Stage of implementations for the Conference Management System

The development of the of the application began with the requirements elicitations for or future application.

First of all, we decided main technologies used by the program. Mainly the Programming Language, Java because the team is currently most proficient with this language. Afterwards the Database server was chosen as PostgreSQL because of its ease of access for us. The ORM was realised through Hibernate, because that was the first ORM all of the team members had prior knowledge on. For the UML diagrams we used StarUML and DrawIO to create them.

We analysed the text and obtained our use cases, form there we created several use cases diagrams for each stage of the lifetime of a conference. After that those whose didn't create the use cases were doing the requirements, from there we obtained the functional requirements and non-functional requirements. These were obtained by careful analysis of the paper describing the functionalities required by the conference management system.

The next stage was creating the Database diagram, the Class diagram and the Architecture diagram. Here it was decided that our application is going to use Angular as the front-end part of our application and on the back-end Spring, Spring Boot, JPA and all the other regular technologies. We also decided to use SendGrid as our API for sending emails. Liquibase to handle our changes to the database which helped us keep track of the current state of our database. MockFlow was used as a GUI prototyping tool.

At this stage the first iteration of the ORM was finished, there were small alterations further down the line. Here was the first occasion when the we can apply the theory learned on the courses to help our progress in the development. For the front-end part, the GUI was first envisioned in our first GUI diagram.

For the skeleton of the GUI MockFlow was used to as a first depiction, then progressively Angular was used to represent the backbone of the front-end, and to realize the ideas created with MockFlow

At this stage the backend was advanced enough that the implementation of the frontend, GUI could begin.

Following these steps, the implementation had the possibility to advance rapidly. The sequence and communication diagrams allowed us the better understand our goals when implementing the program. The backend part was ready, changes only occurred only at the request of the people working on the GUI and the frontend logic of our application. Given that each team member clearly understood his given part in the development any issue that came up was easily handled. The diagrams were a useful tool to define how our application behaves, and with it. Evidently this part was realised as the las component of our application. From then on, we only had to polish it.