* Diversity and inclusion = divide the tasks equally; something happened to another group and we can contact the TA privately
* He asked for a small demo of features
* Testing = unit testing and the other ones that are not included in the third assignment should be done ASAP
* Progress = other groups have code and had things set up last week and have progressed in terms communication
* He is looking only at the Main branch on Wednesday, so now we don’t have any progress. We need to pick up the pace now
* We need a really good demo next week and visible communication between microservices - both functionality within the microservice itself and successful integration with other ones through requests.
* Looked at the other branches - only one microservice done
* Design patterns need to be done by Friday at 18:00
* Seems that we break up into small teams and work separately, but that is not the goal of the course
* Grade is partly due to organization. This week he was expecting progress
* Would be fine if we pick up the pace this week and present the work we need for next time
* The issues yesterday were not grade, but we should have fixed it
* Milestones should be closed when the sprint is finished and if we didn’t finish some move them to the next sprint
* Git Inspector is only for Main, so that is the only progress that counts
* The draft is good = our reasoning is on point;
  + The UML diagram need the components and system tags (notation) - compared to other groups we are doing fine on the diagram
* Always need to have the database and notation for it to the diagram (different notation)
* Do we do retrospectives? = we planned and should start doing them
* **QUESTIONS**
* Each microservice can verify a token by itself in the template = normally we shouldn’t have that - verify the token before doing any action in general
* Can the primary key of faculty be the name ? - think about persistence issues when we change the name, but we have to update it at other places as well.
* Having a request class for each microservice is not a problem, not an overlap
  + We are not going to store the info for the node we just create class that requests that information
* Token should access the node? = token of the user in order to verify that they have access - only the person that put in the node can remove it. So the same token as for verifying
* Roles in authentication - employee and sys admin, but what about faculty = faculty should be separate from sys admins - the token is the same of course, but the roles are different.
* Should sys admin have all of the faculty functionality? = up to us; easiest is hierarchical, but the other one is fine if they are three separate roles.
* For next week we agree that we will have:
  + All MUST haves = basic interactions - requests sending and creations
  + Connection between two microservices? = not explicitly stated in MOSCOW, but they should be in the MUST
  + Authentication can be tackled by multiple people, semi-independently
    - It can be integrated in the User microservice - it’s only getting information
    - We can leave the template as well, but most people merge them.
  + The week after that we will implement smaller functionalities and restrictions for methods
* We will have a meeting in-person on the 15th next week.
* MUST haves by next meeting and then we have three days for Design Patterns
  + We can integrate Design Patterns from the start and that would save us refactoring
* There is always something to improve for the assignment, so we shouldn’t write bad code in order to have a bigger gap in refactoring.
* The prototype is graded before the third assignment, so we lose points if we start out badly.