

به نام خدا

مستند اپلیکیشن یادمان

تیم برنامه نویسی:
محمد امین سالار کیا
مهدی قنبری
محمد تابع امام
امیر حسین مهدی نژاد
امیر سده

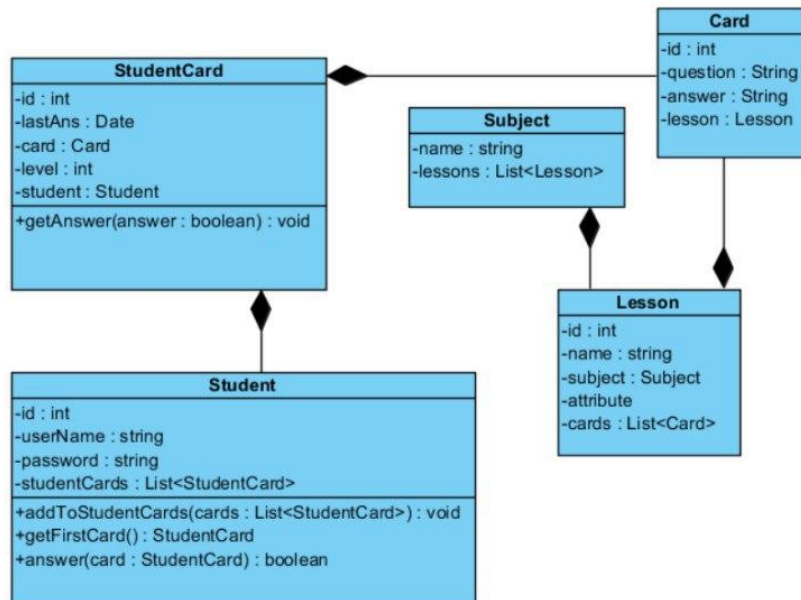
۱۳۹۶/۱۲/۱۶

۱ مقدمه

اپلیکیشن یادمان نرم افزاری جهت یادگیری آسان دروس حفظی کنکور به روش لایتنر است. در این اپلیکیشن با استفاده از دیتابیس جامعی از دروس حفظی کنکور دانش آموز با اضافه کردن کارت به جعبه لایتنرش میتواند دروس را مرور کند. مخاطب این اپلیکیشن تمام دانش آموزان دبیرستانی و به خصوص کنکوری ها هستند. در این مستند سعی شده تمام کدهای پیاده سازی شده و بخش های اپلیکیشن توضیح داده شود. ابتدا طراحی مدل های برنامه و دیتابیس توضیح داده شده و سپس مازول های برنامه آورده شده. کدهای اپلیکیشن را میتوانید در [این مخزن](#) ببینید.

۲ طراحی مدل ها و دیتابیس

نمودار uml مدل ها را در شکل ۱ میتوانید ببینید.



شکل ۱: نمودار uml مدل‌ها

مدل StudentCard نشان‌دهنده کارت‌های اضافه شده به جعبه است. مدل‌های Card, Lesson, Subject مربوط به دیتابیس استاتیک بوده و به ترتیب مبحث (ادبیات، زبان، ...)، درس (درس یک، درس دو، ...) و کارت‌های هر درس هستند.

به غیر از این مدل‌ها مدل‌های دیگری هم (مثل Notification, QueueObject, Quiz, ...) برای پیاده‌سازی ماژول‌های مختلف ساخته شده که در توضیح همان ماژول آورده شده.

۳ ماژول‌ها

تمام فایل‌های برنامه در فولدرهای activities, database, models, ui, utils پیاده‌سازی شده. در فولدر ui فایل‌های مربوط به رابط کاربری (دیالوگ‌ها، لیست‌ها، کنترل فونت و ...) قرار داده شده. در فولدر database هم فایل‌های مربوط به خواندن، اضافه کردن، آپدیت کردن و پاک کردن مدل‌ها از دیتابیس برنامه قرار داده شده. در ادامه ماژول‌های برنامه آورده شده که فایل‌های مربوط به هر ماژول توضیح داده شده. این

فایل‌ها در یکی از پنج فولدر بالا قرار دارند.

۱.۳ ورود و ثبت‌نام و خروج

فایل‌های مربوطه:

LoginActivity, EmailSignUpActivity, SignUpActivity, SignUpDetailsActivity, VerificationActivity, Connection, SharedPreferencesHandler, ConnectionUi

ماژول ورود و ثبت‌نام از طریق سرور اولین ماژولی است که کاربر با آن مواجه می‌شود. تمام ارتباط‌ها با سرور از طریق کلاس Connection انجام می‌شود. واسط کاربری در حین ارتباط‌ها با سرور در ConnectionUi پیاده‌سازی شده. ابتدا کاربر وارد LoginActivity می‌شود برای ورود. در صورت کلیک بر روی گزینه ثبت‌نام وارد SignUpActivity می‌شود برای ثبت‌نام. بعد از وارد کردن شماره تلفن وارد صفحه SignUpDetailsActivity می‌شود و بعد از تایید شماره تلفن وارد VerificationActivity می‌شود برای تکمیل اطلاعات ثبت‌نام. اگر کاربر بخواهد از طریق ایمیل ثبت‌نام کند (نه شماره تلفن) وارد صفحه EmailSignUpActivity می‌شود. اطلاعات کاربر وارد شده در cache برنامه ذخیره می‌شود که این کار از طریق SharedPreferencesHandler انجام می‌شود.

۲.۳ شروع کار

فایل‌های مربوطه:

FirstPageActivity, HomeActivity, DatabaseDownloader, FileDownloader, TutorialViewer

در اولین بار ورود کاربر فایل دیتابیس دروس دانلود می‌شود. اینکار از طریق DatabaseDownloader و FileDownloader انجام می‌شود. در ضمن پس از اولین ورود آموزش استفاده از اپلیکیشن نمایش داده می‌شود. که این کار از طریق TutorialViewer انجام می‌شود. هر دفعه که اپلیکیشن باز می‌شود آیکون یادمان در یک صفحه برای مدت کمی نشان داده می‌شود که این کار در FirstPageActivity انجام می‌شود.

۳.۳ انتخاب کارت، جعبه لایتنر

فایل‌های مربوطه:

SelectCardsActivity, CardActivity, ImageDownloader, DateTimeUtils, Tick

در بخش انتخاب کارت کاربر میتواند با تیک زدن درس یا مبحث یا سال خاص و زدن دکمه تایید تعداد کارت وارد جعبه لایتنرش کند یا از آن خارج کند. این کار در SelectCardsActivity و مدل Tick انجام شده.

در بخش مرور کارت کاربر میتواند کارت‌های جعبه را با سیستم لایتنر ۰، ۲، ۴، ۸، ۱۵ مرور کند. هر کارت یک سوال دارد که با زدن گزینه نمایش پاسخ، پاسخ آن نمایش داده میشود. این قسمت در CardActivity پیاده‌سازی شده.

تاریخ‌ها به صورت میلادی در دیتابیس ذخیره میشوند. تمام کارهای مربوط به تغییر یا تبدیل تاریخ در DateTimeUtils پیاده سازی شده.

برای کارت‌هایی که عکس داشته باشند هر دفعه عکس‌ها دانلود میشوند که این کار در ImageDownloader پیاده‌سازی شده.

۴.۳ فروشگاه

فایل‌های مربوطه:

ShoppingActivity و درون برنامه‌ای بازار

در بخش فروشگاه کاربر میتواند کارت‌های مربوط به مباحث مختلف را بخرد. پس از خرید که از طریق ارتباط با سرور انجام میشود درس‌های خریده شده در دیتابیس برنامه ثبت میشوند. این کار در ShoppingActivity انجام شده. در ضمن از ماژول پرداخت درون برنامه‌ای بازار (com.example.android.trivialdrivesample.util) برای پیاده‌سازی پرداخت استفاده شده.

۵.۳ آزمون‌ها

فایل‌های مربوطه:

SelectCardsActivity, Quiz, QuizCenter, Connection

دانش آموز میتواند در بخش انتخاب کارت با انتخاب آزمون‌های قلمچی، گزینه‌دو، گاج و سنجش، کارت‌های مربوط به هر آزمون را به جعبه لایتنر خود اضافه کند. دریافت اطلاعات هر

آزمون از طریق سرور انجام میشود. مدل‌های آزمون‌ها در Quiz و QuizCenter پیاده‌سازی شده. انتخاب آزمون هم در SelectCardsActivity انجام میشود.

۶.۳ اطلاع‌رسانی (Notification)

فایل‌های مربوطه:

NotificationsActivity, NotificationEventReceiver, NotificationIntentService, NotificationServiceStarterReceiver, Notification

در طول زمان سه نوع پیام به کاربر داده میشود. یک نوع پیام مربوط به یادآوری مرور کارت است که هر روز به کاربر داده میشود. یک نوع پیام مربوط به یادآوری هماهنگ‌سازی اطلاعات با سرور است که اگر سه روز از آخرین هماهنگ‌سازی اطلاعات با سرور گذشته باشد به کاربر داده میشود. و یک نوع دیگر پیام، پیام‌هایی است که در سرور در نظر گرفته میشوند (مثلاً برای مناسبت‌های خاص یا تبلیغ یا ...) که برنامه هر ۶ ساعت یکبار چک میکند که اگر در سرور پیام جدیدی موجود بود به اطلاع کاربر میرساند.

زمانبندی و ارسال پیام در NotificationEventReceiver و NotificationIntentService و NotificationServiceStarterReceiver انجام میشود. در ضمن همه پیام‌ها در NotificationActivity نشان داده میشوند. که مدل Notification در دیتابیس ذخیره میشود و برای نشان دادن همه پیام‌ها استفاده میشود.

۷.۳ هماهنگ‌سازی با سرور

فایل‌های مربوطه:

HomeActivity, DataSyncer, QueueObject

تمام کارهای کاربر به صورت یک صف در دیتابیس داخلی از طریق مدل QueueObject ذخیره میشود. بعد از اینکه کاربر دکمه هماهنگ‌سازی با سرور را در HomeActivity کلیک کرد تمام این صف از طریق DataSyncer در سرور اعمال میشود و سپس کل کارت‌های سرور در دیتابیس داخلی قرار داده میشوند.

۸.۳ تنظیمات

فایل‌های مربوطه:

SettingsActivity, Connection, SharedPreferencesHandler

تنظیمات اکانت (تغییر ایمیل، تغییر رمز عبو، خروج از حساب و حذف اکانت) و اندازه فونت و تنظیمات ارسال پیام و ذخیره‌سازی عکس کارت‌ها در SettingsActivity قابل تغییر است.

۴ ایرادات و توسعه‌های آتی

در اینجا چند ایراد که رفع آن‌ها در بهتر کردن کد موثر است آورده شده:

- در کار با دیتابیس میتوان از کتابخانه‌های ORM استفاده کرد که هم به بهتر شدن کد و هم به سرعت برنامه کمک میکند.
- در ارسال Push-Notification میتوان از سرویس‌های آماده مانند Google Firebase استفاده کرد.
- هماهنگسازی اطلاعات با سرور میتواند با الگوریتم‌های بهتر و با سرعت بالاتر انجام شود
- بهتر است برای مازول‌های مختلف Test های مختلف نوشته شود که در این بین میتوان از سرویس های تست نویسی اندروید (مانند Monkey-test و ...) استفاده کرد.