

EACK XOA 2.0

سینا قاسمی نژاد

اردیبهشت ۱۴۰۰

این داک برای توضیح فاز دوم پروژه برنامه نویسی پیشرفته تهیه شده است و در آن به طور مختصر به توضیح کارکرد این پروژه پرداخته می شود.

۱. منابع استفاده شده

برای ساخت پروژه از بیلد تول gradle، برای گرافیک پروژه از JavaFx، از برای لاگ کردن برنامه از کتابخانه log4j، برای کانفیگ کردن از کتابخانه Properties، و برای ذخیره سازی فایل مدل های پروژه از کتابخانه Gson استفاده شده است.

در فاز یک از محمدامین رئیسی برای راه اندازی کتابخانه log4j کمک گرفته شده بود که در این فاز نیز از همان فایل ها استفاده شده است.

برای راه اندازی فایل های کانفیگ، از کلاس Config.java که در پکیج utils پروژه درس بی پی در ترم پیش وجود داشت (که سید امیرمحمد سادات شکوهی کد آن را زده است) استفاده شده و متودهای اضافه آن دیلیت شده اند.

برای پیاده سازی معماری MVC و تمیزی کد نیز در این فاز، از مریم مقدس کمک گرفته شده است.

۲. ساختار کلی کد و نقاط قوت و ضعف آن

همان طور که پیش تر گفته شد، این پروژه از معماری MVC یا همان Model, View, Controller استفاده می کند که در کلاس درس نیز به مزایای استفاده از این روش اشاره شده است.

این پروژه علاوه بر موارد بالا، از Listener ها نیز استفاده می کند، اشیائی که اتفاقات کلاس های View را به کلاس های Controller و برعکس اطلاع می دهند و اخبار و اطلاعات تحلیل شده را منتقل می کنند. برای برخی Listener ها نیز از FormEvent ها که کلاس هایی اکستند شده از کلاس EventObject هستند نیز استفاده شده است تا دیتای موجود در کلاس های View راحت تر منتقل شود.

علاوه بر موارد بالا، کلاس‌های موجود در پکیج `util` نیز کلاس‌هایی کمکی اند که صرفاً کدهایی که در گروه‌های دیگر قرار نمی‌گرفت، ولی در ساختار پروژه بسیار حیاتی بود در این پکیج قرار گرفته اند. از کلاس‌های مهم این پکیج می‌توان به `Validations` نام برد که در آن از `Regex` برای کردن معتبر بودن ورودی‌های کاربر استفاده شده است. کد باقی کلاس‌های این پکیج نیز از اینترنت پیدا شده اند که لینک آن‌ها به‌صورت کامنت در هر کلاس آمده است.

پکیج `db` نیز مسئول ذخیره‌سازی و لود کردن دیتای کاربران، توئیت‌ها و مسیج‌هاست. این پکیج داخل خود یک `interface` دارد که تمام کلاس‌های دیگر از آن `implement` کرده و توابع خود را از آن می‌گیرند. کلاس `DBUtils` نیز برای جلوگیری از تکرار کد در کلاس‌های مشابه `DB`‌های مختلف ساخته شده و کاربردی غیر از آن ندارد.

در پروژه چند نوع بخش گرافیکی مختلف داریم: پنل‌های اصلی فریم‌ها و صفحات برنامه:

۱- پنل‌های اصلی پنل‌هایی اند که در ابتدای برنامه اجرا می‌شوند. مانند صفحه اول برنامه، صفحه ورود، صفحه ثبت‌نام و پنل اصلی (صفحه‌ای که بعد از ورود به اکانت خواهد دید و همیشه روی برنامه خواهد ماند روی این صفحه یک `Pane` برای قرارگیری صفحات برنامه قرار دارد).

۲- فریم‌ها، مسئول بخش‌هایی اند که کاربر مایل به انجام عملیاتی است که نیاز به یک صفحه کامل نداشته و به‌صورت جداگانه و ریز می‌تواند برای او نمایش داده شود. مانند دیدن عکس پروفایل، توییت زدن، ادیت پیام، اد کردن یک یوزرنیم به چت گروهی و غیره. این فریم‌ها برای زیبایی بیش‌تر برنامه طراحی شده اند. در صورتی که می‌شد با همان صفحات برنامه نیز آن‌ها را به سادگی پیاده‌سازی کرد.

۳- صفحات برنامه نیز شامل بخش‌هایی از برنامه اند که روی پنل اصلی سوار می‌شوند. صفحاتی مانند تایم‌لاین، پروفایل، تنظیمات و ...

نقاط قوت: پیشرفت بسیار زیاد در زمینه تمیزنویسی کد (چه در ساختار `MVC` و چه در عدم وجود متغیر اضافی، کد تکراری و هرگونه `warning` در کد)، تمیزی نسبی محیط گرافیکی و قوی بودن بخش `logic` برای هندل کردن حالات مختلف. الگوریتم نسبتاً خوب برای بک‌زدن در برنامه (استفاده از ساختاری `Stack` طور). لاگ‌ری با سطوح اهمیت لاگ متفاوت.

نقاط ضعف: هنوز برخی کلاس‌ها `Singleton` می‌باشند و با وجود این‌که هیچ متغیری `public` نیست، همچنان برخی متغیرها حدود یک یا دو بار بین کلاس‌های مختلف پاس داده می‌شوند. ضعف بالا در امنیت فایل‌های کاربران، چه در حالتی که فایلی پاک شود، چه در حالتی که فردی غیر خود کاربر بخواهد به فایل دسترسی پیدا کند.

3. Resources

فایل‌ها بخش اعظمی از برنامه را تشکیل می‌دهند. دایرکتوری **configurations** محل ذخیره‌سازی فایل‌های کانفیگ برنامه است که به ۳ دسته آدرس‌ها، پترن‌ها و رنگ‌ها تقسیم می‌شوند. همچنین برای خود این فایل‌های کانفیگ نیز یک فایل کانفیگ وجود دارد. فایل آدرس‌ها به شخصه به من در تغییر پکیج‌بندی کمک شایانی کرد.

دایرکتوری **database** محل ذخیره‌سازی اطلاعات کاربران، توئیت‌ها و چت‌هاست. علاوه بر آن لیست ایمیل‌های ثبت‌شده، شماره تلفن‌های ثبت‌شده، آیدی‌های ثبت‌شده و آخرین آیدی موجود برای یوزر و چت نیز در این دایرکتوری قرار دارند.

دایرکتوری **graphic** محل ذخیره‌سازی فایل‌های **fxml** است که گرافیک پروژه را هندل می‌کنند. هر یک از این فایل‌ها نیز در صورت لزوم کنترلری گرافیکی (علاوه بر کنترلرهای اصلی برنامه که برای اشتباه نشدن نام، **logic** نام‌گذاری شده اند)، ای با پسوند **FXML** دارند.

دایرکتوری **images** نیز شامل دو عکس است که در صورت نبود یک عکس پروفایل برای کاربر، یا در صورت نبود یک عکس برای یک توئیت/مسیج نمایش داده می‌شوند. هر دوی این عکس‌ها را از اینترنت پیدا کردم.

فایل **log4j2.xml** برای راه‌اندازی لاگر برنامه استفاده شده و تمام اطلاعات را در **logs.log** ذخیره می‌کند. دقت کنید که این فایل **append true** دارد. یعنی با هر بار اجرای برنامه، لاگ‌های قبلی پاک نمی‌شوند.