*Report*

for project

**Discipline**: Object-Oriented Programming

**Project name**: University system

**Professor**: Shamoi Pakita

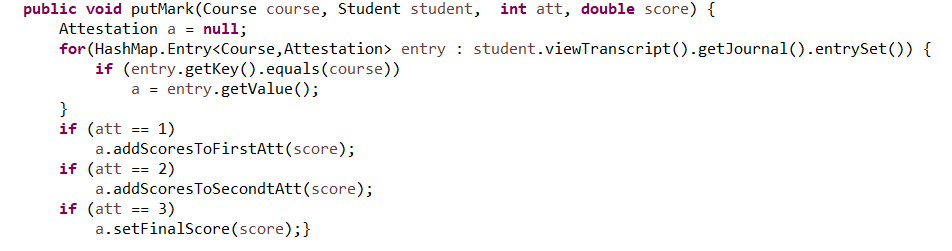
**Group**: Sailau Danagul(leader),

Saliev Dias,

Zhumabayeva Nazym

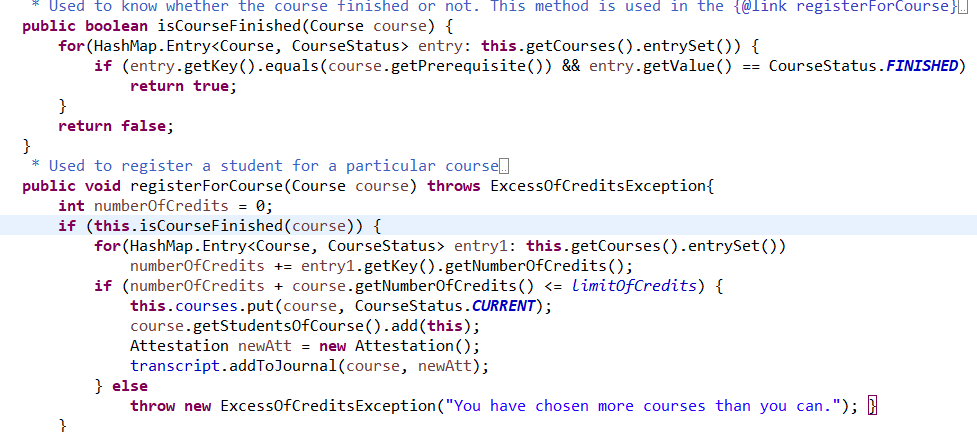
KBTU, December, 2020

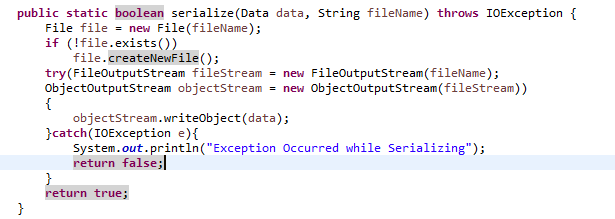
The goal of our project was to recreate a university system, for example: the WSP. We had to make actors, such as student, teacher; create methods for each of them with implementation, such as: putMark from the teacher - puts a certain grade to an individual student of some course; and build in this way the relationship between the different classes. We create different controllers for each user by abstract method userController and wrote its implementation in subclasses of User. Due to this, in login method we invoked user.userController, which is an example of polymorphism. Another example is updateInfo method, which is called in Admin to change specific fields of User.



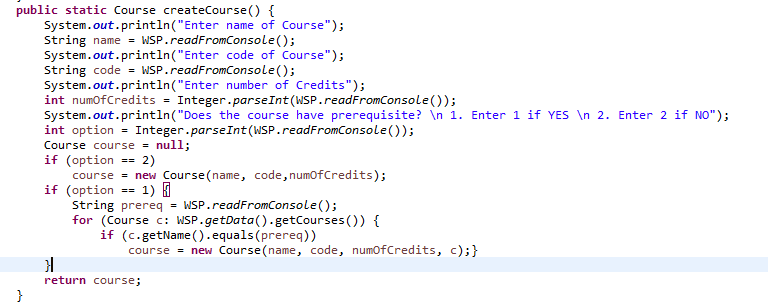
Example for polymorphism in project:

One more example for methods:



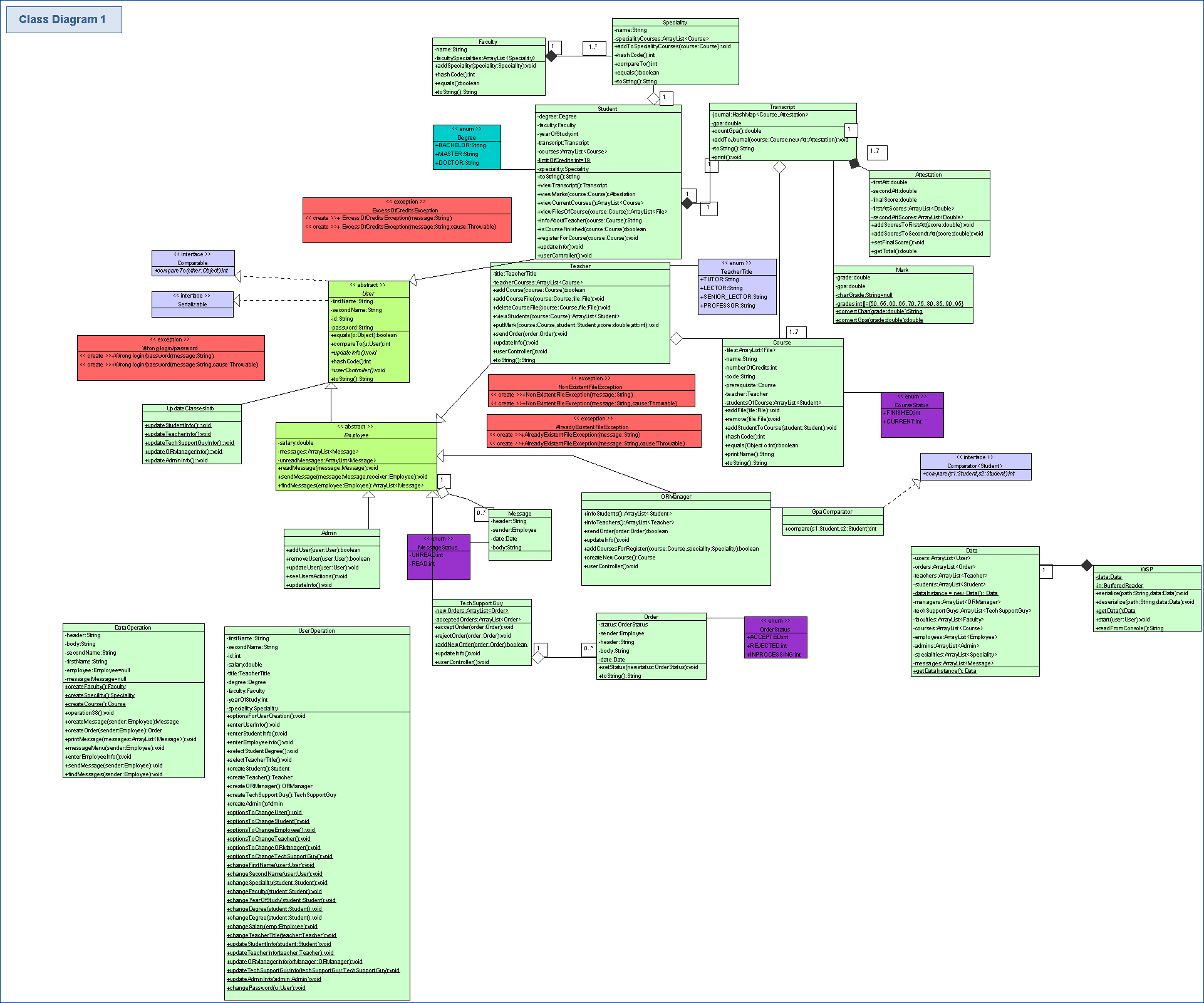
Also, we used pattern Singleton in Data class to create just one object of our class. We did this to address our created objects such as students, teachers, orders through one object data. However, we faced some problems during serialization, when we tried to invoke it with parameter data.

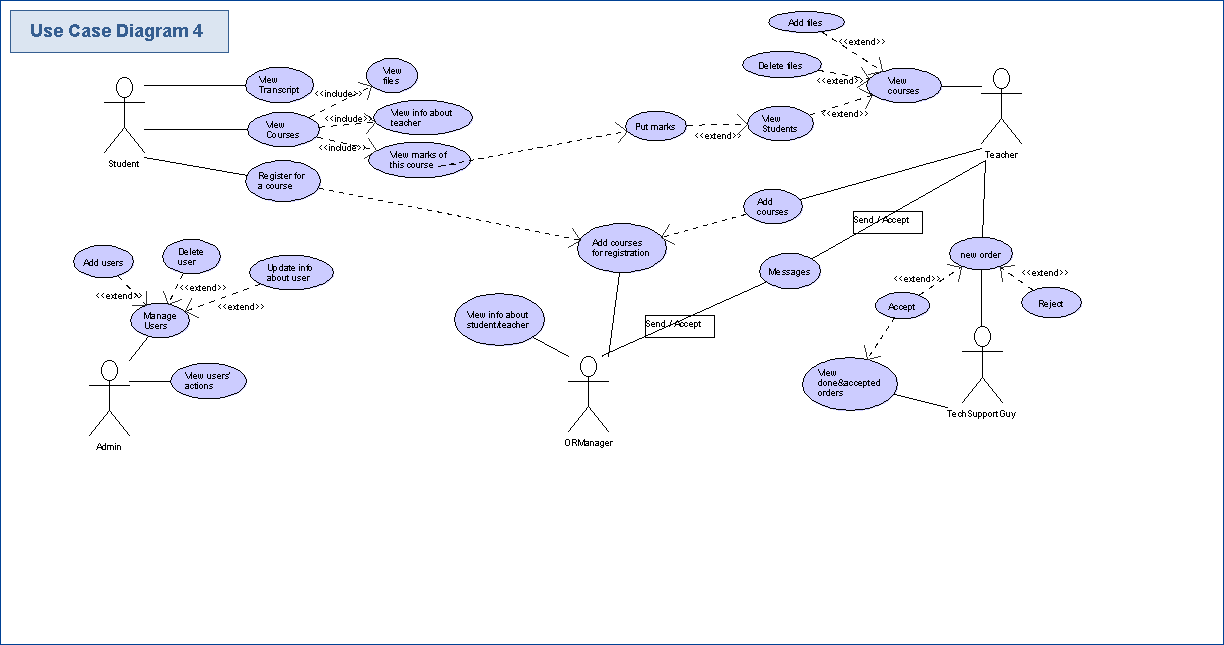
Moreover, we used Factory to create objects of serializable classes. Examples are in UserOperation and DataOperation classes.

****

Additionally we used GpaComparator to sort list of students.

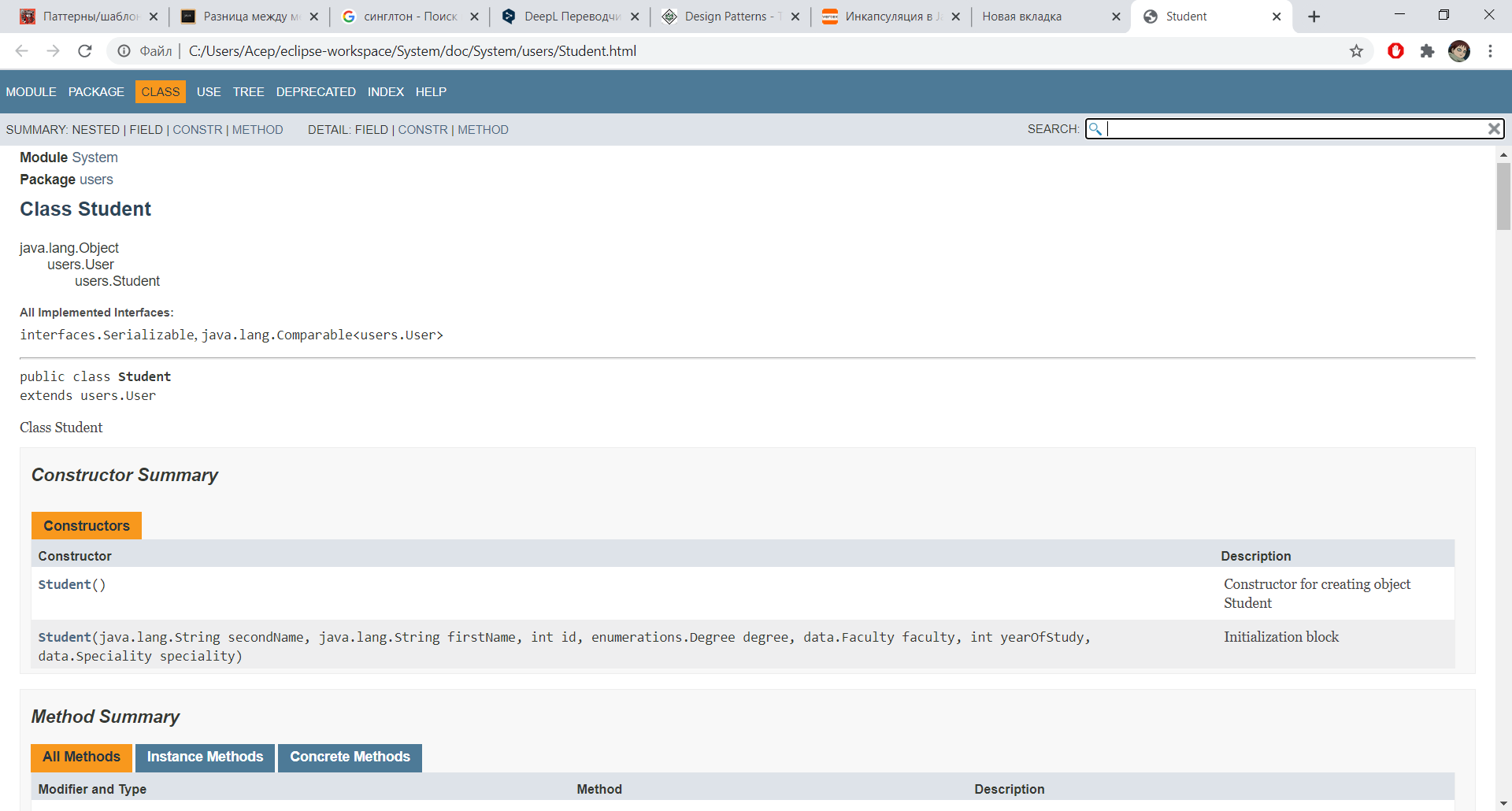
**Class and use case diagram:**

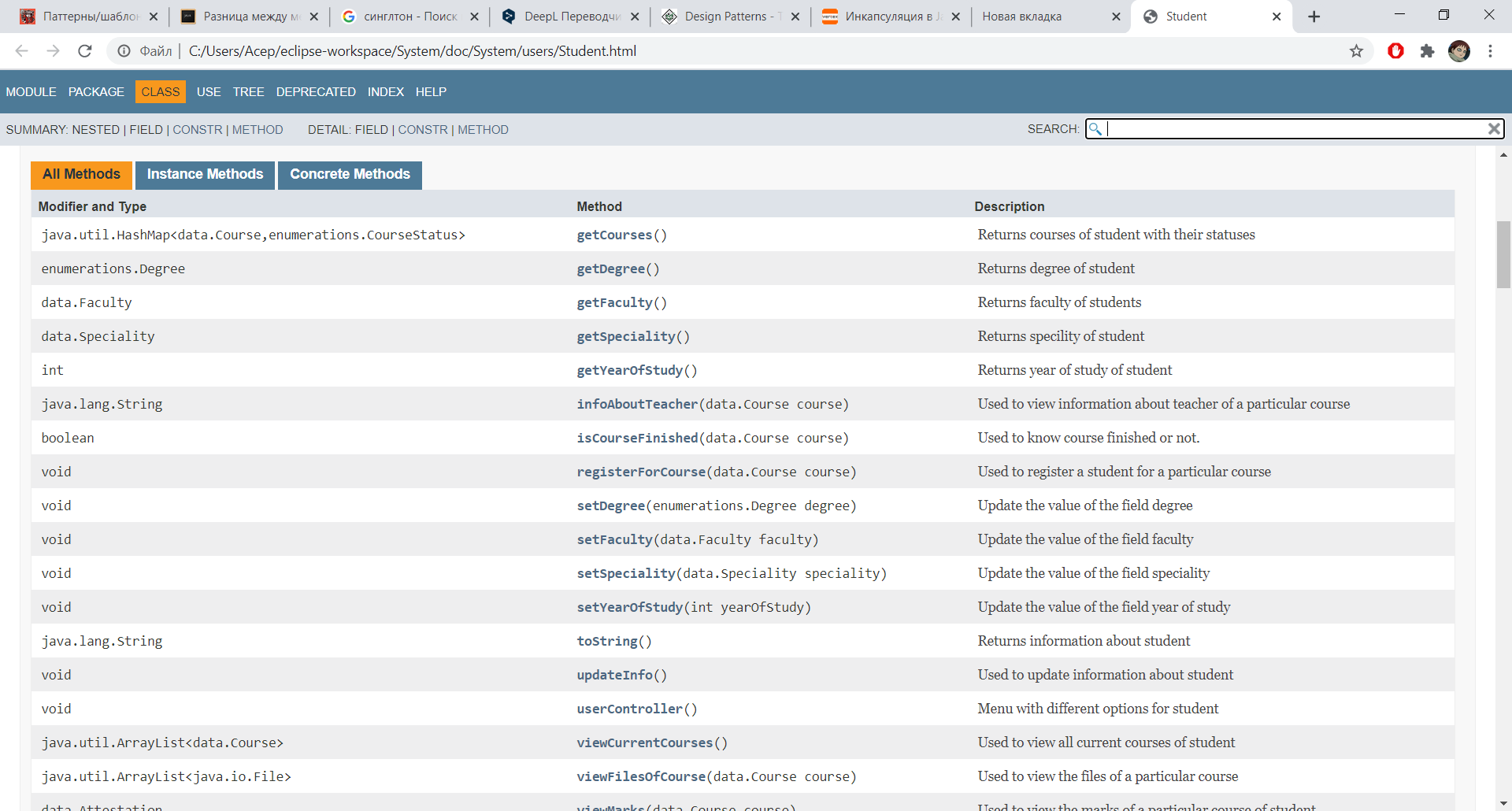


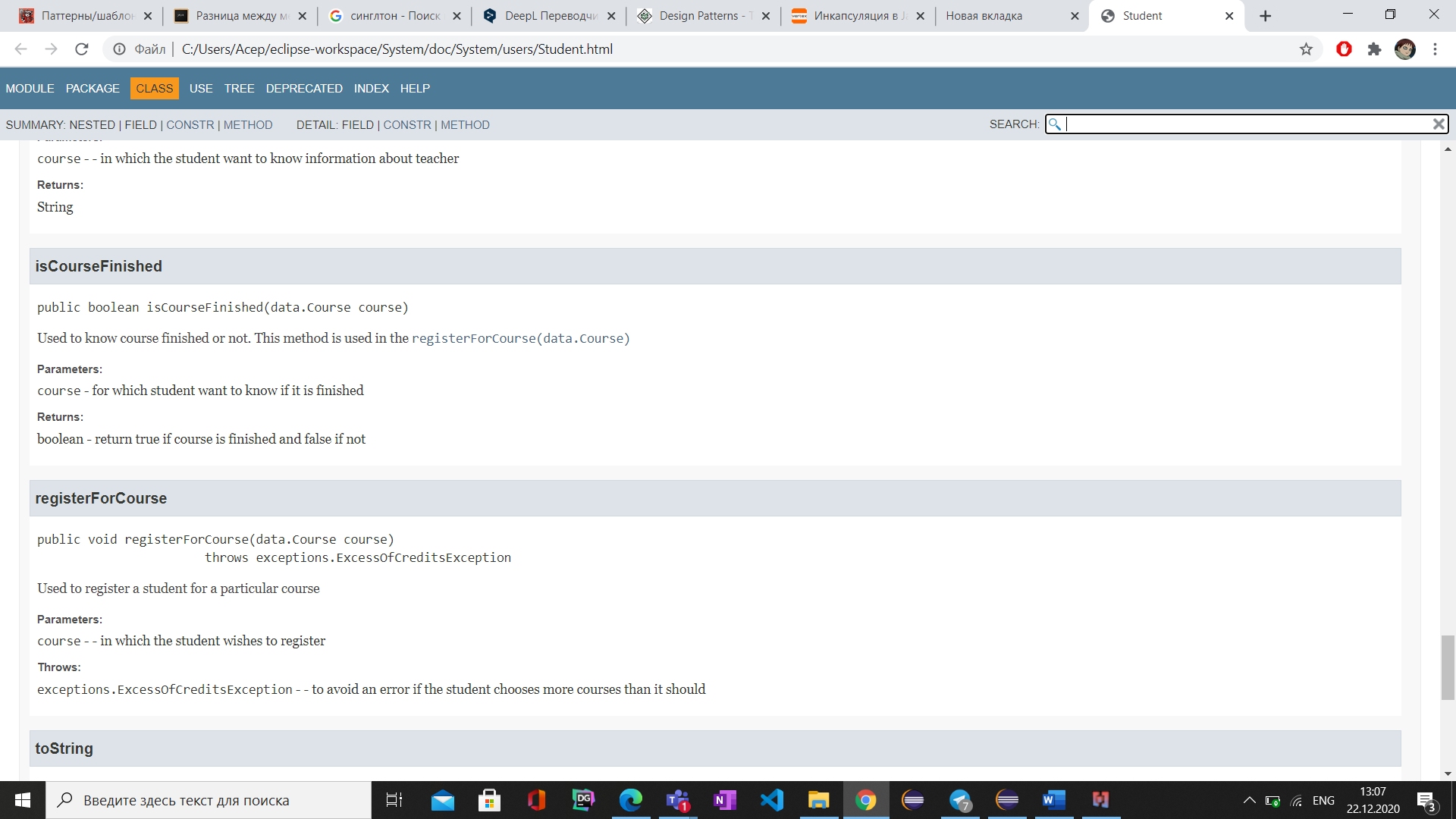


**Documentation**

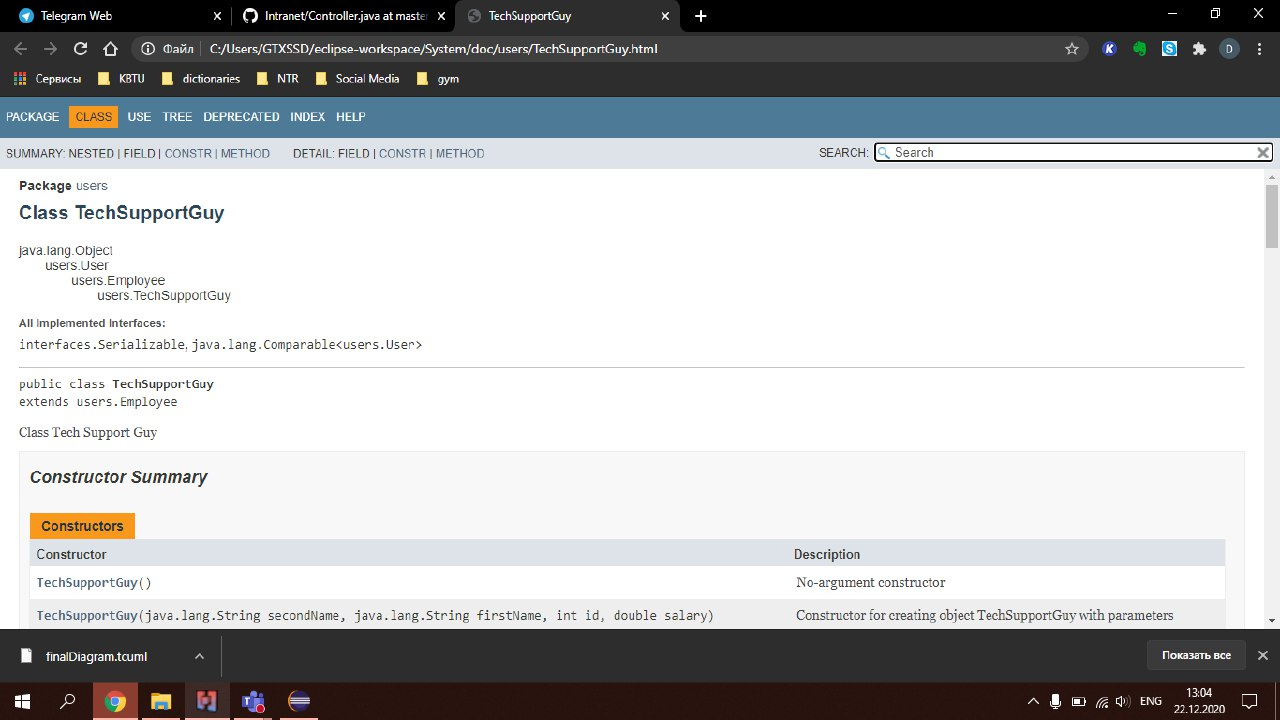
Documentation for class Student

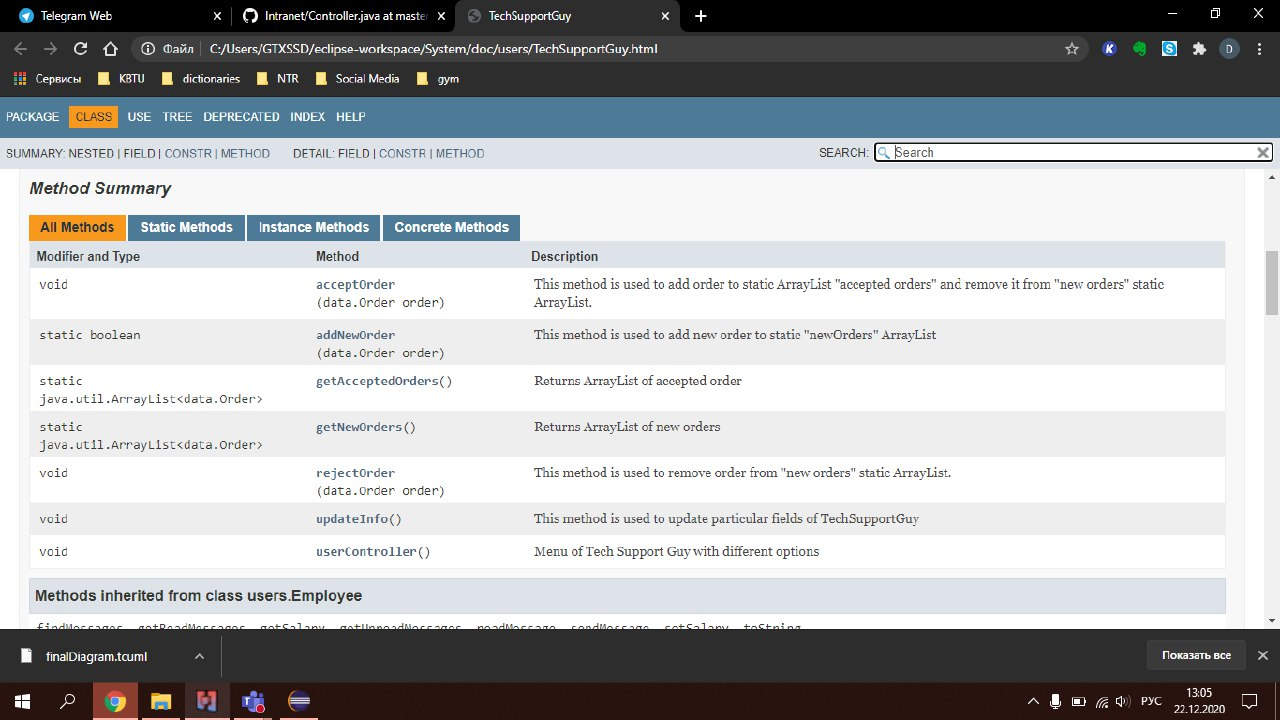


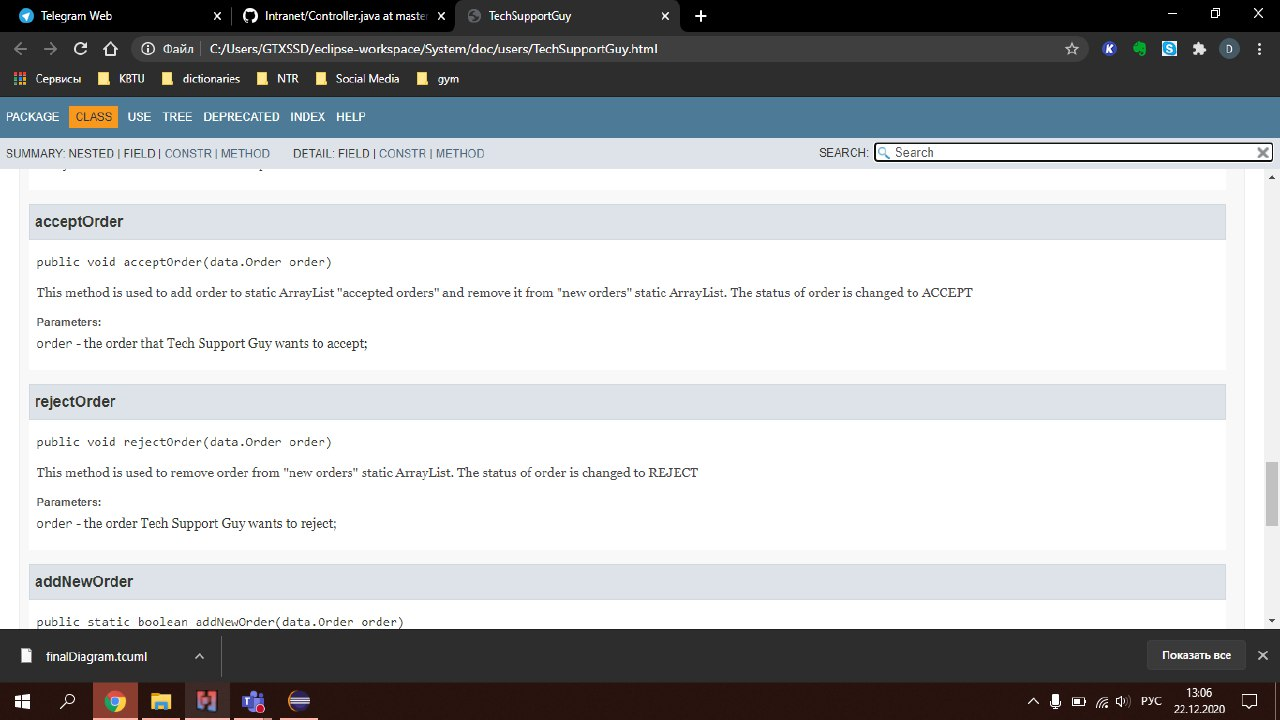




Documentation for class TechSupportGuy





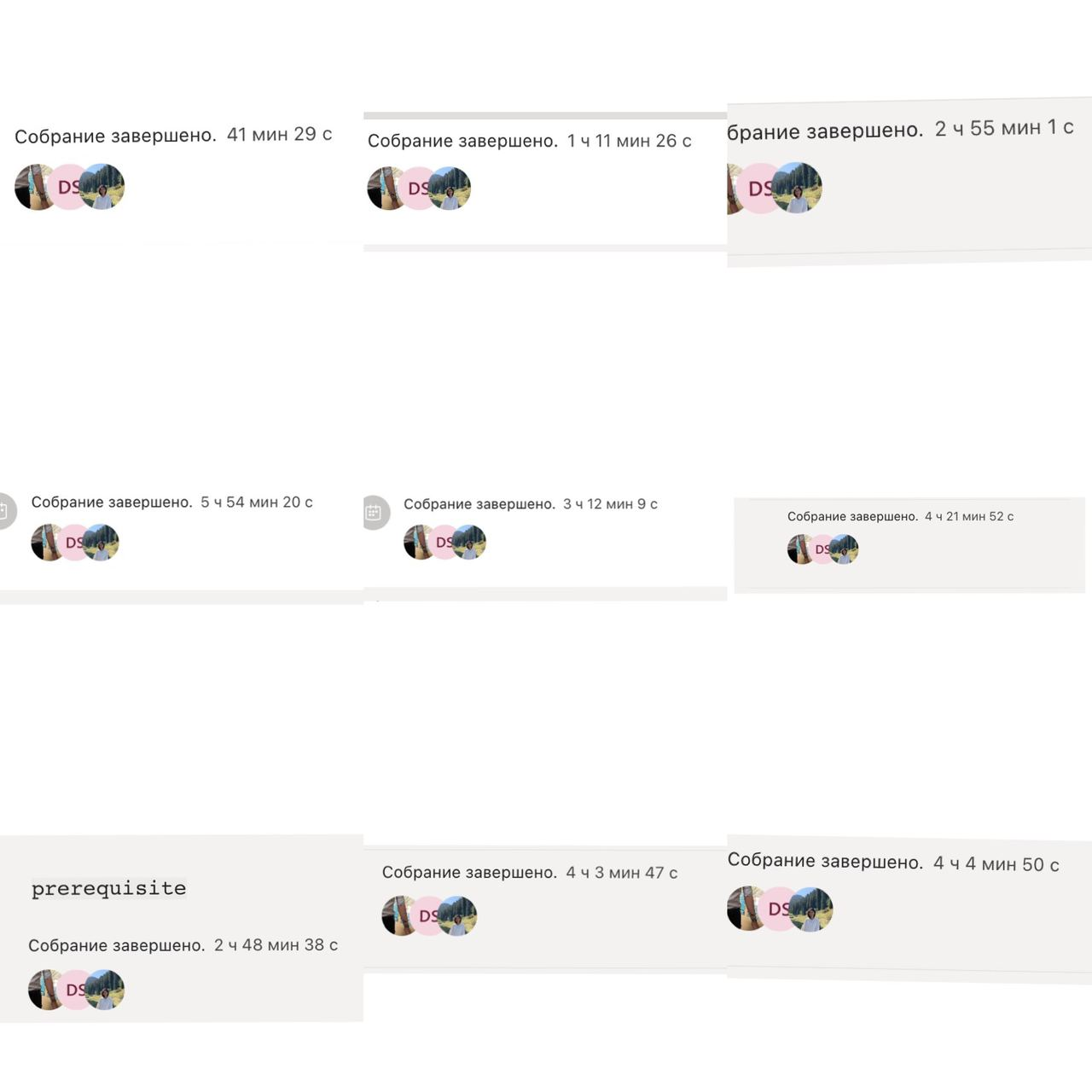


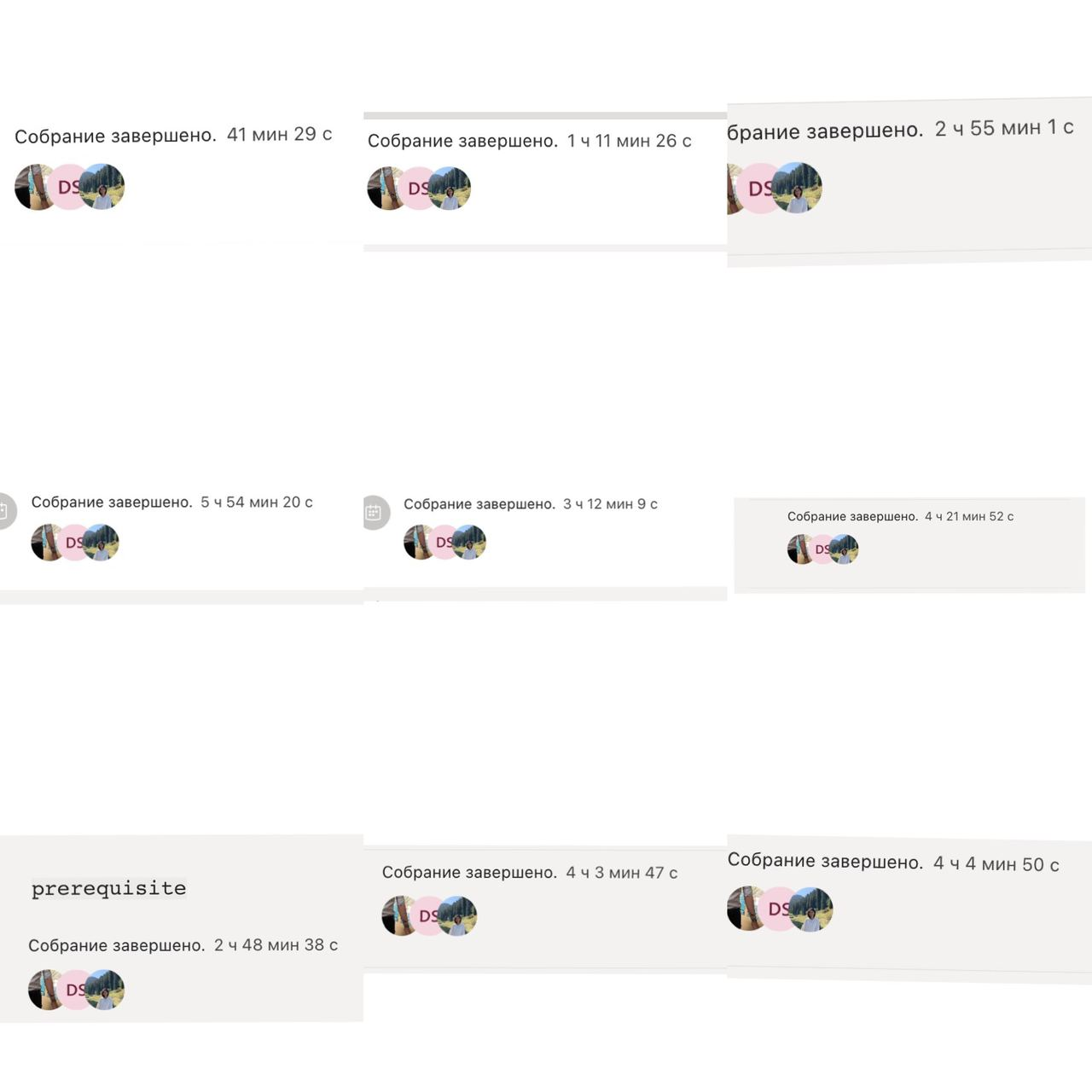
**Project management**

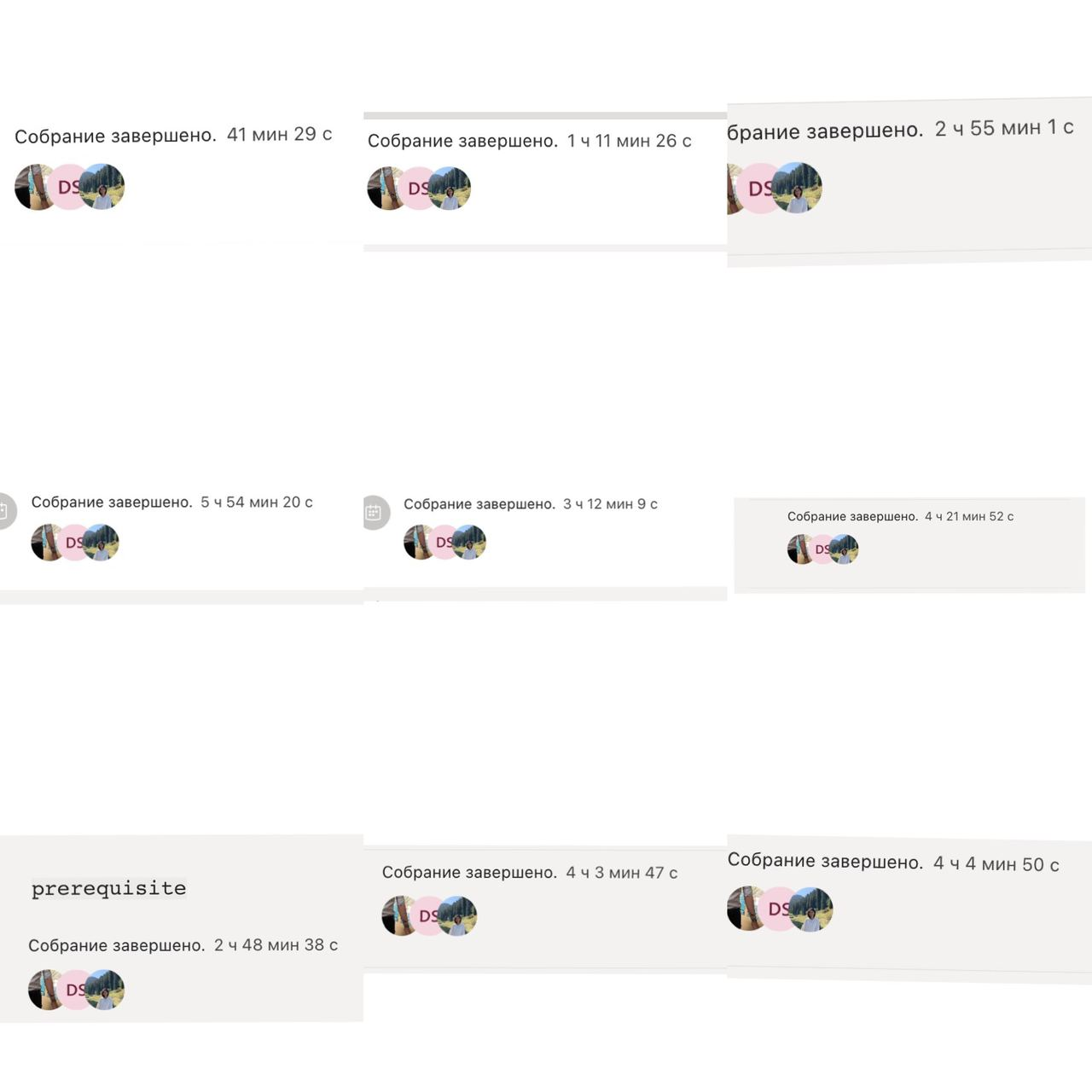
**Teamwork**

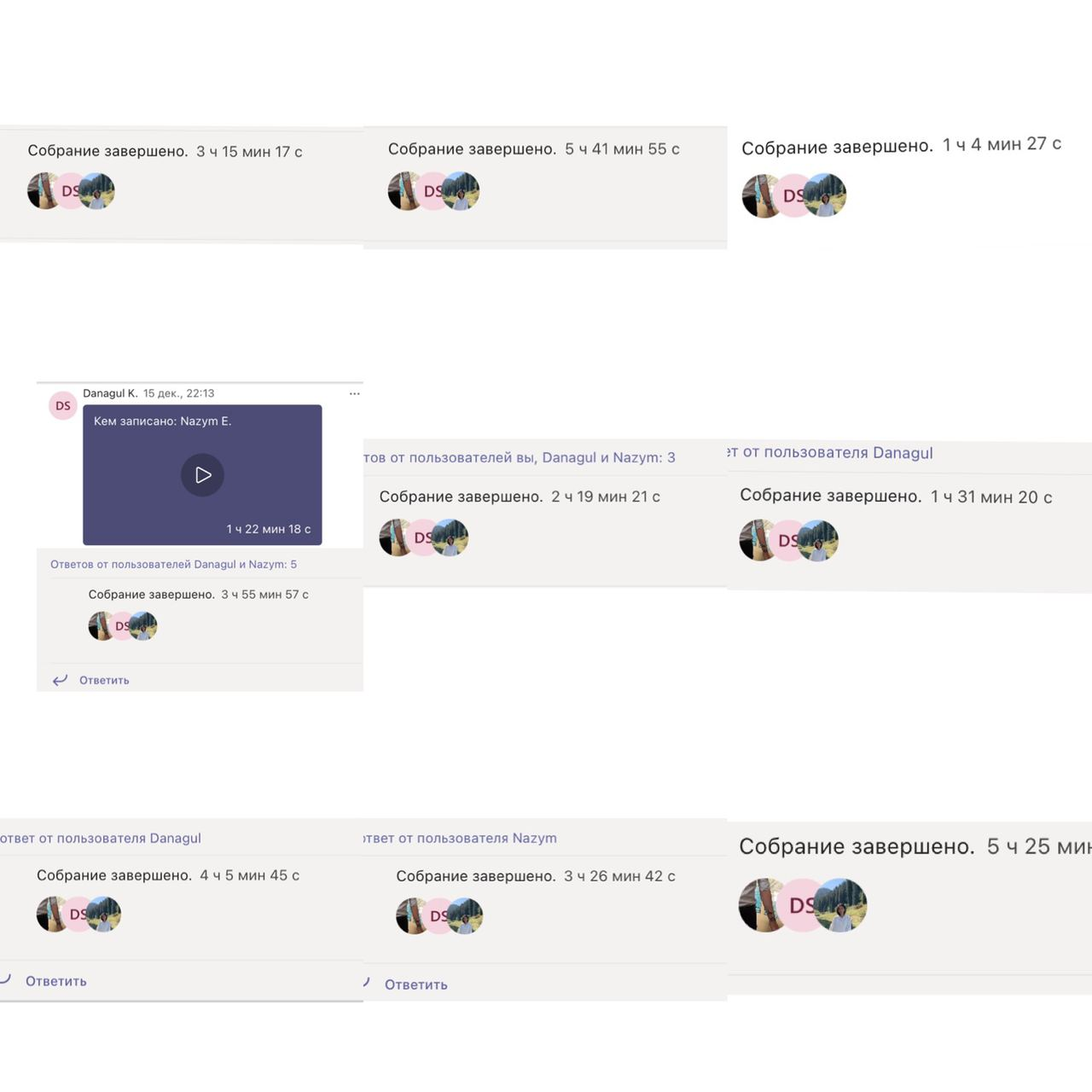
We are all from three different cities, so we had to arrange meetings in Microsoft Teams.

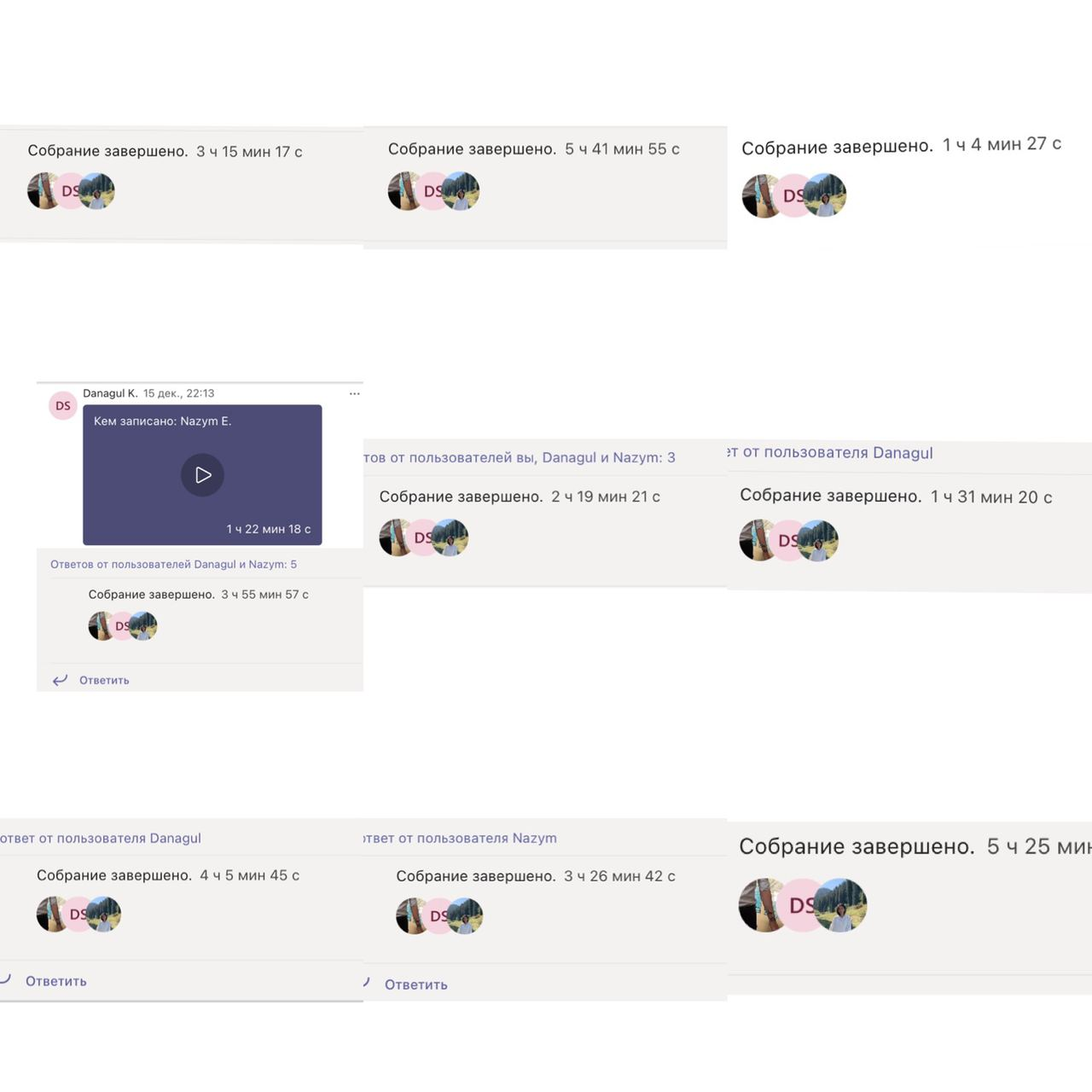
We held a total of 80 hours of meetings:

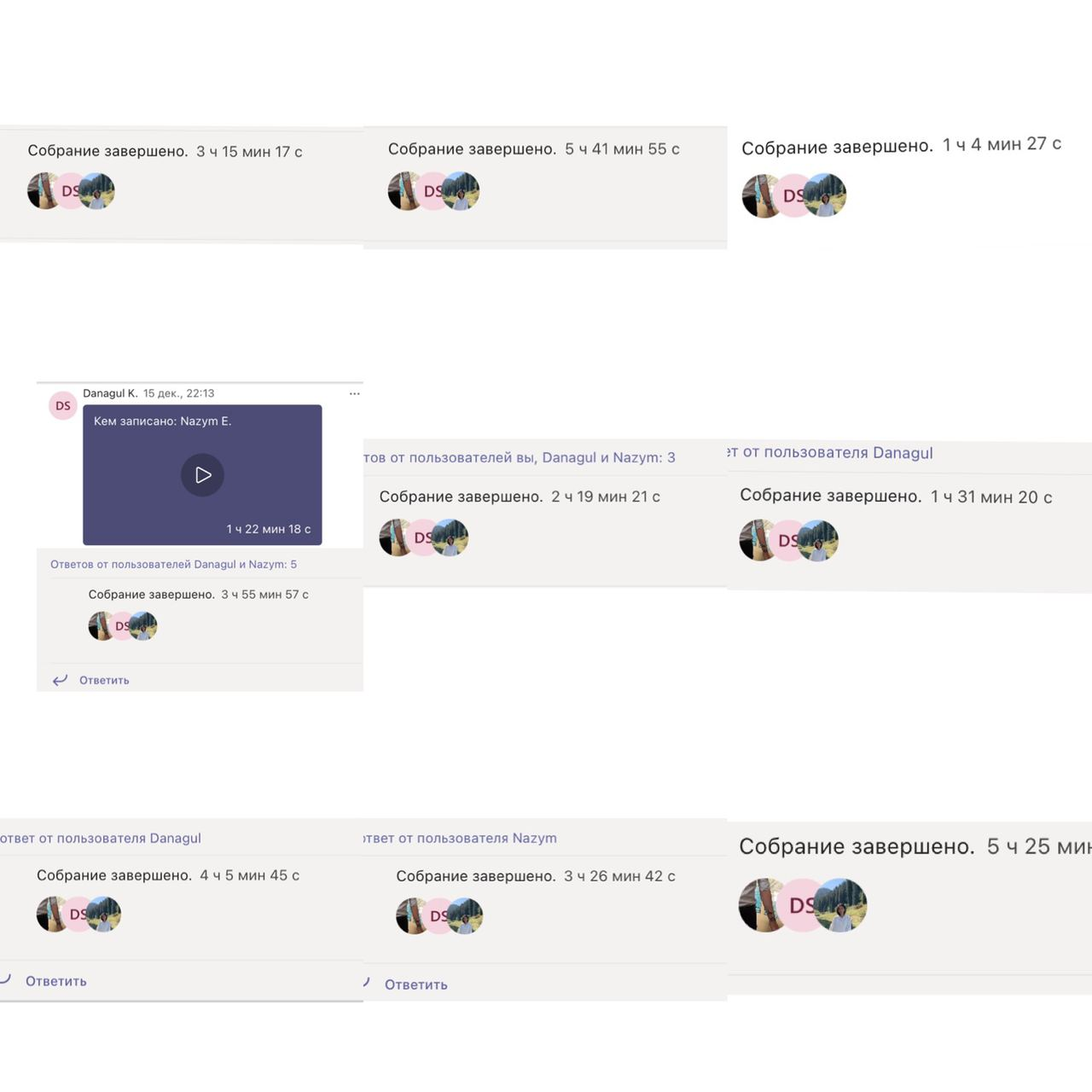


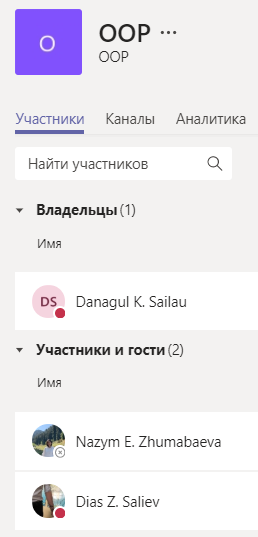












So we also delegated the work of.

Danagul create: Teacher, Course, Data, DataOperation, WSP, TeacherController

Nazym create: Student, Attestation, Mark, Transcript, Faculty, Speciality, Student and Admin controllers

Dias create: ORManager, TechSupportGuy, Employee, Admin and controllers for both, Order

Other classes, interfaces, exceptions we created together.