توضیحات فاز 2

منابع به کار رفته در پیاده سازی کد:

1-Google.com
2-Stackoverflow.com
3-Javatpoint.com
4-https://www.javatpoint.com/javafx-tutorial
5-https://stackoverflow.com/questions/41851501/how-to-design-chatbox-gui-using-javafx/41851855
6-کد قسمت کانفیگ در پروژه برنامه نویسی مقدماتی پاییز 99

در روند طراحی کد از افراد مختلفی مشورت صورت گرفته که اصلی ترین آنها افراد زیر میباشند:

1-امیرمحمد سادات شکوهی
2-محمد امین رئیسی
3-سید عرفان موسویان
4-مهرشاد تازیکی
5-فرزام کرجی بانی
...

کتابخانه های استفاده شده:

1-Gson جهت کار با فایل های json و سیو و لود پروژه
2-Log4j جهت لاگ کردن اطلاعات برنامه
3-JavaFX برای پیاده سازی گرافیک برنامه

نحوه عملکرد کد:

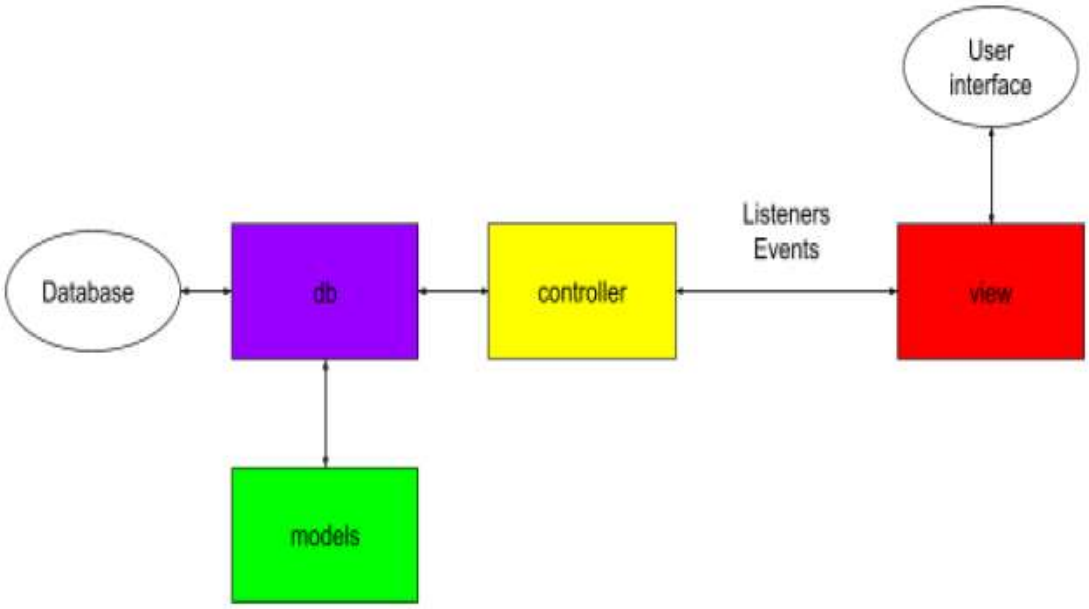
ساختار کلی برنامه:

برنامه از 5 بخش اصلی Controller,View,Listener,Event و Model تشکیل شده که هر بخش به طور جداگانه و همچنین در ارتباط با بقیه بخش ها با همکاری یکدیگر عملکرد پروژه را تشکیل میدهند.

توضیحات کامل هر کدام از بخش ها در ادامه آمده است اما نحوه ارتباط این بخش ها با یکدیگر به این صورت است که بخش Controller منطق اصلی برنامه را تشکیل داده و کلاس های دیتابیس و اطلاعات کاربر تحت کنترل این قسمت است که بقیه قسمت ها با تعامل با این قسمت اطلاعات مورد نیاز را به دست می آورند.

بخش View گرافیک اصلی برنامه است و کلاس های Event و Listener برای برقراری ارتباط بین View و Controller طراحی شده اند.

Model ها نیز کلاس هایی هستند که اشیا برنامه را تشکیل میدهند و اطلاعات مورد نیاز موقت برنامه در آنها نگه داری میشود.



Controller:

این کلاس از دو پکیج *DataBase* و *Pages* تشکیل شده است که در *DataBase* دو کلاس *DataBase* که پایگاه داده اصلی برنامه است و *TweetHandler* که مسئول پیگیری توییت های حاضر در حال نمایش و مدیریت آنها است میباشد.

پکیج *Pages* صفحات ثابت برنامه میباشند که چیزی در خود ذخیر نمیکنند اما با توابعی که دارند اطلاعات را از کاربر دریافت کرده و عملیات مورد نیاز را روی آنها انجام میدهند.

Event:

ایونت ها نشان دهنده اتفاقات و رویداد هایی هستند که در برنامه به وجود می آیند(مانند ساختن یک کاربر جدید یا ورود کاربر به برنامه) و تنها حامل اطلاعاتی اند که کلاس های Listener با استفاده از آنها فعالیت های مربوط را انجام میدهند.

Listener:

هر کلاس Listener مربوط به یک کلاس View میباشد که آن را به قسمت مربوط در Controller وصل میکند و وظیفه اش برقراری ارتباط بین منطق و گرافیک برنامه میباشد.همچنین توابع Listener معمولا ورودی هایی از کلاس های Event دارند که اطلاعات لازم را از آنها دریافت میکنند.

View:

در پکیج *View* یک کلاس به نام *MainView* وجود دارد که به شکل *SingleTone* طراحی شده و در ابتدای برنامه یک شی از آن ساخته میشود و مسئول عوض کردن صفحه های *View* ، برگشتن به صفحه قبل و به طور کلی مدیریت کلاس های گرافیک است.
بقیه کلاس های *View* هر کدام مربوط به یک فایل *fxml* میباشند که برای مدیریت توابع و فیلد های آن فایل و گرافیک مربوط طراحی شده اند همچنین برخی از این کلاس ها نیز به فایل *fxml* خاصی مربوط نمیشوند(مانند *MiniProfileViewer*) که مسئولیت این کلاسها تولید تعدادی شکل و شی گرافیکی به صورت پویا و اضافه کردن آن به مکان های مربوط در گرافیک میباشد.

Model:

کلاس های مدل همانطور که گفته شد اشیا نگه دارنده اطلاعات برنامه هستند و بقیه قسمت های برنامه با استفاده از این اشیا عملیات های مورد نیاز خود را انجام میدهند
مثال :

User,Message,ChatRoom,Group,...

نقاط قوت برنامه:

- استفاده از فایل های جیسون برای ذخیره سازی اطلاعات
- استفاده از اصول کلین کد و *MVC* جهت بالابردن خوانایی کد و افزایش بازدهی
- تلاش در جهت هر چه کوتاه تر کردن سایز های کلاس ها
- طراحی مستقل از یکدگیر سرور و کلاینت جهت سهولت در فاز های بعدی
- طراحی متد های مختلف برای وظایف مشخص به جهت استفاده بهینه در نقاط مختلف برنامه

نقاط ضعف برنامه:

- کامل نبودن برقراری ارتباط برنامه با کاربر به طور مثال عدم وجود پیام هایی در جهت نشان دادن ارورهای برنامه و خطاهای کاربر در ورودی دادن
- تکرار کد در برخی قسمت ها و عدم استفاده از توابع جهت کوتاه سازی کد
- در صورت وجود نداشتن فایل های مربوط به اطلاعات برنامه روند برنامه مختل میشود