DevIncubator Management System

©2019 – DevIncubator (all rights reserved)

Member role

Actions	Member
Watch the **Member's Manage Grid**	-
Add, edit, and delete a member on **Member's Manage Grid**	-
Watch the **Member's Progress grid**	-
Watch the **Tasks Manage Grid**	-
Add, edit, and delete a **New task**	-
Watch the **Member's Task Manage grid**	+
Set the **Member task's state** as Success or Fail	-
Watch the **Subtasks Manage Grid** of the current **Task**	+
Add, edit, and delete a **Subtasks** of the current Task	+

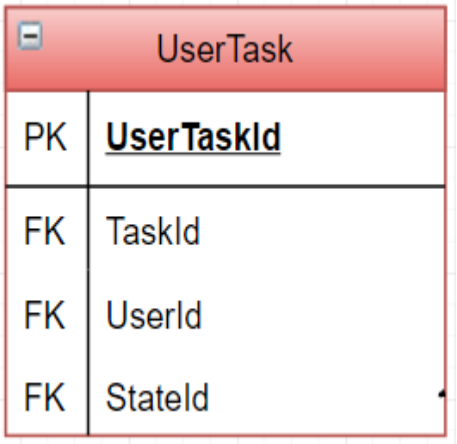
1st week:

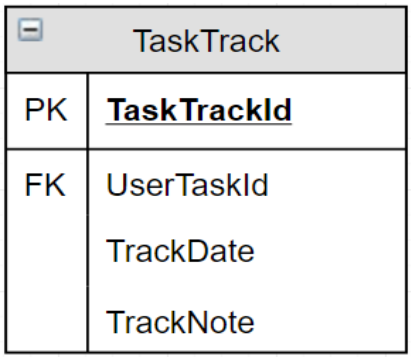
Death valley rewriting. You should rewrite you first project following n-layer architecture pattern: database layer, business layer, presentational layer. Here is detailed plan about what you should do. Additionally you should write unit tests to repositories and controllers. It will be super if you add also 2-3 integration tests.

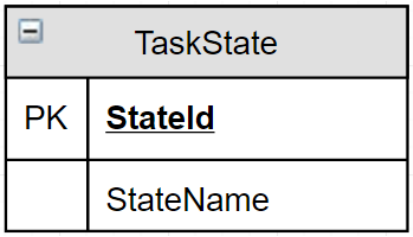
2nd week:

You should implement **UserTask, TaskTrack, TaskState** tables**, vUserTask** view with all needed constrains (remember! you must use only the table level constraints not the column one).

Use the **Task** table as base table. To get **UserId** and **State** join the **Task** table with **UserTask** table and **TaskState** table. As you can see, you have a lot of dependencies of your partners. If they don't implement all needed tables, please push them!







TaskState is a typical constant values table. You need to add three constant values to the **SeedData** which will be inserted all time in the publish process. They are **Active**, **Success**, **Fail.**

And you need to create the **SetUserTaskAsSuccess** and the **SetUserTaskAsFail** procedures, which take the **UserId** and **TaskId** as input parameters and change the **StateId** in the **UserTask** table to **Success** or **Fail** respectively.

3rd week:

Learn how to write unit-tests properly and write unit-tests for you repositories. Read about different approaches, **rule of "AAA"**, **right naming** and **code convention** relating unit testing.

Look at the following tools.
For test coverage of application we will use **dotCover**. Another popular libraries:

- **OpenCover**
- **ncrunch**
- **etc**

For unit tests writing we will use **NUnit**. Another popular libraries:

- **xUnit**
- **MSTest**

For fake/mock/stub we will use **moq**. Another popular libraries:

- **nsubstitute**
- **fakeiteasy**
- **etc**

4th week:

Members' Tasks Manage Grid page a simple grid with typical CRUD actions. You need to implement the UserTask Data Transfer Object.

Hi, dear Ivan! This is your current tasks:

#	Name	Start	Deadline	Status		(Available only for Admin)
1	Create the DB	06.12.2017	12.12.2017	Sucess	Track	Success Fail
2	Implement the procs	13.12.2017	20.12.2017	Active	Track	Success Fail

5th weeks:

Implement controllers with all needed actions, ViewModels, and Views. Look at existing SampleController, and do the same. The Create/Edit pages look similar, but Detail page needs to be as a read-only version. The **Track** is the link to **Task Tracks Manage(Subtasks) grid** page. If this page hasn't been implemented yet simple create this button without particular path.When your partner make their piece of work, you will add all needed links to this button.

6th week:

Complete your presentation layer. add nice styles. If you will work with front-end partner, provide him all routes and data that he can get from them accordingly.

7th week:

Put all parts together, add all logic related you main role. Write unit-tests for you controllers, business logic and 2 integration tests.

8th week:

Test your area of responsibility very intensively and fix all bugs. After this week you should provide a demo with description of business logic of your part in English.