



КЕЙС VK



Задание для хакатона: MVP Витрины приложений RuStore

Партнер: VK

Сложность: Начинаящий / Средний

Контекст и проблема

RuStore — это официальный российский магазин приложений для Android. Одна из его ключевых задач — дать российским разработчикам удобную и современную площадку для распространения своих продуктов.

Перед участниками стоит задача разработать MVP (Minimum Viable Product) витрины приложений. Основная цель — не просто отобразить список приложений, а создать увлекательный и современный пользовательский опыт, который будет мотивировать пользователей изучать и устанавливать новые приложения. Это может быть как Android приложение (в приоритете), так и web с адаптивной версткой, использовать SPA. Необходимо проанализировать существующие решение и улучшить его

Обязательные требования к приложению

*Примечание: для данных вы можете использовать описание настоящих приложений с RuStore или даже добавить «описание» для некоторых своих будущих приложений. Проанализируйте существующее решение и предложите свой подход, который изменит RuStore.

Ваше приложение должно содержать следующие экраны:

1. Онбординг

- Показывается только при первом запуске приложения (или авторизации).
- Содержит:
 - Логотип RuStore.
 - Приветственный текст (например, «Добро пожаловать в RuStore! Откройте для себя мир российских приложений»).
 - Кнопку «Начать», которая ведет на главную витрину.
 - Демонстрация по шагам работы витрины (подсветка по очереди основных элементов интерфейса, чтобы пользователь понял, что за что отвечает)

2. Главный экран (Витрина приложений)

- **Список приложений:** Каждая карточка в списке должна отображать:
 - Иконка приложения.
 - Название.
 - Оценка приложения
 - Категория (используйте: «Финансы», «Инструменты», «Игры», «Государственные», «Транспорт». Можете выделить свои категории).
- **Кнопка категорий:** В верхней части экрана должна быть кнопка или элемент навигации для перехода к экрану «Список категорий» (своеобразные фильтры).

3. Карточка приложения

- **Навигация:** При тапе на приложение открывается его карточка.
- Кнопка «Назад» (или использование системной кнопки назад).
- Детальная информация:
 - Иконка (большого размера).
 - Название.
 - Разработчик.
 - Категория.
 - Возрастной рейтинг (0+, 6+, 12+, 16+, 18+).
 - Полное, развернутое описание.
- **Горизонтальный список скриншотов:** При тапе на любой скриншот должно открываться полноэкранное галерея с возможностью листать все скриншоты этого приложения.

4. Экран категорий

- Отображает список всех доступных категорий.
- При выборе категории пользователь попадает на экран, аналогичный главной витрине, но с отфильтрованным по выбранной категории списком приложений.

Технические детали и данные

- **Источник данных:** Данные о приложениях (название, описание, иконки, скриншоты, категории) должны храниться **локально** в приложении (например, в виде JSON-файла в assets).
- **Опциональное задание (фича!):** Реализовать загрузку данных из простого Mock API. Вы можете использовать JSON-файл, размещенный на любом бесплатном хостинге для статики. Это позволит продемонстрировать работу с сетью. Или вы можете использовать загрузку через режим администратора, чтобы

показать возможность обновления данных, а также это может пригодиться для системы рекомендаций и «Выбор редакции».

Ключевые необходимые фишки и критерии оценки (то, что поможет сделать вашу работу выделяющейся)

1. Продуманный и современный UI/UX:

- Использование актуальных компонентов, например, Jetpack Compose (или аналоги), если делаете Android приложение
- Плавные анимации переходов между экранами, особенно при открытии/закрытии галереи скриншотов.
- Кастомные элементы, например, красивый "шейк" для загрузки контента (skeleton loading), а не стандартный ProgressBar.

2. Система рекомендаций:

- Реализовать простой алгоритм «Похожие приложения» на экране карточки. Например, на основе категории. Если пользователь чаще смотрит приложения категории «Игры», то в первую очередь предлагать именно эти приложения. Также по описанию карточек можно делать выводы и предлагать приложения со схожими тематиками.
- На главном экране реализовать доп.категоризатор:
 - «Популярные приложения» (те, которые чаще всего выбирают пользователи).
 - «Новинки» - если обновили БД (добавили новые приложения), то новые приложения помечаются как новые.
 - «Выбор редакции» - администратор назначает лучшие.

3. Поиск и фильтрация:

- Добавьте на главный экран строку поиска, которая позволяет искать приложения по названию.
- Реализуйте возможность сортировки (например, «По популярности», «По новизне»).

Рекомендации по реализации (Web-версия)

Если команда выбирает Web направление, приложение должно быть не статичной страницей, а полноценным SPA (Single Page Application).

- **Фреймворк:** Желательно Использование React, Vue.js или Angular.
- **Адаптивность:** Корректное отображение на разных разрешениях (популярные мобильные телефоны, планшеты).

- **Интерактивность:** Все те же требования к анимациям галереи, плавным переходам и современному UI.

Дополнительные фишки к заданиям, которые выделяют ваше решение на фоне других:

1. Интерактивная система рейтингов и отзывов

- **Голосование без регистрации:** Система лайков/дизлайков
- **Интеграция с VK:** Возможность оставить отзыв через аккаунт VK (используя VK ID)
- **Визуализация рейтинга:** Анимированные звезды, графики распределения оценок

2. Персонализация на клиенте

- **Локальная история просмотров:** "Вы недавно смотрели"
- **Умные рекомендации:** Простой алгоритм коллаборативной фильтрации

3. Игровые механики

- **Достижения за исследование:** Badge-система ("Исследователь финансов", "Гуру игр", "Посмотрел 10 приложений")
- **Прогресс-бар изучения:** Визуализация того, сколько % приложений в категории пользователь просмотрел
- **Ежедневная рекомендация:** Алгоритм, который каждый день показывает одно случайное приложение с интересным фактом

4. Социальные фишки (без бэкенда)

- **Генерация постов для VK:** Кнопка "Поделиться находкой", которая создает красивую картинку с описанием приложения для соцсетей
- **Локальные списки:** Пользователь может создать "Мой топ-5 приложений" и поделиться ссылкой
-

5. Аналитика и обратная связь

- **Heatmap кликов:** Анонимная визуализация популярных приложений
- **A/B тестирование интерфейсов:** Разные варианты отображения карточек
- **Сбор фидбека:** Встроенная форма "Что улучшить в каталоге?"

7. Реализовать установку приложений через PackageInstaller

- если хранить данные локально, арк должны храниться так же локально
- если вы реализовали бэкенд, арк должны храниться на бэкенде. Файл большой, рекомендуется поддержать скачку в сервисе
- при нажатии на кнопку Установить, арк должна установиться на устройство
- при установке, нужно предусмотреть, что у юзера нет разрешения на скачку и установку приложений из неизвестных источников. Их нужно запросить
- дополнительно, реализовать удаление приложения по кнопке "Удалить"

ЧЕК-ПОИНТ 15.11, начало в 18.00

Необходимо показать промежуточное решение, может даже не рабочее еще, но объяснить к чему стремитесь.

СТОП-КОД 16.11 в 10.00

Необходимо подготовить и прислать в личку @idmtest:

0. Обязательно написать название команды!
1. Скринкаст – 2-3 мин, видеозапись, где вы показываете суть вашего решения. Не нужно описывать проблематику, команду и прочее, это все на защите – только как работает ваше решение и что вы для этого использовали. Чем больше задач закрыли, чем больше фишек реализовали – тем лучше.
«Продайте» ваше решение за 3 мин. Можете залить в облако или на видеохостинг, или прислать файлом (если не несколько гигабайт)
2. Ссылка на репозиторий с кодом (github, не забудьте сделать репозиторий публичным). Если не умеете пользоваться github, то можно архивом. В Readme опишите, как это все работает.

3. Драфт презентации. На защите может быть измененная преза. В драфте постарайтесь показать архитектуру вашего проекта.

По регламенту на защиту допускается ТОП-10 команд по результатам ознакомления с проектами после Стоп-кода и чек-поинту. Но! Если будет больше классных решений, конечно, увеличим это число.

Защиты начинаются в 13.00 16.11. Формат защит уточняется, возможен гибрид, но большинство жюри будет на площадке, показать вживую всегда лучше 😊. На защиту 5 мин и 5 мин на вопросы.

Критерии оценки:

Релевантность поставленной задаче

Работоспособность прототипа в общем

Реализованные основные задачи

Реализованные дополнительные задачи

UI/UX решения

Описанная документация проекта: архитектура, как что работает, как что запустить

Выступление команды