**Conference Management System (CMS)**

**Documentation**

**IDE**

One of the first steps we took in approaching this project was chasing the tools. This led to the ultimate decision of using Visual Studio as IDE, C# as language supported by .NET framework.

Firstly, we chased Visual Studio for being a tremendous IDE which enables easier app development for any platform in all languages, also taking into consideration its extensions which facilitate code writing in a great measure.

**Programming language and Framework**

In terms of framework and language we chased NET C#. It was a decision taken by all the members who have expressed their wish through voting, a decisive factor being the fact that it was the most popular language between us. In addition, the documentation offered by Microsoft was a great aid. Being highly elaborated and extremely well organized, it helped us find solutions to our answers and queries very fast.

The .NET Framework provides a comprehensive programming model which enables creating all kinds of applications that work for Windows being convenient for mobile, web and even for desktop.

The easiness in working with windows forms and the familiarity of the members with this type of applications were key factors that ultimately led to the decision of designing a desktop application. The decision was the most clever one because it helped us save time and avoid future misunderstandings and conflicts between the members.

When we saw the final result, everyone was pleased for having made this first decision as a team.

**Evolution of the application**

In order to create such an application, it was necessary to take small steps. At first, there were a lot of things that needed to be organized and set up before anything.

One of the first tasks that we all had to do was to better understand the problem by reading it as many times as it was necessary as well as identifying the main story lines and flows. We divided the problem into four distinct phases in order to make sure that what we will have accomplished until each phase it will be functional.

After making sure that we all understood the requirements of the project, the next step was to install the necessary tools and to get familiar with them.

Diagrams followed to facilitate understanding and writing code in the future. We all agreed that for better knowledge regarding the project, we worked together on all the tasks that included diagrams: case diagrams, class and sequence diagrams.

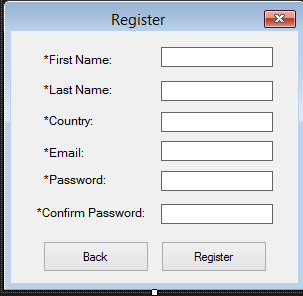
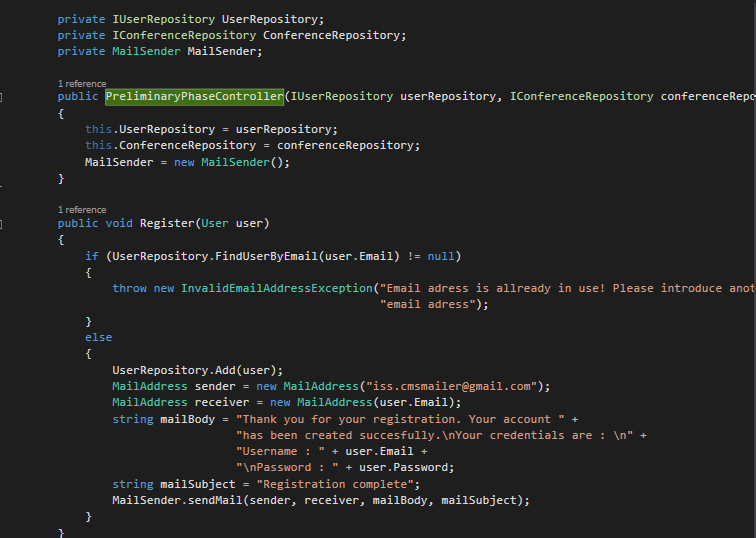
When we all finished the group work, individual tasks were assigned and we were able to start coding. Implementation is probably the most difficult phase in the development of a software application. However, we did not seem to have too much trouble with this. Having the diagrams well made, implementation has gone quite well. Some of us have worked on the backend side, others on frontend or testing, alternating for each phase of the project. Binding all of this together was another important and difficult task. Fixing bugs and warnings gave us some headaches but we managed to succeed. Everything that we worked went through a review made by our leader to ensure that everything is written correctly and in accordance with the conventions originally set.

After the code was developed it was tested against the requirements to make sure that the application is actually solving the needs addressed and gathered during the requirements phase.

As soon as we finished testing, the application was ready and can be used without any problems.

**Presentation of the application**

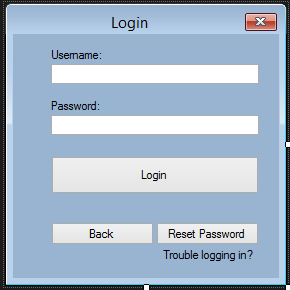
**Register form**

We have a register form connected to a PreliminaryPhaseController that enables user registration.

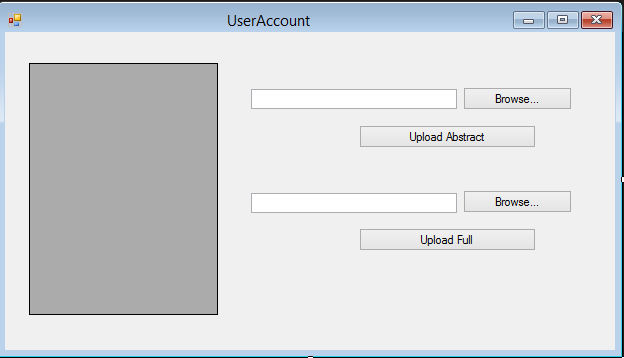
Once the account is validated, the submitter logins, submits the abstract and fulfills the meta-information required: the name of the proposal, the keywords, the topics, the possible list of authors and their meta-information.

We did this with Login Form and with UserAccount form.

**Login Form**

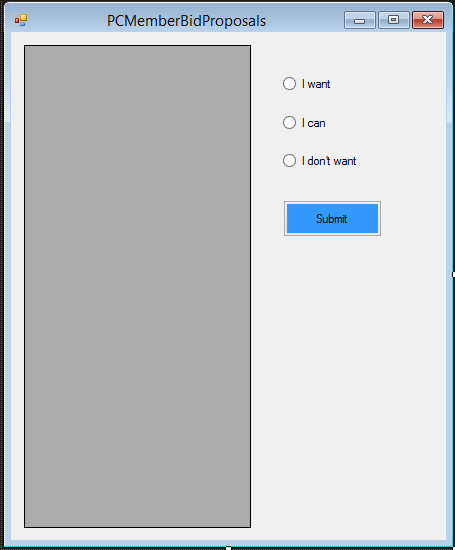
****

**UserAccount Form**

****

In the second phase, PC members are required to bid the proposals. We did this with another form.

**PCMemberBidProposals Form**

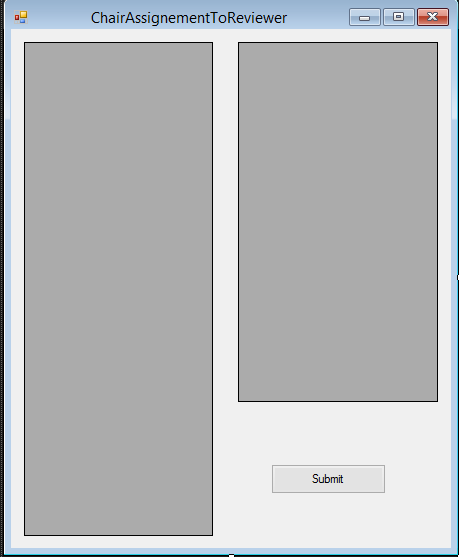
****

Each PC member has to do a brief analyze of abstracts or papers in order say if :

* they are pleased to review some papers
* they are could evaluate some papers
* they refuse to evaluate other papers.

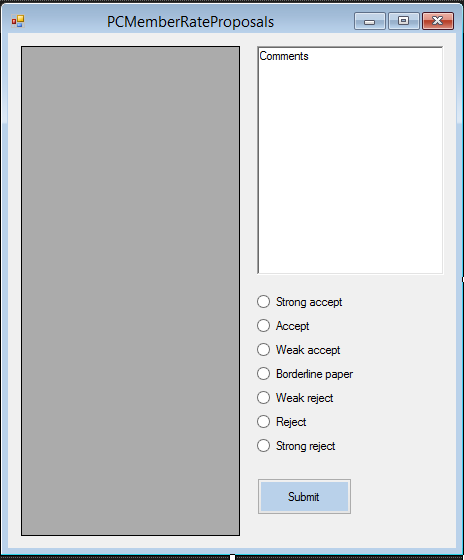
Once the bidding process closed (there is a deadline for bidding), the conference chair or co-chairs assign to each reviewer the papers required to be evaluated.

**ChairAssignementToReviewer Form**

****

The result of each reviewer carries one of the following qualifiers: strong accept, accept, weak accept, borderline paper, weak reject, reject and strong reject. The papers whose evaluation does contain any level of reject are accepted by default.

**PCMemberRateProposals Form**

****

Submissions having been rated, the authors are notified via email about their results.

After that process we move to the third phase which concerns the conference activities.

In the third phase one or more conferences are active and the application manages the schedule or other functionalities that makes easier for people to attend the event.

**Team structure**

We worked on four phases, each phase with its corresponding team: a team for front-end, a team for back-end, a team for the tests (testing) and a team for reviewing code.

Each member had the opportunity to work in front-end and back-end teams but only a few had the opportunity to work in the team that reviewed code.

The front-end team’s main task was to present and put data or relationships in context for a good interaction between the application and the client making things look good, as good-looking things sell better.

The back-end team was the team who created logical back-end and core computational logic of a software or information system. The team created components and features that are indirectly accessed by a user through the front-end’s work.

**GIT**

We used GIT for tracking changes in the application and coordinating work amongst the team members.

Every GIT directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server.

We cloned the repository from <https://github.com/ovidiuenache/iss-project> . The project was equally split between the team members. Each member worked on his part of the project and after that the content was pushed on GIT for all the members to see the changes.

**Trello**

Trello is a collaboration tool that organizes your projects into boards. In one glance, Trello tells you what is being worked on, who is working on what, and where something is in a process.

It made it easier for us to assign each team and its members a task and a deadline and we could supervise every process and the way members deal with given tasks.

**What We Would Like to Change**

The application is subject to change as we would like to convert it into a web application. We want to make it portable and easy to use for phones and tablets.