

«Национальный исследовательский университет  
«Высшая школа экономики»  
Лицей

Индивидуальная выпускная работа  
**ИТ-ПРОЕКТ: Туалеты Тут и Там**  
<https://github.com/kepper104/toilet-finder-project>

*Выполнил(а): Пучков Кирилл Дмитриевич*  
*Группа: ИИ2*

Москва – 2023

## **1. Вступление**

После изучения языка Python и создания на нем нескольких проектов, я захотел попробовать себя в другой области разработки ПО. Я заинтересовался тем, как создаётся то, чем мы пользуемся каждый день - мобильные приложения. Данная ИВР стала для меня хорошей возможностью освоить новый стек технологий и создать собственное полноценное и практичное приложение.

## **2. Проблемное поле**

Выбор темы приложения оказался не слишком сложным: думаю, что почти каждый человек когда-то, идя по улице, задавался вопросом “А где ближайший туалет?” и, возможно, даже пользовался одним из немногих приложений, предложенных в интернете. Однако несмотря на наличие приложений, отвечающих на данный вопрос, это не помогало: приложения низкосортные, зависают, устарели, полные рекламы, платные или не имеют базы туалетов и пользователей в нашем регионе. Проведя своё мини-исследование и придя к таким выводам, я понял, что стоит заполнить эту нишу, и делать приложение, решающее проблему легкодоступной, систематизированной информации о туалетах поблизости.

## **3. Образ продукта**

Создаваемое приложение должно превзойти все другие подобные продукты, собрав от каждого самые удобные функции и интересные концепты. Так, пользователь должен иметь возможность пользоваться приложением без регистрации, при запуске приложения сразу видеть своё местоположение на карте, а также местоположения ближайших туалетов. Для удобства, туалеты также можно просматривать в виде списка, где они отсортированы по расстоянию и о каждом туалете можно одним взглядом получить всю самую важную информацию. Приложение должно быть удобно для всех пользователей, поэтому у каждого туалета есть информация о его удобствах для матерей, а также людей, с ограниченными возможностями передвижения. Сам же банк данных о туалетах в регионе пополняется пользователями приложения. Зарегистрированные пользователи могут создавать новые туалеты на карте,

указав всю необходимую информацию. Другие пользователи могут оставлять оценки и отзывы к туалетам, а также жаловаться на конкретный туалет, если указанная информация недостоверна.

#### **4. Сделанный продукт**

Конечный продукт реализует все запланированные пользовательские сценарии, которые кратко описаны ниже.

- [ Сценарий 1 - "Логин пользователя"]

Сценарий реализован полностью. Приложение делает запрос на сервер, проверяя данные пользователя (логин, пароль).

- [ Сценарий 2 - "Регистрация пользователя"]

Сценарий реализован полностью. Приложение делает запросы на сервер для проверки занятости логина, а затем для регистрации. Также можно продолжить работу в приложении без аккаунта, без доступа к некоторым функциям приложения.

- [ Сценарий 3 - " Просмотр туалетов в режиме карты и просмотр деталей туалета"]

Сценарий реализован полностью, за исключением модификации шага 12 данного сценария, предполагающего верификацию подлинности информации о туалетах. Изначально задуманная система верификации в силу комплексности реализации и потенциальных проблем была заменена на более простую и надежную кнопку «Пожаловаться» (она реализует основную задачу – сообщение пользователем о недостоверной информации).

- [ Сценарий 4 - "Настройка приложения"]

Сценарий реализован полностью. Пользователю предлагается настроить вид приложения выбором светлой или темной темы, а также цветовой гаммы карты.

- [ Сценарий 5 - "Прокладывание маршрута"]  
Сценарий реализован полностью. При выборе конкретного туалета у пользователя появляется опция проложить маршрут до туалета, используя приложение Google Maps.
- [ Сценарий 6 - "Фильтрация туалетов по удобствам"]  
Сценарий реализован полностью. На экранах карты и списка пользователю предлагается применить фильтры, уточняющие его требования к искомому туалету.
- [ Сценарий 7 - "Просмотр туалетов в режиме списка"]  
Сценарий реализован полностью. На экране списка можно листать карточки туалетов, которые отсортированы по удалению от пользователя и на каждой карточке сжато указана вся самая важная информация о туалете.
- [ Сценарий 8 - "Добавление туалета"]  
Сценарий реализован полностью, за исключением упрощения системы верификации (шаг 11). Пользователь использует интуитивный интерфейс, чтобы выбрать местоположение нового туалета и добавить всю информацию.
- [ Сценарий 9 - "Добавление отзыва"]  
Сценарий реализован полностью. Зарегистрированные пользователи могут оставлять как оценки (рейтинг туалета от 0 до 5), так и отзывы, включающие рейтинг и текстовое описание.

## 5. Строеение Backend

Большая часть проекта — это сервер, на котором находится база данных PostgreSQL, используемая написанным мной REST API сервером на Python и Flask, отвечающим за коммуникацию мобильного приложения и сервера. База данных имеет три таблицы: `users_`, `toilets_` и `toilet_reviews_`.

## 6. Средства разработки

Приложение написано только под платформу Android на рекомендованном Google языке Kotlin. Также используется рекомендованный Google реактивный

фреймворк Jetpack Compose, использующийся для быстрого построения удобного интерфейса с помощью кода. Используемые внутри Android проекта библиотеки включают: compose-destinations, maps-compose, Room, retrofit2, dagger-hilt. Для планирования проекта был использован сайт Trello.com, для проектировки базы данных - erd.dbdesigner.net .

## **7. Этапы работы над продуктом**

Этапы работы над продуктом включали:

- Первоначальная заявка. Отправлена до дедлайна, получен примерный образ продукта. После уточнения и реформулировки пары пунктов заявки был получен зачёт.
- Пользовательские сценарии. Всего подготовлено 9 сценариев, описывающих функционал будущего продукта. Сценарии помогли спланировать, как приложение должно быть построено в части интерфейса и общей структуры.
- Выбор инструментов и фреймворков для backend. Сразу было решено, что backend будет запущен на моём домашнем сервере и будет использовать Python для создания API и взаимодействия с базой данных. Выбор базы данных был между MySQL, с которой я работал ранее на нескольких проектах, и не менее популярной PostgreSQL. Решено выбрать Postgres, чтобы познакомиться и получить опыт работы с ещё одной системой.
- Выбор инструментов и фреймворков для приложения. Изучив актуальные и рекомендованные технологии, я выбрал Kotlin и Jetpack Compose (альтернативой были бы Java и XML based views, но они считаются устаревшими и были бы менее удобными для использования, даже несмотря на большее количество документации и ресурсов для изучения). Выбор API для самой карты был между Yandex Map API и Google Maps API. Сравнив документации и виджеты карт, предоставляемые двумя компаниями, был выбран Google Maps API, так как он, в отличие от Yandex, современной, имеет больше документации и информации в сети и поддерживает используемые мной Kotlin и Jetpack Compose.

- Проба пера. После изучения основ технологий, примерно за две недели я написал приложение Note Buddy – простое хранилище для заметок, которое позволило мне до конца понять многие концепты и архитектуру MVVM, которую я позже использовал в основном приложении.
- Написание backend. Пройдясь по пользовательским сценариям и определив структуры данных, которые будут нужны, я создал схемы таблиц базы данных, а также развернул их на рабочем сервере. Для написания REST API я использовал Flask, библиотека для Python, которой я уже неоднократно пользовался. Для взаимодействия Python с PostgreSQL использовалась библиотека psycopg2, а для хэширования и верификации паролей – argon2. Написав все endpoints моего API, а также Python клиент для отправки и тестирования запросов, я начал имплементировать их в приложении, используя retrofit2, популярный REST API фреймворк для Java и Kotlin.
- Финальная заявка. Окончательно понимая, что возможно реализовать из ранее заявленных требований, я отредактировал свою заявку и пользовательские сценарии – было убрано упоминание строгой привязки туалета к ресторану, кафе, или прочему заведению в случае, если туалет не общественный.
- Написание самого приложения. За три месяца было написано основное приложение.
- Тестирование. Установщик приложения был разослан нескольким родным и друзьям, которые помогли найти и исправить некоторые баги и неудобства в интерфейсе.
- Написание отчёта и KDoc (документации) кода.

## 8. Рефлексия

Написание приложения для Android получилось отличным от моего типичного опыта написания приложений из-за более строгой архитектуры и принципов мобильных приложений. Несмотря на большое количество уже

реализованных функций, у приложения есть направления для улучшения и расширения функций: добавление фото туалетов, модификация системы отзывов и автоматическая верификация без ручной проверки, а также прочие улучшения внешнего вида приложения.

Я впервые в жизни работал над проектом такого масштаба, с дедлайнами, формальным описанием требований и взаимодействием пользователя с продуктом. Мною были приобретены ценные навыки планирования, самоорганизации и доведения проекта до конца, а также навыки написания native Android приложений, которые я смогу использовать в своих дальнейших проектах и, возможно, на работе. Риски проекта включали в себя трудности работы с гео-сервисами, нехватку времени для изучения нового стека технологий, а также возможную неэффективность и проблемы с реализацией системы верификации пользователей и туалетов. В результате я столкнулся со всеми вышеперечисленными рисками, но смог преодолеть все, кроме последнего. Я недооценил сложность реализации системы голосования: она включает в себя существенное взаимодействие клиента и сервера и может быть подвержена злоупотреблению со стороны пользователя. В связи с этим она была заменена на упрощенную систему репортов пользователя, которые просто рассматриваются вручную.

## **9. Заключение**

Опыт, полученный в процессе работы над этой ИВР, был очень полезен, я получил ценное представление о том, как происходит разработка больших, используемых обычными пользователями приложений, состоящих из нескольких частей, которые должны взаимосвязанно работать, а также, насколько важны дедлайны и планирование.