Process report

Group 4

Group members:

Martin Chingov

Ivaylo Hristov

Ivan Stoilchev

Kaloyan Ivanov

Table of contents:

1. Division of labor amongst the group members: who has accomplished what task; which tasks were carried out as a group.
2. An explanation of the choices we made for our program.
3. The problems we encountered and how we solved these problems.
4. Evaluation: everybody's personal view on the assignment and why this project was important for us (what we learned from it).

Division of labor

URS:

During the working process we were able to divide our work for the URS very equally. We tried to gather as much as possible to do as much as possible for the deliverable at hand. In that part we tried to do everything as a team and discuss the things we were going to do and give as a final version.

Design document:

In this part we were also trying to work together as much as possible. Of course that was not always feasible but when we could not gather we used VOIP software to talk and make decisions about the design and the user experience that was to be implemented later in the project.

Source code:

For the source code we used GitHub, which helped us work on the project simultaneously even if we weren’t gathered. We used Google Drive to split our tasks, so it was easy for everybody to check the progress and get on a task. We split the classes in such way that everybody can work on the application without having to wait for somebody else to complete a certain task. We split the component classes equally so they would be ready quickly. Afterwards Kaloyan started implementing the drawing methods for the links, while at the same time Martin was doing the drawing methods for the components. Then we split the work in the Form class so everybody can implement the paint events for his part. Ivaylo and Ivan took a major part in removing the components/links, and saving the state of the project as well as loading a previously saved one.

Choices made

After we started coding we encountered most of the problems mentioned above, so we needed to change the class diagram for something that was going to work out for us. We removed the classes which were going to be responsible for the types of inputs/outputs from components, and we also had a hard time implementing the exact things we wanted to and the workings of the methods were not as we desired

Problems encountered

Our team was good and that’s why we didn’t have a lot of problems. One of the things we had to fix a few times was the class diagram, because we weren’t sure what we wanted and we had to add some new ideas which led to updates to the class diagram. And after we thought it was good, we forgot about one major thing in C# and that is that there could not be multiple inheritance. So that led to a major change in the structure of the project (code) as now we don't have some of the classes we stated in the class diagram which we thought would be useful. Because of that we had to move the code which was supposed to go into the missing classes to the ones who were going to inherit from them. The other main problem we encountered during the project was that some of the method calls in our project were organized very poorly or entirely misplaced, which resulted in incorrect and weird behavior, which we unfortunately only had enough time to partially fix.

Evaluation

**Ivan Stoilchev**: For me the assignment was interesting. We had to do it in a group and that was positive for our communication skills and also it was fun to do it with friends. I think we all learned some new stuff from the assignment. For me the most important part was communicating with everybody and writing the code.

**Kaloyan Ivanov**: Having the opportunity to work on a project with a group of more than 2 people has been beneficial to me. It taught me a lesson on how we are all think differently and might have different ways of working but when put together as a team we must act as one and fill the holes of the other member which can be interpreted as supporting one another. In the future it will help me when I start working in a company because most probably there no one is working alone and everyone has a team, depending on the company usually of at least 5 members, which is really close to what we did in this project. I also have to say it was quite challenging. Now after done with the project I see why the waterfall method is used less and less in the ICT field as it is very strict which comes of that it is sequential so you do not have the advantages of the agile approach to have iterations which lets you go back and fix the thinks you did not like and be more flexible. In the future I would like to be more focused and try to have a good vision of what I what to accomplish. As to the project itself I must admit we were not in the right direction with the class diagram or should I say the URS and the design document as a whole. I only now understand and appreciate the value of good User Requirements Specification and the Design Document. A lot was missed by us and we should have elaborated more and to plan better with the idea to to achieve more than the minimal requirements. In the end it was an extremely helpful experience. If I had the opportunity to do anything different it would be to even more dig into the classes and optimize a lot of our application more so that it is even more object oriented and also leave more time to handle all the exceptions and small bugs interfering with the user experience.

**Martin Chingov**: The project really helped me to understand how important is to work in a team and also how to use the graphical class. I can say that it is very similar to the ProCp subject that we have at the moment. The only difference is the methodology that we use in both projects. With OOD we have to follow the Waterfall method, which is very straightforward and every step depends on the previous one, which means that a single mistake in the past may lead to very serious problems later. While doing the documentation, I was personally very uncertain how the project could succeed if we have to implement everything in the application as we have described it in the documents. That is why we had a lot of feedback from Mr. Gestel, our project mentor, who navigated us and helped us realise how the workflow should happen using this methodology. I think we learned a lot about how plans may go wrong anytime and how we must handle all these problems as a team.

**Ivaylo Hristov**: The distribution of tasks across time for this project was very helpful, interesting and thought-provoking. Usually, we are forced to start working on code not long after we come up with some initial ideas on how to implement something, if not immediately after, but in OOD2 we were given most of the block to only focus on documentation and think of the most efficient and elegant ways to implement the program before even starting to work on it, resulting in what I believe is a far better grasp of how to actually go about implementing the program, even if we stumbled on some parts during the actual coding.