Schedule Manager Application

Laborator Inginerie Software

**Project description**

The application is meant for employers as a tool for managing and being more aware of their employees schedule.

Use Cases

Register Use Case

* An unregistered user, is able to make an account in order to use the application.
* A user can have two roles, a normal user, with limited privileges, and a SMA (Schedule Manager Admin).

Login Use Case

* If a user already has an account, the user will be able to directly login into the application. After that, he can either use the available features or log out.

Logout Use Case

* A user that has logged in, can easily logout. Also, there is a session timeout implemented, if a user stays too much without doing anything on a page, the session will expire and the user will need to log in again.

Create Business Trip Use Case

* An authorized user, is able to create a business trip, which will be later approved by an administrator.

View Business Trips Use Case

* An authorized user, is able to see the business trips that he created, and is able to cancel them, this will trigger a notification to the administratior.

O imagine care conține diagramă, text, cerc, linie

Descriere generată automat

Architecture

The application uses a domain driven design approach (DDD). This is an architectural pattern in which the developers try to model the software to match the domain/business as closely as possible. Our application is divided in a few projects:

* ScheduleManager.Web (The Web project)
  + It serves as the interface, handling all client interactions.
  + It provides a decoupled frontend, that can be easy to work with, especially when using Mediator Design Pattern.
* ScheduleManager.Application
  + This contains the application logic and manages the flow of data between the domain layer and the user interface
  + Makes sure that the presentation layer is as decoupled as possible from the business layer.
* ScheduleManager.Domain
  + It contains the business logic, mostly the needed Interfaces and necessary entity models.
* ScheduleManager.DataAccess
  + Manages all operations regarding data or any operations that is interacting with the database.
  + It is a very good practice to have a separate layer dedicated only to data access.
* ScheduleManager.Notification
  + Handles the delivery of notifications, for this application, it uses emails.

Flow charts:

Register flow:

O imagine care conține diagramă, text, Plan, captură de ecran

Descriere generată automat

O imagine care conține text, cerc, captură de ecran, diagramă

Descriere generată automat

Login flow:

* Given the use of the Mediator design pattern and the domain driven design, the flow charts are quite similar, in the sense that each request is handled by the mediator.

O imagine care conține text, diagramă, captură de ecran, Plan

Descriere generată automat

ViewBusinessTripsForUser flow

O imagine care conține diagramă, text, Plan, Dreptunghi

Descriere generată automat

Techonlogies used:

For developing the application I’ve used ASP.NET Razor Pages. It is a web development framework that simplifies the creation of web pages. It uses a multi-page approach. The main advantage is that it uses the Razor syntax for integrating C# with HTML, in ‘cshtml’ files. It can also be integrated with JavaScript for som more complicated functionalities.

I have used it together with Microsoft Identity framework, it provides a good way to handle authentication and identity management. In this way, we are never going to store the actual password for a user in the database, for example.

Downsides I’ve found:

Given the strict rules of the architecture and the design pattern, it can increase in complexity very quickly. I also found that, to follow strictly the design pattern chosen, we have to write a lot of code and new files, for example, each action, even the Logout one, needs a class for a request and for a handler. Also, for a model in the domain layer. Examples:

O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat

O imagine care conține text, captură de ecran

Descriere generată automat