Project requirements document

Project title: POLLAID

Authors: Miguel Sánchez-Brunete Álvarez, Adrián Díez Álamo

Date: 12-11-2020

0. Document versions

Version 1: First delivered document for the first semester

Version 2: Change made by 12-11-2020: reorganization of the group and tasks due to one person left the team and reconsideration about how we want the final product to be.

1. Project components (project products)

- There will be an API (implemented using Spring Boot) implemented during the first and second semester.
- There will be an API (implemented using Python) implemented during the second semester.
- There will be a relational database (implemented using PostgreSQL) implemented during the first and second semester.
- The Spring REST API documentation will be available in Swagger and will be accessible by users. Calls can be made from there as well.
- There will be a web application (implemented using React js) where users will be able to manage everything implemented during the first and the second semester.

2. Project boundaries

- This will be a fully functional application that will be entirely done according to product description previously provided in Project Vision document.
- We do not know what the clients' expectations are and till the first version is deployed and we collect some feedback from real users we won't be able to decide if we can include those features or not.

3. Functional requirements list

First semester

- Creating a user profile through the web application will allow users to create polls and manage the information coming from them (statistics, previous polls, results...).
- Different types of polls will be allowed, such as one-choice, multiple-choice, open questions…

- Users will be able to create surveys from the web application with collections of polls and share the link with other people in order to complete it.
- Users will have the possibility to create questionnaires in which they will be able to provide their opinion (Open-ended questions)

Second semester

- Open-ended questions will be semantically analyzed and compared with the other ones in order to create a poll with the most voted answers and avoid similar / duplicated answers.
- In polls there will be a first round with all the possible answers and after that people will vote again between the most voted options (provided that an absolute majority has not been reached in the first round) - as in the presidential electoral voting process
-
- A Python microservice (API) will be created in order to semantically analyze answers coming from the open-ended questions.
- A new field “Priority” will be created for polls and open-ended questions to remark the priority of the poll / open-ended question for the users.
- A filter will be added in order to sort existing polls / open-ended questions by the status, questions, priority…

4. Nonfunctional requirements list

Comentado [MW1]: do you support all browsers? do you use java in some version, python in some version?

- Java 8 (supporting all browsers)
- Python 3.x

5. Measurable implementation indicators

- There will be an API (implemented using Spring Boot). We will create a Docker image and run it locally with Tomcat.
- There will be an API (implemented using Python). We will create a Docker image and run it locally with Tomcat.
- There will be a relational database (implemented using PostgreSQL) and the main engine will be Hibernate.
- There will be a web application (implemented using React js) and we will run it locally using npm.

6. Project acceptance criteria for semester one

Required:

- Creating a user profile through the web application will allow users to create polls and manage the information coming from them (statistics, previous polls, results...).
- Different types of polls will be allowed, such as one-choice, multiple-choice, open questions…
- Users will have the possibility to create questionnaires in which they will be able to provide their opinion (Open-ended questions)

Expected:

- Users will be able to create surveys from the web application with collections of polls and share the link with other people in order to complete it.

Planned:

- All previous points

7. Project acceptance criteria for semester two

Required:

- Open-ended questions will be semantically analyzed and compared with the other ones in order to create a poll with the most voted answers and avoid similar / duplicated answers.
- In polls there will be a first round with all the possible answers and after that people will vote again between the most voted options (provided that an absolute majority has not been reached in the first round)
- Like a presidential electoral voting process
- A new field “Priority” will be created for polls and open-ended questions to remark the priority of the poll / open-ended question for the users.
- A filter will be added in order to sort existing polls / open-ended questions by the status, questions, priority…

Expected:

- A Python microservice (API) will be created in order to semantically analyze answers coming from the open-ended questions.

Planned:

- Test of the application in real-world conditions with real users

Comentado [MW2]: added planned point

8. Teamwork organization

Adrián Díez Álamo: Back-end developer

Miguel Sánchez-Brunete Álvarez: Project manager & back-end developer:

- Will be in charge of defining the tasks to be done, as well as distributing them among the different members of the team and checking that everything meets the desired quality standard

1. The tasks are stored in Jira's backlog and as the sprints are finished, which do not have a fixed duration, the tasks are assigned to the developers taking into account their role in the project.
2. The duration of the sprints depends exclusively on the workload of each team member. If in a sprint not all tasks are completed, they will be passed on to the next sprint along with others.
3. All the updated documentation regarding the project can be found in our Confluence page

Comentado [MW3]: where do you storee tasks? decided is task is done? when you mere task to common branhc? what tools do you use (git, jira?)?

4. All tasks are created in JIRA and as soon as the task is started the task is moved to “In progress” section, and when the task is finished (for coding tasks it is done when the pull request has been accepted and the branch merged to master) is moved to “Done” section.
5. We use GIT

9. Project risks

- There is a risk that the metric used for the semantic analysis will not be as accurate as planned.

10. Milestones

Comentado [MW4]: moved this to teamwork . You HAVE TO have a milestones like finishin developmetn of X before that date or develop this, this, this in stage 1, this, this, this in stage 2 if there is no plan it would never work. So I believe that there is some plan, please describe it

- Finish development before 18.01.2021
    - Working at the same time in the webpage and in the Java API
- Finish Python API before 15.12.2020