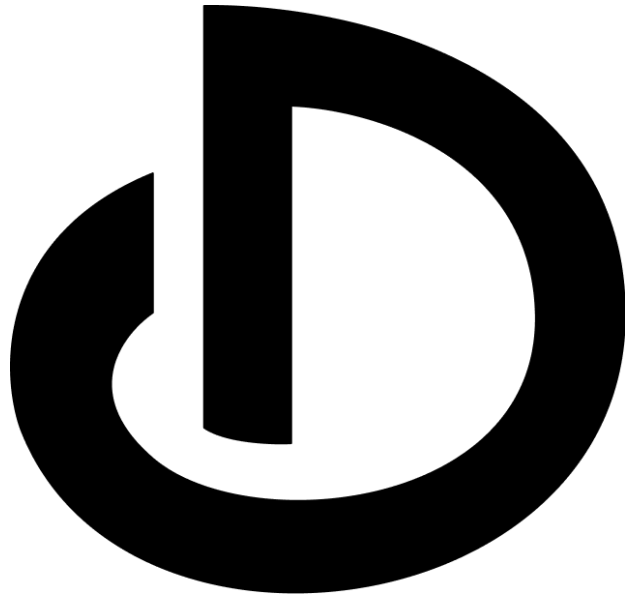


گزارش فنی پروژه‌ی دُنْگِی



اعضای تیم

محمدامین لطفی

محمد مهدی عبدی

علیرضا کاظمینی

استاد

مهرداد احمدی نیک

دانشگاه صنعتی شریف

۱۳ تیر ۱۴۰۳

فهرست

3.....	هدف پروژه.....
4.....	معماری سیستم.....
5.....	قابلیت‌های سیستم.....
6.....	تصاویر از رابط کاربری برنامه.....
9.....	کارهای تکمیل شده.....
9.....	1. شروع پروژه.....
9.....	2. اتصال به سرور.....
9.....	3. درخواست HTTP.....
9.....	4. صفحه شروع.....
9.....	5. صفحه خوش‌آمدگویی.....
10.....	6. صفحه ورود.....
10.....	7. صفحه ثبت‌نام.....
10.....	8. صفحه لیست گروه‌ها.....
10.....	9. صفحه افزودن گروه.....
11.....	10. صفحه پروفایل.....
11.....	11. خروج از حساب.....
11.....	12. مدیریت هزینه‌ها.....
11.....	13. لیست دعوت‌ها (تایید یا رد دعوت).....
12.....	پیوندها.....

هدف پروژه

پروژه دُنْگِی با هدف ایجاد یک ابزار کارآمد و ساده برای مدیریت و تقسیم هزینه‌ها بین افراد طراحی شده است. این اپلیکیشن به کاربران این امکان را می‌دهد که هزینه‌ها را بین گروهی از افراد تقسیم کنند و وضعیت پرداخت‌ها را دنبال کنند.

معماری سیستم

کلاینت

اندروید Kotlin

چرا انتخاب شد: Kotlin به دلیل قابلیت‌های پیشرفته، ایمنی در مقابل خطاهای NullPointerException و سازگاری کامل با کتابخانه‌های جاوا، انتخاب شده است. همچنین، Kotlin به دلیل سینتکس مدرن و خوانایی بالا برای توسعه اپلیکیشن‌های اندرویدی مناسب است.

رابط کاربری (UI): طراحی شده با XML و اجرای منطق برنامه با استفاده از Kotlin.
ارتباط با سرور: از Retrofit برای ارسال و دریافت داده‌ها از سرور استفاده می‌شود.
مدیریت توکن: استفاده از SharedPreferences برای ذخیره و مدیریت توکن احراز هویت.

سرور:

Node.js

چرا انتخاب شد: Node.js به دلیل قابلیت هندل کردن تعداد زیادی درخواست همزمان و کارایی بالا، انتخاب شده است. همچنین، Node.js با استفاده از JavaScript که یک زبان محبوب و پرکاربرد است، توسعه و نگهداری آسان‌تری دارد.

Node.js: به عنوان محیط اجرایی سرور.
Express.js: برای مدیریت درخواست‌ها و مسیرهای API.
JWT: برای مدیریت احراز هویت و امنیت.

پایگاه داده:

پایگاه داده مورد استفاده در این پروژه MongoDB است. این پایگاه داده غیر رابطه‌ای به دلیل انعطاف‌پذیری و قابلیت گسترش آسان، انتخاب شده است.

قابلیت‌های سیستم

تقسیم هزینه‌ها بین افراد:

- کاربران می‌توانند گروه‌هایی ایجاد کنند و هزینه‌ها را بین اعضای گروه تقسیم کنند.

مدیریت دعوتنامه‌ها:

- ارسال دعوتنامه برای اضافه کردن اعضای جدید به گروه.
- پذیرش یا رد دعوتنامه‌ها توسط کاربران.

مدیریت پروفایل:

- کاربران می‌توانند اطلاعات پروفایل خود را مشاهده و ویرایش کنند.

احراز هویت و امنیت:

- استفاده از JWT برای احراز هویت کاربران.
- رمزنگاری داده‌های حساس در سرور.

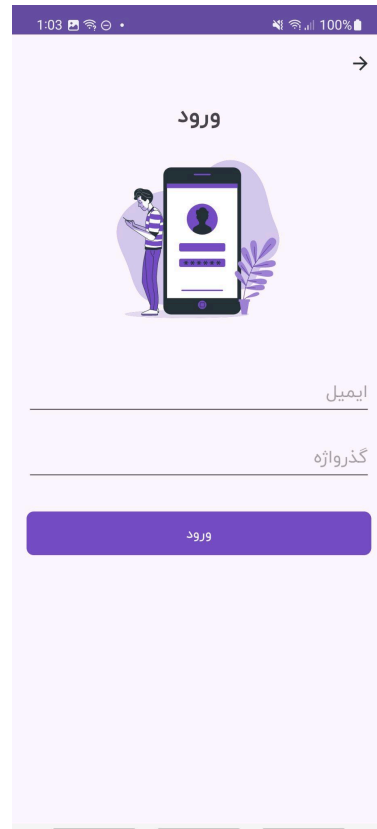
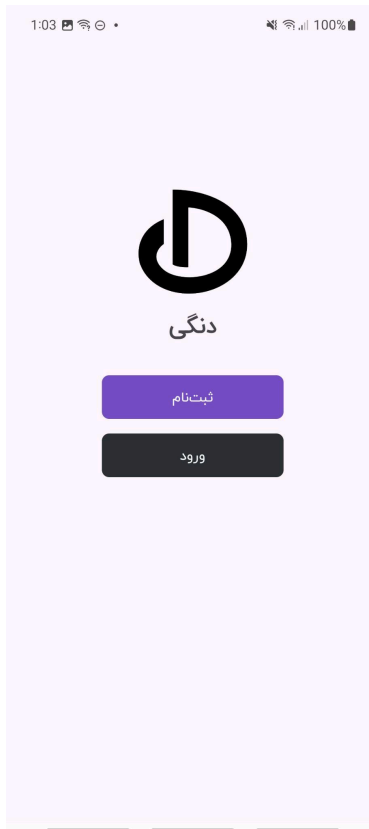
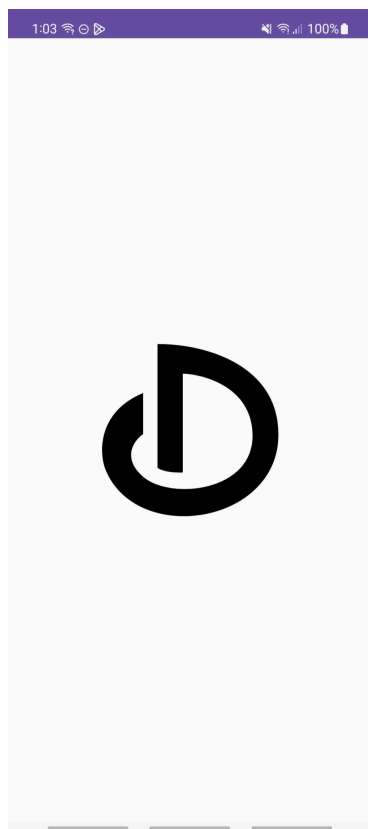
ارتباط با سرور:

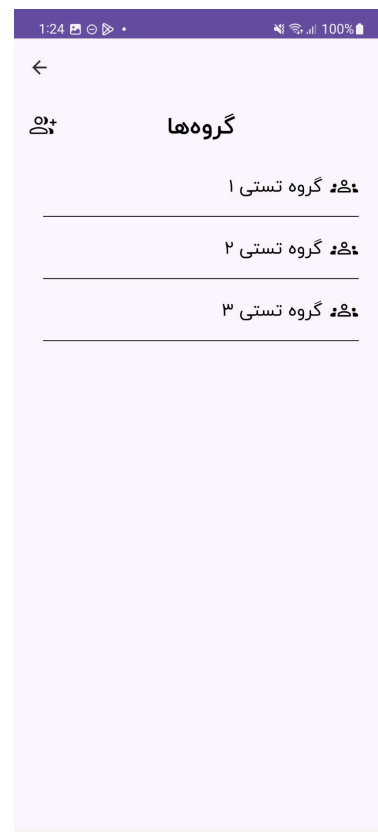
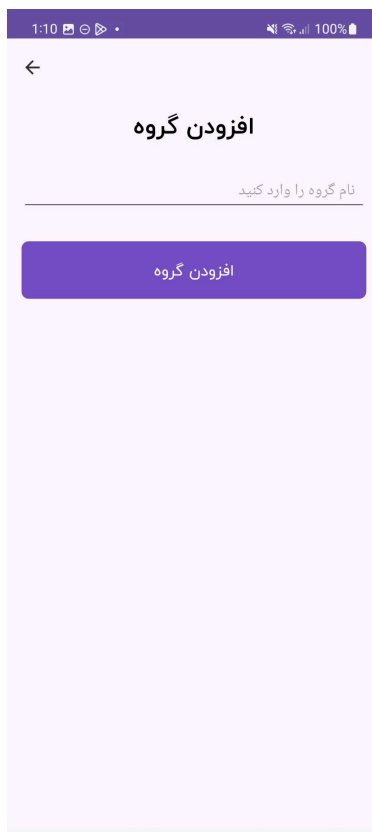
- استفاده از Retrofit برای ارتباط با API‌های سرور.

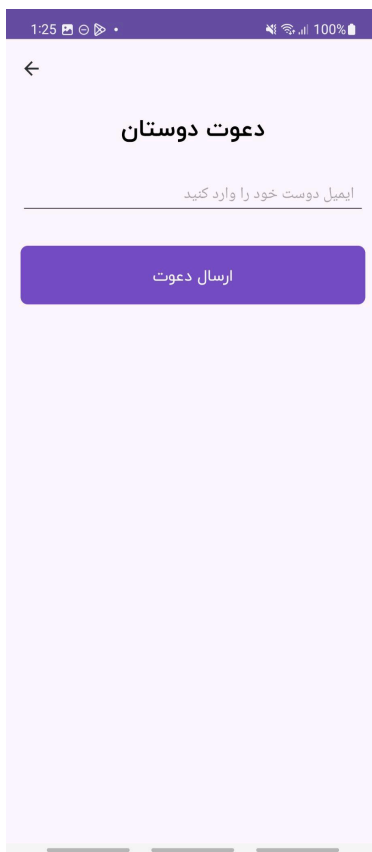
رابط کاربری کاربرپسند:

- طراحی رابط کاربری ساده و کاربرپسند برای استفاده راحت‌تر کاربران.

تصاویر از رابط کاربری برنامه







کارهای تکمیل شده

1. شروع پروژه

- توضیحات: تنظیم و پیکربندی اولیه پروژه اندروید.
- پیاده‌سازی:
 - ایجاد یک پروژه جدید اندروید در Android Studio.
 - پیکربندی تنظیمات پروژه از جمله نسخه SDK، و وابستگی‌ها.
 - تنظیم کنترل نسخه با استفاده از Git.

2. اتصال به سرور

- توضیحات: برقراری اتصال بین برنامه اندروید و سرور.
- پیاده‌سازی:
 - ادغام Retrofit به عنوان کلاینت HTTP برای مدیریت درخواست‌های API.
 - پیکربندی Retrofit با آدرس سرور.
 - افزودن مجوزهای لازم برای دسترسی به اینترنت در `AndroidManifest.xml`.

3. درخواست HTTP

- توضیحات: مدیریت درخواست‌های HTTP برای تعامل با سرور.
- پیاده‌سازی:
 - تعریف مدل‌های داده برای بدنه‌های درخواست و پاسخ.
 - ایجاد یک رابط `ApiService` با روش‌هایی برای ورود، ثبت‌نام، دریافت گروه‌ها، و افزودن گروه.
 - پیاده‌سازی مدیریت خطا برای درخواست‌ها و پاسخ‌های شبکه.

4. صفحه شروع

- توضیحات: صفحه شروع اولیه که هنگام راه‌اندازی برنامه ظاهر می‌شود.
- پیاده‌سازی:
 - ایجاد یک `SplashActivity` برای نمایش صفحه شروع.
 - اعمال یک تأخیر برای نمایش صفحه شروع برای مدت کوتاهی.
 - پس از تأخیر، توکن احراز هویت را در `SharedPreferences` بررسی می‌کند و به صفحه مناسب هدایت می‌شود (`GroupsActivity` اگر توکن موجود باشد، `WelcomeActivity` اگر نباشد).

5. صفحه خوش‌آمدگویی

- توضیحات: این صفحه گزینه‌هایی برای ورود یا ثبت‌نام به کاربر ارائه می‌دهد.
- پیاده‌سازی:
 - ایجاد یک `WelcomeActivity` با دو دکمه: "ورود" و "ثبت‌نام".

- ناولبری به `LoginActivity` و `SignUpActivity` بر اساس انتخاب کاربر مدیریت می‌شود.

6. صفحه ورود

- توضیحات: به کاربران اجازه می‌دهد به حساب‌های خود وارد شوند.
- پیاده‌سازی:
 - ایجاد یک `LoginActivity` با فیلدهایی برای ایمیل و رمز عبور.
 - اعتبارسنجی ورودی کاربر انجام می‌شود.
 - پس از ورود موفق، توکن احراز هویت در `SharedPreferences` ذخیره می‌شود و کاربر به `GroupsActivity` هدایت می‌شود.

7. صفحه ثبت‌نام

- توضیحات: به کاربران جدید اجازه می‌دهد حساب کاربری ایجاد کنند.
- پیاده‌سازی:
 - ایجاد یک `SignUpActivity` با فیلدهایی برای نام، نام خانوادگی، ایمیل و رمز عبور.
 - اعتبارسنجی ورودی کاربر انجام می‌شود.
 - پس از ثبت‌نام موفق، توکن احراز هویت در `SharedPreferences` ذخیره می‌شود و کاربر به `GroupsActivity` هدایت می‌شود.

8. صفحه لیست گروه‌ها

- توضیحات: لیستی از گروه‌هایی که کاربر عضو آن‌هاست را نمایش می‌دهد.
- پیاده‌سازی:
 - ایجاد یک `GroupsActivity` با یک `RecyclerView` برای نمایش لیست گروه‌ها.
 - داده‌ها از سرور با استفاده از Retrofit دریافت می‌شود و لیست به صورت پویا به‌روز می‌شود.
 - یک آیکن "افزودن گروه" به نوار ابزار اضافه شد تا به `AddGroupActivity` هدایت شود.

9. صفحه افزودن گروه

- توضیحات: به کاربران اجازه می‌دهد یک گروه جدید ایجاد کنند.
- پیاده‌سازی:
 - ایجاد یک `AddGroupActivity` با فیلدی برای وارد کردن نام گروه.
 - یک `Toolbar` برای ناولبری و نمایش عنوان اضافه شد.
 - API ایجاد گروه فراخوانی می‌شود و پس از موفقیت، کاربر به `GroupDetailsActivity` هدایت می‌شود.

10. صفحه پروفایل

- توضیحات: این ویژگی به کاربران اجازه می‌دهد اطلاعات پروفایل خود را مشاهده و ویرایش کنند.

- پیاده‌سازی:

- ایجاد یک `ProfileActivity` با فیلدهایی برای نمایش و به‌روزرسانی اطلاعات کاربر.
- ادغام API برای دریافت و به‌روزرسانی داده‌های پروفایل.

11. خروج از حساب

- توضیحات: این ویژگی به کاربران اجازه می‌دهد از حساب‌های خود خارج شوند و توکن احراز هویت را پاک کنند.

- پیاده‌سازی:

- افزودن دکمه خروج در `ProfileActivity`.
- پیاده‌سازی قابلیت پاک کردن توکن از `SharedPreferences` و هدایت کاربر به `WelcomeActivity`.

12. مدیریت هزینه‌ها

- توضیحات: این ویژگی به کاربران اجازه می‌دهد هزینه‌های خود را مدیریت کنند.

- پیاده‌سازی:

- ایجاد یک صفحه برای مشاهده و افزودن هزینه‌ها.
- ادغام API برای دریافت و مدیریت داده‌های هزینه.

13. لیست دعوت‌ها (تایید یا رد دعوت)

- توضیحات: این ویژگی به کاربران اجازه می‌دهد دعوت‌ها را مشاهده و تایید یا رد کنند.

- پیاده‌سازی:

- ایجاد یک صفحه برای نمایش لیستدعوت‌ها.
- ادغام API برای دریافت لیست دعوت‌ها.
- افزودن قابلیت تایید یا رد دعوت‌ها با استفاده از دکمه‌های مناسب در هر آیتم لیست.
- ارسال درخواست به سرور برای تایید یا رد دعوت و به‌روزرسانی وضعیت در لیست.

پیوندها

- [مخزن پروژه‌ی اندروید دُنْگِی](#)
- [مخزن پروژه‌ی Node.js دُنْگِی](#) + (مستندات API)
- [برد کانبان](#)