

به نام خدا



برنامه نویسی موبایل

استاد مهرداد احمدی نیک

گزارش پروژه Event Planner

کیارش کیانیان - ۹۹۱۰۵۶۷۸

امیررضا ابوطالبی - ۹۹۱۰۵۱۹۷

مهرشاد برزمینی - ۹۹۱۷۰۳۶۱

بهار ۱۴۰۳

قسمت اول: معرفی و ویژگی‌های برنامه

معرفی

پروژه‌ای که در حال پیاده‌سازی آن هستیم، یک برنامه ریزری رویدادها است که به کاربران کمک می‌کند تا وظایف خود را به صورت مؤثری مدیریت کنند. این برنامه به صورت یک ساختار مشتری-سرور پیاده‌سازی می‌شود که بخش سرور با استفاده از پایتون و بخش کلاینت با استفاده از جاوا و XML توسعه داده شده است.

ویژگی‌های برنامه

1. احراز هویت کاربران:

- ثبت‌نام کاربر: کاربران می‌توانند با استفاده از ایمیل و رمز عبور در برنامه ثبت‌نام کنند.
- ورود و خروج: کاربران می‌توانند وارد سیستم شده و از آن خارج شوند.
- تغییر رمز عبور: امکان تنظیم مجدد رمز عبور برای کاربران فراهم شده

2. مدیریت وظایف:

- ایجاد، ویرایش و حذف وظایف: کاربران می‌توانند وظایف خود را ایجاد، ویرایش و در صورت نیاز حذف کنند.
- دسته‌بندی وظایف: وظایف می‌توانند به دسته‌های مختلفی مانند "برای انجام"، "در حال انجام" و "تکمیل شده" دسته‌بندی شوند.
- تعیین اولویت وظایف: وظایف می‌توانند اولویت‌بندی شوند (بالا، متوسط، پایین) تا کاربر بتواند به ترتیب اولویت آن‌ها را انجام دهد.
- تعیین مهلت انجام وظایف: کاربران می‌توانند برای هر وظیفه مهلت انجام تعیین کنند تا از زمان‌بندی صحیح اطمینان حاصل کنند.

3. برنامه‌ریزی وظایف:

- برنامه‌ریزی خودکار وظایف: برنامه وظایف را بر اساس مهلت‌ها و اولویت‌ها به صورت خودکار برنامه‌ریزی می‌کند.
- برنامه‌ریزی دستی وظایف: کاربران می‌توانند وظایف را به صورت دستی با استفاده از رابط کاربری کشیدن و رها کردن برنامه‌ریزی کنند.
- نمای وظایف: برنامه دارای نمای وظایف است که نمای کلی از وظایف را به کاربر نشان می‌دهد.

4. اعلان‌ها و یادآوری‌ها:

- اعلان‌های پیش‌موعد: برای وظایفی که مهلت آن‌ها نزدیک است، اعلان‌های پیش‌موعد به کاربران ارسال می‌شود.

- یادآوری‌های وظایف با اولویت بالا: برای وظایف با اولویت بالا، یادآوری‌های خاصی ارسال می‌شود تا کاربر آن‌ها را فراموش نکند.

5. رابط کاربری زیبا و کاربرپسند:

- تا جای ممکن سعی شده است که یک رابط کاربری زیبا طراحی کرده تا باعث راحتی کاربر در فضای اپلیکیشن شود. درواقع سعی بر پیاده سازی رابط کاربری بصورت user-friendly شده است.

6. استفاده از پایگاه داده PostgreSQL:

- ذخیره‌سازی اطلاعات کاربر و وظایف: اطلاعات کاربران و وظایف در یک پایگاه داده PostgreSQL ذخیره می‌شود که امکان مدیریت بهتر داده‌ها را فراهم می‌کند.

قسمت دوم: مراحل پیاده‌سازی برنامه

فاز 1: راه‌اندازی اولیه

1. راه‌اندازی مخازن پروژه:

- مخازن کد منبع برای بخش سرور و کلاینت در یک سیستم مدیریت نسخه (مثل Git) ایجاد می‌شود.
- ساختار اولیه پوشه‌ها و فایل‌های پروژه تعیین و ایجاد می‌شود.

2. پیکربندی محیط توسعه:

- محیط توسعه با نصب و پیکربندی ابزارها و کتابخانه‌های مورد نیاز آماده می‌شود.
- برای سرور از پایتون و جنگو (Django) استفاده می‌شود و برای کلاینت از اندروید SDK و ابزارهای مرتبط.

3. پیاده‌سازی احراز هویت کاربران:

- قابلیت ثبت‌نام، ورود و خروج کاربران در بخش سرور پیاده‌سازی می‌شود.
- API‌های مربوط به احراز هویت با استفاده از JWT برای امنیت بیشتر طراحی و پیاده‌سازی می‌شود.

فاز 2: توسعه ویژگی‌های اصلی

1. توسعه ویژگی‌های مدیریت وظایف:

- API‌های مربوط به ایجاد، ویرایش، حذف و مشاهده وظایف در بخش سرور پیاده‌سازی می‌شود.
- مدل‌های داده مربوط به وظایف و دسته‌بندی‌ها در پایگاه داده تعریف و پیاده‌سازی می‌شود.

2. پیاده‌سازی منطق برنامه‌ریزی وظایف:

- الگوریتمی برای برنامه‌ریزی وظایف بر اساس مهلت‌ها و اولویت‌ها توسعه داده می‌شود.
 - از Celery برای پردازش وظایف پس‌زمینه و برنامه‌ریزی وظایف استفاده می‌شود.
- ## 3. طراحی و توسعه رابط کاربری اصلی:
- رابط کاربری برای نمایش لیست وظایف، جزئیات وظایف و رابط کاربری برنامه‌ریزی طراحی و پیاده‌سازی می‌شود.

فاز 3: ارتقاء و بهبودها

1. اضافه کردن اعلان‌ها و یادآوری‌ها:

- قابلیت ارسال اعلان‌ها و یادآوری‌ها برای وظایف با مهلت نزدیک پیاده‌سازی می‌شود.
- کاربران می‌توانند تنظیمات اعلان‌ها را مطابق نیازهای خود تنظیم کنند.

2. پیاده‌سازی ویژگی‌های گزارش‌دهی و تحلیل:

- گزارش‌های آماری از وظایف انجام شده، پیشرفت هفتگی/ماهانه و روندهای بهره‌وری کاربران تهیه می‌شود.
- این گزارش‌ها به کاربران کمک می‌کند تا بهره‌وری خود را بهبود بخشند.

3. بهینه‌سازی الگوریتم برنامه‌ریزی وظایف:

- الگوریتم برنامه‌ریزی وظایف با استفاده از بازخورد کاربران و تست‌های مختلف بهینه‌سازی می‌شود.
- بهبود کارایی و دقت برنامه‌ریزی وظایف.

فاز 4: تست و استقرار

1. انجام تست‌های کامل:

- تست‌های واحد برای کد بخش سرور (APIها و منطق برنامه‌ریزی) نوشته و اجرا می‌شود.
- تست‌های رابط کاربری و تست‌های مربوط به سمت سرور برای اطمینان از پیاده‌سازی درست برنامه انجام می‌شود.

نتیجه‌گیری

با پیاده‌سازی این برنامه، کاربران می‌توانند وظایف خود را به صورت مؤثر مدیریت کرده و از ابزارهای متنوع برای برنامه‌ریزی، اعلان‌ها و گزارش‌دهی استفاده کنند. برنامه با استفاده از پایگاه داده PostgreSQL برای ذخیره‌سازی داده‌ها و Celery برای برنامه‌ریزی وظایف، از لحاظ فنی قوی و قابل اعتماد است. با استفاده از این برنامه، کاربران می‌توانند بهره‌وری خود را افزایش داده و وظایف خود را به بهترین شکل ممکن مدیریت کنند.

توضیحات کلی و جزئیات پروژه سرور

ساختار کلی پروژه

پروژه سرور ما شامل چندین اپلیکیشن مختلف است که هر کدام وظایف خاصی را انجام می‌دهند. این اپلیکیشن‌ها شامل:

- **authapp** برای مدیریت احراز هویت و کاربران
 - **tasks** برای مدیریت برنامه‌ها و تسک‌ها
 - **event_planner_server** برای تنظیمات کلی و مدیریت سرور هر کدام از این اپلیکیشن‌ها
- فایل‌ها و ماژول‌های خاص خود را دارند که وظایف مربوطه را انجام می‌دهند.

اپلیکیشن authapp

admin.py

این فایل مدیریت نمایش و مدیریت کاربران و پروفایل‌ها در پنل ادمین جنگو را انجام می‌دهد. از مدل‌های User و UserProfile استفاده شده و قابلیت‌های اضافی به ادمین اضافه می‌شود.

apps.py

این فایل پیکربندی اولیه اپلیکیشن را تعریف می‌کند و نام اپلیکیشن authapp را مشخص می‌کند.

custom_password_reset_views.py

این فایل حاوی ویوهای سفارشی برای ریست کردن رمز عبور است. دو ویو مهم در اینجا وجود دارد:

- **CustomPasswordResetView**: برای شروع فرآیند ریست رمز عبور

- CustomPasswordResetConfirmView: برای تأیید و تکمیل فرآیند ریست رمز عبور

models.py

مدل UserProfile برای ذخیره اطلاعات اضافی کاربران ایجاد شده است. این مدل شامل فیلدهایی مانند user و raw_password است.

serializers.py

سریالایزرها برای تبدیل داده‌ها بین اشیاء پایگاه داده و فرمت JSON استفاده می‌شوند. در اینجا سریالایزهای مختلفی برای ثبت‌نام، ورود و ریست رمز عبور تعریف شده‌اند.

tests

تست‌های مختلفی برای بررسی عملکرد صحیح ویوهای مربوط به احراز هویت نوشته شده‌اند. این تست‌ها شامل تست‌های مربوط به ثبت‌نام، ورود و ریست رمز عبور هستند.

urls.py

این فایل شامل آدرس‌های URL مربوط به احراز هویت است. هر ویو با یک آدرس URL مشخص مرتبط شده است.

views.py

این فایل شامل ویوهای مربوط به احراز هویت است. ویوهای LoginView و RegisterView به ترتیب برای ثبت‌نام و ورود کاربران استفاده می‌شوند.

اپلیکیشن tasks

admin.py

این فایل مدیریت نمایش و مدیریت تسک‌ها و برنامه‌ها در پنل ادمین جنگو را انجام می‌دهد. از مدل‌های Task و Plan استفاده شده و قابلیت‌های اضافی به ادمین اضافه می‌شود.

apps.py

این فایل پیکربندی اولیه اپلیکیشن را تعریف می‌کند و نام اپلیکیشن tasks را مشخص می‌کند.

consumers.py

این فایل حاوی کد مربوط به وبسوکت‌ها است. NotificationConsumer برای ارسال نوتیفیکیشن‌های لحظه‌ای به کاربران استفاده می‌شود.

models.py

مدل‌های Plan و Task برای مدیریت برنامه‌ها و تسک‌ها ایجاد شده‌اند. مدل Notification برای ذخیره نوتیفیکیشن‌های کاربران استفاده می‌شود.

routing.py

این فایل شامل آدرس‌های URL مربوط به وبسوکت‌ها است. از re_path برای تعریف الگوهای URL استفاده شده است.

serializers.py

سریالایزرها برای تبدیل داده‌ها بین اشیاء پایگاه داده و فرمت JSON استفاده می‌شوند. در اینجا سریالایزرهای مختلفی برای تسک‌ها، برنامه‌ها و نوتیفیکیشن‌ها تعریف شده‌اند.

tasks.py

این فایل شامل وظایف مربوط به Celery است. وظیفه check_task_deadlines برای بررسی ددلاین‌های تسک‌ها و ارسال نوتیفیکیشن‌ها به کاربران استفاده می‌شود.

urls.py

این فایل شامل آدرس‌های URL مربوط به اپلیکیشن tasks است. هر ویو با یک آدرس URL مشخص مرتبط شده است.

views.py

این فایل شامل ویوهای مربوط به مدیریت تسک‌ها، برنامه‌ها و نوتیفیکیشن‌ها است. ویوهای مختلفی برای لیست کردن، ایجاد، به‌روزرسانی و حذف تسک‌ها و برنامه‌ها تعریف شده‌اند.

tests

تست‌های مختلفی برای بررسی عملکرد صحیح ویوها، مدل‌ها و سریالایزرهای مربوط به تسک‌ها نوشته شده‌اند. این تست‌ها شامل تست‌های مربوط به ایجاد، به‌روزرسانی، حذف و لیست کردن تسک‌ها و برنامه‌ها هستند.

تنظیمات پروژه (settings.py)

تنظیمات کلی

در این فایل تنظیمات کلی پروژه مانند SECRET_KEY، DEBUG و ALLOWED_HOSTS تعریف شده‌اند.

تنظیمات پایگاه داده

از کتابخانه dj_database_url برای پیکربندی پایگاه داده PostgreSQL استفاده شده است. اطلاعات مربوط به پایگاه داده از فایل env خوانده می‌شوند.

تنظیمات احراز هویت و JWT

تنظیمات مربوط به احراز هویت و JWT در اینجا تعریف شده‌اند. از کتابخانه rest_framework_simplejwt برای مدیریت توکن‌های JWT استفاده شده است.

تنظیمات Celery

تنظیمات مربوط به Celery برای مدیریت وظایف پس‌زمینه در اینجا تعریف شده‌اند. از Redis به عنوان Backend و Broker استفاده شده است.

تنظیمات Channels

تنظیمات مربوط به Channels برای مدیریت ارتباطات وب‌سوکت‌ها در اینجا تعریف شده‌اند.

الگوریتم زمان‌بندی تسک‌ها

الگوریتم زمان‌بندی تسک‌ها بر اساس اولویت و ددلاین تسک‌ها عمل می‌کند. این الگوریتم تسک‌ها را بر اساس نسبت اولویت به زمان باقی‌مانده تا ددلاین مرتب می‌کند. تسک‌هایی که زودتر ددلاین دارند و اولویت بالاتری دارند، در اولویت بالاتری قرار می‌گیرند.

سرویس نوتیفیکیشن

از Celery برای مدیریت وظایف پس‌زمینه و ارسال نوتیفیکیشن‌های زمان‌بندی‌شده استفاده شده است. وظیفه check_task_deadlines به صورت دوره‌ای اجرا می‌شود و تسک‌هایی که ددلاین

نزدیک دارند را شناسایی کرده و نوتیفیکیشن‌های مربوطه را برای کاربران ارسال می‌کند. نوتیفیکیشن‌ها از طریق وب‌سوکت‌ها به کاربران ارسال می‌شوند.

پایگاه داده

پایگاه داده پروژه PostgreSQL است که یکی از قوی‌ترین و محبوب‌ترین سیستم‌های مدیریت پایگاه داده رابطه‌ای است. جداول مختلفی برای ذخیره اطلاعات کاربران، پروفایل‌ها، برنامه‌ها، تسک‌ها و نوتیفیکیشن‌ها در این پایگاه داده وجود دارد.

کامپوننت‌های سیستم

- **وب‌سوکت‌ها:** برای ارسال نوتیفیکیشن‌های لحظه‌ای به کاربران استفاده می‌شود.
- **Celery:** برای مدیریت وظایف پس‌زمینه و ارسال نوتیفیکیشن‌ها استفاده می‌شود.
- **Redis:** به عنوان Backend و Broker برای Celery استفاده می‌شود.
- **Django Rest Framework:** برای ایجاد API‌های RESTful استفاده می‌شود.
- **SimpleJWT:** برای مدیریت احراز هویت مبتنی بر توکن‌های JWT استفاده می‌شود.

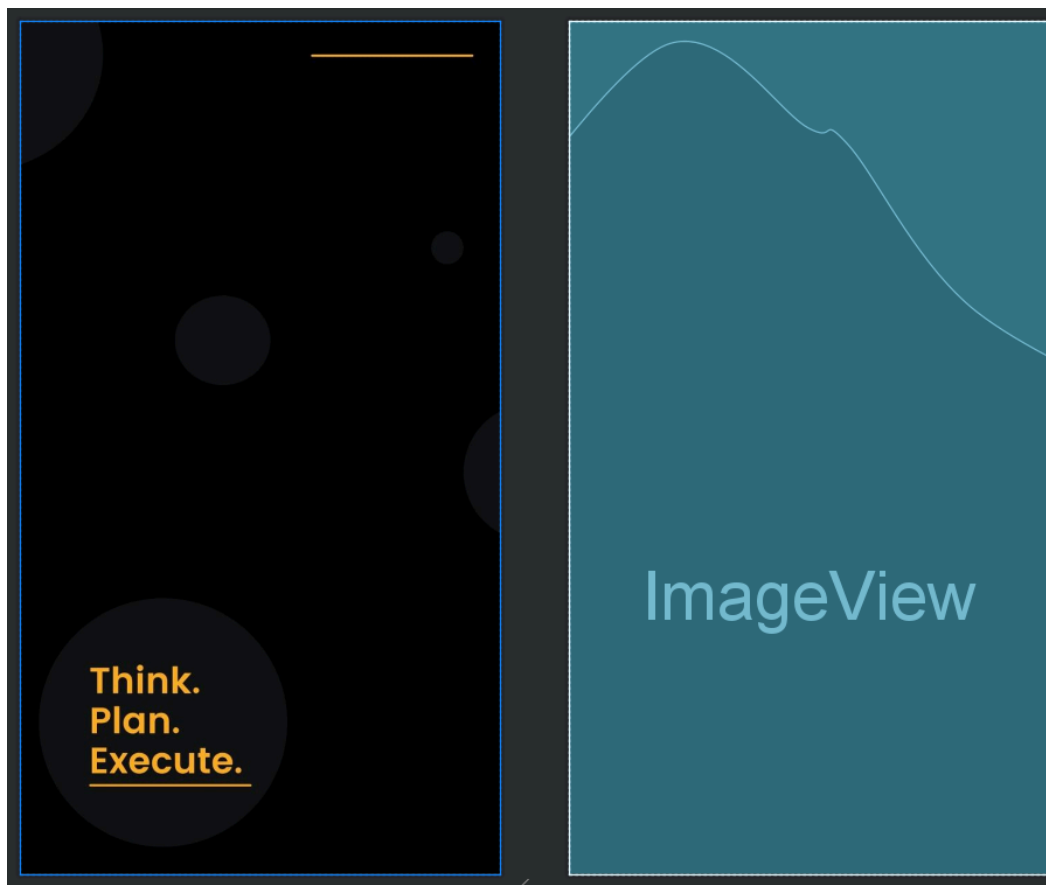
این توضیحات شامل جزئیات کاملی از ساختار کلی پروژه، تنظیمات، الگوریتم‌ها و سرویس‌های مختلفی است که در پروژه سرور ما استفاده شده‌اند.

توضیحات مربوط به پیاده سازی کلاینت (استفاده از java, xml)

رابط کاربری

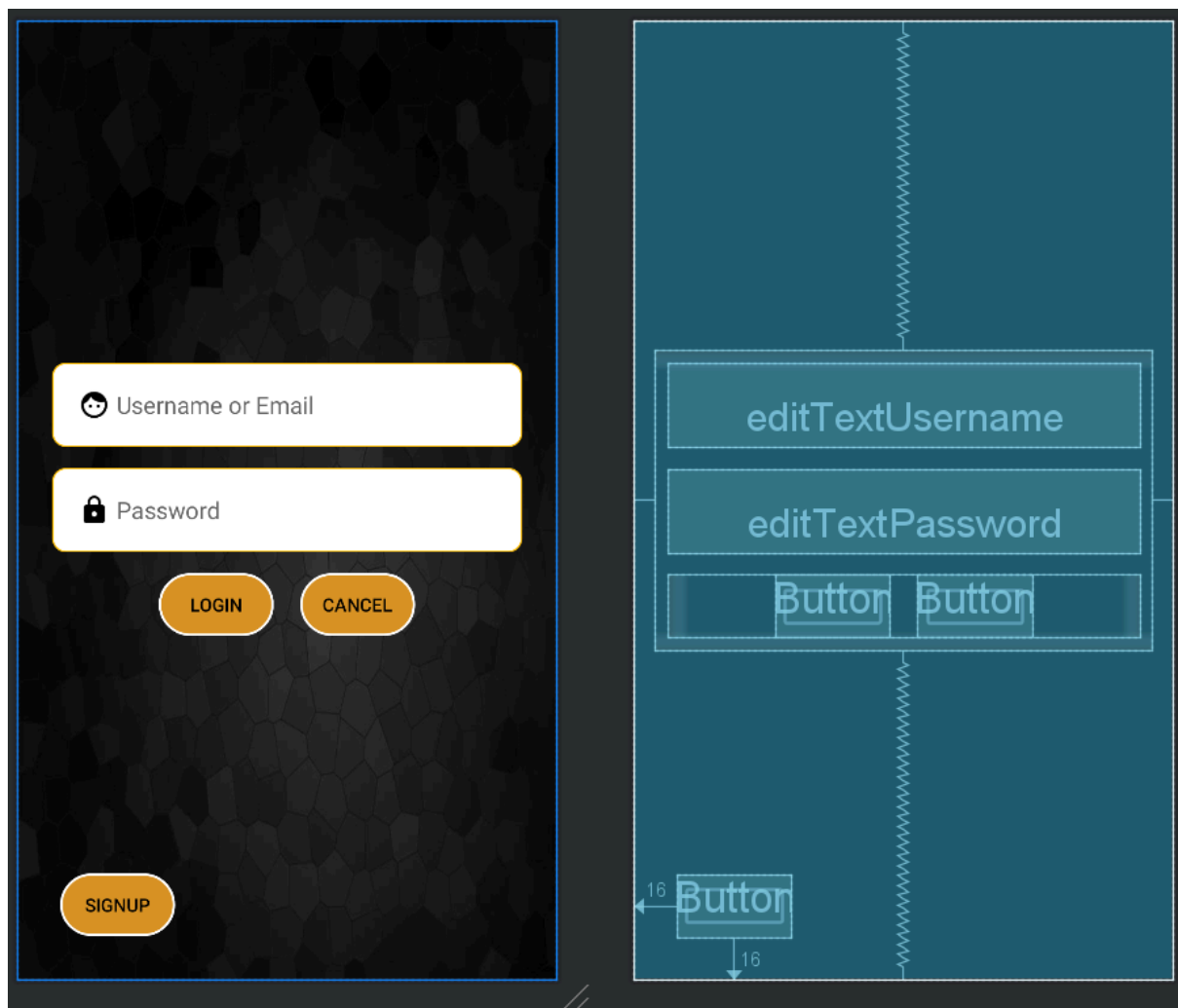
Splash Screen

در این بخش قرار است تمام فایل های xml را نشان دهیم و از ترتیبی که در اجرای برنامه دارند استفاده میکنیم. در مرحله اول برای چند ثانیه activity_splash به عنوان main launcher اجرا میشود و به کاربر برای چند ثانیه نشان داده میشود.



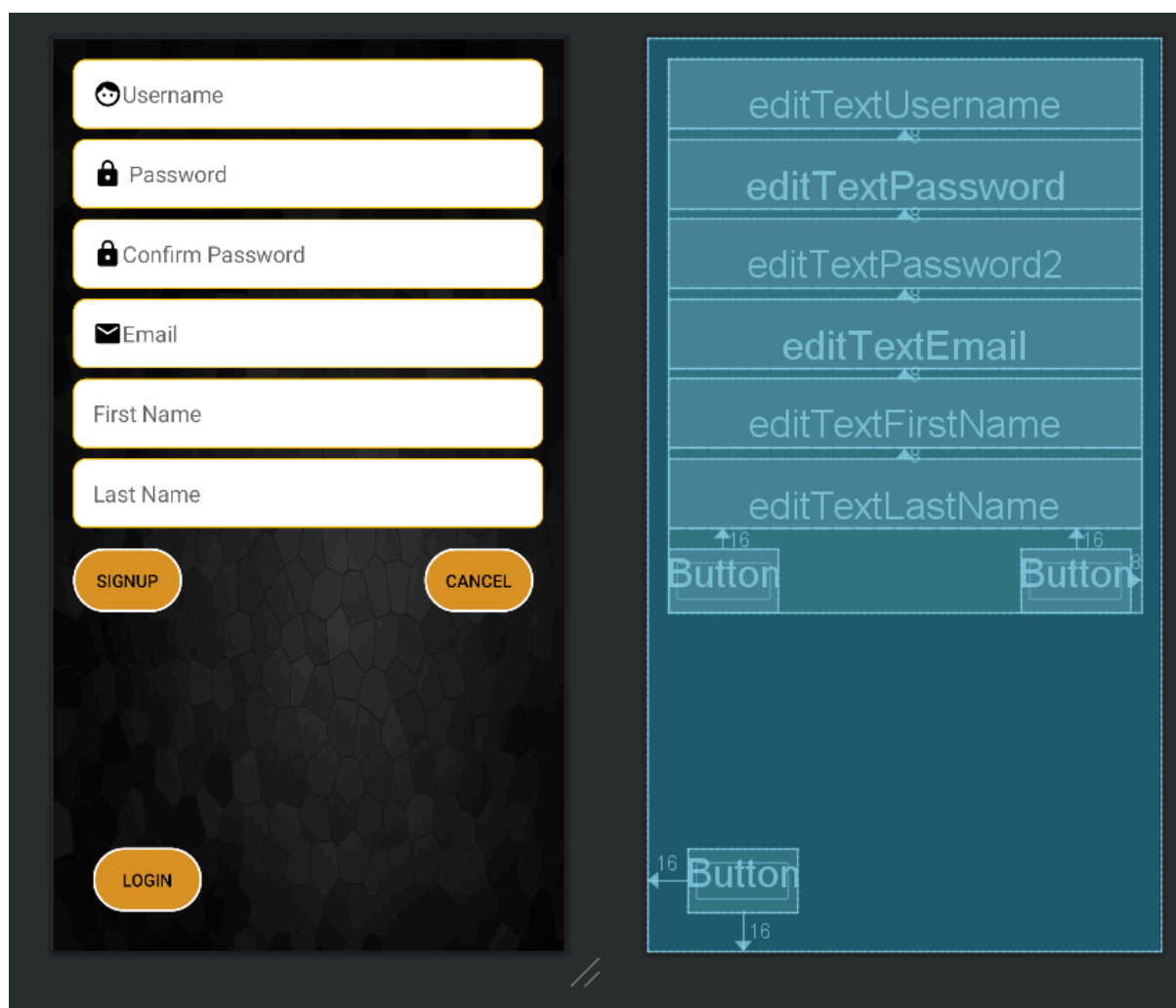
LoginActivity

پس از نمایش activity_splash.xml به کاربر برنامه اندروید با ساخت Intent (با استفاده از دو کلاس ActivityEnum و ActivityChanger) به LoginsActivity رفته و activity_login.xml را به کاربر نشان میدهد تا کاربر login کند.

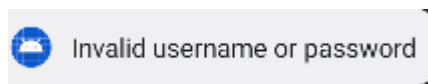


SignupActivity

در صورتی که کاربر حسابی در برنامه تا به حال نداشته با استفاده از دکمه Signup به صفحه ثبت نام رفته تا در آن ثبت نام کند. (activity_signup برای کاربر به نمایش گذاشته میشود)

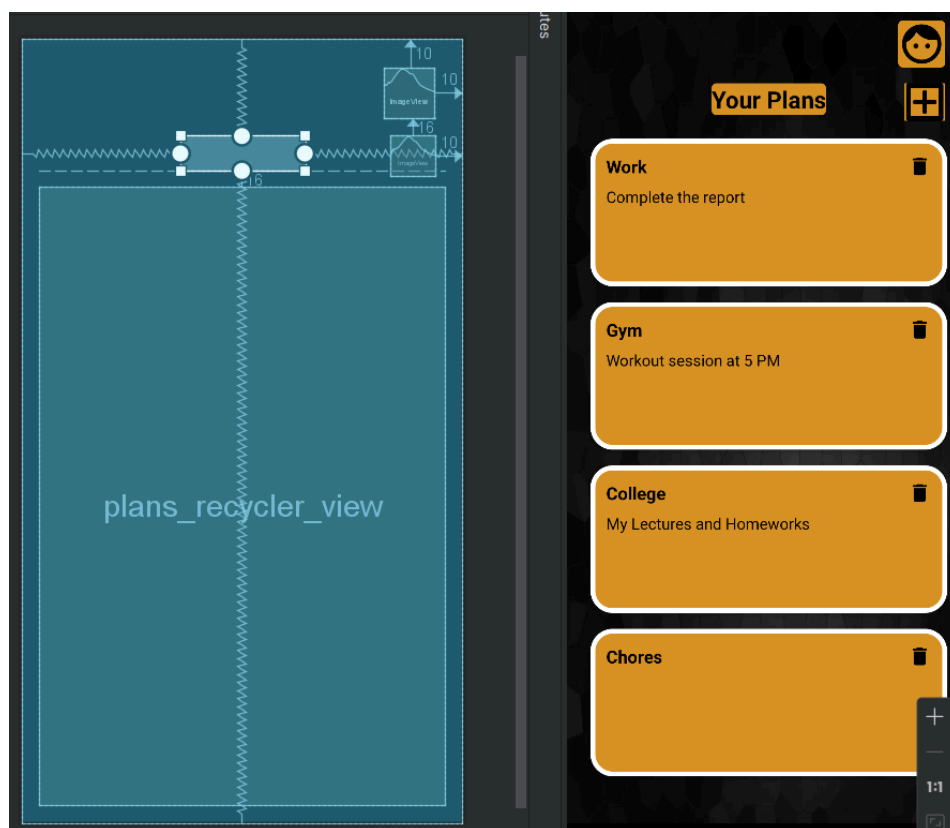


در صورت ثبت نام صحیح، کاربر به Login منتقل میشود تا Login کند. در این دو activity پس از هر عملیات موفق یا ناموفق به صورت Toast برای کاربر به نمایش گذاشته میشود. به عنوان مثال:



MainActivity

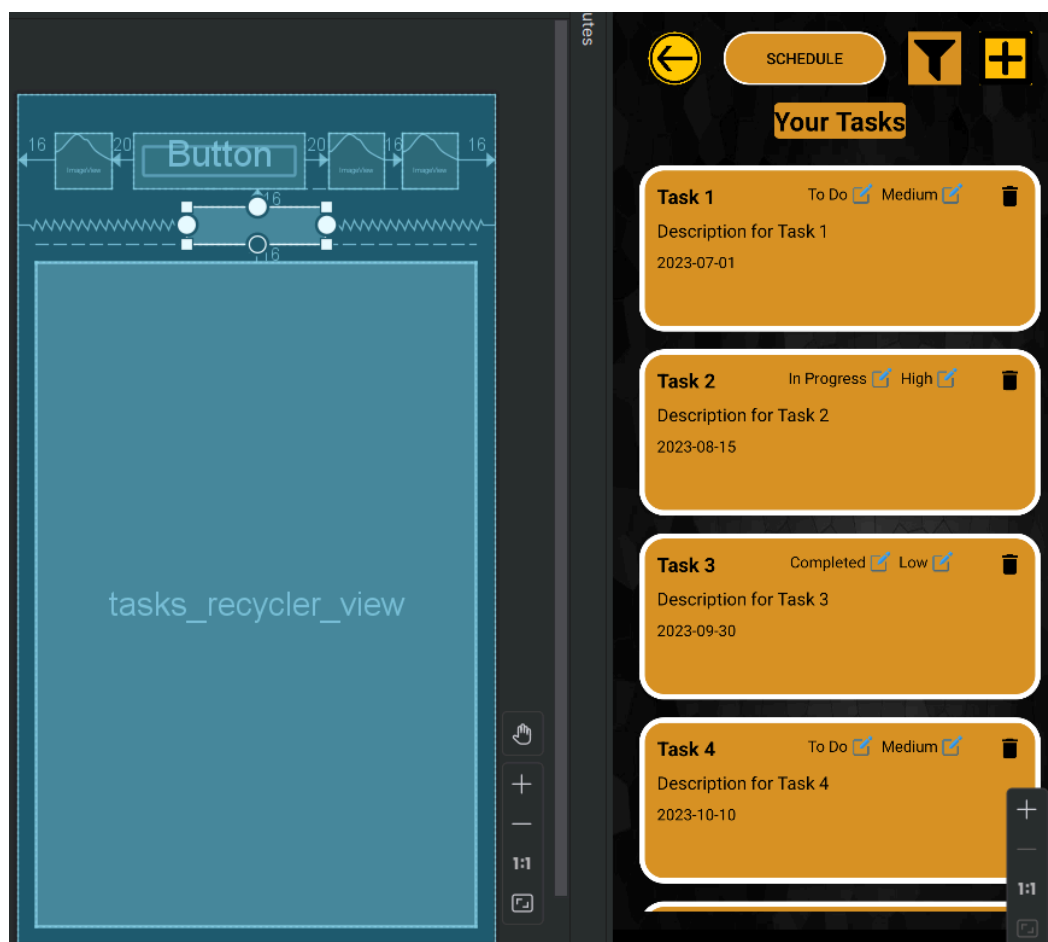
پس از ثبت نام و ورود به حساب کاربری توسط کاربر از LoginActivity به MainActivity میرویم و در آنجا Plan های کاربر در یک RecyclerView به عنوان item_plan به نمایش گذاشته میشوند.



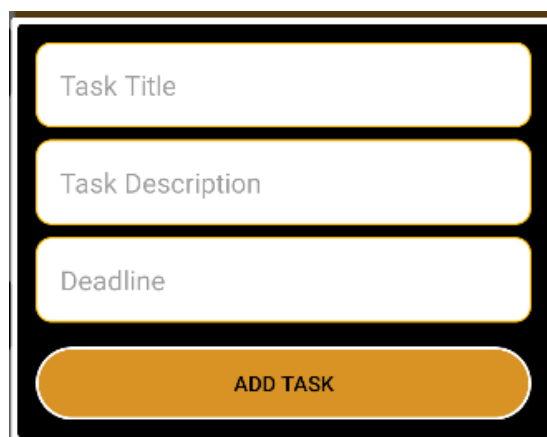
که یک دکمه برای اضافه کردن Plan است. هر Plan شامل یک Title و یک Description است. پس از فشردن دکمه اضافه یک popMenu به شکل زیر بالا خواهد آمد و کاربر با وارد کردن مشخصات Plan مورد نظر خود، آن را به Plan ها اضافه خواهد کرد.

TasksActivity

پس از آنکه کاربر یکی از Plan ها را کلیک کرد به داخل آن Plan میرویم و تسک های مربوط به آن Plan همانند Plan ها به عنوان item_task در RecyclerView لیست میشوند.



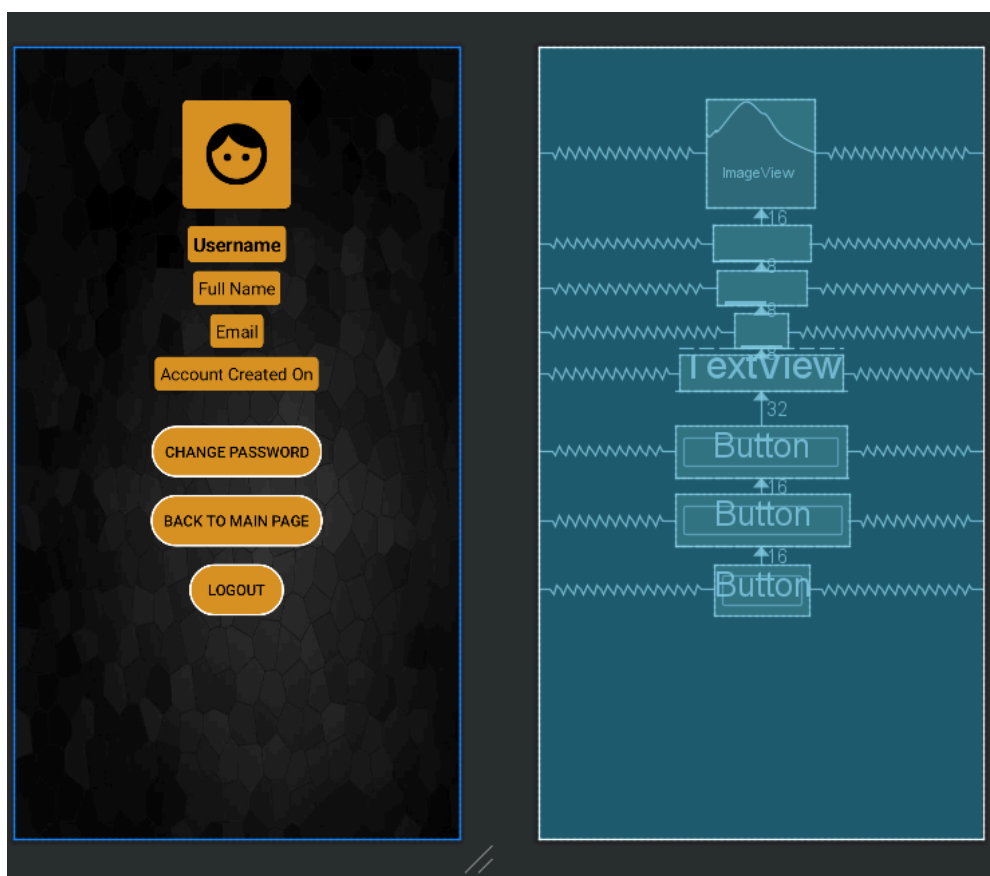
کاربر توانایی Delete کردن، اضافه کردن و عوض کردن Status و Priority تسک ها را دارد. همچنین یک دکمه برای Filter کردن تسک های به نمایش گذاشته شده بر اساس Status آنها وجود دارد. هر تسک شامل 5 ویژگی است که عبارتند از Title، Description، Deadline، که در هنگام اضافه کردن یک تسک الزامی هستند؛ و دو ویژگی دیگر نیز همان Status و Priority هستند. دقیقاً همانند add plan پیاده سازی شده.



A task form with a black background and yellow rounded rectangular input fields. The fields are labeled 'Task Title', 'Task Description', and 'Deadline'. Below the fields is a yellow rounded rectangular button labeled 'ADD TASK'.

UserProfileActivity

اگر در صفحه Plans باشیم و کاربر بر روی آیکون user profile کلیک کند وارد صفحه profile میشویم.



ScheduledTasksActivity

اگر در صفحه تسک ها باشیم و کاربر بر روی دکمه Schedule بزند به صفحه ای میرویم که در آن تسک های Schedule شده برای ما به نمایش گذاشته میشود.

