Structurio – Dokumentation

Lastenheft

Ziel des Projekts:
Structurio soll ein einfaches, intuitives Projektmanagement-Tool sein, das die Planung, Organisation und Nachverfolgung von Aufgaben in Projekten erleichtert.
Zusätzlich soll die Software dem Benutzer ermöglichen, den Überblick über das Budget und die eigenen Arbeitszeiten zu behalten.

Zielgruppe:
Schüler:innen, Studierende und kleine Teams, die eine unkomplizierte Lösung für das Verwalten von Projekten und Aufgaben suchen.

Anforderungen:

- Übersichtliche Darstellung von Projekten und Aufgaben, sowie Finanzen und Zeiterfassung.
- **Kernfunktion:** Aufgaben und Aufgabenverwaltung, sowie Abrechnungen mit Belegen und Zeiterfassung stehen im Mittelpunkt der Anwendung
- Möglichkeit, Projekte und Aufgaben anzulegen, zu bearbeiten und zu löschen
- Statusverwaltung für Aufgaben durch Ziehen in verschiedene Spalten im Kanban (z. B. offen, in Bearbeitung, erledigt)
- Möglichkeit, Abrechnungen zu erstellen, bei denen Rechnungen hochgeladen und im Programm aufgerufen und angeschaut werden können
- Zeiterfassung: Verknüpfung der Zeiterfassung mit Benutzeraktivitäten, inklusive Start-/Endzeit und Dauer. Benutzer können Check-in und Check-out durchführen sowie die erfasste Zeit nachträglich bearbeiten.
- Einfache Benutzeroberfläche, die ohne Einarbeitung nutzbar ist
- **Datenhaltung:** Die Daten werden über eine Swagger-Schnittstelle auf eine PostgreSQL-Datenbank geladen und dort gespeichert.
- Rechnungen und weitere Projekterelevante uploadbare Dateien werden lokal gespeichert, da wir keinen Server haben.

Umsetzungsdetails (Pflichtenheft)

Softwarevoraussetzungen

- .NET Core (WPF)
- Microsoft.NETCore.App
- Microsoft.WindowsDesktop.App.WPF

Verwendete NuGet-Packages im Frontend:

- Extended.Wpf.Toolkit (4.7.25104.5739)
- LiveChartsCore.SkiaSharpView.WPF (2.0.0-rc5.4)
- MaterialDesignThemes (5.2.1)

- Microsoft.Web.WebView2 (1.0.3296.44)
- Moq (4.20.72)
- MSTest.TestAdapter (3.9.3)
- MSTest.TestFramework (3.9.3)
- Newtonsoft.Json (13.0.3)
- PdfiumViewer (2.13.0)
- PdfiumViewer.Native.x86.v8-xfa (2018.4.8.256)
- PdfiumViewer.Native.x86_64.v8-xfa (2018.4.8.256)
- Serilog (4.3.0)
- Serilog.Sinks.Console (6.0.0)
- Serilog.Sinks.File (7.0.0)
- SkiaSharp (3.119.0)
- System.Drawing.Common (9.0.6)

Python-Requirements im Backend:

- connexion[swagger-ui] >= 2.6.0; python_version>="3.6"
- connexion[swagger-ui] <= 2.3.0; python_version=="3.5" or python_version=="3.4"
- connexion[swagger-ui] <= 2.14.2; python_version>"3.4"
- werkzeug == 0.16.1; python_version=="3.5" or python_version=="3.4"
- swagger-ui-bundle >= 0.0.2
- python_dateutil >= 2.6.0
- setuptools >= 21.0.0
- Flask == 2.1.1
  allgemein requirments.txt

Funktionsblöcke / Architektur

Detaillierte Beschreibung der Umsetzung

Mögliche Probleme und ihre Lösung

Wie wurde die Software getestet?

- **Frontend:**
  Die Anwendung wurde auf verschiedenen Laptops und PCs sowie von unterschiedlichen Personen getestet, um Bedienbarkeit und Stabilität sicherzustellen.
- **Backend:**
  Die Backend-API wurde mit Postman und Swagger UI getestet.
- **Automatisierte Tests:**
  Für das Frontend wurden Unit-Tests mit MSTest durchgeführt.

Bedienungsanleitung

Quellen