



دانشکده مهندسی کامپیوتر

گزارش کارآموزی

نام و نام خانوادگی کارآموز: محمد درخشان

شماره دانشجویی: ۹۵۲۶۶۰۳

استاد کارآموزی: دکتر مازیار پالهنگ

سرپرست کارآموزی: دکتر جهانیان

تلفن: ۰۳۱-۳۷۹۳۳۱۷۵

محل کارآموزی: پژوهشکده‌ی مدیریت و اقتصاد دانشگاه اصفهان

آدرس: اصفهان، دانشگاه اصفهان، ساختمان مرکزی، طبقه‌ی همکف، پژوهشکده‌ی مدیریت

تاریخ پایان: ۱۳۹۹/۰۶/۲۰

تاریخ شروع: ۱۳۹۹/۰۴/۲۸

فرم شماره ۱

گزارش خلاصه ای از فعالیتهای هفتگی

شماره دانشجویی: ۹۵۲۶۶۰۳

نام و نام خانوادگی کارآموز: محمد درخشان

دانشکده: مهندسی کامپیوتر

رشته تحصیلی: کامپیوتر (نرم افزار)

<ul style="list-style-type: none"> - آشنایی با مفاهیم مدیریت مالی، دارایی جاری، بدھی جاری و سرمایه در گردش - آشنایی با چرخه‌ی عملیاتی و چرخه‌ی نقدینگی - آشنایی با مفاهیم اولیه‌ی بازار‌های مالی، ساختار سرمایه 	<p>هفته اول از تاریخ: ۱۳۹۹/۰۴/۲۸ لغایت تاریخ: ۱۳۹۹/۰۵/۰۲</p>
<ul style="list-style-type: none"> - آشنایی با نحوه‌ی نگارش و تدوین پروپزال ، سنجش رقبا و بازار هدف - آشنایی با آمیختگی بازاریابی و مفاهیم 4P - آشنایی با نحوه‌ی هزینه‌یابی شرکت‌ها و روابط بدست آوردن هزینه به درآمد 	<p>هفته دوم از تاریخ: ۱۳۹۹/۰۵/۰۴ لغایت تاریخ: ۱۳۹۹/۰۵/۰۹</p>
<ul style="list-style-type: none"> - آشنایی اولیه با بلاک چین و مفاهیم این حوزه - آشنایی با نرم افزار‌های توزیع شده یا DApp - آشنایی اولیه با سالیدیتی و نحوه‌ی راه اندازی سرور نرم افزار‌های DApp 	<p>هفته سوم از تاریخ: ۱۳۹۹/۰۵/۱۱ لغایت تاریخ: ۱۳۹۹/۰۵/۱۶</p>
<ul style="list-style-type: none"> - آشنایی با چگونگی راه اندازی نرم افزار‌های اندرویدی - ایجاد صفحه‌ی لაگین مبتنی بر مکانیزم OTP برای احراز هویت کاربر - آشنایی با ایجاد shape و پیاده سازی صفحه‌ی اصلی نرم افزار 	<p>هفته چهارم از تاریخ: ۱۳۹۹/۰۵/۱۹ لغایت تاریخ: ۱۳۹۹/۰۵/۲۳</p>

نام و نام خانوادگی سربرست کارآموزی: دکتر جهانیان

امضاء و مهر سربرست
[Handwritten Signature]

فرم شماره ۱ (ادامه)

گزارش خلاصه ای از فعالیت های هفتگی

شماره دانشجویی: ۹۵۲۶۶۰۳

نام و نام خانوادگی کارآموز: محمد درخشان

دانشکده: مهندسی کامپیوتر

رشته تحصیلی: کامپیوتر (نرم افزار)

<p>- بهره گیری از دیزاین پترن Singleton برای ایجاد پایگاه داده، طراحی مدل های جداول، استفاده از Live data برای دسترسی به اطلاعات ذخیره شده در پایگاه داده</p> <p>- آشنایی با معماری MVVM در اندروید</p> <p>پیاده سازی منطق (logic) صفحه های پروفایل در قالب معماری MVVM</p>	<p>هفته پنجم</p> <p>از تاریخ: ۱۳۹۹/۰۵/۲۵ لغایت تاریخ: ۱۳۹۹/۰۵/۳۰</p>
<p>آشنایی با مفهوم Dagger و استفاده از فرم وورک 2</p> <p>آشنایی با ایجاد Http request در اندروید و دریافت اطلاعات از سورس سالیدیتی</p> <p>آشنایی با Repository Design pattern در اندروید</p>	<p>هفته ششم</p> <p>از تاریخ: ۱۳۹۹/۰۶/۰۱ لغایت تاریخ: ۱۳۹۹/۰۶/۰۶</p>
<p>آشنایی با ایجاد انیمیشن با استفاده از Motion layout و پیاده سازی صفحه های پروفایل با استفاده از این لایه</p> <p>آشنایی با Navigation Component و Navigation Graph و بهره مندی از آن برای حرکت بین منوها های پروفایل</p>	<p>هفته هفتم</p> <p>از تاریخ: ۱۳۹۹/۰۶/۱۰ لغایت تاریخ: ۱۳۹۹/۰۶/۱۳</p>
<p>آشنایی با سرویس FireBase و استفاده از آن برای تحلیل آمار، گزارش خودکار کردن نرم افزار، بررسی عملکرد و بهینه بودن نرم افزار و تست نرم افزار های اندرویدی با استفاده از FireBase Test Lab</p>	<p>هفته هشتم</p> <p>از تاریخ: ۱۳۹۹/۰۶/۱۵ لغایت تاریخ: ۱۳۹۹/۰۶/۲۰</p>

نام و نام خانوادگی سربرست کارآموزی: دکتر جهانیان

امضاء و مهر سربرست
دکتر جهانیان

فرم شماره ۲

گزارش سرپرست کارآموزی

نام و نام خانوادگی سرپرست کارآموزی: دکتر جهانیان

نام واحد صنعتی: پژوهشکده مدیریت و اقتصاد دانشگاه اصفهان

شماره دانشجویی: ۹۵۲۶۶۰۳

نام و نام خانوادگی کارآموز: محمد درخشنان

دانشکده: مهندسی کامپیوتر

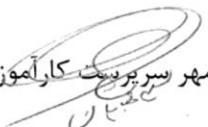
گرایش تحصیلی: کامپیوتر (نرم افزار)

ردیف	اظهار نظر سرپرست کارآموز	ضعیف متوسط خوب عالی
1	رعایت نظم و ترتیب و انضباط در محیط کار	/
2	میزان علاقه و همکاری با دیگران	/
3	علاقه به فراغیری مطالب علمی و فنی	/
4	پیگیری وظایف و میزان پشتکار	/
5	ارزش پیشنهادات کارآموز در جهت بهبود کار	/
6	کیفیت گزارش‌های کارآموزی (حداقل فرمهای شماره (1)	/
7	میزان بهره گیری از امکانات موجود جهت ارتقاء توانایی علمی و فنی	/

تعداد روزهای غیبت: نیازد

تعداد روزهای مرخصی: نیازد

پیشنهادات سرپرست کارآموزی جهت بهبود دوره کارآموزی:

امضاء و مهر سرپرست کارآموزی


فرم شماره ۳

نظرات و پیشنهادات

(در پایان دوره تکمیل شود)

شماره دانشجویی: ۹۵۲۶۶۰۳

نام و نام خانوادگی کارآموز: محمد درخشان

دانشکده: مهندسی کامپیوتر

رشته تحصیلی: کامپیوتر (نرم افزار)

شرح نظرات و پیشنهادات:

1. در مورد دوره کارآموزی و مراحل مختلف آن:

کارآموزی تمرینی برای آشنایی با محیط کار و مهارت هایی است که در طول تحصیل فراغرفته شده است. لذت کار کردن گروهی و به ثمر نشستن پژوهه ها از قسمت های خوب این دوران می باشد. توجه به نیاز صنعت در حین تحصیل از تکات مثبت این دوره است.

اگر دانشجویان به چشم بیگاری به این دوره نگاه نکنند و در زمان و مکان درست باشند، این دوره از بهترین دوره های تحصیل آنان خواهد بود، جایی که علاوه بر یادگیری مهارت و تمرین عملی دروس، نظم و انضباط و همکاری گروهی را نیز فرا می گیرند.

2. در مورد امور پژوهشی، فنی و تولیدی محل کارآموزی:

مجموعه‌ی داده پردازان خوارزمیک، که زیر نظر پژوهشکده مدیریت و اقتصاد دانشگاه اصفهان کار میکند از جمله مراکز فعال در زمینه‌ی بازار های مالی می باشد. این مجموعه در راستای تحقق اهداف خود اقدام به تولید محصولات نرم افزاری در همین زمینه نموده اند. حضور در این مجموعه به عنوان کارآموز نه تنها منجر به آشنایی بیشتر بnde با توسعه‌ی نرم افزار های اندرویدی شد بلکه منجر به آشنایی با مفاهیم مالی و بازار های مالی گردید.

توجه: علاوه بر ارائه فرم نظرات و پیشنهادات در پیوست گزارش تفصیلی، در صورت تمایل یک کپی از این فرم را به دفتر ارتباط با صنعت دانشکده تحويل نمایید.

امضاء کارآموز

فهرست مطالب

۶.....	چکیده
۷.....	فصل اول معرفی محل کارآموزی
۷.....	مقدمه
۷.....	معرفی شرکت
۸.....	فصل دوم کارهای انجام شده
۸.....	مقدمه
۸.....	گام اول، آشنایی با مفاهیم مالی و مدیریتی
۹.....	گام دوم، درک ارتباط سرور و کلاینت
۱۰.....	گام سوم، شروع توسعهٔ نرم افزار
۱۰.....	پیاده سازی صفحات لاگین و خانه
۱۲.....	اتصال پایگاه داده و کد زنی منطق صفحهٔ پروفایل
۱۶.....	آشنایی با الگوی طراحی Repository و اتصال به سرور
۱۷.....	پیاده سازی UI مربوط به صفحهٔ پروفایل
۲۱.....	استفاده از سرویس Firebase
۲۷.....	فصل سوم پیوست ها

چکیده

دوره‌ی کارآموزی از دوره‌های کارآمد در طول تحصیل هر دانشجو می‌باشد. در طول این دوره دانشجو با شیوه‌ی بهره‌گیری از دانسته‌های فراگرفته شده در دانشگاه و ترکیب آن‌ها با صنعت، آشنا می‌شود. اگر دانشجو در طول این دوره سعی بر حل مشکلات و مسائل موجود با بهره‌گیری از علوم دانشگاهی نماید، هدف اعلای این دوره تحقق پیدا کرده است.

شرکت داده پردازان خوارزمیک، که زیر نظر پژوهشکده‌ی مدیریت و اقتصاد دانشگاه اصفهان، مشغول فعالیت است با جذب کارآموزان دانشگاهی سعی بر بهبود فضای کاری خود دارد. در طول دوره‌ی کارآموزی، ابتدا به دانشجو مفاهیم اولیه‌ی مربوط به بازار‌های مالی و مفاهیم ریاضیاتی این زمینه آشنا می‌شود تا بتواند در ک درستی از نرم افزاری که در حال توسعه‌ی آن است بدست آورد. در ادامه پس از آشنایی مختصری با تیم برنامه‌نویسی سمت سرور که در حال توسعه‌ی BackEnd با استفاده از سالیدیتی می‌باشند، به صورت تخصصی به فراگیری شیوه‌ی پیاده سازی نرم افزار اندروید پرداخته می‌شود. در طول دوره تلاش شد که مفاهیم مهم و مطرح در دنیای توسعه‌ی نرم افزار‌های اندرویدی به خوبی پوشش داده شود که به لطف خدای منان، این هدف محقق شد.

فصل اول

معرفی محل کارآموزی

۱-۱- مقدمه

شرکت داده پردازان خوارزمیک، زیر نظر پژوهشکده مدیریت و اقتصاد دانشگاه اصفهان، از شرکت های فعال در عرصه بزار های مالی و تولید نرم افزار هایی در این زمینه است. این مجموعه تحت نظر جناب امیر حسین صنعتگر، مدیر عامل شرکت، اداره می شود. از دیگر بنیان گذاران این مجموعه می توان به سر کار خانم آرزو کرباسیون و مریم جان ثاری اشاره نمود. هر سه تن، از فارغ التحصیلان اسبق دانشگاه صنعتی اصفهان هستند.

۱-۲- معرفی شرکت:

شرکت داده پردازان خوارزمیک واقع در خیابان شیخ بهائی، از شرکت های فعال در زمینه دیجیتال مارکتینگ، مشاوره بازار های مالی و تولید نرم افزار های مرتبط با این بازار می باشد.

این مجموعه توسط سه تن، بنیان گذار، راه اندازی گردیده و هم اکنون بیش از ۱۵ نفر در این مجموعه در حال فعالیت هستند. اعضای این مجموعه در دو تیم مجزا مشغول فعالیت هستند: تیم تحلیل و آنالیز بازار های مالی و تیم طراحی و توسعه نرم افزار.

هر چند در ظاهر این دو تیم فعالیت های مجزایی از یک دیگر دارند، اما در انتها به یک هدف منجر می شوند. بهره گیری از سیستم های هوش مصنوعی و طراحی بات هایی برای مدیریت بازار های مالی خارجی، نقطه ایست که اهداف این دو تیم را همسو می کند.

راه اندازی سایت برای چند برنده صادراتی و بازرگانان ایرانی - خارجی، ارائه می شاوره های مارکتینگ، تولید تیز های تبلیغاتی و توسعه نرم افزار "آی بورس" از جمله فعالیت های حاضر در روزمه تیم برنامه نویسی این شرکت می باشد.

فصل دوم

کارهای انجام شده در دوره کارآموزی

۱-۲- مقدمه

در دوره‌ی دو ماهه‌ی کارآموزی در مجموعه‌ی داده پردازان خوارزمیک سعی بر آن شد که کارآموز با شیوه‌ی توسعه‌ی نرم افزارهای اندرویدی آشنا شود. از آنجایی که محصول خروجی و نتیجه‌ی کار، نرم افزاری در زمینه‌ی بازارهای مالی می‌باشد و تسلط توسعه‌دهنده به این مفاهیم و درک از سیستمی که برنامه نویس در حال تدوین آن می‌باشد، در بهبود کیفیت خروجی و سرعت توسعه‌ی نرم افزار بسیار موثر است، لذا لازم بود در وهله‌ی اول، کارآموز با مفاهیم این بازار آشنا شود. زین سبب در گام اول، آموزش‌هایی در زمینه‌ی مدیریت و استراتژی‌های مالی به وی آموزش داده شد. در ادامه مقدماتی از برنامه نویسی سالیدیتی بیان گردید تا درک ارتباط BackEnd و FrontEnd ملموس‌تر باشد. در ادامه در طول ۵ هفته‌ی باقی مانده، شیوه‌ی توسعه‌ی نرم افزار در محیط اندروید استودیو و تست نرم افزار با استفاده از سرویس FireBase محقق شد. در طول فرآیند کار، توسعه‌ی نرم افزار در قالب اسکرام، و در اسپرینت‌های یک هفته‌ای صورت گرفت. هم چنین از نرم افزار trello برای مدیریت و تخصیص task به هر فرد استفاده گردید.

۲-۲- گام اول، آشنایی با مفاهیم مالی و مدیریتی:

بدون شک درک توسعه‌دهنده، از محصولی که در حال ایجاد آن می‌باشد از عوامل موثر در فراهم شدن خروجی مطلوب است. لذا در گام اول لازم بود با مفاهیم مورد نیاز در تدوین نرم افزار آشنا شوم.

در هفته‌ی اول فراغرفته شد که در هر شرکت برخی از اموال و سرمایه‌ها هستند که به صورت معقول قابلیت تبدیل شدن به وجه نقد، در زمانی کوتاه، را دارند. به این دسته از اموال، دارایی‌های جاری می‌گویند. این دارایی‌ها از آن جهت برای شرکت‌ها مورد توجه هستند که می‌توانند به عنوان منبعی برای تامین نیازهای مالی روزانه تلقی شوند. افزایش بازدهی سرمایه‌گذاری‌های انجام شده در شرکت و تلاش برای بالا بردن فرآوری حاصله از مهمترین اهداف اقتصادی- مدیریتی شرکت محسوب می‌شود. در حقیقت، تلاش مدیران و برنامه‌ریزان شرکت‌ها و موسسات اقتصادی بر این است که با مطالعه اطلاعات و آمار و ارقام موجود در صورت‌های مالی، شما بی‌از عملکرد شرکت به

دست آورند و بتوانند با بررسی وضعیت حاضر و در نظر گرفتن اهداف استراتژیک و بلند مدت سازمان، برای افزایش فرآوری، حداکثر کردن ثروت سهامداران و سود شرکت، از نقاط قوت و ضعف خویش آگاهی یافته و با اتخاذ راهبردهای شایسته، موجبات بهینه سازی فعالیت‌ها و ساختار منابع موجود و به دنبال آن بهره جویی قابل قبول از سرمایه‌گذاری‌های انجام شده را فراهم آورند.

هر شرکت بسته به شرایط خود دارای سلامت مالی می‌باشد که از عوامل بسیار مهم و تاثیر گذار در ارزش گذاری آن شرکت و جذب سرمایه گذار می‌باشد. سرمایه‌ی درگوش فاکتوری است که با استفاده از آن می‌تواند سلامت مالی شرکت را ارزیابی نمود. به عبارت دیگر، سرمایه‌ی درگوش میزان نقدشوندگی و کارایی شرکت را نشان می‌دهد و از تفرقه‌بدھی‌های جاری و سرمایه‌ی جاری حاصل می‌شود.

بر این اساس، تجزیه و تحلیل کارآمد صورت‌های مالی می‌تواند زمینه لازم را برای شناسایی مشکلات پیش روی شرکتها ایجاد کند و راهگشای بسیاری از تصمیم‌گیری‌ها و آینده نگری‌ها باشد. همچنین سرمایه‌گذاران با استفاده از داده‌های صورت‌های مالی شرکت می‌توانند نسبت به وضعیت بنیادی شرکت تصمیماتی را اتخاذ نمایند که منجر به خرید یا فروش سهام آن شرکت در بازارهای مالی گردد.

به هنگام خرید سهام یک شرکت ارزشمند بودن آن شرکت را از دو منظر می‌توان مورد بررسی قرار داد:

- از منظر اطلاعات بنیادی شرکت

- از منظر اطلاعات تکنیکال شرکت

در هنگام بررسی اطلاعات بنیادی شرکت بهره‌گیری از سایت‌هایی همچو `tsetmc` و هم چنین فیپ ایران می‌تواند کمک کننده باشد. و اطلاعات تکنیکال شرکت را با استفاده از فرمول‌ها و متدهایی ریاضی می‌توان بدست آورد. از آنجایی که خوانش اطلاعات و `scrap` این دیتا‌ها از سایت‌های مذکور برای اعمال تحلیل بنیادی، توسط سمت سرور صورت گرفت و هم چنین اعمال روابط ریاضی بررسی تکنیکال، بر روی قیمت سهام نیز از جمله وظایف سمت سرور بود، جزئیات دقیق‌تر این موضوع خارج از بحث اصلی، و آموزش‌های فراگرفته شده در این دوره‌ی کارآموزی می‌باشد.

۲-۳-گام دوم، درک ارتباط سرور و کلاینت:

پس از آنکه با برخی از مفاهیم بازارهای مالی آشنایی اولیه صورت گرفت و درک مورد نیاز برای تدوین نرم افزار اندروید فراهم گردید، مرحله‌ی توسعه‌ی نرم افزار آغاز شد.

با توجه به این امر که بازار های مالی، بازار هایی حساس و آسیب پذیر هستند و هرگونه نقص امنیتی منجر به از دست رفتن سرمایه‌ی عظیمی می‌شود، لذا لازم بود فراتر از مکانیزم‌های عادی توسعه‌ی نرم افزار، محصول پیش روید. زین سبب بهره‌مندی از زبان برنامه نویسی سالیدیتی و ایجاد سرور در بستر بلاکچین از نقاط مورد توجه این محصول می‌باشد. سالیدیتی زبان برنامه نویسی است که برای ایجاد قرار داده‌های هوشمند اتریوم مورد استفاده قرار می‌گیرد، با بهره‌گیری از این زبان می‌توان به توسعه‌ی نرم افزار‌های غیر مت مرکزی یا DApp پرداخت. نکته‌ی مهم و قابل توجه در این نرم افزار‌ها آن است که با توجه به توزیع نرم افزار بر روی چندین سرور، پایداری و امنیت آن‌ها به مراتب بیشتر از سایر متدهای توسعه‌ی نرم افزار است. پس از آشنایی اولیه با تکنولوژی مورد استفاده در سمت سرور، به صورت مفصل، بر روی هدف کارآموزی یعنی فرآگیری توسعه‌ی نرم افزار‌های اندرویدی پرداخته شد.

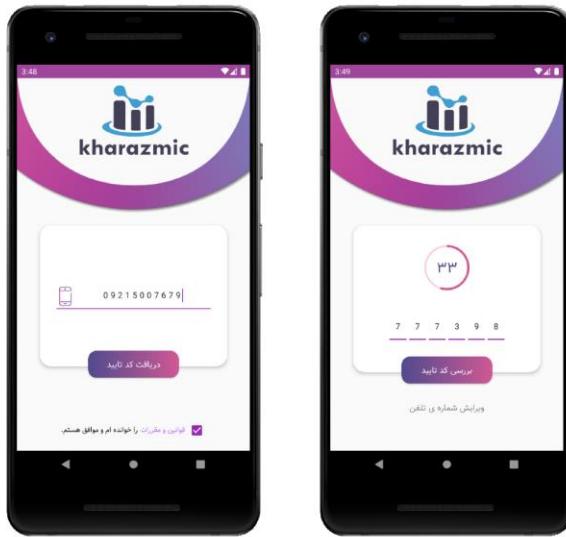
۴-۲- گام سوم، شروع توسعه‌ی نرم افزار:

۱- پیاده‌سازی صفحات لاغین و خانه:

در نرم افزار‌های اندرویدی، لایه‌ها و رابط کاربری در قالب فایل‌های XML پیاده‌سازی می‌شوند. لذا در گام اول پس از فرآگیری نحوه‌ی ایجاد اشکال اولیه و ابتدایی نظیر دایره، مستطیل و حلقه، بر اساس فایل تصویری که تیم گرافیست شرکت در اختیار تیم برنامه نویس قرار داده‌اند، اقدام به طراحی صفحات نرم افزار، در قالب کد‌های XML می‌شود.

بدون شک، از مهم‌ترین قسمت‌های یک نرم افزار، صفحه‌ی احراز هویت کاربر می‌باشد. بسیاری از قابلیت‌های نرم افزار در گروی احراز هویت صحیح می‌باشد. مکانیزم‌های امنیتی زیادی وجود دارند که اخیراً مکانیزم رمز عبور یک‌بار مصرف مورد توجه بسیاری قرار گرفته است. رمز عبور یک‌بار مصرف یا به اختصار OTP مکانیزمی امنیتی است که در آن کاربر به ازای هر بار ورود از سرور رمز عبوری تصادفی دریافت می‌کند و با این رمز می‌تواند فرآیند احراز هویت خود را انجام دهد. در این پروژه در گام اول، از کاربر شماره‌ی موبایل وی دریافت گردید و سپس سرور کدی ۶ رقمی به صورت تصادفی ایجاد می‌کند و برای موبایل کاربر پیامک می‌شود. چنانچه کاربر در نرم افزار، در قسمت مشخص شده، کد پیامک شده را به درستی وارد کند، توکنی که به کاربر اختصاص پیدا کرده است به عنوان ریسپانس دریافت می‌گردد و بدین گونه کاربر می‌تواند وارد نرم افزار شود. در فرآیند احراز هویت کاربر از

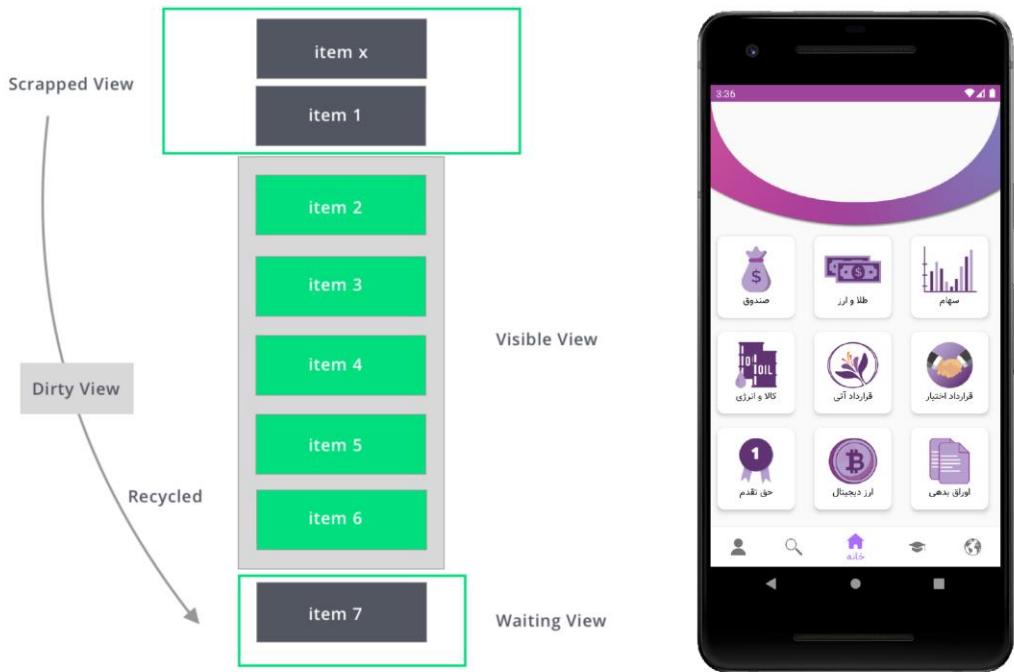
فرمت JSON استفاده گردید. پس از آنکه توکن اختصاصی کاربر دریافت شد، وی به صفحه‌ی اصلی نرم افزار هدایت می‌شود. هم چنین توکن در فایل XML مرسوم به Sharedpreferences که می‌توان آنرا به عنوان یک پایگاه داده‌ی key-value نیز تلقی کرد، ذخیره می‌شود. این کار برای این صورت میگیرد که هر بار نیاز به ایجاد ارتباط به سرور وجود داشت، این توکن در هدر بسته‌ها، باید ارسال شود.



تصویر(۱-۲) سمت چپ، صفحه‌ی ورود شماره موبایل
سمت راست، محل وارد کردن کد تایید پیامک شده

بعد از آنکه فرآیند احراز هویت کاربر و ذخیره سازی توکن دریافت شده از سمت سرور با موفقیت صورت گرفت، می‌بایست کاربر به صفحه‌ی اصلی هدایت شود.

با توجه به اینکه در طراحی صفحه‌ی اصلی، از View‌های تکراری استفاده شده است، به منظور افزایش عملکرد نرم افزار از یک View خاص به نام RecyclerView استفاده گردید. این المان که می‌توان از آن به یک المان فوق العاده کاربردی و هوشمند نام برد، بدین شکل عمل میکند که یک لایه دریافت نموده و بعد از آن اطلاعات درون یک آرایه را به ترتیب درون این لایه قرار میدهد. نکته‌ی لازم به ذکر که منجر به افزایش عملکرد نرم افزار نسبت به طراحی تک تک لایه‌ها می‌شود آن است که این المان، به جای ایجاد کردن لایه‌های جدید به تعداد اعضای آرایه، لایه‌ها را بازیافت می‌کند! بدین معنی که اگر لایه‌ای از صفحه‌ی نمایش خارج شود، به جای آنکه آنرا حذف کند و برای المان دیگری که به صفحه‌ی نمایش وارد می‌شود یک مورد جدید بسازد، صرفاً دیتا‌های موجود در آن لایه را با دیتا‌های جدید آرایه جایگزین می‌کند و به جای ایجاد لایه‌ی جدید، از همان لایه‌ی ساخته شده بهره میگیرد.



تصویر (۲-۲) سمت راست، UI مربوط به صفحه‌ی اصلی نرم افزار

تصویر سمت چپ، نحوه‌ی عملکرد المان RecyclerView

۲- اتصال پایگاه داده و کد زنی منطق صفحه‌ی پروفایل:

در توسعه‌ی نرم افزار بهره‌گیری الگوهای طراحی یا Design Pattern‌ها از همیت ویژه‌ای برخوردار است. چرا که در این الگوهای روش‌هایی بهینه و مناسب برای پاسخ‌دهی به مشکلات نرم افزاری است. یکی از این الگوهای نرم افزاری، الگوی طراحی Singleton است. به زبان ساده، با بهره‌گیری از این الگو این اطمینان به شما داشته می‌شود که از یک کلاس، تنها یک نمونه در کل نرم افزار وجود داشته باشد و در صورتی که در قسمت‌های مختلف نرم افزار دسترسی به این کلاس نیاز باشد، همان نمونه‌ی اولیه‌ی ساخته شده، در اختیار همه‌ی بخش‌ها قرار می‌گیرد. این الگوی طراحی دقیقاً همان چیزی است که برای ایجاد پایگاه داده در نرم افزارهای اندرویدی بدان نیازمندیم. چرا که ما مایل هستیم در برنامه‌ی اندروید صرفاً یک پایگاه داده ساخته شود و بعد از آن در همان پایگاه داده تمامی اطلاعات قرار گیرد.

نرم افزارهای اندرویدی از پایگاه داده‌ی SQLite که یک پایگاه داده‌ی رابطه‌ای است استفاده می‌کنند. در سال‌های اخیر گوگل برای آسان نمودن شیوه‌ی استفاده از این پایگاه داده توسط برنامه‌نویسان اندروید، کتابخانه‌ای

تحت عنوان Room را معرفی نموده است که با استفاده از این کتابخانه بدون آنکه در گیر کانفیگ های پیچیده برای دسترسی به پایگاه داده شویم، به سادگی می توانیم از Sqlite نمونه ای ایجاد نموده و اطلاعات مورد نیاز خود را در آن ذخیره کنیم. Room در اصل همچو یک ORM عمل نموده و دسترسی به پایگاه داده و عملیات کوئری زدن را بسیار ساده تر کرده است.

پس از آنکه بر اساس موارد نوشته شده در اسپرینت، و هماهنگی با تیم Backend مشخص شد که کاربر دارای چه ویژگی هایی است، کد های مربوط به ایجاد جدول اطلاعات کاربر و کوئری های مربوط به درج و واکشی این اطلاعات زده شد.

یکی از بحث های مورد توجه و جدید در توسعه‌ی نرم افزار های اندروید استفاده از المان های اندروید یا به عبارت دیگر Android Architecture Components AAC می باشد. از مهم ترین موارد AAC می توان به Live Data اشاره کرد. در اصل یک دیتا کلاسی است که نسبت به تغییرات آگاه است. بدین معنی که اگر نمونه ای از آن ساخته شود، چنانچه تغییری در محتوای آن صورت بگیرد، متوجه این تغییر می شود.

این ویژگی دقیقا همان چیزی است که در هنگام خواندن و ویرایش اطلاعات در داخل پایگاه داده بدان نیازمندیم. برای آنکه موضوع قابل درک تر شود به ارائه مثالی می پردازم:

فرض کنید صفحه‌ای داریم که به محض ورود کاربر، اطلاعات وی را از سرور دریافت می کند و در پایگاه داده ذخیره می کند و به او نشان می دهد. هم چنین نرم افزار ما شامل صفحه‌ی دیگری نیز می باشد که در آن کاربر می تواند مشخصات خود را ویرایش کند. اگر ما از live data استفاده نکنیم، چنانچه که کاربر اطلاعات خود را ویرایش کرد و در سمت سرور ثبت شد، لازم است وقتی به صفحه‌ی نمایش مشخصات کاربر بر میگردیم، مجدداً به سرور متصل شویم و اطلاعات بروز شده را دریافت و در پایگاه داده ذخیره کیم. حال فرض کنید در چندین بخش از نرم افزار برخی از مشخصات کاربر مورد استفاده قرار گیرند و تمام این صفحات قبل از بار اطلاعات را دریافت کرده‌اند. از آنجا که هیچ کلاسی آگاه به تغییر وجود ندارد، هیچ کدام از صفحات دیتا‌های جدید را بارگذاری نمی کنند و داده‌های قدیمی که قبل از پایگاه داده دریافت نموده‌اند را لود می کنند.

برای رفع این مشکل دو راه حل وجود دارد، یا هر بار که کاربر اطلاعاتی را ویرایش نمود، تمامی صفحاتی که به گونه‌ای، از برخی اطلاعات کاربر استفاده می کنند Reload شوند و یا از Live data استفاده شود.

از آنجایی که Live data نسبت به تغییرات آگاه است، کافی است خروجی کوئری های پایگاه داده‌ی خود را از

جنس Live data تعریف کنیم. در این صورت چنانچه کاربر اطلاعات خود را ویرایش نماید، چون محتوای موجود در Live data تغییر کرده است و این کلاس از تغییرات آگاه است، در کلیه‌ی قسمت‌های نرم افزار که این اطلاعات به نحوی استفاده شده است، تغییرات اعمال می‌شود و نیاز به بارگذاری مجدد کل صفحه نمی‌باشد.

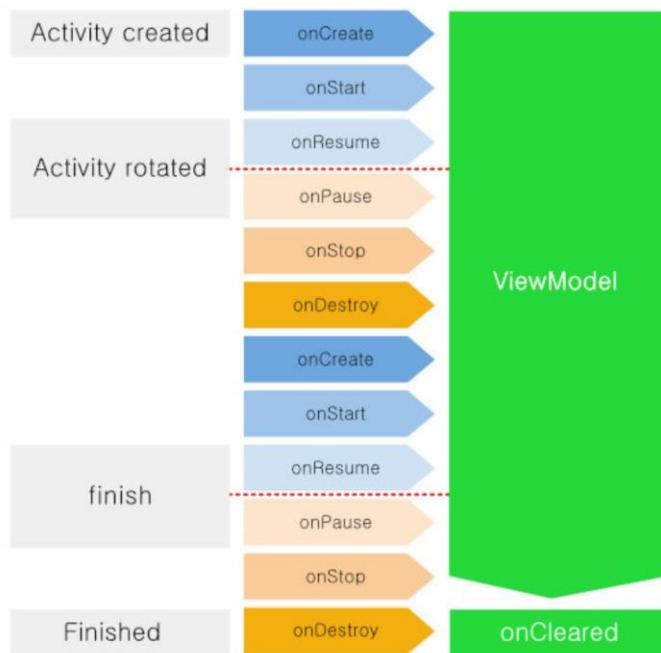
از دیگر بحث‌های مهم موجود در توسعه‌ی نرم افزار انتخاب معماری نرم افزار است. در گذشته برای نرم افزار‌های اندرویدی از معماری MVP استفاده می‌شد اما چندین سال است که گوگل به صورت رسمی استفاده از معماری نرم افزار MVVM را مطرح کرده است. این معماری که مخفف کلمه‌ی Model - View - View Model است، به خوبی با سایر AAC‌ها منطبق می‌شود و از آنها بهره می‌برد. برای پیاده‌سازی این معماری لازم است هر ویژگی نرم افزار را به چند کلاس تقسیم نمود:

یک لایه مربوط به واسط کاربری که به آن View گفته می‌شود، یک کلاس مربوط به ادغام منطق و لاجیک نرم افزار و المان‌های بصری که به آن Model گویند و یک کلاس دیگر برای ایجاد لاجیک و قسمت‌های تصمیم گیرنده‌ی نرم افزار که به آن View Model گویند. در نرم افزار‌های اندرویدی، کلاس Model برابر با همان مفهوم Activity می‌باشد.

مهم ترین ویژگی استفاده از این معماری در کلاس مربوط به View Model نهفته است! در نرم افزار‌های اندروید هر کلاس چرخه‌ی حیات مخصوص به خود را دارد که بر اساس رفتار کاربر ممکن است خاتمه یابند و یا ایجاد شوند. نکته‌ی حائز اهمیت این است که View Model نسبت به تغییرات منزه است، بدین معنا که حیات آن صرفاً وابسته‌ی مرحله‌ی ساخته شدن و یا تخریب کلاس Model وابسته است و به سایر چرخه‌های حیات کلاس Model نیست. برای درک بهتر اجازه دهید با بیان یک مثال کاربری که در پیاده‌سازی صفحه‌ی پروفایل از این روش استفاده شده است، به شرح موضوع پردازم:

در صفحه‌ی پروفایل لازم است اطلاعاتی از پایگاه داده دریافت گردد و در کلاس Model این اطلاعات با لایه‌ی بصری نرم افزار یعنی View ادغام شود. تصویر کاربر در جای مناسب نشان داده شود و مشخصات کاربر در قسمت‌های مشخص شده نوشته شوند. اگر به صورت عادی و بدون استفاده از معماری MVVM اقدام به انجام این کار کنیم مجبور هستیم منطق نرم افزار یعنی دریافت اطلاعات از پایگاه داده و خوانش اطلاعات را در کلاس Model انجام دهیم. نکته‌ای که وجود دارد بر اساس چرخه‌ی حیات این کلاس، چنانچه کاربر گوشی خود را از حالت عمودی به

افقی بچرخاند یا چنانچه از صفحه‌ی کنونی به صفحه‌ی دیگری رود (جایه جایی Fragment مدنظر می‌باشد) و مجدداً به همین صفحه بازگردد، کل کلاس Model بازسازی شده، اطلاعات مجدداً از پایگاه داده دریافت می‌شوند و سپس با لایه‌ی View ادغام می‌شوند. حال آنکه فرض کنید به جای وجود پایگاه داده‌ی محلی و آفلاین، لازم بود اطلاعات از سرور دریافت شود. بدون شک این حالت باعث ایجاد پردازش زائد در نرم افزار و در نتیجه کاهش عملکرد نرم افزار خواهد شد. حال آنکه با بهره‌گیری از معماری MVVM و انتقال فرآیند خوانش اطلاعات از پایگاه داده (و یا دریافت اطلاعات از سرور)، انجام فرآیند های محاسباتی و ... در کلاسی موسوم به ViewModel نیاز به بازخوانی این اطلاعات و لود زائد بر روی نرم افزار نخواهد بود. چرا که چرخه‌ی حیات این کلاس صرفاً وابسته به مرحله‌ی ایجاد و تخریب کلاس Model است.



تصویر(۲-۳) نشان دهنده‌ی چرخه‌ی حیات Activity و چرخه‌ی حیات کلاس ViewModel

۳- آشنایی با الگوی طراحی Repository و اتصال به سرور:

از دیگر الگوهای طراحی نرم افزار، الگوی مخزن یا Repository می‌باشد. به بیان ساده با استفاده از این الگوی طراحی، لایه‌ی منطق، نیازی به دانستن در مورد چگونگی تامین دیتا‌های مورد نیازش ندارد و صرفاً دیتا‌ها را از کلاس مشخصی دریافت می‌کند. این الگوی طراحی در بسیاری از قسمت‌های نرم افزار مورد استفاده قرار گرفت. برای مثال در طراحی صفحه‌ی پروفایل، کلاس ViewModel مربوط به نمایش اطلاعات صفحه‌ی پروفایل، اطلاعات را از کلاسی تحت عنوان ProfileRepository دریافت می‌کند. کلاس ViewModel اطلاعی از آن ندارد که این اطلاعاتی که دریافت کرده است را از پایگاه داده‌ی محلی دریافت نموده و یا اینکه این اطلاعات از سرور دریافت شده‌اند، وی تنها بر روی وظیفه‌ی خود متمرکز است. در کلاس ProfileRepository، بررسی می‌شود که آیا اطلاعات کاربر قبلاً در دیتابیس قرار دارند و به روز هستند یا خیر، چنانچه که پاسخ مثبت باشد، دیتا‌های موجود را برمی‌گرداند و چنانچه که اطلاعات کاربر در دسترس نباشد، آن را از سرور درخواست کرده و در پایگاه داده‌ی محلی ذخیره می‌کند. استفاده از این الگوی طراحی علاوه بر بهینه کردن عملکرد نرم افزار و جلوگیری از ایجاد Redundancy، باعث می‌شود که مشکل زدایی و عیب‌یابی نرم افزار نیز تصویریح یابد.

برای ایجاد چنین الگوی طراحی نیازمند آن هستیم که از چندین کلاس نمونه بسازیم و در قسمت‌های مختلف نرم افزار از این کلاس‌ها استفاده کنیم. ایجاد نمونه از چند کلاس و استفاده از قسمت‌های مختلف نرم افزار منجر به کاهش وابستگی کلاس‌ها به یک دیگر می‌شود. این عامل سبب آن است که چنانچه در یکی از این کلاس‌ها تغییری صورت گیرد نیازمند صرف زمان زیادی برای اعمال این تغییر در کلیه‌ی کلاس‌هایی باید که به نحوی از این کلاس استفاده کرده‌اند. برای مثال فرض کنید در وهله‌ی اول سازنده‌ی کلاسی پارامتری را دریافت نمی‌کند اما در ادامه‌ی کار، سازنده تغییر می‌کند و نیازمند دریافت پارامتری به عنوان ورودی است. در این حالت باید در تمام کلاس‌هایی که از کلاس مذکور نمونه ای ایجاد شده‌است، پارامتر مورد نیاز سازنده را به آن ارجاع دهیم. این مشکل سبب به وجود آمدن مفهومی تحت عنوان تزریق وابستگی یا Dependency Injection که به اختصار به آن DI نیز می‌گویند، شد. تزریق وابستگی منجر به سهولت در نمونه سازی از کلاس‌ها و استفاده از آن‌ها در سایر کلاس‌ها می‌شود. یکی از بهترین ابزارهایی که برای این کار در اختیار برنامه نویسان اندروید قرار گرفته است، Dagger نام دارد که یک چهارچوب نرم افزاری برای تزریق وابستگی حساب می‌شود. هم اینکه این چهارچوب توسط تیم گوگل در حال توسعه می‌باشد. شما با استفاده از این چهارچوب، یک بار وابستگی‌ها را تعریف می‌کنید و

بعد از آن در هر قسمت از نرم افزار که نیازمند آن بودید که از کلاسی نمونه گیری کنید، کافی است این وابستگی را تزریق کنید. این کار با بهره گیری از **Annotation** ها صورت میگیرد.

پس از آنکه الگوی طراحی مخزن ایجاد شد و هم چنین وابستگی های مورد نیاز برای پیاده سازی این الگو، از طریق **چهار چوب Dagger** بدان تزریق شد، لازم است ارتباط کلاینت و سرور برقرار شود.

در نرم افزار های اندرویدی چندین راه برای ارتباط با سرور وجود دارد. یکی از مهم ترین و بهترین راه های ارتباط با سرور استفاده از کتابخانه **Volley** است که توسط گوگل توسعه پیدا کرده است. با استفاده از این کتابخانه می توان تصویر، رشته و یا قالب های جیسون از سرور دریافت نمود. منظور از قالب های جیسون **JsonObject** و یا **JSONArray** می باشد. از ویژگی های مهم این کتابخانه نسبت به سایر کتابخانه ها، سیستم کش آن می باشد.

با توجه به اینکه در هنگام دریافت اطلاعات کاربر، تنها یک مورد مشخصات دریافت می شود، لذا برای ارتباط نیازمند ایجاد **JsonObjectRequest** هستیم. با استفاده از کتابخانه **JSON** که توسط خود گوگل توسعه پیدا کرده است، **JsonObject** دریافتی از سرور را به جدول اطلاعات کاربر که قبلا آن را ایجاد کردیم، **Map** می کنیم. سپس با استفاده از کوئری نوشته شده برای درج اطلاعات، این اطلاعات را درون پایگاه داده ذخیره می کنیم.

همانگونه که قبلا ذکر شد، چون نوع خروجی جدول را از نوع **Live data** تعریف کردیم، پس از اینکه رکورد جدیدی در پایگاه داده درج شد، اطلاعات کاربر در تمامی کلاس هایی که به نحوی از این اطلاعات استفاده کرده بودند بروزرسانی می شود و یکپارچگی نرم افزار نیز حفظ می شود.

۴ - پیاده سازی UI مربوط به صفحه **i** پروفایل

در قسمت های قبل با نحوه ای استفاده از معماری نرم افزار **MVVM** و هم چنین الگوی طراحی **Repository** آشنا شدیم. هم چنینی فرا گرفته شد که در نرم افزار های اندرویدی از چه پایگاه داده ای استفاده کنیم و چگونه از این پایگاه داده بهره ببریم. پس از آنکه منطق صفحه **i** پروفایل پیاده سازی شد، لازم به آن است که **UI** این صفحه نیز بر اساس طرحی که تیم گرافیست به تیم توسعه دهنده دادند، پیاده سازی شود.

نکته ای که در طراحی این صفحه وجود دارد لایه ای جدید با نام **MotionLayout** است که از سال ۲۰۱۸ توسط

گوگل معرفی گردید. با استفاده از این لایه می توان نسبت به ری اکشن های کاربر مثل اسکرول کردن، جیدمان صفحه را تغییر داد.

هم چنین در صفحه‌ی پروفایل نیاز است که تصویر کاربر نشان داده شود. برای آنکه بتوان تصاویر کاربر را نشان داد دو استراتژی ممکن از پیش گرفته شود:

۱) دانلود تصویر از سرور، ذخیره‌ی آن در گوشی کاربر و سپس نمایش آن در محل مورد نظر

۲) نمایش تصویر به صورت آنلاین

از آنجایی که کتابخانه‌های بسیار قدرتمندی برای نمایش تصاویر آنلاین وجود دارد و این کتابخانه‌ها امکان کش این تصاویر را می‌دهند، و همچنین عملیات خواندن و نوشتمن بر روی حافظه‌ی گوشی کاربر نیازمند دریافت دسترسی از کاربر است، ترجیح بر استفاده از روش دوم صورت گرفت.

کتابخانه‌ای که برای این کار مورد استفاده قرار گرفت، کتابخانه‌ی Glide می‌باشد که از کتابخانه‌های معروف در این زمینه است. با استفاده از این کتابخانه و با بهره‌گیری از سرویس کش قدرتمند آن، می‌توان تصاویر را حتی بدون نیاز به اینترنت، پس از یک بار بارگذاری، نمایش داد.

برای اینکه بتوان از این کتابخانه در کنار معماری MVVM بهره برد، می‌بایست از قابلیت زیان برنامه نویسی کاتلین تحت عنوان Extension Function استفاده کرد. با استفاده از این ویژگی، برای المان تصویر (Image View) خود، تابعی تعریف می‌کنیم که آدرس عکس را دریافت کند و پس از دریافت، با استفاده از کتابخانه‌ی Glide عکس را از آدرس دریافت شده، بارگذاری نموده و در المان تصویر مذکور نشان می‌دهیم.



تصویر(۴-۲) سمت چپ، حالت اولیه‌ی صفحه‌ی پروفایل

سمت راست، حالتی که کاربر صفحه را به سمت بالا اسکرول نماید

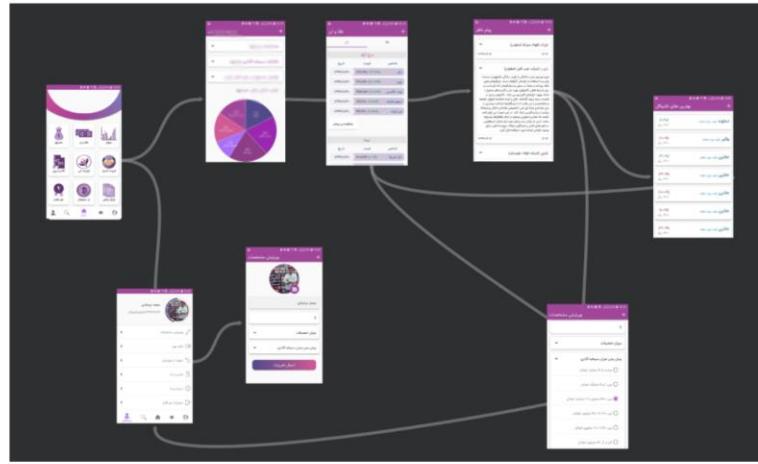
از دیگر المان‌هایی که باید بدان در این وله توجه نمود، پیمایش بین صفحات می‌باشد. به محض اینکه کاربر وارد نرم افزار می‌شود اولین صفحه‌ای که مشاهده می‌کند، صفحه‌ی خانه، یعنی صفحه‌ی اصلی نرم افزار می‌باشد. حال آنکه لازم است کاربر بتواند بین صفحات حرکت کند و از صفحه‌ی خانه به صفحه‌ی پروفایل و از صفحه‌ی پروفایل، فرضاً، به صفحه‌ی ویرایش مشخصات رود. برای این کار روش‌های مختلفی وجود دارد. در گذشته، با استفاده از کلاسی تحت عنوان `Fragment Manager` این کار صورت می‌گرفت. اما هم اینک کامپوننت دیگری به AAC‌ها اضافه شده است که با عنوان `Navigation Component` شناخته می‌شود. با استفاده از این کامپوننت شما می‌توانید بین صفحات (`Fragments`) جا به جا شوید. استفاده از این کامپوننت به منظور پیاده‌سازی الگوی طراحی موسوم به `SingleActivity` می‌باشد که هم اکتون در نرم افزاری همچو اینستاگرام از آن بهره برده شده است. این

الگویان می کند که در نرم افزار های اندرویدی، لازم است که تنها یه صفحه ای اصلی (Activity) ایجاد شود و بعد از آن برای ایجاد سایر صفحات، والد هایی برای این صفحه در نظر گرفته شود (Fragment).



تصویر(۵-۲) نمایی از صفحه ای ویرایش مشخصات

در Navigation Component برای تعریف مسیر جریان نرم افزار و ارتباط صفحات به یک دیگر از فایل XML موسوم به Navigation Graph استفاده می کنیم. به کمک این گراف، ما Fragment ها را به یک دیگر متصل میکنیم و امكان حرکت کاربر از یک صفحه به صفحه ای دیگر را فراهم می سازیم. هم چنین با استفاده از قابلیت هایی که Navigation Graph در اختیار برنامه نویس قرار میدهد، وی می تواند اینیمیشن هایی را به هنگام انتقال از یک Fragment به Fragment دیگر تنظیم کند. همچنین می توان چیدمان استک صفحات برنامه به هنگام فشار دادن دکمه ای بازگشت گوشی، را تنظیم نمود. در زیر نمونه ای از گراف مسیر را می توانید مشاهده کنید که به علت سیاست های حفظ حقوق معنوی نرم افزار، برخی از قسمت ها محو شده اند.



تصویر(۶-۲) گراف مسیر رسم شده برای قسمتی از نرم افزار

۵ - استفاده از سرویس Firebase

سرویس Firebase از معروف ترین و پرکاربرد ترین سرویس های موجود برای توسعه دهنده کان اندروید می باشد. این سرویس که توسط خود گوگل عرضه شده است خدمات بسیاری را در اختیار توسعه دهنده کان اندروید و حتی سایر پلتفرم ها قرار میدهد که بسیاری از این خدمات رایگان و برخی نیز پولی هستند. از جمله سرویس هایی که توسط Firebase ارائه می شود می توان به موارد زیر اشاره کرد:

- سرویس ارسال نوتیفیکشن یا FCM

- سرویس دریافت آمار نرم افزار

- سرویس بررسی میزان سرعت و عملکرد نرم افزار

- سرویس گزارش کرش و خطایابی نرم افزار

- سرویس تست نرم افزار

به غیر از موارد فوق، خدمات دیگری نیز توسط این سایت ارائه می شود که در طول دوره ی کارآموزی، به علت کم کاربرد و یا غیر رایگان بودن آنها، فراگرفته نشد.

۱ - سرویس ارسال نوتیفیکشن یا FCM:

در اصل کلمه‌ی FCM مخفف کلمه‌ی Firebase Cloud Messaging می‌باشد و به معنای ارسال پوش اعلان است. بدون شک در انواع نرم افزار‌ها مشاهده کرده اید که برخی از اطلاعیه‌های مهم در قالب یک اعلان، توسط نرم افزار به شما نشان داده می‌شود. فایربیس این امکان را به شما می‌دهد که برای همه یا دسته‌ای از کاربران خود اعلان ارسال کنید. هم چنین، می‌توانید با بهره‌گیری از کتابخانه‌های وسیعی که برای این سرویس نوشته شده است، سرویس ارسال اعلان را به سرور خود متصل نموده و چنانچه تغییری در پایگاه داده‌ی شما رخ داد با تنظیم کردن یک عدد Trigger، این تغییر را به کاربر گوش زد کنید. اگر بخواهیم مثالی عینی که در این پروژه استفاده شد را بیان کنم، در صورتی که کاربر درخواست فعال سازی اعلان را داده باشد، تغییرات قیمت دلار به وی به صورت Notification نشان داده می‌شود.

از امکانات خوبی که در کنار ارسال اعلان شما می‌توانید از آن استفاده کنید، بررسی آمار تعامل کاربر با آن اعلان است. به عبارت دیگر شما می‌توانید متوجه شوید که این اعلان را چند نفر از کاربران شما دریافت کرده و دیده اند و چند نفر از کاربران شما، بر روی اعلان کلیک کرده و توسط آن وارد نرم افزار شده اند.

Notification	Status	Platform	Start / Send	End	Sends	Opens
▶ ...	✓ Completed	...	Aug 30, 2020 9:22 PM	-	<1000	25%
▶ ...	✓ Completed	...	Aug 26, 2020 12:12 PM	-	<1000	67%
▶ ...	✓ Completed	...	Aug 26, 2020 12:11 PM	-	<1000	33%

تصویر (۷-۲) نمایی از پنل ارسال اعلان سرویس Firebase

۲ - سرویس دریافت آمار نرم افزار:

به نظر شخص بنده، از بهترین و مهم ترین خدماتی که توسط Firebase به شما ارائه می‌شود، بخش تحلیلات آماری آن است. با استفاده از این بخش شما می‌توانید میزان کاربران جدید در طول دوره‌های زمانی مختلف، میزان حضور کاربران در نرم افزار، تعداد کل کاربران نرم افزار، رده‌ی سنی کاربران، جنسیت و حتی موقعیت جغرافیایی کاربران را بدست آورید.

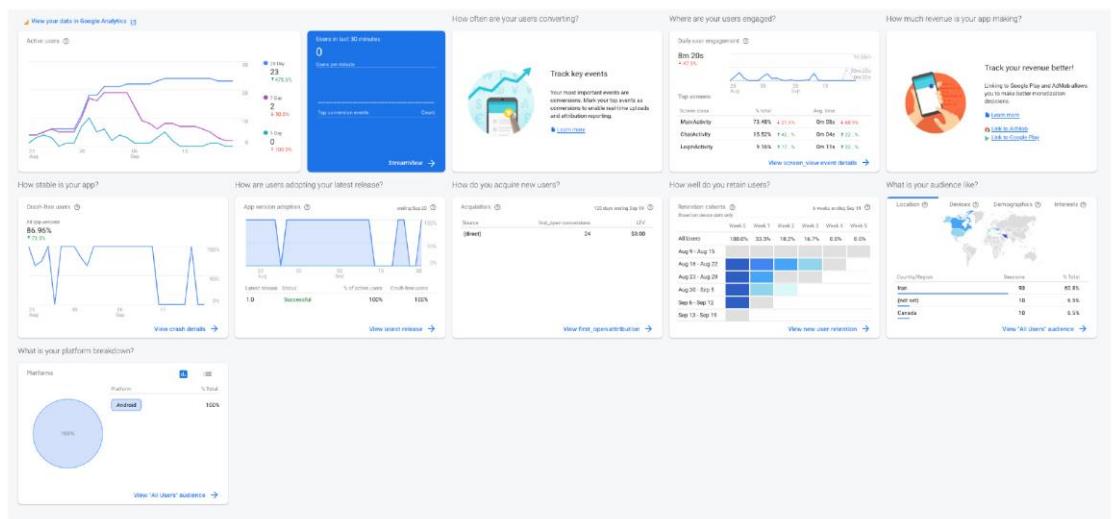
شما می‌توانید به صورت زنده اطلاعات، کاربرانی که هم اکنون در حال استفاده از نرم افزار هستند را بر روی نقشه‌ی گوگل مشاهده کنید (این نقشه به دلایل امنیتی، تا میزان شهر، بزرگ نمایی می‌شود و موقعیت دقیق و محل زندگی

کاربران را نمی توان توسط آن بدست آورد(۱)

هم چنین این بخش با در دست داشتن سوابق گذشته‌ی نرم افزار، میزان تعامل و جذب کاربر را در دوره‌ی مشابه قبل نشان می‌دهد. به عبارت دیگر، بیان می‌کند که در این ماه میزان کاربران چه میزان رشد و یا کاهش داشته است.

هم چنین با استفاده از این قسمت شما می‌توانید بفهمید کاربران برای استفاده از نرم افزار شما از کدام نسخه‌ی اندروید استفاده می‌کنند.

هم چنین چنانچه که نرم افزار شما، به صورت غیر رایگان عرضه شود و برای دریافت وجه از سرویس‌های گوگل پلی بهره‌گیرید، شما می‌توانید اطلاعاتی مالی نیز بدست آورید.



تصویر(۲-۸) داشبورد مربوط به آمار کاربران نرم افزار در سرویس Firebase

۳ - سرویس بررسی میزان عملکرد نرم افزار:

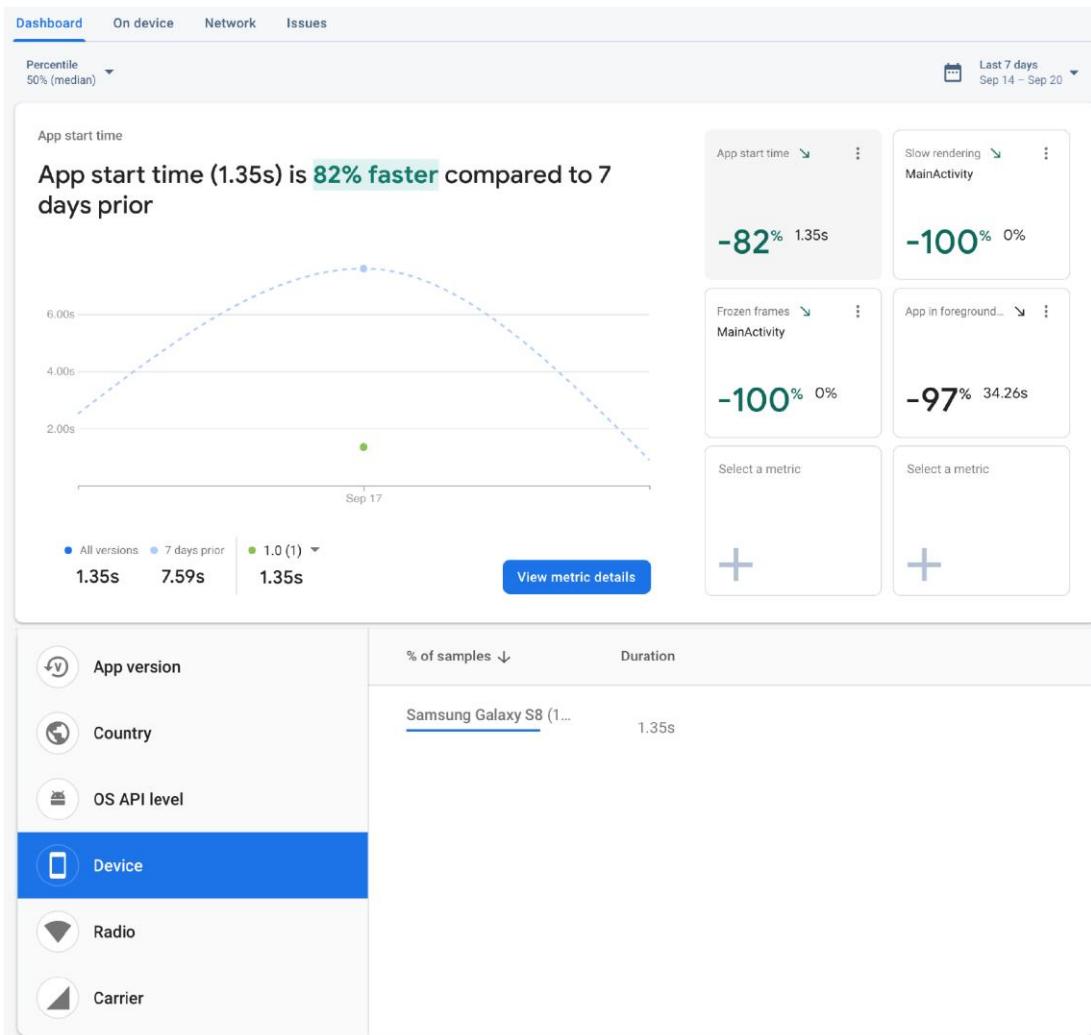
نرم افزار‌های اندرویدی بسیار به وضعیت فیزیکی گوشی کاربر وابسته‌اند، ممکن است یک نرم افزار بسیار روان و با عملکردی کاملاً مناسب بر روی یک گوشی اجرا شود و در گوشی دیگری، دارای هنگی‌های فراوان باشد.

با استفاده از این سرویس شما می‌توانید عملکرد نرم افزار خود را بررسی نمایید و از خوب و بهینه کار کردن نرم افزار بر روی گوشی تمام کاربران آگاه شوید.

این سرویس به شما این امکان را می‌دهد که میزان موفقیت آمیز بودن ارتباطات شبکه، اجرای مناسب و بدون وقفه‌ی نرم افزار، اجرا شدن سریع نرم افزار و هم چنین بررسی عملکرد نرم افزار هنگامی که در پس زمینه در حال اجرا

است را مورد ارزیابی قرار دهید.

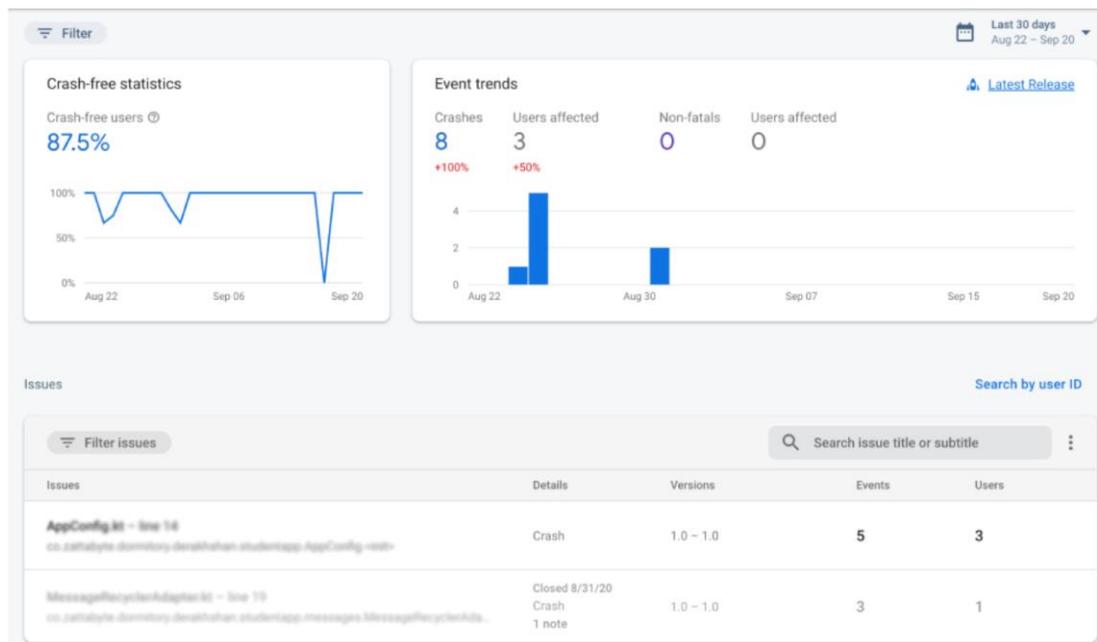
در صورتی که نرم افزار بر روی دستگاهی به درسی اجرا نشود و عملکرد ضعیفی داشته باشد، این سرویس به شما این امکان را می دهد که از اپراتور شبکه، نسخه‌ی اندروید نرم افزار، مدل دقیق گوشی، موقعیت جغرافیایی کاربر و همچنین نوع اتصال (داده تلفن همراه یا وای‌فای) مطلع شوید.



تصویر(۲-۹)نمایی از صفحه‌ی تحلیل عملکرد نرم افزار در سایت Firebase

۴ - سرویس گزارش کوش و خطایابی نرم افزار:

بدون شک توسعه‌ی یک نرم افزار بدون وجود هیچ گونه خطا و ایراد امکان پذیر نیست. برنامه نویسان تا حد بسیار زیادی سعی بر آن دارند که قبل از منتشر کردن نرم افزار تمامی مشکلات و نقصان‌های نرم افزار را بر طرف نمایند. اما این حقیقت بر کسی پوشیده نیست که باز هم ممکن است در طول فرآیند استفاده از نرم افزار، مشکلی به وجود آید. خیلی از کاربران به خصوص کاربران نرم افزار‌های اندروید، به محض مشاهده‌ی خطا، به جای ارسال گزارش خطأ به تیم برنامه نویس، نرم افزار را حذف کرده و نرم افزار مشابه دیگری را نصب می‌کند. همین عامل اهمیت عاری بودن از خطا در نرم افزار‌های اندرویدی را چندین برابر کرده است. از مهم ترین خدماتی که شما توسط سایت **FireBase** به آن دسترسی دارید، سیستم گزارش خطأ آن است. با استفاده از این سرویس هنگامی که نرم افزار کوش کند، به صورت کامل، علت کوش، مشخصات فیزیکی گوشی کاربر، ساعت و تاریخ کوش و موقعیت جغرافیایی کاربر را به شما گزارش می‌دهد. شما با مشاهده‌ی علت کوش به صورت کامل می‌توانید متوجه شوید که کاربر در چه وضعیتی بوده است که چنین خطایی رخداده و این دقیقاً همان چیزی است که هر برنامه نویس اندروید بدان نیازمند است.



تصویر(۱۰-۲) نمایی از صفحه‌ی گزارش خطأ

۵- تست نرم افزار:

یکی از دیگر خدماتی که گوگل از طریق این سایت به شما ارائه می دهد، امکان اجرا و تست نرم افزار بر روی گوشی های مختلف است. همان طور که قبل نیز ذکر شد، صحت عملکرد نرم افزار های اندرویدی به شدت به نوع گوشی کاربر و مشخصات فیزیکی آن وابسته است. از آنجایی که امکان تست نرم افزار بر روی دسته‌ی وسیعی از گوشی های مختلف برای خیلی از افراد امکان پذیر نیست، گوگل اقدام به ارائه‌ی این سرویس نموده است. با استفاده از این سرویس در طرح رایگان، شما می توانید حداقل از ۱۰ گوشی مجازی و ۵ گوشی واقعی را بهره‌گیرید و نرم افزار خود را بر روی آن تست کنید. این گوشی ها بسیار قابلیت شخصی سازی دارند و هر نسخه‌ای از اندروید که مایل باشید را می توانید بر روی این گوشی ها تست کنید. هم چنانچه که شما خود برای نرم افزار تست ننوشته باشید، گوگل این امکان را به شما می هید که از ربات تست نرم افزار استفاده کنید. این ربات در اصل قطعه کدی است که نرم افزار شما تحت آن اجرا می شود و تمامی مسیر های نرم افزار برای آن بررسی میگردد بدین صورت که این ربات سعی در پیدا کردن دکمه ها و ورودی ها در لایه های نرم افزار نموده و با وارد کردن مقدار های مختلف و تصادفی، عملکرد نرم افزار شما را بررسی می کند.

در پایان نیز گزارش جامعی از فرآیند تست، کلیی کوتاه از مسیر اجرای نرم افزار به شما ارائه می شود.

پیوست ها

بخشی از کدهای زده شده در طول پروژه

کد مربوط به UI صفحه اصلی نرم افزار:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <layout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools">
5
6      <data>
7
8      </data>
9
10
11     <androidx.constraintlayout.widget.ConstraintLayout
12         android:layout_width="match_parent"
13         android:layout_height="match_parent">
14
15         <androidx.core.widget.NestedScrollView
16             android:layout_width="match_parent"
17             android:layout_height="0dp"
18             app:layout_constraintBottom_toTopOf="@+id/bottom_menu"
19             app:layout_constraintTop_toTopOf="parent">
20
21             <LinearLayout
22                 android:layout_width="match_parent"
23                 android:layout_height="wrap_content"
24                 android:orientation="vertical">
25
26
27                 <ImageView
28                     android:id="@+id/top_banner"
29                     android:layout_width="match_parent"
30                     android:layout_height="wrap_content"
31                     android:scaleType="fitXY"
32                     android:src="@mipmap/login_top_banner"
33                     app:layout_constraintTop_toTopOf="parent" />
34
35
36                 <androidx.recyclerview.widget.RecyclerView
37                     android:id="@+id/recyclerView"
38                     android:layout_width="match_parent"
39                     android:layout_height="wrap_content"
40                     android:overScrollMode="never" />
41
42             </LinearLayout>
43         </androidx.core.widget.NestedScrollView>
44
45     </androidx.constraintlayout.widget.ConstraintLayout>
46 </layout>
```

کد مربوط به ایجاد پایگاه داده بر اساس الگوی Singleton

```

package com.kharazmic.app.database

import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase
import com.kharazmic.app.database.model.*

@Database(
    entities = [UserInfoModel::class, SignalsModel::class, SearchHistoryModel::class, BestStockModel::class],
    version = 7,
    exportSchema = false
)
abstract class MyDatabase : RoomDatabase() {

    abstract val userDao: UserDao
    abstract val signalDao: SignalDAO
    abstract val searchHistoryDao: SearchHistoryDAO
    abstract val bestStockDao: BestSignalDAO

    companion object {

        @Volatile
        private var INSTANCE: MyDatabase? = null

        fun getInstance(context: Context): MyDatabase {
            synchronized(this) {
                var instance = INSTANCE

                if (instance == null) {
                    instance = Room.databaseBuilder(
                        context.applicationContext,
                        MyDatabase::class.java,
                        "database"
                    )
                        .fallbackToDestructiveMigration()
                        .build()
                    INSTANCE = instance
                }
                return instance
            }
        }
    }
}

```

کد مربوط به UI صفحه‌ی پروفایل و استفاده از MotionLayout

```

<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <data>

        <variable
            name="viewModel"
            type="com.kharazmic.app.main.profile.ProfileViewModel" />
    </data>

    <androidx.constraintlayout.motion.widget.MotionLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layoutDirection="rtl"
        app:layoutDescription="@xml/profile_motion">

        <RelativeLayout
            android:id="@+id/avatarHolder"
            android:layout_width="140dp"
            android:layout_height="140dp"
            android:layout_marginTop="10dp"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent">

            <com.mikhaelopez.circularprogressbar.CircularProgressBar
                android:id="@+id/remainingSubscription"
                android:layout_width="140dp"
                android:layout_height="140dp"
                app:cpb_background_progressbar_color="@color/light_pink"
                app:cpb_progressbar_color="@color/dark_purple" />

            <de.hdodenhof.circleimageview.CircleImageView
                android:id="@+id/avatar"
                loadImage="@{viewModel.image}"
                android:layout_width="120dp"
                android:layout_height="120dp"
                android:layout_centerInParent="true"
                android:scaleType="centerCrop" />

        </RelativeLayout>
    
```

```
<TextView
    android:id="@+id/name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{viewModel.name}"
    android:textAppearance="@style/vazir_font" />

<TextView
    android:id="@+id/expire"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{viewModel.subscription}"
    android:textAppearance="@style/vazir_font"
    android:textColor="@color/gray" />

<ImageView
    android:id="@+id/profileBanner"
    android:layout_width="match_parent"
    android:layout_height="170dp"
    android:scaleType="fitXY"
    android:src="@mipmap/profile_top_banner" />

<LinearLayout
    android:id="@+id/content"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:elevation="10dp"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/profileBanner">

    <androidx.recyclerview.widget.RecyclerView
        android:overScrollMode="never"
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>

</androidx.constraintlayout.motion.widget.MotionLayout>

</layout>
```

کد مربوط به بخشی از کلاس ViewModel صفحه‌ی پروفایل:

```

class ProfileViewModel(val context: Context, private val database: MyDatabase) : ViewModel() {

    private val viewModelJob = Job()
    private val scope = CoroutineScope(Dispatchers.Main + viewModelJob)
    val name = MutableLiveData<String]()
    val subscription = MutableLiveData<String }()
    private val followers = MutableLiveData<String>()
    private val following = MutableLiveData<String>()
    val signals = MutableLiveData<String>()
    val image = MutableLiveData<String>()
    lateinit var isTokenExpired: TokenExpired

    fun getUserInfo() {
        val request = object : JsonObjectRequest(Method.GET, Address().userInfoAPI, null,
            Response.Listener {
                scope.launch {
                    async(Dispatchers.IO, CoroutineStart.DEFAULT, block = {
                        Utils(context).profileID = it.optString("_id")
                        val userInformation =
                            UserInfoModel(
                                name = Arrange().persianConverter(it.optString("name")),
                                image = Address().base + it.optJSONObject("avatar")
                                    .getString("url"),
                                education = it.optInt("education"),
                                experience = it.optInt("experience"),
                                followers = Arrange().persianConverter(it.optString("followers")),
                                following = Arrange().persianConverter(it.optString("following")),
                                maxDays = it.optInt("maxDays").toFloat(),
                                remainingDays = it.optInt("remainingDays").toFloat(),
                                signals = Arrange().persianConverter(it.optString("signals")),
                                subscription = Arrange().persianConcatenate(
                                    middle = "انجام اقتصادی",
                                    first = it.optString("subscription")
                                ),
                                worth = it.optInt("worth")
                            )
                        database.userDAO.add(userInformation)
                    }).await()
                }
            }, Response.ErrorListener {
                try {
                    if (it.networkResponse.statusCode == 401)
                        isTokenExpired.expire(true)
                    else
                        Log.i(
                            "Log",
                            "error in ProfileViewModel ${String(
                                it.networkResponse.data,
                               Charsets.UTF_8
                            )}"
                        )
                } catch (e: Exception) {
                    Log.i("Log", "error in ProfileViewModel $it")
                }
            })
        }) {
            @Throws(AuthFailureError::class)
            override fun getHeaders(): MutableMap<String, String> {
                val params = HashMap<String, String }()
                params["Authorization"] = "Bearer ${Utils(context = context).token}"
                params["Accept"] = "Application/json"
                return params
            }
        }

        request.retryPolicy =
            DefaultRetryPolicy(10000, 5, DefaultRetryPolicy.DEFAULT_BACKOFF_MULT)

        Volley.newRequestQueue(context).add(request)
    }
}

```