# meeting 1: *11 November 2020*

do this with moscow method

### **security requirements**

* should be an authentication system (username and password) username is **netID** password is gonna be decided by the user, then it has to be stored securely
  + won't be working on the whole authentication system, only parts of it
  + will have to create database and API to connect to it
  + no need to have a registration area
* this won't be connected to TU Delft sign-on
* netID will be unique ID identifying each user
* recommended is Spring Security for all of this

### **general description**

* tu delft wants to offer
* students on campus every 2 weeks
* entire software needs to be modular

### **tech specs**

* needs to be modular to change with changing rules
* communicate through APIs
* needs to be scalable for high demands
  + high demands will have to be figured out on our own (500 1st years, 350 2nd and 3rd year)
* need to use Java 11
* need to use Spring Boot and Gradle

### **domain specs**

* if a room has more than 200 seats, max 30% capacity
* less than 200 -> 20%
* students must attend campus at least once every 2 weeks
* 45 min gap between each lecture
* must be able to change the room configuration (the 20% and the 30%)
* students and teachers must be able to see the classes they must attend and which class it's in
* if a teacher has corona -> lectures cancelled, put online
* if student doesn't wanna attend -> update on the system

### **housekeeping**

* Gitlab:
  + make sure the email you've got on github is your TU Delft email (sometimes there's issues with permissions otherwise)
  + merges should be done through merge requests.
    - add comments(aim for minimum of 3-4).
  + use gitlab Issues boards to keep requirements organized
* starting next week will check if at least 5 commits (merge excluded)
* all need to merge using merge requests
* they must be reviewed by at least two people
  + they must write a detailed review
  + if the merge request is good, say a few points that you like, ask a question about choices
* make sure everyone has equal amounts of work
* in the meetings, everyone should be participating, offering opinions and asking questions, nobody should be silent during the whole meeting
* final thing: about setting up the repo: you've been granted access, so set it up with the template from Brightspace, make the Read-me page (short description about the project and present yourself)
* send to Ayush the list of requirements on Mattermost

q's:

logs?

time start end?

user types:

|  |  |
| --- | --- |
| Teacher | Student |
| Can cancel lecture by reporting sick with corona.  • Lecture should be moved online or cancelled | Should be able to change their attendance to lectures.  • Should update the number of students attending. |
|  | Will get notified if the lecture is cancelled. |
| would see their time slots, rooms when requesting their schedule. | would see their on campus time slots, rooms when requesting their schedule. |
|  | On campus at least once every 2 weeks. |

# Meeting 2: *18 November 2020*

feedback for requirements:

up to us whether we implement load balancing

we can change the details of the requirements document as we work

we need to include how the teacher will add lectures to the time table in the must haves

split up the requirements more, use bullet points to explain broader requirements in detail

we can populate the database with made up data

admin role for updating corona rules is optional

**Database questions**

* Do we need a classroom name, faculty, location etc?
* Lecture name, subject?
* Multiple teachers for a lecture?
* Is a lecture certainly moved online when it’s cancelled?
* Adding a course for lecture?
* Storing the rules?