

## قسمت کدهای برنامه:

### پکیج model:

همه‌ی مدل‌ها از کلاس Model ارث می‌برند که دارای ویژگی‌های زمان ساخته شدن (createdAt) ، آخرین زمان تغییر (lastModified) و یک آیدی است. همچنین isDeleted مشخص می‌کند این مدل پاک شده یا خیر، همچنین تمامی فیلدهای مدل‌ها private هستند و با getter و setter صدا زده می‌شوند. در model.fields فیلدهایی خاص نگهداری شده:

فیلد LockedField که یک AccessLevel را با یک تایپ pair می‌کند و برای فیلدهایی از یوزر تعریف می‌شود که سطح دسترسی در آن‌ها مهم است.

فیلد NotificationType که از سه نوع REPORT (ریپورت‌هایی که مستقیم به یوزر می‌شود) ، INFO و REQUEST است.

### پکیج db:

شامل بخش‌های زیر است:

#### • پکیج dbset

- وظیفه‌ی اتصال برنامه به دیتاباز را به عهده دارد و عملیات‌های ذخیره‌سازی و لود کردن را انجام می‌دهد.
- در آن ImageDBSet وظیفه‌ی ذخیره‌سازی عکس‌ها و نسبت دادن رشته‌ی آیدی به آن‌ها را به عهده دارد.
- در ImageDBSet برای سرعت بیشتر در لود شدن عکس‌ها لیستی به نام cache داریم که عکس‌هایی که حداقل یک بار لود شده‌اند در آن نگهداری می‌شوند تا نیازی به خواندن دوباره از Hard نباشد.
- کلاس abstract DBSet
  - به صورتی کار می‌کند که اگر مدل داده شده id نداشته آن را می‌سازد و ذخیره می‌کند اما اگر مدل مورد نظر id داشته باشد، آن را آپدیت می‌کند.
  - تابع validate که ولیدیشن‌های مربوط به مدل را انجام می‌دهد.
  - توابع getAll و getFirst که در queryBuilderها کاربرد دارند و فیلترینگ مدل‌ها را انجام می‌دهند.

- **پکیج exception**
  - کلاس `ConnectionException` که خطا در اتصال با دیتا را نشان می‌دهد.
  - کلاس `validation` که وظیفه‌ی نشان دادن خطاها روی `field`های مدل است و توابعی برای راحتی لاگ کردن و دسترسی به اوروهای هر فیلد دارد.
- **پکیج gsonAdapter**
  - توابعی برای شناساندن `LocalDate` و `LocalDateTime` به کتابخانه `gson`.
- **پکیج queryBuilder**
  - شامل فیلترهای پرکاربرد مورد استفاده برای هر مدل می‌شود که یک `Predicate` خروجی می‌دهند و در توابع `getFirst` و `getAll` در `DBSet` استفاده می‌شوند.
- **کلاس Context**
  - شامل یک `instance` از همه‌ی `DBSet`هاست.

#### پکیج util:

دارای کلاس `Config` است که برای لود کردن کانفیگ‌ها از آن استفاده می‌شود.  
 پکیج `validator` که ولیدیتورهای ایمیل و شماره تلفن را شامل می‌شود.

#### پکیج view:

شامل گرافیک `Dialog`های برنامه و `ViewManager` است.

#### کلاس ViewManager:

کلاس `static` است که وظیفه‌ی مدیریت صفحه‌ها و نمایش آن‌ها را دارد.  
 تابع `changeCenter` در آن اطلاعات اصلی صفحه را تغییر می‌دهد و آن را در `history` پوش می‌کند تا به کمک آن تابع `back` (رفتن به صفحه قبل) پیاده سازی شود.  
 تابع `connectionFailed` نیز خطاهای اتصال به فایل‌های دیتا را به کاربر نشان می‌دهد.

#### پکیج controller:

شامل کنترلرهای اصلی و همچنین کنترلرهایی برای استفاده راحت‌تر از `DBSet`هاست به عنوان مثال تابع `follow` که وظیفه‌ی انجام عملیات فالو کردن یوزر دیگر (به همراه نوتیفیکیشن و ...) دارد.

پکیج apps:

در واقع کنترلرهای جدا شده‌ی هر بخش در آن قرار دارد که وظیفه‌ی کنترل کردن viewها و تغییر آن‌ها را دارند.

## قسمت Resources برنامه:

فولدر config:

شامل اطلاعات مکان فایل‌های fxml است.

همچنین در lang.properties متن‌های ثابت در برنامه و Iconهای آن آمده که قابل تغییرند.

فولدر view:

شامل فایل‌های fxml است که سعی شده تا جای ممکن از هم مستقل و قابل باز مصرف باشند در همین جهت در بعضی appها از Cardها استفاده شده که در واقع قسمت‌های کوچک‌تر از گرافیک هستند و می‌توانند با لود شدن در view دیگری مورد استفاده قرار گیرند.

## کتابخانه‌های استفاده شده:

کتابخانه‌ی javafx

کتابخانه‌ی gson برای save/load کردن مدل‌ها با json

کتابخانه‌ی log4j برای لاگ کردن

کتابخانه‌ی fontawesome-fontawesomefx برای آیکون‌ها

کتابخانه‌ی jfoenix برای برخی اجزای گرافیکی

برای build پروژه نیز از gradle استفاده شده

منابع:

مشورت انجام شده با امیرسالار صفایی در مورد گرافیک برنامه

تبدیل BufferedImage به Image قابل استفاده در javafx ([لینک](#))

آداپتورهای gson ([لینک](#))