# *Sprint retrospective, Milestone #3*

| User story | Task | Assignee/s | Estimated effort | Actual effort | Done | Notes |
|---|---|---|---|---|---|---|
| Standardise "Submit Button" by extracting it into a component, such that all secondary pages use the same button. | #48, #49 | Vlad Nitu + Rares Toader | 2h + 3h = 5h | 2h + 4h (comple teaza Rares) | Completed | Vlad: After the  end of sprint #2 standup, we agreed to  follow the next approach: we extract the button that was previously used in "checkURL" and "checkTwoURLs", respectively, into the previous component `submitButton.js` that was used in the other two secondary pages: "checkText" and "checkTwoTexts". We do this in order for all secondary pages to use the same button. Rares: Using the same button proved to be tricky, and it caused some problems for small screens for the "checkText" and "checkTwoTexts" pages. I fixed this issue, along with another small screen bug, which was the services in the main page, which looks alright now. I also made some small changes such that the keyword "plagiarism" was no longer displayed. |

| | | | | | | |
|---|---|---|---|---|---|---|
| As a user, I want to see a progress bar that visually represents the percentage of overlapping between two articles | #51 | Vlad Nitu | 4h | 5h 23m | Completed | Implemented two progress bars, one circular (using linear interpolation between green and red) and one linear (stacked 5 colors: Green, Yellow, Light Orange, Dark Orange and Red) + a pointer to show what percentage of overlapping the article obtains. @cciacu helped in approaching the third-party library used for the circular progress bar by adding the percentage of overlapping inside the circle. |
| Research algorithm feasibility solutions | #47 | Vlad Nitu, Diana Micloiu, Mirica Matei, Cristian | - | 14h 12h 7h 15h | closed | Tried different alternatives to make trade-off accuracy vs efficiency, such as: subsampling, modifying the hash function, reading Diana's proposed paper that got great results by using Winnowing with a specific set of parameters and implemented Rabin-Karp hashes. Only the Diana's paper approach remains in the latter code, as Rares's two collection approach seems the most efficient in practice |

| | | | | | | |
|---|---|---|---|---|---|---|
| As a user, I want to enter the URL of two news articles, so that I can be able to see a rendered side-by-side comparison of the two web pages. | #6 | Cristian Ciacu | 15h | 18h | Closed | Add side-by-side rendering of two news articles. |
| As a user, I want to enter the link of a news article, so that I will be given an overlap warning if the article has high similarity with one document in the database. | #12 | Rares Toader | 8h | 9h | Completed | Finished implementing the connection between the frontend and backend, with a version that worked decently fast so far. Main challenges were to persist the news articles which took some time, but it was needed to get an accurate estimate of how long the server took to respond. Another challenge was to be able to parallelise some of the code to make it work faster. I also used the improvements suggested by Diana and Matei (fast computation of Jaccard Similarity), Vlad (saving in batches to make persisting way faster), Cristian (only considering most informative hashes). |

| | | | | | | |
|---|---|---|---|---|---|---|
| Refactor one of the frontend components to include a design pattern (strategy). | #50 | Rares Toader | #8h | #2h30m | Completed | In order to promote code reusability and extensibility, I refactored two types of already existing components (with 3 instances each, thus 6 components), into only two components, according to the strategy pattern. `BodyCheck…` has been refactored into `BodyCheckGeneric` and `ForwardToX..` has been refactored into `ForwardToPage`. I severely overestimated the time it would take to implement this feature, as I thought I would need to make a lot of changes, but this was in fact easier than expected. |
| As a user, I want to enter the link of a news article so that I will get back similar articles, together with their plagiarism similarity ratio. | #13 | Diana Micloiu, Matei Mirica | 6h | 5h | Completed | Modified the API for */urlsimilarity/* to integrate retrieving the five most similar articles together with their similarity ratio (in decreasing order). This was one of the must haves we had left and is now completed by using a fixed length min heap structure. |

| As a user, I want to enter the link of a news article, so that I will get back similar articles, together with their respective URLs | #11 | Diana Micloiu, Matei Mirica | 6h | 5h | Completed | Created a special ResponseUrlEntity to encapsulate all the details (url, similarity and in the future other attributes) of the similar articles that need to be sent to the frontend as a response. Modified the front end to integrate displaying the similar articles with their URL, similarity and the bar visualization implemented by Vlad |
|---|---|---|---|---|---|---|
| | | | | | | |