# **Лабораторная работа 11. Разработка мобильного приложения**

# Задачи лабораторной работы:

• Разработать собственное приложение с применением функционала, изученного в рамках курсах ПВМС.

• распределить работы между участниками группы и управлять проектом разработки, используя Github Projects.

• для контроля версий использовать git , разместив проект на github в репозитории для команды;

• документировать проект в Readme и wiki репозитория согласно требованиям, представленным ниже;

• непрерывная сборка проекта в Travis CI и автоматические тесты.

## Описание проекта.

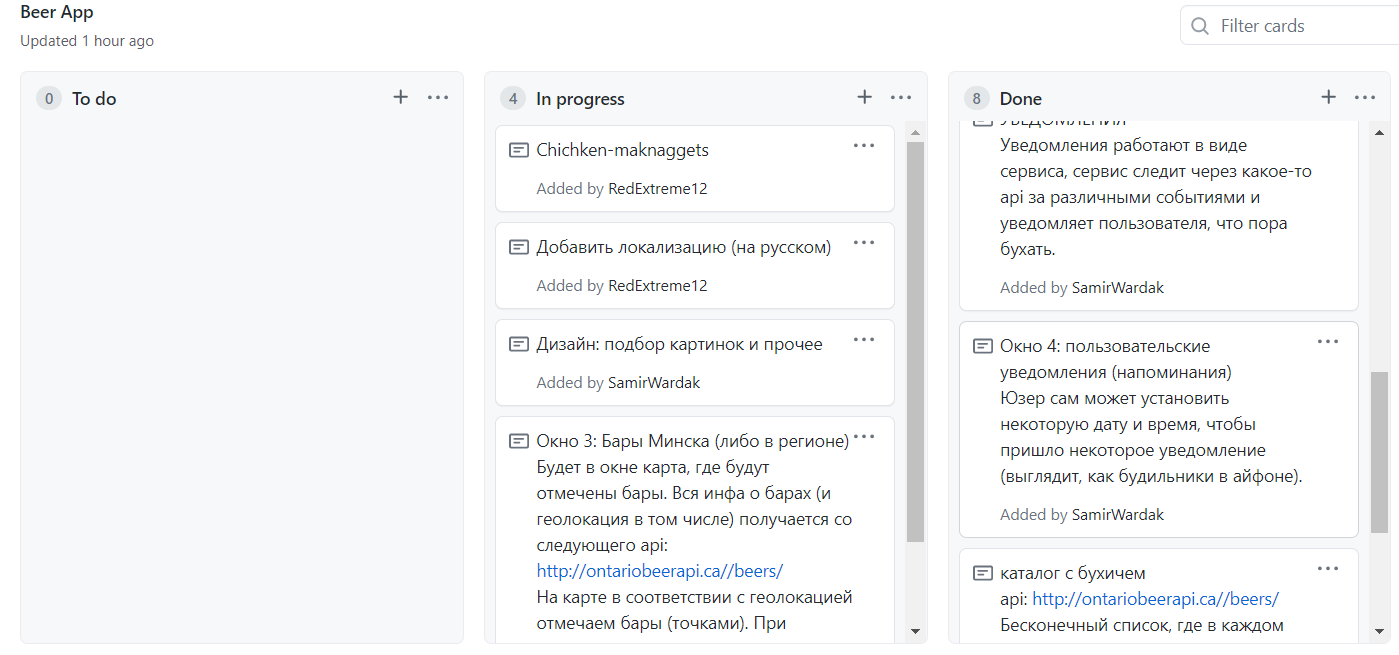
Это приложение подойдет людям, часто употребляющих пиво. Имеетя возможность просматривать различные типы алкоголя, загружаемые в приложение. Также имеется возможность добавить выбранный алкоголь в избранное.

Имеется возможность просмотреть события, чтобы отпраздновать их с пивом. . Также имеется возможность добавлять и оценивать свои собственные события. Вы можете установить это событие в дальнейшем уведомлении.

Имеется карта для указания местных локальных заведений для употребления пива.

## Распределение задач.

Для распределнеия задач использовали github projects.



Пример использования github projects.

Роли проекта:

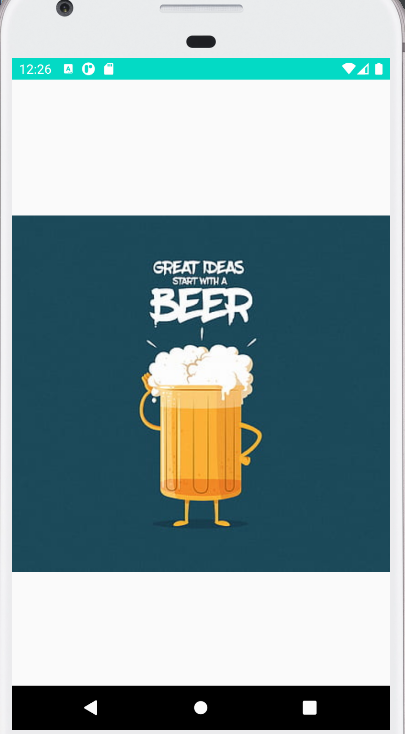
Иван Шишлянников — project manager, backend/frontend, travis ci.

Дмитрий Рогозенко — backend/frontend , database, tests.

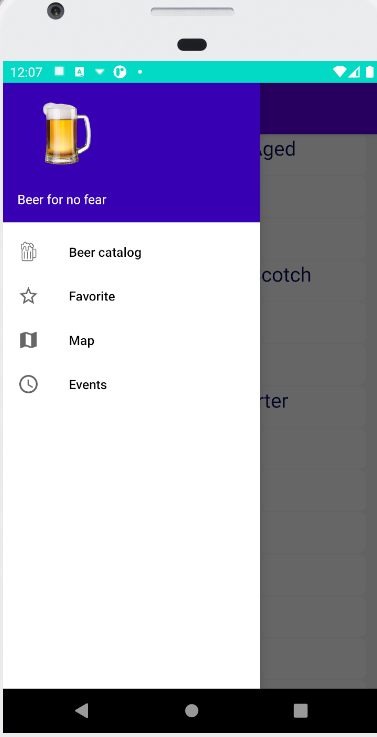
Самир Вардак — — backend/frontend, work with API.

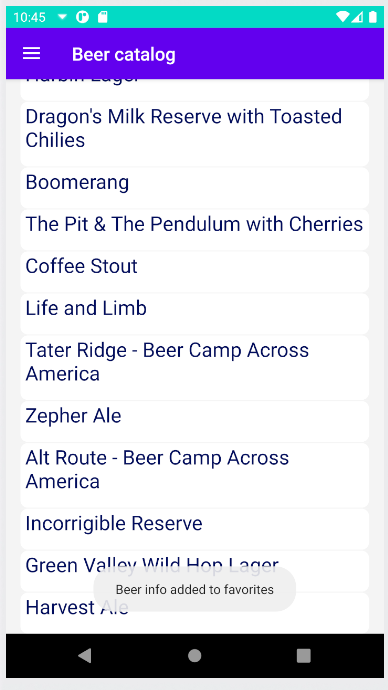
## Демонстрация работы

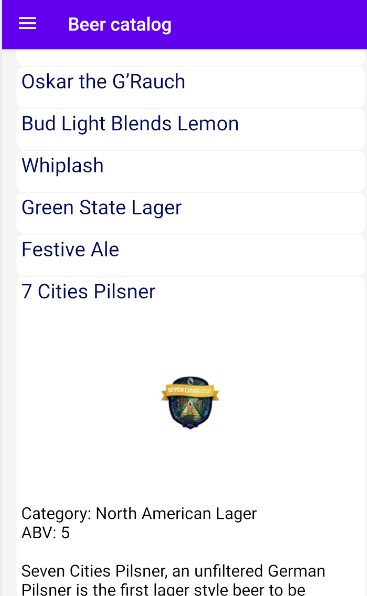
Загрузочный экран:



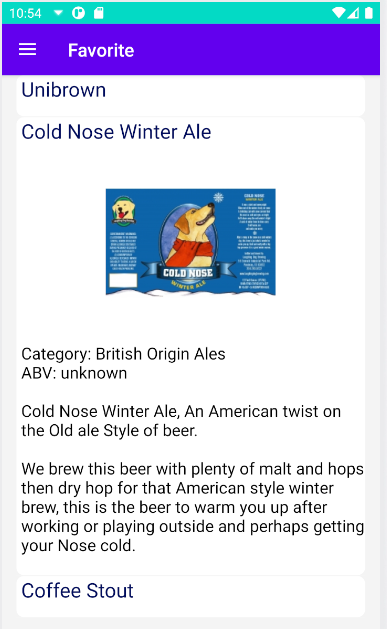
Главное выпадающее меню:



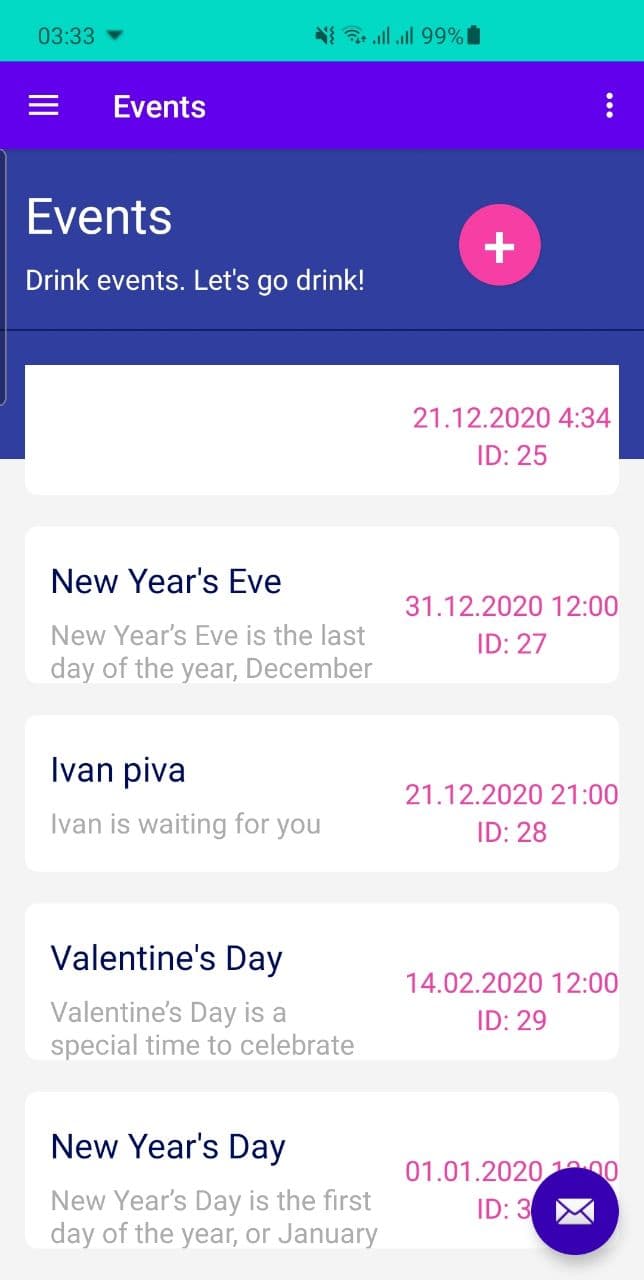


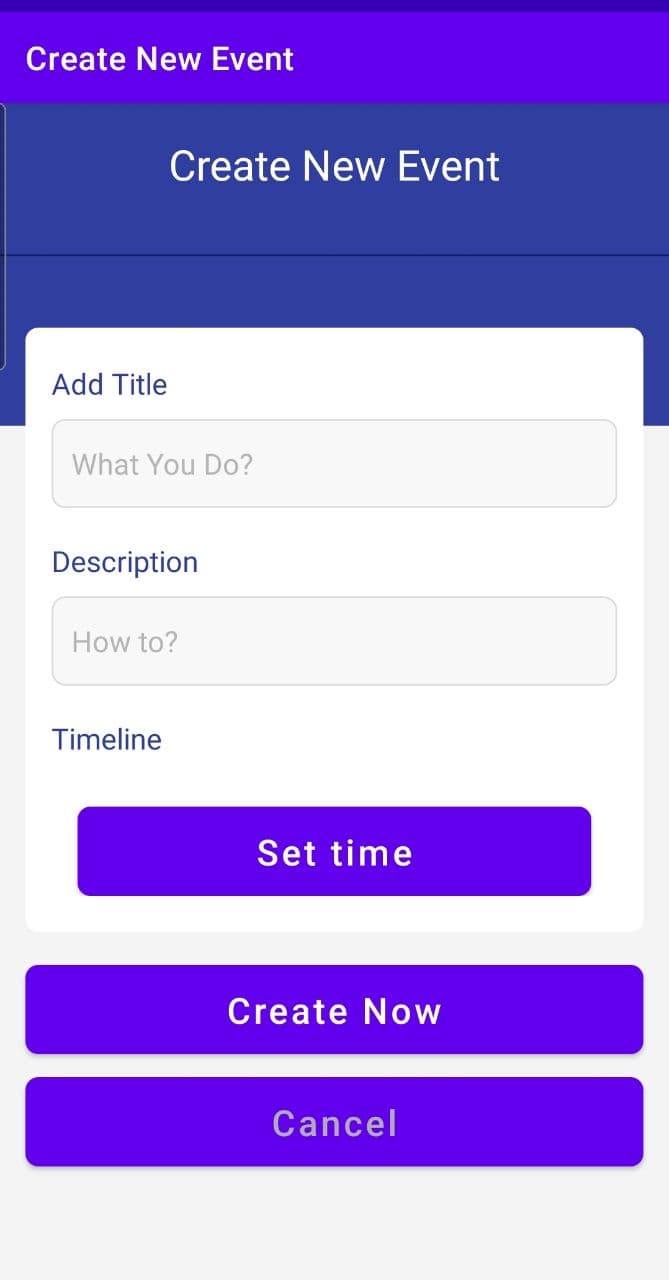


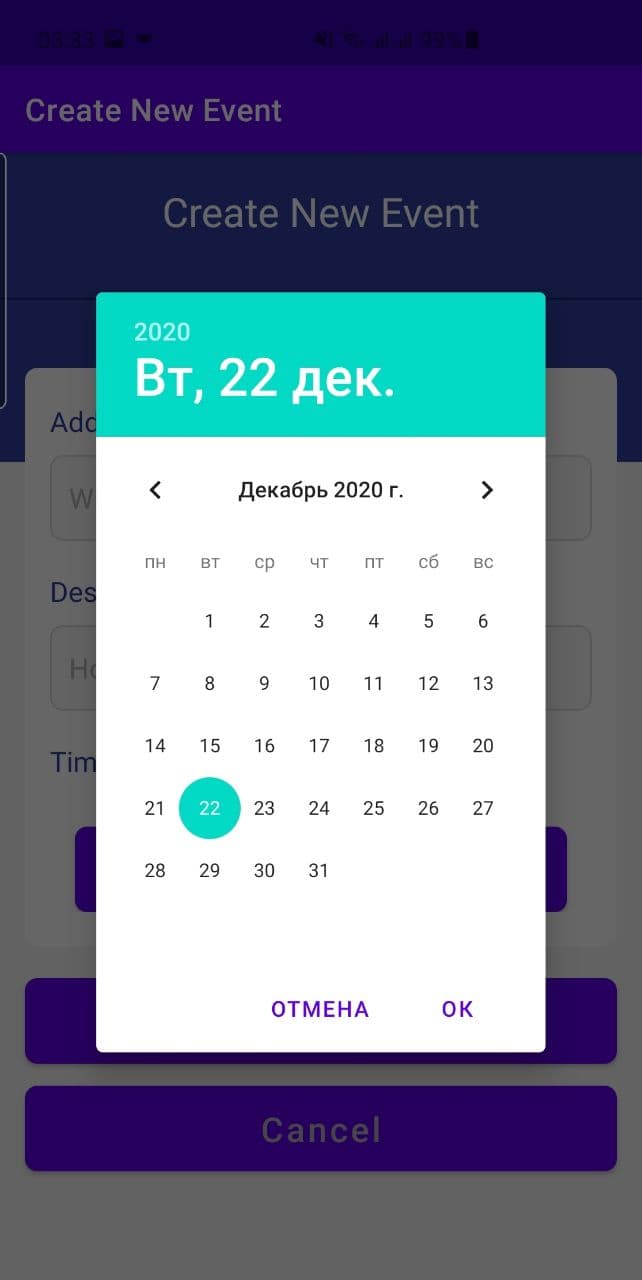
Каталог пива. Информация получается с помощью сервиса brewerydb.com.



Каталог сохраненного пива. Данные хранятся в db sqlite.





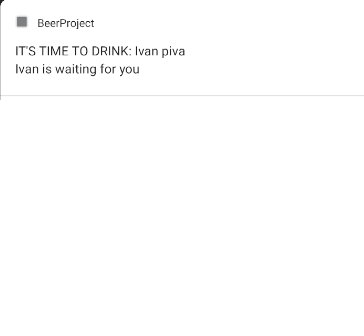


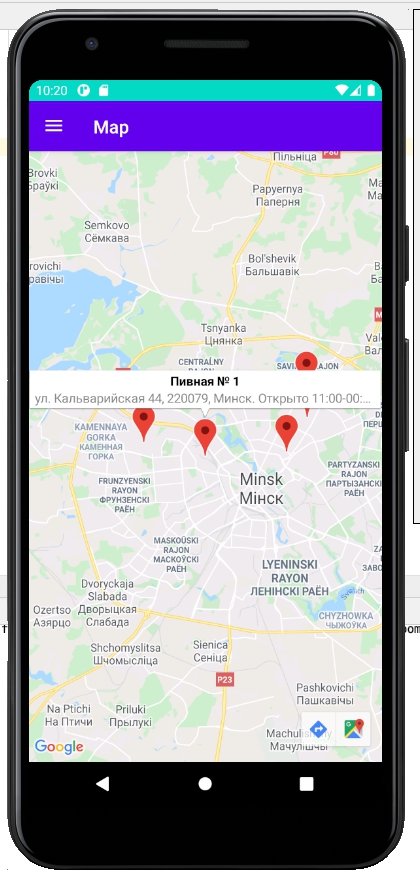
Редактор событый. События получаются с помощью сервиса calendarific.com.

Так же можно добавлять свои собственные события.

Пользователь получает уведомления о ближайших событиях. Уведомления присылаются сервисом работающем в foreground.

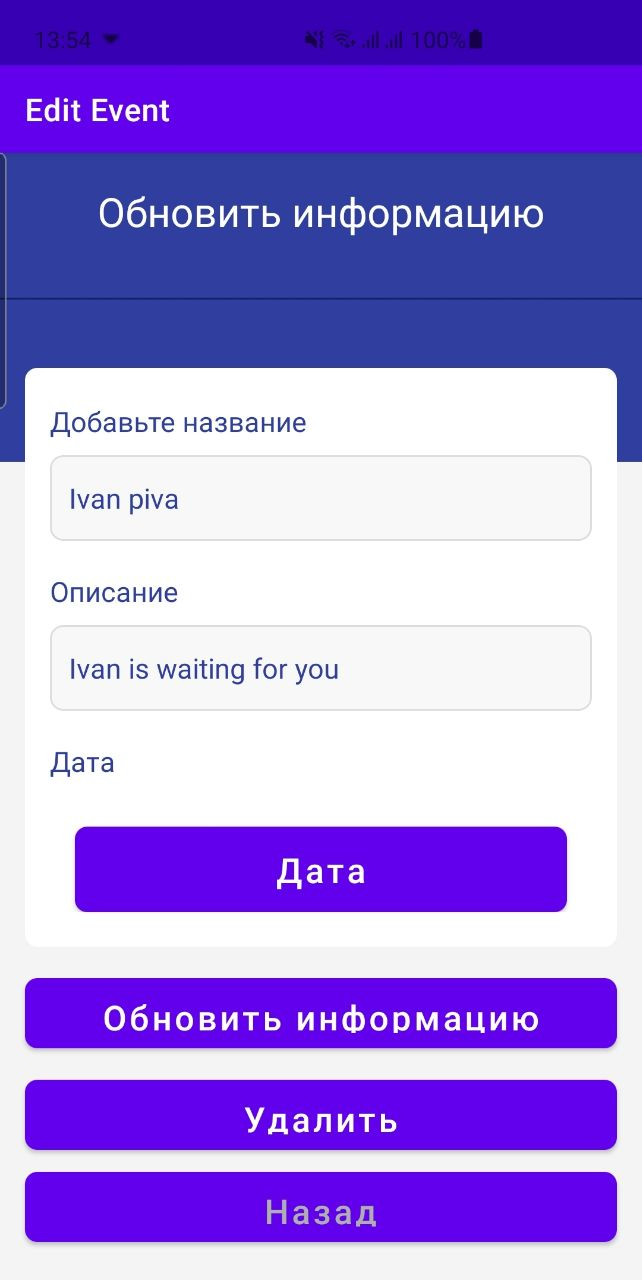
Пример уведомлений:

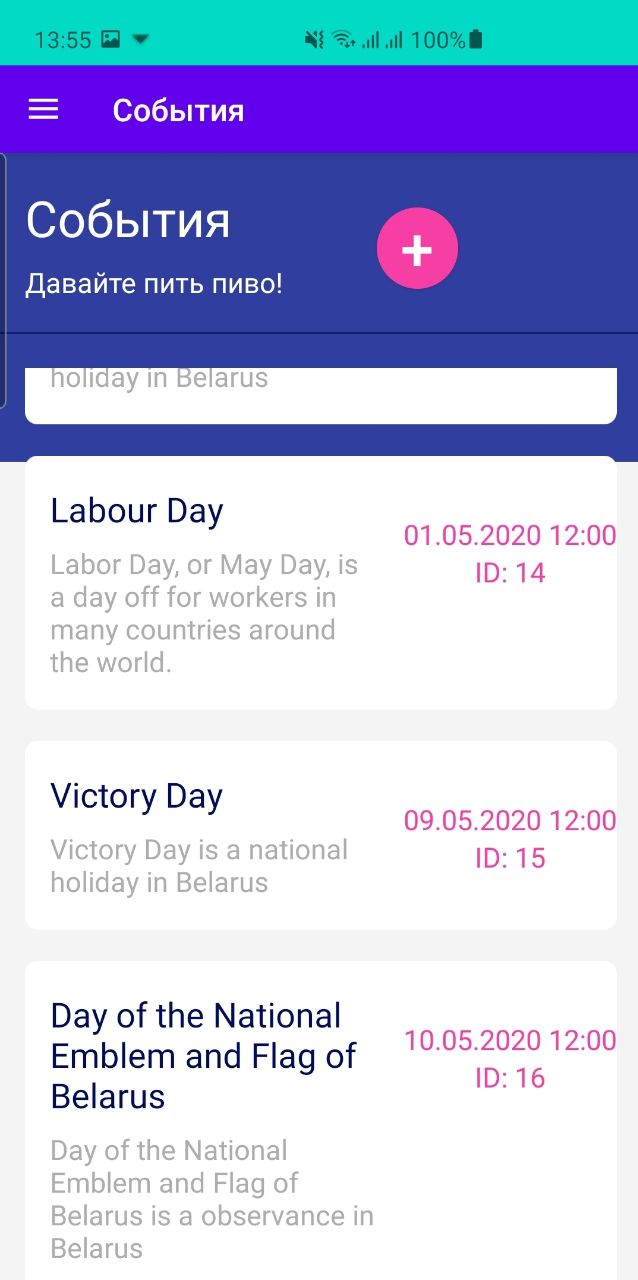




Работы карты. Пользователь имеет возможность получить дпоплнителную информацию в каком-либо заведении.

В приложении поддерживается локализация на русском и английском языках.





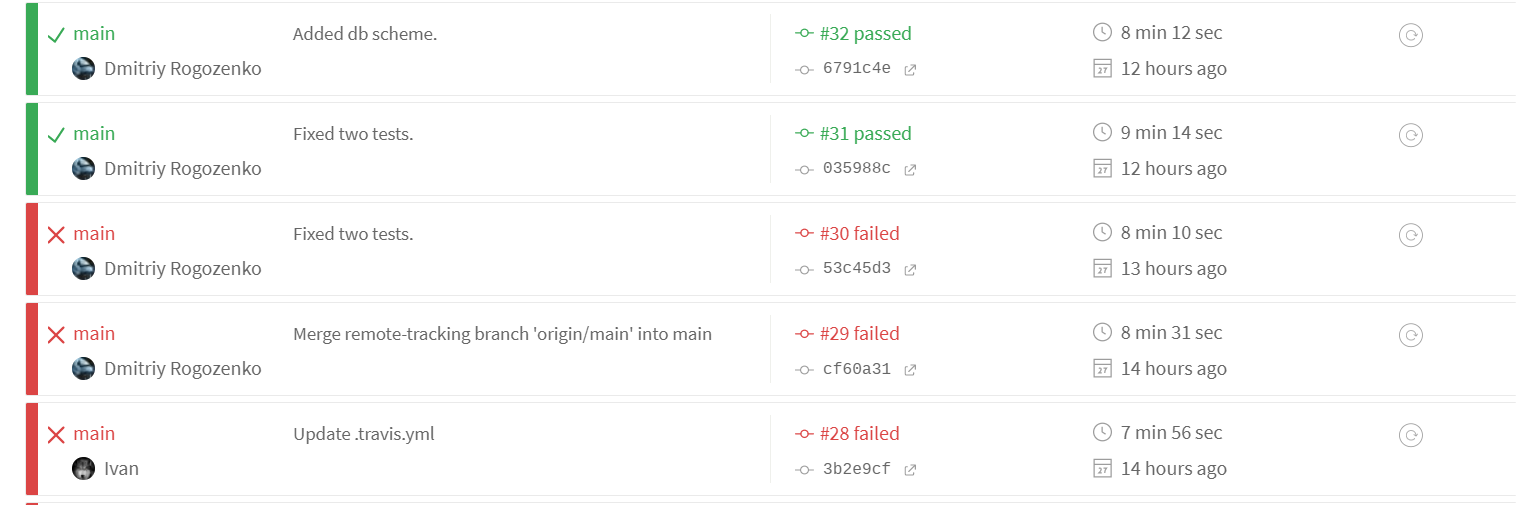
## Непрерывная сборка и тестирование.

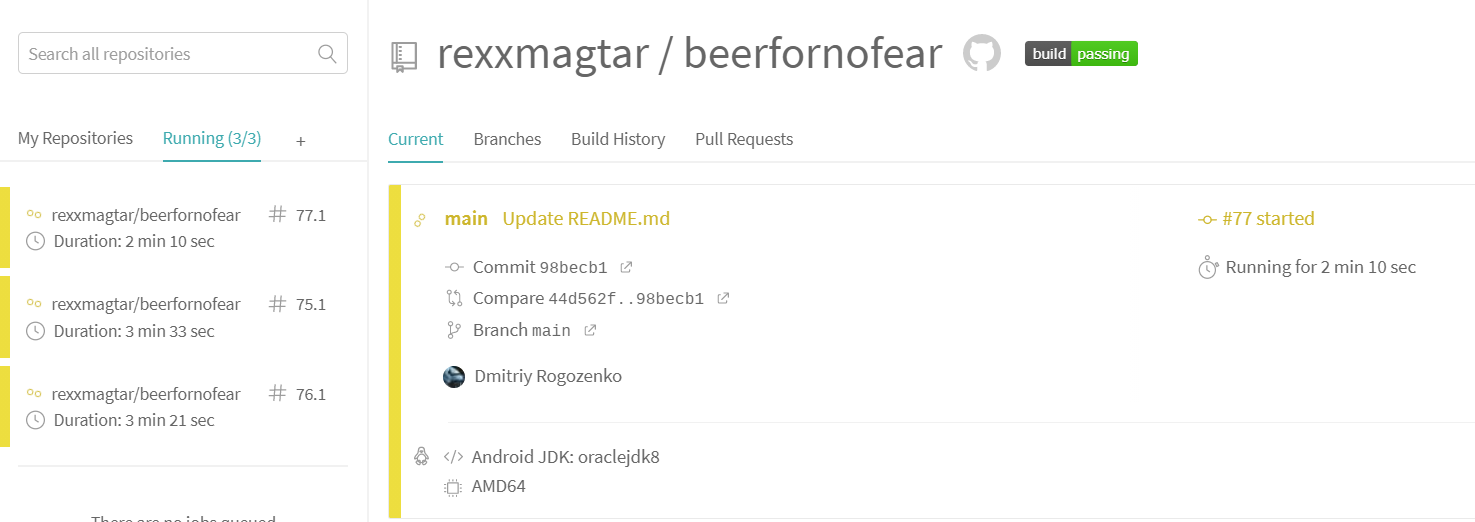
Примеры тестов:

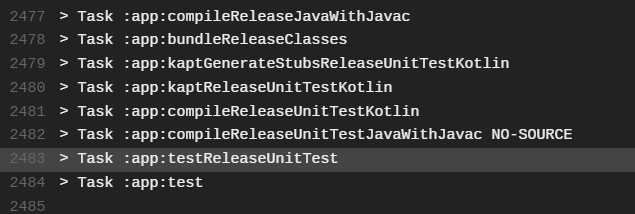
@Test  
fun testDBUpdateRow() {  
 // Context of the app under test.  
 val appContext = InstrumentationRegistry.getInstrumentation().*targetContext* // assertEquals("com.example.beerproject", appContext.packageName)  
 val dbHelper = DataBase(appContext)  
  
 val idVodka = dbHelper.insertIntoEventTable("drink\_vodka",  
 "Svayak is coming", "31.12.2020 23:55")  
  
 assertEquals(1, dbHelper.updateDataInEventTable(idVodka.toString(), "Drink Every Day",  
 "GO DRINK!", "25.12.2020 12:35"))  
  
 dbHelper.deleteRowFromEventTable(idVodka.toString())  
}  
  
@Test  
fun testDBDeleteRow() {  
 val appContext = InstrumentationRegistry.getInstrumentation().*targetContext* val dbHelper = DataBase(appContext)  
  
 val idVodka = dbHelper.insertIntoEventTable("drink\_vodka",  
 "Svayak is coming", "31.12.2020 23:55")  
  
 assertEquals(1, dbHelper.deleteRowFromEventTable(idVodka.toString()))  
}

Для сборки и тестирования использовалась платформа travis ci.

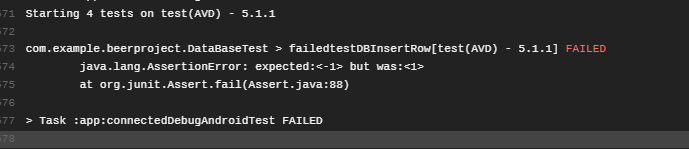
Фрагмент работы с travis ci.











Фрагмент .travis.yml

language: android

jdk: oraclejdk8

sudo: **false**

os: linux

android:

components:

- tools

- platform-tools

- build-tools-22.0.1

- android-22

- extra-google-m2repository

- extra-android-m2repository

- extra-android-support

# Specify at least one system image,

# if you need to run emulator(s) during your tests

- sys-img-armeabi-v7a-addon-google\_apis-google-22

- sys-img-armeabi-v7a-android-22

licenses:

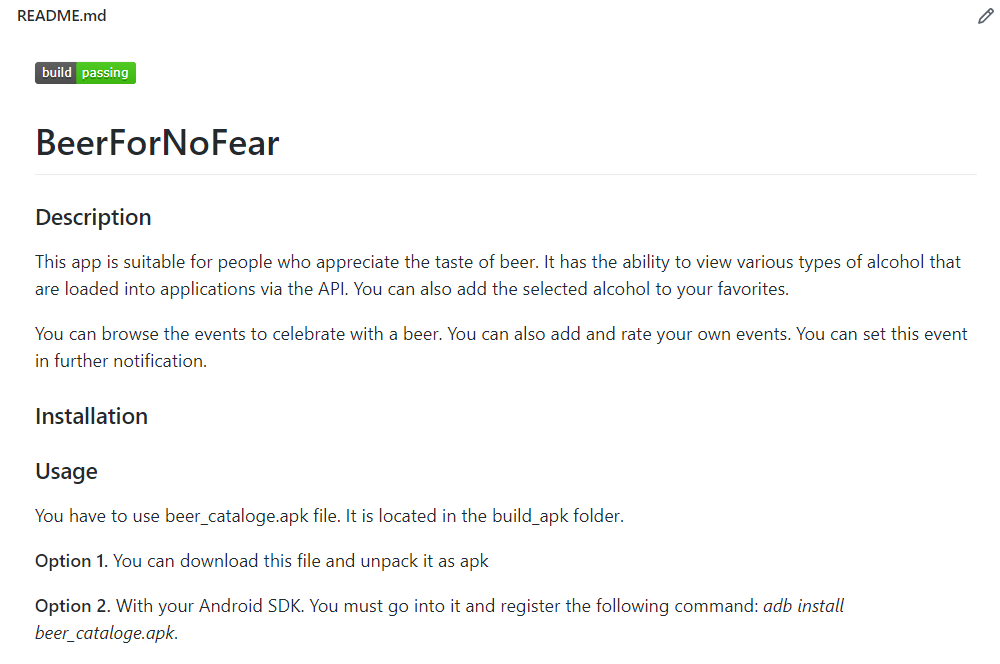
- 'android-sdk-license-.+'

- 'google-gdk-license-.+'

- 'android-sdk-preview-license-.+'

## Документация

Фрагмент README



Фрагмент Wiki

