Техническая документация

Основные возможности

- Пагинация списка актуальных игровых предложений (60 элементов на страницу).

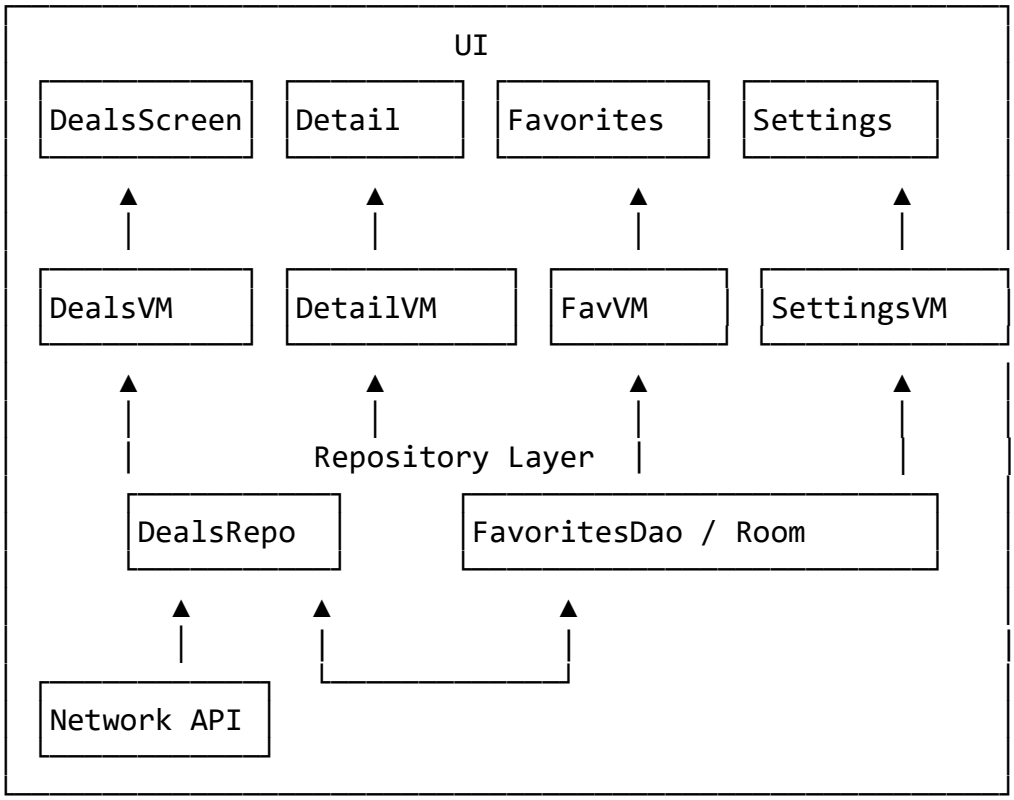
- Клиент-сайд фильтрация по диапазону цены и сортировка по возрастанию/убыванию.
- Поиск по названию, выполняющийся на сервере CheapShark.
- Отображение деталей игры и цен во всех магазинах.
- Избранное: добавление/удаление сделок, хранение в локальной БД Room.
- Фоновая периодическая проверка по заданному интервалу и порогу скидки через WorkManager → нотификации.
- Настройки: минимальный порог скидки и частота проверки.
- Темная тема, кастомная цветовая палитра, шрифт Montserrat.

Технологии и зависимости

- Kotlin 2.0
- Jetpack Compose 1.6
- AndroidX Room 2.6
- WorkManager 2.9
- Paging 3.3
- Retrofit 2.11 + kotlinx-serialization
- Coil Compose 2.5
- Koin 3.5
- DataStore Preferences 1.0
- Robolectric + WorkManager-testing + Room-testing для unit-тестов

Архитектура проекта

Используется MVVM + Repository + Clean-style. Модули:



Модульная структура

- data.network — API и DTO
- data.local — Room entities и DAO
- data.prefs — DataStore Preferences (SettingsRepository)
- data.repository — репозитории и PagingSource
- data.worker — PriceCheckWorker
- di — Koin модули
- ui.deals, ui.detail, ui.fav, ui.settings — Composable экранов + ViewModel
- ui.theme — цвета, типографика, формы
- util — StoreNames

Data Layer

Network

- **CheapSharkApi** (Retrofit + kotlinx-serialization)

    - getDeals(pageNumber, pageSize, storeID?, title?): список сделок.
    - getDealDetail(id): детали одной сделки + gameInfo, cheaperStores.
    - getDealsBySteamApp(steamAppID): все предложения по игре.
- **DTO**:
    - DealDto
    - DealDetailDto, GameInfoDto, StoreDealDto

Local

- **Entity**: FavoriteEntity (Room), primaryKey = dealId
- **DAO**: FavoritesDao
    - all(): Flow<List<FavoriteEntity>>
    - add(FavoriteEntity), remove(dealId), find(dealId)
- **Database**: AppDatabase : RoomDatabase

Preferences

- **SettingsRepository**: DataStore Preferences
    - minDiscount: Flow<Int> (дефолт 30%)
    - intervalHours: Flow<Int> (дефолт 6 ч)
    - save(min, hrs)

Repository

- **DealsRepository**
    - getDealsPager(storeId?, title?) : Flow<PagingData<DealDto>>
    - getDealDetail(id): DealDetailDto
    - getDealsForGame(steamAppId): List<DealDto>
    - addToFav(dto), removeFav(id)
    - observeFavorites(): Flow<List<FavoriteEntity>>
- **DealsPagingSource**: Paging3 source, передаёт title для server-side search.

Domain / Use Cases

- **Просмотр списка сделок**
- **Поиск** (server-side по title)

- **Фильтр** (client-side по диапазону цены)
- **Сортировка** (ASC/DESC)
- **Избранное** (добавить/убрать)
- **Переход в детали** и просмотр всех цен
- **Периодическая проверка** (WorkManager → нотификация)
- **Настройки** (порог скидки, период)

UI Layer

Navigation

- AppNavGraph (Compose Navigation)
  - deals
  - settings
  - favorites
  - deal/{dealId}

Скрин Deals (DealsScreen.kt)

- **TopBar**: SearchField (OutlinedTextField), кнопки фильтра, избранное, настройки
- **AnimatedVisibility**: цена-фильтр (RangeSlider)
- **LazyColumn**: DealCard
- Навигация → Detail по Uri.encode(deal.id)

DealCard (DealCard.kt)

- AsyncImage (Coil) с capsule_616x353, ContentScale.Crop, aspectRatio(16/9)
- Заголовок, цены, скидочный Chip, название магазина, кнопка «сердечко»

Скрин Favorites (FavoritesScreen.kt)

- Сортировка по title
- DealCard → onFavClick = removeFavorite
- Пустой кейс «Пусто»

Скрин Settings (SettingsScreen.kt)

- Поля для Мин. скидка и Частота проверки

- Кнопка Save → prefs.save() + enqueuePriceChecker()
- **Debug-кнопка** «Проверить сейчас» → единоразовый WorkRequest

Скрин Detail (DealDetailScreen.kt)

- AsyncImage (capsule_616x353)
- Текущая цена, обычная цена, скидка
- Кнопка «сердечко» (toggleFav)
- LazyColumn всех сделок (storeDeals) во всех магазинах

Dependency Injection

Используется **Koin**:

- networkModule – OkHttpClient, Retrofit, Json
- storageModule – Room, FavoritesDao, SettingsRepository
- workerModule – WorkManager, PriceCheckWorker
- repositoryModule – DealsRepository
- viewModelModule – DealsVM, DetailVM (with SavedStateHandle), FavVM, SettingsVM

Запуск в GameDealsApp.onCreate:

```
startKoin {
 androidContext(this@GameDealsApp)
 modules(
   networkModule,
   repositoryModule,
   viewModelModule,
   storageModule,
   workerModule
 )
}
enqueuePriceChecker(get())
```

Стили и тема

- **Цвета** (ui/theme/Color.kt):
  - Black, CardDarkGrayPurple, Coral/Emerald/Sky
- **Typography** (ui/theme/Type.kt):
  - Google Font Montserrat (regular, semibold, bold)
- **Shapes** (AppShapes): small=12.dp, medium=16.dp
- **Theme** (ui/theme/Theme.kt):
  - Светлая / Тёмная схема с принудительным background = Black

    - dynamicColor = false
    - Transparent status bar, Edge-to-edge

Уведомления и WorkManager

- PriceCheckWorker (CoroutineWorker):
    - doWork():
        - prefs.minDiscount.first()
        - dao.all().first() → список избранного
        - Для каждой: api.getDeals(...).first { it.id == fav.dealId }
        - Если discount ≥ minDiscount && newPrice < oldPrice:
            - notifyPriceDrop(title, newPrice)
            - dao.add(fav.copy(currentPrice = newPrice))
- enqueuePriceChecker(wm, hours) → PeriodicWorkRequest (ExistingPeriodicWorkPolicy.UPDATE)
- NotificationChannel price_drop, NotificationCompat.Builder

Пользовательская документация

GameAggregator — Android-приложение на Jetpack Compose, которое показы-вает скидки на игры с API CheapShark, позволяет искать по названию, фильтровать по цене, добавлять в избранное и получать уведомления о снижении цены.

С его помощью вы можете

- Искать игры по названию среди всех магазинов.
- Фильтровать по диапазону цены и сортировать по возрастанию/убыванию.
- Добавлять понравившиеся предложения в «Избранное».
- Просматривать детальную информацию о каждой игре и сравнивать цены во всех поддерживаемых магазинах.
- Получать push-уведомления, когда цена в «Избранном» падает ниже заданного порога.

Быстрый старт

- **Список сделок**
  - На главном экране отображаются текущие акции (60 элементов на страницу).
  - Проведите вниз для подгрузки следующих позиций.
- **Поиск и фильтры**
  - Введите название игры в поле поиска — приложение выполнит запрос к серверу CheapShark и покажет все подходящие результаты.
  - Нажмите значок «фильтр», чтобы открыть диапазон цен (ползунок).
  - Выберите стрелками «Цена ↑/↓» для сортировки.
- **Избранное**
  - Нажмите на сердце в карточке, чтобы добавить/удалить игру из «Избранного».
  - Перейдите во вкладку «Избранное» (иконка сердца в шапке), чтобы увидеть все сохранённые позиции.
- **Детали игры**
  - Кликните по карточке — откроется экран с обложкой, текущей и обычной ценой, процентом скидки.
  - Ниже — список всех предложений по этой игре в разных магазинах с ценами и скидками.
- **Настройки**
  - Задайте **минимальный порог скидки** и **частоту проверки** (в часах).
  - Кнопка **«Проверить цены сейчас»** отправляет единоразовую проверку.
  - После сохранения настроек приложение автоматически будет проверять «Избранное» в фоне и присылать уведомления.

Уведомления

- Приложение шлёт уведомление «Цена упала!» при снижении цены на любую игру из «Избранного» ниже ранее сохранённого уровня.