Факультет компьютерных наук

Департамент
Программной инженерии

Android приложение для организации студенческой жизни в общежитии

A Mobile App for Organizing Student Life in a Dormitory

Индивидуальный | Программный

Выполнил: Хазеев Амир Айратович, БПИ247
Научный руководитель: Резуник Людмила Александровна
преподаватель департамента «Программной инженерии» ФКН ВШЭ

Москва 2026

Предметная область

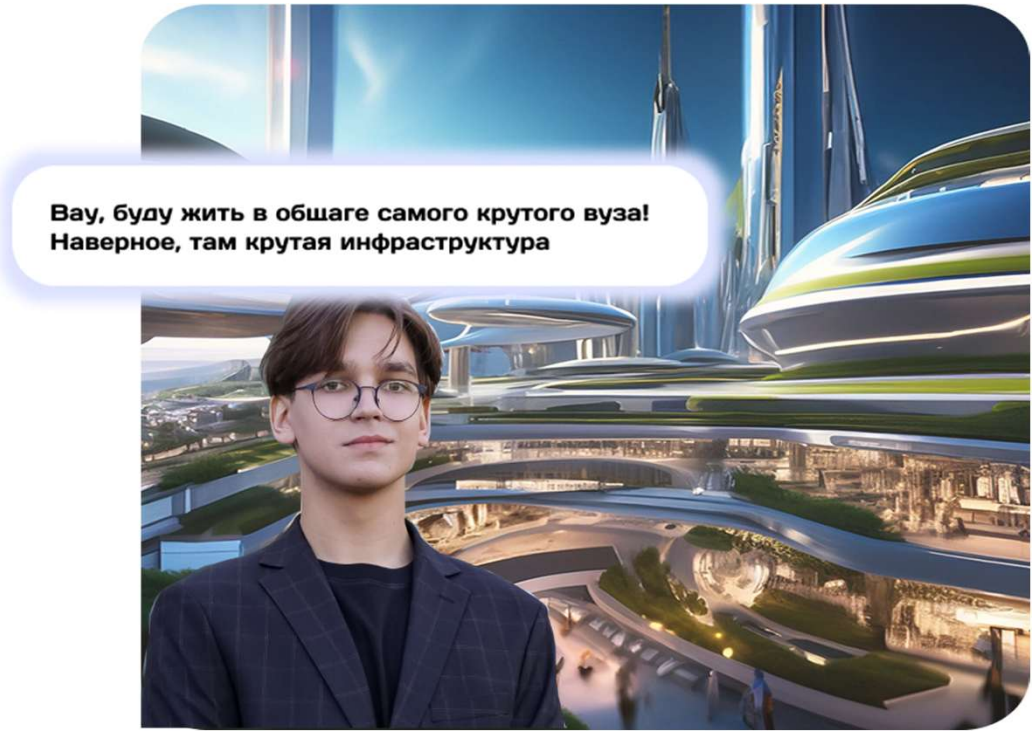
Предметная область проекта — повседневная студенческая жизнь в общежитии.

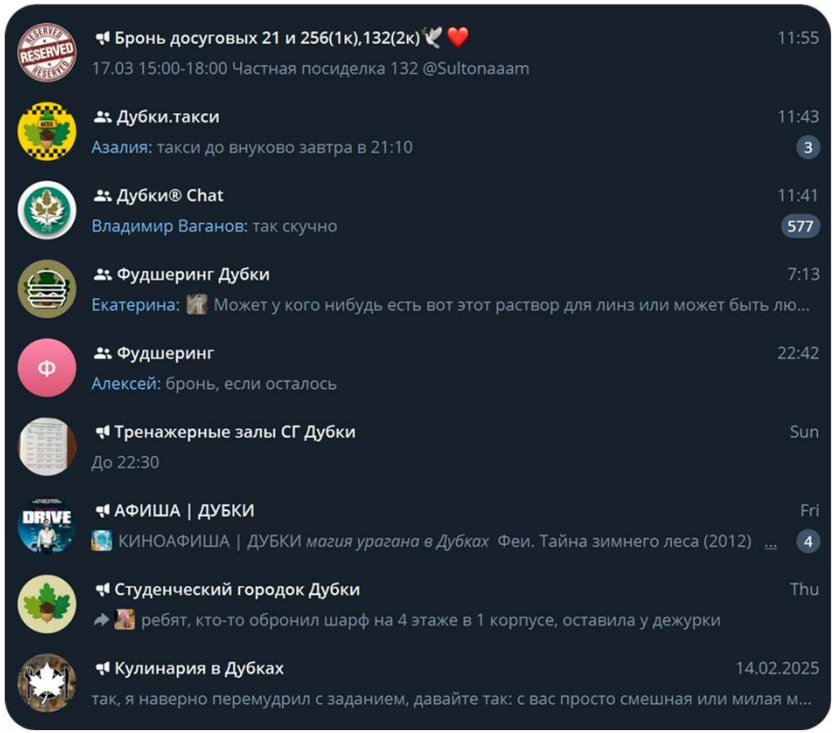
В такой среде у студентов регулярно возникают повторяющиеся задачи: найти вещь для временного использования, предложить свою вещь другим жильцам, проверить занятость общей комнаты или создать бронь.

Поэтому приложение сфокусировано на трех направлениях:

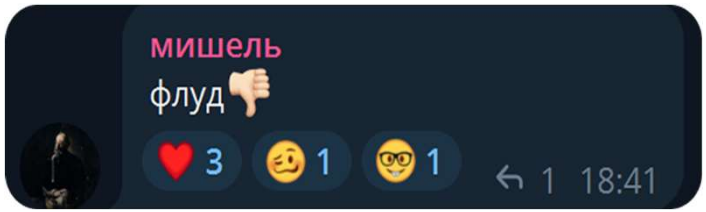
- Шеринг вещей
- Бронирование помещений
- Работа с профилем пользователя

Ожидание \ Реальность

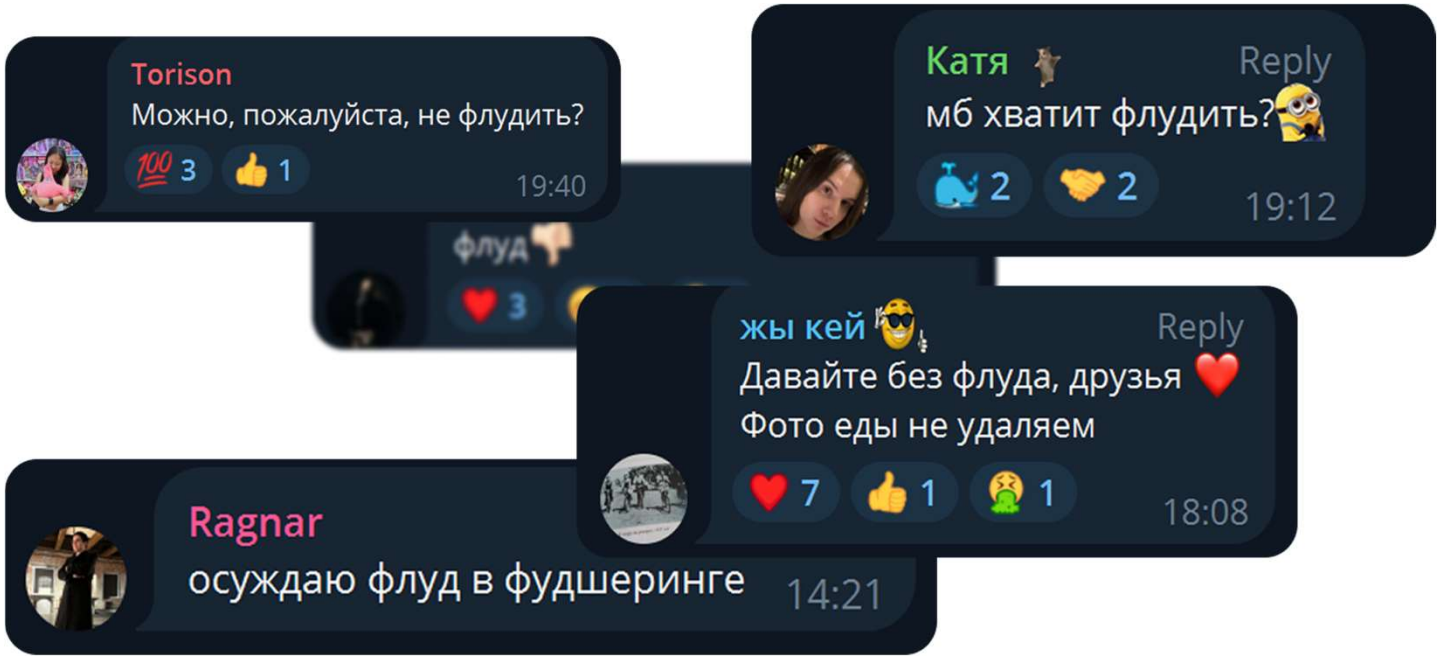




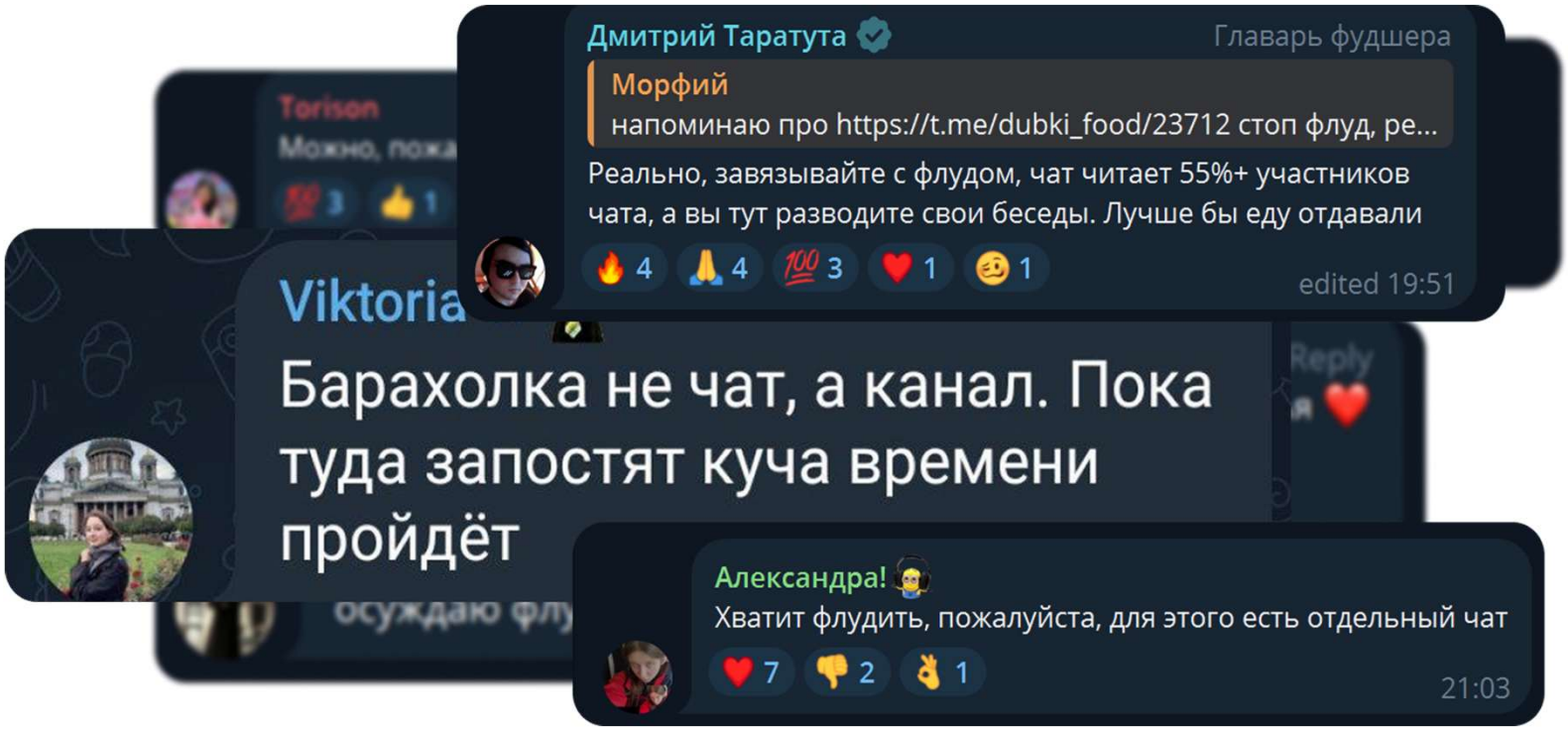
Что не так?



Что не так?


Torison
Можно, пожалуйста, не флудить?
💯 3 👍 1
19:40
Катя
Reply
мб хватит флудить?
2 2
19:12
флуд
3
жы кей
Reply
Давайте без флуда, друзья ❤️
Фото еды не удаляем
7 👍 1 🤮 1
18:08
Ragnar
осуждаю флуд в фудшеринге
14:21

Что не так?


Torison
Можно, пожа
Дмитрий Таратута
Главарь фудшера
Морфий
напоминаю про https://t.me/dubki_food/23712 стоп флуд, ре...
Реально, завязывайте с флудом, чат читает 55%+ участников чата, а вы тут разводите свои беседы. Лучше бы еду отдавали
4 4 3 1 1
edited 19:51
Viktoria
Барахолка не чат, а канал. Пока туда запостят куча времени пройдёт
Reply
Александра!
Хватит флудить, пожалуйста, для этого есть отдельный чат
7 2 1
21:03

Актуальность проекта

Организация бронирования и шеринга сейчас – **хаос**

- Объявления теряются в потоке сообщений
- Неактуальные предложения остаются в чатах
- Бронирование - сплошная путаница
- Ручная модерация

Цель и задачи проекта

Цель:
Разработать Android-приложение для жителей общежития, объединяющее шеринг вещей и бронирование общих пространств в едином интерфейсе, чтобы упростить поиск доступных ресурсов, оформление заявок и организацию повседневных бытовых сценариев.

Задачи:

- Спроектировать основные пользовательские сценарии
- Обеспечить взаимодействие с API, локальный кэш и хранение сессии
- Заложить грамотную архитектуру для дальнейшей поддержки и реализовать Android-приложение на Kotlin

Сравнительный анализ

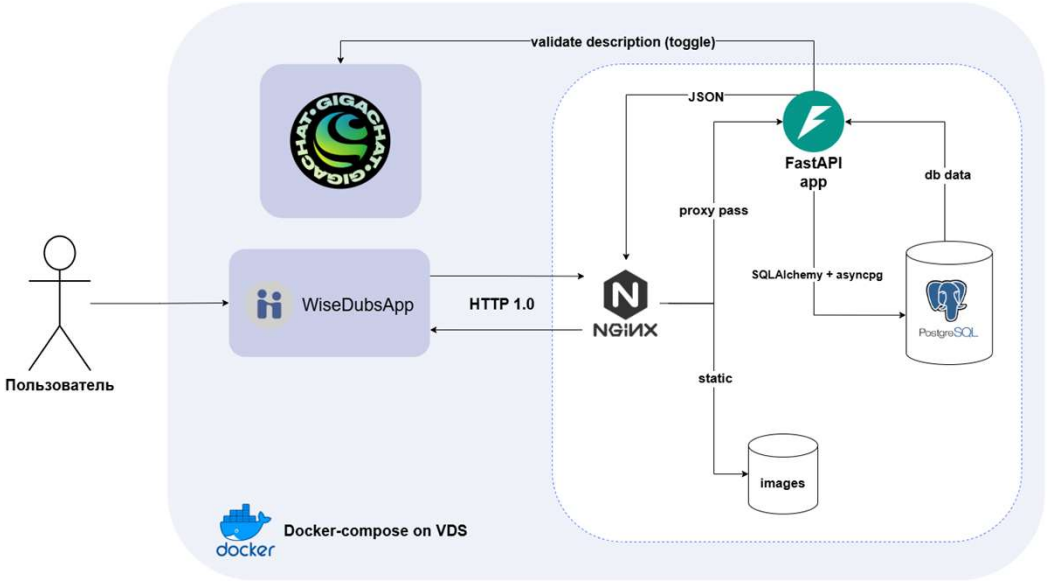
Критерий	Telegram	HeyHub	Dubki App	Дубовозки	**WiseDubsApp**
Быстрый доступ к информации	+	+	+	+	+
Пользовательский контент	+	+	-	-	+
Поиск и фильтрация	+/-	+	-	-	+/-
Обмен вещами	+/-	-	-	-	+
Бронирование общих пространств	-	+	+	-	+
ИТОГО	**4/5**	**4/5**	**2/5**	**1/5**	**5/5**

Основной функционал

1. Регистрация, вход и хранение пользовательской сессии.
2. Просмотр ленты объявлений и карточки объявления.
3. Создание, редактирование и удаление собственных объявлений.
4. Резервирование доступного объявления и отмена резерва.
5. Бронирование общих помещений с проверкой занятости интервала.
6. Просмотр личных постов, забронированных объявлений, броней и профиля.
7. Работа с изображениями объявлений и точками выдачи.

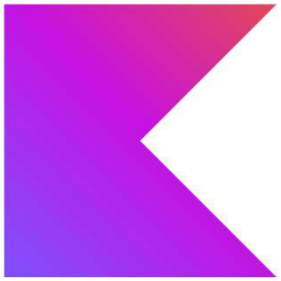
Серверная часть

- Авторизация: регистрация, вход, текущий пользователь, bearer-токен
- Шеринг: объявления, изображения, резервирование и отмена резерва
- Бронирование: комнаты, проверка занятости, создание и изменение брони
- Комнаты и локации пользователя: справочники комнат и сохраненные точки выдачи



Ссылка на OpenApi с описанием контракта

Стек технологий

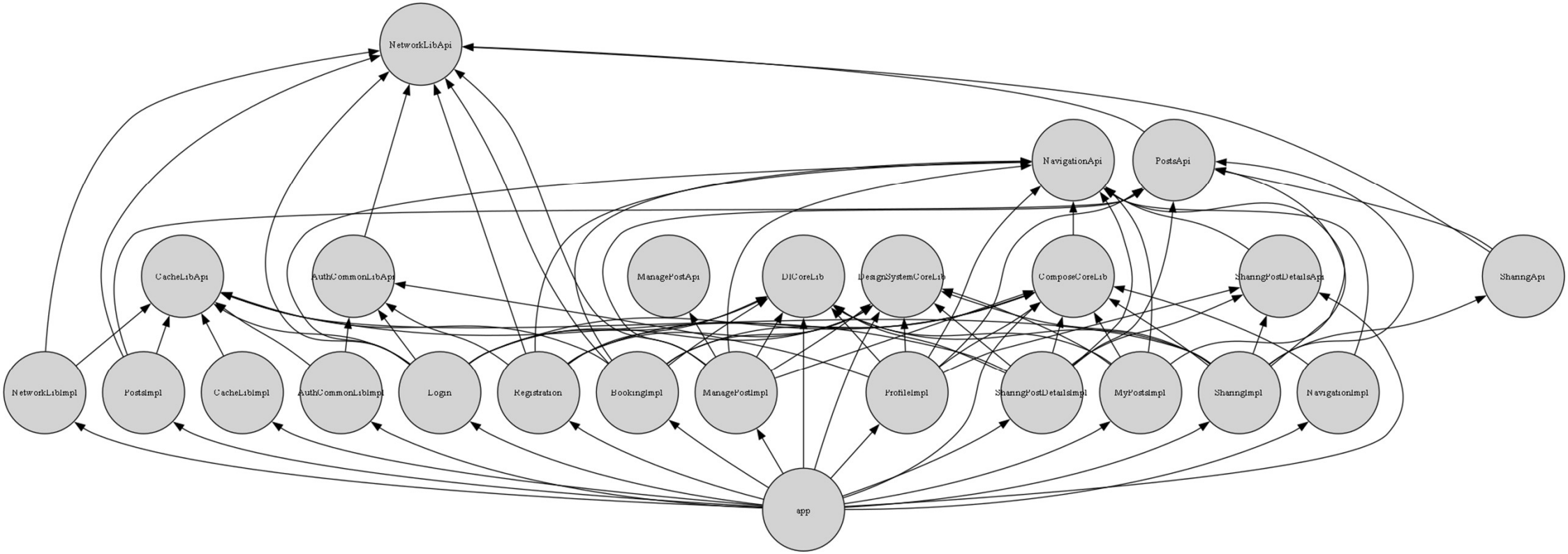


Kotlin

Jetpack Compose

Retrofit

Архитектура приложения (взгляд свысока)



Dependency Injection

CacheLibApi → CacheApi

CacheLibImpl → CacheLibComponent : CacheApi

Feature/App-модули → context.deps<CacheApi>()

Dagger остается в impl-слое, а внешние модули видят только интерфейс.

```
interface CacheApi {
    val dataStoreHelper: DataStoreHelper
    val postDataSource: PostDataSource
    val bookingDataSource: BookingDataSource
}

@Singleton
@Component(modules = [CacheLibModule::class])
interface CacheLibComponent : CacheApi
```

```
val component = rememberScopedComponent {
  context.run {
    DaggerMyPostsComponent.factory().create(deps<PostsApi>())
  }
}
```

Navigation

Навигация по экранам:

- Каждый provider реализует общий RouteEntryProvider
- Dagger собирает providers в Set<RouteEntryProvider> через @IntoSet
- Это попадает в NavigationImpl либу и там регистрируются руты для всех экранов

Глобальная навигация:

- Поделена на две части: PreLogin и PostLogin
- Это позволяет разделить некоторые зависимости для более строгой структуры

```
interface RouteEntryProvider {
    fun EntryProviderScope<NavKey>.provideRoute()
}
```

```
@Binds
@IntoSet
@PostLoginRoutes
fun bindBookingRouteProvider(
    provider: BookingScreenRouteProvider
): RouteEntryProvider
```

Демонстрация

Результаты и выводы

- Разработано Android-приложение WiseDubsApp, объединяющее ключевые сценарии студенческой жизни в общежитии: шеринг вещей и бронирование общих пространств
- Повторяющиеся бытовые задачи перенесены из неструктурированных чатов в единый мобильный интерфейс
- Пользователь получает более прозрачный способ узнать доступность вещей и общих помещений
- Многомодульная архитектура позволяет добавлять новые сценарии общежития без переработки всего приложения, также фичи могут разрабатываться независимо
- Цель работы достигнута

Дальнейшее развитие проекта

1. Связаться с инициативной группой IT-клуба общежития Дубки для проработки совместного решения
2. Расширение функциональности: добавление возможности совместных поездок на такси, ленты объявлений и тд.
3. Переезд на Kotlin Multiplatform для поддержки обеих платформ (IOS + Android)
4. Реализация ролей в приложении
5. Возможность масштабировать решение под разные общежития
6. Интеграция с Hse App X....

Список использованных источников

1. Android Developers. Документация по Android, Jetpack Compose, Navigation и Room [Электронный ресурс]. URL: https://developer.android.com/ (дата обращения: 20.05.2026).
2. Kotlin Documentation. Официальная документация языка Kotlin [Электронный ресурс]. URL: https://kotlinlang.org/docs/home.html (дата обращения: 20.05.2026).
3. Dagger. User Guide [Электронный ресурс]. URL: https://dagger.dev/dev-guide/ (дата обращения: 20.05.2026).
4. Square Open Source. Документация Retrofit и OkHttp [Электронный ресурс]. URL: https://square.github.io/ (дата обращения: 20.05.2026).
5. FastAPI. Официальная документация фреймворка [Электронный ресурс]. URL: https://fastapi.tiangolo.com/ (дата обращения: 20.05.2026).
6. Dubki App [Электронный ресурс]. URL: https://project1013191.tilda.ws/ (дата обращения: 20.05.2026).
7. HeyHub [Электронный ресурс]. URL: https://heyhub.com/ (дата обращения: 20.05.2026).
8. «Дубовозки» [Электронный ресурс]. URL: https://play.google.com/store/apps/details?id=com.alad1nks.dubovozki (дата обращения: 20.05.2026).


ВЫСШАЯ • ШКОЛА • ЭКОНОМИКИ •