Assignment 7

After a weekend conference the new CEO has heard that projects can be done on the web, and you can get more users that way. So now we need to move the project to the web. There are a bunch of new hires, but the deadline has not changed. It's as if these managers are under the impression that 9 women can deliver a baby in a month.

- Create a new GitHub repository with the name: UBB-SE-2024-[Posting / Groups / Messaging / Business / Games / Music / Evil] (e.g. UBB-SE-2024-Posting). Invite the client (imre.zsigmond@ubbcluj.ro) as a collaborator so they can take a look. Join with the subgroup from the list below and solve the assignment together.

  921/1 + 922/2
  921/2 + 923/2
  923/1 + 925/1
  924/1 + 926/2
  927/2 + 926/1
  925/2 + 927/1
  924/2 + 922/1

- You are one team with four team leads. Grades will be awarded for the new team at the second lab (although it is a good idea to present at least half at the first). The Four team leads will break down the grade between those that have contributed. The grade is for the entire team of 20+ students. Any student who did not contribute will be considered as if they abandoned the project. So if we have a team of 20 people who get a 5.0, but only 10 contributed, then everybody who did will get a 10.
- Migrate your solution to ASP.NET core MVC. This should only affect your GUI; the business logic, and data storage are just copied. Prioritize the initial/previous requirements by what would fit best together and what you can do in a reasonable amount of time, then implement them in that order. If you do not get to the end of the list, it's ok if the application is reasonable in size and quality.
- By default, ASP.NET core MVC comes with jQuery and Bootstrap. If you want to use any extra javascript / css framework(s) you need to get the express approval of every team, and the client (at one of the courses).
- Your desktop/mobile applications must run in parallel with your website. The data access has already been moved to the server, and your business logic should be already separated so just copy it to the server. Now you must remove the business logic from the desktop/mobile application. Change the APIs created for the last lab to give the desktop/mobile application what they need. Change the desktop/mobile applications to call the new APIs. The server will have their ASP.NET core MVC GUI, and ASP.NET core API both use the same business logic, which will in turn use the same database access.
- All functionalities need to be guarded from unauthorized users.
- Create a component diagram for your solution.

Deadline: lab 7

For the bonus:

- Team leads may hold onto their pick for "a +1 lab grade to any student" until the last lab (5th of June) to coordinate with others.

- If you are aiming for the "all projects in one bonus", then create the repo "UBB-SE-2024-Adonalsium", also invite the client. The bonus may be presented until the end of the last lab as one, or in smaller pieces during the last two weeks by each team. The joined solution must include all functionalities that were implemented for their specific lab (desktop/mobile/web) from all groups, using the same database, working on a deployed server (ask the teacher for one), with every component able to communicate with any other without a direct reference. The desktop/mobile applications do not have to be merged, just connected to the merged server.
- One or two teams can abandon their current project (no longer needs to be upgraded or merged), and focus on merge specific problems, like the shared menu, inter project communication etc. They would be graded on what they did for this aspect even if the merge ultimately doesn’t happen.