Komandos „TELPA“ projekto techninė ataskaita

Projekto išeities tekstai:  
<https://github.com/PauliusMilmantas/TELPA>

Pasirinkta technologinė platforma: .NET Core

Sistemos struktūra sudaryta iš 3 architektūrinių sluoksnių – vartotojo sąsajos, dalykinio funkcionalumo ir duomenų prieigos.

Vartotojo sąsaja ir visas dalykinis funkcionalumas realizuoti naudojant Angular Web karkasą.

Duomenų prieigos sluoksnis realizuotas naudojant server-side ASP.NET MVC karkasą.

Kokybinių reikalavimų realizacijų pavyzdžiai:

1. Concurrency:
   1. <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Controllers/SessionController.cs>  
      TELPA\TELPA\TELPA\Controllers\SessionController.cs  
      (38 eilutė – „Me“ metodas)
   2. <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Controllers/EmployeeController.cs>  
      TELPA\TELPA\TELPA\Controllers\EmployeeController.cs  
      (400 eilutė – „UpdateEmployee“ metodas)
   3. <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Startup.cs>  
      TELPA\TELPA\TELPA\Startup.cs  
      (182 eilutė – „Configure“ metodas)
2. Security:  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Controllers/EmployeeController.cs>  
   TELPA\TELPA\TELPA\Controllers\EmployeeController.cs  
   (213 eilutė – „getEmployeesAndLeadersForSupremeLeader“ metodas)
3. Data Access:  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Controllers/EmployeeController.cs>

TELPA\TELPA\TELPA\Controllers\EmployeeController.cs

(383 eilutė – „CreateEmployee“ metodas)

1. Data consistency; Optimistic locking:

<https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/ClientApp/src/app/topic-edit/topic-edit.component.ts>

TELPA \TELPATELPAClientApp\src\app\topic-edit\topic-edit.component.ts

(175 eilutė – „updateTopic()“ metodas ir 232 eilutė – „updateTopicLink()“ metodas)

1. Memory management:  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Components/SessionService.cs>  
   TELPA\TELPA\TELPA\Components\SessionService.cs  
   (94 eilutė – „Dispose“ metodas)
2. Reactive programming; Asynchronous/non-blocking communication:  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/ClientApp/src/app/calendar/calendar.component.ts>  
   TELPA\TELPA\TELPA\ClientApp\src\app\calendar\calendar.component.ts  
   (57 eilutė – „getBackendData“ metodas)
3. Cross-cutting functionality/Interceptors:  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Controllers/EmployeeController.cs>  
   TELPA\TELPA\TELPA\Controllers\EmployeeController.cs  
   (397 eilutė – „Logged“ atributas)  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA.Extensions.Logic/Components/FileLoggerService.cs>  
   \TELPA\TELPA.Extensions.Logic\Components\FileLoggerService.cs  
   (21 eilutė – „OnActionExecuted“ metodas)  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/appsettings.json>  
   TELPA\TELPA\appsettings.json  
   (45 eilutė – „Interface“ savybė)
4. Extensibility/Glass-box extensibility:  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/Startup.cs>  
   TELPA\TELPA\TELPA\Startup.cs  
   (63 - 137 eilutės – „ConfigureServices“ metodas)  
   <https://github.com/PauliusMilmantas/TELPA/blob/master/TELPA/TELPA/appsettings.json>  
   TELPA\TELPA\appsettings.json  
   (28 eilutė – „Services“ savybė)