlerCount, getLayoutData, getParent, isAttached, isOrWasAttached, onAttach, onBrowserEvent, onDetach, onLoad, onUnload, removeFromParent, setLayoutData, sinkEvents

Methods inherited from class com.google.gwt.user.client.ui.UIObject
addStyleDependentName, addStyleName, ensureDebugId, ensureDebugId, ensureDebugId, getAbsoluteLeft, getAbsoluteTop, getElement, getOffsetHeight, getOffsetWidth, getStyleElement, getStyleName, getStyleName, getStylePrimaryName, getStylePrimaryName, getTitle, isVisible, isVisible, onEnsureDebugId, removeStyleDependentName, removeStyleName, resolvePotentialElement, setElement, setElement, setHeight, setPixelSize, setSize, setStyleDependentName, setStyleName, setStyleName, setStyleName, setStyleName, setStylePrimaryName, setStylePrimaryName, setTitle, setWidth, setVisible, setVisible, sinkBitlessEvent, sinkEvents, toString, unsinkEvents

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface com.google.gwt.user.client.ui.HasVisibility
isVisible, setVisible

Methods inherited from interface com.google.gwt.user.client.EventListener
onBrowserEvent

Methods inherited from interface com.google.gwt.event.logical.shared.HasAttachHandlers
addAttachHandler, isAttached

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Methods inherited from interface `com.google.gwt.user.client.ui.IsWidget`
`asWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets.ForIsWidget`
`add, remove`

Methods inherited from interface `com.google.gwt.user.client.ui.HasWidgets`
`add, clear, iterator, remove`

Methods inherited from interface `java.lang.Iterable`
`iterator`

Methods inherited from interface `com.google.gwt.user.client.ui.HasOneWidget`
`getWidget, setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.AcceptsOneWidget`
`setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.SourcesPopupEvents`
`addPopupListener, removePopupListener`

Methods inherited from interface `com.google.gwt.user.client.EventPreview`
`onEventPreview`

Methods inherited from interface `com.google.gwt.user.client.ui.HasAnimation`
`isAnimationEnabled, setAnimationEnabled`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasCloseHandlers`
`addCloseHandler`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Constructors

VR_Download

```
public VR_Download()
```

This constructor is creating the PopUp - Panel and starts the update

Methods

updateDatabase

```
public void updateDatabase()
```

This method is calling a method on the server which updates the database

se.lnu.client.view.menu
Class VR_menu

```
java.lang.Object
  |
  +-se.lnu.client.view.menu.VR_menu
```

public class **VR_menu**
extends java.lang.Object

Constructor Summary

`public`	`VR_menu()` This constructor calls the menu Panel

Method Summary

`com.google.gwt.user.client.ui.MenuBar`	`initialize()` This method creates the MenuBar which will be set into the MenuPanel
`void`	`setMenuPanel()` This method will create the menu panel and sets the MenuBar and the current University name besides that
`static void`	`updateUniname(java.lang.String uniname)` This method sets the current selected universitys name next to the menu

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

VR_menu

`public VR_menu()`

This constructor calls the menu Panel

Methods

updateUniname

`public static void updateUniname(java.lang.String uniname)`

This method sets the current selected universitys name next to the menu

Parameters:
`uniname`

setMenuPanel

```
public void setMenuPanel()
```

This method will create the menu panel and sets the MenuBar and the current University name besides that

initialize

```
public com.google.gwt.user.client.ui.MenuBar initialize()
```

This method creates the MenuBar which will be set into the MenuPanel

Returns:
MenuBar

se.lnu.client.view.menu
Class VR_SetUserAccess

```
java.lang.Object
  |
  +-com.google.gwt.user.client.ui.UIObject
      |
      +-com.google.gwt.user.client.ui.Widget
          |
          +-com.google.gwt.user.client.ui.Panel
              |
              +-com.google.gwt.user.client.ui.SimplePanel
                  |
                  +-com.google.gwt.user.client.ui.PopupPanel
                      |
                      +-com.google.gwt.user.client.ui.DecoratedPopupPanel
                          |
                          +-se.lnu.client.view.menu.VR_SetUserAccess
```

All Implemented Interfaces:

com.google.gwt.user.client.ui.HasVisibility, com.google.gwt.user.client.ui.IsWidget, com.google.gwt.event.logical.shared.HasAttachHandlers, com.google.gwt.user.client.EventListener, com.google.gwt.user.client.ui.HasWidgets.ForIsWidget, com.google.gwt.user.client.ui.HasOneWidget, com.google.gwt.event.logical.shared.HasCloseHandlers, com.google.gwt.user.client.ui.HasAnimation, com.google.gwt.user.client.EventPreview, com.google.gwt.user.client.ui.SourcesPopupEvents

public class **VR_SetUserAccess**
extends com.google.gwt.user.client.ui.DecoratedPopupPanel

Fields inherited from class `com.google.gwt.user.client.ui.UIObject`
`DEBUG_ID_PREFIX`

Constructor Summary	
`public`	`VR_SetUserAccess()` This constructer will start the PopUp-Panel

Methods inherited from class `com.google.gwt.user.client.ui.DecoratedPopupPanel`
`clear, doAttachChildren, doDetachChildren, getCellElement, getWidget, iterator, remove, setWidget`

Methods inherited from class `com.google.gwt.user.client.ui.PopupPanel`
`addAutoHidePartner, addCloseHandler, addPopupListener, center, getContainerElement, getGlassElement, getGlassStyleName, getOffsetHeight, getOffsetWidth, getPopupLeft, getPopupTop, getStyleElement, getTitle, hide, hide, isAnimationEnabled, isAutoHideEnabled, isAutoHideOnHistoryEventsEnabled, isGlassEnabled, isModal, isPreviewingAllNativeEvents, isShowing, isVisible, onEventPreview, onKeyDownPreview, onKeyPressPreview, onKeyUpPreview, onPreviewNativeEvent, onUnload, removeAutoHidePartner, removePopupListener, setAnimationEnabled, setAutoHideEnabled, setAutoHideOnHistoryEventsEnabled, setGlassEnabled, setGlassStyleName, setHeight, setModal, setPopupPosition, setPopupPositionAndShow, setPreviewingAllNativeEvents, setTitle, setWidget, setWidth, setVisible, show, showRelativeTo`

Methods inherited from class `com.google.gwt.user.client.ui.SimplePanel`

add, getContainerElement, getWidget, iterator, remove, setWidget, setWidget

Methods inherited from class com.google.gwt.user.client.ui.Panel
add, add, adopt, adopt, clear, disown, doAttachChildren, doDetachChildren, orphan, remove, remove

Methods inherited from class com.google.gwt.user.client.ui.Widget
addAttachHandler, addBitlessDomHandler, addDomHandler, addHandler, asWidget, asWidgetOrNull, createHandlerManager, delegateEvent, doAttachChildren, doDetachChildren, fireEvent, getHandlerCount, getLayoutData, getParent, isAttached, isOrWasAttached, onAttach, onBrowserEvent, onDetach, onLoad, onUnload, removeFromParent, setLayoutData, sinkEvents

Methods inherited from class com.google.gwt.user.client.ui.UIObject
addStyleDependentName, addStyleName, ensureDebugId, ensureDebugId, ensureDebugId, getAbsoluteLeft, getAbsoluteTop, getElement, getOffsetHeight, getOffsetWidth, getStyleElement, getStyleName, getStyleName, getStylePrimaryName, getStylePrimaryName, getTitle, isVisible, isVisible, onEnsureDebugId, removeStyleDependentName, removeStyleName, resolvePotentialElement, setElement, setElement, setHeight, setPixelSize, setSize, setStyleDependentName, setStyleName, setStyleName, setStyleName, setStyleName, setStylePrimaryName, setStylePrimaryName, setTitle, setWidth, setVisible, setVisible, sinkBitlessEvent, sinkEvents, toString, unsinkEvents

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface com.google.gwt.user.client.ui.HasVisibility
isVisible, setVisible

Methods inherited from interface com.google.gwt.user.client.EventListener
onBrowserEvent

Methods inherited from interface com.google.gwt.event.logical.shared.HasAttachHandlers
addAttachHandler, isAttached

Methods inherited from interface com.google.gwt.event.shared.HasHandlers
fireEvent

Methods inherited from interface com.google.gwt.user.client.ui.IsWidget
asWidget

Methods inherited from interface com.google.gwt.user.client.ui.HasWidgets.ForIsWidget
add, remove

Methods inherited from interface com.google.gwt.user.client.ui.HasWidgets

`add, clear, iterator, remove`

Methods inherited from interface `java.lang.Iterable`
`iterator`

Methods inherited from interface `com.google.gwt.user.client.ui.HasOneWidget`
`getWidget, setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.AcceptsOneWidget`
`setWidget`

Methods inherited from interface `com.google.gwt.user.client.ui.SourcesPopupEvents`
`addPopupListener, removePopupListener`

Methods inherited from interface `com.google.gwt.user.client.EventPreview`
`onEventPreview`

Methods inherited from interface `com.google.gwt.user.client.ui.HasAnimation`
`isAnimationEnabled, setAnimationEnabled`

Methods inherited from interface `com.google.gwt.event.logical.shared.HasCloseHandlers`
`addCloseHandler`

Methods inherited from interface `com.google.gwt.event.shared.HasHandlers`
`fireEvent`

Constructors

VR_SetUserAccess

`public` **`VR_SetUserAccess`**`()`

This constructer will start the PopUp-Panel

Package
se.lnu.server.controller

se.lnu.server.controller
Class Database_ServiceImpl

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-com.google.gwt.user.server.rpc.AbstractRemoteServiceServlet
              |
              +-com.google.gwt.user.server.rpc.RemoteServiceServlet
                  |
                  +-se.lnu.server.controller.Database_ServiceImpl
```

All Implemented Interfaces:
Database_Service, java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable, com.google.gwt.user.server.rpc.SerializationPolicyProvider

---

public class **Database_ServiceImpl**
extends com.google.gwt.user.server.rpc.RemoteServiceServlet
implements com.google.gwt.user.server.rpc.SerializationPolicyProvider, java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig, java.io.Serializable, Database_Service

---

Fields inherited from class `com.google.gwt.user.server.rpc.AbstractRemoteServiceServlet`
`perThreadRequest`, `perThreadResponse`

Constructor Summary

`public`	`Database_ServiceImpl()`

Method Summary

`java.util.ArrayList`	`getAccessList()` This method returns a ArrayList with users who are allowed to make a database update
`java.util.ArrayList`	`getAllUniversitys(java.lang.String minValue, java.lang.String maxValue)` This method returns all UniversityEntries depending on the the range of the slider
`java.util.ArrayList`	`getData(java.lang.String uni, java.lang.String minValue, java.lang.String maxValue)` This method returns all DataEntries in a ArrayList which has the given unversity name and is also within the range of the slider
`java.util.ArrayList`	`getFrequentlyWords(java.lang.String description)` This method returns a ArrayList of Strings and contains the 15 frequently words
`void`	`getMultiCoordinatesData()` This method will create a fresh Comma Seperated Values (CSV) - file which conatains the data to create a parallel coordinates system.
`void`	`setNewUser(GWT_Access newAccess)` This method will set a new user

java.lang.String	updateDB() This method will trigger the update

Methods inherited from class com.google.gwt.user.server.rpc.RemoteServiceServlet
checkPermutationStrongName, doGetSerializationPolicy, getSerializationPolicy, onAfterResponseSerialized, onBeforeRequestDeserialized, processCall, processPost, shouldCompressResponse

Methods inherited from class com.google.gwt.user.server.rpc.AbstractRemoteServiceServlet
doPost, doUnexpectedFailure, getPermutationStrongName, getThreadLocalRequest, getThreadLocalResponse, onAfterRequestDeserialized, processPost, readContent

Methods inherited from class javax.servlet.http.HttpServlet
doDelete, doGet, doHead, doOptions, doPost, doPut, doTrace, getLastModified, service, service

Methods inherited from class javax.servlet.GenericServlet
destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.servlet.Servlet
destroy, getServletConfig, getServletInfo, init, service

Methods inherited from interface javax.servlet.ServletConfig
getInitParameter, getInitParameterNames, getServletContext, getServletName

Methods inherited from interface com.google.gwt.user.server.rpc.SerializationPolicyProvider
getSerializationPolicy

Methods inherited from interface se.lnu.client.controller.database.Database_Service
getAccessList, getAllUniversitys, getData, getFrequentlyWords, getMultiCoordinatesData, setNewUser, updateDB

Constructors

Database_ServiceImpl

```
public Database_ServiceImpl()
```

Methods

(continued from last page)

getAllUniversitys

```
public java.util.ArrayList getAllUniversitys(java.lang.String minValue,
         java.lang.String maxValue)
```

This method returns all UniversityEntries depending on the the range of the slider

getData

```
public java.util.ArrayList getData(java.lang.String uni,
         java.lang.String minValue,
         java.lang.String maxValue)
```

This method returns all DataEntries in a ArrayList which has the given unversity name and is also within the range of the slider

getFrequentlyWords

```
public java.util.ArrayList getFrequentlyWords(java.lang.String description)
```

This method returns a ArrayList of Strings and contains the 15 frequently words

getMultiCoordinatesData

```
public void getMultiCoordinatesData()
```

This method will create a fresh Comma Seperated Values (CSV) - file which conatains the data to create a parallel coordinates system. The external javascript picks the data up and creates a parallel coordinate visualization out of it (D3.js)

updateDB

```
public java.lang.String updateDB()
```

This method will trigger the update

getAccessList

```
public java.util.ArrayList getAccessList()
```

This method returns a ArrayList with users who are allowed to make a database update

setNewUser

```
public void setNewUser(GWT_Access newAccess)
```

This method will set a new user

Package se.lnu.server.controller.charts

se.lnu.server.controller.charts
Class CountFrequentlyWords

```
java.lang.Object
  |
  +-se.lnu.server.controller.charts.CountFrequentlyWords
```

public class **CountFrequentlyWords**
extends java.lang.Object

Constructor Summary

public	CountFrequentlyWords()

Method Summary

java.util.ArrayList	countWords(java.lang.String strin) This method returns the last 15 frequently words in a String
boolean	isFillers(java.lang.String strin) This method checks if the word is a stop word

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

CountFrequentlyWords

`public CountFrequentlyWords()`

Methods

countWords

`public java.util.ArrayList countWords(java.lang.String strin)`

This method returns the last 15 frequently words in a String

Parameters:
`String`

isFillers

`public boolean isFillers(java.lang.String strin)`

This method checks if the word is a stop word

(continued from last page)

Parameters:

`String`

Returns:

boolean

se.lnu.server.controller.charts
Class Map_UniversitysList

```
java.lang.Object
  |
  +-se.lnu.server.controller.charts.Map_UniversitysList
```

public class **Map_UniversitysList**
extends java.lang.Object

Constructor Summary

public	Map_UniversitysList()

Method Summary

java.util.ArrayList	createUniList(java.lang.String minValue, java.lang.String maxValue)
static java.lang.String[]	getUniNames()

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Map_UniversitysList

```
public Map_UniversitysList()
```

Methods

getUniNames

```
public static java.lang.String[] getUniNames()
```

createUniList

```
public java.util.ArrayList createUniList(java.lang.String minValue,
                                        java.lang.String maxValue)
```

se.lnu.server.controller.charts
Class ParallelCoordinates

```
java.lang.Object
  |
  +-se.lnu.server.controller.charts.ParallelCoordinates
```

public class **ParallelCoordinates**
extends java.lang.Object

Constructor Summary

public	ParallelCoordinates()

Method Summary

java.lang.String	generateCsvFile(java.util.ArrayList source) This method will create the CSV file

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ParallelCoordinates

```
public ParallelCoordinates()
```

Methods

generateCsvFile

```
public java.lang.String generateCsvFile(java.util.ArrayList source)
```

This method will create the CSV file

Parameters:
`ArrayList`

Returns:
String

Package
se.lnu.server.controller.database

se.lnu.server.controller.database
Class DataEntries

```
java.lang.Object
  |
  +-se.lnu.server.controller.database.DataEntries
```

public class **DataEntries**
extends java.lang.Object

Constructor Summary

public	`DataEntries()`

Method Summary

`java.util.ArrayList`	`createArray(java.lang.String uni, java.lang.String minValue, java.lang.String maxValue)` This method calls the database and gets all necessary data to create a Arraylist with all universities
`void`	`createNewEntry(java.sql.ResultSet rs)` This method creates a new Entry

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DataEntries

```
public DataEntries()
```

Methods

createArray

```
public java.util.ArrayList createArray(java.lang.String uni,
                                       java.lang.String minValue,
                                       java.lang.String maxValue)
```

This method calls the database and gets all necessary data to create a Arraylist with all universities

Parameters:
`String`

Returns:

(continued from last page)

ArrayList

createNewEntry

```
public void createNewEntry(java.sql.ResultSet rs)
  throws java.sql.SQLException
```

This method creates a new Entry

Parameters:

`rs`

Throws:

`SQLException`

se.lnu.server.controller.database
Class DB_Access

```
java.lang.Object
  |
  +-se.lnu.server.controller.database.DB_Access
```

public class **DB_Access**
extends java.lang.Object

Constructor Summary

public	DB_Access()

Method Summary

java.util.ArrayList	get() This method gets a list of accepted user
void	set(GWT_Access access) This method creates a new user with his password into the database

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DB_Access

`public DB_Access()`

Methods

get

`public java.util.ArrayList get()`

This method gets a list of accepted user

Returns:
ArrayList

set

`public void set(GWT_Access access)`

This method creates a new user with his password into the database

(continued from last page)

Parameters:

`access`

se.lnu.server.controller.database
Class Dictionary

```
java.lang.Object
  |
  +-se.lnu.server.controller.database.Dictionary
```

public class **Dictionary**
extends java.lang.Object

This is not used yet This file contains two methods to search in a directory which is quite good.

Constructor Summary

`public`	`Dictionary()`

Method Summary

`java.lang.Object[]`	`getDic()` the other way to search in the directory so that the xml file will be used instead of the database
`static void`	`main(java.lang.String[] args)`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Dictionary

`public Dictionary()`

Methods

main

`public static void main(java.lang.String[] args)`

Parameters:
`args`

getDic

`public java.lang.Object[] getDic()`

(continued from last page)

the other way to search in the directory so that the xml file will be used instead of the database

Returns:
Object[]

se.lnu.server.controller.database
Class MySQL

```
java.lang.Object
  |
  +-se.lnu.server.controller.database.MySQL
```

public class **MySQL**
extends java.lang.Object

Constructor Summary

`public`	`MySQL()` This method is a constructor and gives the instruction to create the necessary tables if the application hast his first run

Method Summary

`void`	`callDB(java.lang.String sqlOrder)` This method is used for INSERT INTO queries to the database
`void`	`close()` This method closes the connection with the database
`void`	`connectDB()` This method establishes a connection to the database
`void`	`createDB()` This method creates the tables for the application
`java.sql.ResultSet`	`getResultsetFromDB(java.lang.String query)` This method will get data from the database , depending which query is fired

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MySQL

`public MySQL()`

This method is a constructor and gives the instruction to create the necessary tables if the application hast his first run

Methods

(continued from last page)

createDB

```
public void createDB()
  throws java.sql.SQLException
```

This method creates the tables for the application

Throws:
`SQLException`

connectDB

```
public void connectDB()
```

This method establishes a connection to the database

close

```
public void close()
```

This method closes the connection with the database

callDB

```
public void callDB(java.lang.String sqlOrder)
  throws java.sql.SQLException
```

This method is used for INSERT INTO queries to the database

Parameters:
`String` - query

Throws:
`SQLException`

getResultsetFromDB

```
public java.sql.ResultSet getResultsetFromDB(java.lang.String query)
```

This method will get data from the database , depending which query is fired

Parameters:
`String` - query

Returns:
ResultSet

se.lnu.server.controller.database
Class UpdateEntries

```
java.lang.Object
  |
  +-se.lnu.server.controller.database.UpdateEntries
```

public class **UpdateEntries**
extends java.lang.Object

Field Summary

public static	updateSize

Constructor Summary

public	UpdateEntries()

Method Summary

java.lang.String	startUpdate() This method will start the update

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

updateSize

`public static int **updateSize**`

Constructors

UpdateEntries

`public **UpdateEntries**()`

Methods

startUpdate

`public java.lang.String **startUpdate**()`

(continued from last page)

This method will start the update

Returns:

String confirmation

Package se.lnu.server.controller.parse

se.lnu.server.controller.parse
Class Parse_DB

```
java.lang.Object
  |
  +-se.lnu.server.controller.parse.Parse_DB
```

public class **Parse_DB**
extends java.lang.Object

Constructor Summary

public	Parse_DB()

Method Summary

void	cleanData() This method sets the values on default
boolean	isValid(java.lang.String text) This method checks if the String is a valid number or not
java.util.ArrayList	parseData(java.lang.String content, int arendeid) This method separates the attributes of a single project from each other and creates a new query to set the project details into the database

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Parse_DB

public **Parse_DB**()

Methods

cleanData

public void **cleanData**()

This method sets the values on default

isValid

public boolean **isValid**(java.lang.String text)

(continued from last page)

This method checks if the String is a valid number or not

Parameters:

`String` - number

Returns:

boolean

parseData

```
public java.util.ArrayList parseData(java.lang.String content,
         int arendeid)
```

This method separates the attributes of a single project from each other and creates a new query to set the project details into the database

Parameters:

`String` - content

`String` - arendeid

Returns:

ArrayList

se.lnu.server.controller.parse
Class Parse_HTML

```
java.lang.Object
  |
  +-se.lnu.server.controller.parse.Parse_HTML
```

public class **Parse_HTML**
extends java.lang.Object

Constructor Summary

public	Parse_HTML()

Method Summary

java.lang.String	formatInUTF8(java.lang.String orginal)
java.lang.String	parseHtml(java.net.URL url)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Parse_HTML

```
public Parse_HTML()
```

Methods

formatInUTF8

```
public java.lang.String formatInUTF8(java.lang.String orginal)
```

parseHtml

```
public java.lang.String parseHtml(java.net.URL url)
  throws java.io.IOException
```

Package
se.lnu.server.model

se.lnu.server.model
Class DB_Model

```
java.lang.Object
  |
  +-se.lnu.server.model.DB_Model
```

public class **DB_Model**
extends java.lang.Object

Constructor Summary

public	DB_Model()

Method Summary

static java.lang.String	getClassDriverJDBC()
static java.lang.String	getDatabasename()
static java.lang.String	getIp()
static java.lang.String	getPasswd()
static java.lang.String	getPort()
static java.lang.String	getTablenameAccess()
static java.lang.String	getTablenameBeviljat()
static java.lang.String	getTablenameDBConfig()
static java.lang.String	getTableNameDictionary()
static java.lang.String	getTablenameSource()
static java.lang.String	getUserForDB()
static void	setClassDriverJDBC(java.lang.String classDriverJDBC)
static void	setDatabasename(java.lang.String databasename)
static void	setIp(java.lang.String ip)

static void	setPasswd(java.lang.String passwd)
static void	setPort(java.lang.String port)
static void	setTablenameAccess(java.lang.String tablenameAccess)
static void	setTablenameBeviljat(java.lang.String tablenameBeviljat)
static void	setTablenameDBConfig(java.lang.String tablenameDBConfig)
static void	setTableNameDictionary(java.lang.String dictionary)
static void	setTablenameSource(java.lang.String tablenameSource)
static void	setUserForDB(java.lang.String userForDB)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DB_Model

`public DB_Model()`

Methods

getTablenameAccess

`public static java.lang.String getTablenameAccess()`

Returns:
the tablenameAccess

setTablenameAccess

`public static void setTablenameAccess(java.lang.String tablenameAccess)`

Parameters:
`tablenameAccess` - the tablenameAccess to set

getTablenameDBConfig

`public static java.lang.String getTablenameDBConfig()`

(continued from last page)

Returns:
the tablenameDBConfig

setTablenameDBConfig

```
public static void setTablenameDBConfig(java.lang.String tablenameDBConfig)
```

Parameters:
`tablenameDBConfig` - the tablenameDBConfig to set

getClassDriverJDBC

```
public static java.lang.String getClassDriverJDBC()
```

Returns:
the classDriverJDBC

getDatabasename

```
public static java.lang.String getDatabasename()
```

Returns:
the databasename

getIp

```
public static java.lang.String getIp()
```

Returns:
the ip

getPasswd

```
public static java.lang.String getPasswd()
```

Returns:
the passwd

getPort

```
public static java.lang.String getPort()
```

Returns:
the port

getTablenameBeviljat

```
public static java.lang.String getTablenameBeviljat()
```

Returns:
the tablenameBeviljat

getTableNameDictionary

```
public static java.lang.String getTableNameDictionary()
```

Returns:
the dictionary

getTablenameSource

```
public static java.lang.String getTablenameSource()
```

Returns:
the tablenameSource

getUserForDB

```
public static java.lang.String getUserForDB()
```

Returns:
the userForDB

setClassDriverJDBC

```
public static void setClassDriverJDBC(java.lang.String classDriverJDBC)
```

Parameters:
`classDriverJDBC` - the classDriverJDBC to set

setDatabasename

```
public static void setDatabasename(java.lang.String databasename)
```

Parameters:
`databasename` - the databasename to set

setIp

```
public static void setIp(java.lang.String ip)
```

(continued from last page)

Parameters:
`ip` - the ip to set

setPasswd

`public static void **setPasswd**(java.lang.String passwd)`

Parameters:
`passwd` - the passwd to set

setPort

`public static void **setPort**(java.lang.String port)`

Parameters:
`port` - the port to set

setTablenameBeviljat

`public static void **setTablenameBeviljat**(java.lang.String tablenameBeviljat)`

Parameters:
`tablenameBeviljat` - the tablenameBeviljat to set

setTableNameDictionary

`public static void **setTableNameDictionary**(java.lang.String dictionary)`

Parameters:
`dictionary` - the dictionary to set

setTablenameSource

`public static void **setTablenameSource**(java.lang.String tablenameSource)`

Parameters:
`tablenameSource` - the tablenameSource to set

setUserForDB

`public static void **setUserForDB**(java.lang.String userForDB)`

Parameters:
`userForDB` - the userForDB to set

se.lnu.server.model
Class SQLRequests

```
java.lang.Object
  └-se.lnu.server.model.SQLRequests
```

public class **SQLRequests**
extends java.lang.Object

Constructor Summary

public	SQLRequests()

Method Summary

static java.lang.String	createBeviljatTable() This method returns a query which creates a table for the the granted money , it contains the year(ar) and the money (pengar), which is related to the year
static java.lang.String	createDictionaryTable() This method returns a query which is currently not used but I leave it her for future work.
static java.lang.String	createMainTable() This method returns a query which creates a table in the database which holds the data of every project from the VetenskapsrÃ¥det
static java.lang.String	createUserAccess() This method returns a query which creates a table for the the user access , it contains the username and the password
static java.lang.String	getAccess() This method returns a query of all user and there password
static java.lang.String	getDataFromUniversity(java.lang.String uni, java.lang.String minValue, java.lang.String maxValue) This method returns a query which contains all data of a single university which is in the range of max and min of the range slider
static java.lang.String	getIDsFromDB() This method returns a String query which returns all project IDs in the database
static java.lang.String	getSumAllUniversitys(java.lang.String[] uni, java.lang.String minValue, java.lang.String maxValue) This method returns a query which sums all granted money of each university
static java.lang.String	setAccess(GWT_Access access) This method returns a query which sets a new user and his/her password into the database
static java.lang.String	setIntoBeviljat(int arendeid, int ar, int pengar) This method returns a query which inserts a new row with year (ar) and money(pengar) depending on the projectID(arendeID)
static java.lang.String	setIntoDictionary(java.lang.String value) This method is currently not in use but it insert a new entry into a directory table

static java.lang.String	`setIntoMain(int arendeid, java.lang.String diarienr, java.lang.String institution, java.lang.String instans, java.lang.String projekttitelsv, java.lang.String projekttitelen, java.lang.String beslutsdat, java.lang.String titel, java.lang.String namn, java.lang.String aemnesomrade, java.lang.String koen, java.lang.String klassifiering, java.lang.String beskrivning, java.lang.String vaerdhoegskola, java.lang.String bidragsform)` This method returns a query which insert into the database a new project

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

SQLRequests

```
public SQLRequests()
```

Methods

createUserAccess

```
public static java.lang.String createUserAccess()
```

This method returns a query which creates a table for the the user access , it contains the username and the password

Returns:
String query

createBeviljatTable

```
public static java.lang.String createBeviljatTable()
```

This method returns a query which creates a table for the the granted money , it contains the year(ar) and the money (pengar), which is related to the year

Returns:
String query

createDictionaryTable

```
public static java.lang.String createDictionaryTable()
```

This method returns a query which is currently not used but I leave it her for future work.

Returns:
String query

createMainTable

```
public static java.lang.String createMainTable()
```

(continued from last page)

This method returns a query which creates a table in the database which holds the data of every project from the VetenskapsrÃ¥det

Returns:

String query

getDataFromUniversity

```
public static java.lang.String getDataFromUniversity(java.lang.String uni,
         java.lang.String minValue,
         java.lang.String maxValue)
```

This method returns a query which contains all data of a single university which is in the range of max and min of the range slider

Parameters:

`String` - uni

`String` - minValue

`String` - maxValue

Returns:

String query

getIDsFromDB

```
public static java.lang.String getIDsFromDB()
```

This method returns a String query which returns all project IDs in the database

Returns:

String query

getAccess

```
public static java.lang.String getAccess()
```

This method returns a query of all user and there password

Returns:

String query

setAccess

```
public static java.lang.String setAccess(GWT_Access access)
```

This method returns a query which sets a new user and his/her password into the database

Parameters:

`access`

Returns:

String query

getSumAllUniversitys

```
public static java.lang.String getSumAllUniversitys(java.lang.String[] uni,
         java.lang.String minValue,
         java.lang.String maxValue)
```

This method returns a query which sums all granted money of each university

(continued from last page)

Parameters:

`String` - uni
`String` - minValue
`String` - maxValue

Returns:

String query

setIntoBeviljat

```
public static java.lang.String setIntoBeviljat(int arendeid,
         int ar,
         int pengar)
```

This method returns a query which inserts a new row with year (ar) and money(pengar) depending on the projectID(arendeID)

Parameters:

`String` - arendeid
`String` - ar
`String` - pengar

Returns:

String query

setIntoDictionary

```
public static java.lang.String setIntoDictionary(java.lang.String value)
```

This method is currently not in use but it insert a new entry into a directory table

Parameters:

`value`

Returns:

String query

setIntoMain

```
public static java.lang.String setIntoMain(int arendeid,
         java.lang.String diarienr,
         java.lang.String institution,
         java.lang.String instans,
         java.lang.String projekttitelsv,
         java.lang.String projekttitelen,
         java.lang.String beslutsdat,
         java.lang.String titel,
         java.lang.String namn,
         java.lang.String aemnesomrade,
         java.lang.String koen,
         java.lang.String klassifiering,
         java.lang.String beskrivning,
         java.lang.String vaerdhoegskola,
         java.lang.String bidragsform)
```

This method returns a query which insert into the database a new project

Parameters:

`String` - arendeid
`String` - diarienr

(continued from last page)

`String` - institution
`String` - instans
`String` - projekttitelsv
`String` - projekttitelen
`String` - beslutsdat
`String` - titel
`String` - namn
`String` - aemnesomrade
`String` - koen
`String` - klassifiering
`String` - beskrivning
`String` - vaerdhoegskola
`String` - bidragsform

Returns:

String query

Linnæus University
School of Computer Science, Physics and Mathematics

SE-391 82 Kalmar / SE-351 95 Växjö
Tel +46 (0)772-28 80 00
dfm@lnu.se
Lnu.se/dfm