

«Национальный исследовательский университет
«Высшая школа экономики»
Лицей

Индивидуальная выпускная работа
ИТ-ПРОЕКТ: Qwiz-Backend-Android
<https://github.com/LeSaR-school/Qwiz>

Выполнил: Ковалев Илья Михайлович
Группа: 11И2

Москва – 2023

Qwiz – приложение для android и iOS, и вебсайт — был задуман как замена популярной платформе для прохождения викторин Kahoot, переставшей работать в России. Несмотря на существование альтернатив, не одна из них не имеет весь функционал, который предоставлял Kahoot. Так, например, Qwizizz – одна из наиболее популярных альтернатив — требует оплату, которую невозможно провести в России. Другой пример — Quizlet – иногда просто перестает принимать ответы посреди проведения викторины. Так как я сам пытался пользоваться этими платформами, и остался неудовлетворен их работой, я решил разработать это приложение.

Изначальная задумка для проекта была таковой: викторины, создаваемые пользователями, можно бы было проходить как самостоятельно, так и в группе. Во втором случае хост викторины выводил бы сайт на большой экран (например с помощью проектора в классе лица), где были бы видны вопросы. На экранах участников, в это время, показывались бы варианты ответов. Также, специально для учителей, есть классы, в которые можно добавлять учеников, и задавать им викторины для самостоятельного прохождения с дедлайном.

В итоговом продукте не удалось реализовать групповое прохождение викторин по следующей причине. В начале планирования группового проекта я взял на себя разработку бэкэнда, андроид приложения и вебсайта. Но из-за ограничений по времени, разработка сайта была передана моему партнеру. Это не создало бы неполадок, если бы не тот факт, что в групповом проекте каждому участнику необходимо написать свой бэкэнд. Так как возможность хостинга квизов предполагалась только для веб версии приложения, которую я передал партнеру, прохождение викторин в группе пришлось отменить (то есть не был сделан 8 сценарий).

Сценарий 1 — Логин / Регистрация / Изменение аккаунта. В связи с тем, что проверка и отправка писем на электронную почту была очень трудной в

реализации, я перешел на систему ников — каждый пользователь может придумать уникальный никнейм. В связи с отсутствием сервиса электронной почты, смена пароля может осуществляться только при наличии данного пароля. Из сценария логина было убрано сообщение о смене пароля, так как если пользователь забыл свой данный пароль, его невозможно сменить. Остальные подсценарии были сделаны успешно (с изменением почты на никнейм).

Сценарий 2 — Создание / Присоединение к классу — работает как и предполагалось. Учитель может создавать классы, добавлять учеников и задавать им викторины (с дедлайнами по желанию)

Сценарий 3 — Создание / Копия / Редактирование квиза. Все пункты, кроме последнего, были реализованы успешно. Пункт 11 — отправить .csv на почту — не был сделан по вышеупомянутой причине.

Сценарий 4 — Публикация квиза — сделан как описано в заявке. Единственный нереализованный аспект — отсутствует ссылка на приватный квиз, так как приложение не поддерживает внешние ссылки.

Сценарий 5 — Прохождение викторины — как сказано выше, сделан только для самостоятельного прохождения. В остальном сделано полностью.

Сценарий 6 — Смена вида приложения — с помощью Android Studio было очень просто сделать смену языка / темы. Смена вида сайта, очевидно, отсутствует.

Сценарий 7 — Просмотр популярных викторин / Поиск по названию. Сделан список популярных/новых викторин и строка поиска на главной странице. Ими можно пользоваться без наличия аккаунта.

Средства разработки

Для разработки backend я выбрал Rust и PostgreSQL, так как они были первыми в соответствующих категориях в [опросе StackOverflow об удовлетворенности разработчиков в 2022 году](#). Ни с одним из этих языков я не

работал до этого, но я смог изучить их за данный период времени, и мне очень понравилось с ними работать.

База данных содержит в себе 9 таблиц:

(pkey – колонка входит в первичный ключ таблицы, fkey – колонка является вторичным ключом)

- media – медиа (картинки, видео, аудио пользователей):
 - uuid (pkey) – случайный UUID4 для ссылок из других таблиц
 - uri – ссылка (локальная на файл или http для youtube видео) на ресурс
 - media_type – вид ресурса - image
- account – аккаунт пользователя:
 - id (pkey) – id пользователя для ссылок из других таблиц
 - username – уникальное имя пользователя (макс. 12 символов)
 - password_hash – hexdump хэша пароля пользователя (128 символов — 64 байта / 512 бит)
 - profile_picture_uuid (fkey1) – ссылка на таблицу media, содержащую аватарку пользователя
 - account_type – тип аккаунта — student / teacher / parent
- class – класс:
 - id (pkey) – id класса для ссылок из таблицы students
 - teacher_id (fkey1) – id аккаунта учителя
 - name – название класса (макс. 100 символов)

Также на таблице сделаны 2 триггера: первый проверяет тип аккаунта учителя, чтобы нельзя было создавать другим пользователям классы, а второй — удаляет все классы учителя при смене типа аккаунта.

- student – ученик в классе:
 - student_id (fkey1) – id ученика

- class_id (*pkey*) – id класса

На этой таблице так же есть 2 триггера, проверяющих тип аккаунта.

- assignment – задание квиза от учителя классу:
 - qwiz_id (*fkey1*) – id квиза, который задали классу
 - class_id (*fkey2*) – id класса, которому задали квиз
 - open_time – время открытия задания
 - close_time – дедлайн прохождения задания
 - id – id задания для ссылки из таблицы completed_assignment

- completed_assignment – сделанное учеником задание:

- assignment_id (*fkey1*) – id задания
- student_id (*fkey2*) – id аккаунта студента

На данной таблице есть триггер, проверяющий, есть ли студент в классе, которому назначено задание

- qwiz – квиз:

- id (*fkey1*) – id квиза для ссылок из таблицы question
- name – название квиза (макс. 100 символов)
- creator_id (*fkey2*) – id аккаунта создателя квиза
- thumbnail_uuid (*fkey3*) - ссылка на таблицу media, содержащую иконку квиза
- public – является ли квиз публичным (boolean)

- question – вопрос в квизе:

- qwiz_id (*fkey1*) (*pkey*) – id квиза, которому принадлежит вопрос
- index (*pkey*) – порядковый номер вопроса в квизе
- body – текст вопроса (макс. 500 символов)
- answer1 - answer4 – четыре ответа на вопрос (первые 2 обязательные, 3 и 4 необязательные) (макс. 200 символов)

- correct – номер правильного ответа (1-4)
- embed_uuid – ссылка на таблицу media, содержащую встроенную в вопрос картинку, видео или аудио
- vote – голос за квиз:
 - voter_id (fkey1) (pkey) – id проголосовавшего за квиз аккаунта
 - qwiz_id (fkey2) (pkey) – id квиза, за который проголосовали

Для написания android-приложения я использовал Android Studio и язык Kotlin. Android Studio был выбран так как это единственная программа для разработки нативных приложений android, а Kotlin – так как это улучшенная версия Java.

Этапы работы

Backend мной разрабатывался с августа по октябрь. До начала разработки я сменил язык с Go на Rust, так как не было времени его выучить. Это был один из рисков в начале работы над проектом, и мне удалось его оперативно разрешить.

Android-приложение я разрабатывал с октября по ноябрь. Не получилось уложиться в дедлайн так как backend потребовал больше времени чем я ожидал.

Вебсайт я отдал на разработку партнеру в конце августа, так как именно тогда осознал что мне не хватит времени на все 3 части проекта.

Рефлексия

В процессе работы возникло две проблемы: нехватка времени на изучение языка Go, которую я предсказал в начале работы над проектом, и успешно устранил за счет перехода на Rust, а также нехватка опыта написания бэкэнда, которая привела к смещению дедлайна для разработки android-приложения.

Продукт в будущем можно будет развить множеством способов, например:

- Добавить работу с электронной почтой
- Связать бэкэнды web-, iOS- и android-версий

- За счет связи бэкэндов добавить групповое прохождение викторин
- Добавить уведомления, когда ученику задан квиз в классе

За время работы над этим проектом я научился:

- Писать backend код на языке Rust
- Писать базы данных и сложные запросы на языке PostgreSQL
- Делать запросы к базе данных на SQL через Rust
- Создавать нативные андроид-приложения на Android Studio с Kotlin

В заключение доклада, я могу сказать, что мне понравилось работать над этим проектом, а именно изучать новые языки программирования (Rust, PostgreSQL, Kotlin), а так же планировать инфраструктуру приложения (т. е. как разные части приложения связаны и общаются друг с другом, как хранится информация в базе данных, и.т.п). Единственный большой минус — это написание различных бэкэндов мной и моим партнером, что привело к удалению некоторого функционала приложения.