SLM – WS2022 – Semesterprojekt „Maintenance Monitor"

Content

Basic information

Authors: Adolf Nathalie (ic21b045), Gelbmann Andreas (ic21b066), Stranz Daniel (ic20b044)

Github: https://github.com/ic21b066/WS2022_SLM_Proj.git

Please see README.md for instructions how to run the program (usage of the software).

Overall process

Team building

The team members agreed on equal payloads as good as possible, which means each member contributes to coding, architecture tasks and documentation. Minor tasks have to be acknowledged by all team members.

User stories

As the scope of the project is stated clearly and the team agreed on the conclusion that the coding part is rather simple, we decided to work with the basic form regarding user stories, though we share the opinion, that in larger, more complex projects a deeper insight via more detailed user stories benefits the outcome.

Github account and public repository

As required a GitHub-account was created and a public repository opened following the instructions provided by the course.

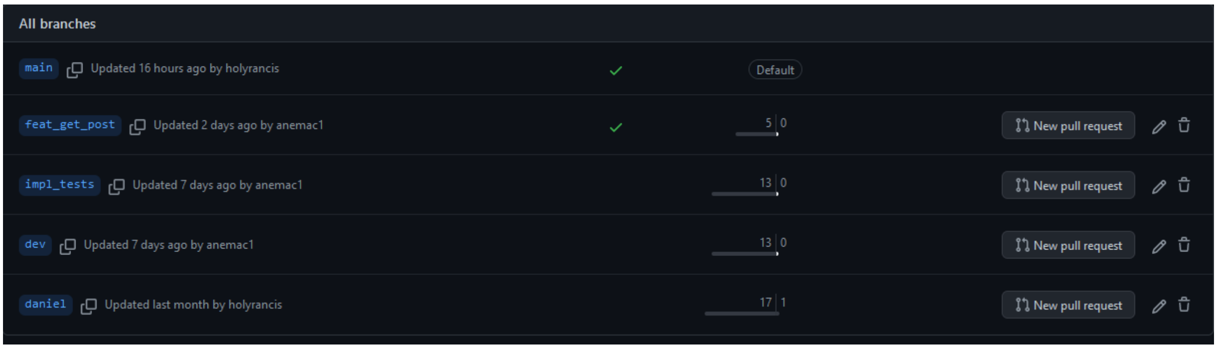
Development

Different approaches

In order to maximize the personal output we decided to not only follow the instructions by the course, but also to build on various documentations provided by the creators of the used tools and software, e.g.: spring.io, jetbrains.com, tutorialspoint.com

Different branches

After the needed knowledge concerning implementation skills was acquired the branch structure was remodeled. Naturally a dev and main branch are part of the project, some rather personal branches have been omitted throughout the implementation process.



All branches

main	Updated 16 hours ago by holyrancis	✓	Default		
feat_get_post	Updated 2 days ago by anemac1	✓	5 | 0	New pull request	
impl_tests	Updated 7 days ago by anemac1		13 | 0	New pull request	
dev	Updated 7 days ago by anemac1		13 | 0	New pull request	
daniel	Updated last month by holyrancis		17 | 1	New pull request	

Code documentation

As stated above the coding assignment is evaluated to be rather trivial and accordingly we only use the basic functions, which are provided by the course. Code documentation is dealt with in a feasible way.

Unit tests

Due to the early start we could not implement a test-driven-development approach from the beginning, as it was introduced in a later stage of the course. Agreeing on the presented benefits of this approach we decided to put a fair amount of work into the unit tests, as they can also function as code documentation.

Github action and artifacts

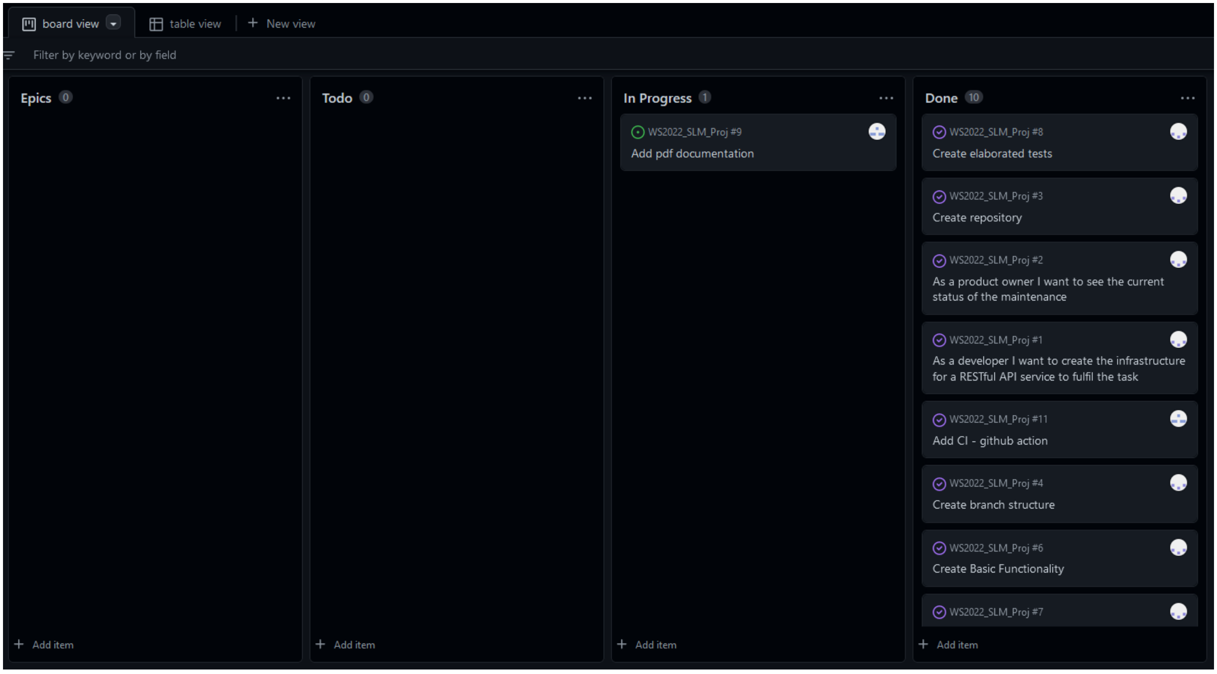
Following the structure of the course the last steps in order to finish our project were implementing continuous integration and delivery via the GitHub action platform ending in a tested .jar file for convenient export.

User stories

As stated above the user stories were written in the most basic form, as they are teached on UAS Technikum Wien („agile project management“), though we acknowledge that more elaborate user stories might be more suitable (to more complex projects).

After receiving the information about the preferred way of task documentation of the lecturer the procedure was changed to define short named user stories with an elaborate Todo description.

Github Kanban – Example



Epics 0

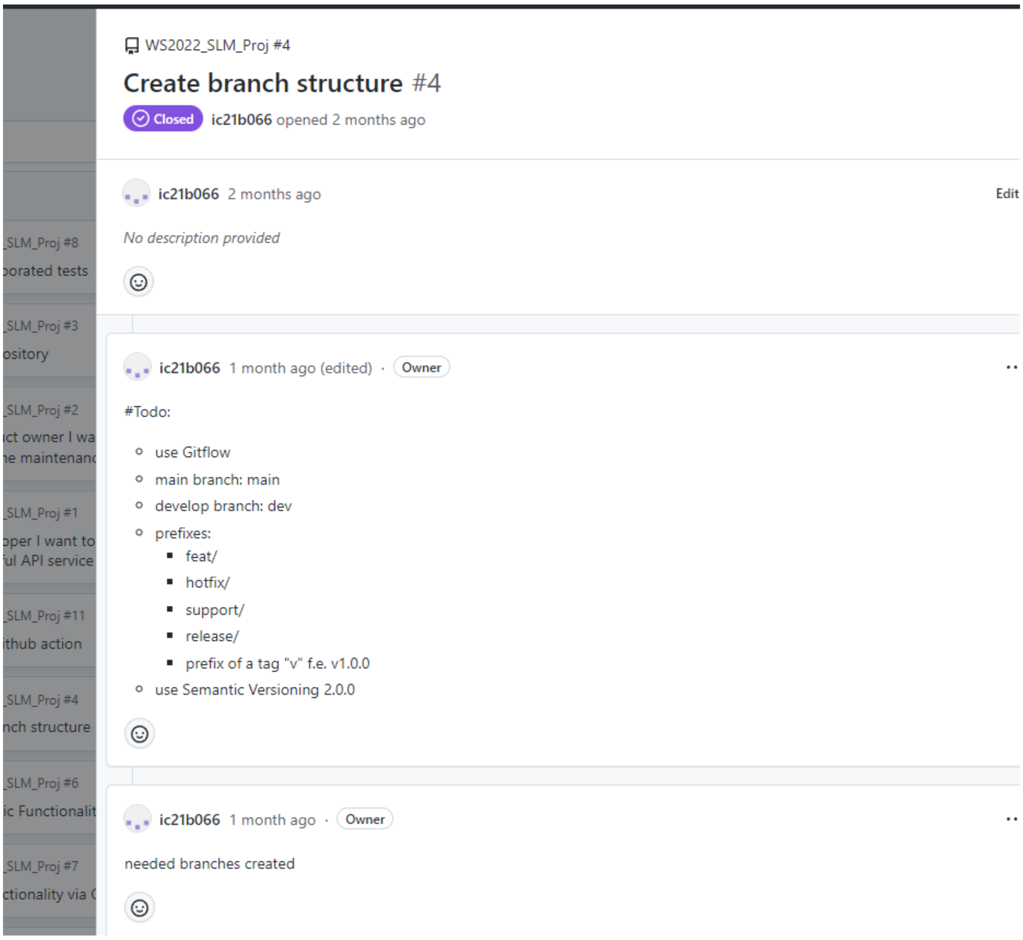
Todo 0

In Progress 1

- WS2022_SLM_Proj #9
  Add pdf documentation

Done 10

- WS2022_SLM_Proj #8
  Create elaborated tests
- WS2022_SLM_Proj #3
  Create repository
- WS2022_SLM_Proj #2
  As a product owner I want to see the current status of the maintenance
- WS2022_SLM_Proj #1
  As a developer I want to create the infrastructure for a RESTful API service to fulfil the task
- WS2022_SLM_Proj #11
  Add CI - github action
- WS2022_SLM_Proj #4
  Create branch structure
- WS2022_SLM_Proj #6
  Create Basic Functionality
- WS2022_SLM_Proj #7



WS2022_SLM_Proj #4

Create branch structure #4

Closed ic21b066 opened 2 months ago

ic21b066 2 months ago

No description provided

ic21b066 1 month ago (edited) · Owner

#Todo:

- use Gitflow
- main branch: main
- develop branch: dev
- prefixes:
  - feat/
  - hotfix/
  - support/
  - release/
  - prefix of a tag "v" f.e. v1.0.0
- use Semantic Versioning 2.0.0

ic21b066 1 month ago · Owner

needed branches created

Usage of the software

Github

Github is used for its version control, Kanban board and CI/CD abilities.

Springboot

As recommended by the course we used SpringBoot to embed the necessary web server.

Insomnia

Insomnia was used to check the implemented REST application functions.

Postman

Postman was used to double-check the implemented REST application functions.