Summary of Algorithmic Performances

Alexander Fomin
University of Vienna

Hermann Hinterhauser
University of Vienna

1 Motivation

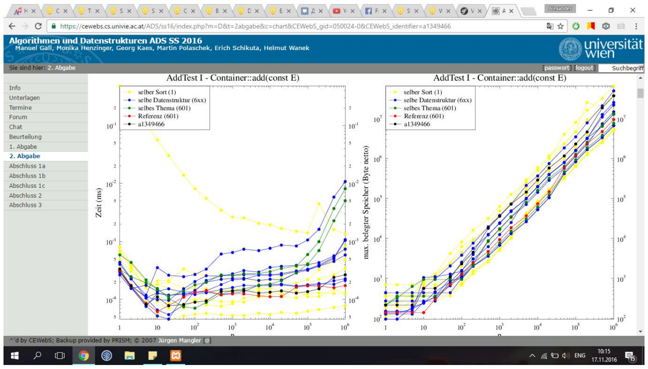
In the lecture ”Algorithm and Data Structures” students have to implement a data structure with a sorting algorithm. Each student has to select one of a pre-selected set of data structures and a sorting algorithm. After the implementation a few test-runs are made and the results are saved. By the end of the course the students can compare their results with all students who used the same data structure and/or sorting algorithm, together with a reference. To compare the results a lot of charts are drawn and one may loose the overview very quickly.

Within this project new possible solutions will be discussed and one of them will be implemented, using interactive charts. Possible interactive features are a zoom function, the possibility to select a student and load his data in the charts, maybe it could also be possible to load data from multiple students and remove lines from the chart or change its scaling.

Possible users for such an application are the students and professors. Maybe such a tool could also be useful for professional programmers who want to see the efficiency of their implemented data structures. The provided data was from the year 2012 and contained the data for 129 students and 21 reference data structures. The results were stored in xml files.

2 Related work

There is an already existing tool on the CeWebs site for ”Algorithm and Data Structures”. An example-image is shown in Figure 1.



https://cewebs.cs.univie.ac.at/ADS/ss16/index.php?m=D&t=2abgabe&c=chart&CEWebS_gid=050024-0&CEWebS_identifier=a1349466

Algorithmen und Datenstrukturen ADS SS 2016
Manuel Gall, Monika Henzinger, Georg Kaes, Martin Polaschek, Erich Schikuta, Helmut Wanek

Sie sind hier: 2. Abgabe

^'d by CEWebS; Backup provided by PRISM; © 2007 Jürgen Mangler

Figure 1: The tool on CeWebs

3 Approach

The Application was designed and implemented in 3 main milestones:

- M1: Project Proposal
- M2: Lo-Fi Prototype
- M3: Hi-Fi Prototype

After the third milestone this report was written and the Implementation was revised yet again to implement its final form.

3.1 Project Proposal

This milestone was the first proposal of the project and included a specification of the project, a short description and the team members planned, which tools could be useful and how the goals may be achieved.

3.2 Lo-Fi Prototype

A first Lo-Fi Prototype was implemented using Tableau. This Lo-Fi prototype consisted of 10 sheets and 3 different dashboards Furthermore, the proposal was revised and different scenarios and smaller milestones were planned out.

The first Dashboard (Figure 2) had 3 simple line-graphs, visualising the calc-time, used memory and the memory efficiency.

A line can be used to show trends very effectively, also the user can see which measuring points belong together. This will become very important, when more than one line is displayed. It is a good candidate for the implementation.

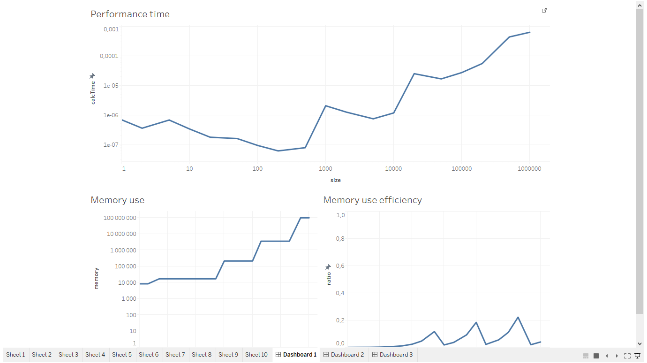


Figure 2: The first Dashboard

The second dashboard (Figure 3) showed a different line-graph to visualise the calc-time and used two bar-charts for memory and memory efficiency. The area below the line was filled blue.

The blue area below the line does not add much to give a visually appealing view than a simple line. Instead it has the potential to overlay other lines, when implementing it. It was an idea, we tried it and rejected it for further designs. The bar charts have a similar problem. They contain a lot of ink, but offer no further visual improvement. It is possible to observe a trend in the data, but what if further data is added? If new data is added, new bars have to be added to the chart, beside the already present bars, making the chart broader. This would require users to scroll

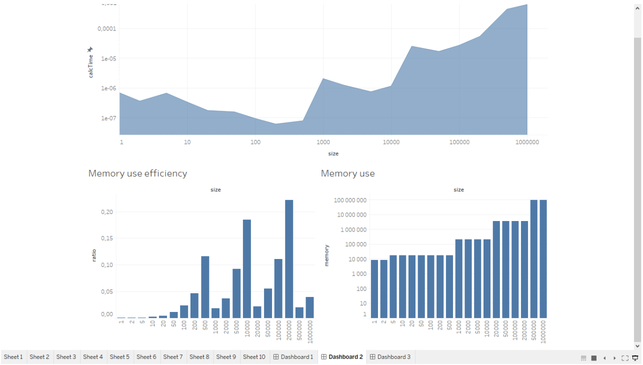


Figure 3: The second Dashboard

sideways very often, to observe a wider range along the x-axis. This would not be appealing and will be rejected for future designs.

On the third dashboard circles were used to visualize the data. It is shown in Figure 4.

The circles highlight the measuring points very good, but it makes it difficult to find the connections between these points. Especially if you look at the memory efficiency it is very hard to get the same expression as the graphs before. It looks very messy. This problem will get even heavier, when further data is added. It would just look like a huge cloud of data and nobody would think that each datapoint belongs to a serie of measurements. Using different colors for the circles may help, but trends would be more obvious by using line charts (which can also use different colors).

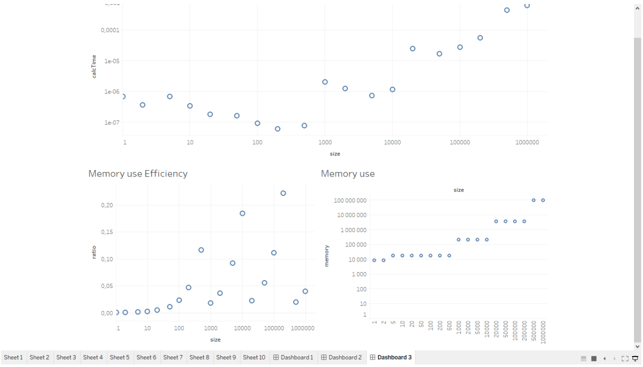


Figure 4: The third Dashboard

Finally, a solution was chosen, that incorporated both the design of the first dashboard and the third dashboard. Such a solution has already been used on the CeWebs-Website. The application draws charts with filled circles and connects these circles with straight lines. This allowed for a very good observation of the trends, while measuring points remain clearly visible. It combines the strengths of both approaches.

3.3 Hi-Fi Prototype

The third milestone included a short reminder, a few Use-Cases and challenges/problems were discussed. Also a final visualization plan was created. It was created using creately.com. We decided to implement a primary function (Figure 5) and a secondary function as shown in Figure 6.

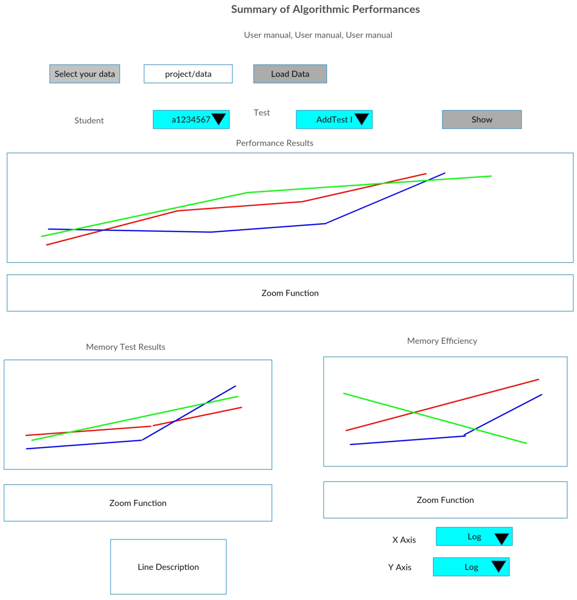


Figure 5: Visualization Plan, Primary Function

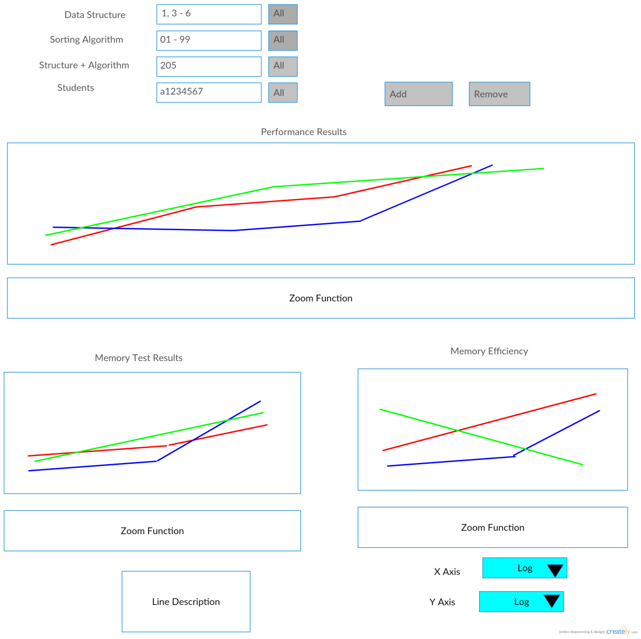


Figure 6: Visualization Plan, Secondary Function

4 Implementation

4.1 Ressources

The application was implemented by using mainly HTML and JavaScript. Also, on some occasions, PHP was used. For drawing the charts, the JavaScript Library D3 was used, which provided many useful functions for drawing charts. Furthermore JSON was used, especially in later versions of the implementation.

4.2 Challenges/Problems

The first big challenge was to get the data from an xml file, none of the project team members knew prior to the start of this project, how this should be done. After looking for a solution on Google.com, one was found on http://www.w3schools.com/xml/ajax_applications.asp and a similar one was implemented first. Later Alexander Fomin expanded the first solution with JSON.

A major challenge was getting the content of the folder containing the data. JavaScript works from the Clients side and does not allow to access the file system. A solution was found using PHP which works on the Server side and therefore can access the file system.

The code also uses JavaScript with parallel execution, which causes a problem with the colors of the lines. While it can be resolve by manually removing the falsely colored line and adding it again, no further solution for this problem could be find due to time constraints.

4.3 Description of the tool

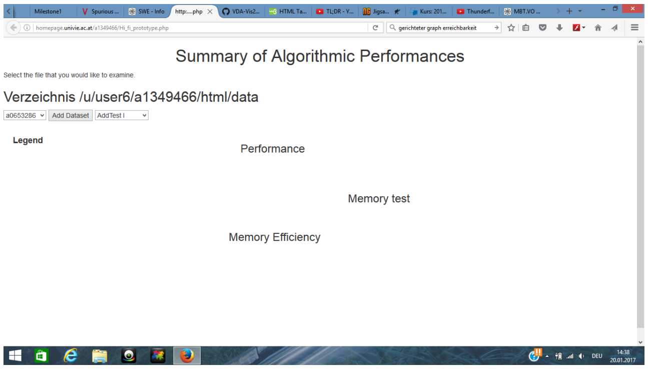
With a drop down menu, a student can be selected, according to his Matrikelnummer. The same can be done with the tests. After selecting a student and a test, the user presses the button ”Add Dataset” and the charts are drawn. This was coded using HTML. The content of the drop down menu for the students was filled by using PHP, the drop down menu for the tests was hard-coded in HTML. The charts are three Line-charts, the x-axis show the size of the tested dataset and the y-axis show: one for the runtime, the second for the used memory and another one for memory efficiency. Line-charts were choosen, because they allow to very easily observe trends. The data for the lines was extracted using JSON and the lines were drawn with D3.

Also a zoom-function was implemented using D3. Every chart has a smaller chart below it, with a grey rectangular. The grey rectangular can be changed in size and position and by doing that, the view of the responsive chart is also changed in size and position. This allows for very quick zooming, while the user simultaneously keeps an overview on which section of the chart the user is currently observing.

5 Results

5.1 Scenarios

In the first scenario, the student with Matrikelnummer ”a0012251” wants to take a look at his test results. After starting the application he immediately gets the view shown in Figure 7.



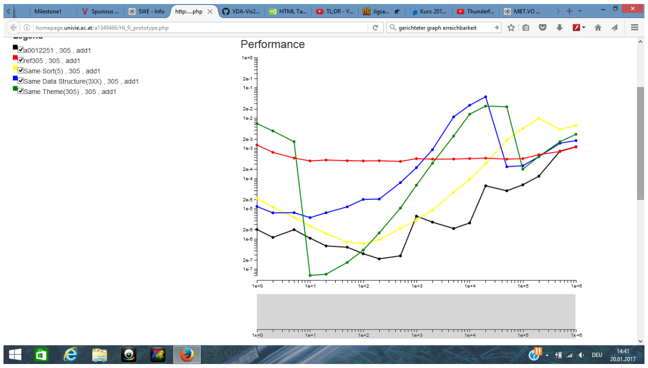
Summary of Algorithmic Performances

Select the file that you would like to examine.

Verzeichnis /u/user6/a1349466/html/data

Figure 7: The tool, ready for use

Now he uses the two drop down menus to first select his Matrikelnummer and the test. The selected test was ”AddTest I”. The application loaded the data and drew different graphs. One table was named ”Performance” (Figure 8), it showed the size of the added data on the x-axis and the run time on the y-axis. Below this chart two other charts were drawn (Figure 9). Their x-axis represented the size too, the y-axis on the right chart showed the used memory and the y-axis on the left chart showed the memory-efficiency.



homepage.univie.ac.at/a1349466/Hi_fi_prototype.php
gerichteter graph erreichbarkeit
■ ☑ a0012251 , 305 , add1
■ ☑ ref305 , 305 , add1
■ ☑ Same Sort(5) , 305 , add1
■ ☑ Same Data Structure(3XX) , 305 , add1
■ ☑ Same Theme(305) , 305 , add1
Performance
1e+0
2e-1
1e-1
2e-2
1e-2
2e-3
1e-3
2e-4
1e-4
2e-5
1e-5
2e-6
1e-6
2e-7
1e-7
1e+0
1e+1
1e+2
1e+3
1e+4
1e+5
1e+6
14:41
20.01.2017

Figure 8: The Performance chart

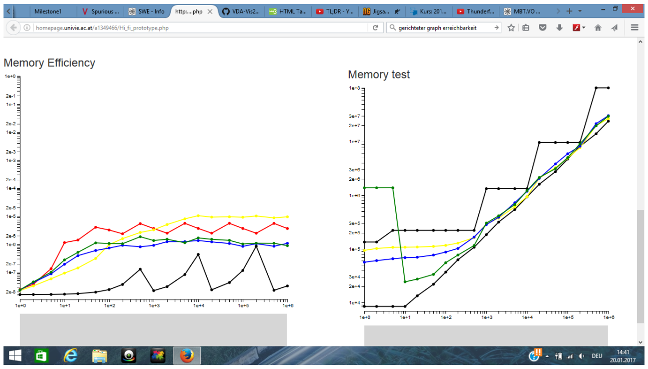


Figure 9: Charts for memory and memory-efficiency

After viewing all the data, the student used the zoom-function to take a closer look on his results. He first clicks on the left side of the grey rectangular below the ”Memory Test” chart and moved it to the right side. The area became smaller and the view of the chart also changed. A similar operation was done with the ”Performance” chart. The student clicked on the left side of the grey area, changed it in size, die the same with the right side and then the student clicked on the area surface and dragged it a little bit to the right. Refer to Figures 10 and 11.

In the second scenario, the student, from scenario 1, wants to compare his results with the results of a close friend. After loading all the data into the application as described in the previous scenario, he changed the Matrikelnummer to ”a0257748” and clicked on ”Add Dataset”. Since he was not interested in the average results from the whole class, he simply removed the lines by un-checking the checkboxes.

The ”Performance” chart showed, that his colleague’s data structure was faster, although his reference was slower (Figure 12). Only at some occasions their results seemed equal. He wanted to examine these data points more closely and zoomed in. There was only a very slight difference, as seen in Figure 13.

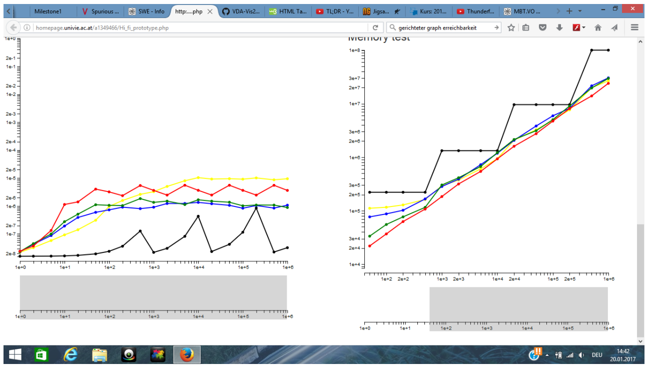


Figure 10: The zoom-function on Memory Test

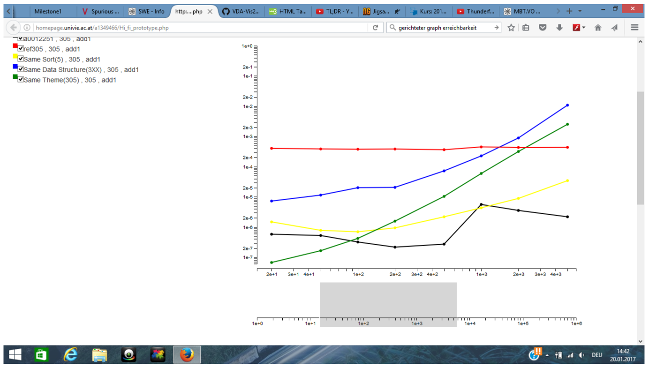
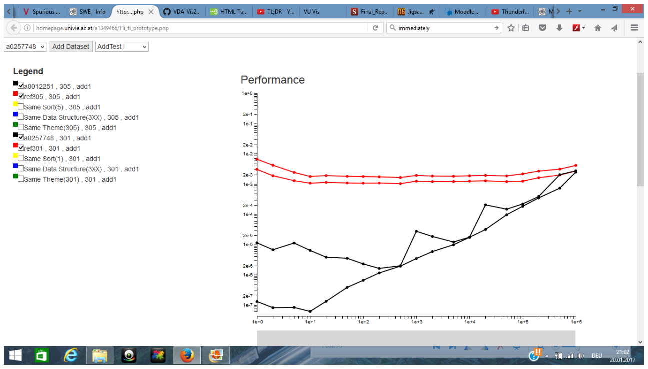

☑ a0012251 , 305 , add1
🟥 ☑ ref305 , 305 , add1
🟨 ☑ Same Sort(5) , 305 , add1
🟦 ☑ Same Data Structure(3XX) , 305 , add1
🟩 ☑ Same Theme(305) , 305 , add1

Figure 11: The zoom-function on Performance


a0257748 ⌄ | Add Dataset | AddTest I ⌄

Legend
⬛ ☑ a0012251 , 305 , add1
🟥 ☑ ref305 , 305 , add1
🟨 ☐ Same Sort(5) , 305 , add1
🟦 ☐ Same Data Structure(3XX) , 305 , add1
🟩 ☐ Same Theme(305) , 305 , add1
⬛ ☑ a0257748 , 301 , add1
🟥 ☑ ref301 , 301 , add1
🟨 ☐ Same Sort(1) , 301 , add1
🟦 ☐ Same Data Structure(3XX) , 301 , add1
🟩 ☐ Same Theme(301) , 301 , add1

Figure 12: Comparing the results of two students

5.2 Performance

The application is a bit slow, but reasonably fast if you consider the amount of data, which is used.

5.3 Feedback

The main functions work fine, it draws the lines as expected. Sometimes the application has a problem with using the right colors. This problem can usually be solved by removing the line and adding it again. Why this occurs is not clear. The interactive features (removing lines and zooming) work good, but the zoom-function can only zoom-in along the x-axis. Sometimes the application fails in drawing charts (Figure 14), then it has to be restarted.

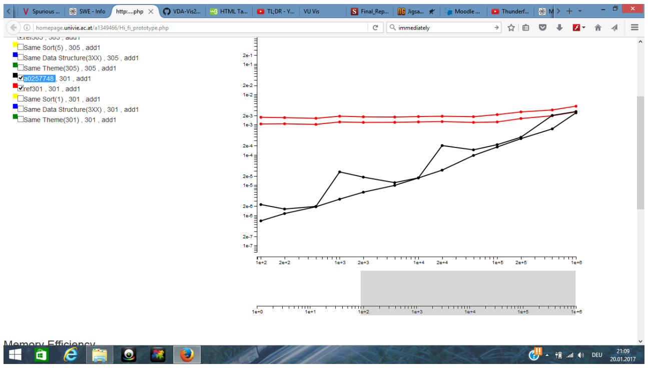

🟨 ☐ Same Sort(5) , 305 , add1
🟦 ☐ Same Data Structure(3XX) , 305 , add1
🟩 ☐ Same Theme(305) , 305 , add1
⬛ ☑ a0257748 , 301 , add1
🟥 ☑ ref301 , 301 , add1
🟨 ☐ Same Sort(1) , 301 , add1
🟦 ☐ Same Data Structure(3XX) , 301 , add1
🟩 ☐ Same Theme(301) , 301 , add1

Figure 13: Comparing the results and zooming in

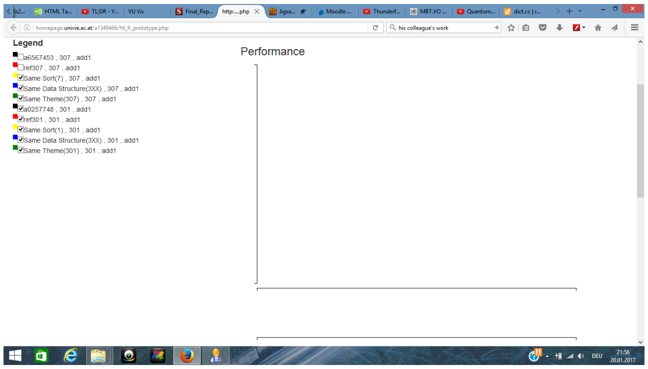

Legend
⬛ ☐ a6567453 , 307 , add1
🟥 ☐ ref307 , 307 , add1
🟨 ☑ Same Sort(7) , 307 , add1
🟦 ☑ Same Data Structure(3XX) , 307 , add1
🟩 ☑ Same Theme(307) , 307 , add1
⬛ ☑ a0257748 , 301 , add1
🟥 ☑ ref301 , 301 , add1
🟨 ☑ Same Sort(1) , 301 , add1
🟦 ☑ Same Data Structure(3XX) , 301 , add1
🟩 ☑ Same Theme(301) , 301 , add1

Figure 14: A failed attempt

6 Discussion

6.1 Strength and Weaknesses

A line chart seemed to be the best choice for this application right from the start on. Lines are an excellent choice if you have multiple measurements and want to observe, how one variable changes while you change the other one. Another type of diagram for such a task, would be a scatter plot. This type of chart also allows you to easily plot one variable against another. It is a good choice, when you have a huge amount of data points, as it is the case for this application. However, what a scatter plot can not show very effectively is the connection between measurement-points. One may use different colors for different measurement-series, but you still have to connect the dots with your eyes (imaginary). That is one of the reasons why, for example physicists, use scatter plots and then fit a trend line into the data. Both chart types require a very little amount of ink to visualise the data, because you do not have to draw huge areas, which represent the data. The scatter plot requires the lesser amount of ink and the observer sees, where all the measured data is.

The final solution was a combination of both approaches. The circles allow the user to see the exact values of the data points and the line can show him the trend and connection between the circles. It combines their strengths, without introducing noteworthy weaknesses.

This design may become a problem, if a huge amount of data sets is added to one chart. The chart may easily become flooded with information and lines can overlap each other. Users can counter that with removing lines manually.

The drop down menu for students was a reasonable choice.

A User has to scroll this menu very long to eventually find his Matrikelnummer. The list is also not ordered making it even more difficult to find the right selection. With less than 200 students, it is not that time consuming, but it can possibly become a bad issue. Another solution, which was used in earlier proto-types, was a text field, where you type in a Matrikelnummer. A very good approach could be a combination of both solutions: a text field and, after typing in some numbers, it gives the user suggestions, like in most search engines nowadays.

Another problem is the selection of the colors. If a user only loads the data from one student, it is not an issue. There are different colors for the student's data, the averages of the other class members and the reference. But, as soon as another student's data is loaded into the application it causes troubles, because the colors used for the data of the second student is the same as for the first one. If one wants to, for example, compare the results of different students, every student gets the same color. The introduction of further colors could solve this problem.

Students might not want their results to be visible to other students, so a solution should be found for this problem. Perhaps the easiest way would be to ask for the student's password, before adding his data.

6.2 Lessons Learned

The main lesson of this project was, that planning ahead is very important. A very big fraction of the workload was not the coding, but more the work before you start coding. At the beginning, coding seems to be very difficult, especially if you have very low experience, but with more experience the coding becomes much easier and the hardest part becomes the reasoning about what you want to achieve. What do you do? Why do you do it? Is it the best or are there better ways? These Questions have to be answered, or you should have at least a very strong idea of what you want to achieve, before you should start programming.

At some points in the project, this was not considered and major changes of the original idea had to be done. It was not that big of a problem, since the project was relatively easy, but one can already see that this may become an issue in greater projects.

You also have to estimate, what you can achieve in the little time. Some features could not be implemented like the secondary function of the application and the function to change the scaling of the axis. There was to little time and we may should have started sooner on some occasions to get more time reserves. Also a short manual for the tool was not implemented and the axis of the charts were not labeled.