

FLOW; TEAM ASSIGNMENT

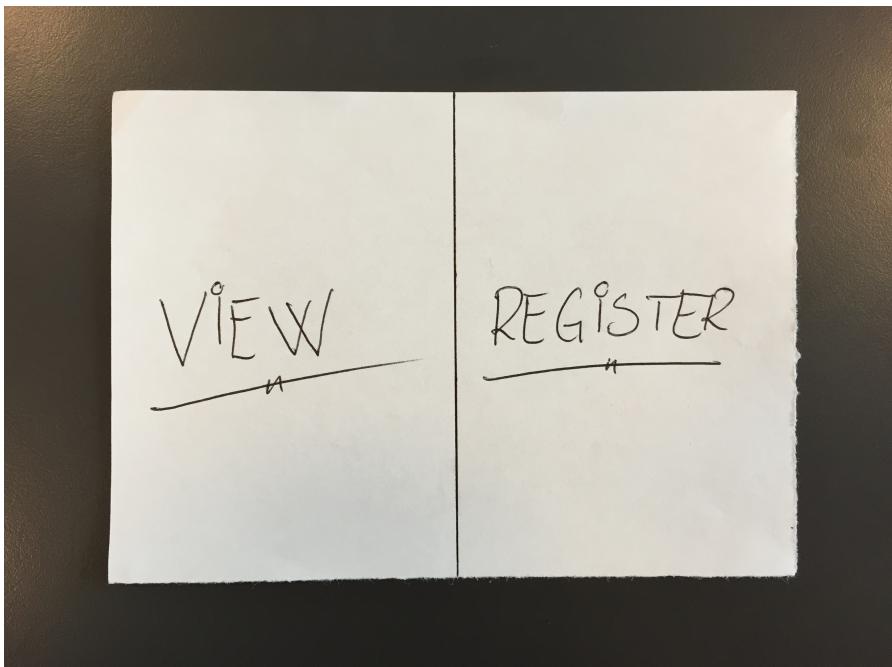
Table of Contents

FLOW; TEAM ASSIGNMENT.....	1
LO-FI prototypes.....	1
Usertest.....	2
Screendumps; final product	3
Evaluation.....	5

LO-FI prototypes

We made a lo-fi prototype in class of our program. This is what we came up with.

So the idea was that once the program is executed, you'd start by having two choices. In correspondence with the handed out assignment requirement for version 1.0, it should be able to view and register a list of candidates for a team. Also it should be able to save and load from a file.



Figur 1: Start window

Figur 2: A list of all entered people. Via the links on screen, you're able to either go back, register and move around on the persons.

The image shows a hand-drawn wireframe sketch of a user interface. At the top center, the word "REGISTER" is written in a large, bold, sans-serif font. Below it, there are two buttons: "BACK" on the left and "VIEW" on the right, separated by a vertical line.

Below the buttons, there are four input fields arranged vertically:

- Administrator:
- Analyst:
- Creative:
- Finisher:

To the right of the first input field, there are labels: "FIRST NAME" above the second input field and "LAST NAME" above the third input field.

Below these fields is a large rectangular box with a double-line border, containing the placeholder text "WRITE YOUR COMMENTS HERE..."

At the bottom of the main screen, there are three buttons: "SAVE" (with a crossed-out "X" over it), "DELETE" (with a crossed-out "X" over it), and another button that is mostly obscured by the "DELETE" button.

Below the main screen, there are two smaller rectangular boxes:

- The left box contains the word "CHANGES" and "SAVED" below it, with a large "OK" button at the bottom.
- The right box contains the text "ARE YOU SURE" followed by "you WANT TO" and "DELETE this Person". It includes two buttons at the bottom: "YES" and "NO".

Figur 3: The register screen where you add a new person to the list. Here you can also choose to delete any person from the list.

UserTest

In our user test we would select a co-student from our class in order to test a person who'd then have background knowledge to- or purpose of the program.

Our selected person, Andreas Poulsen, accepted to participate in our user test.

We showed him our start screen and asked him, what would he do?

He 'clicked' on the view-link and we switched to the view screen. He then realized that the list was empty, since noone had been entered into our list, so he clicked on the register-link and we switched to the register screen. He then imaginary inserted data in the form and saved. He then clicked back on the view screen to see his list.

Andreas' feedback was very important to us because of the current program's actual missing loading-from-a-file part.

We also talked with the tutors about coding the program based on our lo-fi prototype combined with Andreas' feedback.

This, however, meant that we sort of had to rethink our entire program. Instead of multiple Frames, we designed an entire new with only one Frame.

Screendumps; final product

Here's some pictures of our final product.

The screenshot shows a Windows-style application window titled "Person profiles". The interface includes a top bar with standard window controls (minimize, maximize, close) and a toolbar with "Add", "Edit", and "Delete" buttons. On the left, there are input fields for "Name" (containing "Jonas Armandi"), "Administrator" (11), "Creative" (10), "Analyst" (10), and "Finisher" (5). To the right of these are "Comments" (containing "I like Cake") and three buttons: "Add", "Edit", and "Delete". Below this is a table displaying data:

NAME	ADMINISTRATOR	ANALYST	CREATIVE	FINISHER	COMMENTS
Jonas Armandi	11	10	10	10	5 I like Cake
Torben	10	5	5	5	10 Cake is good
Henrik	5	10	5	5	10 Hello
Bob	10	5	10	5	Hello
Lars	10	5	10	5	I dont like cake
Jonas Armandi	11	10	10	10	5 I like Cake

At the bottom are three buttons: "Save", "Load", and "Print".

Figur 4: Our final product.

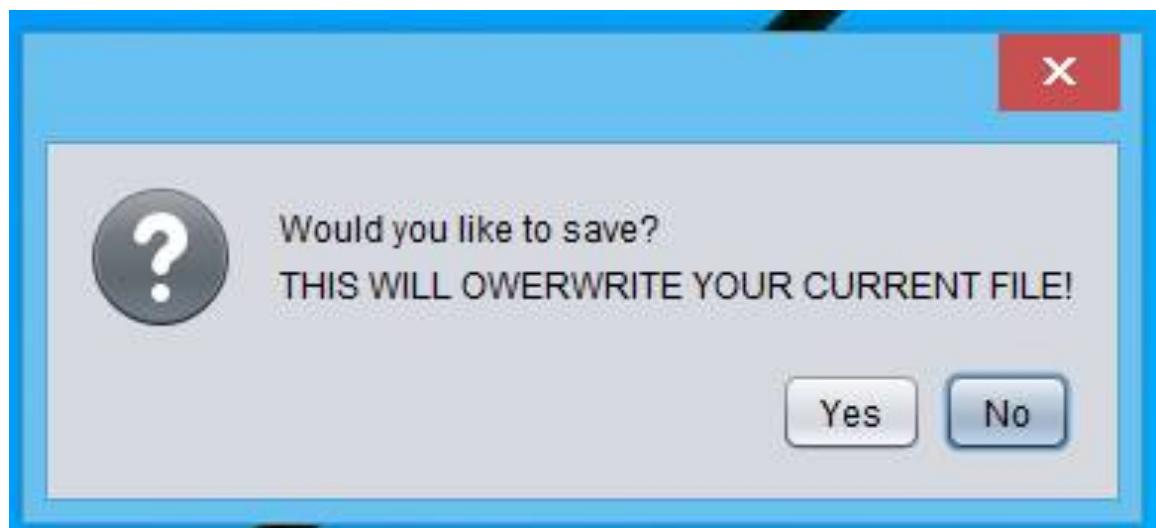


Figure 5: Checking if you're sure to save.

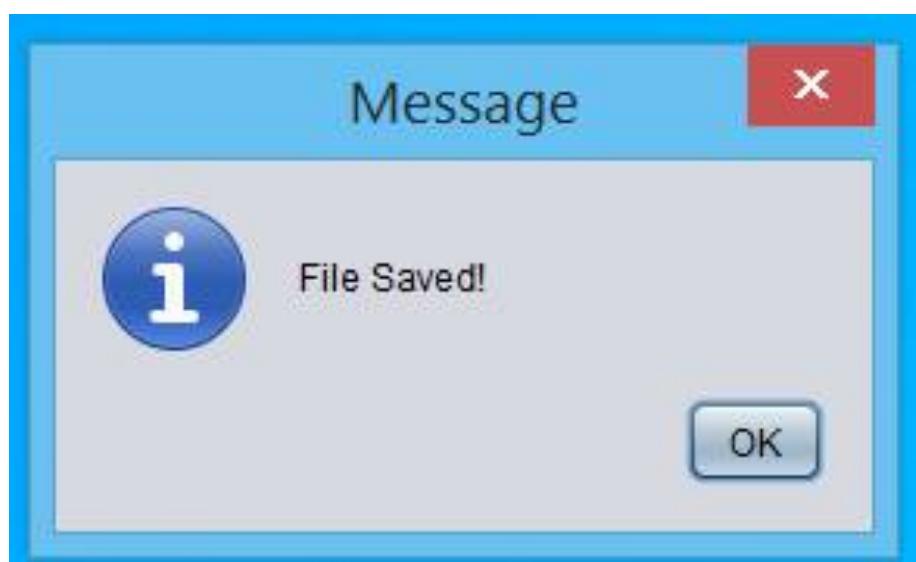
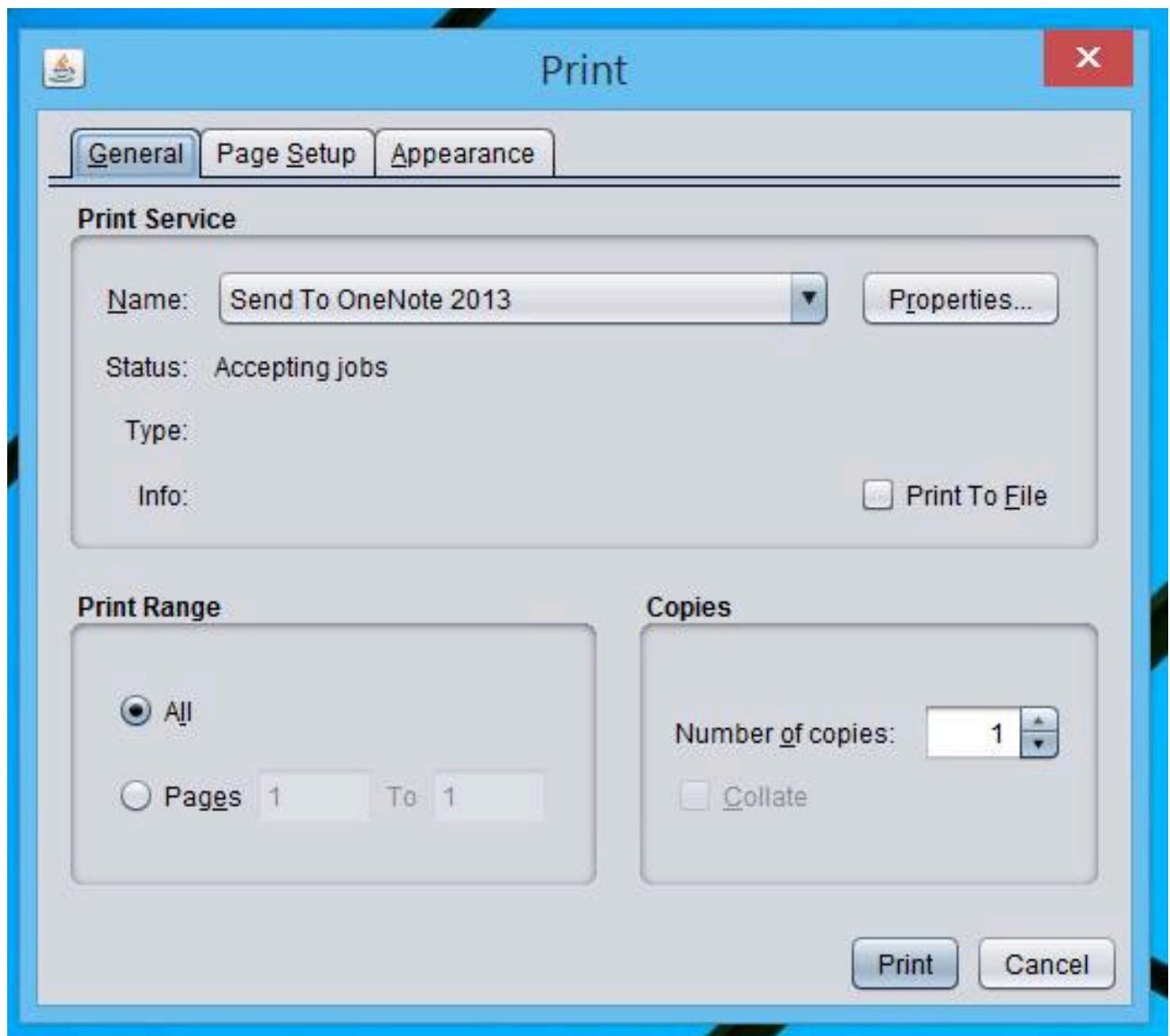


Figure 6: Status of saving job.



Figur 7: Extra feature we added!

Evaluation

During the development of our program we had some difficulties, but with a lot of help and some cool youtube videos we actually figured most things out.

It was confusing that one guy helping us saying that using JTables is a good thing and then the next guy helping us saying that it's a bad thing. In the end we came through with our wish to use the Jtable and got the basic table up and running with an add, edit and delete to/from the table feature.

After we implemented the coding from the FileHandlerStat.class for loading and saving, which was very complex, and got it to work, we added a pop-up box with a 'are you sure?' option, and then it shows OK.

We thought it'd be cool for the team-leader if we had a printing method there as well, and we found that build-in method on the JavaDoc.

We think that the program runs well and it meets the team-leader's requirements.

We also made a user-test of the actual program with the same test person as before and Andreas was completely blown away! No, but he did like what we had done with the final product.