**GROUP 7 PROJECT REPORT**

Teemu Halmi

Tapio Paloniemi

Oulu University of Applied Sciences

contents

[1 Introduction 3](#_Toc501013037)

[2 THE WORK ENVIRONMENT 4](#_Toc501013038)

[3 Definition 5](#_Toc501013039)

[4 Implementation 7](#_Toc501013040)

[5 Testing 8](#_Toc501013041)

[6 POSSIBILITIES OF FURTHER DEVELOPMENT 9](#_Toc501013042)

[7 conclusion 10](#_Toc501013043)

[APPENDIX User Interface Design MATERIAL 11](#_Toc501013044)

# 1 Introduction

This is report describes a small web development project we did in December 2017 as a part of our studies in Oulu University of Applied Sciences. The idea of the project was to combine all the studies so far into a single work.

Our task was to design and implement a web based mathematical tool which would offer six different kinds of mathematical functionalities for the user. These were number systems conversion tool, a number systems output, combinatorics tool, random values tool and one free choice tool for which we decided to implement a tool that solves the roots of the second degree polynomial.

The program we designed and implemented works on a single web page and it uses vertical tabs to display different functionalities of the program. The user has complete freedom to move between different tools at will.

# 2 THE WORK ENVIRONMENT

Project team members worked in Oulu University of Applied Sciences classrooms and from home also. Tasks were divided so that both members could work individually from whatever location they chose to. Both team members had their own personal laptop computers to do all the necessary work with.

The user interface mockup was made with browser version of the Balsamiq (<https://balsamiq.com>) software. This allowed both members to edit the design without the need to constantly share files and what changes which team member has made to the design.

The actual of coding of components was done using several different tools. The choice what tool to use for the actual coding was not limited in any way and it was each team members own personal preference. The tools used for the HTML, CSS and JavaScript were Visual Studio Code (<https://code.visualstudio.com>), Coda (<https://panic.com/coda/>) and Vim (<http://www.vim.org>).

The written code was shared between the team members mainly through GitHub (<https://github.com>) but also Tower (<https://www.git-tower.com/mac/>)

was used for that purpose. Doing this allowed more flexibility for the team members because it made it possible to work on the project individually.

The testing was done using several different browsers, namely Chromium (<https://www.chromium.org>), Chrome (<https://www.google.com/chrome/>), Firefox (<https://www.mozilla.org/en-US/firefox/>) and Safari (<https://www.apple.com/lae/safari/>).

This report and the project plan were written with Word Online (<https://office.live.com/start/Word.aspx>#) .

# 3 Definition

This is not a stand-alone software. The user must have a browser to use this software, preferably Chromium or Chrome to get the best user experience. Also, the Quadratic Equation tool depends on MathJax (<https://www.mathjax.org>) to render graphics.

When the user enters the webpage or loads the project's *index.html* file, he is presented with a program that has six different mathematical functionalities implemented. These are placed in a vertical tab structure with the tabs on the left side. And by clicking these tabs the user can switch between different mathematical functionalities.

First one of these is the Number Systems tool which is open by default. With this tool the user can make number conversion between decimal, binary, octal and hexadecimal numbers. User is presented here with four different input boxes where he can enter numbers and they are then converted to other number bases. If the user inputs incorrect values he is presented with a warning dialog informing him of the error.

When the user chooses Number Tables tab he is initially presented with a page consisting only of two buttons and an empty table. By pressing the *Generate* button a table is generated which decimal numbers from 0 to 50 and their corresponding values in binary, octal and hexadecimal. If the table is generated, further pressing of the *Generate* button does nothing. The *Clear* button clears the table.

The next is Combinatorics tools. In this tab the user presented with three different sets to choose from. He does that by pressing a button with the appropriate set name in it. After choosing the set two previously disabled radio buttons and a drop-down menu are made active. With these radio buttons the user can choose between combinations and permutations and from the drop-down menu he can choose the number of elements. Default is one. If *Permutations* radio button is chosen two additional radio buttons become available: *Replacement* and *No replacement*. With these the user can choose how the permutations are done on a given set. If combinations option is chosen the *Calculate* button becomes available but if permutations were chosen the user must choose either *Replacement* or *No replacement* before this happens. After pressing the *Calculate* button results are displayed under *Results*.

Under the next tab is the Truth Tables tool. The user is presented with a two-variable truth table and on its left side is a column of buttons where the user can choose a logical operator. These logical operations are: *negation* (for first variable), *and*, *or*, *exclusive or*, *implication*, and *equivalence*. When the user presses any of these buttons the last column is filled with corresponding truth values. *Clear* button under the truth table empties the whole table and *Fill* button fills columns p and q with values of T and F.

Next tool is Random Values. When the user first clicks this tab, he is presented with input boxes for *Min*, *Max*, *Quantity* and *Generate* button. Min box is used to define the minimum value for random numbers and Max for maximum value. *Quantity* determines the amount of random numbers generated and *Generate* button generates those random values according to parameters given. After pressing the *Generate* button random numbers are displayed below, where y-coordinate shows random number value and x-coordinate the ordinal number of generated random value. Black dots mark the generated random numbers and a red line shows the running average. By moving a mouse cursor over black dot, the user can see its random number value and ordinal number.

Final tool is called Quadratic Equation. With this the user can find the roots of any second degree polynomial with rational coefficients. There are three input boxes labeled *'a='*, *'b='* and *'c='* for the coefficients of the polynomial. Below those is the *Solve* the equation button which attempts to find the roots. If the user entered a = 0 or non-number to any of the boxes an error message is displayed below the text *Solution* after pressing *Solve the equation* button. If all the coefficients are integers an exact solution is given and other cases answer is presented in decimal form which may or may not be an exact solution to the equation.

# 4 Implementation

At the first phase an initial project plan was made which included preliminary schedule and the steps needed to finish the project successfully. The responsibilities for designing and implementing was divided between team members so that each one had to design and implement three functionalities.

Once each functionality was at least in basic working order the integration was done.

All the HTML code to display the starting page is included in the file called *index.html*. This is the page user must open if he wishes to use these tools. Tab structure is implemented with CSS and JavaScript code in *default.css* and *simple-tabs-pure-js.js* (<https://www.cssscript.com/simple-javascript-tabs-library-simple-tab/)> which handles the tab switching. File *demo.css* contains the styling for tab structure.

HTML code for each of the tools is also included in the *index.html*. This is code is however changed on-the-fly by the scripts located in the scripts folder.

The necessary scripts for Number Systems tools are located in the file called *numbersystems.js* located in scripts subfolder and styling information in the *numbersystem.css* which is located in *css* subfolder. All of the scripts for the tools are in scripts subfolder and styling sheets in *css* subfolder.

The scripts for tools Number Table, Combinatorics and Quadratic Equation are all in a file called *ownscripts.js* and styling information in *default.css*.

Scripts for tool Truth Tables are in a file *truthtables.js* and styling information in *truthtables.css.*

Random Values tool uses two different script files: *Chart-bundle.js* (<http://www.chartjs.org>) to generate graphics for displaying the distribution of random values and *random.js* for the rest. Styling is in the file *random.css*.

# 5 Testing

Each team member was mainly responsible for the testing of components they designed and implemented. Due to the nature of project each individual tool could be coded and tested separately before moving into integration phase. No official testing procedure was deemed necessary for this project.

Once the integration was done the testing of the user interface, its components, functionalities and compatibility with the browsers was done by both team members. The way the project work was divided caused some minor issues during the integration process, mainly because of conflicting stylesheets. These issues were easily fixed and they didn't cause any significant delay.

# 6 POSSIBILITIES OF FURTHER DEVELOPMENT

Since this is a school project with no obvious application outside this course the further development is not likely to happen.

# 7 conclusion

The project was finished successfully and well before the deadline. No major issues were encountered during the work and only small issues during the integration process, which was to be expected. Those kinds of problems are a normal part of the development process.

In this project, the tools were completely independent of each other which made it possible to divide the design and implementation of the tools between team members and give each member freedom with the design and implementation, only constraints being the limitations set by the user interface.

If the tools would have been dependent on each other in some way this kind of approach would not have worked. Much more detailed initial design would have been necessary and also much more communication during the actual coding.

# APPENDIX User Interface Design MATERIAL

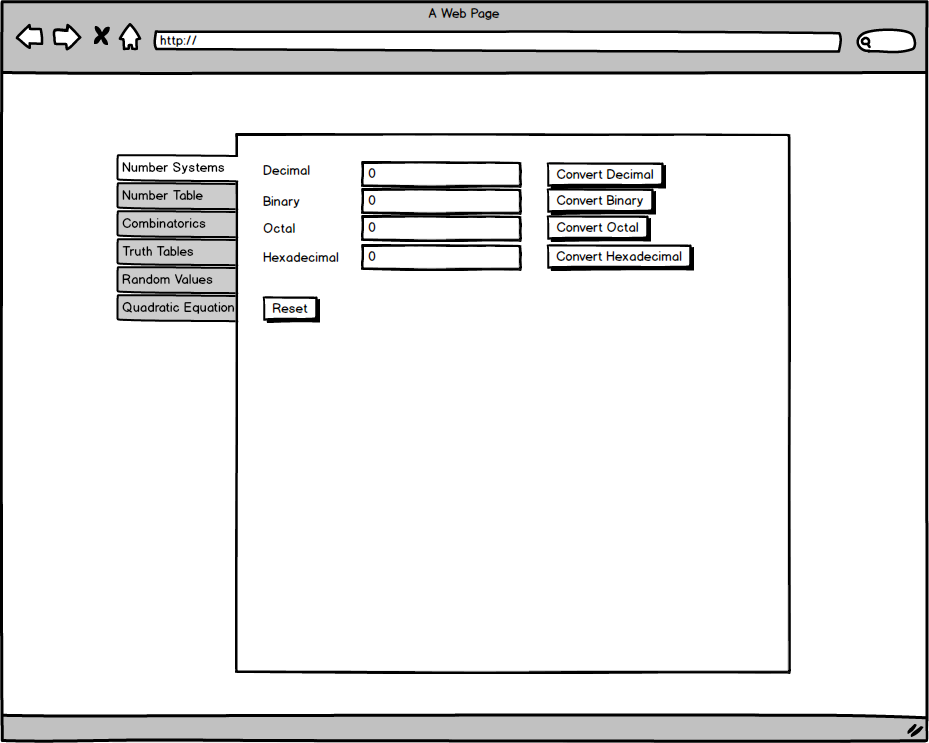


Figure 1 Opening page, Number Systems tab

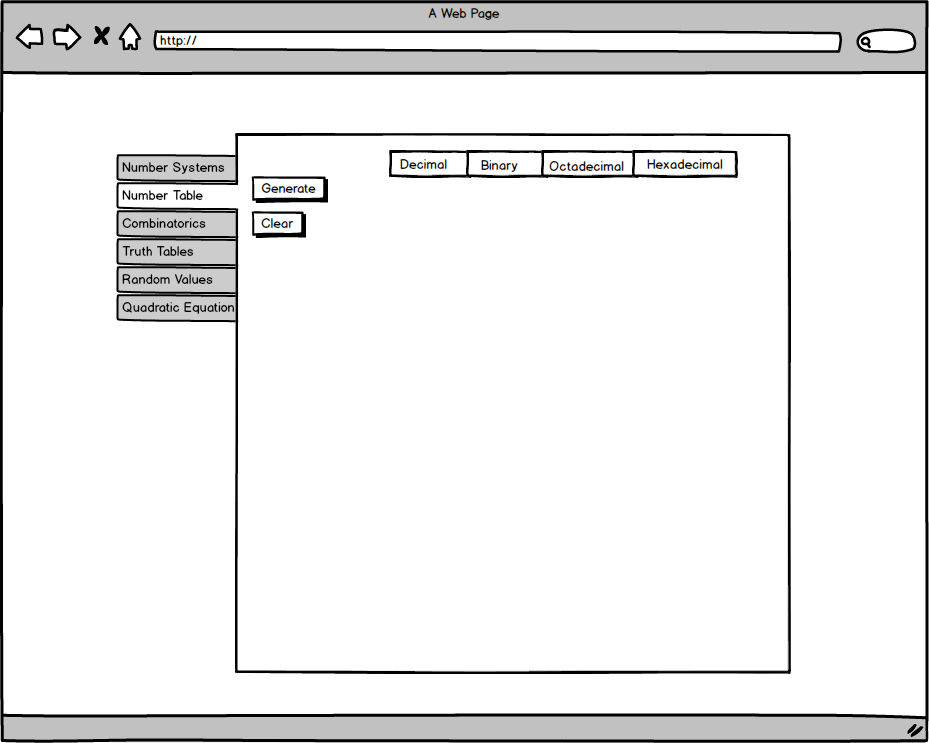


Figure Number Table tab

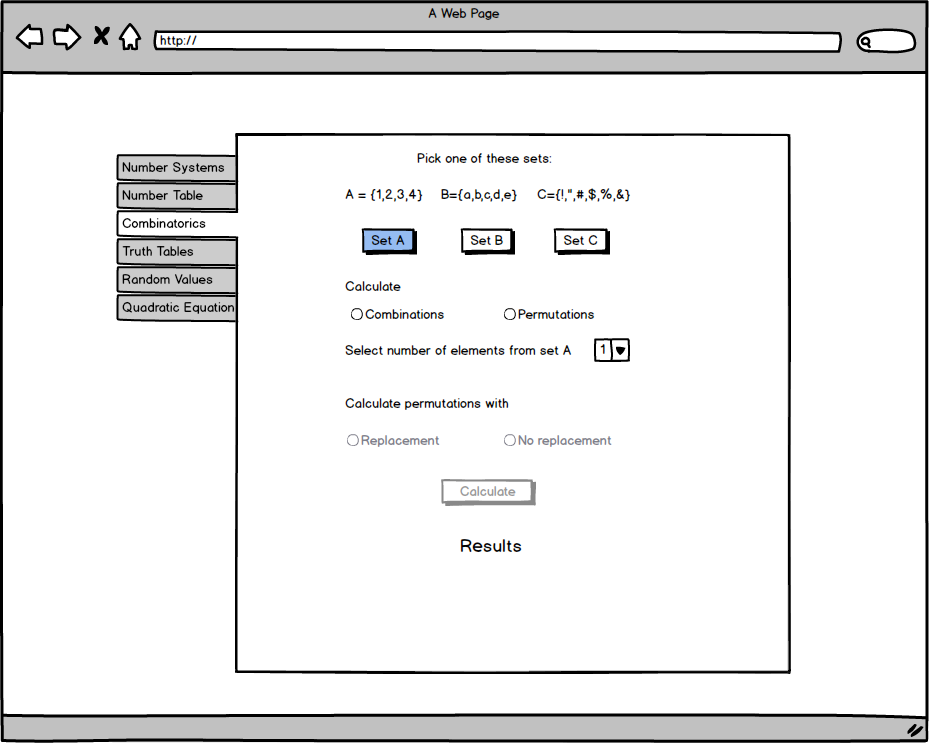


Figure Combinantorics tab

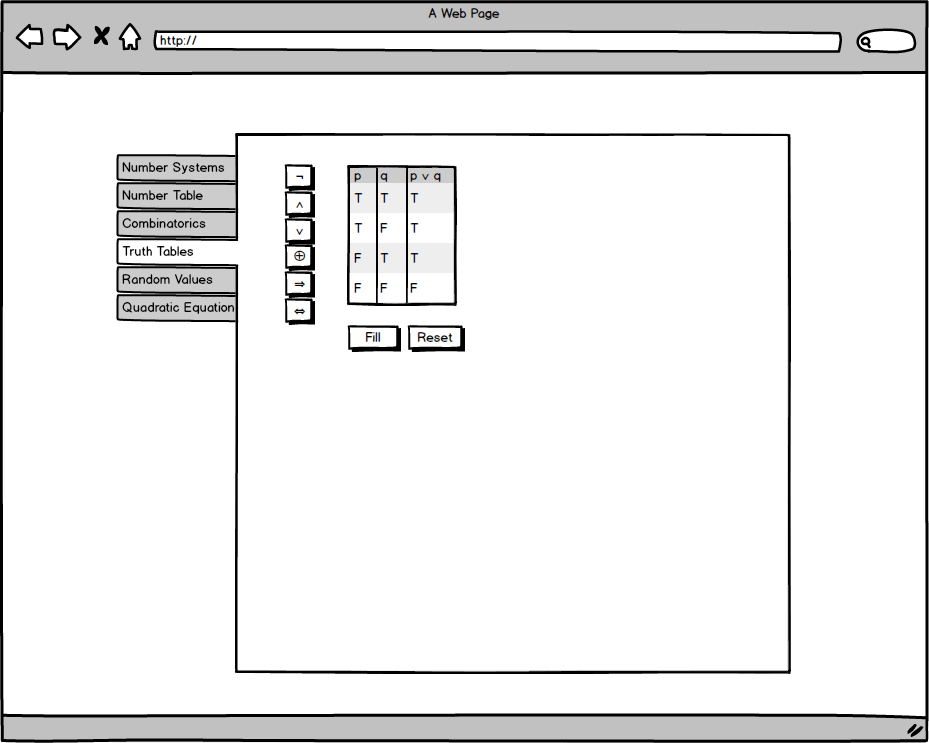


Figure Truth Tables tab

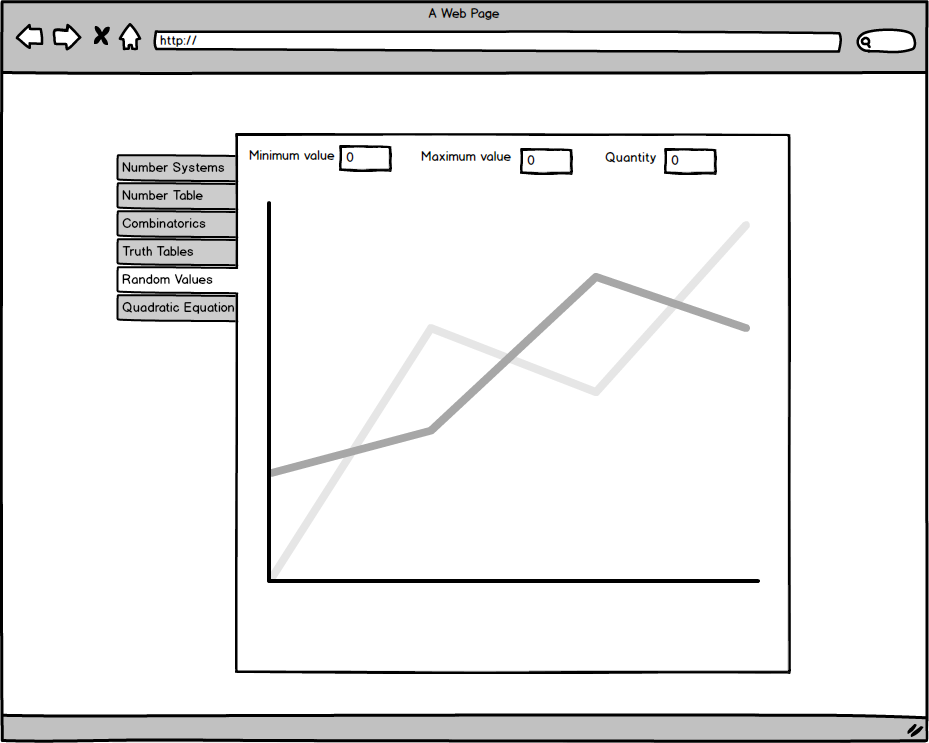


Figure Random Values tab

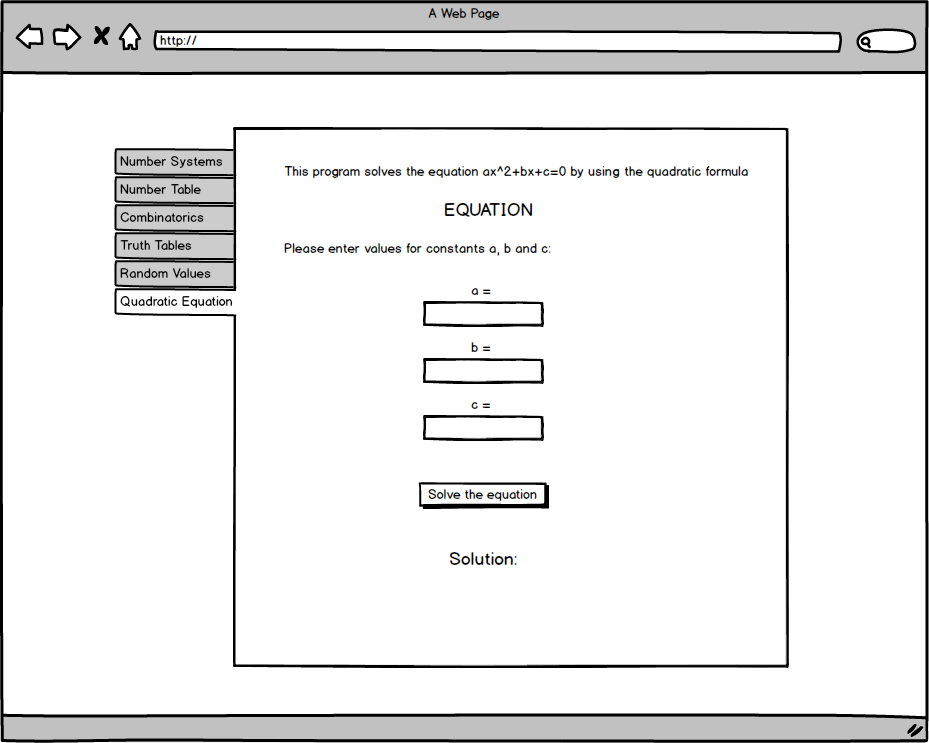


Figure Quadratic Equation tab