We’ll have 3 applications. One for the company, one for the server and one for the clients. The rules can be uploaded to the server, and the clients can connect to the server and see. The company part is a Windows Forms app. the Client is also a Windows forms app. The server is a console app or an ASP.net application. We can have a lot in for the clients (students). We could make the students per house a list of objects. The things can be changed by every member. We can have global rules and local rules. The global rules can be set by the company, and the local rules can be updated by the clients. WE can have a list of global rules on the left, and a calendar on the other side .This should be doable with a lot of labels and buttons. We can use a lot of forms. When you click a day (it is a button), you open a form with instructions for that day. We can have an add button per day, and then that will show for everyone. We then have to think about how to distinguish rooms. It might be too hard to make the calendar. It might be too much effort. If we have the global rules, each member can add and subtract from the local rules. YOu can add that and how frequently it needs to be done, and then the others need to agree. This should be done with a lot of logging in and logging out.

We all have some knowledge, but mostly self learned. Dima says he should be able to make the server. Company should not be able to see the student's individual profiles. Sometimes someone comes by and uploads and takes things from the server from the house. SO the company is not necessary according to Robert. THere will be a local server, and an employee will go to a building. Robert thinks that from what is said that someone will come by and take complaints, which are anonymous. The person that told the complaint is not identified Robert thinks that someone that gets a lot of complaints should be notified ot that .People should be able to set if others can see the complaint or not. If we have just one application, with clients and the administrator logging in on one computer. This is easier because it is only one application, without any connections. The GUI for this would have maybe a tab for the ruleset. The moderator from the company needs to go to the application, if we don’t want this we should use a server, but we can’t do this. The student houses are local networks. Robert thinks that is should be complex enough without the internet stuff. SO we should not do that. THe GUI would have a student name, mandatory rules, student made rules, which you can add or remove. THere is only one student house. We can have a schedule, and a complaint button. Dima: Do we send complaints about a person or in general? Robert: If you know you can report it, but if you don’t the others will know. That will be for the students. The schedule should be something like a calendar, and we can make that like the fhict app. Every student should have a shared application, and then also a shared database. In this database we should have a name, student number etc. Encryption is too far, so we can use the same password for everything. A user can log in on the thing, and then choose a room with a password. Someone will need a personal password too. WE could have another application for the admin or the employee. On the left of the GUI we can have a button that would let you enter a room. When we present, we should show the removing as the last person. We can store the password locally. We should have a local server according to Dima. An employee will be able to retrieve the info. We need to save things in a json or txt file. Reading and writing a file should be too difficult for this project. Dima will explain everything if someone doesn’t understand. Dima thinks we can make a perfect version and then simplify it. This will take a lot of time because of us having a website and databases. The application should work perfectly, thus we shouldn’t go a lot further than we have right now.

If some students have submitted some complaints, and the employee comes, he can see the complaints. We aren’t explained how connections work, so we should not use that. Dima says that we should be ambitious, but Robert says we should not do too much.

In each building we have a local server. It will just redirect things. We have two files, one for rules and one for complaints. The employee can connect to the server, and he can update rules. Robert and Colin agree that a simple program on one computer should be more in scope of what we can do. Robert thinks we don’t have enough time to make a very big application. Robert thinks a local server should be kept as a nice addition but not an implementation. Colin agrees, but Dima thinks it should be an implementation. Robert says it is not the idea that you make a very difficult application, but to work in a team is the biggest thing that they are going to test. Dima proposes that we can make the application and he will make the local server. Servers are trickier, and files are already something new. Dima explains that json reading is quite simple. Robert and Colin say that we should have something that everyone knows.

We stick with one application, and a different application for the admin and one for the client, and it changes on the login credentials. The login screen should be one form. So we have three forms, one for login, one for the client and one for the employee. If we have enough time we can make some local servers. What we have now is the prototype.

We should push a folder before we try to do something with the coding.

Robert says there is no use in using Gitlab we are going to work on different files.

We will start coding next week (after the assessment friday).

Problems:

* Employee has to go to every building
* Students have to meet with the employee
* Appointed people do not clean the shared responsibilities
* Common groceries aren’t done or paid
* Garbage disposal is not done on time
* Unannounced parties
* Employee can’t keep up with the local rules because of the local rules in many houses
* Will the problems be solved

TODO:

Make project Analysis

GIT Repository

Present our found problems

Process minutes