



دانشگاه پیام نور استان تهران
مرکز / واحد مرکز تهران شمال
گروه مهندسی کامپیوتر و فناوری اطلاعات

پروژه کارشناسی
رشته‌ی مهندسی کامپیوتر
گرایش نرم افزار

عنوان پروژه:
ساخت اپلیکیشن آشپزی
اندرویدی

استاد راهنما:
جناب دکتر سید علی رضوی ابراهیمی
تهیه کننده:
محسن گلی
تیرماه ۱۴۰۰

بِحَمْدِ اللَّهِ الْعَلِيمِ الرَّحِيمِ



دانشگاه پیام نور استان تهران
مرکز / واحد مرکز تهران شمال
گروه مهندسی کامپیوتر و فناوری اطلاعات

پروژه کارشناسی
رشته‌ی مهندسی کامپیوتر
گرایش

عنوان پروژه:
ساخت اپلیکیشن آشپزی
اندرویدی

استاد راهنما:
جناب دکتر سید علی رضوی ابراهیمی
تهیه کننده:
محسن گلی
تیرماه ۱۴۰۰

کلیه حقوق مادی مترتب بر نتایج مطالعات، ابتکارات
و نوآوری های ناشی از این پروژه متعلق به :
"دانشگاه پیام نور استان تهران/مرکز تهران شمال "
می باشد.

ساخت اپلیکیشن آشپزی

اندرویدی

استاد راهنما: دکتر سید علی رضوی ابراهیمی

نام و نام خانوادگی: محسن گلی

شماره دانشجویی: ۹۷۰۱۶۱۴۶۷

ترم تابستان سال تحصیل ۱۴۰۰ - ۹۹

تاریخ شروع: ۱۴۰۰/۹/۲۰

تاریخ پایان: ۱۴۰۰/۱۲/۲۰

چکیده

در عصر توسعه فناوری، تصور انجام فعالیت یا استراحت بدون استفاده از گجت‌ها (Gadget)^۱ غیرممکن به نظر می‌رسد. به لطف دستگاه‌های سیار، کاربران می‌توانند در هر زمانی بدون در نظر گرفتن موقعیت مکانی‌شان، انواع امور را با استفاده از **وب جهان گستر**^۲ اجرا کنند. برنامه‌های اینترنتی به گونه‌ای طراحی شده‌اند که دسترسی کاربران به عناصری که برای راه‌اندازی یک کسب‌وکار نیاز دارند یا ابزارهایی که در زندگی روزمره به کار می‌برند را به راحتی فراهم کنند.

دسترسی به اینترنت پایدار، دامنه وسیع خدمات و راحتی استفاده، معیارهایی هستند که حتی کاربران مشکل‌پسند، دستگاه‌های اندرویدی یا IOS^۳ را جذب می‌کنند. برنامه‌های موضوع‌محور، طوری مهندسی می‌شوند که انتظار کاربران در رده‌های سنی متفاوت را برآورده سازند. **رابط کاربری**^۴ نقش مهمی را بازی می‌کند. هر روز توسعه‌دهندگان برنامه‌های موبایل با بهینه‌سازی محصولات‌شان مواجه می‌باشند. این موضوع نه تنها به افزایش کیفیت و اجرای بهتر برنامه‌ها کمک می‌کند بلکه باعث افزایش رقابت در بازارهای کسب‌وکار فناوری نیز می‌شود. تعادل میان کیفیت و قیمت محصولات، تقاضای اصلی در بازار را می‌سازد.

ساختن یک برنامه‌ی آشپزی، بدان معنی است که قبل از هر چیز، کاربران امکان انتخاب دستور تهیه غذا و حتی امکان نحوه آماده‌سازی به روش خود را داشته باشند. **بانک اطلاعاتی**^۵ یک برنامه، می‌تواند شامل هزاران دستور تهیه از سراسر دنیا باشد. شما به راحتی می‌توانید طرز تهیه‌ای شامل مواد مناسب را یافته و ایده خود را تحقق بخشید. استفاده از برنامه به عنوان کتاب آشپزی این امکان را که دستورهای تهیه مورد علاقه را جهت استفاده بعدی مشخص نمایید به راحتی فراهم می‌کند. محاسبه کالری هم به این وسیله ممکن می‌گردد.

قبل از مشخص نمودن مراحل ساخت و طراحی یک برنامه آشپزی تحت بستر اندروید، سوالاتی می‌بایست پاسخ داده شوند: چطور شما می‌توانید یک برنامه دستور پخت را به بهترین وجه بسازید؟ چه نکاتی در طی ساخت برنامه بایستی در نظر گرفته شود؟ چه عواملی می‌تواند باعث نقص طراحی گردد؟

بسیار ارزش‌مند است تا با انواع برنامه‌های موجود در این زمینه نیز آشنا شویم. پنج نوع برنامه برای آشپزی وجود دارد که عبارتند از:

- مجموعه‌ای از دستورهای تهیه
- برنامه‌های آشپزی با قابلیت شخصی‌سازی
- برنامه‌های رسانه اجتماعی
- کتاب‌های مرجع مواد لازم برای پخت

^۱ ابزارک

^۲ World Wide Web

^۳ Iphone Operating System

^۴ User Interface

^۵ Data Base

فهرست مطالب

عنوان.....	صفحه.....
چکیده.....	یک.....
فهرست مطالب.....	دو.....
فصل اول.....	۱.....
مقدمه.....	۱.....
مقدمه.....	۲.....
۱-۱ معرفی پروژه.....	۳.....
۲-۱ ساختار گزارش.....	۳.....
فصل دوم.....	۴.....
مقایسه انواع برنامه های رایج.....	۴.....
مقدمه.....	۵.....
۲-۲ مجموعه ای از دستورهای تهیه.....	۵.....
۲-۲ برنامه های آشپزی با قابلیت شخصی سازی.....	۵.....
۳-۲ برنامه های رسانه اجتماعی.....	۶.....
۴-۲ کتاب های مرجع مواد لازم برای پخت.....	۶.....
۵-۲ ابزارهای اندازه گیری.....	۷.....
۶-۲ جمع بندی.....	۷.....
فصل سوم.....	۸.....
طراحی و پیاده سازی.....	۸.....
مقدمه.....	۹.....
۱-۳ ساخت BottomNavigation، FrameLayout و SelectorColor.....	۱۰.....
۲-۳ TabLayout ViewPager، Adapter ViewPager و Custom Shape.....	۱۷.....

۳-۳ پایگاه داده Sqlite.....	۲۰
۳-۴ کتابخانه‌ی پیکاسو و Show Item.....	۲۳
۳-۵ ساخت صفحه جزئیات هر Item.....	۲۵
۳-۶ قسمت جستجوی هر Item.....	۲۸
۳-۷ اضافه کردن Item به قسمت علاقه‌مندی.....	۲۹
۳-۸ صفحات SplashScreen و Login.....	۳۰
۳-۹ جمع بندی.....	۳۲
فصل چهارم.....	۳۳
نتایج پروژه.....	۳۳
مقدمه.....	۳۴
۴-۱ چرا برنامه‌های آشنی بسیار محبوب هستند؟.....	۳۵
۴-۲ جمع بندی.....	۳۵
فصل پنجم.....	۳۶
جمع بندی و پیشنهادها.....	۳۶
مقدمه.....	۳۷
۵-۱ چه نوع برنامه‌ای می‌خواهیم، بسازیم؟.....	۳۷
۵-۲ ویژگی‌هایی که برنامه‌های آشنی باید داشته باشند.....	۳۷
۵-۲-۱ ویژگی‌های کاربری.....	۳۷
۵-۲-۲ ویژگی‌های مدیریتی.....	۳۸
مراجع.....	۴۰
واژه‌نامه.....	۴۱
واژه‌نامه فارسی به انگلیسی.....	۴۲
واژه‌نامه انگلیسی به فارسی.....	۴۳

فهرست اشکال

- شکل ۱-۳ ساخت navigationbottom در فایل activity\main.xml ۱۰
- شکل ۲-۳ ساخت فایل Android Resource File و اختصاص نام menu ۱۰
- شکل ۳-۳ افزودن item های person, search, home در فایل menu\bottom.xml ۱۱
- شکل ۴-۳ ساخت icon برای هر سه item در پوشه drawable ۱۱
- شکل ۵-۳ استفاده از ویجت FrameLayout در فایل activity\main.xml ۱۲
- شکل ۶-۳ مرتب کردن پروژه براساس Directory ۱۲
- شکل ۷-۳ استفاده از تابع onCreate در SearchFragment.kt ۱۳
- شکل ۸-۳ ساخت Directory برای اختصاص Selector به Fragment ها ۱۳
- شکل ۹-۳ مقداردهی به Selector فایل color\selector\bottom.xml ۱۴
- شکل ۱۰-۳ اضافه کردن دو ویژگی در bottomNavigation و مقداردهی به آنها ۱۴
- شکل ۱۱-۳ افزودن کدهای هر Bottom برای رفتن به Search, Home و Person Fragment ها ۱۵
- شکل ۱۲-۳ خروجی برنامه و طراحی BottomNavigation در اندروید استودیو ۱۵
- شکل ۱۳-۳ ورود به هر Fragment های دکمه های خانه، جستجو و شخصی ۱۶
- شکل ۱۴-۳ افزودن RelativeLayout برای ساخت Action Bar ۱۶
- شکل ۱۵-۳ اضافه نمودن TextView برای نمایش متن Fragment ها در نوار ابزار ۱۷
- شکل ۱۶-۳ textCategory برای findView شدن Textview ۱۷
- شکل ۱۷-۳ تنظیم fragment\home و نمایش Fragment ۱۸
- شکل ۱۸-۳ استفاده از تابع cast() در فایل HomeFragment.kt ۱۸
- شکل ۱۹-۳ ایجاد پوشه tablayout و ایجاد Fragment های Lunch, Dinner, Cake, Brunch ۱۹
- شکل ۲۰-۳ اضافه نمودن Fragmen ها ساخته شده به HomeFragment.kt و تنظیم ViewPager ۱۹
- شکل ۲۱-۳ خروجی برنامه پس از ساختن Fragment های صبحانه، ناهار، شام و دسر ۲۰
- شکل ۲۲-۳ ایجاد فایل selector\tab\layout.xml ۲۰
- شکل ۲۳-۳ ایجاد و مقداردهی به رکوردها در پایگاه داده Sqlite ۲۱
- شکل ۲۴-۳ ایجاد فیلدها با استفاده از زبانه Browse Data ۲۱
- شکل ۲۵-۳ افزودن اطلاعات به فیلدهای پایگاه داده ۲۲
- شکل ۲۶-۳ ایجاد کلاس DataBase در DataBase Package ۲۲
- شکل ۲۷-۳ وارد نمودن مقادیر مربوط به هر متغیر در فایل Info\db ۲۳
- شکل ۲۸-۳ استفاده از Database در فایل SearchFragment.kt ۲۳
- شکل ۲۹-۳ افزودن کتابخانه Picasso به فایل build.gradle ۲۳
- شکل ۳۰-۳ وارد کردن کد در فایل AdapterFood.kt برای استفاده از کتابخانه Picasso ۲۴
- شکل ۳۱-۳ نمایش Item ها بعد از افزودن کتابخانه Picasso ۲۴
- شکل ۳۲-۳ دادن Permission برای دسترسی به اینترنت ۲۵
- شکل ۳۳-۳ دریافت تصاویر بعد از دادن Permission اینترنت به برنامه ۲۵
- شکل ۳۴-۳ افزودن ImageView برای ساخت نوار ابزار ۲۶

شکل ۳-۳۵	ساختن شکل Icon علاقه‌مندی در نوار ابزار صفحه جزئیات هر Item	۲۶
شکل ۳-۳۶	activity\detail.xml بعد از افزودن TextView و ImageView	۲۷
شکل ۳-۳۷	وارد کردن TextView و ImageView در فایل MainActivity.kt	۲۷
شکل ۳-۳۸	شکل نهایی صفحه جزئیات در برنامه	۲۸
شکل ۳-۳۹	شکل نهایی قسمت جستجو پس از اجرای برنامه	۲۹
شکل ۳-۴۰	افزودن توابع getStatus, getCategory, getValue, getFav	۲۹
شکل ۳-۴۱	اضافه شدن موارد انتخاب شده به قسمت علاقه‌مندی‌های صفحه شخصی	۳۰
شکل ۳-۴۲	صفحه اول SplashScreen برای خوش آمدگویی	۳۱
شکل ۳-۴۳	صفحه خوش آمدگویی و صفحه ورود برنامه	۳۱

فهرست جدول ها

.....

فهرست علائم اختصاری

IDE: Integrated Development Enviroment

محیط یکپارچه توسعه

IOS: Iphone Operating System

سیستم عامل آی فون

JVM: Java Virtual Machine

ماشین مجازی جاوا

WWW: World Wide Web

وب جهان گستر

فصل اول

مقدمه

مقدمه

پس از اتمام دوره‌های آموزشی و مباحث نظری و همچنین آشنایی با مفاهیم و محیط‌های برنامه‌نویسی، انجام پروژه‌های مختلف جهت تکمیل این فرآیند و توسعه مهارت برنامه‌نویسی از اهمیت خاصی برخوردار است. به هر موضوعی که علاقه‌مند هستید، آن را می‌توان در قالب یک برنامه^۱ قابل نصب در هر نوع بستری^۲ تبدیل کرد. **گوشی‌های هوشمند**^۳ یکی از این بسترها می‌باشند که امکان نصب و استفاده از این گونه برنامه‌ها را ممکن می‌سازند.

یکی از موضوعات برای تبدیل به برنامه‌ای قابل دسترس بر روی گوشی هوشمند که در این گزارش نیز به آن پرداخته شده، برنامه آموزش آشپزی می‌باشد. دسترسی سریع و آسان به فهرستی از غذاها و دستور تهیه آن‌ها به صورت متنی و همراه با تصویر از مزیت‌های این چنین برنامه‌هایی می‌باشد. به وسیله این برنامه ساده می‌توانید برای هر وعده غذایی، انتخابی داشته باشید و از طریق دستور تهیه، به راحتی آن را آماده کنید.

در این پروژه از زبان برنامه‌نویسی کاتلین^۴ استفاده گردیده است که منبع باز^۵ بوده و قابل اجرا بر روی ماشین مجازی جاوا^۶ (JVM) می‌باشد. این زبان در سال ۲۰۱۷ توسط شرکت گوگل به عنوان یک زبان رسمی توسعه اندروید معرفی گردید. ساختار آن بسیار شبیه به زبان جاوا بوده و یکی از ویژگی‌های زبان کاتلین استفاده از تمام کتابخانه‌ها و چارچوب‌های **نرم‌افزاری**^۷ زبان جاوا است. تمرکز کاتلین بر خوانایی و قابل فهم بودن کدها بوده، که موجب ساده‌تر شدن بازبینی آن‌ها گردیده است.

برای ساخت این برنامه از محیط برنامه‌نویسی اندروید استودیو استفاده گردیده است که یکی از IDE^۸های قابل استفاده برای این منظور می‌باشد.

^۱ Application

^۲ Platform

^۳ Smart Phone

^۴ Kotlin

^۵ Open Source

^۶ Java Virtual Machine

^۷ Framework

^۸ Integrated Development Enviroment

۱-۱ معرفی پروژه

با اجرای برنامه چه در محیط اندروید استودیو و چه بر روی گوشی هوشمند اندرویدی خود، در ابتدا با **صفحه آغازین**^۱ برنامه مواجه می گردید که برای چند ثانیه ای پیغام خوش آمدگویی به همراه تصویری را نمایش می دهد. در قسمت بعد وارد صفحه ورود کاربر شده که با وارد نمودن نام کاربری و شماره تلفن امکان رفتن به مرحله بعد ممکن می شود. صفحه بعدی از سه بخش^۲ شامل:

۱. خانه، برای نمایش تمامی موارد

۲. جستجو، جهت جستجوی موارد

۳. شخصی، برای نمایش فهرست علاقه مندی ها

تشکیل گردیده است.

صفحه خانهای برنامه از چهار بخش تشکیل گردیده است که شامل صبحانه، ناهار، شام و دسر می گردد. در این صفحه موارد مربوط به هر دسته بندی به صورت مجزا نمایش داده می شود. هر مورد شامل یک تصویر، نام و خلاصه ای از غذای مربوطه می گردد. برای دسترسی به جزئیات بیش تر هر مورد با کلیک بر روی آن، وارد صفحه جزئیاتش می شویم. صفحه جزئیات شامل بخشی برای نمایش نام و نوع آن، تصویر، مواد لازم و طرز تهیه این مورد می باشد که اگر متن آن از اندازه صفحه نمایش بیش تر بود با پیمایش امکان دیدن تمامی متن فراهم می گردد. هم چنین شامل بخشی برای افزودن این مورد در صورت علاقه به فهرست علاقه مندی ها می باشد.

در بخش جستجو، می توان هر کدام از موارد را با نام جستجو نمود و در بخش شخصی، علاوه بر نام کاربر و شماره تماس او در صورتی که موردی را قبلا به فهرست علاقه مندی ها اضافه نموده باشیم، آن را مشاهده می کنیم.

۲-۱ ساختار گزارش

فصل آخر نیز به جمع بندی کلی پروژه و پیشنهادهایی برای بهبود و توسعه آن مطرح می گردد.

^۱ Splash Screen

^۲ Fragment

فصل دوم

مقایسه انواع برنامه‌های رایج

مقدمه

آشپزی به عنوان یکی از رضایت‌بخش‌ترین فعالیت‌هایی است که واقعا مهارت‌های هنری ما را می‌سنبجند. از آن جایی که ما در هر کاری به برخی از راهنمایی و توصیه‌ها نیازمندیم، آشپزی نیز از این قاعده مستثنی نیست. این روزها، می‌توانیم با به کارگیری گوشی‌های هوشمند یا لپ‌تاپ برای هر سبک آشپزی یک دستور تهیه‌ای را بیابیم.

یک راهنمایی قدم به قدم، ما را آگاه می‌سازد که چه چیزی را درست یا نادرست انجام می‌دهیم. بنابراین برنامه‌های زیر برای کامل‌تر کردن و بهتر نمودن این مهارت بسیار مبتکرند. این برنامه‌ها راهنمای واضحی برای راحتی و مهیج کردن تجربه آشپزی‌تان فراهم می‌کنند.

همان‌طور که اشاره گردید، پنج نوع برنامه آشپزی وجود دارد:

- مجموعه‌ای از دستورهای تهیه
- برنامه‌های آشپزی با قابلیت شخصی‌سازی
- برنامه‌های رسانه اجتماعی
- کتاب‌های مرجع مواد لازم برای پخت
- ابزارهای اندازه‌گیری

در ادامه‌ی این فصل به معرفی و توضیح هریک از موارد بالا می‌پردازیم.

۲-۱ مجموعه‌ای از دستورهای تهیه

یکی از رایج‌ترین انواع برنامه‌ها که برای آشپزی ساخته می‌شوند، برنامه‌هایی هستند که شامل مجموعه‌ای از دستورالعمل‌ها می‌باشند. آن‌ها کتابخانه‌ای از دستورالعمل‌ها می‌باشند که طوری درست شده‌اند تا کاربر با استفاده از چند معیار مثل مواد تشکیل‌دهنده، وعده‌های غذایی و سایر گزینه‌ها امکان جستجو داشته باشد.

اطلاعات در دسته‌های مشخصی تفکیک شده‌اند و به همراه فیلم‌های آموزشی و دستورالعمل سرآشپزان، بخشی جدایی‌ناپذیر از این گونه برنامه‌ها می‌باشند. وقتی چنین برنامه‌ای می‌سازید بایستی به این نکات توجه خاصی نمود:

- ذخیره و بازنمایش فیلم‌های آموزشی
- به‌روز رسانی محتویات و نشان دادن اطلاعات
- فهرست غذاهای فصلی
- یکپارچه‌سازی با تقویم برای برنامه‌ریزی

۲-۲ برنامه‌های آشپزی با قابلیت شخصی‌سازی

این نوع از برنامه‌ها دارای کتابخانه‌ی مهمی از دستور تهیه نیستند. در عوض این فرصت را برای کاربران مهیا کرده‌اند تا به‌طور مستقل بتوانند دستور تهیه خود را ساخته و به صورت مجموعه درآورند. این نوع از برنامه معمولاً به صورت قدم به قدم دستور

تهیه را فراهم نموده و راهنمای آشپزی به کاربر هر مرحله را اطلاع می دهد. برنامه هایی از این دست امکان انتخاب دستورالعمل مشخص بر روی اینترنت و با امکان افزودن دستور تهیه به کتابخانه کاربر ساخته می شوند. راحتی کار با این برنامه ها به این گونه است که کاربر در زمان پخت قادر به قفل کردن صفحه نمایش می باشد. این ویژگی به کاربر در شرایطی که نیاز به روشن بودن مداوم صفحه نمایش گوشی دارد و برای جلوگیری از تغییر وضعیت نمایش و رفتن به ”حالت خواب“^۱ بسیار کمک می رساند.

در حین ساخت این گونه برنامه ها بایستی به الگوریتم پاسخ گو برای موارد زیر توجه خاصی شود :

- ساخت برنامه های تغذیه ای
- دریافت دستورالعمل از اینترنت
- تشخیص متن^۲ در زمان خواندن داده از صفحات اینترنتی

۲-۳ برنامه های رسانه اجتماعی

برنامه های آشپزی اجتماعی برای آن دسته از کاربران ساخته شده اند که در رسانه های اجتماعی فعال هستند. فضاهایی اینترنتی هستند که کاربران می توانند، عکس های دست آوردهای آشپزی شان را در آن با یک دیگر به اشتراک گذاشته و راجع به دستور تهیه های جالب شان به بحث بپردازند. برنامه های دارای طبیعت اجتماعی به کاربران، یک فرصت برای ساخت دست آوردهای وعده های غذایی جالب شان از صفر و بحث درباره همه ی مزایا و معایب غذاها و مواد تشکیل دهنده آنها می دهند. کاربر، فرصت دیدن فرآیند پخت وعده های غذایی به صورت **برخط**^۳ را داشته و هم چنین نشانه گذاری دستورالعمل مورد علاقه اش ممکن می گردد. وقتی این نوع برنامه ها را می سازیم، بایستی مهم ترین نقش را به جمع آوری و تقارن عکس های دستگاه های کاربران اختصاص داد.

۲-۴ کتاب های مرجع مواد لازم برای پخت

برنامه هایی هستند که از دامنه وسیعی از اطلاعات تشکیل شده اند، از ساده ترین مواد تشکیل دهنده شروع شده و با بخش های نادر و کم یاب که به صورت راهنماهای آشپزی ارائه می گردند به پایان می رسند. نوعی فرهنگ لغات، مناسب مواد تشکیل دهنده هستند که نه تنها برای افراد حرفه ای بلکه برای دوست داران غذاهای ساده نیز مفید می باشند. داده ها در قالب ویژگی های خاص بخش های (سبزیجات، میوه ها، ادویه جات، سس ها و سایر) ارائه می گردند. فرق جزئی طعم ها، روش های آماده سازی و این که منشا آنها کجاست، نشان داده می شود. این نوع از برنامه ها معمولا شامل یک تحلیل کننده درونی می باشند که به کاربر ارزش غذایی مواد تشکیل دهنده هر وعده را می تواند پیشنهاد دهد.

^۱ Sleep Mode

^۲ Text Recognition

^۳ Online

سازنده این گونه از برنامه‌ها می‌بایست به دو معیار توجه کند:

- تامین اطلاعات به جذاب‌ترین و مفیدترین روش
- ارائه به‌روز رسانی منظم محتوا

۲-۵ ابزارهای اندازه‌گیری

پنج‌مین و آخرین نوع برنامه‌های آشپزی، برنامه‌هایی با ابزارهایی ابداعی برای اندازه‌گیری درجین پخت می‌باشند. در واقع، ابزارهای آشپزخانه، ساده‌ترین برنامه‌های آشپزی هستند. این گونه نرم‌افزارها برای محاسبه حجم، دما یا وزن مواد تشکیل‌دهنده برحسب دستورالعمل پخت طراحی شده‌اند. برنامه، هم‌چنین می‌تواند به‌عنوان یک زمان‌سنج قابل حمل مورد استفاده قرار گیرد. این نوع نرم‌افزارها بلااستثنا، بسیار رایج و برای هرکسی سودمند می‌باشند. هم برای مبتدیانی که درحال آموزش مهارت‌های آشپزی و یادگیری مطالب پایه‌ای می‌باشند و هم برای افراد حرفه‌ای می‌خواهند محاسبات مواد تشکیل‌دهنده هر غذایی را هرچه سریع‌تر انجام دهند.

برنامه‌های آشپزی، برای رقابت در بازار می‌بایست کاربردی آسان و دارای رابطی مطابق با معیارهای روز باشد که به‌طور برابر در تمامی گجت‌ها قابل اجرا باشند.

۲-۶ جمع‌بندی

طراحی منحصر به فرد برنامه‌های آشپزی و فرصت‌های فراوان برای ساخت تنوع گسترده وعده‌های غذایی توجه هم‌تازه‌کاران و سرآشپزان حرفه‌ای را در گوشه و کنار دنیا جلب خواهد کرد. یک مرورگر ابداعی به کاربران اجازه می‌دهد تا علاقه‌مندی‌های خود را ذخیره کنند و نظرات‌شان را با خانواده، دوستان و همکاران‌شان به اشتراک بگذارند. به‌لطف قابلیت برنامه‌ریزی، کاربر می‌تواند فهرست غذایش را برای تمام خانواده تعمیم دهد. کتاب آشپزی هم‌راه با امکان یادداشت به کاربران امکان حرکت به سمت حرفه‌ای شدن را می‌دهد. پخت هر وعده به‌لطف یک برنامه آشپزی به واقعیت تبدیل شده است.

فصل سوم

طراحی و پیاده‌سازی

مقدمه

در این فصل شروع به ساخت یک برنامه اندرویدی با استفاده از زبان قدرت مند کاتلین می کنیم. می خواهیم یک برنامه دستور پخت غذا را ساخته، جایی که به توان دستور تهیه هر نوع غذایی و برای هر وعده ای را یافته و آن را آماده کرد.

برنامه پیش رو با دریافت نام کاربر و شماره تلفن وارد صفحه خانه شده که دارای سه بخش اصلی می باشد، به غیر از خانه بخش های جستجو و شخصی در این برنامه افزوده شده که هر کدام کاربردی دارند. در قسمت خانه که دارای نوار ابزاری در بالا برای نمایش هر وعده غذایی می باشد، لیستی از غذاها نیز نمایش داده می شود که با کلیک بر روی هر یک از وعده ها موارد مربوط به آن را برای ما فهرست می کند. هر یک از این موارد شامل تصویر و قسمتی برای نمایش نام غذا و مختصری از نحوه تهیه آن می باشد. با کلیک بر روی هر یک از این غذاها به صفحه دیگری رفته که شامل یک نوار ابزار در بالا است که نام وعده و غذا را نمایش می دهد و دکمه ای برای بازگشت به صفحه قبل و شکلی که با انتخاب آن، غذای مربوطه به لیست علاقه مندی ها افزوده شود، در آن تعبیه شده است و هم چنین دارای قسمت هایی برای مواد اولیه و طرز تهیه آن می باشد. قابل ذکر است تصاویر برنامه از کتابخانه ی Picasso بوده و از طریق اینترنت دریافت می شوند.

در صفحه جستجو، امکان جستجوی غذا براساس نام آن ممکن شده است. در صورتی که مورد یافت شود در پایین آن را مشاهده خواهیم نمود.

در قسمت شخصی، در صورتی که موردی را قبلا با کلیک بر روی شکل قلب به عنوان مورد علاقه انتخاب کرده باشید، آن مورد در این صفحه برای شما نمایش داده خواهد شد.

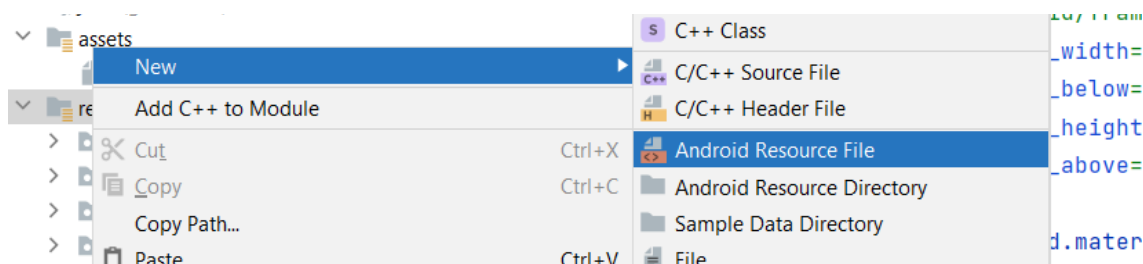
۱-۳ ساخت BottomNavigationView، FrameLayout و SelectorColor

صفحه اول برنامه که همان activity_main.xml می باشد که شامل یک BottomNavigationView، یک FrameLayout است که سه Fragment را با استفاده از این دو Widget نمایش می دهیم. عرض BottomNavigationView و ارتفاع آن را تعیین نموده و یک id نیز به آن اختصاص می دهیم. هم چنین برای آن menu، background و layout_alignParentBottom نیز در نظر می گیریم (شکل ۱-۳).

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_navigation"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:layout_alignParentBottom="true"
    android:background="@color/bg_widget"
    app:itemIconTint="@color/color_selector_bottom"
    app:itemTextColor="@color/color_selector_bottom"
    app:menu="@menu/menu_bottom" />
```

شکل ۱-۳ ساخت navigationbottom در فایل activity_main.xml

حال بر روی پوشه res برنامه کلیک راست کرده و بر روی گزینه Android Resource File کلیک می کنیم و از پنجره بعدی File Name را menu_bottom قرار داده و سپس Resource Type را Menu انتخاب می کنیم (شکل ۲-۳) تا هم مسیر و هم فایل مربوطه ساخته شود.



شکل ۲-۳ ساخت فایل Android Resource File و اختصاص نام menu

فایل menu_bottom.xml را باز کرده و سه item مورد نیاز home، search و person را به آن می افزاییم (شکل ۳-۳).

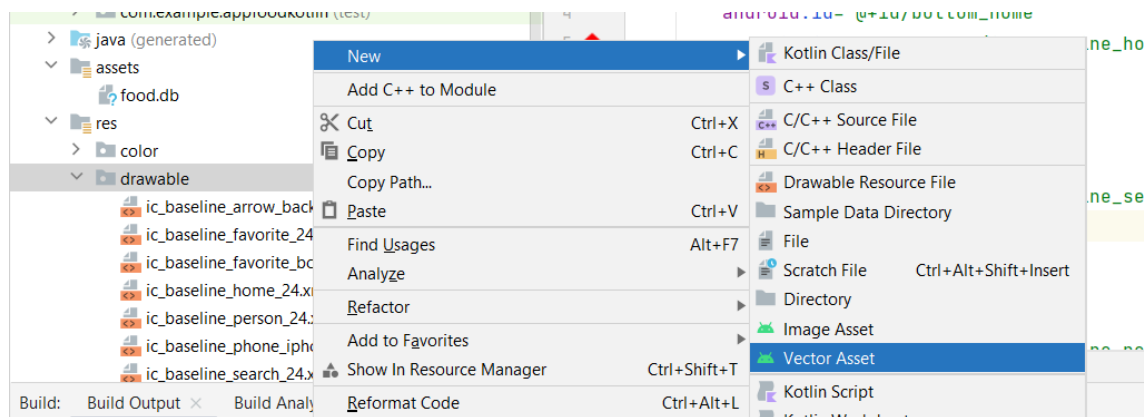
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3     <item
4         android:id="@+id/bottom_home"
5         android:icon="@drawable/ic_baseline_home_24"
6         android:title="خانه" />
7
8     <item
9         android:id="@+id/bottom_search"
10        android:icon="@drawable/ic_baseline_search_24"
11        android:title="جستجو" />
12
13    <item
14        android:id="@+id/bottom_person"
15        android:icon="@drawable/ic_baseline_person_24"
16        android:title="شخصی" />
17 </menu>

```

شکل ۳-۱ افزودن item های home search.person در فایل menu_bottom.xml

برای دسترسی به icon های هر سه item، به پوشه drawable که قبلا در مسیر پوشه res ساخته ایم رفته و با کلیک راست روی آن و انتخاب فایل New Vector Asset را می سازیم تا فایل برای ما Add شود (شکل ۴-۳).



شکل ۴-۳ ساخت icon برای هر سه item در پوشه drawable

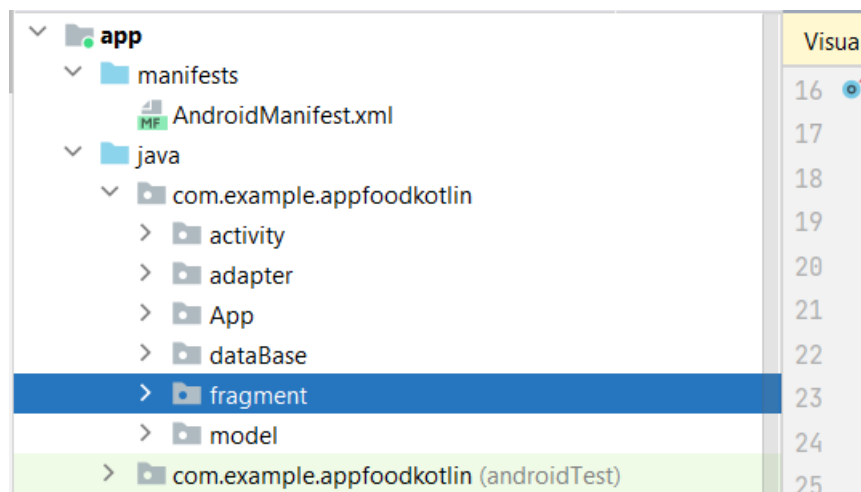
پس از این مرحله به فایل activity_main.xml رفته و در قسمت app:menu را مساوی @menu/menu_bottem قرار داده تا بتوانیم از این Icon ها استفاده کنیم.

برای نمایش FrameLayout از ویجت آن <FrameLayout/> استفاده نموده که دارای عرض و ارتفاع به مقدار match_parent می‌باشد تا تمام صفحه نمایش را پوشش دهد و در بالای Bottom Navigation قرار گیرد. با استفاده از Attribute^۱ به نام layout_above آنرا اعمال می‌کنیم (شکل ۳-۵).

```
<FrameLayout
    android:id="@+id/frame_layout"
    android:layout_width="match_parent"
    android:layout_below="@id/toolbar"
    android:layout_height="match_parent"
    android:layout_above="@id/bottom_navigation" />
```

شکل ۳-۵ استفاده از ویجت FrameLayout در فایل activity_main.xml

حال برای این که پروژه ما براساس Directory^۲، مرتب باشد، هر یک از کلاس‌های مربوط را در همان دایرکتوری قرار می‌دهیم (شکل ۳-۶).



شکل ۳-۶ مرتب کردن پروژه براساس Directory

با کلیک بر روی پوشه fragment، کلاس مربوط به Home، Search و Person را که دارای سازنده^۳ fragment می‌باشند را ایجاد می‌کنیم و از تابع^۴ onCreateView برای بارگذاری^۵ View استفاده می‌کنیم (شکل ۳-۷).

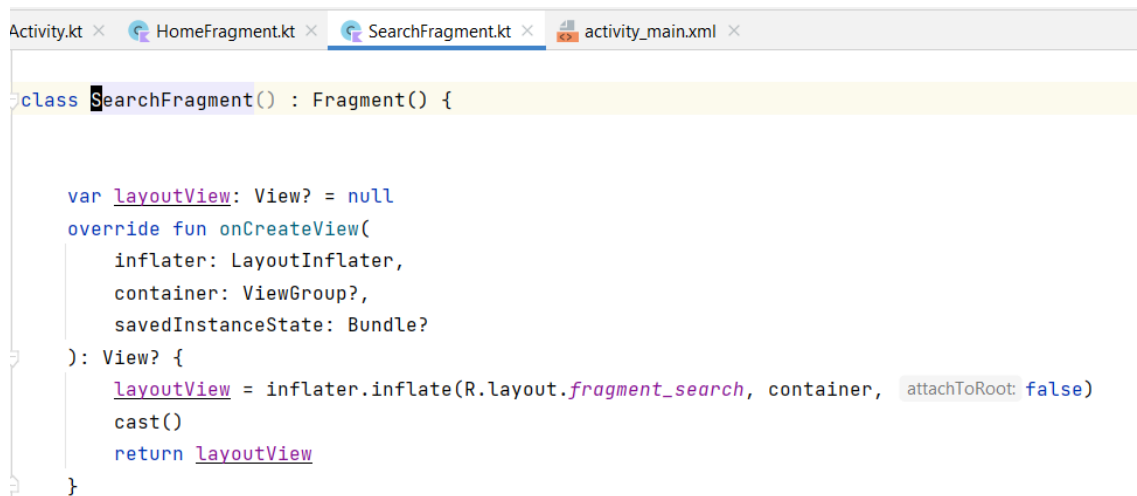
^۱ خاصیت

^۲ مسیر

^۳ Constructor

^۴ Function

^۵ Load



```

class SearchFragment() : Fragment() {

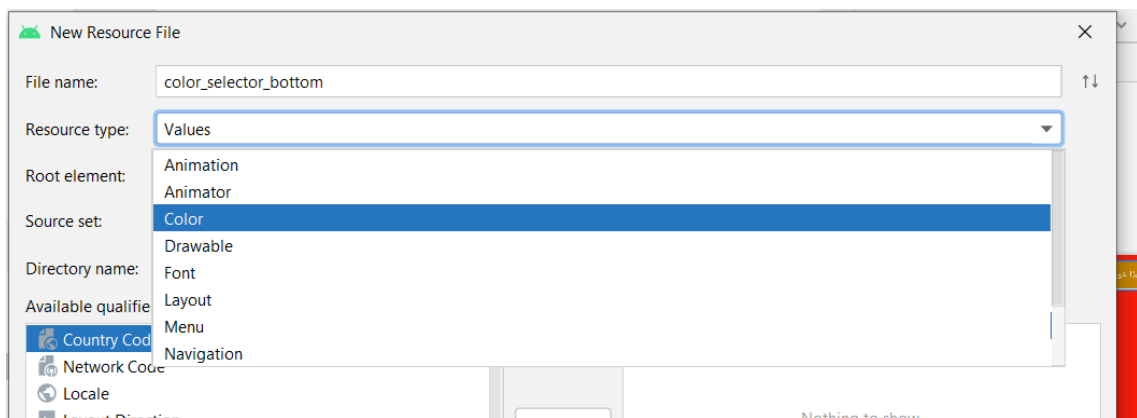
    var layoutView: View? = null
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        layoutView = inflater.inflate(R.layout.fragment_search, container, attachToRoot: false)
        cast()
        return layoutView
    }
}

```

شکل ۷-۳ استفاده از تابع onCreateView در SearchFragment.kt

به همین ترتیب سایر Fragment ها را می‌سازیم.

بایستی به هر یک از این Fragment ها یک Selector اختصاص دهیم. کافیست بر روی پوشه Resource کلیک راست کرده و از گزینه New یک Android Resource Directory ساخته و در پنجره‌ای که باز می‌شود نام فایل را color_selector_bottom گذاشته و نوع آن را color قرار می‌دهیم (شکل ۸-۳).



شکل ۸-۳ ساخت Directory برای اختصاص Selector به Fragment ها

حال در فایل color_selector_bottom.xml برای حالتی که دکمه‌ای انتخاب می‌شود رنگ آن تغییر کند، تا در زمان غیرفعال بودن متفاوت باشد، Item ها را مقداردهی می‌نماییم (شکل ۹-۳).

```

nActivity.kt × color_selector_bottom.xml × activity_main.xml ×
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:color="@color/back_ground" android:state_selected="true" />
    <item android:color="@color/item_color" android:state_selected="false"/>
</selector>

```

شکل ۹-۳ مقداردهی به Selector فایل color_selector_bottom.xml

سپس در فایل activity_main.xml و در قسمت BottomNavigation دو Attribute دیگر را اضافه می‌کنیم یکی itemTextColor و itemIconTint را مقداردهی می‌نماییم (شکل ۱۰-۳).

```

<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_navigation"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:layout_alignParentBottom="true"
    android:background="@color/bg_widget"
    app:itemIconTint="@color/color_selector_bottom"
    app:itemTextColor="@color/color_selector_bottom"
    app:menu="@menu/menu_bottom" />

```

شکل ۱۰-۳ اضافه کردن دو ویژگی در bottomNavigation و مقداردهی به آنها

با افزودن سه خط کد به فایل MainActivity.kt با کلیک بر روی هر کدام از دکمه‌های Home، Search و Person وارد Fragment آن می‌شویم (شکل ۱۱-۳).

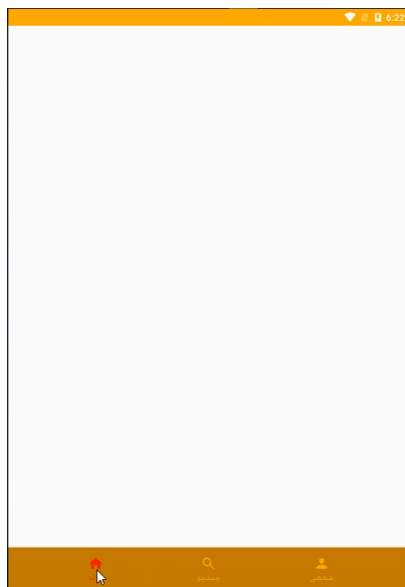
```

when (it.itemId) {
    R.id.bottom_home -> {
        supportFragmentManager.beginTransaction()
            .replace(R.id.frame_layout, HomeFragment()).commit()
        txtCategory!!.text = "خانه"
    }
    R.id.bottom_search -> {
        supportFragmentManager.beginTransaction()
            .replace(R.id.frame_layout, SearchFragment()).commit()
        txtCategory!!.text = "جستجو"
    }
    R.id.bottom_person -> {
        supportFragmentManager.beginTransaction()
            .replace(R.id.frame_layout, PersonFragment()).commit()
        txtCategory!!.text = "شخصی"
    }
}
true ^setOnNavigationItemSelectedListener
}

```

شکل ۱۱-۳ افزودن کدهای هر Bottom برای رفتن به Fragment های Home، Search و Person

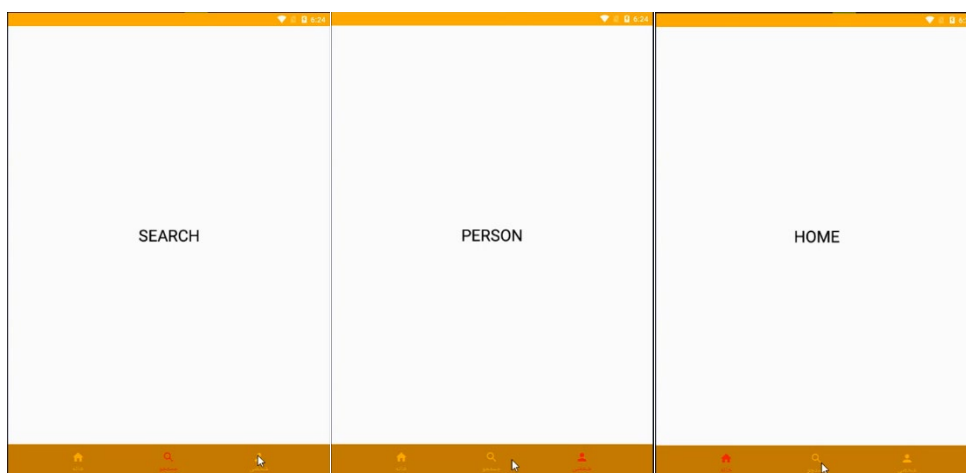
تا این جای کار با استفاده از شبیه ساز^۱ اندروید استودیو برنامه را اجرا کنیم، خروجی به شکل زیر خواهد بود (شکل ۱۲-۳).



شکل ۱۲-۳ خروجی برنامه و طراحی BottomNavigation در اندروید استودیو

^۱ Emulator

شکل زیر نشان می‌دهد با کلیک بر روی هر یک از دکمه‌ها وارد آن Fragment می‌شویم (شکل ۳-۱۳).



شکل ۳-۱۳ ورود به هر Fragment های دکمه‌های خانه، جستجو و شخصی

برای ساخت Action Bar در فایل activity_main.xml یک <RelativeLayout> اضافه می‌کنیم تا Widget هایی که بعداً می‌خواهیم به آن بیفزاییم به راحتی ساخته شوند (شکل ۳-۱۴).

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:background="@color/bg_widget"
    android:id="@+id/toolbar">
```

شکل ۳-۱۴ افزودن <RelativeLayout> برای ساخت Action Bar

برای نمایش Tool Bar^۱ در Fragment در فایل activity_main.xml و در FrameLayout، layout_below را مساوی @+id/toolbar قرار می‌دهیم. در ادامه <TextView/> را نیز می‌نویسیم تا متن خانه، جستجو و شخصی در نوار ابزار نمایش داده شود (شکل ۳-۱۵).

^۱ نوار ابزار

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:id="@+id/txt_category_bottom"
    android:textColor="@color/white"
    android:textSize="20sp"/>
```

شکل ۳-۱۵ اضافه نمودن <TextView/> برای نمایش متن Fragment ها در نوار ابزار

پس از آن در فایل MainActivity.kt بایستی TextView را findViewById کنیم و برای هر کدام textCategory را اضافه می‌کنیم (شکل ۳-۱۶).

```
...
        .replace(R.id.frame_layout, HomeFragment()).commit()
        txtCategory!!.text = "خانه"
    }
    R.id.bottom_search -> {
        supportFragmentManager.beginTransaction()
            .replace(R.id.frame_layout, SearchFragment()).commit()
        txtCategory!!.text = "جستجو"
    }
    R.id.bottom_person -> {
        supportFragmentManager.beginTransaction()
            .replace(R.id.frame_layout, PersonFragment()).commit()
        txtCategory!!.text = "شخصی"
    }
}
```

شکل ۳-۱۶ textCategory برای findViewById شدن TextView

۳-۲ Custom Shape و Adapter ViewPager، TabLayout ViewPager

در فایل fragment_home.xml از TabLayout برای تنظیم نمودن این Fragment و برای نمایش آن‌ها از ViewPager استفاده می‌کنیم (شکل ۳-۱۷). سپس در فایل HomeFragment.kt با اضافه نمودن متغیرهای tablayout و viewPager آن‌ها را می‌توانیم استفاده کنیم.

```

<com.google.android.material.tabs.TabLayout
    android:id="@+id/tab_layout"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_centerInParent="true"
    android:layout_marginHorizontal="10dp"
    android:background="@color/bg_widget"
    app:tabIndicatorColor="@color/back_ground"
    app:tabTextColor="@color/selector_tab_layout" />
</RelativeLayout>

```

```

<androidx.viewpager.widget.ViewPager
    android:id="@+id/view_pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/rel"
    android:layout_marginTop="10dp" />

```

شکل ۳-۱۷ تنظیم fragment_home و نمایش Fragment

به منظور استفاده از این دو Widget تابع cast() را تعریف می‌کنیم (شکل ۳-۱۸).

```

var tabLayout: TabLayout? = null
var viewPager: ViewPager? = null
override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    viewLayout = inflater.inflate(R.layout.fragment_home, container, false)

    cast()

    return viewLayout
}

private fun cast() {
    tabLayout = viewLayout!!.findViewById(R.id.tab_layout)
    viewPager = viewLayout!!.findViewById(R.id.view_pager)
    adapter()
}

```

شکل ۳-۱۸ استفاده از تابع cast() در فایل HomeFragment.kt

حال بایستی برای هر یک از Fragment ها Adapter نیز معین کنیم. بر روی پوشه Adapter کلیک راست کرده و یک کلاس جدید کاتلین به نام AdapterViewPager می‌سازیم. برای استفاده از این فایل بایستی در داخل فایل HomeFragment.kt یک تابع Adapter بسازیم. در پوشه fragment یک پوشه دیگر به نام tablayout ساخته و در این پوشه Fragment های Brunch، Cake، Dinner و Lunch را ایجاد می‌کنیم (شکل ۳-۱۹).



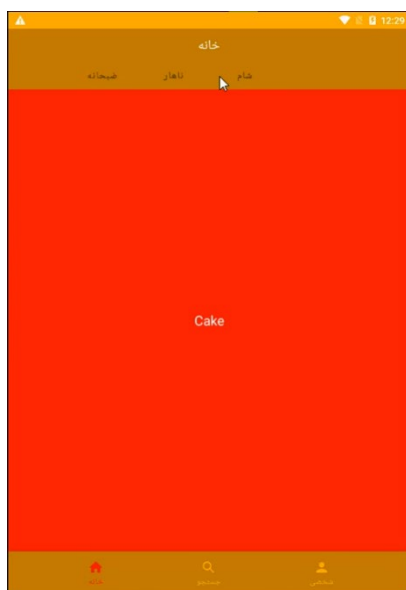
شکل ۳-۱۹ ایجاد پوشه tablayout و ایجاد Fragment های Brunch, Cake, Dinner و Lunch

برای هر یک از این Fragment ها بایست در مسیر res و در داخل پوشه layout فایل xml مربوط به هر کدام را ایجاد نمود. بعد از تمام مراحل بالا در فایل HomeFragment.kt هر چهار Fragment را اضافه نموده و همین طور ViewPager را تنظیم می نماییم (شکل ۲۰-۳).

```
private fun adapter() {
    val adapterViewPager = AdapterViewPager(childFragmentManager)
    adapterViewPager.setFragment(BrunchFragment(), title: "صبحانه")
    adapterViewPager.setFragment(LucnhFragment(), title: "ناهار")
    adapterViewPager.setFragment(DinnerFragment(), title: "شام")
    adapterViewPager.setFragment(CakeFragment(), title: "دسر")
    viewPager!!.adapter = adapterViewPager
    tabLayout!!.setupWithViewPager(viewPager)
}
```

شکل ۳-۲۰ اضافه نمودن Fragmen ها ساخته شده به HomeFragment.kt و تنظیم ViewPager

پس از اجرا گرفتن شکل زیر را در خروجی خواهیم داشت (شکل ۲۱-۳).



شکل ۳-۲۱ خروجی برنامه پس از ساختن Fragment های صبحانه، ناهار، شام و دسر

برای پس زمینه Selector را در مسیر پوشه res و در داخل پوشه color به نام selector_tab_layout.xml ایجاد می کنیم (شکل ۳-۲۲).

```
ainActivity.kt ^ fragment_home.xml ^ HomeFragment.kt ^ selector_tab_layout.xml ^ AdapterView.kt ^
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:color="@color/back_ground" android:state_selected="true" />
  <item android:color="@color/white" android:state_selected="false"/>
</selector>
```

شکل ۳-۲۲ ایجاد فایل selector_tab_layout.xml

لازم است در فایل fragment_home.xml این TabLayout را معرفی کنیم و Attribute های آن را نیز بیفزاییم.

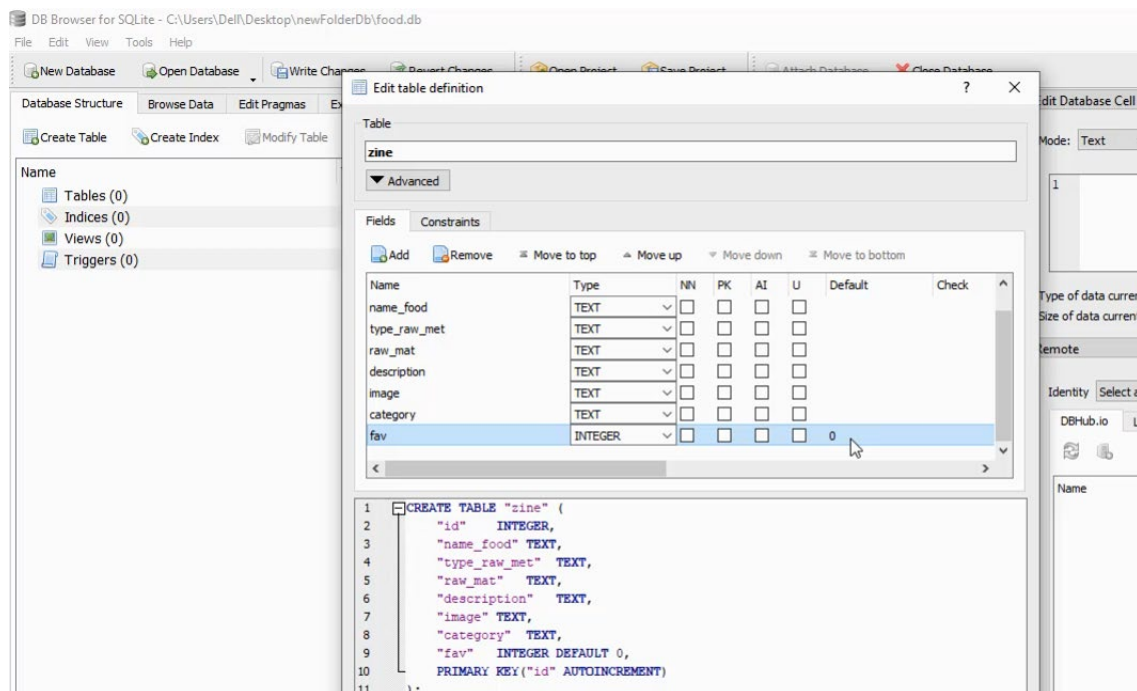
در این قسمت در مسیر پوشه drawable فایل shap_tab_layout.xml را ایجاد کرده و در داخل فایل fragment_home.xml آن را معرفی می نماییم.

۳-۳ پایگاه داده^۱ SQLite

برای استفاده از پایگاه داده ما نرم افزار SQLite را به کار می بریم. پس از اجرای این نرم افزار New Database را انتخاب نموده و مسیر ایجاد آن را معین نموده و برای آن نامی را انتخاب می کنیم. در قسمت بعدی در جدول باز شده پس از انتخاب نام برای آن با استفاده از گزینه Add هر تعداد که نیاز باشد رکورد ایجاد می نماییم. اولین Field ما id است و دومین Field را به

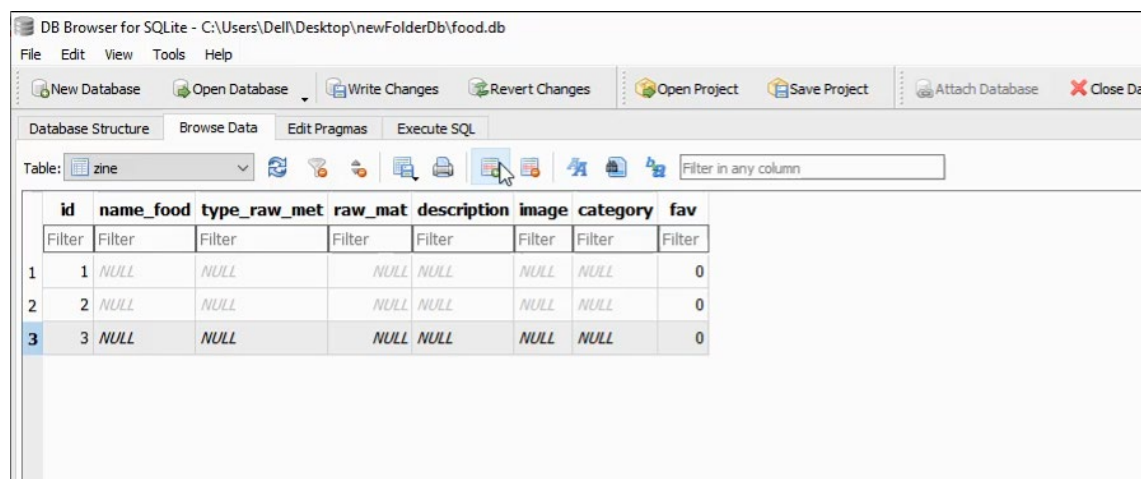
^۱ Database

name_food تغییر می‌دهیم و گزینه‌های PK و AI را نیز انتخاب می‌کنیم، باقی Field ها را مطابق شکل نام‌گذاری و مقدار هر یک را مشخص می‌کنیم (شکل ۳-۲۳).



شکل ۳-۲۳ ایجاد و مقداردهی به رکوردها در پایگاه‌داده SQLite

حال از زبانه Browse Data جدول خود را انتخاب نموده که در این جا zine می‌باشد و برای پر کردن آن از دکمه مثبت برای افزودن Field ها استفاده می‌کنیم (شکل ۳-۲۴).



شکل ۳-۲۴ ایجاد فیلدها با استفاده از زبانه Browse Data

بعد از پر کردن این فیلدها در نهایت فایلی به شکل زیر خواهیم داشت (شکل ۳-۲۵).

id	naem_food	type_raw_material	raw_material	discription	image	fav	category
16	آلسکا	شامل...	عدد 3	مرحله اول...	https://...	0	cake
17	کیک سبب و دارجین	نخم مرغ	عدد 4	در اولین...	https://...	0	cake
18	خورشت فورمه سبزی	بهار متوسط	عدد 2	مرحله اول...	https://...	0	luncher
19	خورشت بامیه	بامیه	گرم 400	در مرحله 1	https://...	0	luncher
20	قیمه نثار فروبی	رب گوجه‌فرنگی	فانشق 1	ابتدا پیاره	https://...	0	luncher
21	آنگوشت دیرک	سبب زمینی	عدد متوسط 2	در مرحله 1	https://...	0	luncher
22	خورشت آلو اسفناج با ...	گوشت گوسفند	گرم 300	در مرحله 1	https://...	0	luncher
23	کباب	گوشت چرخ کرده	کیلو گرم 1	مهم ترین	https://...	0	luncher
24	خورشت کنگر	کنگر	گرم 200	در اولین	https://...	0	luncher
25	خورشت کدو	کدو سبز	عدد 4	...	https://...	0	luncher
26	غوره مسما	مرغ	تیکه 5	ابتدا در	https://...	0	luncher
27	ته چین مرغ	سینه مرغ	عدد 1	مرحله 1-	https://...	0	luncher
28	سوپ خو فرمز	هویج متوسط	عدد 2	اولین 1-	https://...	0	dinner
29	کشتک بادمجان	بادمجان	کیلو 1	...	https://...	0	dinner
30	کوکو سبزی	نخم مرغ	تا 3 عدد 2	در ابتدا	https://...	0	dinner
31	تاس کباب	گوشت گاو و با	گرم 400	خرد کر 1-	https://...	0	dinner
32	سالاد مرغ فاست	سبب زمینی پخته ش	عدد 3	مرغ پخته	https://...	0	dinner

شکل ۳-۲۵ افزودن اطلاعات به فیلدهای پایگاه داده

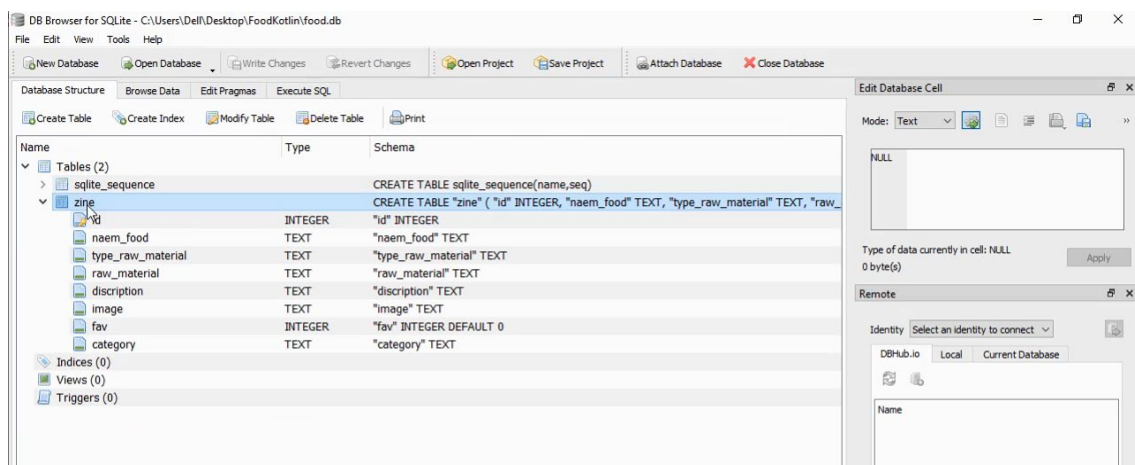
حال در داخل اندروید استودیو پوشه assets را ایجاد نموده و در داخل آن پایگاه داده‌مان را می‌افزاییم. پس از آن بایستی کلاس پایگاه داده را ایجاد کنیم.

برای این منظور در زیرشاخه java بروی com.example.appfoodkotlin راست کلیک کرده و از قسمت New گزینه New Package را انتخاب کرده و بسته 'dataBase' را می‌سازیم. داخل این Package کلاس DataBase را ایجاد می‌کنیم (شکل ۲۶-۳).

```
class DataBase(val context: Context) :
    SQLiteOpenHelper(context, dbName, factory: null, Info_db.dbVersion) {
    override fun onCreate(db: SQLiteDatabase?) {
    }
}
```

شکل ۳-۲۶ ایجاد کلاس DataBase در DataBase Package

درون پوشه DataBase فایل دیگری به نام Info_db برای وارد کردن متغیرهای مان ایجاد کرده تا از شلوغی کدها در فایل DataBase اجتناب کنیم. نکته این که مقادیر هر متغیر بایستی با نام فیلدهای پایگاه داده‌مان یکی باشد (شکل ۲۷-۳).



شکل ۳-۲۷ وارد نمودن مقادیر مربوط به هر متغیر در فایل Info_db

یک فایل به نام Food.kt در پوشه model ساخته و کلاس Food را در آن ایجاد می‌کنیم و آنرا در تابع getAll درون فایل DataBase.kt وارد می‌کنیم.

حال برای جلوگیری از ساختن Database برای هر Fragment یک Package به نام App در سرشاخه com.example.foodkotlin.activity ساخته و درون آن کلاسی به نام Base ایجاد می‌کنیم. این اولین کلاسی است که در اندروید استودیو اجرا می‌شود. از این کلاس در داخل MainActivity.kt استفاده می‌کنیم و در داخل Ftagment برای مثال SearchFragment یک تابع به نام cast ساخته دستورات مان را برای استفاده از Database وارد می‌کنیم (شکل ۲۸-۳).

```
recyclerView.adapter = GridLayoutManager(Base.activity!!, spanCount = 3)
list = Base.dataBase!!.getAll()
adapterSearch = AdapterSearch(Base.activity!! list!!)
```

شکل ۳-۲۸ استفاده از Database در فایل SearchFragment.kt

در قسمت بعدی Layout هر یک از Itemها را ساخته و اطلاعات را به هر Adapter ارجاع دهیم.

۳-۴ کتابخانه‌ی پیکاسو و Show Item

برای افزودن کتابخانه پیکاسو به فایل build.gradle به آدرس <https://github.com/square/picasso> مراجعه نموده در پایین صفحه عبارت 'com.squareup.picasso:picasso:2.71828' را کپی و در داخل فایل وارد می‌کنیم (شکل ۲۹-۳).



شکل ۳-۲۹ افزودن کتابخانه Picasso به فایل build.gradle

بعد از اضافه کردن کتابخانه پیکاسو از آن در Fragment ها استفاده می کنیم. کافی است در فایل AdapterFood.kt کد مربوطه را وارد کنیم که در شکل مشخص شده است (شکل ۳۰-۳).



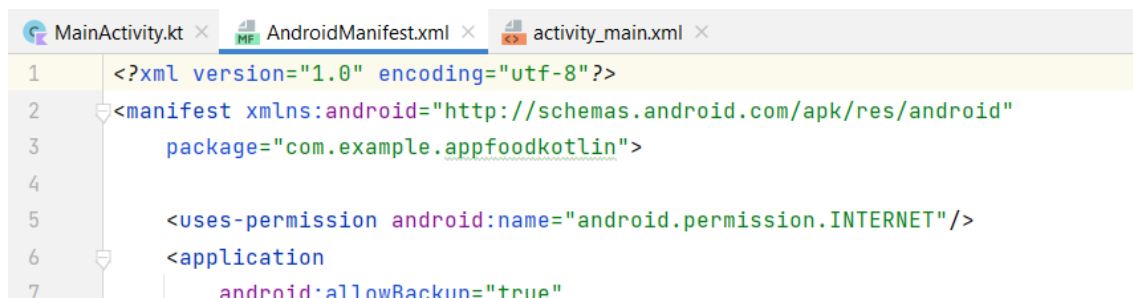
شکل ۳۰-۳ وارد کردن کد در فایل AdapterFood.kt برای استفاده از کتابخانه Picasso

حال به Fragment هایی که قرار است داده های ما را نشان دهند، رفته و از فایل fragment_brunch.xml شروع کرده و RecyclerView ها را اضافه نموده و سپس در داخل BrunchFragment.kt تابع cast را ساخته با یک متغیر در نهایت کلاس RecyclerView آن را فراخوانی می کنیم. برای سایر Fragment ها نیز RecyclerView ها را وارد می کنیم. در نهایت از برنامه اجرا گرفته و Item ها را برای ما نمایش می دهد (شکل ۳۱-۳).



شکل ۳۱-۳ نمایش Item ها بعد از افزودن کتابخانه Picasso

دلیل این که تصاویر را نمایش نمی دهد بدین خاطر است که بایستی در فایل AndroidManifest.xml به اینترنت Permission باید داده شود (شکل ۳۲-۳).



شکل ۳-۳۲ دادن Permission برای دسترسی به اینترنت

بعد از اجرای مجدد برنامه خواهیم دید که تصاویر نیز برای ما به نمایش در خواهند آمد (شکل ۳-۳۳).



شکل ۳-۳۳ دریافت تصاویر بعد از دادن Permission / اینترنت به برنامه

در قسمت بعد به قسمت جزئیات هر Item می‌پردازیم.

۳-۵ ساخت صفحه جزئیات هر Item

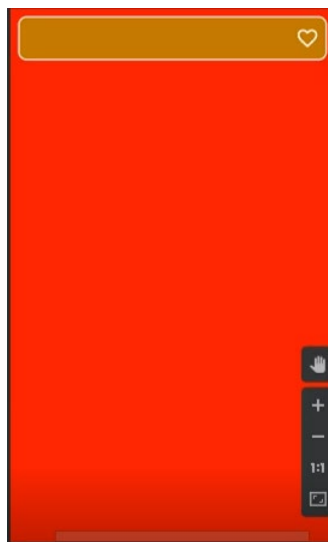
برای ایجاد صفحه Detail^۱ یک Activity با نام MainActivity.kt ایجاد می‌کنیم و همین‌طور Layout آنرا هم پیاده‌سازی می‌کنیم به وسیله فایل activity_detail.xml برای آن Layout را می‌نویسیم. در نوار ابزاری که در بالای صفحه می‌سازیم، نوع و نام غذا را نمایش می‌دهیم و دکمه‌ای برای افزودن آن به لیست علاقه‌مندی‌ها و هم چنین یک دکمه برای بازگشت تعبیه می‌کنیم. یک <ImageView/> ساخته و مقادیر آنرا وارد می‌کنیم (شکل ۳-۳۴).

^۱ جزئیات

```
<ImageView
    android:id="@+id/btn_favorite"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:layout_marginStart="10dp"
    android:src="@drawable/ic_baseline_favorite_border_24" />
```

شکل ۳-۳۴ افزودن <ImageView/> برای ساخت نوار ابزار

برای افزودن تصویر Favorite در پوشه drawable یک Vector Asset ایجاد کرده و شکل آن را می‌سازیم (شکل ۳-۳۵).



شکل ۳-۳۵ ساختن شکل Icon علاقه‌مندی در نوار ابزار صفحه جزئیات هر Item

برای نمایش نوع و نام غذا <TextView/> و برای نمایش دکمه بازگشت یک <ImageView/> دیگر می‌سازیم. در مرحله بعد <ImageView/> را با یک CardView در داخل یک قاب قرار می‌دهیم. یک <TextView/> برای مواد اولیه نیز به آن می‌افزاییم (شکل ۳-۳۶).



شکل ۳-۳۶ activity_detai.xml بعد از افزودن <TextView/> و <ImageView/>

حال به فایل DetailActivity.kt رفته و برای نمایش نام و نوع غذا متغیرهای txtRawMat و txtTypeMat را به آن می‌افزاییم. به همین شکل متغیر txtDescription برای توضیحات هر وعده غذایی و همین‌طور یک متغیر ImageView برای نمایش تصاویر نیز ایجاد می‌کنیم که تمامی این موارد بایستی findViewById شوند (شکل ۳-۳۷).

```
class DetailActivity : AppCompatActivity() {
    var txtName: TextView? = null
    var txtTypeRawMat: TextView? = null
    var txtRawMat: TextView? = null
    var txtDescription: TextView? = null
    var imageView: ImageView? = null

    var btnFavorite: ImageView? = null
    var id: Int? = null
    var value: Int? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        @SuppressLint("SetTextI18n")
        private fun getData() {
            private fun cast() {
                txtName = findViewById(R.id.txt_name_food_detail)
                txtTypeRawMat = findViewById(R.id.txt_type_raw_mat)
                txtRawMat = findViewById(R.id.txt_raw_mat)
                txtDescription = findViewById(R.id.txt_description)
                imageView = findViewById(R.id.image_detail)
                btnFavorite = findViewById(R.id.btn_favorite)
                val back = findViewById<ImageView>(R.id.btn_back_detail)
                back.setOnClickListener {
                    it: View!
                }
            }
        }
    }
}
```

شکل ۳-۳۷ وارد کردن TextView و ImageView در فایل DetailActivity.kt

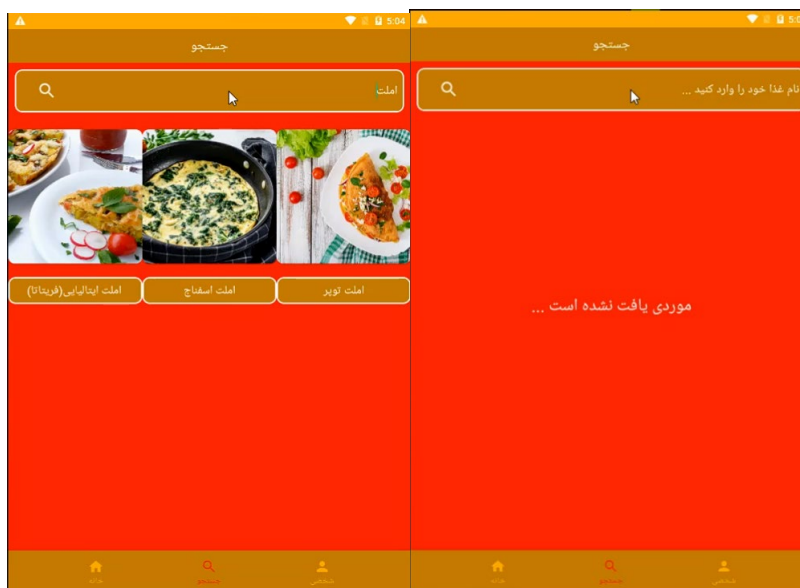
در نهایت شکل زیر را خواهیم داشت (شکل ۳-۳۸).



شکل ۳-۳۸ شکل نهایی صفحه جزئیات در برنامه

۳-۶ قسمت جستجوی هر Item

ابتدا کدهای مربوط به `fragment_search.xml` را می‌نویسیم. سپس در فایل `SearchFragment.kt` با استفاده از تابع `cast` کلیه `layoutView` ها را تعریف می‌کنیم. حال یک فایل `AdapterSearch.kt` نیز می‌سازیم و درون آن یک `search_layout.xml` ایجاد می‌کنیم. اولین تابعی که اطلاعات را از پایگاه داده می‌گیرد تابع `getAll()` می‌باشد که در فایل `SearchFragment.kt` آن را خواهیم ساخت و مقدار آن را به متغیر `list` انتساب می‌دهیم. برای پس‌زمینه جستجو نیز یک فایل `back_search.xml` می‌سازیم. بعد از اجرای برنامه خواهیم دید که در قسمت جستجو اگر موردی یافت نشود پس‌زمینه را نشان می‌دهد در غیر این صورت مواردی را که بیابد حتماً نشان خواهد داد (شکل ۳۹-۳).



شکل ۳-۳۹ شکل نهایی قسمت جستجو پس از اجرای برنامه

۳-۷ اضافه کردن Item به قسمت علاقه‌مندی

در کلاس DataBase سه تابع برای قسمت علاقه‌مندی داریم. اولین تابع `getFav()` می‌باشد که تمامی موارد را دریافت می‌کند. تابع دوم که مقدار هر یک از Id ها را خواهد گرفت به نام `getValue()` می‌باشد. در آخر تابع `getStatus()` که دو مقدار را دریافت خواهد کرد (شکل ۳-۴۰).

```

fun getAll(): ArrayList<Food> {...}

fun getCategory(category: String): ArrayList<Food> {...}

fun getFav(): ArrayList<Food> {...}

fun getValue(id: Int): Int {...}

fun getStatus(status: Int, id: Int) {...}

```

شکل ۳-۴۰ افزودن توابع `getFav`، `getValue`، `getCategory` و `getStatus`

در فایل `DetailActivity.kt` از این توابع استفاده خواهیم کرد. برای این که بدانیم آیا موردی به قسمت علاقه‌مندی‌هایمان اضافه شده است یا خیر یک تابع `checkFav()` ایجاد می‌کنیم.

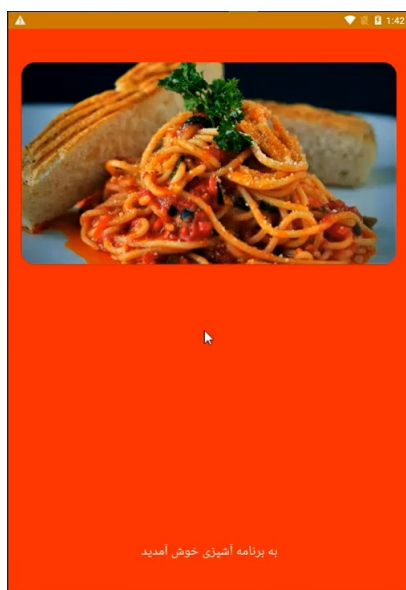
بعد از اجرا چند مورد را برای اضافه شدن به قسمت علاقه‌مندی‌ها انتخاب می‌کنیم اما برای دیده شدن در قسمت شخصی باید PersonFragment.kt را نیز تکمیل کنیم. ابتدا در فایل fragment_person.xml کدهای مربوط به RecyclerView را وارد می‌کنیم سپس به فایل PersonFragment بازگشته و تابع cast را به آن می‌افزاییم. پس از تکمیل و اجرای برنامه موارد اضافه شده به لیست علاقه‌مندی‌ها را در قسمت شخصی مشاهده خواهیم کرد (شکل ۳-۴۱).



شکل ۳-۴۱ اضافه شدن موارد انتخاب‌شده به قسمت علاقه‌مندی‌های صفحه شخصی

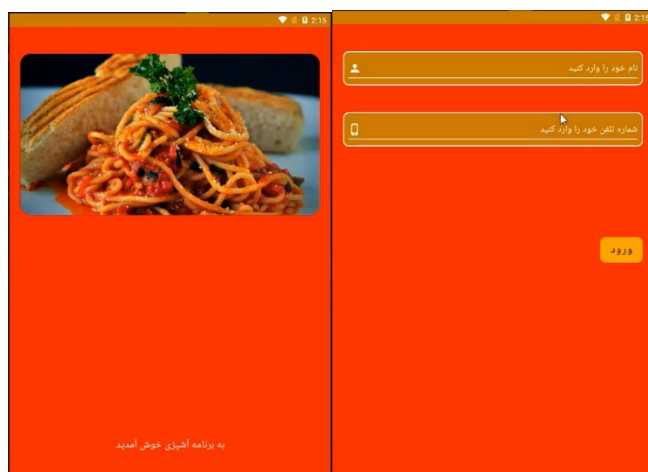
۳-۸ صفحات SplashScreen و Login

دو فایل activity_login.xml و activity_splash.xml را می‌سازیم. در صفحه SplashScreen قالبی می‌سازیم که یک تصویر را برای ما به مدت چند ثانیه نشان دهد و یک پیام خوش‌آمدگویی به کاربر نشان دهد. تصویری که قبلاً دریافت کرده‌ایم را در پوشه drawable ذخیره کرده و قرار می‌دهیم. یک <TextView/> برای نمایش پیام خوش‌آمدگویی نیز به آن اضافه می‌کنیم. حال به فایل SplashActivity.kt رفته و با استفاده از کتابخانه Handler زمان را برای نمایش صفحه خوش‌آمدگویی تنظیم می‌کنیم. پس از اجرای برنامه، این صفحه را مشاهده خواهیم نمود (شکل ۳-۴۲).



شکل ۳-۴۲ صفحه اول SplashScreen برای خوش آمدگویی

حال به صفحه Login می‌رسیم که دو ورودی برای آن داریم. دو فایل `activity_login.xml` و `LoginActivity.kt` را ایجاد کرده و در فایل `activity_login.xml`، قالب‌های ورود نام و شماره تماس را خواهیم ساخت. برای این منظور از `<RelativeLayout>` و `<EditText/>` استفاده می‌کنیم تا محل ورود نام کاربر را بسازیم به همین شکل برای ورود شماره تلفن نیز اقدام کرده و از `<ImageView/>` برای افزودن شکل به محل ورود هر دو متغیر استفاده می‌کنیم. با استفاده از `<Botton/>` یک دکمه ورود برای بعد از وارد شدن نام و شماره تلفن ایجاد می‌کنیم. برای این دکمه یک شکل هم در پوشه `drawable` می‌سازیم با نام `shape_login.xml` و سپس در فایل `LoginActivity.kt` دستوراتی برای زمانی که ورودی‌ها خالی و پر هستند، خواهیم نوشت. حال یک کلاس `SharedPref.kt` می‌سازیم تا از این سه متغیر را بعداً استفاده کنیم. فقط برای دکمه ورود در فایل `Base.kt` یک متغیر از جنس `SharedPreferences` می‌افزاییم. اگر برنامه را اجرا کنیم ابتدا صفحه خوش آمدگویی را خواهیم داشت و بعد از چهار ثانیه به صفحه ورود خواهیم رسید (شکل ۳-۴۳).



شکل ۳-۴۳ صفحه خوش آمدگویی و صفحه ورود برنامه

۳-۹ جمع بندی

در این فصل ابتدا صفحه MainActivity را ساختیم که شامل یک BottomNavigation و یک FrameLayout است که دسته‌بندی مربوط به هر یک از این سه Fragment را با استفاده از این دو ویجت نمایش می‌دهد. در قسمت بعدی هر یک از این Fragment ها را داخل صفحه HomeFragment تنظیم نمودیم. سپس با استفاده از نرم‌افزار DB Browser Sqlite پایگاه داده‌ای برای برنامه ایجاد کردیم. بعد از این مرحله Category هر یک از Layout ها را ایجاد نموده و اطلاعات موجود در پایگاه داده را به هر کدام از این Adapter ها مرتبط ساختیم. پس از افزودن پایگاه داده به پروژه مان Adapter Food را کامل نمودیم، در ادامه کتابخانه Picasso را از سایت محبوب Github به Gradle پروژه افزودیم و برای هر Fragment به ساخت RecyclerView پرداختیم. قسمت مربوط به صفحه جزئیات را در ادامه تکمیل نموده به شکلی که با کلیک بر روی هر یک از این موارد داخل Activity Detail آن مورد شده و جزئیات آن را به ما نشان داد و هم‌چنین به صفحه قابلیت Scroll را افزودیم. صفحه جستجو قسمت بعدی بود که به آن پرداخته شد و سپس تکمیل قسمت مورد علاقه‌ها انجام شد. صفحه Login و SpalshScreen مرحله بعدی بود که با استفاده از کلاس handler در داخل SplashActivity زمان نمایش صفحه خوش آمدگویی را تعریف نمودیم.

فصل چہارم

نتائج پروژہ

مقدمه

در دهه گذشته، استفاده از برنامه‌های تحت گوشی‌های هوشمند افزایش چشم‌گیری در سراسر دنیا داشته است. تقاضای بسیار برای برنامه‌های موبایلی از سوی اکثر سازمان‌های بزرگ و کسب‌وکارها وجود دارد. در میان ارائه خدمات و کالا به مشتریان، صاحبان کسب‌وکار از طریق این برنامه‌ها، سرعت بازگشت سرمایه‌شان را سرعت می‌بخشند. امروزه، بیش تر ترافیک برخط به‌خاطر دستگاه‌های سیار می‌باشد و مردم از برنامه‌های موبایلی برای هر چیزی استفاده می‌کنند، برای ارتباط با دیگران، نحوه زندگی، خرید، غذا، ورزش، کسب‌وکار، بازی و بسیاری دیگر و وقتی که مردم نتوانند از خانه خود خارج شوند، از این که هنوز به بسیاری از دانش‌ها از طریق این برنامه‌ها و در خانه‌شان دسترسی دارند، خیال‌شان راحت می‌شود.

امروزه مردم، بیش تر و بیش تر به سمت این برنامه‌ها برای جستجوی طرز تهیه غذاها روی می‌آورند و زمان کتاب‌های آشپزی به‌سر آمده است. امروزه همه چیز دیجیتالی شده و مردم این که همه انواع طرز تهیه غذاها را تحت یک بستر در اختیار داشته باشند، بسیار دوست دارند. میلیون‌ها انسان در سراسر جهان به‌دنبال یافتن دستورهای پخت و برنامه‌های آشپزی هستند. بنابراین اگر شما هم به‌دنبال ساخت یک برنامه آشپزی برای دستگاه‌های هوشمند هستید، انتخاب درستی نموده‌اید.

۴-۱ چرا برنامه‌های آشپزی بسیار محبوب هستند؟

- **لذت تجربه آشپزی:** بسیاری از نسل حاضر امروزی، به تخصیص بخشی از اوقات فراغت خود به آشپزی علاقه‌مند می‌باشند، و در این میان وعده‌های غذایی جدید و ناآشنا را ترجیح می‌دهند و بسیاری از برنامه‌های آشپزی ممکن است به دستیاری باوفا برای آنان تبدیل شوند.
- **ارتباط بیش‌تر:** هم‌چنین این گونه برنامه‌ها ممکن است، منجر به یافتن افرادی هم‌سلیقه و هم‌فکر و ارتباط با آنها شوند. بسیاری از کاربران می‌توانند دوستان جدیدی بیابند و در مورد دستورهای پختی که به آنها علاقه دارند و وعده‌های غذایی متفاوت با هم به بحث و ابراز عقیده بپردازند.
- **افزایش محبوبیت نمونه‌های عندالمطالبه:** امروزه همه ما دائما در حال عجله‌ایم و بنابراین دوست داریم تا قادر به گرفتن هر خدمتی به‌صورت برخط و در هر جایی و هر زمانی باشیم. تا کسی اسنپ یک نمونه از آن می‌باشد. به ماشین نیاز داریم، کافی‌ست چند ضربه بزیم و نزدیک‌ترین تاکسی را بیایم. همین حالت برای آشپزی نیز اتفاق افتاده است. می‌خواهیم چیزی بپزیم بدون این که زمان زیادی برای جستجوی دستور پخت درستی صرف کنیم. بنابراین بایستی برنامه‌ی مناسبی برای این هدف خاص داشته باشیم که این مهم‌ترین دلیل برای ساخت یک برنامه آشپزی است.
- **دیجیتالی شدن سراسری:** بسیاری از شرکت‌ها تلاش می‌کنند برای برآورده کردن نیاز مشتریان، خدمات‌شان را به‌صورت برخط ارائه دهند و آشپزی هم از این قاعده مستثنا نیست.

۴-۲ جمع بندی

مطالب بالا بایستی شما را متقاعد کند که ساخت یک برنامه آشپزی ارزشش را دارد، به شرطی که پروژه ساخت آن به‌درستی و بهترین روش انجام پذیرد. اول از همه شما بایستی تصمیم بگیرید که برنامه شما بر روی چه دستگاهی قرار است اجرا شود. برای این منظور بهتر است کاربران آتی‌تان را در نظر بگیرید و تشخیص دهید که آنها از چه دستگاهی استفاده خواهند کرد. اگر بیش‌تر آنها دستگاه‌های هوشمند اپل را انتخاب می‌کنند، پس ساخت یک برنامه مبتنی بر IOS را آغاز کنید. اگر می‌خواهید موفق شوید، بایستی برنامه موبایلی شما بر روی تمامی دستگاه‌ها قابل اجرا باشد. پیشنهاد قبلی ما برای برنامه‌های مقید است که تازه ساخته شده‌اند. وقتی برنامه‌ای ساخته می‌شود و برای شما سودآور شد بایستی قابلیت‌های آن را توسعه بخشید. انتخاب زبان برنامه‌نویسی، یکی دیگر از مولفه‌ها می‌باشد. موارد بسیاری موجود است که نمی‌توان تمامی آنها را نام برد. کارایی برنامه، مقیاس‌پذیری برنامه و هزینه ساخت برخی از مواردی هستند که طراحی و ساخت برنامه شما به آنها بستگی دارد.

فصل پنجم

جمع‌بندی و پیشنهادها

مقدمه

ساخت برنامه آشپزی در این سال‌ها خواهان بسیاری پیدا کرده است و می‌توان انتظار داشت که مردم بیش‌تر و بیش‌تر به سمت استفاده از این گونه برنامه روی بیاورند. ساخت این برنامه‌ها برای کسب و کارهای نوپا ایده‌ی بسیار خوبی است. طراحی و توسعه یک برنامه آشپزی پیچیده و طولانی است. در این پروژه سعی گردید تا نوعی از برنامه‌های آشپزی که در گزارش بدان پرداخته شد، طراحی و پیاده‌سازی شود که قطعاً دارای نواقص و ایراداتی نیز هست. در ادامه سعی می‌گردد تا پیشنهادهایی برای توسعه و بهتر شدن پروژه ارائه گردد که قاعدتاً دانش و زمان بیش‌تری را طلب می‌کند.

۵-۱ چه نوع برنامه‌ای می‌خواهیم، بسازیم؟

مهم نیست که چه نوع برنامه‌ای می‌خواهید بسازید، مطمئن شوید، برنامه‌ای که می‌سازید دارای صفحه آغازینی باشد که کاربر را تشویق و ترغیب کند تا برای مدت طولانی مایل به استفاده از برنامه شما باشد. گذاشتن نوار جستجو در صفحه، کاربر را ترغیب می‌کند تا دنبال غذایی که می‌خواهد گشته و مواردی را که علاقه دارد به فهرست علاقه‌مندی‌هایش بیفزاید. اگر محتوای برنامه شما نتواند رضایت کاربر را تامین کند، حوصله آن‌ها را سر برده و باعث می‌شود تا برنامه شما را ترک کنند.

۵-۲ ویژگی‌هایی که برنامه‌های آشپزی باید داشته باشند

زمانی که برنامه آشپزی می‌سازید، مطمئن شوید که دو نسخه از آن را طراحی می‌کنید، هر کدام با مجموعه‌ای از ویژگی‌های خودش.

۵-۲-۱ ویژگی‌های کاربری

- **ثبت نام:** با ساده‌ترین چیز شروع می‌کنیم، که ثبت نام می‌باشد. مهم‌ترین ترفند این است که نباید کاربر را مجبور به گذراندن مراحل غیرضروری کرد چرا که از نصب برنامه شما منصرف شده و ایده‌ی دیگری را برمی‌گزیند. راحت‌ترین راه برای این منظور از طریق رسانه‌های اجتماعی می‌باشد.
- **ساختن حساب کاربری:** کاربری که ثبت نام کرده، حال لازم است به دیگر کاربران خود را معرفی کند. او را به پرکردن یک‌سری سوالات نظیر سن، جنسیت، محل، شغل، علاقه‌ها و دیگر چیزها ترغیب کنید. به هر حال کاربری که از طریق رسانه‌های اجتماعی اقدام به ثبت نام کرده است، اطلاعات کلیدی او از طریق فیس‌بوک، توییتر و اینستاگرام قابل جمع‌آوری است.
- **جستجوی دارای فیلتر:** تمامی برنامه‌های آشپزی دارای نوار جستجو هستند، اما کاربر می‌تواند با اعمال فیلتر هنگام جستجو، مشخص کند که به کدام نوع از دستور پخت علاقه‌مند است.
- **فهرست دستورهای در دسترس:** کاربر چیزی را که می‌خواهد مشخص می‌کند و دکمه جستجو را می‌زند، حال برنامه شما بایستی تمامی موارد مطابق با مورد جستجو را نمایش دهد.

- **آموزش ویدئوی:** این خوب است که دستور تهیه غذا به همراه جزئیات وجود دارد، اما بهتر است کاربر به تواند فرآیند تهیه غذا را نیز مشاهده کند. این دقیقاً زمانی موثر است که تمامی فیلم‌های آموزشی در دسترس قرار گیرند.
- **به اشتراک گذاری:** کاربران عاشق به اشتراک گذاری محتوا در صفحات اجتماعی هستند که شامل آشپزی هم می‌شود. بنابراین دکمه به اشتراک گذاری به کاربر امکان می‌دهد تا دستورهای پخت مورد علاقه‌اش را در فیس‌بوک و سایر صفحات اجتماعی با دیگر کاربران به اشتراک بگذارد.
- **مدیریت تنظیمات:** به کاربر اجازه دهید تا حساب کاربری خود را مدیریت کرده و در صورت لزوم تغییر دهد.
- **رتبه‌بندی:** کاربران از این ویژگی استقبال می‌کنند. به این وسیله آن‌ها قادر به رتبه‌بندی دستورهای پختی هستند که مایلند دیگر کاربران آن‌ها را امتحان کنند و نظرات دیگر کاربران را در مورد آن وعده خاص مطالعه کنند قبل از این که خودشان بخواهند آن را آماده کنند.

۵-۲-۲ ویژگی‌های مدیریتی

قسمت دوم ویژگی‌های مدیریتی را هدف قرار داده است. این به معنای مدیریت سیستم می‌باشد مثل: صورت حساب، پرداخت، کاربران، محتوا و سایر موارد.

- **فهرست خرید:** فرض کنید که کاربر به تهیه یک دستور پخت علاقه دارد ولی برخی از مواد لازم را در یخچال خود ندارد. چه کاری می‌تواند انجام دهد؟ درست است، او می‌تواند آن‌ها را از طریق برنامه شما به لیست خریدش اضافه کند. این یکی از فواید برنامه‌های آشپزی است.
- **تخمین هزینه دستور تهیه:** قسمت قبل منطقاً منجر به این می‌شود، زمانی که فهرست خریدی ساخته شد، مهم است که هزینه آن چقدر است. تخمین هزینه به کاربر کمک می‌کند تا بودجه‌اش را با دستور تهیه منطبق کند.
- **یافتن نزدیک‌ترین فروشگاه:** کاربری که فهرست خرید دارد، نیاز دارد بداند، از کدام فروشگاه می‌تواند مواد لازم را تهیه کند. شما می‌توانید نزدیک‌ترین فروشگاه را به او به وسیله‌ی برنامه‌تان پیشنهاد دهید.

این موارد تنها بخشی از قابلیت‌های است که می‌توان به برنامه اضافه نمود، امید است تا با افزایش دانش و مهارت به توان در آینده به این هدف نائل شد.

مراجع

مراجع

<https://github.com/MohsenGol/picasso>, 17 February 2022.

واژه‌نامه

واژه‌نامه فارسی به انگلیسی

Gadget	ابزارک
Permission	اجازه
Details	جزئیات
Attribute	خاصیت
Directory	مسیر
Toolbar	نوار ابزار

واژه‌نامه انگلیسی به فارسی

Application	برنامه
Constructor	سازنده
Background	پس زمینه
Data Base	پایگاه داده
Emulator	شبیه ساز
Fragment	بخش؛ تکه
Framework	چارچوب
Function	تابع
Kotlin	نوعی زبان برنامه نویسی
Load	بارگذاری
Online	برخط
Open Source	منبع باز
Platform	بستر
Sleep Mode	حالت خواب
Smart Phone	تلفن هوشمند
Splash Screen	صفحه آغازین برنامه
Text Recognition	تشخیص متن
User Interface	رابط کاربری
Package	بسته

Abstract

In the age of technology development, it seems impossible to imagine activity or rest without the use of gadgets. Thanks to mobile devices, users can perform a variety of tasks using the World Wide Web at any time, regardless of their location. Internet applications are designed to provide users with easy access to the elements they need to start a business or the tools they use in their daily lives.

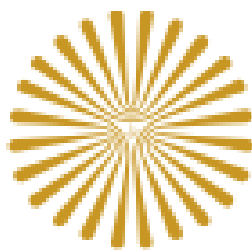
Stable internet access, wide range of services and ease of use are criteria that attract even troubled users of Android or IOS devices. Subject-oriented applications are engineered to meet the expectations of users of all ages. The user interface plays an important role. Every day, mobile app developers are faced with optimizing their products. This not only helps to increase the quality and better execution of programs, but also increases competition in the technology business markets. The balance between quality and price of products is the main demand in the market.

Creating a cooking application means that, first of all, users can choose the recipe and even the way they prepare it. An application database can contain thousands of commands from around the world. You can easily find a way to prepare the right ingredients and realize your idea. Using the application as a cook book makes it easy to specify your favorite recipe for later use. Calorie calculation is also possible with this device.

Before defining the steps of building and designing a cooking application under the Android platform, questions must be answered: How can you make a recipe app in the best way? What points should be considered during the construction of the application? What factors can cause design defects?

It is very valuable to get acquainted with the types of applications available in this field. There are five types of cooking applications:

- A set of preparation instructions
- Customizable cooking applications
- Social media apps
- Materials reference books for cooking
- Measuring tools



Payame Noor University

Faculty of Engineering

Department of Computer Engineering and Information Technology

Payame Noor University (Tehran Shomal Centre)

B.Sc. Final Project Report

**Design and Implementation of Cooking Application
for Android**

Under Supervision of:

Dr. Seyed Ali Razavi Ebrahimi

By:

Mohsen Goli

June 2021