Specifications

A Data Harmonization Portal

Athens University of Economics and Bussiness
Charoula Velissarakou
11/4/2017

1. Scope

1.1 Overview

The Data Harmonization Portal is a tool designed for laboratory use, so that its registered users will be able to manage medical variables in a way that reliability is guaranteed.

This document will cover the registration and log in process and the detailed ways that users follow to upload their data, edit it, and download it. It will also describe the second purpose of the tool, the data harmonization, which will be achieved with the help of all the crowd sourcing functionalities that the tool will use to combine all these data that come from all the different registered users.

1.2 Objectives

The objectives of that web platform application are as following (this specification covers just a small part of that main application because it is not complete yet) :

- To support user login and registration to application
- To effectively upload meta-data and download it.
- To share meta-data with other members of the community.
- To easily manage and combine data from heterogeneous sources into an integrated information product.
- To enhance the quality of the data and reduce errors.
- To eliminate redundancies and duplications of data.

2. Flowchart

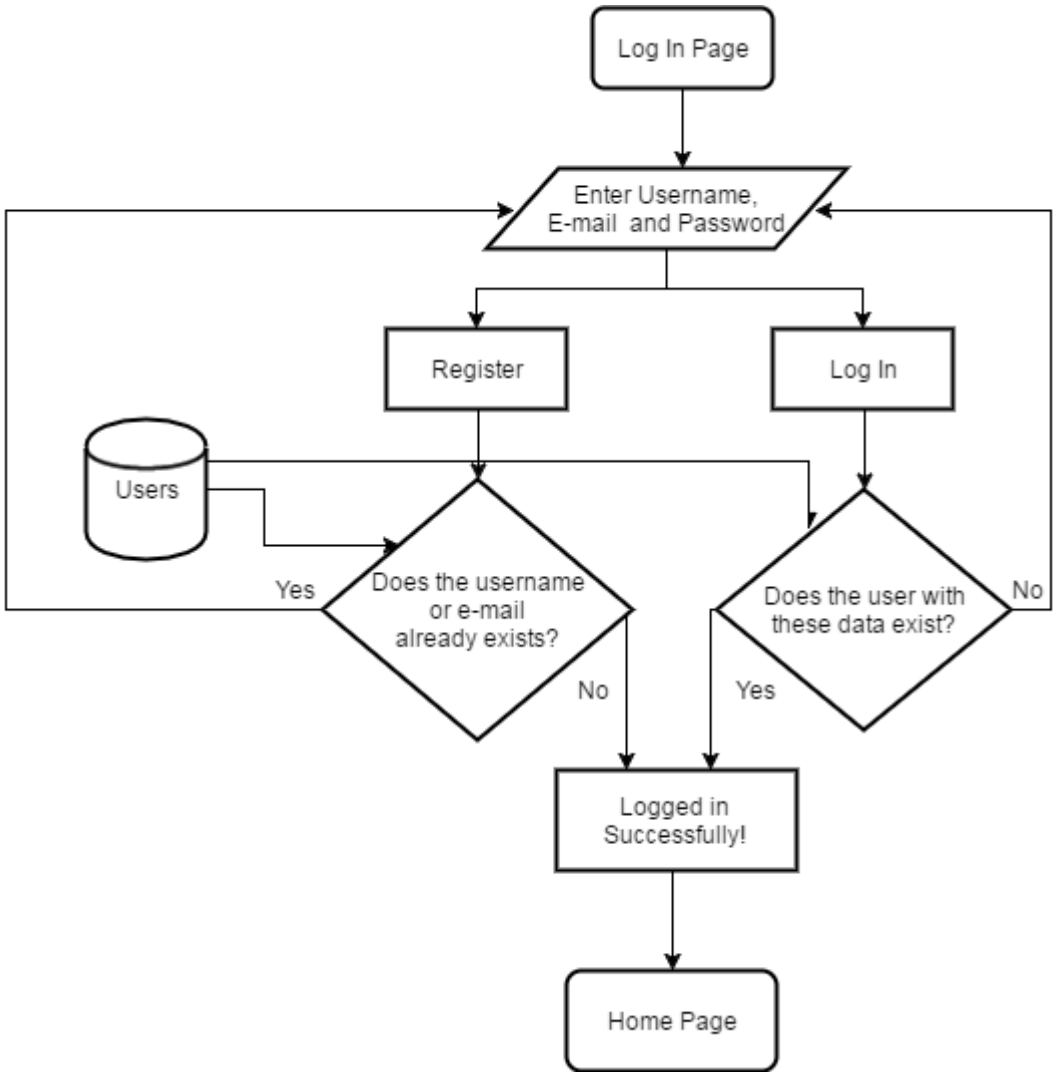
2.1 Log in Flowchart

When the user opens the Data Harmonization Portal for the first time, the log in Page will be shown.

The users should be registered so as to use the tool and their data should exist in the "users" database which is created for that purpose.
After typing a valid username, e-mail and a password there will be two use cases.

- Register to Application:
  If the imported username, mail and password do not exist in the database, the user will be able to log in automatically with these data, once they are stored into the database. This will be done, when the user presses the "Register" button. The server has a folder for each user referred to as their username. A Public and a Private folder will be created in which they will save their data.

- Log In to Application:
  If the imported e-mail and password exist in the database the user will be logged in, otherwise they will have to correct the username, e-mail or the password again.
  The Log in flowchart is the following:



2.2 Home Page Flowchart

When the user logs in, the Home Page will be shown. Then the user can manage the data as they wish.
On the left, there is a navigation menu that provides the following functionalities:

1. **Upload meta-data:**
   The user will have to choose a file, that has a .json extension once this button is clicked. This JSON file contains all the data of grouped medical variables, that are universal and commonly accepted by the community that uses this tool. A file sample should be like this:

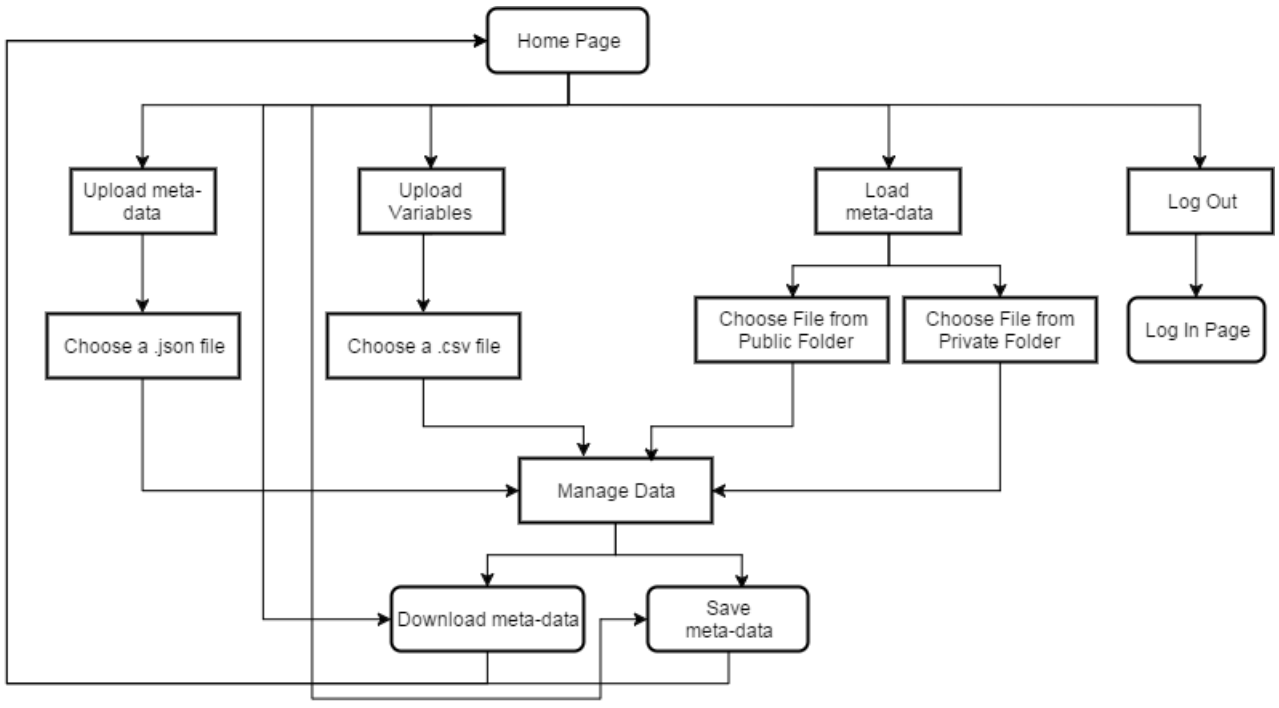

```
[
 {
  "code": "Group 1",
  "label": "Group 1",
  "parent": "None",
  "description": "Group 1",
  "children": [
   {
    "code": "Group 3",
    "label": "Group 3",
    "parent": "Group 1",
    "description": "Group 3",
    "children": [
    ]
   }
  ]
 },
 {
  "code": "Group 2",
  "label": "Group 2",
  "parent": "None",
  "description": "Group 2",
  "children": [
   {
    "code": "var 6",
    "label": "var 6",
    "type": "Binominal",
    "group": "Group 2",
    "description": "var 6",
    "methodology": "var 6"
   }
  ]
 }
]
```


   Each group has a unique code, a label, a parent, a description and an array with its children. The array can include variables and also other groups with their own attributes. In this way, a hierarchy is formed. With the use of the

D3 javascript library this hierarchy is plotted on the first panel on the left so that the user can see this file's visualized result.

2. **Download meta-data:**
   When the user is finished, they can download the new meta-data.
3. **Upload Variables:**
   The user will have to choose a file, that has a .csv extension once this button is clicked. This CSV file contains all the data of ungrouped medical variables. Each variable has a unique code, a label, type, description and methodology. When the file is uploaded the user can manage these variables which will be shown at the variable table on the first panel on the right, and add them in a group that he/she will create or upload.
4. **Load meta-data:**
   The user will have to choose a file to load from either the public or the private folder that exist in server. if it is a json file, the hierarchy that has been created so far will be replaced with the new one and the new variables will be added to the old ones. Then the user can choose which variables they want to keep or delete or even edit.
5. **Save meta-data:**
   The user can save their data to the server. If they choose the Public folder, their friends will be able to see and use their data for later use. If they choose the Private Folder the files will be accessible only by them.

The Home Page flowchart is the one below.



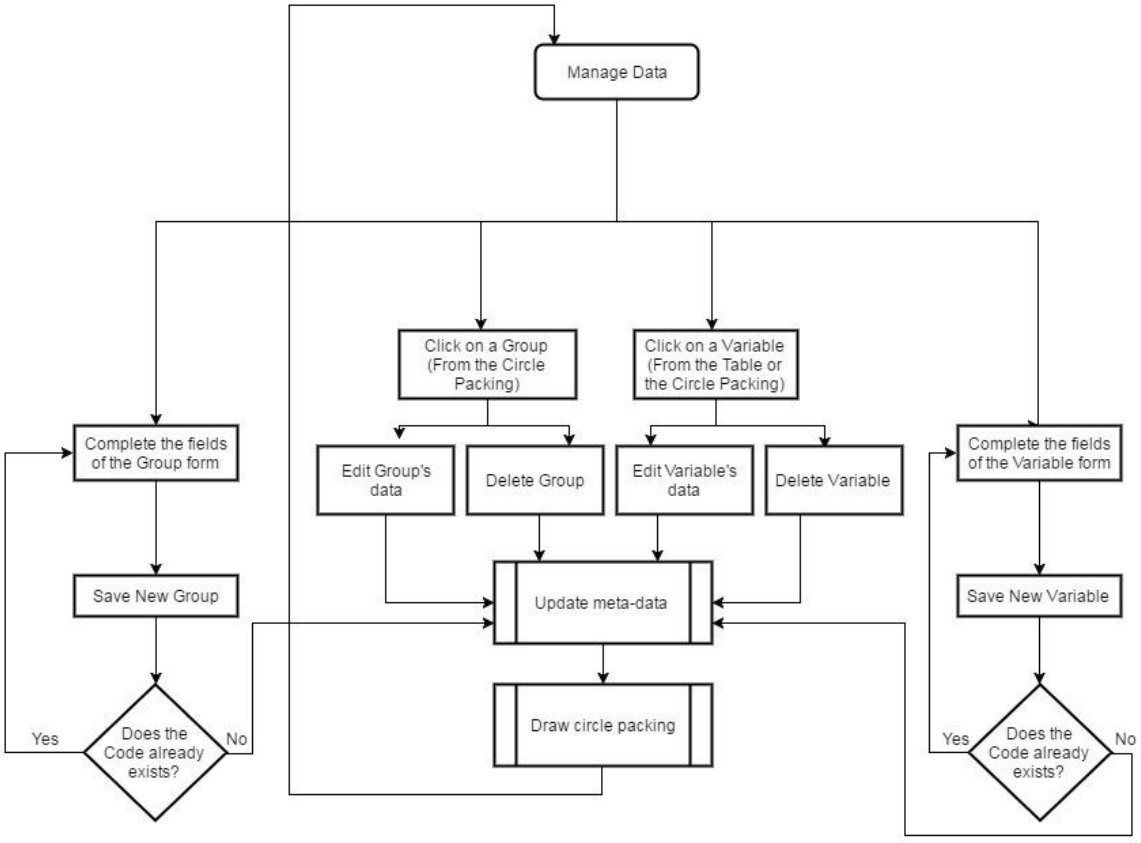
The "Manage Data" action is analyzed by a flowchart too.

2.3 Manage Data Flowchart

There are four ways to manage data.

- Add a new Group:
  Using the "Selected Group" form, the user will be able to fill the data of the new group, he/she needs to add. Code and Label are required fields. If the code is already taken by another group that is saved to the meta-data (JSON), an alert message will appear and the user will have to change it so that it can be added to the pack. Then the new group will be saved to the meta-data and will be plotted as a circle on the first panel on the left. Then, a click event will be added to that circle and whenever a user clicks that group, its details will be showed up at the "Selected Group" form.
- Click on a group circle:
  When the appropriate form will be filled, the user can edit or delete the selected group. If that happens, the JSON meta-data will be updated properly and the circle packing will be drawn again.
- Add a new Variable:
  Using the "Selected Variable" form, the user will be able to fill the data of the new variable, he/she needs to add. Code and Label are required fields. If the code is already taken by another variable that exist at the table, an alert message will appear and the user will have to change it so that it can be added to the table. Then the new variable will be saved to the meta-data, its details will be added to the variable table and it will be plotted as a circle on the first panel on the left. Then, a click event will be added to that circle and whenever a user clicks that variable, its details will be showed up at the "Selected Variable" form and the row that has its details, will get highlighted.
- Click on a variable circle:
  When the "Selected Variable" form will be filled, the appropriate row will be highlighted and the user will be able to edit or delete the selected variable. If that happens, the JSON meta-data will be updated properly and the circle packing will be drawn again because the hierarchy might change (If the parent is different after editing or if the variable is deleted from a group). If the user choose to edit the variable, the highlighted row will be deleted from the table and a new one with the new data will be added. If the user choose to delete the row then it will be permanently deleted from the table.

The "Manage Data" flowchart is the following:



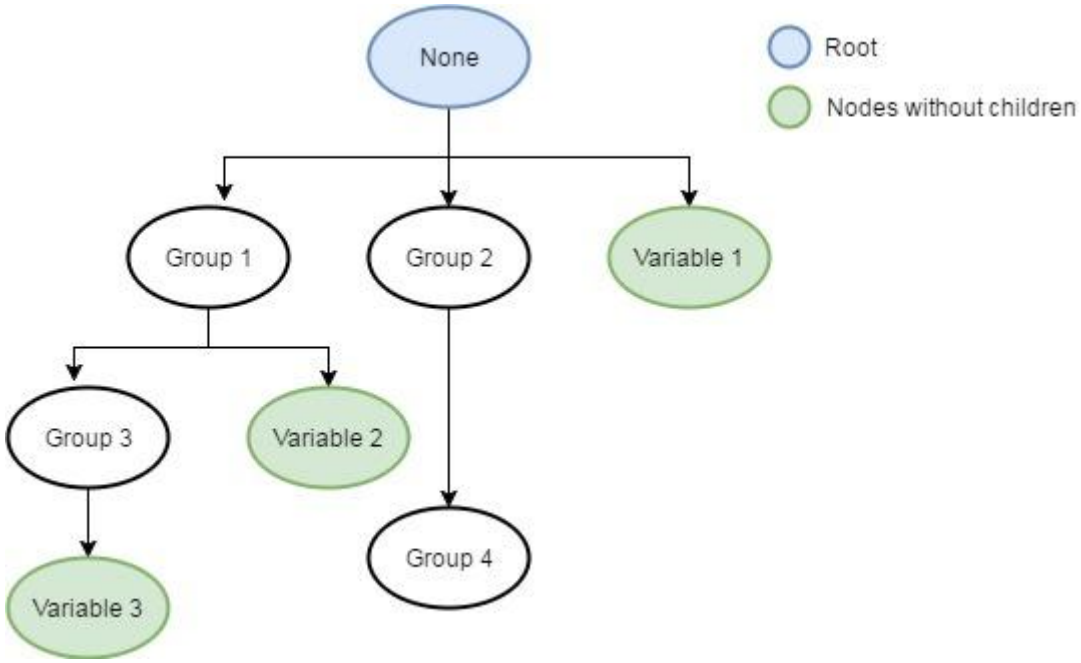
3. Update functionalities

The Data Harmonization Portal is basically based on the meta-data creation that happens during a user's session. These meta-data need to get updated after every change made on them. Consequently, the hierarchy is constantly changing, so the plotting should also change after the meta-data update.
Based on the "Manage Data" flowchart the functionalities that happen each time the data need to update are:

- Update meta-data
- Draw Circle Packing

3.1 Update meta-data

The data are saved in JSON format in the form of a hierarchy as shown before. A visualized hierarchy example is:



"None" is the root of the hierarchy, it is the supergroup that contains every node that is created. Group nodes can have children, unlike variable nodes.

Whenever an update (edit, delete or add) takes place, there will be a need to find the parent of the specified node and then the child that the action will be done to. As far as the edit action is concerned, the delete action will take place first and then the add action should follow. This happens because the edited node might have a different parent, so it will have to be deleted from the old one first, so that it can be added to the new one later.

The updated nested JSON object will be used for the circle packing update.

Note: So as to never lose the hierarchy or the variables that have been created so far while browsing through the tabs of the tool, two variables are saved to the localStorage of the browser. One variable is used for storing nested JSON object and the other one for storing the variables. Whenever a user logs out those variables will be cleared.

3.2 Update circle packing

This is done with the help of the D3 javascript library, using the JSON hierarchy to pack the groups together. Zoom events on circle click are initialized and the labels are added in the middle of each circle.

4. Combining Data - Data Harmonization

This topic is not completed yet.

5. Glossary

Meta-data	Data that provides information about other data.
JSON	**J**ava**S**cript **O**bject **N**otation JSON is a syntax for storing and exchanging data.
CSV	**C**omma **S**eparated **V**alues CSV stores data in a table structure format.