

[Описание](#)

[Пользовательская аудитория](#)

[Фичи приложения](#)

[Мокапы экранов пользовательского интерфейса](#)

[Главный экран](#)

[Ключевые аспекты](#)

[Работа с данными](#)

[Крайние и конечные случаи UX](#)

[Используемые библиотеки](#)

[Внешние сервисы](#)

[Список задач:](#)

[Задача 1: Настройка проекта](#)

[Задача 2: Добавить UI для каждой Activity и Fragment](#)

[Задача 3: Обеспечить работу RecyclerView](#)

[Задача 4: Создать модель данных](#)

[Задача 5: Создать логику для работы с данными](#)

[Задача 6: Обеспечить возможность делиться шуткой](#)

GitHub Username:

- Daniily000 - Даниил Юрков
- Debosch - Влад Шильцин

ChuckApp

Описание

Читайте, делитесь и создавайте отборные шутки про настоящего героя голливудских боевиков - Чака Норриса.

Приложение предоставляет возможность читать, загружать, хранить, делиться шутками про Чака, а также создавать их, имея при этом возможность проверить, существует ли уже похожая шутка, и сохранить ее за авторством себя.

Никакой регистрации в приложении не требуется - скачай и пользуйся. Приложение распространяется на бесплатной основе и не содержит рекламы.

Пользовательская аудитория

Приложение рассчитано на широкую аудиторию, которая часто пользуется мобильным телефоном. Основной возрастной сегмент - 18-45 лет. Вполне пригодится тем, у кого в течение дня есть время, когда нечем заняться (поездки на автобусе, метро).

Фичи приложения

- Делиться шутками
- Искать шутки
- Создавать шутки
- Создавать категории для шуток
- Сохранять шутки в Избранное

Мокапы экранов пользовательского интерфейса

Эту часть работы можно сделать и от руки и просто сфотографировать и вставить в этот блок. Или же используйте программы: Google Drawings, www.ninjamock.com, Paper by 53, Photoshop или Balsamiq.

Главный экран



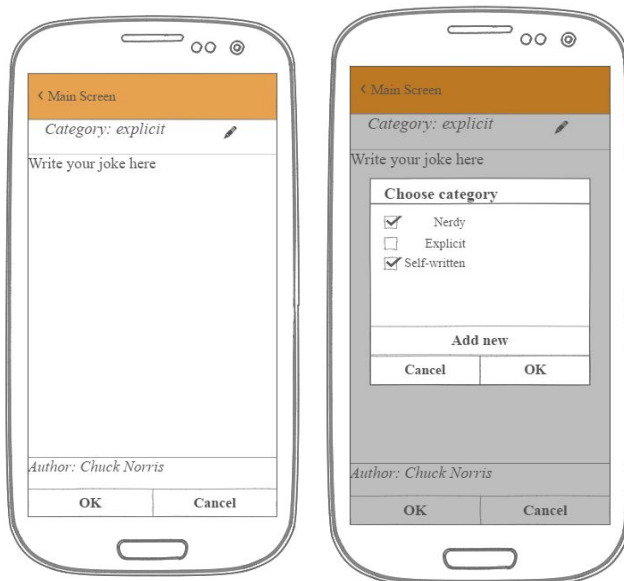


Главный экран отображает общий список из случайно выбранных шуток. Тулбар имеет две кнопки - для поиска и для дополнительных опций. Дополнительные опции содержат вкладку “категории”, по нажатию на которую открывается фрагмент с выбором категорий.



При невозможности загрузить шутки отображается следующее.

Экран новой шутки



Нажатие на кнопку “New joke” открывает экран с добавлением шутки, где выбираются подходящие под новую шутку категории, пишется сама шутка и указывается авторство. Выбор категорий точно такой же, как фрагмент с выбором категорий на прошлом экране.

Ключевые аспекты

Работа с данными

Приложение работает с API api.chucknorris.io. Далее по тексту базовым адресом будет считаться <https://api.chucknorris.io/>. API по запросу отдает шутки про Чака.

API умеет:

- Давать случайную шутку: `jokes/random`
- Отдать шутку(-и) из категории(-ий): `jokes/random?category={category}`
 - Получить категории: `jokes/categories`
 - Известных на данный момент категорий: 16
- Осуществлять поиск по базе: `jokes/search?query={query}`

Запросы осуществляются через Retrofit2, парсятся через Gson. Локально хранятся в SQLite. Процесс получения шутки:

1. Проверка существования шутки в локальном хранилище

2. Если шутки нет
 - а. Получить шутку от сервера
 - і. Если невозможно, выдать оповещение в UI и прекратить получение
 - б. Сохранить шутку в хранилище
3. Отдать шутку из хранилища

Далее шутка проходит через адаптер и отображается в своем View

Новая шутка записывается в отдельную таблицу в локальной базе.

Крайние и конечные случаи UX

Нажатие на лупу в тулбаре изменит вид тулбара - текст сменится на EditText с предложением ввести номер шутки для поиска. Нажатие на Enter запустит процесс получения и отображения шутки во фрагменте. При загрузке шутки предполагается отображение пустого контейнера, после подгрузки будет проигрываться анимация появления (alpha: 0 -> 1).

При нажатии на кнопку "поделиться" будет предложен выбор платформы, через которую пользователь захочет поделиться шуткой

Используемые библиотеки

Retrofit2 - для осуществления HTTP запросов и управления возвращаемыми результатами

Gson - для парсинга Json-файла.

Material design - визуальная составляющая приложения.

Внешние сервисы

Не планируются

Список задач:

Задача 1: Настройка проекта

Задачи по настройке проекта:

- ❖ Создать репозиторий
- ❖ Создать проект
- ❖ Сделать Initial commit
- ❖ Сделать Readme

Задача 2: Создать UI

- ❖ Создать layout для отображения основной активности с шутками
- ❖ Создать layout для Itema шутки (предусмотреть возможность переиспользования для фрагмента с одной шуткой)

Задача 3: Обеспечить работу RecyclerView

- ❖ Создать адаптер
- ❖ Реализовать сортировку
- ❖ Реализовать обновление данных

Задача 4: Создать модель данных

- ❖ Создать дата классы для хранения шуток

Задача 5: Создать логику для работы с данными

- ❖ Создать интерфейс Service
- ❖ Создать интерфейс Data Access Object
- ❖ Создать SQL базу данных для кэширования
- ❖ Осуществить возможность получения данных через Retrofit и их кэширования
- ❖ Осуществить возможность получения объекта из кэша

Задача 6: Обеспечить возможность делиться шуткой

- ❖ Реализовать возможность распространения шутки по социальным сетям

Дальнейшие действия

- После завершения заполнения всех блоков, поделись по ссылке документом с Машей. Убедись, что указал права на редактирование
- Жди комментариев
- После подтверждения, что можно приступить к разработке, скачайте этот документ как PDF [File → Download as PDF]
 - Проверьте, что PDF сохранился с именем
“KODE_Итоговый_проект_Команда_”

- Создайте новый GitHub-репозиторий для проекта. Назовите его **“Final Project”**
- Добавьте PDF в GitHub вашего проекта