

Week 2

Chair: Sever

Notes: Irtaza

Goals:

Research about serverside development and spring. And how the client sends the request and the sever responds to it.

Next, meet up on **Wednesday 20/02/2019** to discuss the server side and how we will start programming it.

This week:

Week 3 (0.4)

- A first version of the client is working (0.1)
- A first version of the server is working (0.1)
- The client can send a request to the server and it can respond to it. (0.2)
- Agenda
- Sprint plan
- Meeting notes
- Scrum review
- Checkstyle report
- Code coverage report
- The release of our current functionality

Authentication:

- Have an email and a password

Database:

- Create a database and use a web API to connect with <https://www.carbonfootprint.com/calculator.aspx>

Application Components and Libraries:

Serverside(Back end): Spring

Clientside (Front end): Spring MVC

GUI: JavaFX

Server + Database: JDBC

Features (1.2)

- Eating a vegetarian meal (0.1)
- Buying local produce (0.1)
- Using bike instead of car (0.1)
- Using public transport instead of car (0.1)
- Lowering the temperature of your home (0.1)
- Installing solar panels (0.1)
- Track the CO2 that you save and compare to your friends (0.2)
- Provide badges, achievements & other stimuli (0.2)

Week 6 (0.6)

- The full workflow of "Eating a vegetarian meal" is implemented (0.2)
 - User clicks on a button to indicate that they bought a vegetarian meal (0.1)
 - The client sends a json request to the server (0.1)
 - The server stores this data in a file or database (0.1)
 - On closing the client and reopening it, it is able to request the data from the server and show it to the user. (0.1)

Week 8 (0.8)

- 4 of the 6 minimal features from Food, Transportation and Energy have been implemented (0.4)
- The full workflow for "Track the CO2 that you save and compare to your friends" is implemented
 - You are able to have an overview of the CO2 that you have produced. (0.2)
 - You are able to have an overview of the CO2 that others have produced. (0.2)

Notetaking:

Scrum: (1.5 points)

Scrumboard (0.4)

- There is a scrum board with a “backlog” and “done” (0.3)
 - Which is being used as a tool, not as paperwork. (0.1)
 - Which contains all work that needs to be done (0.1)

The process (1.1)

- Every week there is a meeting with the TA
 - With an agenda that is committed to git (0.2)
 - While meeting-notes are being made and committed to git (0.1)
 - Where every week someone else is chair (0.1)
 - Where everyone is asked what they did and if they have blocking issues. (0.1)
- Every week a sprint plan is made in GitLab on which they explain what they will be doing this week. (0.3)
- Every week a sprint review is held, to see what can be done better, which is committed to git (0.2)
 - They reflect if in the past proposed improvements have been implemented correctly. (0.1)

Code: (1.5 points with 0.5 bonus points)

- Libraries are not used where necessary (0.4)

Checkstyle

- Every week a Checkstyle screenshot is committed to Git (0.2)
- Checkstyle is being used (less than 20 errors) (0.3)
- *Checkstyle is being used perfectly (less than 5 warnings and every suppressed warning has a good reason)* (0.2)

Testing

- Code is being tested (0.1)
- More than 80% is being tested (0.4)
- *More than 95% of the code is covered* (0.2)
- Mockito is being used for at least 5 different classes (0.3)

Report: (0.75 points)

For this project a report of the process, design decisions, points of improvement and individual feedback needs to be made. The full requirements of these can be found in the introduction slides on BrightSpace.

Git: (1.2 points with 1.1 bonus point)

Organisation

- Certain things are added to the README of the repository before the end of the second week (0.1)
 - Each member has made a commit to add their own netid and picture to the README
 - Each member has committed a personal development plan
- 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)
- *There is a **great** **readme*** (0.1)

Process

- Correct commit messages are self-explanatory (0.1)
- Branches are being used correctly and get sensible names (no names of people) (0.2)
- The default CI service is in use for at least 4 weeks of the project (0.2)
- *Custom CI configuration has been set up to use the latest features of Java for at least 2 weeks of the project* (0.2)
- *An example of a relatively large PR is given that follows best practices. (Code review, approvals, waiting for CI, etc.)* (0.2)
- *There is evidence of sustained code review taking place.* (0.2)
- *Pull based development is being used for every merge. Master is protected* (0.3)
- *Every week there is a release with a tag* (0.1)

Presentation (1.3 points)

At the end of the quarter, you will hold a presentation about your project. This will be judged by a member of the teaching team. For this part, you are able to receive 0.5 points. A guideline will be made available later in the course.

Question and Answers

- Questions are being answered (0.3)

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

Responsible Computer Science (0.5 points)

During this course, you will have to write in your report about the morality of the work you are doing. This part will be 0.5 of your grade. More information about this part will follow.

Personal Growth

During the course, you will be asked to achieve a personal goal. Due to the fact that this part is very hard to grade, it has been decided to not have an explicit part in the rubric about this.

Information Literacy (0 points)

For this course an exercise on Information Literacy has to be made. If you have not done this 0.5 points will be deducted from your grade.