УДК 004.422.8

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИНФОРМАЦИОННОГО ОНЛАЙН СЕРВИСА

Чоповенко А. О., Артемов А. О.

Национальный технический университет Украины «Киевский политехнический институт», Украина, Киев

В данной статье рассмотрена проблема нахождения достопримечательностей и сведений о событиях городов. В статье обобщен новый материал по исследуемой теме, а также проанализированы существующие решения и их недостатки. На основе проведенного исследования было принято решение использовать архитектуру MVP с дальнейшей оптимизацией приложения. Главное достоинство: анализ оптимизации практической деятельности, на основе которого было разработано приложение на платформе Android, которое бы удовлетворяло большинство требований. Такой взгляд будет интересен специалистам в области разработки программных продуктов для операционной системы Android.

Ключевые слова: Android разработка, оптимизация производительности, пользовательский интерфейс, бизнес логика приложений, API интеграция.

Чоповенко А. О., Артемов А. О. Мобільний додаток для інформаційного онлайн сервісу / Національний технічний університет України "Київський політехнічний інститут", Україна, Київ

У даній статті розглянута проблема знаходження пам'яток і відомостей про події міст. У статті узагальнено новий матеріал з досліджуваної теми, а також проаналізовані існуючі рішення і їх недоліки. На основі проведеного дослідження було прийнято рішення використовувати архітектуру MVP з подальшою оптимізацією додатки.

Головне достоїнство: аналіз оптимізації практичної діяльності, на основі якого було розроблено додаток на платформі Android, яке б задовольняло більшість вимог. Такий погляд буде цікавий фахівцям в області розробки програмних продуктів для операційної системи Android.

Ключові слова: Android розробка, оптимізація продуктивності, призначений для користувача інтерфейс, бізнес логіка додатків, API інтеграція.

Chopovenko A. O., Artemov A. O. The mobile application for online information service / National Technical University of Ukraine "Kyiv Polytechnic Institute", Ukraine, Kiev

This article deals with the problem of finding information on attractions and events in the city. The article summarizes the new material on the subject in question, as well as the analysis of existing solutions and their weaknesses. On the basis of the study, it was decided to use the MVP architecture to further optimize applications. The main advantage: the analysis of optimization practices, on the basis of which an application has been developed on the Android platform, which would satisfy most requirements. Such a view would be of interest to specialists in the field of software development for the Android operating system.

Key words: Android development, performance optimization, user interface, business logic applications, the API integration.

Введение. В большем городе трудно найти достопримечательные места или узнать о событиях, которые происходят или будут происходить в нем. Эта проблема связана с большим количеством не актуальной информации и актуальна как для жителей города, так и для туристов. Существующие информационные ресурсы ограничены в количестве информации и являются полезными только в совокупности. В связи с этим возникает потребность в объединение как можно большего количества источников в один ресурс с постоянно обновляющейся базой данных.

Анализ проблемы и осмотр существующих решений. Сравнивая такие аналоги как: «Интересный Киев», «Киев City Guide», «Киев Достопримечательности», «Афиша» и другие подобные мобильные приложения, помимо ограниченного количества данных и их неактуальности имеют и более серозные недостатки. В большей степени это не удобный интерфейс приложения, устаревшие и неактуальные базы данных, отсутствие поиска и разбития объектов по за раннее отсортированным категориям, а также недостаточно информативное представление данных. Немало важным является гибкое API ресурса, благодаря которому клиент сможет получать нужные ему данные, не прибегая к их обработке на клиентской стороне. То есть, ответ сервера на запрос с клиента должен содержать только те поля и данные которые запросил клиент, для уменьшения трафика и нагрузки на серверную часть.

Решение проблемы. Благодаря появлению сервиса KudaGo.com необходимость в самостоятельном поиске и организации данных пропадает. Так же сервис предоставляет закрытый API для реализации кроссплатформенных клиентских приложений. API реализовано благодаря технологии REST [1] (рисунок 1), а данные предоставлены в структуре Json [2]. Получив доступ к API было начато работу над Android [3] версией приложения.

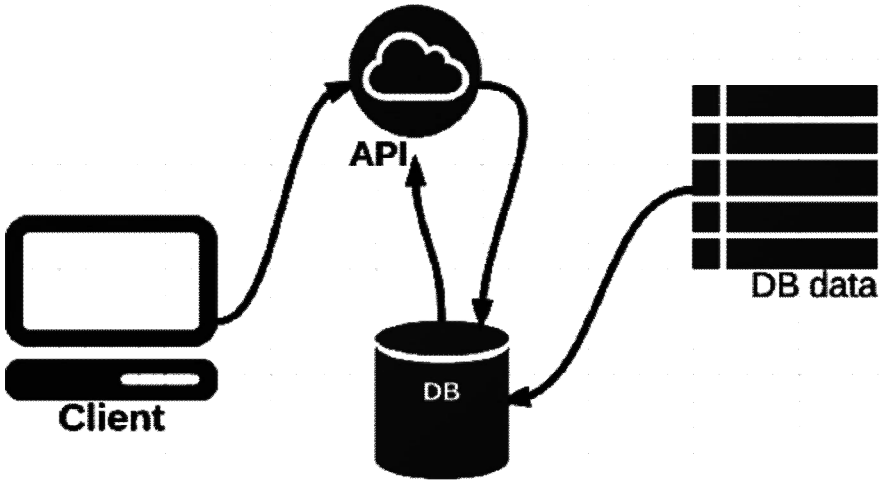

Рисунок 1. Организация REST API

Архитектура приложения изначально реализована таким образом, что при изменении данных на сервере или появление новых городов и категорий, приложение не нуждается в доработке или критическом изменение программного кода. Не смотря на загрузку информации с сервера, изображения и текст кэшируются в внутренней памяти устройства, а устаревшая информация удаляется, что позволяет эффективнее использовать память [4].

В связи с большим количеством запросов на сервере в приложении реализован механизм экспоненциальной выборки [5] для предотвращения блокировки запросов. Таким образом, если сервер блокирует запрос, клиент оживает, некоторое время, перед повторной отправкой запроса. Время повторной отправки рассчитывается по формуле: $T = x^2 + n * log4$, где n – номер попытки запроса.

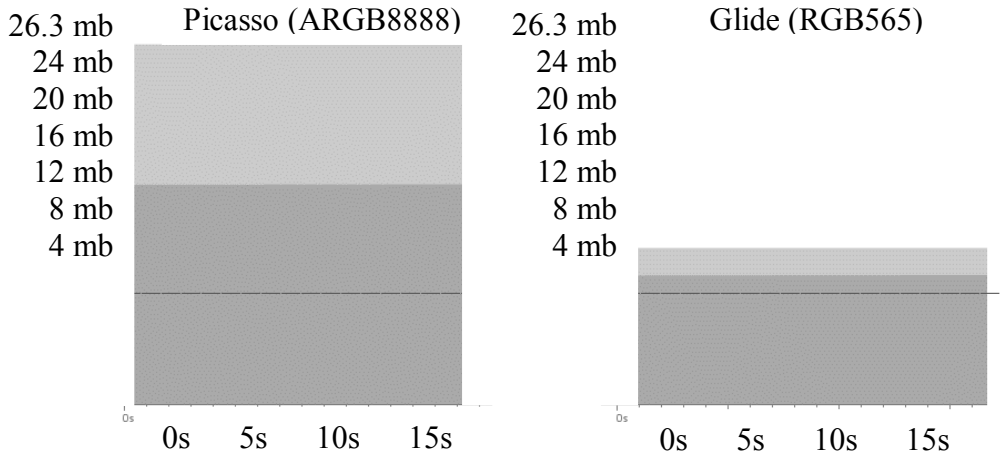

Рисунок 2. Разница в используемой памяти двух библиотек

Дизайн приложения разработан на основе Material Design Guidelines [6], актуальной версии рекомендаций Google по созданию графического интерфейса для мобильных приложений на базе ОС Android. Для подгрузки изображений с сети было принято решение использовать библиотеку Glide, которая не только быстрее всем известной Picasso от Google, но и использует меньше памяти для кеширования уже загруженных изображений. (рисунок 2) Скоп данных предоставлен в виде карточек

(CardView) с наиболее необходимой информацией в данной категории, а более детальная информация отображается при нажатии на карточку и переходе на следующий экран приложения.

Для удобной работы с базой данных SQLite, в программе используется ORM (Object-Relational Mapping) ActiveAndroid. ActiveAndroid — библиотека, позволяющая работать с БД без написания SQL-запросов. Библиотека может сохранять объекты в таблицу и извлекать их также в виде объектов (рисунок 3). Данная ORM очень схожа с такими известными библиотеками, как Hibernate, EclipseLink и другими и полностью покрывает понятие ORM. На рисунке ниже показано, как устроена работа ORM.

Поскольку приложение тесно завязано на работе с несколькими потоками, то было принято решение использовать библиотеку EventBus для упрощенного «общения» между ними. EventBus это реализация шаблона Издатель-подписчик (Publisher-Subscriber) (рисунок 4), при котором объекты передают и получают данные из общей шины событий (данных).

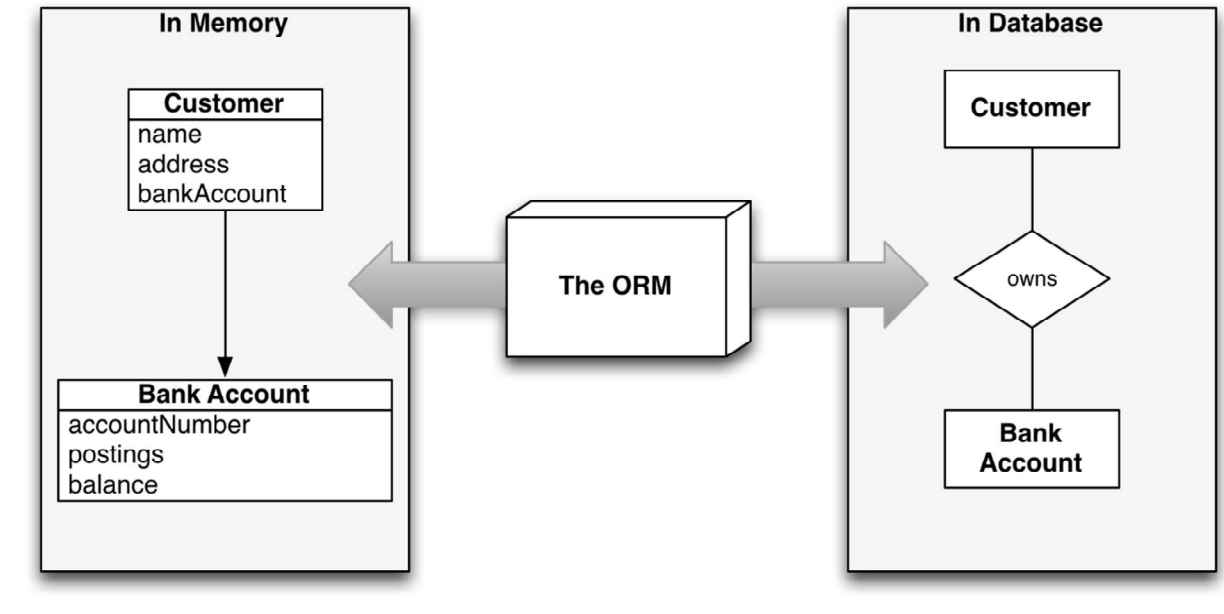

Рисунок 3. Работа с базой данных через ORM

В одном месте программы событие (event) в виде объекта передается в шину событий, а в класс, где реализован метод приема данного объекта приходит данное событие.

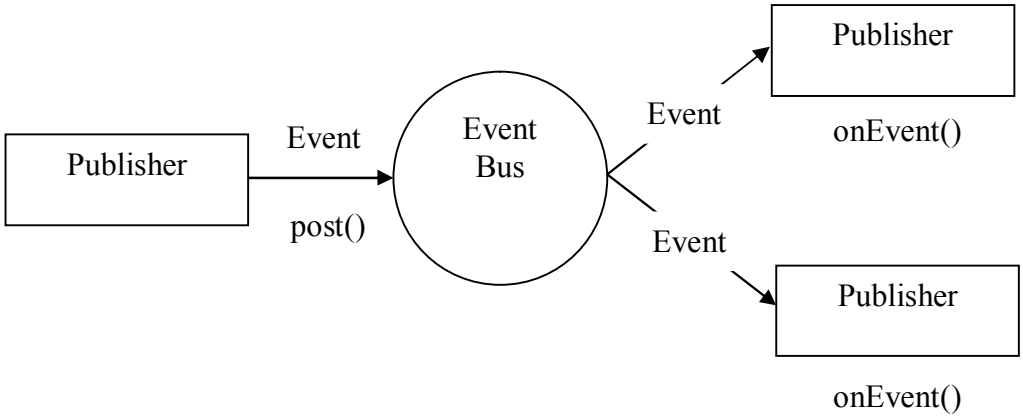

Рисунок 4. Процесс передачи event через EventBus

Для оптимизации использования мобильного трафика были использованы упрощенные запросы и тем самым клиент-серверные операции получили неплохой прирост в скорости выполнения запросов и получения ответов от сервера. Для подключения к серверу мы используем OkHttp - это альтернативный HTTP-клиент, основанный на исходных кодах HttpURLConnection, и реализующий множество дополнительных полезных функций. В частности, в OkHttp:

- добавлена поддержка протоколов HTTP 2 (draft), SPDY 3 (draft);
- реализовано автоматическое восстановление соединения, при возникновении распространенных сетевых проблем (например, проблем с прокси-сервером);
- реализован пул соединений, обеспечивающий повторное использование HTTP и SPDY соединений, за счет чего увеличивается пропускная способность и снижается время ожидания.

Для формирования запросов используется известная библиотека Retrofit. Описание запросов к серверу происходит в интерфейсе. Над каждым методом должна стоять аннотация, с помощью которой Retrofit «узнает», какого типа запрос. Так же с помощью аннотаций можно указывать параметры запроса.

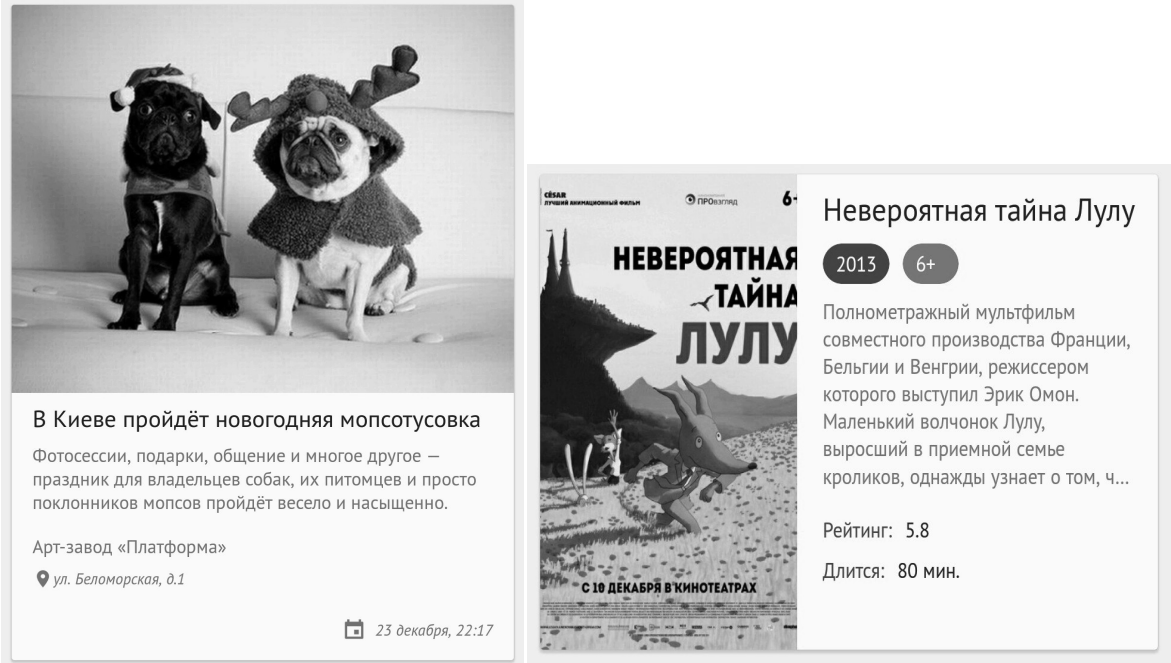

Рисунок 5. Представление данных в различных категориях

Помимо стандартных категорий, таких как Новости, Места, События и дополнительной категории Фильмы (рисунок 5), в приложение было реализовано несколько уникальных решений. Это поиск мест по близости (рисунок 6) и дополнительное приложение для платформы Android Wear [7] (рисунок 7). Android Wear — версия операционной системы Google Android, созданная для умных часов и других носимых устройств.При соединении смартфона на Android версии 4.3+ и умного устройства, Android Wear интегрирует в него функциональность Google Now и позволит получать входящие уведомления со смартфона на носимое устройство. Дополнительное приложение так же реализует поиск мест по близости, но данные предоставляются в виде списка мест, при нажатии на которые их можно отобразить на карте, открыть в стандартном приложение либо позвонить туда.

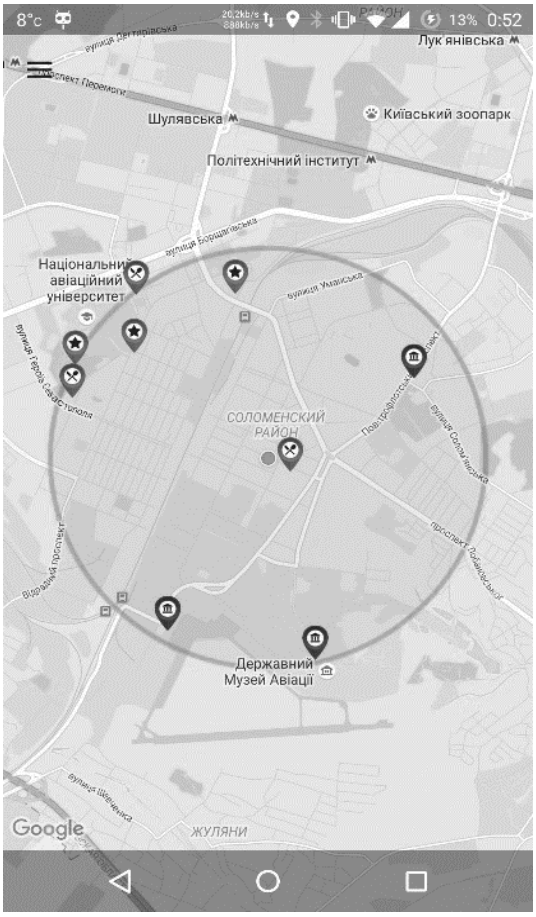

Рисунок 6. Места по близости

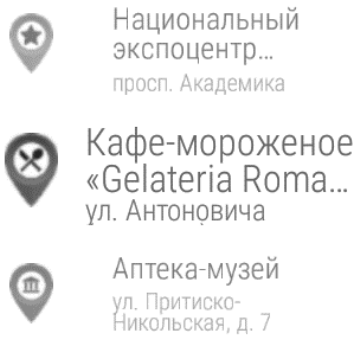

Рисунок 7. Места по близости на Android Wear

Выводы:

Благодаря недостаткам, найденным в приложениях, имеющих схожий функционал, в созданном приложение их удалось избежать. Так же была решена проблема использования излишнего количества оперативной памяти и мобильного трафика.

Литература:

1. REST [Электронный-ресурс] / Википедии [Офиц. сайт]. URL: https://ru.wikipedia.org/wiki/REST.

2. JSON [Электронный-ресурс] / Википедии [Офиц. сайт]. URL: https://ru.wikipedia.org/wiki/JSON.

3. Android - Architecture / [Электронный-ресурс] / Tutorialspoint [Офиц. сайт]. URL: http://www.tutorialspoint.com/android/android_architecture.htm.

4. Introduction to Glide, Image Loader Library for Android, recommended by Google [Электронный-ресурс] / The Cheese Factory: Blog [Офиц. сайт]. URL: http://inthecheesefactory.com/blog/get-to-know-glide-recommended-by-google/en.

5. Exponential distribution [Электронный-ресурс] / Wikipedia [Офиц. сайт]. URL: https://en.wikipedia.org/wiki/Exponential_distribution

6. Material design - Introduction [Электронный-ресурс] / Google [Офиц. сайт]. URL: https://www.google.com/design/spec/material-design/introduction.html.

7. Android Wear by Google [Электронный-ресурс] / Android [Офиц. сайт]. URL: https://www.android.com/wear/.

8. П. Дайтел. Android для программистов. Создаем приложения. Питер. - 2013. – С. 111-124.

9. С. Хашими. Разработка приложений для Android. Питер. -2011. – С. 50-72.

10. А. Голощапов. Google Android. Программирование для мобильных устройств. БХВ-Питербург. -2011. – С. 201-230.

References:

1. The REST [E - resource] / Wikipedia [official website]. The URL: https://ru.wikipedia.org/wiki/REST .

2. The JSON [E - resource] / Wikipedia [official website] The URL: https://ru.wikipedia.org/wiki/JSON .

3. Android - Architecture / [E - resource] / Tutorialspoint [official website]. URL: http :// www . tutorialspoint . com / android / android _ architecture . htm .

4. Introduction to Glide, Image Loader Library for Android, recommended by Google [E - resource] / The Cheese Factory: Blog [official website]. The URL: http://inthecheesefactory.com/blog/get-to-know-glide-recommended-by-google/en .

5. Exponential distribution [E - resource] / Wikipedia [official website]. The URL: https://en.wikipedia.org/wiki/Exponential_distribution .

6. Material design - Introduction [E - resource] / Google [official website]. The URL: https://www.google.com/design/spec/material-design/introduction.html .

7. Android Wear by Google [E - resource] / Android [official website]. The URL: https://www.android.com/wear/.

8. P. Daytel. Android for programmers. Create application. Peter. -2013. – S. 111-124.

9. S. Hashimi. Development applications for Android. Peter. -2011. – S. 50-72.

10. A. Goloshchapov. Google Android. Programming for mobile BHV devices - Piterburg. -2011. – S. 201-230.