I started this project as simple as possible to don't waste to much time, was difficult to find free time to do it so I did it as quickly as possible.

I tried to follow a kind of ONION architecture with UI > Service > Dal > database, but as I told you on phone interview, I am studying to become software architecture, I am 28 years old now, I expecting to achieve it at a minimum of next 3 years. One of my weakness I would say that it is Desing Patterns. I find myself really good at solving problems but I am improving myself as a developer to know and understand more and more about design patterns.

I tried to follow this approach because for what I have been studying that It provides better maintainability as all the codes depend on layers or the center. It also provides better testability as the unit test can be created for separate layers without an effect of other modules of the application and It develops a loosely coupled application as the outer layer of the application always communicates with inner layer via interfaces.

This UI project is using Ioc as well with "Unity", it decouples your classes, as well as allowing easier unit testing because you can mock/stub the dependencies and also if your controllers have dependencies then an IOC framework allows you to easily declare these dependencies by interface on the controllers and not worry about how they get supplied.

I could have done it following different approach but as I said, I tried to save time and focused on the main objective, but I am 100% sure that I could have tried have it built following different architecture approach.

This system has a simple database with 3 entities, User, TaskList and Tasks.

DbContext is using migrations, you will need to run "update-database" pointed to DAL project in a way to have its database created, please do not forget to set connection string on web.confing (UI) and app.config (DAL). I did that using migration just because I find it easier than creating a database and using entity data model, otherwise, I would be forced to send you scripts of database or a backup file to be restored or database file to be attached and it would depend on SQL version etc.

Improvements if I had more time to do it:

- create universal method setup on tests class

 - create a history of changes on tasks

 - create functionality to attach documents on tasks (pdf,png,etc)

 - create the functionality of cloning list and tasks

 - set up logging errors and apply to try catch blocks (I know it is easy and should have done already)

 - create functionality to change the password

 - create the functionality of "forget my password"

- Create the functionality of sharing Lists and tasks