Abstract

Over the past few years, a lot of research has been done on volcanic lakes. Research on this subject is important, as the chemical composition of volcanic lakes can clearly reflect upcoming eruptions, both in pre-eruption degassing of rising magma, as in intrusions into the convection zone of the hydrothermal system of such a lake. By having knowledge over such systems, eruptions may be predicted better, and the moment of evacuation may be placed more accurately. However, to be able to recognise the changes in water composition and to place it in the light of an oncoming eruption, reliable methods are needed to get such information from other, similar volcanoes. Only then, it will be possible to correlate the data to the activity of the volcanic system underneath the lake.

This thesis proposes a system to store and retrieve such data. While individual scientists and research groups make use of their own systems for data storage, a global system connecting all those individual systems is not readily available. Seeing the importance of readily available data, such a system can prove essential for the prediction of volcanic events.

A first version of this system is ready for release at the time of writing. A system providing the necessary functionalities for safely processing the data has been created. The system makes use of various methods to insure system and data integrity. It is therefore a safe and secure system, ready to be placed online, and provide the data tool that is needed.

Introduction

When a volcano erupts, liquid stone (magma) is pushed towards the surface. As the viscosity (resistance to flow) of such a liquid is relatively high, a lot of pressure is needed for this. When the eruption subsides, so does the magma pressure. This, combined with the overlying rock collapsing into the magma chamber can cause a caldera, or volcanic crater. When rainwater fills this caldera and it is not drained, a lake forms.

Being on top of a dormant (but not dead) volcano, the water from the lake interacts with the magmatic system in various ways. When the water intrudes into cracks in the rock, it can warm up due to the temperature of the magma chamber underneath the volcano. The warmer water rises, and makes room for cooler water, and eventually this generates a complete hydrothermal system. Other then heat, the rising water can also bring along dissolved minerals, changing the chemical composition of the water. Preceding the next eruption, volcanic gasses can be injected in the water as well. This changes the water's properties more radically than the every day interaction of the water with the underlying rock.

When an eruption is upcoming, the rising magma interacts with the water on two different ways.

The first way is when gasses escape from the magma, and get dissolved in the water. The second way is when the rising magma itself starts to interact with the hydrothermal system. These two steps are clearly visible in the composition of volcanic lakes.[1]

Changes in the water's properties can easily be measured by means of sampling the water and analysing those samples. Various methods of water analysis can be used. Naturally, temperature will have to be measured at the field, but chemical composition is usually analysed at the lab. Depending on what is known about the volcano's activity pattern, different elements and physical variables are tested for.

While each volcano is different, information gathered prior to an eruption of one volcano might prove very important in predicting an eruption of a different volcano. However, a system that gathers information on these volcanic lakes is not yet available. Therefore, testing if a volcano is near to eruption has to be done by guessing it's activity, rather then correlating data with that of other volcanic events. A system to test and correlate data can therefore prove vital in predicting upcoming eruptions. When considering that this also means that volcanologists can better pinpoint the moment evacuation is needed, available information can well save lives.

While sampling data on volcanic lakes is abundant, actually getting the data can be quite a hassle.[2] Data mining and machine learning techniques might be used on these forms of data as well, but for that, large data sets are needed. To ensure a fast and viable way to get that data, a central database in which sampling data is collected is needed.

Within the Earth Sciences community, a lot of work is put in large databases containing all sorts of data on various topics. An example of this is the GEOROC Database[3] in which is stored what kinds of minerals and rock types can be found at certain locations.

This thesis describes the work on a similar project, but then for information on volcanic lakes. The main question this thesis will focus around is: What is the most efficient way to completely store all available sampling data concerning volcanic lakes, in such a way that both storing as retrieving the data is a logical and streamlined process?

This thesis will answer the main question by describing the development process, outlining the problems involved and the considerations made. At first, we will define the difficulties the main question brings with itself. We will briefly look at the methods and tools used. After that, we will focus on the considerations made during development. Finally we will look at the changes and additions that can still be made to the system.

Problem Definition

To completely store all sampling data from all volcanic lakes where this data is available, the system must support all these kinds of data values. The system should support physical values such as *temperature*, *conductivity*, *salinity*, *density*, *pH-value* etc. Next to that, concentrations of all sorts of elements are measured as well. However, as different volcanoes require different approaches, not every volcano requires storage for the same element concentrations. To completely store them though, full support for all these elements is required.

Sampling data will of course have to be associated to a single volcano. And to actually make the data useful for predicting eruptive events, information about eruptions will have to be stored as well. for eruption data, it is important to store both the *starting date* as the *ending date*. As similar volcanoes often have a similar type eruptive style, the *length* and *intensity* of an eruption of a different volcano might say useful things about upcoming events from other volcanoes. Important for the intensity of the eruption is the volume of material erupted. In geological data, difference is made between material leaving the mountain in liquid form (*lava*) and material leaving the mountain by air, solidified upon landing (*tephra*).

As said, the dates of eruption are very important, to see how the water changes prior to an eruption, and to give an approximate time to eruption. Unfortunately, predicting eruptions is not an exact science, and estimations about time to eruptions and the approximate intensities are mostly not very accurate. The more information is available, though, the more educated the guess can become.

Methods

For the development of the system, a standard software engineering approach is used. After the requirements (appendix 1) for the system were established and motivated (appendix 2), a database design (the Entity-Relationship diagram from Fig 1) and a system model (the UML diagram from Fig. 2) were made.

After the models were established, programming could begin on the system. The interface was placed in a web environment. A separate program would have been possible too, but a web environment is easily spreadable. Seeing that similar databases are put in a web environment as well, this option was then selected. For the development, the language PHP (version 5.3.6) was used.

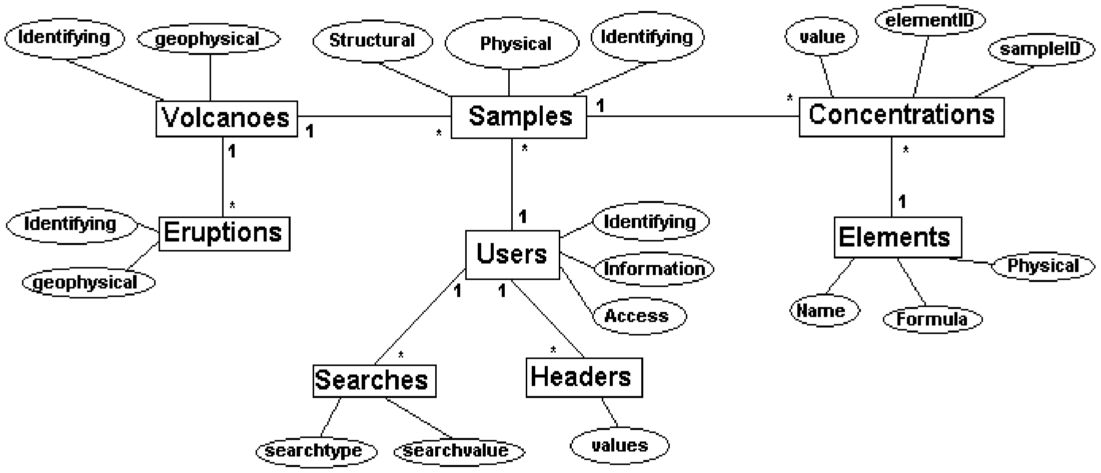

Fig 1: Simplified version of the ER-Diagram

The central table is the samples table. This contains all physical values about a sample, as w ell as the identifying variables such as the volcano it belongs to, the time and date and the depth of sampling. A sample record is defined by a unique sample ID number. The volcano table is identified by the volcano numbering used by the Smithsonian Institution. Other identifying variables for the volcano include the coordinates and the height. Geophysical variables include the sizes and volume of the lake. For an eruption, the time and date of begin and end are stored. A separate integer is used to identify the eruption, but each eruption is associated w ith a volcano. Geophysical eruption data include eruption type and magnitude. The Users table contains the identifying number, each user account has a unique ID number and nickname. The passw ord w ith w hich a user is logged in is stored in an encrypted w ay. Information about the user contains background information, email address and if he allow s other regular users to send him email. The user's access level is defined here as w ell, and the user can see various parts of the system based on this. The searches table contain the searches that a user has conducted on the system and has saved. Important to note here is that there are various types of searches. The headers table is used w hen defining headers for the input process.

The tw o other tables concern the concentrations of the various elements found in the w ater. All elements used are stored in the Elements table. This includes their name, chemical formula and physical variables as molar mass and electrovalence. The concentrations table is a cross table betw een elements and samples. A concentration record is therefore defined by both it's Sample ID as it's Element ID – and in case of a ratio even by it's second Element ID.

PHP allows for easy and effective page customisation and interfacing with a database. As Database Management System, MySQL was selected. Both because of the free availability, as well as the complete interface between MySQL and PHP. To administer the database for easy set up, the package LAMP (Linux Apache, MySQL, Php) was used. Furthermore, the PhpMyAdmin was used as well. The PhpMyAdmin is a tool that allows easy setup and administration of the raw data from the database.

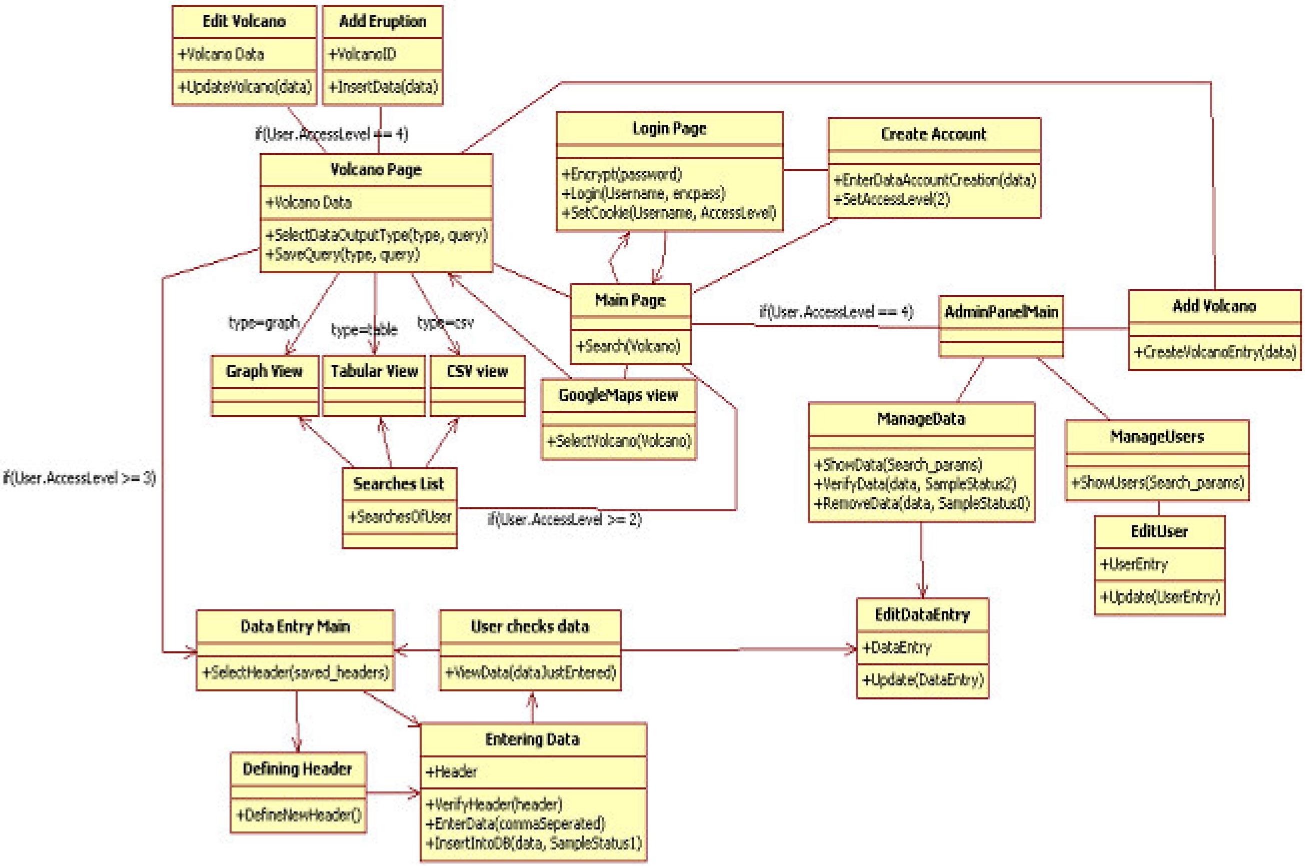

Fig 2: The UML web structure. The central page is the Main Page. This page contains information to search for a volcano. The other pages are presented in a module-based structure. Just above the Main Page is the account management module. In the final design, this module also includes profile editing and finding other users.
Just below the main page is the Google Maps view, using the main page or this view, users will be able to select a volcano to get to the volcano page. This page is located just left of the main page. Output modules are located just below the volcano page. The input module, containing all the various steps of the input process is below that. Above the volcano page and to the right of the main page are the administrative modules.

Summary and Outlook

A requirements document has been completed, and from that an ER diagram an a UML model of the system have been derived. At of the time of writing, the system is fully functional. All essential functionality has been added and tested.

Future Work

There are a few points on which this project can still be improved upon.

- The creation of the header can be improved with the use of JQuery. This is a JavaScript library allowing for more interactive elements on a web page. This would allow for the header to be defined the way it is outlined in Requirement 1.3.2.3. However, as the JQuery library proved to be more difficult then expected, and an other method was available, dynamic definition of the header is moved to future work.
- Related to the above point: as of now, editing a record can only be done by rejecting it and resubmitting it. However, being able to edit a single record would be very useful as well. Unfortunately, to prevent lengthy and confusing editing procedures, JQuery will have to be used here as well.
- Searching for a volcano can be done in different ways. One of the planned ways was by using Google Maps to navigate to a volcano marker. However, as functionality was more important then this, the Google Maps feature was also moved to Future Work.
- An other desirable but not mandatory feature would be adding support for the imperial units. Though this could make input easier for those that make use of it every day, at the moment, only SI units are supported.
- As it is now, the Administration Panel is rather redundant. As it turned out, most functions can be reached through other ways as well. Making the Administration Panel more useful though, requires reconsideration of the web structure. This is not a straightforward task, so it will be future work as well.
- At the moment, all interfacing with the data is done through comma separated file formats. Most spreadsheet programs are able to output and input this, so there is an interface with them. However, as it is still not a straightforward way to do so, this process will have to be smoothed out as well.
- One feature that will have to be added is support for multiple date/time fields in the header. Some researchers might use different columns for different date values (a column for the year, one for the month and one for the day, for example). As of this moment, the system does not support this.
- When there is sufficient data in the system, it might be useful to work on some Data Mining and Data Analysis modules as well. However, as this fell beyond the scope of the project, this has not been done yet.
- Extensive testing and “trollproofing” (a “troll” is defined as a malicious user on the internet) will have to be done as well. As a large part of the code is only tested for it’s functionality when it’s used as intended, there is a possibility that malicious users might be able to compromise the integrity of the system or of the data. This is not desirable. However, this process requires more time then was available.
- While the system is fast enough as it is, more users mean more load on the system. It is therefore a goal to optimize the programming such that pages are loaded as fast as possible.

  A few examples of this are the function to recalculate one value in an other. This function currently is very big and contains a number of redundant lines. An other example is the function that lists the options in a search. At the moment, this function makes a number of queries to the database, and is called for every drop down menu. This can be improved by executing the function once and saving the result.

  Also included at this point is better commenting of the code. As the system will need to be maintained, it is inevitable that there will be people working with the programming code. Commented code will make sure that the code is easier understandable.

It is our hope that this system will be able to contribute to a better understanding of the hydrothermal systems in volcanic lakes, and that it will eventually lead to more accurate predictions of upcoming volcanic events.

Acknowledgments

At first, I want to thank Dr. M. Emmerich and Dr. M van Bergen, my supervisors, for the help and insight that they have provided. Secondly, I would like to thank Jaron Viëtor and Carina van der Meer for their support in the programming.

I enjoyed working on this thesis, and I enjoyed being able to help the scientific community further through it.

References

[1] Christenson, B.W. - Geochemistry of fluids associated with the 1995-1996 eruption of Mt. Ruapehu, New Zealand: signatures and processes in the magmatic-hydrothermal system - Journal of Volcanology and Geothermal Research, volume 97, issues 1-4, April 2000, pages 1-30
[2] Varekamp, J.C., Pasternack, G.B. and Rowe Jr, G.L. - Volcanic lake systematics II. Chemical constraints - Journal of Volcanology and Geothermal Research, volume 97, issues 1-4, April 2000, pages 161-179
[3] http://georoc.mpch-mainz.gwdg.de/georoc/Start.asp per July 31, 2011
[4] http://www.volcano.si.edu/world/volcanocriteria.cfm#VolcanoNumber per August 12, 2011
[5] http://www.geology.sdsu.edu/how_volcanoes_work/Variability.html per August 15, 2011

User Manual

Unregistered / unlogged in users

Finding a volcano

From the home page, select a volcano by it’s volcano name, or by it’s lake name. Alternatively, you can select a country and see all volcanoes from that country. If there are several entries for one volcano, you will see a page listing basic information about those volcanoes, and giving the possibility to see the full record of each of those.

The volcano page

On the volcano page, you can see all information available about a volcano. This includes eruption data. Data elements that are not visible on this page has not been filled in. From this page, sampling data can be accessed.

Accessing sampling data

To access the sampling data, click one of the three buttons on the volcano page. You can either search for data put in a table, data put between commas (Comma Separated Values or CSV), or you can view a graph from data. Select the data columns that you want present in your data. If needed, you will be asked to specify some of the columns (this concerns the units in which you want the data to be output). After this has been done, the data will be made visible.

Importing data in Microsoft Excel

To import data in Microsoft Excel, make a CSV search first. When you have the CSV data, you will see it presented in a text area. Copy everything from that area. Paste it into a plain text editor (e.g. notepad), and save it as *yourname*.csv. Then open this file with spreadsheet program, and you can copy the rows into your main spreadsheet.

Registering an account

To register a new account, select “register” from the menu or from the home page. Fill in the form. All fields except your background information are mandatory. Your email address will be displayed publically if you select to allow others to send you email. However, other unregistered users will not be able to see your email address. Whichever option you select, the Administrator will always be able to view your email address.

Logging in

To log in, select “Log in” in the menu. Fill in your username and password. If the login is successful, you will be redirected to the home page.

Registered users

Saving search

During the making of a search, you will see an option to name your search. If you give your search a name, it will automatically be saved for further use. When naming your search, make sure that you only have one search with the same name and the same type. For instance, you can have both a tabular search and a CSV search with the same name, but no two tabular searches with the same name.

Editing your profile

You can edit your profile directly from the link in the menu, or you can navigate to your profile page using the Find a User link and navigating to your profile. Here, you can edit the information on your profile. You can not edit your username - only an Administrator can do that. Naturally, you can not edit your rank as well. To edit your password, you will have to fill in your old (current) password, and your new password twice - to make sure there are no typing errors involved. To edit your email address, you will have to fill in both email fields. Leaving the second field empty tells the system you will not be changing your email address. All other fields can be freely edited to your liking.

Recognized / Verified users

Defining a new header

To define a new header, first go to the volcano page. Under the record, click the button to add data to the database. On that page, you can select the header, and also make a new one. Select the option to make a new header.

First, select the header name. This name has to be unique to you. This means that there can be other users with the same header name, but you can not have two headers with the same name. In this step, you can also select the amount of columns in your excel file. This amount can not be edited later on.

In the second step, select the column types of your excel file. If you have a column that is not in the menu, it is not yet supported.

In the third step, you can select column specifications. In case of the element concentrations, you will be asked to select the unit in which you have your data. This is also the case for temperature and density.

For the Date / Time field - the only field that is mandatory in the header - you will have to make the format in which your Date / Time field is. For this, a shorthand code is used.

Year	**Ex.**	**Month**	**Ex.**	**Day**	**Ex.**	**Hour**	**Ex.**	**Min.**	**Ex.**
		MMM	Jan						
YYYY	2004	MM	01	DD	07	hh	09	nn	08
YY	04 *	M	1	D	7	h	9	n	8

* It is assumed that when using this notation, the date falls within the past 100 years.

With months, days, hours and minutes, two letters mean with leading zeroes (for months, MM means 01 through 12) and one letter means without leading zeroes. MMM is three-letter month notation (Jan - Dec). Every other character is read as the format in which the data is in. Some examples:

YYYY-MM-DD hh:nn	2004-01-07 09:08
MMM DD, YYYY	Jan 07, 2004
D M ‘YY - h.nn	7 1 ‘04 - 9.08

Unfortunately, support for two separate date/time fields is not built in.

Editing a header

When you have headers defined, you are also able to edit them. This uses roughly the same steps as for defining a header. However, editing the name and defining the columns is one step, and the amount of columns is not editable. The button to edit a header is in the table displayed on the page where you select the header to use.

Deleting a header

When a header is no longer needed, it can be removed. However, this can not be undone. To make sure you don't accidentally delete a header, you will be asked if you really want to remove the header.

Adding data

To add data, you must first have a header defined. From the page that you can reach by clicking the Add Sampling Data button from the volcano page, you can select a header to use. Then you will be lead to a page with 2 fields. One big data field, and one small field marked "Seperator".

The best way to submit the data is to copy the data directly from your excel file, and paste it in the field marked with Data. Then select the character that separates the fields from each other. Paste this character in the field marked with Separator. This ensures that the system can read your data well.

When this is done, you will be lead to a different page, displaying all the data that you have just submitted in a tabular form. From this, you can check if the data entered is correct. There will be a few operations available at this point. You can either edit one record (Editing multiple records will have to be done step by step), reject the entry (All records will be removed) or accept the entry and continue. You will then be lead back to the Volcano Page.

Your data will be usable by you immediately. However, before it is available for use everywhere, the Administrator will have to validate the data. This basically means that the Administrator will have to check if the types of data entered match the expected columns according to your header. Until this has been done, however, you need to be logged in to be able to view your data and work with it.

Administrators

Editing a user profile

To edit a user profile, first navigate to the profile by using the "Find a user" link from the menu. Using this page, you can select all users with their name starting with a certain letter. Then view their profile using the button on the line of their user record. From this, you can edit their profile as admin. On the page - which is similar to the profile page where you edit your own profile - You can edit their data. Though the page looks like the regular edit-profile page, there are some differences. As a first one, you are able to edit a user's username. The username still has to be unique, and the system checks for this. To edit their password, you do not need to know their current password. You do, however, have to type in the new password twice, to make sure there are no typing errors in it. The last difference is that you can edit a user's rank. This change is immediately, and you will have to take into account that the user can immediately perform the tasks that people of their rank are able to do.

Note: You will be able to edit your own rank as well. The moment you take yourself off the rank "Administrator", you will not be able to revert this. Only an other Administrator can set your access back.

Adding a volcano

To add a new volcano record to the system, you can use the link from the menu. The Smithsonian Volcano Label is the volcano numbering according to the Smithsonian Institution. Please make sure that you have this correct, as you will not be able to edit that label later on. On the form, all values marked with * are mandatory. All other fields can be left empty.

It is also possible to upload an image of the volcano. This image will then be displayed on the volcano page. This image will be uploaded directly to the server.

After successfully submitting the form, you will be taken to the volcano page of that volcano.

Editing a volcano record

When a mistake is made in a volcano record, this can easily be edited. The only uneditable value is the volcano label. When an image is already uploaded for the volcano, uploading a new image will overwrite the old one. The old image will then be lost.

Adding an eruption

Adding an eruption is also done from the volcano page. Go to the page to add an eruption using the button to do so. The form you will have to fill in has one mandatory field - this is the starting date. When the starting date is not known exactly, an estimate may be provided. Like is stated on that page: If any of the date and time fields are unknown, please leave it to the first value or - in case of the starting year – provide an estimate. The ending year may be left empty.

If filling in the ending date is not needed, please leave all fields empty. If any field is filled in, all fields must be filled in.

Other fields from this page may be left empty or filled in according to the data available.

Editing an eruption

Editing an eruption is done in a similar way as adding it. Once again, the starting date and time are mandatory, and the ending date and time should either be completely filled in or completely left empty.

It is therefore possible to add the eruption when it begins, and edit in the ending date later, when the eruption is added. This way, the data on eruptions is kept actual.

Validating data

To validate data, you will see a similar screen as the user sees just after the data is entered. However, there is a difference. While the user only sees the “batch” of data he has just entered, you will view all the data that has not been validated. This data is sorted by batch. Every batch has their own table with the appropriate header. Per batch, you can validate or reject all data. You can also opt to edit a single record. Furthermore, you can validate or reject all data in one go.

System administrator

Setting up the system

To set up the system, all files from the repository (listed in Appendix 3) will have to be present.

The following list of software will have to be installed:

- A webserver (Apache is used in the development stage, but any webserver will do)

- MySQL (MySQL is recommended as every reference to the database in the code is by the MySQL interface commands)
- PHP
- Recommended: PhpMyAdmin (for easy administration of the database).

First, to set up the database, make a user with the username and password that are specified on line 6 of header.php. The mysql_connect() command requires as arguments: location, username, password.
Set up a database whose name is specified in line 7 of header.php. Grant the user that you have made in the previous step the following rights over the database:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- ALTER

Run the file volcaniclake.sql. This sets up the database completely, so that it is ready for use.

Next, configure the filesystem permissions so that the webserver has reading rights over the folder in which all files are located. Create a folder within this folder named “images”, and give the webserver writing rights in this folder as well.

Once these steps are finished, configure the router so that the server can be reached as webserver. Register a web address name for as far as this has not yet been done.

Now, the reCAPTCHA, used in the registering of an account will have to have it’s keys adapted to the new web address. For this, visit http://www.google.com/recaptcha and register the web address under “my account”. A public and a private key will be generated. Copy the public key into the $publickey variable in register.php, line 4. Place the private key in the line below.

Once this has been done, the system is ready for use.

Appendix 1

Requirements Document V3

Scope of the project

In the scientific studies of volcanoes, there is quite some information about crater lakes. However, this information isn't stored in one central spot, and in a unified manner. This makes it difficult to draw conclusions from the data, conclusions that can be vital in predicting upcoming eruptions, and thus saving lives.

This project has the purpose of setting up such a central spot and unified manner of storing the data collected from measurements on crater lakes all around the world. This requires the use of a database, and a usable and workable interface. This document lists those requirements needed for this project, and a motivation for the requirements themselves.

Stakeholders and actors in the requirements process

For the establishing of the requirements, different conversations have been conducted with both Dr. van Bergen as Dr. Emmerich.

The following stakeholders have been identified.

Volcanologists who enter the data. These are the people that enter the data into the database. Their primary concern is the speed and efficiency with which they can enter their data.

Volcanologists who use the data. These are the people who want to make use of the data found in the database. It is important for these stakeholders that the data that they find is correct, and easily findable.

Project administrator. This is the person or organisation responsible for both keeping the system up and running, and maintaining and checking data integrity.

Glossary of terms

mg/L: Unit value. Milligrams per litre. Also a value for density.

mol/L: Unit value. Moles per litre or Molarity. This unit value is concerned with the amount of particles in a certain volume.

PPB: Unit value. Parts per billion. Mostly used as weight-ppb.

PPM: Unit value. Parts per million. Mostly used as weight-ppm.

Sample: Water sample, taken from the water, and analysed.

Smithsonian volcano label: All volcanoes are labeled by the Smithsonian Institution, and these labels are accepted as a uniform identifier for a volcano.

Volcanic lake: Other name for crater lake.

WT%: Unit value. Weight-percent. Percentage of weight taken by the element, relative to the total weight of the water volume.

Requirements

Functional Requirements

<u>1: Storing data</u>

1.1: Requirements on data to be stored

1.1.1: Data stored should be per sampling of water from a crater lake. Values stored are:

- Identifying values

   - Smithsonian volcano label (cross reference to the volcano table)
   - Time and date of sampling
   - Depth of sampling (stored in meters below water level)
- Data values
   - Chemical composition of the water (see 1.1.1.1)
      - major cations
      - major anions
      - dissolved gas species
   - Physico-chemical values
      - pH-value (both in the field as in the lab, at 25 degrees Celcius)
      - Electrical conductivity
      - Density (in the field and in the lab)
      - Temperature of the water
      - Air temperature
      - Water salinity
   - (if known) Date and time of last eruption before the measurement
   - (if known) Date and time of first eruption after the measurement
   - Analytical techniques used
   - Remarks concerning field observations
   - Other remarks
   - Data source: literature reference, unpublished report, personal communication etc.

Of the concentrations and temperatures, the following will be saved:

- Original value
- Original unit
- Calculated value in a standard unit

1.1.1.1: Breakdown of the chemical values that need to be stored

Anions

S total	SO_4	$S_4O_6^{2-}$	$S_5O_6^{2-}$	$S_6O_6^{2-}$	HCO^{3-}
NO_{3-}	Cl	F	Br	I	

Dissolved gas species

SO_2	H_2S	CO_2	O_2

Cations

Ag	Al	As	Au	B	Ba
Be	Bi	Ca	Cd	Ce	Co
Cr	Cs	Cu	Dy	Er	Eu
Fe total	Fe^{2+}	Fe^{3+}	Ga	Ge	Hf
Hg	Ho	K	La	Li	Lu
Mg	Mn	Mo	Na	Nb	Nd
Ni	P	Pb	Pr	Rb	Sb
Sc	Se	Si total	SiO_2	Sm	Sn
Sr	Ta	Tb	Th	Ti	Tl
Tm	U	V	Y	Yb	Zn
Zr					

1.1.2: Data should be stored about the used volcanoes as well:

- Smithsonian Volcano Label [4]
- Name of the volcano
- Name of the volcanic lake
- Country the volcano is in
- State/province
- Optional: image
- Longitude and latitude
- Altitude of the volcanic lake
- Surface area of the lake
- Volume of the lake
- Maximum diameter of the lake
- Minimum diameter of the lake
- Lake colour
- Maximum depth of the crater lake
- (If known) Date and time of last eruption

1.1.3: The data about eruptions:

- Date and time of the begin of the eruption
- Date and time of the end of the eruption
- Magnitude of the eruption on the VEI-scale [5].
- Type of eruption
- Volume of lava erupted
- Volume of tephra erupted
- Notes about the eruption

1.2: Requirements on how to store the data

1.2.1: Data should be validated by administration before being able to be used.

1.2.2: Data is stored in the database by means of a web interface.

1.2.3: Data on chemical composition can be delivered in the following units (but will be stored in a unified manner in the database):

- mg/L [kg/m^3]
- mg/kg [kg/kg]
- µg/L [kg/m^3]
- µg/kg [kg/kg]
- ppm (parts per million) [kg/kg]
- ppb (parts per billion) [kg/kg]
- ppt (parts per trillion) [kg/kg]
- mol/L [mol/m^3]
- mmol/L [mol/m^3]
- µmol/L [mol/m^3]
- mol/kg [mol/kg]
- mmol/kg [mol/kg]
- µmol/kg [mol/kg]
- meq/L

1.2.4: Data on temperatures can be delivered in C, F or K, but will be stored in a unified manner in the database.

1.3: Requirements on the storing process

Storing data is divided into two subcategories, volcanoes and sampling data.

1.3.1: Volcanoes

1.3.1.1: Data storage on the volcanoes will be done by the Administrator (as defined in Non

Functional 2.1.4).
1.3.1.2: There will be a webpage form in which the Administrator can enter the required data (1.1.2) to create a record about a volcano.
1.3.1.3: From a volcano page - reachable from the webinterface's home page - the Administrator can push a button to edit the information about a volcano.
1.3.1.4: From a volcano page, the Administrator can push a different button to make an entry about an eruption of that volcano. The data required is mentioned in 1.1.3

1.3.2: Sampling data
1.3.2.1: From the volcano page - which is viewable for everyone - a Recognised User (as defined in non-functional requirement 2.1.3) can access the data input page for that volcano.
1.3.2.2: As most scientists use Microsoft Excel or similar programs to store their data, the system will support .csv input (comma seperated values).
1.3.2.3: The system will allow the user to dynamically define the header of the .csv file. The process is as such:

1. The user selects a field from a drop-down menu
2. If the field in question is a field of which different input units are accepted, an other drop down menu will appear, allowing the user to select the desired unit.
3. An other drop down menu will appear, allowing the user to select an other field, continuing in the same loop

The user is able to save his header for further usage.
1.3.2.4: Mandatory data fields are:

- Date of sampling (time is optional)
- Depth of sampling

The system will not accept the header if these two values are not selected.
1.3.2.5: Once the header is defined and accepted, the user will find a text box, in which the data content of the .csv file can be pasted.
1.3.2.5: The data will be parsed by the system, validated (to see if the data values match the expected input types as defined by the header) and entered into the database.
1.3.2.6: The user will see a confirmation screen (table output as defined in req. 2.2) in which he / she can check his data.
1.3.2.7: The Administrator will have to validate the data before it is added into the general database. However, the user inputting the data will be able to use it already.

1.4 Other requirements on data storage
1.4.1: The database should have 3 tables concerned with the storage of the sampling data. The first table is the main table for the samples, the second table is a table with the different elements that can be referenced to. The third table is a crossed table between the second and the first, storing the concentrations.

1.4.2: For reference of the sample in the third table mentioned in1.4.1, the sample itself will have an ID number attached to it, as will the element itself, to be able to handle different occurrences of the same element.

1.4.3: Other then the tables used for storing sampling data, the database will have the following tables:

- Volcano table, with all elements mentioned in 1.1.2
- Eruption table, linking to the Volcano table, with all elements mentioned in 1.3.1.4.1
- User table, with all information about a single user, including permission levels
- Searches table, in which searches saved by users are stored.
- Data header table, in which the data headers as defined in 1.3.2.3 are stored.

2: Retrieving the data

2.1: Every user should be able to retrieve the data by means of the web interface.

2.2: Data can be output in various ways:

- Tabular. The web interface provides a table with the requested values. Similar to 1.3.2.3, the user will be able to select the columns of the table, and place constraints on them.
- Comma-separated. The web interface provides a .csv output, readable by Microsoft Excel and similar programs. Once again, a header and constraints can be determined.
- Graph. The web interface provides graphs with trend-lines on the requested values. These trend lines can be generated from single values, or ratio’s (to cancel out weather interference)
- Google Maps. The web interface shows the locations of the volcanoes on Google Maps, linking to the queries associated with that volcano

2.3: There will be a page displaying information about a volcano, with all data (including any known eruptions) presented in a user friendly way. From this page, searches can be conducted about sampling data.

2.4: Users with a user account will be able to save and name their tabular, comma-separated and graph searches. These searches should also be reachable from the home page, without accessing the volcano page first.

3. Working with the data

3.1: The chemical values - while being stored in a unified manner - should be retrievable in all previously mentioned units (See 1.2.3)

Non functional requirements

1: Safety

1.1: Users should not be able to delete data-entries, only Administrators should be able to do this.

1.2: To prevent accidental deletion, data should still be retrievable after deletion.

2: Security

2.1: The system allows for creation and management of user accounts. For this, four types of users are recognized:

1. Unregistered user - The user has not logged in or not created an account
2. Registered user - The user has made an account and can save searches
3. Recognized user - The user has made an account, can save searches and is allowed to enter data. Data should be validated by an Administrator first.
4. Administrator - The administrator can save searches, is allowed to enter, remove and validate data; and can change the permission level of a user.

2.2: Users will have to be recognized by an Administrator as scientist collecting data to input in the database, before being allowed to do so.
2.3: Administrators will be able to remove data, but should only do so if the data is faulty or entered in malice, and they should be warned before doing so.
2.4: The Administrator can assign other Administrators
2.5: Administrators will have an environment of their own to do their administrative tasks. There will be a user panel, where users can be selected and their permissions altered, and a panel where unvalidated data can be validated.

3: Usability

3.1: The system should be usable by relative inexperienced computer users.
3.2: For 3.1 to go into effect, the system should provide help functions on every page, outlining the use and usage of the page.
3.3: At any moment, it should be clear to the user where he is in the system, by using “breadcrumbs” - navigational links on the top of each page

4: Reliability

4.1: The system should be foolproof, data integrity should be preserved.
4.2: Malicious users should be able to be dealt with, by removing the data they input, as well as having their account deactivated.

5: Performance

5.1: The system should be able to handle concurrent accesses and entries of data while keeping data integrity.
5.2: Access to the more sophisticated parts of the system (graph generation and interfacing with Google Maps) should not take more then a reasonable time. Even with many users making use of the system, generating a graph should not take more then 10 seconds.

6: System requirements

6.1: The system should be a webpage-based interface.
6.2: The system should be compatible with the most recent versions of the most used browsers (Internet Explorer, Firefox, Opera)
6.3: The system should be programmed using PHP, Javascript, HTML, CSS, SQL and PHP-integrated SQL
6.4: The database system will be MySQL.

Appendix 2

Requirements Motivation

This document deals with motivating the requirements, outlining the considerations and choices being made.

Functional Requirements
1.1.1: This requirement deals with the data values that need to be saved on the samples. The identifying values are pretty much self explanatory and identify a single sample. As multiple samples can be taken at the same time but from different depths, this needs to be a function as well.
In the conceptual model of the database, there is a single field with SampleID, as this is a more convenient way of referring to a single sample, for instance in the concentrations table.
The actual data values are mostly stored in the concentrations table, in order to reduce redundancy. Physical data will be stored in the samples table.

1.1.2: This requirement deals with the values that should be stored about volcanoes. The Smithsonian Volcano Labeling is the international standard for volcano labeling, and thus will be used as identifier for the volcano - both in this table as in other tables referencing the volcano table. The distinction between country and state/province is made because in large countries, the type of volcano often depends on the geographical location, adding the state or province allows for a better chance of putting the volcano in it's geographical location - for as far the longitude and latitude don't do so already.
The volcano table itself will not reference the eruptions, but the eruptions table references the volcano. The volcano table will - however - reference the date and time of the last eruption, for copying into the sample table if needed.

1.1.3: The main consideration that had to be made here was the distinction between lava and tephra. As tephra is the volcanic material that gets shot into the air and solidifies there (volcanic bombs, ash, etc), and lava the volcanic material that still has some form of flow on the ground, this distinction is made. This distinction is also made in many studies on volcanoes.

1.2.1: The administration is responsible for data integrity. Their means of checking is intentionally left vague, as this is more a procedure question then a development question.

1.2.2: A webinterface is a type of interface building that matches the developer's experiences, and that also allows for multiple concurrent accesses.

1.2.3 / 1.2.4: As different people use different units for different values, they all need to be captured. However, to make a good graphical representation, the data should be stored in a unified manner.

1.3.1: These records will be fully entered by the administrator of the system. The administrator does the data entry on volcanoes, and registers eruptions. The entering of a new volcano will be done from the Admin Panel. However, to ensure the right volcano is being edited, the editing of a volcano, and the adding of an eruption will be done from the volcano page. The admin panel will link to this page if a new volcano is made.

1.3.2: The most important consideration here is in the creation of the header. As every scientist names his columns differently in their spreadsheet files, it will be quite hard for the system to read those correctly. The scientist will have to make his header manually at first, to prevent ambiguity. However, as the process of creating the header is quite lengthy, the user can select the header to be saved. The user can also name the header.
The data itself is then entered in .csv format. This method is selected, because it is the most easy format to work with, considering most scientists use spreadsheet programs to store their data in. Exporting to csv is a function that all those programs support. In time, it might be possible to directly interface with .xls files, but that is quite some work, and therefore not in the scope of this project.
The Administrator will have to validate the data, after it is entered, however, this process should be fairly easy - from a system’s point of view. An important consideration made in 1.3.2.7, is that the user should be able to use his data directly. As the administrator has not yet validated the data, other users can not be sure of the validity of this data. However, the original user, the one that entered the data, can be, and should be able to use his own data immediately.

1.4.1 / 1.4.2: A critical descision here was using three tables instead of one. As not all scientists always measure all elements, if all elements were one row of the database, there would be a lot of redundancy here, as many of the cells would be empty (but the space would be reserved). The three tables does mean more tables in the database, but the overall file size of the database itself would be a lot smaller.

1.4.3: These tables are those that were deemed necessary for the system to run smoothly. This includes tables to save the headers (as discussed in 1.3.2), and a table to save a user’s searches. There seems to be no reason to limit the amount of headers and searches a user can save.

2.1: As data on this should be freely available, every user, even those who are not logged in should be able to have full access to the data.

2.2: The four ways of displaying the data were considered to be the most useful ones. Although the Google Maps output type isn't an output type as the others. On Google Maps, markers are shown for each volcano where information is available on, leading to the volcano page of the volcano selected. The sampling data will be displayed in any of the other three data output types. CSV output is done so that uses can easily import the data into their own spreadsheets. Studying of the tables is done by selecting the tabular view. And if the user wants to make a graph of a certain concentration, or some other value (supported are: concentrations, pH values, temperatures, conductivity and density), this function is supported.

2.3: The layout of the volcano page like is greatly inspired by the layout of the volcano pages of the Smithsonian Institution. Other then the information that is known about the volcano itself, the volcano page will also feature the known data on the eruptions, in chronological order with the oldest one being on top.

2.4: Although an account will not be required for data access, having an account should come with some benefits, even if the account is not validated to enter data. The obvious choice then is letting people save their searches, as search queries could take a while to configure to a user’s wishes, and it can be assumed that users will probably want to reuse their searches. As the Google Maps view isn’t really suited for saving searches on, the searches will only be on tabular, graph or csv searches.

3.1: It was said before that different scientists have different methods of working with their data, and thus want different units on the numerical data. The function that calculates the entered value in the corresponding value in the standard unit, should be able to calculate the values corresponding to all other units as well.

Non Functional Requirements:

1.1: To prevent users going off to delete data, only the administrator should be able and allowed to do so.

1.2: Even the administrator makes mistakes, and it is very well possible that entered data is not validated although it is valid. Deleted data should therefore be able to be restored.

2.1: Account management is one of the key elements in creating a safe web environment. As we also recognise unregistered users, they are a part of the account management system. Furthermore, a tight structure is needed for a project like this.

2.2: The most important consideration in this project is data integrity. The data needs to be good, and for that, a good structure is needed. Not every account should be able to enter data, and

those that should be allowed to do so, should be manually selected by the administration, to keep track of everything.
2.3: Of course good data should not be removed. There should be safeguards in place to prevent accidental deletion.
2.4: Administrators should be able to chance the access level of any user. This implies that an administrator can set other administrators and remove his own rank as well.
2.5: The admin panel is a part that is shielded off from the normal view, and is essentially separated from the rest of the website. It is the environment in which the administrator does it's job. To make sure the administrator can find it's way around, all administrative pages are focused here.

3.1: As various degrees of computer proficiency can be expected among the people working with the system, the system should be understandable by those with a low level of computer proficiency as well.
3.2: Help functions can outline the difficulties in a page, and provide a solid and clear structure to the users.
3.3: Breadcrumbs are small links indicating the position in the system (for instance Main Page > volcanoes > Askja > Tabular). They can be essential for users to find their way around the system, and to maintain a view on it's structure.

4.1: In a system like this, data integrity is perhaps the most important requirement of all. There should be safeguards in place to maintain data integrity
4.2: As the system will be hosted on the webinterface, and the internet does have quite a few malicious users, the scenario of a malicious user attempting to compromise data integrity will have to be taken into account. Both the user account as the data the user entered should be able to be dealt with accordingly.

5.1: As the system might be used world wide, and by many people at a time, it is important that concurrent accesses can be handled, and that this doesn't give any problems with the data, both in entering data as in displaying it.
5.2: This is more of a requirement on server strength, and is deliberately left vague. The 10 second limit is set, as it is proven that most users view 10 seconds as a maximum acceptable waiting time for a website. Google Maps integration might take some more time, especially on slower PCs, but the user will know this problem, most likely. However, as graph generation has to be done completely serverside, it needs to be possible to handle various different accesses to that system at the same time.

6.1: A web system is both a relatively simple interface to build upon, as well as a very versatile one. And on top of that, it doesn't require any special downloads for the system to work, making it a good environment for this system.
6.2: As different people use different browsers, compatibility issues should be resolved.
6.3: This is a short oversight of what languages are needed in order to accomplish this system.
PHP: This is an HTML preprocessor language. It can run scripts to dynamically define a

webpage before it is transmitted to the user.
JavaScript: This is a script that runs on the PC of the user during the display of the webpage, to dynamically alter the page during it's displaying.
HTML: The basic markup language that all internet pages use.
CSS: Language to define the style of a page, for uniform looks of all the pages.
SQL: Query language to interface with the database.
PHP integrated SQL: Using PHP to define SQL queries and send them to the database.
6.4: MySQL is a free database system with which I have experience already. The MySQL package also comes with clear and ready functions for PHP to interface with it.

Appendix 3

Considerations

The deviations from the requirements are outlined per change of the requirements. The numbers match those of the requirements in Appendix 1.

1.1.1: Also able to be stored is the ratio between two elements. Considering that a ratio tells more about the status of a volcano then the concentration of a single element. The concentration of a single element can be changed by meteorological conditions as well. Rain decreases the concentration, evaporation increases it. The difference between evaporation and volcanic activity is not seen when looking at a single element. Looking at a ratio between two elements - of which one is exhumed by the volcano, and the other is not - cancels this out.
Of the temperature and the density, instead of the original value and unit, the recalculated value and unit is saved. This is done because of the ease of recalculation in these cases. For concentrations, the original value and unit are saved, and the recalculation in other units is done in the output stage.
1.1.2: The volume of the lake is not expressed in minimum and maximum volume, as there will not be much variation in this. The (average) volume of the lake is usually sufficient.
1.3.2.3: Dynamic definition of the header proved to be more difficult then expected. It is now included in Future Work.
The way it is now, the user first defines a name and the amount of columns present in the header. The user then selects the type of columns in the next step, and the column specifications (units, formatting of the date / time) in the third step. This is hardly any slower then the proposed dynamic definition.
1.3.2.4: Depth of sampling is not a mandatory value. The sampling depth will usually be at the water surface, and most users simply presume this is the case.
1.4.1: For the process of validation of the data, an extra table is used in the system. This is the "Batches" table. Sampling data is entered in batches (multiple rows in one entry, all submitted at the same time). As the headers of different records can vary, putting all unvalidated data in one

big table would make checking very difficult due to many empty cells within this table. However, as all sample records from one batch have the same header, grouping those together in a table proved a viable solution. The header itself is saved alongside the other batch data as well. Administrators then can validate or reject a single record, a complete batch, or all data unvalidated at that point.

2.2: The original plan included a feature in Google Maps, placing markers on the map on locations where the volcanoes entered in the system are located. Due to time issues and a good findability of the volcanoes as it is, this feature is moved to Future Work.

3.1: To minimise the amount of calculations, the concentration values will not be stored in a unified manner, but rather in the unit in which they were saved. They are still retrievable in all mentioned units, but the recalculation will take place in the output stage.

The use of SI units is largely enforced. The only non-SI units that are allowed to be used are Kelvin and Fahrenheit in the temperature scales. This is done because Fahrenheit is a more natural value to many of the users.

The imperial system (miles, ounces, gallons, etc) is not supported as of this point. The recalculation within SI units is already quite complex, adding the complete imperial system only makes this function more complicated. It is presumed that the scientific community is able to work in SI units.

Non functional requirements

2.5: The Administration panel did not turn out the way it was intended. While it's intention was to have a central portal for the Administrator in which they can perform their tasks, it is basically a page that tells a few statistics, and that allows the Administrator to go and validate data. Other functions are either reachable from the menu, or spread out across the system. Making the Administration Panel a more useful page is on the to-do list, but the functionality of the system is not impaired by the state it is in now.

3.3: Instead of breadcrumbs, a menu is used. This allows for easy navigation to all necessary functions from every page on.

Other important decisions

The original plan includes saving the date and time in a human-readable format in the database (DD-MM-YYYY hh:nn). However, it turned out to be quite difficult in use, and sorting data took too much time - especially with larger sets of data. A notation in UNIX timestamps was briefly considered, but as the UNIX timestamp does not allow for any dates before 1970, a different method would have to be found. The solution as of now is an approximation of the minutes before or after the year 0. The approximation lies in it that for simplicity, the time value of one month is calculated as the time value of 31 days. Months with less then 31 days, and leap years, are not taken into account. As it is, the range is large enough, ranging from about 3000 BC to 3000 AD.

Appendix 4

File and function overview

adddata.php	The first page in the process of adding data. On this page, the user selects a header to work with, or the user can define a new header, or edit / remove an existing one.
adderupt.php	This file is used by the Administrator to add an eruption record to a volcano.
adminpanel.php	Contains a few general statistics about the data stored, and is (meant to be) a portal to the administrator's functions.
checkdata.php	The last step in adding data. A user can check his data here, and reject it if it's not good. An administrator can reject or validate all unvalidated data here.
countries.php	A file containing a list of all countries.
datacsv.php	Using this file, a user can output data from the database in comma separated format.
datagraph.php	Using this file, a user can output data from the database in graph format.
datatable.php	Using this file, a user can output data from the database in tabular format.
defineheader.php	Verified Users and above can define and edit their data headers here.
editeruption.php	The Administrator can edit an eruption record in this file.
editvolcano.php	The Administrator can edit a volcano record in this file.
excanvas.min.js	A javascript library file needed for display of the graph in datagraph.php
finduser.php	Using this file, the user list can be seen and a user profile can be selected.
header.php	The header file in which the main functions are located. ● clearsession - Logging out a user ● explcoords - turning a coordinate string into a html output format ● datetoint - converting the numerical values of the date into the integer value saved in the database ● inttodate - converting the integer of a date into separate numerical values. ● inttoctime - converting the integer of a date into a UNIX

	timestamp ● monthtostr - converting a numerical representation of a month into a textual representation of that month. ● options - outputting all options of different types of much used drop-down menus. ● options2 - a variation on options - merging into the first options function is possible, but for future work ● headerval - returning the user readable format of a header value from the simpler format in which they are saved in the database. ● translateheader - converting the saved header into a readable format using the headerval function iteratively
index.php	The main page of the system.
inputdata.php	The file where the data is actually input into the system. The header has been selected at this point.
jquery.flot.js	JQuery library used in displaying the graph in datagraph.php.
jquery.js	JQuery library
login.php	The page from which users can log in into the system.
logo.php	The file where the logo is displayed.
logo1.png logo2.png logomain.png	The image files from which the logo is made up.
menu.php	The menu, included in all other pages. If the menu has to be edited, only this file will have to be adapted.
newvolcano.php	The page from which the administrator can add a new volcano record.
profile.php	The page from which a user record (a profile) can be edited.
recalculate.php	File containing the recalculate function. This function recalculates a concentration value from one unit to a different one.
recaptchalib.php	Library for the CAPTCHA used in register.php
register.php	The page from which a new user can register an account.
style.css	The stylesheet of the environment.
texture1.jpg	The background texture used.
viewprofile.php	The profile can be viewed from this page. Editing the profile is linked through from here.

volcano.php	The volcano page, displaying all information about the volcano and the known eruptions, and giving the possibility to view the sample data, add the data and edit the volcano and it’s eruptions. Available functions are based on access level.