Rubric

Legend:

- 🟩 Done
- 🟨 To be decided
- 🟥 Not done
- 🟦 Notes

Features: (Base of 2.4 with 1 bonus point)

- The code compiles (0)
    - All features of the basic assignment are present (2.4)
    - *Additional features (or improvements of existing features) (0.1-1)*

Scrum: (Base of 2 with no bonus)

- Every week a screenshot of the scrumboard is uploaded to the git repository (0) Some screenshots from before the Christmas Break are missing. We didn’t upload it to Github.
    - Every week there is a meeting with the TA (0.2)
        - With an agenda for the demo weeks (0.1)
        - While meeting-notes are being made for the demo weeks (0.1)
        - Where every round someone else is chair (0.1)
        - Where everyone is asked what they did and if they have blocking issues. (0.1)
    - There is a scrum board (0.2)
        - With a “backlog” (0.1)
        - With a “ready to start” (0.1)
        - With a “done” (0.1) We had this at the start but decided to just move issues that are done to closed (since we close issues with pull requests via close keywords)
        - Which has <u>everything</u> that the team needs to do (0.1)
        - Which is being used as a tool, not as paperwork. (0.1)
        - On which issues reside (0.1)
            - Every issue starts in the backlog or ready to start
            - Every feature is in the form of a user story
    - Every sprint a sprint plan is made (0.1)
        - On which they explain what they will be doing this sprint. This is committed to the repository (0.1)
    - Every sprint a sprint review is held, to see what can be done better (0.2)
        - They reflect if in the past proposed improvements have been implemented correctly. (0.1)

Code: (Base of 1.5 with 0.4 bonus points)

- The code compiles (0)
    - *Libraries are used where necessary (0.1)*
    - Checkstyle is being used (less than 20 errors) (0.2)
        - Checkstyle is well used(less than 5 errors) (0.4)
        - *Checkstyle is being used perfectly (less than 5 warnings and every suppressed warning has a good reason) (0.1)*
    - There is being tested (0.1)
        - More than 70% is being tested (0.8)
        - *More than 95% of the code is covered (0.1)*
        - *Mockito is being used to test certain classes (0.1)*

UML (Base of 0.5 with 0 bonus points)

- A correct UML diagram containing the “emergent architecture” is created following the UML guidelines

Report: (Base of 1.5 with no bonus points)

- There is a report, which contains a weekly screenshot of the code-coverage of the team and a weekly screenshot of the scrumboard.
    - [for every item that should be in the report] is correctly written and described why the choice was made (1)
        - extensively and perfectly described (0.5)

Git: (Base 0.6 with 1.0 bonus point)

- There is a git repository, which contains all the code the team made. All team members contributed evenly (0) Some members contributed more than others but we tried to balance this out during the last couple of weeks.
    - There is a readme, where each member separately put their netID and picture. It also explains the goal of the project. (Only valid in 2nd week) (0.1)
        - *There is a great readme (https://jfhbrook.github.io/2011/11/09/readmes.html) (0.1)*
    - Correct commit messages are being used (0.1)
    - There is a .gitignore, which contains all files that should never be committed. (0.1)
        - *Jars of libraries are put in the pom.xml and added to the .gitignore (0.1)*
    - Branches are being used (0.1)
        - *Branches are being used correctly (name and feature branches) (0.2)*
        - *Pull based development is being used (0.4)*
        - *A ready to review is added to the scrum board (0.1)*
        - *Continuous integration is being used (0.2)*
    - *Every week there is a release with a tag (0.1)*

Presentation (Base of 0.75 with no bonus)

Q&A (Base of 0.75)

- Questions are being answered (0.25)
  - Questions are being answered in a way that shows that the students know a lot about the topic of the question. (0.5)