

# **Тестовое (тренировочное) задание**

## **Управление задачами**

Version 1.2 (ASP NET Core)

## Содержание

<b>ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ .....</b>	<b>3</b>
Общее описание процесса .....	3
Функциональные требования.....	3
Требования к интерфейсу пользователя .....	4
<b>СРЕДА ВЫПОЛНЕНИЯ, ИНСТРУМЕНТЫ, ТЕХНОЛОГИИ .....</b>	<b>4</b>
База данных.....	4
Приложение .....	4
<b>ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ .....</b>	<b>4</b>
Основные .....	4
Дополнительные.....	5

## Требования к приложению

Необходимо разработать систему управления задачами. Система должна позволять редактировать иерархию задач, а также проводить некоторые вычисления по их полям данных.

### Общее описание процесса

Стандартная схема работы с системой выглядит следующим образом:

1. Задачи заносятся в систему (создаются, редактируются, удаляются).
2. К любой задаче могут быть добавлены подзадачи.
3. Удалена, может быть только терминальная задача. Удаление поддерева не требуется.
4. Структура задачи и подзадачи одинакова. Подзадача не может принадлежать более чем одной задаче. Количество уровней подзадач не ограничено.
5. Поля «Плановая трудоёмкость задачи» и «Фактическое время выполнения» вычисляемые и складываются из сумм подзадач, входящих в данную задачу, и самой задачи (при отображении задачи содержащей подзадачи время выполнения подзадач необходимо показывать отдельно).

Каждая задача, отслеживаемая системой, характеризуется следующим набором атрибутов:

- Наименование задачи
- Описание задачи
- Список исполнителей (простое текстовое поле, ссылочность не нужна)
- Дата регистрации задачи в системе
- Статус задачи: Назначена, Выполняется, Приостановлена, Завершена (число статусов для системы фиксировано и сами статусы неизменны).
- Плановая трудоёмкость задачи
- Фактическое время выполнения
- Дата завершения

### Функциональные требования

Идентификатор	Описание требования
FR001	Система должна позволять создавать задачи и подзадачи.
FR002	Просмотр и редактирование всех полей задачи (кроме вычисляемых)
FR003	Удаление задач и подзадач
FR004	Просмотр списка задач
FR005	Необходимо предусмотреть следующие ограничения при переходе между статусами задач: <ol style="list-style-type: none"> <li>1. При завершении задачи, все её подзадачи также должны быть переведены в статус «Завершена» (если какая либо из задач в дереве не может быть «завершена» ни одна из задач дерева не должна поменять статус).</li> <li>2. Статус «Завершена» может быть присвоен только после</li> </ol>

	<p>статуса «Выполняется».</p> <p>3. Статус «Приостановлена» может быть присвоен только после статуса «Выполняется».</p>
--	---

### Требования к интерфейсу пользователя

Идентификатор	Описание требования
UI001	Иерархия задач должна быть представлена в виде дерева
UI002	Список всех подзадач выбранной задачи должен выводиться в виде плоского списка
UI003	Интерфейс должен быть организован по типу “Проводника Windows” (слева дерево задач, справа описание выбранной задачи/подзадачи)
UI004	При переключении между задачами страница не должна перезагружаться полностью (должны использоваться AJAX запросы)

Больше ограничений на интерфейс пользователя нет. Внешний вид приложений, состав форм и граф перехода реализуется кандидатом в соответствии с его представлением о задаче.

## Среда выполнения, инструменты, технологии

### База данных

Все данные приложения должны быть размещены в БД MS SQL Server. Среда разработки: MS Visual Studio 2017.

### Приложение

Приложение должно быть реализовано с использованием ASP.NET Core и Razor.

В качестве языка реализации следует выбрать C#.

Среда выполнения: .NET Core 2.x

## Требования к реализации

### Основные

1. Необходимо использовать объектно-ориентированные подходы при построении системы. Ориентироваться на паттерн MVC.
2. Необходимо придерживаться единой схемы обработки исключений (не использовать коды возврата, публиковать исключения только в контроллерах, использовать отдельные типы исключений для описания различных исключительных ситуаций)
3. Использовать ресурсы для хранения UI-строк

4. Именованые типов и переменных должно быть выдержано в единой схеме (желательно в рекомендованной Microsoft для .Net приложений, но другие также допустимы).
5. UI должен быть построен с учётом интересов пользователя. В создании UI руководствоваться требованиями руководством Microsoft по дизайну UI. Пользоваться им должно быть удобно (а лучше и интуитивно понятно).

**Дополнительные**

1. Использование многослойной архитектуры с отдельным слоем доступа данных и бизнес логики будет дополнительным плюсом.
2. Использование Sass/Less и TypeScript будет дополнительным плюсом.