**1. Краткое описание проекта (суть)**

Название мобильного приложения: FiFi Sql.

Описание: Мобильное приложение содержит в себе обсуждение пользователей, теорию и практику, чтобы закреплять материал. Так же будет бот для помощи по основным вопросам пользователей. Система мотивации будет поощрять пользователей (уровни, совместные задания с другими пользователями, рейтинг проектов) изучать и работать вместе.

Цель: Обучение пользователей языку программирования sql.

Аналоги: Learn SQL, Mimo, SQL Practice, SQL Play, SQLPhone.

Дата сдачи — 17.11.2025.

**2. Основные требования (Функциональные)**

**2.1. Пользовательские роли**

* Пользователь — регистрирование/вход, прохождение курсов, практики, совместных заданий, улучшение уровней, участие в обсуждении, получение ответов на основные вопросы в боте.
* Администратор (школа/компания) — возможность управлять бд, просматривать данные (как в курсаче).

**2.2. Базовый функционал (MVP)**

1. Аутентификация: логин/пароль.
2. Профиль пользователя: имя, уровень.
3. Курс с заданиями: категории курса (практический, теория и т.д.).
4. Анализ прогресса: узнать сильные, слабые стороны и отчетность у бота.
5. Обсуждение по курсу: написать вопрос, прокомментировать, отправить файл.
6. Оценка: возможность оценить полезность задания.
7. Уведомления push о занятиях.
8. Проект: создание в команде и голосование за лучшего.
9. Панель администратора (раздел в приложении) —изменить данные, просмотреть данные, сортировка данных, .?
10. Бот: ответ на основные вопросы, предоставление дополнительного материала при необходимости.

**2.3. Нефункциональные требования**

* Кроссплатформенность: iOS 13+/Android 8+.
* Время отклика UI: <200 ms на основные экраны (при наличии сети).
* Безопасность: HTTPS, безопасное хранение токенов, ограничение доступа к файлам.
* Конфиденциальность: соответствие общим требованиям (анонизированные логи), возможность удаления аккаунта.
* Локализация: RU.
* Доступность: контраст, масштабирование шрифта.

**3. Предложенный стек технологий (рекомендация для сдачи в сроки)**

* Frontend (мобильное): **Back4App Parse Backend**— платформа с низким кодом, которая позволяет разработчикам создавать высокомасштабируемые и продвинутые приложения без управления операциями на стороне сервера.
* Базы данных: Railway (облачная MySQL)
* CI/CD: GitHub Actions → автоматические сборки / тесты.
* JS (Cloud) +MySQL + Railway (облачная MySQL)

Обоснование выбора **Back4App Parse Backend** в двух-трех строчках: работает с нужными мне инструментами (MySQL) и

**4. Рекомендуемые инструменты разработки**

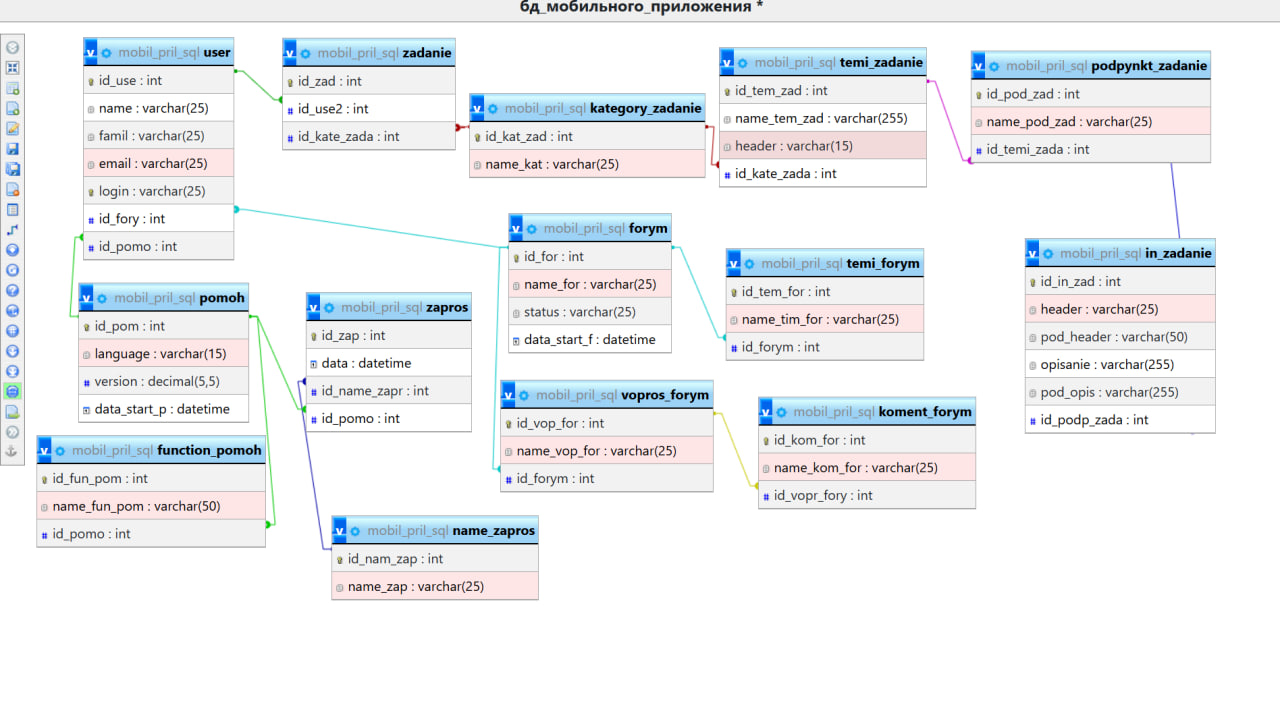
* IDE: Android Studio.
* Figma — прототипы / дизайн.
* GitHub — исходники + Actions.
* Debugger/ Logcat — отладка.
* Postman — если есть API.

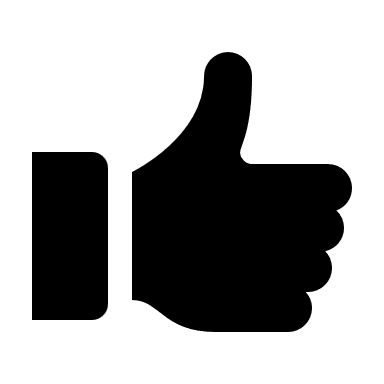
**5. Архитектура и данные**

**5.1. Архитектурная схема (вкратце)**

Мобильное приложение (Parse Server) ↔ Back4App и Storage; Автоматическое API (REST и GraphQL), Аутентификация (email, соцсети, Google, Facebook), Push-уведомления, админка (Parse Dashboard), Хостинг, для бизнес-логики JS или MySQL с Railway; .

**5.2. Структура БД**



**6. API / Endpoints (пример для собственного backend)**

Если вы решите делать собственный backend (Node/Express), пример endpoints:

* POST /auth/register
* POST /auth/login
* GET /users/:id
* GET /courses
* POST /courses
* GET /courses/:id/assignments
* POST /courses/:id/assignments
* POST /assignments/:id/submissions
* GET /users/:id/notifications
* POST /courses/:id/messages  
  …и т.д. (авторизация — JWT/Firebase tokens).

**7. UI/UX: основные экраны + требования**

1. Экран входа / регистрации — простая, крупные поля, Google sign-in.
2. Главная / задания — нахождения перечня заданий обучения (категорий).
3. Окна тем заданий – описание темы и обучение (теоретической, практической, проект).
4. Обсуждение— список сообщений, ввод, вложения, комментарии, файлы.
5. Профиль — данные, настройки уведомлений, выход.
6. Админ/преподаватель — возможность управлять бд, просматривать данные.
7. Бот – ответ на основные вопросы, предоставление доп. информации при необходимости.

Wireframe: минималистичный, карточный интерфейс; крупные кнопки, читаемые списки.

**8. Тестирование**

* Unit tests.
* Integration: минимальный сценарий «войти → открыть курс → сдать задание».
* Регрессионное тестирование перед сдачей.
* Тестирование на реальных устройствах (Android & iOS).
* PyCharm Debugger – бот.
* MySQL Workbench/phpMyAdmin – база данных.

**9. Релиз**

* Репозиторий: Git (feature-branch workflow).
* Pull Request template + code review.
* GitHub Actions: запускает lint, unit tests, билд apk.
* Автоматизированные сборки для тестовых релизов (Internal testing).
* Release: RuStore.

**10. Безопасность и приватность**

* Все запросы по HTTPS.
* Токены в безопасном хранилище (EncryptedSharedPreferences / Android Keystore).
* Контроль доступа на уровне БД (Parse Server даёт CLP (Class-Level Permissions) и ACL (Access Control Lists)).
* Ограничение загрузки файлов (В Parse есть File Adapter**,** size/type), сканирование расширений.
* Возможность удаления аккаунта / данных.

**11. Роадмап / план по дням с конкретными датами**

**Sprint 0 — Подготовка / Discovery (16 — 21 сентября 2025) — 6 дней**

**Цели:** финальные требования, приоритеты, дизайн-гайд, инфраструктура.

* 16–17.09: Сбор требований, определение MVP, Stretch (push-уведомления), согласование форматов файлов, определить формат хранения в MySQL.
* 18.09: Составление ТЗ (этот документ), определение требования для сервера, бота, приложения.
* 19–20.09: Быстрые наброски экранов + прототип Figma.
* 21.09: Настройка репозитория, CI (начало), базовая структура проекта.

**Доставка:** финализированные требования, Figma прототипы главных экранов, рабочий репозиторий.

**Sprint 1 — Core: аутентификация, профиль, список курсов, расписание (22 сентября — 5 октября 2025) — 2 недели**

**Задачи:**

* Реализовать аутентификацию (email/password).
* Экран профиля.
* Модель данных и интеграция Railway.
* Экран обсуждения (просмотр тем и вопросов).
* Базовые задания (просмотр теории / практики / проекта).
* Unit tests для Auth и моделей.

**Доставка:** рабочее приложение с логином, просмотром профиля, теории / практики / проекта и обсуждением; backend (Back4App) со структурой данных.

**Sprint 2 — Assignments & Submissions, Materials, Notifications (6 — 19 октября 2025) — 2 недели**

**Задачи:**

* CRUD для админов.
* Функция загрузки файлов в задания и в материалы (EncryptedSharedPreferences / Android Keystore).
* Реализация сдачи задания (практическая часть).
* Локальные и push-уведомления о занятиях.
* Фишки, оценки заданий.
* Кеширование данных для оффлайн (Room).
* Тесты: интеграционные сценарии «задание студента → проверка системы».

**Доставка:** функциональные задания + сдачи, проверка/просмотр материалов, уведомления.

**Sprint 3 — Чат/Форум, оценки, UI polish (20 октября — 2 ноября 2025) — 2 недели**

**Задачи:**

* Railway (бд);
* Окошки;
* Профиль;
* Доработка регистрация/авторизация.

**Доставка:** Railway, Окошки, Профиль, Доработка.

**Sprint 4 — Интеграция, тесты, админ-панель, подготовка к UAT (3 — 16 ноября 2025) — 2 недели**

**Задачи:**

* Сервер;
* Курс с заданиями;
* Рейтинг.

**Доставка:** Сервер, курсы.

**Stabilization & UAT fixes (17 — 25 ноября 2025) — 9 дней**

**Задачи:**

* Админы;
* Обсуждение.

**Доставка:** Админы, обсуждение.

**Final polishing & submission (26 — 28 ноября 2025) — 3 дня (включая саму сдачу)**

**Задачи:**

* Финальные баг-фиксы;
* Бот.

**Финиш:** сдача до **28.11.2025** (включительно).

**12. Уникальные функции проекта (если останется время)**

* Больше возможностей, фишек профиля.
* Отправка уведомлений с настройкой, сколько учиться будешь и т.д.
* Темная тема, настраиваемая, другими словами - кастомизация.
* Добавление доп. заданий и фишек к ним.
* Мини-игрулька.