

Trellomize Report

ارشیا صمیمی و امیرحسین رئیس سمیعی

وظایف انجام شده مشترک

- چیتش ساختار پروژه و کلاس‌های اصلی (کاربر، پروژه و وظیفه)
- طراحی ابزارهای کمکی
- تست و دیباگ کلی

وظایف انجام شده توسط ارشیا صمیمی

- لاگ کردن وقایع
- آزمون واحد (تست و بررسی متودها و کارایی کلاس‌های کاربر، پروژه و وظیفه)
- بخش مدیر سیستم (ساختن مدیر جدید، مدیریت کاربران از جمله غیرفعال کردن و...)

وظایف انجام شده توسط امیرحسین رئیس سمیعی

- مدیریت فایل
- رابط کاربری (ورود و عضویت، مدیریت گذرواژه، مدیریت و نمایش پروژه و وظایف، کامنت‌ها و تاریخچه)

توضیحات کلی پروژه

پروژه به زبان پایتون طراحی شده که زبانی قدرتمند و دارای ابزارهای فراوان است. از طرفی ساختارهای برنامه‌نویسی را در مقایسه با بعضی زبان‌های دیگر مانند C++ کمتر حمایت می‌کند که منجر به بهینگی پایین‌تر است و ساختارهای ضعیف‌تری قابل پیاده‌سازی خواهد بود. با این وجود این سیستم مدیریت پروژه به خوبی قابلیت ساخت و مدیریت پروژه و اعضای آن را داشته و در محیطی آسان کاربرد این قابلیت‌ها را ارائه می‌دهد.

برنامه با هر بار شروع فعالیت کل داده‌های کاربران، پروژه‌ها و وظایف را لود می‌کند که این کار موجب درگیر شدن دائمی بخشی از حافظه است، اما از طرفی حین برنامه از تداخل بین داده‌ها و همچنین اتلاف زمان برای باز و بسته کردن مکرر فایل داده‌ها جلوگیری می‌کند. استفاده از ساختمان‌های مختلف داده می‌تواند راه‌حل مناسبی برای این موضوع باشد، اما ساختار فعلی کاملاً پاسخگوی نیازهای حجم قابل توجهی از کاربران می‌باشد.

معماری ارائه شده با توجه به تلاش زیادی که جهت رعایت اصول برنامه‌نویسی شی‌گرایانه صورت گرفته، امکان قابل قبولی برای توسعه و گسترش امکانات و قابلیت‌ها فراهم می‌کند.

چالشی که این سیستم در دنیای واقعی با آن روبرو است توانایی آن در مقابل حجم‌های بالاتر داده‌های کاربری است. هش کردن حجم زیادی از اطلاعات موجب کارایی پایین‌تر برنامه شده که این موضوع در حجم داده زیاد چشمگیر است و نیازمند راه‌حل‌های کارآمد از جمله سیستم‌های بهینه‌تر مدیریت داده مثل دیتابیس‌ها و... می‌باشد.

کتابخانه‌های استفاده شده

در این برنامه از کتابخانه‌های زیادی استفاده شده است که هر کدام سهولت قابل توجهی در پیاده‌سازی اهداف پروژه فراهم کرده‌اند. برای مثال از کتابخانه **rich** برای جلوه‌های تصویری برنامه استفاده شده است و همچنین چندین کتابخانه دیگر که اهداف مشابهی داشته و از زمان گذاشتن روی کدنویسی مجدد کدهایی که به صورت آماده وجود دارند جلوگیری کرده‌اند. تعدادی دیگر از کتابخانه‌ها از بین شماری کتابخانه که اهداف یکسانی دارند انتخاب شده‌اند از جمله **loguru** و **pytest**. در هر مورد علت انتخاب راحتی بیشتر برنامه‌نویسان پروژه با کتابخانه موردنظر در بین کتابخانه‌های موجود بوده است، گرچه در مواردی از دو کتابخانه برای تکمیل یک بخش از پروژه استفاده شده است. برای مثال بخش اعظم آزمون واحد با کتابخانه **pytest** صورت گرفته است اما در کنارش از برخی قابلیت‌های **unittest** نیز بهره برده‌ایم.