

## Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Чипатаев Нурсултан

**Название:** Чипатаев Антиплагиат(без картинок).docx

**Координатор:** Саламат Калдыбеков

**Коэффициент подобия 1:1,3**

**Коэффициент подобия 2:0,7**

**Тревога:1**

**После анализа Отчета подобия констатирую следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....  
.....  
.....  
.....  
.....  
.....

16.05.19

Дата

*Handwritten signature*

Подпись Научного руководителя

## ОТЗЫВ

### НАУЧНОГО РУКОВОДИТЕЛЯ

на \_\_\_\_\_ дипломный проект \_\_\_\_\_  
(наименование вида работы)  
Чипатаев Нурсултан Келисулы \_\_\_\_\_  
(Ф.И.О. обучающегося)  
5B070400 - Вычислительная техника и программное обеспечение \_\_\_\_\_  
(шифр и наименование специальности)

Тема: «Разработка мобильного приложения для агропромышленной площадки на базе Android».

Перед студентом Чипатаевым Нурсултаном стояла задача разработать мобильное приложение для агропромышленной площадки на базе ОС android. Данная тема одна из наиболее актуальных т.к. использование информационных технологий в сфере сельского хозяйства является достаточно перспективным направлением. Приложение должно представлять из себя онлайн доску сельскохозяйственных объявлений.

Был использован официальный стек технологий для разработки мобильных приложений под ОС android: Язык программирования Java, интегрированная среда разработки Android Studio и средство сборки Gradle.

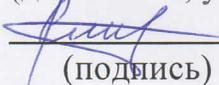
Чипатаев Нурсултан успешно освоил MVP архитектуру android приложений и разработал удобный интерфейс. В Документации дается теоретическая информация об использовании библиотек и технологий. Необходима дальнейшая доработка и улучшение программного продукта.

Поставленные перед студентом задачи решались самостоятельно. Основные результаты дипломной работы внедрены и используются в ИП «Maltabu.kz», о чем свидетельствует акт о внедрении.

Подводя итоги, дипломный проект соответствует специальности 5B070400 – «Вычислительная техника и программное обеспечение», студент заслуживает присвоения академической степени «бакалавра» и может быть допущен до защиты.

#### Научный руководитель

Маг.техн.наук, сениор-лектор  
(должность, уч. степень, звание)

 Калдыбеков С. Б.  
(подпись)

«03» 05 2019г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекомм уникационных технологий

Кафедра "Программная инженерия"

Чипатаев Н.К.

Разработка мобильного приложения для агропромышленной площадки на базе  
Android

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное  
обеспечение

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

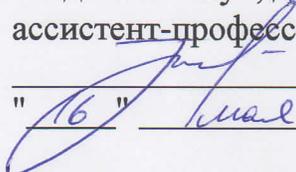
Кафедра "Программная инженерия"

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой ПИ

канд. техн. наук, доцент,

ассистент-профессор

 Р. Юнусов

" 16 " мае 2019 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

На тему: " Разработка мобильного приложения для агропромышленной  
площадки на базе Android"

по специальности 5В070400 – Вычислительная техника и программное  
обеспечение

Выполнил

Чипатаев Н.К.

Научный руководитель

Маг. техн. наук, сениор-лектор

 С.Б.Калдыбеков

" 10 " 06 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

**УТВЕРЖДАЮ**

Заведующий кафедрой ПИ  
канд. техн. наук, доцент,  
ассистент-профессор

Р. Юнусов

" 16 " мая 2019 г.

**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Обучающемуся Чипатаеву Нурсултану Келисулы

Тема: Разработка мобильного приложения для агропромышленной площадки на базе Android

Утверждена приказом проректора по академической работе № 1162 - б

от "16" октября 2018 г.

Срок сдачи законченного проекта

"17" мая 2019 г.

Исходные данные к дипломному проекту: Паспорт проекта, техническая документация по применению технологии, техническое задание.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) реализация архитектуры мобильного приложения в соответствии с концепцией MVP;

б) проектирование и разработка пользовательского интерфейса;

в) разработка приложения;

г) отладка и тестирование программного комплекса.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 15 слайдов презентации.

Рекомендуемая основная литература: из 7 наименований.

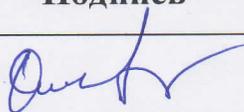
## ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области, разработка технического задания на проектирование программного комплекса	14.01.19	-
2. Реализация классовой структуры модели базы	17.02.19	-
3. Разработка пользовательского интерфейса, реализация представлений для исполнения функций "Авторизация", "Регистрация", "Подача объявлений"	11.03.19	-
4. Разработка структуры приложения с реализацией логической цепочки иерархических вызовов	21.03.19	-
5. Разработка дизайна интерфейса. Тестирование приложения	3.04.19	-
6. Написание пояснительной записки к дипломному проекту	15.04.19	-

### Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	З.М.Өмірбекова ассистент	6.05.19.	
Нормоконтролер	Г.А.Омарова лектор	6.05.19.	

Научный руководитель \_\_\_\_\_

С.Б. Калдыбеков

Задание принял к исполнению обучающийся \_\_\_\_\_

Н.К. Чипатаев

Дата

"10" "05" 2019 г.

## АННОТАЦИЯ

Данная дипломная работа предназначена для агропромышленной площадки. Созданное решение представляет собой мобильное приложение – онлайн доску объявлений для платформы андроид. Целевой аудиторией сервиса являются частные лица, желающие продать или купить сельскохозяйственных животных и продукты фермерства, а также предприятия сельскохозяйственного производства. Функционал приложения предполагает использование таких ресурсов мобильного устройства как звонки, камера и карта памяти поэтому требует специальных разрешений.

В этой статье описан процесс планирования, проектирования и разработки данного проекта. Начало посвящено исследованию предметной области и сравнению аналогичных проектов. В следующем разделе происходит анализ технологий, в результате которого был выбран нативный подход к разработке приложения и соответствующий ему стек технологий. Для написания приложения были использованы официальные инструменты разработки для ОС android. Проектная часть описывает проектирование бизнес прецедентов и интерфейса приложения. Основной частью статьи является экспериментальный раздел, в нем подробно изложен процесс разработки моделей, интерфейса и функционала системы.

## АҢДАТПА

Бұл дипломдық жұмыс агроөндірістік алаң үшін жасалған. Жасалған проект андроидқа арналған мобильді қосымша. Сервистің аудиториясы – малшаруашылығымен айналысатын кәсіпкерлер мен ауылшаруашылық өндірістер. Қосымша арнайы шешімдер талап етеді өйткені ол камера, қоңыраулар және жад картасын пайдаланады.

Бұл мақалада жоспарлау және құру процесстері сипатталған. Алғашқы бөлімі зерттеуге және аналогтық проекттерді саластыруға арналған. Келесі бөлімде технологияларға анализ беріледі. Қосымшаны жасау үшін андроид операциялық жүйеге арналған ресми әзірлеу құралдары пайдаланылады. Проекттік бөлімінде бизнес-прецеденттерді және интерфейті жобалау сипатталады. Мақаланың басты бөлімі – экспериментальді бөлім. Онда жүенің модельді, интерфейсті және функционалды жүйесі толық белгіленген.

## **ANNOTATION**

This thesis is intended for the agro-industrial site. The created solution is a mobile application. Target audit services are individuals who want to sell or buy agricultural products.

This article describes the process of planning, designing and developing this project. The beginning is devoted to the research of the subject area and the comparison of similar projects. In the next section, an analysis of the technologies takes place. For the writing of the application were used official development tools for the OS android. The design part describes the design of business use cases and the application interface. The main part of the article is an experimental section, it describes the process of developing models, interface and functionality of the system.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1 Исследовательский раздел	9
1.1 Цель разработки системы	9
1.2 Термины и сокращения	10
1.3 Существующие аналоги	10
2 Технологический раздел	11
2.1 Способ разработки мобильных приложений	11
2.1.1 Нативный способ разработки	11
2.1.2 Кроссплатформенный способ разработки	11
2.1.3 Гибридный способ разработки	12
2.1.2 Сервис для хранения исходного кода	12
2.2 Преимущества используемых технологий	12
2.2.1 Преимущества Java	13
2.2.2 Android studio	13
3 Проектная часть	14
3.1 Архитектура системы	14
3.2 Архитектура приложения	14
3.3 Разработка модели бизнес-прецедентов	16
3.4 Проектирование интерфейса	18
3.4.1 Создание концепции	18
3.4.2 Создание прототипа	19
4 Экспериментальный раздел	20
4.1 Создание моделей	20
4.1.1 Создание модели Post	20
4.1.2 Создание моделей Catalog и Category	21
4.2 Создание представлений	22
4.2.1 Создание главного экрана	22
4.2.2 Создание экрана категории	23
4.2.3 Создание экрана PostDetails	24
4.3 Функциональность приложения	26
4.3.1 Подача объявления	26
4.3.2 Регистрация и Авторизация пользователя. Восстановление пароля	29
4.3.3 Продвижение объявления	31
4.3.4 Поиск объявления	32
4.3.5 Фильтрация объявлений	33
4.3.6 Смена языка интерфейса	35
ЗАКЛЮЧЕНИЕ	37
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	38
Приложение А. Техническое задание	39
Приложение Б. Текст программы	41
Приложение В. Акт о внедрении результатов дипломной работы	101
Спецификация	102

## ВВЕДЕНИЕ

Сельское хозяйство – один из ведущих секторов казахстанской экономики. В сельской местности проживает почти половина населения страны, и от степени развития сельскохозяйственного производства во многом зависит жизненный уровень не только тех, кто работает здесь, но и тех, кто в той или иной мере связан с этой сферой.

Одной из острых проблем агропромышленного производства являются невысокие оперативность и эффективность принимаемых управленческих решений ввиду недостаточного использования новых информационных технологий.

Сельское хозяйство – идеальная среда для применения информационных технологии. В связи с этим для эффективного и устойчивого роста продуктивности сельского хозяйства необходимо применять передовые информационные технологии, позволяющие выявить недостатки и привлечь инвестиции.

Одной из основных отраслей сельского хозяйства в Казахстане является животноводство. Животноводство обеспечивает население продуктами питания, а легкую промышленность – сырьем. Сегодня чтобы продать или купить скот необходимо ехать на базар, что занимает время и средства, или переплачивать посредникам.

Мобильные технологии имеют решающее значение. Использование приложений дает доступ ко всей необходимой информации и аккумулирует вокруг себя только целевую аудиторию.

Дополнительные преимущества мобильных приложений:

- интерфейс на основе привычек пользователей;
- использование функций устройства (телефон, камера, карта памяти);
- использование push-уведомлений.

Результатом данного дипломного проекта является мобильное приложение на платформе Android – Maltabu.kz. На момент проектирования приложения, уже существовали: web-приложение и мобильная версия сайта. Maltabu.kz – сервис бесплатных сельскохозяйственных объявлений, который позволит продать или купить животных, товары, услуги, продукты животноводства и фермерства у простых людей.

## 1 Исследовательский раздел

### 1.1 Цель разработки системы

Данная дипломная работа представляет собой мобильную версию сервиса бесплатных сельскохозяйственных объявлений Maltabu.kz, разработанной для платформы Android. Главным преимуществом приложения является доступность т.к. не у каждого жителя сельской местности есть персональный компьютер или ноутбук, в то время как, смартфон на ОС Android, которая является самой распространённой ОС для мобильных устройств, имеется у подавляющего большинства населения планеты. Также к плюсам такого подхода можно отнести увеличение пользователей сервиса и аккумулярование целевой аудитории.

В возможности приложения входит:

- Просмотр объявлений;
- Подача объявления;
- Создание личного кабинета;
- Поиск, сортировка и фильтрация объявлений.

Онлайн доска объявлений – популярный вид приложений в Казахстане, но Maltabu.kz один из немногих проектов в сфере сельского хозяйства.

Для использования приложения необходимо интернет-соединение, а реализация должна происходить с использованием архитектуры «Клиент – Сервер».

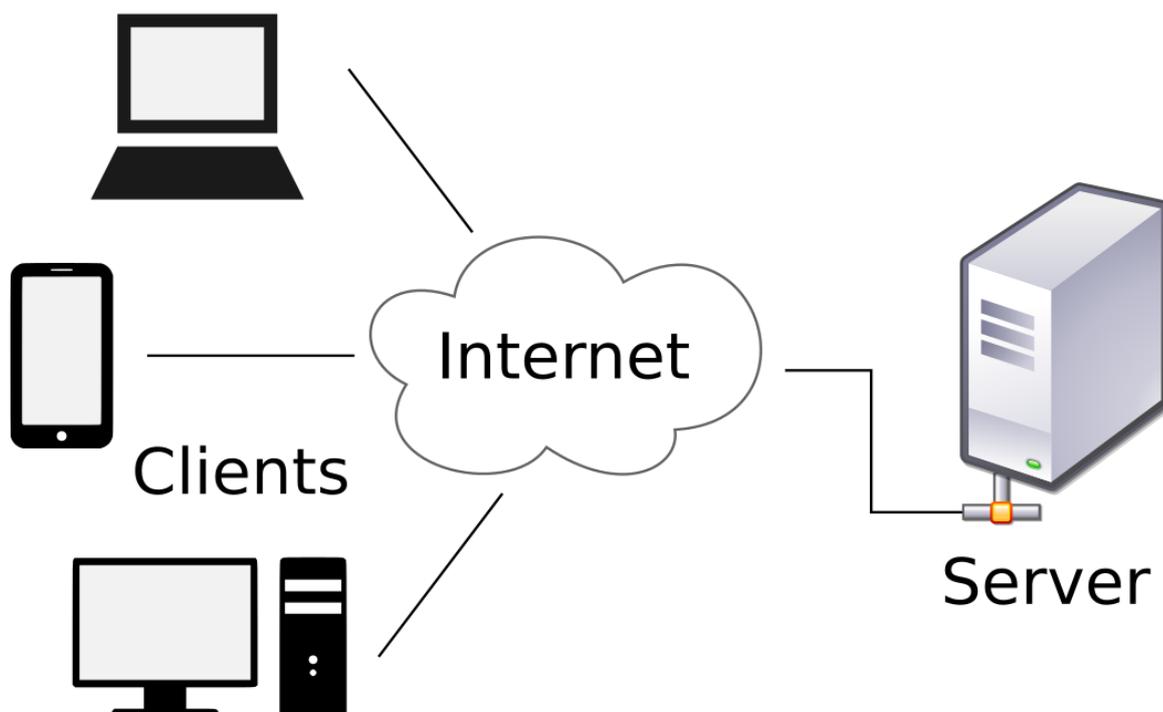


Рисунок 1.1 – Архитектура «Клиент –Сервер»

## 1.2 Термины и сокращения

Термины и сокращения, использованные при разработке, а также специфические термины, связанные с реализацией и используемыми технологиями отображены в таблице 1.

**Таблица 1 – Термины , сокращения, и их определения**

Сокращение или термин	Определение
БД	База данных
ОС	Операционная система
API	Application Programming Interface
IDE	Интегрированная среда разработки
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
SDK	Software Development Kit
JDK	Java Development Kit

## 1.3 Существующие аналоги

Данный проект берет за основу уже существующие решения в области мобильных приложений, реализующих сервис онлайн объявлений. К самым популярным можно отнести: Kolesa, Olx и Market. Эти приложения вышли в прокат несколько лет назад и уже имеют многомиллионную аудиторию.

Также, как и вышеупомянутые сервисы, наш проект избавляет покупателей и продавцов от посредников, и создает площадку для интернет торговли в своей области.

Сравнительные характеристики популярных мобильных приложений в Казахстане представлена в таблице 2.

**Таблица 2 – Сравнительные характеристики приложений**

Характеристика	Kolesa	Olx	Market
Рейтинг	4,6	4,5	4,6
Поддержка государственного языка	Нет	Нет	Нет

## Продолжение таблицы 2

Тех. поддержка	Есть	Есть	Есть
Тип разработки	Нативный	Гибридный	Нативный
Дизайн	5	4	5
Юзабилити	4,5	4,8	4,5
Цена	Бесплатно	Бесплатно	Бесплатно

## 2 Технологический раздел

### 2.1 Анализ технологии

#### 2.1.1 Способ разработки мобильных приложений

На сегодняшний день существует несколько основных подходов к разработке мобильных приложений. В зависимости от выбранной реализации определяются используемый стек технологий. Основными подходами к мобильной разработке являются нативный, кроссплатформаенный и гибридный.

##### 2.1.1.1 Нативный способ разработки

В android разработке в стек технологий при таком подходе входят язык программирования Java или Kotlin, интегрированная среда разработки Android Studio и система автоматической сборки Gradle. Эти технологии являются официальными инструментами разработки для ОС Android.

При нативной разработке приложение, как правило, имеет максимальный доступ к ресурсам смартфона и считается более предпочтительным способом.

##### 2.1.1.2 Кроссплатформенный способ разработки

Приложения написанные данным способом пишутся одновременно для всех платформ, не только android, и адаптированы к большинству устройств, потому что для их работы используется браузерный движок. Создаются на языке разметки и стилей. Лучшие инструменты кроссплатформенной разработки: Apache Cordova, Xamarin и Unity.

Еще одним популярным фреймворком для создания кроссплатформенных приложения является React Native. Этот фреймворк предназначен для создания кроссплатформенных приложений для IOS и

Android платформ, в основе которого лежит, Java script библиотека React. Данный фреймворк ориентирован на мобильные устройства.



**Рисунок 2.1 – Логотип React Native**

Основными проблемами, с которыми сталкиваются разработчики, использующие кроссплатформенную методологию, стали аппаратные и программные ограничения мобильного устройства. Этот недостаток устранится со временем – в более новых версиях все меньше ограничений, но полного доступа к ресурсам смартфона, как в нативных приложениях, в скором времени не предвидится.

### **2.1.1.3 Гибридный способ разработки**

Как понятно из названия, гибридный подход сочетает в себе преимущества нативного и кроссплатформенного способов. Смысл в том, что в приложение, написанное нативным методом вставляются графические элементы – web-view, в которых отображаются страницы, ранее написанные кроссплатформенным способом, что значительно сокращает время разработки.

### **2.1.2 Сервис для хранения исходного кода**

Жизненный цикл современных программных продуктов невозможен без обновлений. Сегодня, для безопасной разработки, необходимо пользоваться системой контроля версий и облачным сервисом для хранения исходного кода. В данной дипломной работе использовалась самая распространенная система управления версиями – git. В качестве облачного хранилища программного кода был выбран сервис Github.

## **2.2 Преимущества используемых технологий**

В связи со спецификой разрабатываемого проекта, был выбран нативный метод написания приложения т.к. определенные функции требуют полный доступ к некоторым ресурсам мобильного устройства, таким как камера,

телефон и внутреннее хранилище данных. Как говорилось ранее, при нативном способе создания android приложения используется следующий стек технологий: Java, Gradle, Android studio.

## 2.2.1 Преимущества Java

Официальными языками программирования для разработки приложений на платформе Android являются Java и Kotlin. Моим выбором стал Java потому что это самый распространенный язык программирования в мире и вокруг него образовалось самое большое сообщество программистов. Также Java – это не только язык программирования, но и огромное количество инструментов и технологий, охватывающие все этапы создания и сопровождения ПО.



Рисунок 2.2 – Логотип Java

## 2.2.2 Android studio

В качестве IDE была выбрана Android Studio. Эта среда разработки также является официальным инструментом разработки под Android, и включает в себя такие возможности как интеграция с популярными системами контроля версий, использование встроенного эмулятора, написание unit-тестов и многое др.

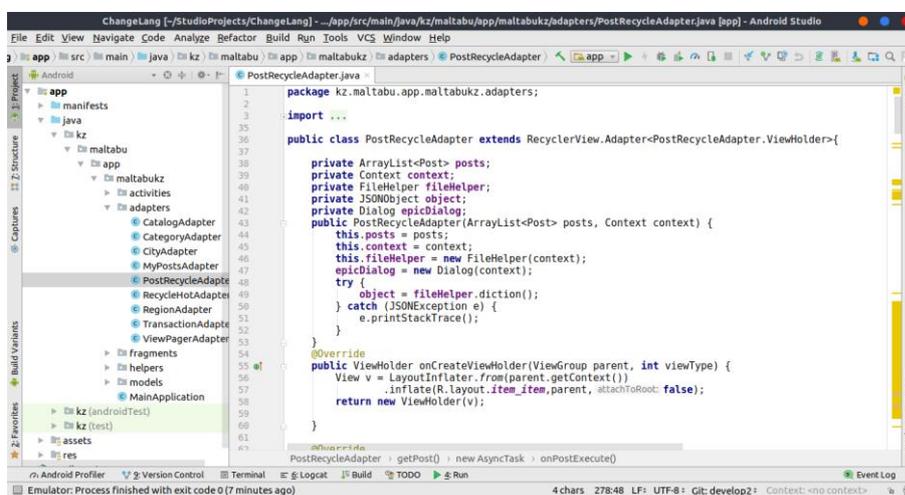


Рисунок 2.3 – Окно Android studio

## 3 Проектная часть

### 3.1 Архитектура системы

Разрабатываемое мобильное приложения является элементом системы Maltabu.kz, в которую также входят NOSQL БД – MongoDB, API написанная на nodejs, SPA приложение разработанное на веб фреймворке AngularJS.

На рисунке 3.1 изображена архитектура системы.

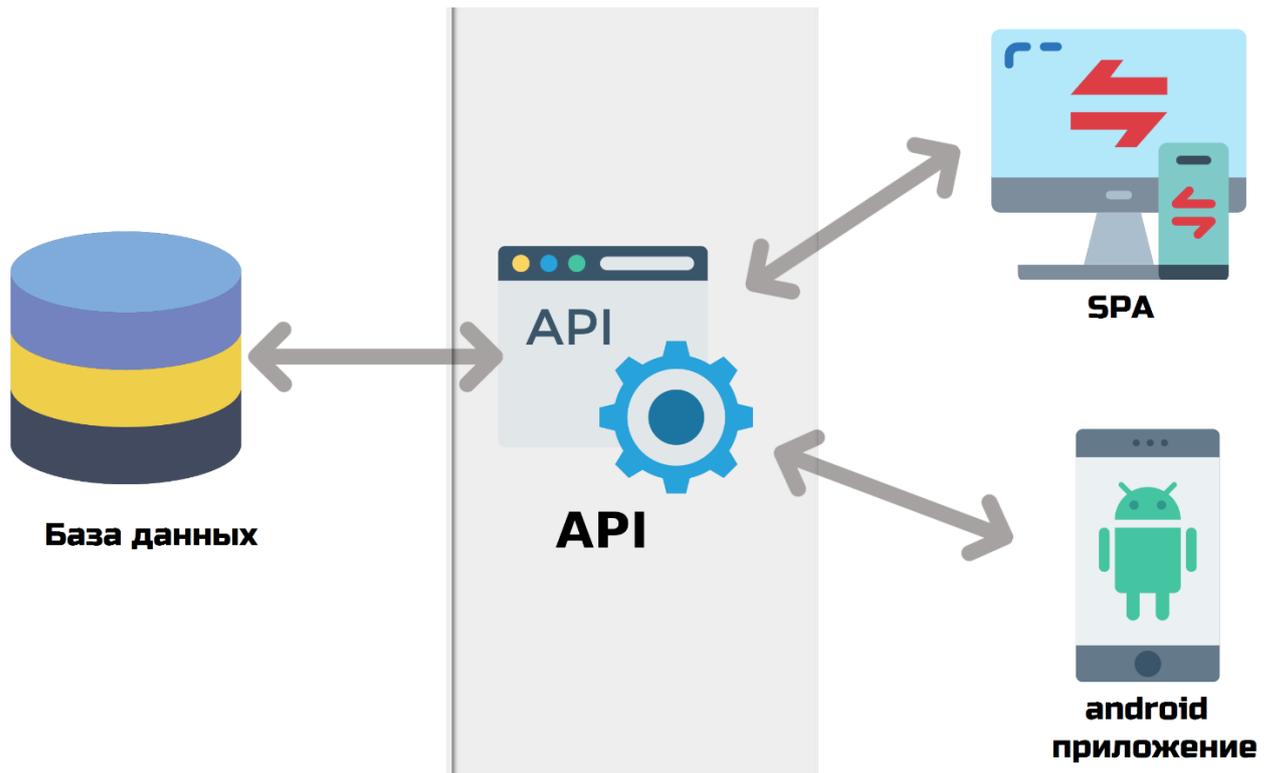


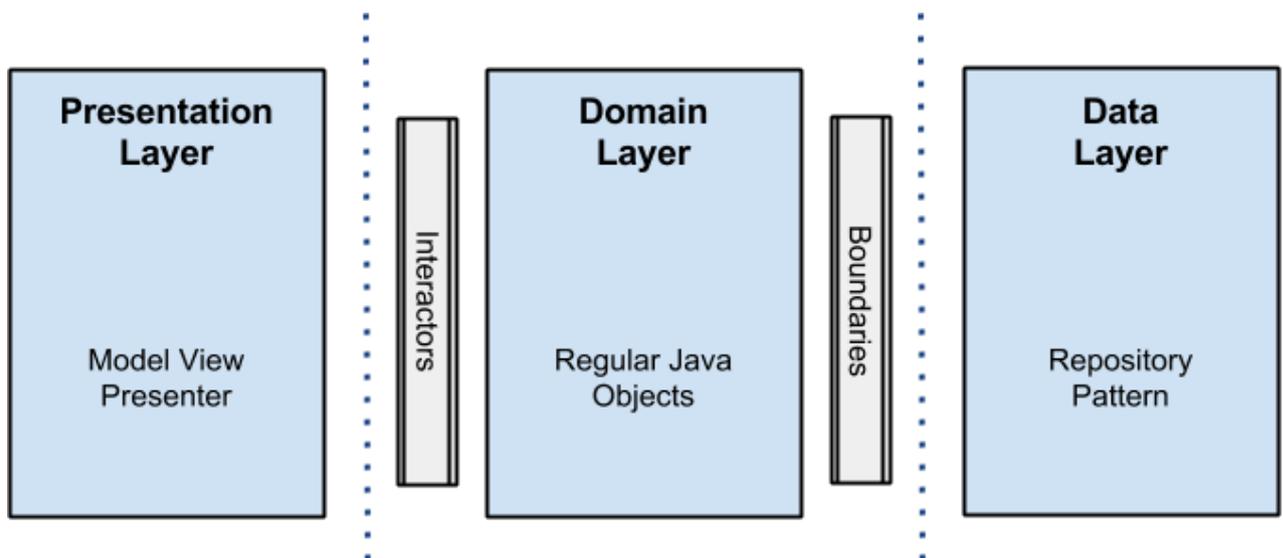
Рисунок 3.1 – Архитектура системы

Тип данного проекта – это онлайн доска объявлений, поэтому приложение постоянно взаимодействует с данными из БД через API посредством http протокола. Основная часть обработки данных производится на серверной стороне, поэтому внутри приложения большое количество http-запросов.

### 3.2 Архитектура приложения

Архитектуру данного приложения можно разделить 3 ключевых слоя:

- Data Layer (Слой данных);
- Domain Layer (Слой бизнес-логики);
- Presentation Layer (Слой представления).

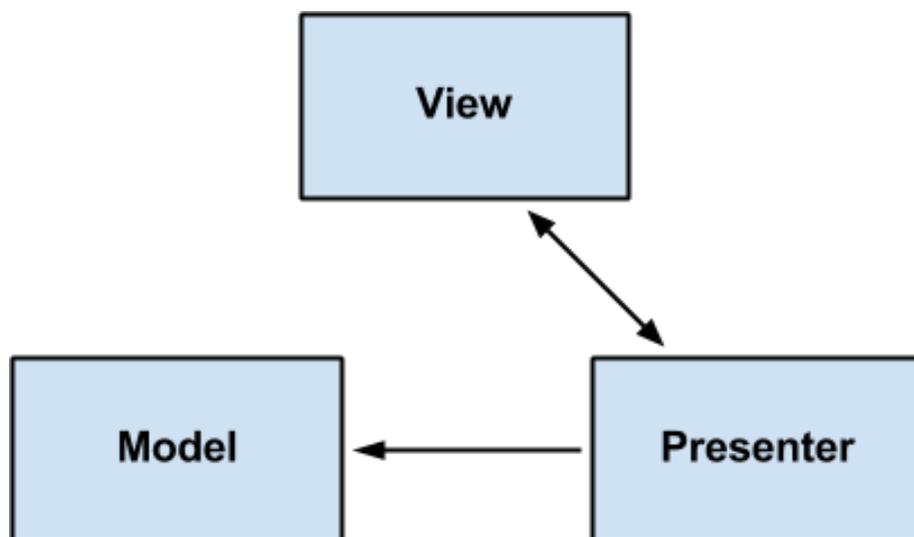


**Рисунок 3.2 – Архитектура приложения**

Как представлено на Рисунке 3.2, каждый из 3 слоев отвечает за свою логику.

В этой архитектуре слой данных, в первую очередь, – это источник информации, отвечающий за получение данных из различных источников. Главным преимуществом отделения слоя данных является то, что остальные слои не имеют доступа к источнику данных. Следующий слой – слой бизнес-логики. К нему обращается слой представления для получения данных и выполнения запросов. В проекте представлен в виде чистого модуля Java без зависимостей от Android-классов. Так происходит потому что, для реализации бизнес логики все необходимое – это, всего лишь, модели и чистый Java. Третий слой отвечает за взаимодействие с пользователем. Здесь все процессы связаны только с UI. Он связывается с экраном мобильного устройства и взаимодействует с данными из слоя данных через слой бизнес логики.

Слой представления реализован с помощью паттерна MVP



**Рисунок 3.3 – Паттерн MVP**

MVP разделяет приложение на следующие уровни:

- Model;
- View;
- Presenter.

Связь между уровнями реализуется посредством Presenter.

Первый уровень Model это – бизнес-логика приложения. Данный уровень контролирует изменения данных (создание, изменение и удаление).

View это графический интерфейс, отображающий информацию (данные). Он направляет события (действия) юзера в Presenter.

Presenter выступает в роли прослойки. Он парсит данные из Model и отображает их в View. Он также обрабатывает события юзера, полученные из View.

Основные правила по которым функционирует паттерн MVP:

- У уровня View нет доступа к уровню Model;
- Presenter отвечает только за один View.

### 3.3 Разработка модели бизнес-прецедентов

Для разработки модели бизнес-прецедентов был использован онлайн конструктор uml диаграмм createUML. В бесплатной версии конструктора есть возможность создать все виды uml диаграмм.

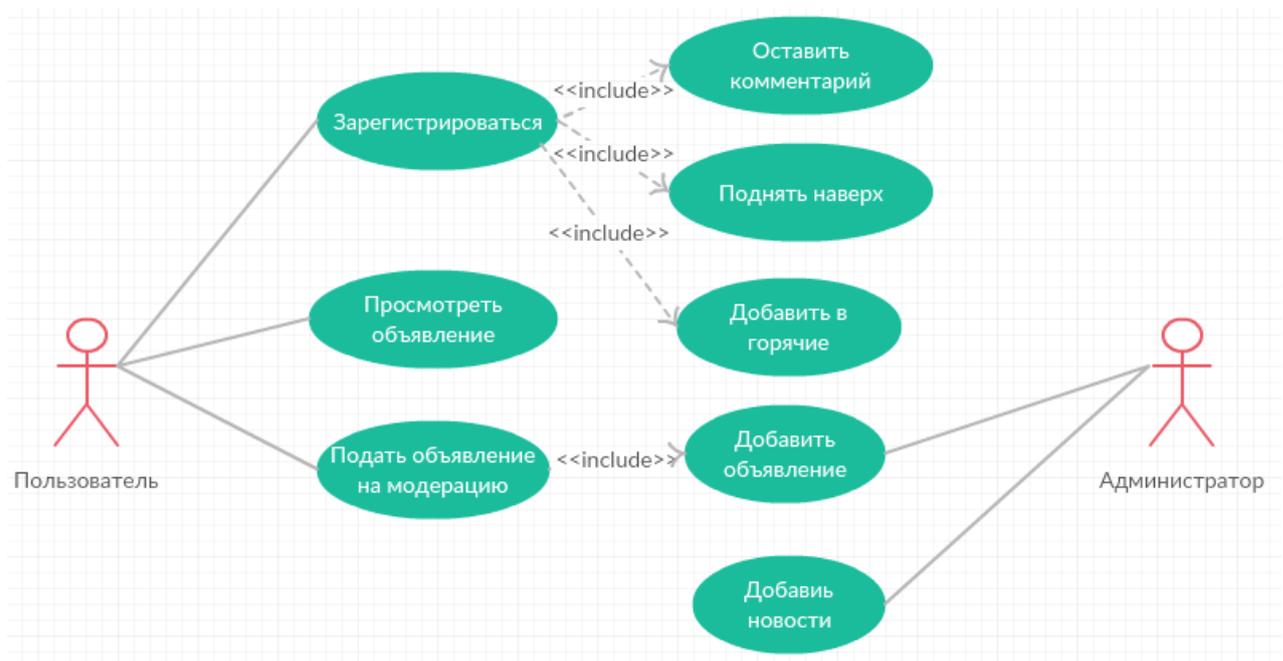


Рисунок 3.4 – Модель бизнес-прецедентов

Любой пользователь просто открывает приложение и просматривает объявления. Как видно на рисунке 3.5, пользователю приложения не обязательно регистрироваться в системе чтобы подать объявление. После подачи объявления на модерацию, оно проверяется администратором сайта и, в зависимости от решения, добавляется или отклоняется. Также администратор может добавлять новости на сайт.

Функции, которыми может воспользоваться только авторизованный пользователь:

- Оставить комментарий;
- Продвинуть объявление – поднять наверх;
- Продвинуть объявление – добавить в горячие.

Для продвижения объявления у пользователя должны быть единицы, которые пополняются через систему моментальных электронных платежей «Касса 24». Подключение этой системы – это попытка монетизации приложения, т.е. использование бесплатного приложения с возможностью покупать дополнительные платные возможности. Стоимость зависит от выбранной услуги.

«Поднять наверх» – функция которая, часто используется в приложениях, связанных с объявлениями. Если пользователь авторизовался в приложении и у него достаточно средств, то после нажатия кнопки «Поднять наверх» объявление поднимется в самое начало списка.

«Добавить в горячие» также используется в онлайн сервисах с объявлениями. При успешном выполнении функции, объявление добавляется в специальных раздел «Горячие» на 24 часа. Этот раздел расположен на стартовом экране, поэтому пользователи часто встречают объявления, добавленные в горячие. Как правило, этот способ продвижения объявления обходится дороже.

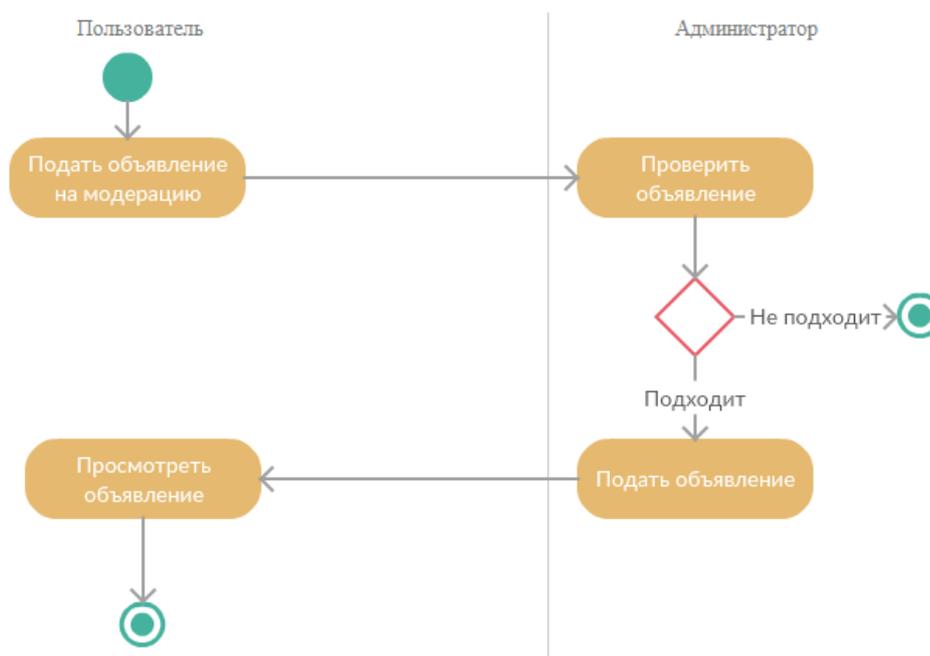


Рисунок 3.5 - Диаграмма видов деятельности

### 3.4 Проектирование интерфейса

UI мобильного приложения играет ключевую роль т.к. первая реакция пользователей на программный продукт происходит, непосредственно, через пользовательский интерфейс. Интуитивно непонятный и неудобный интерфейс оттолкнет пользователей. Прежде всего, проектировщик должен учитывать потребности пользователей. Значительно упрощает задачу определенность в целевой аудитории.

Основные задачи пользовательского интерфейса:

- Внести ясность;
- Удерживать внимание пользователя;
- Дать пользователям ощущение контроля.

Проектирование интерфейса приложения можно разделить на следующие этапы:

- Создание концепции;
- Создание прототипа.

#### 3.4.1 Создание концепции

На этом этапе определяются основные элементы UI приложения шрифты, цвета, отступы, расположение элементов на экране и др. На момент проектирования интерфейса уже было создано веб приложение, поэтому UI разрабатываемого приложения основывался на интерфейсе сайта.

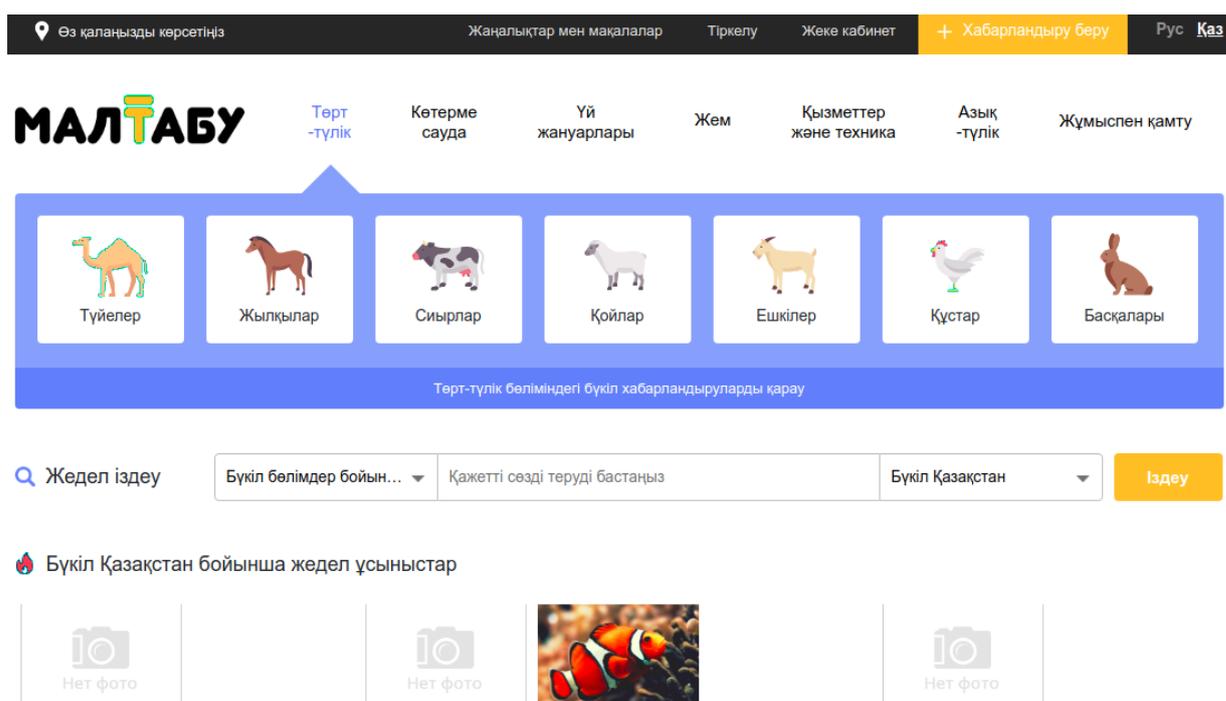


Рисунок 3.6 – Главная страница сайта Maltabu.kz

### 3.4.2 Создание прототипа

Прототипирование интерфейса велось с помощью Moqups. Moqups – это веб приложение для создания прототипов GUI приложения.

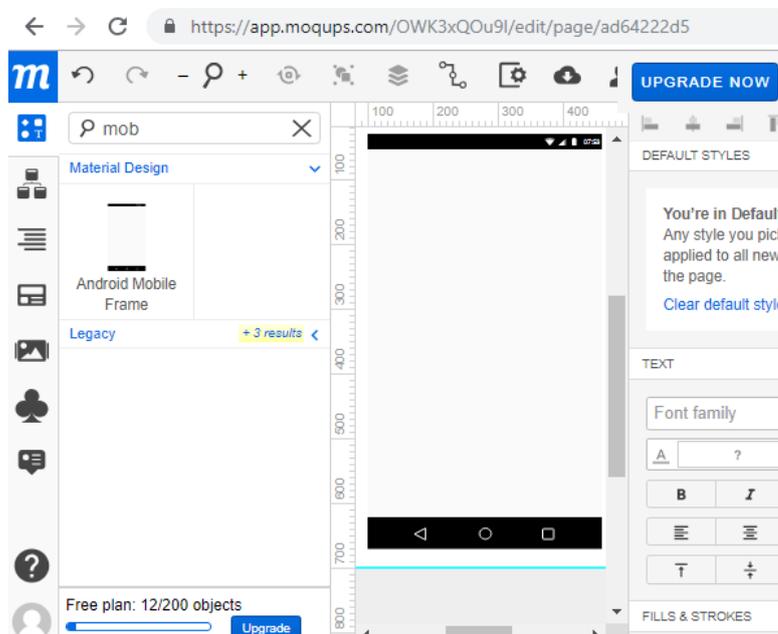


Рисунок 3.7 – Создание прототипа в moqups

Интерфейс приложения берет за основу UI аналогичных приложений. Отличительной чертой данных приложений является боковое меню, находящееся слева. В Android Studio данное меню реализовано с помощью специального элемента Navigation Drawer. Меню, как правило, разделяется на 2 части верхнюю и нижнюю. В верхней части находится логотип и ярлык поиска, в нижней – меню.

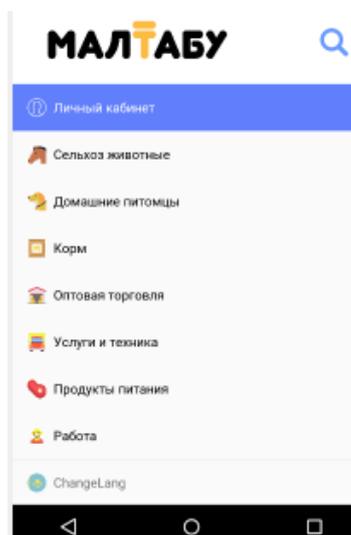


Рисунок 3.8 – Прототип бокового меню

## 4 Экспериментальный раздел

### 4.1 Создание моделей

Объявления представляют собой набор данных полученных с сервера, посредством http-запросов. Эти данные получаются в формате JSON. Основная задача конвертировать полученную информацию для дальнейшего использования. Чтобы решить эту проблему создаются модели данных, которые создаются на основе данных из сервера, и с которыми и будет происходить манипуляции.

#### 4.1.1 Создание модели Post

Модель Post представляет собой java класс, который содержит в себе всю необходимую информацию:

- Заголовок объявления;
- Дата создания объявления;
- ID объявления;
- Описание объявления;
- Массив изображений, прикрепленных к объявлению;
- Номер телефона, указанный в объявлении;
- Город или Область;
- Массив комментариев.

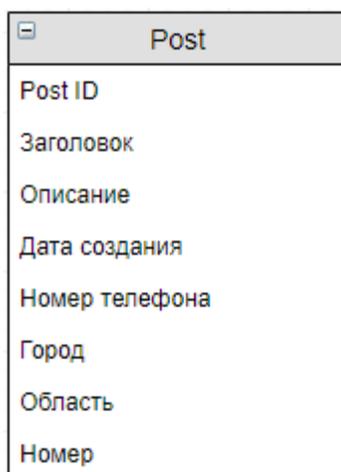


Рисунок 4.1 – Сущность Post

Вышеуказанные поля являются свойствами класса Post и являются private, поэтому доступ к ним производится только через специальные функции. Также здесь есть необходимые методы для манипуляций с объектом класса: несколько конструкторов (включая пустой), а также геттеры и сеттеры. Еще

одна особенность этого класса – это то, что он реализует Parcelable. Дело в том, что есть необходимость передавать объект класса Post в другой activity.

```
Post.java x
1 package kz.maltabu.app.maltabukz.models;
2
3 import ...
7
8 public class Post implements Parcelable{
9     private String createdAt;
10    private String phones;
11    private String updatedAt;
12    private String title;
13    private String visitors;
14    private String content;
15    private String catalogID;
16    private String categoryID;
17    private String regionID;
18    private String cityID;
19    private String price;
20    private String priceValue;
21    private String address;
22    private String number;
23    private ArrayList<Comment> comments;
24    private ArrayList<Image> images;
25    private boolean exchange;
26
27    @
46    protected Post(Parcel in) {...}
47    public static final Creator<Post> CREATOR = new Creator<Post>() {...};
57    public ArrayList<Comment> getComments() { return comments; }
60    public void setComments(ArrayList<Comment> comments) { this.comments = comments; }
63    public String getPhones() { return phones; }
66    public void setPhones(String phones) { this.phones = phones; }
69    public String getCreatedAt() { return createdAt; }
```

Рисунок 4.2 – Java класс Post

#### 4.1.2 Создание моделей Catalog и Category

Каждое объявление относится к определенному каталогу, который, в свою очередь, относится к определенной категории. Эта информация также получается с сервера в виде JSON, поэтому для нее создаются модели.

Эти модели создаются аналогично модели Post, с той разницей, что каждая категория содержит в себе несколько каталогов. на Рисунке 4.3 мы видим отношения один ко многим.

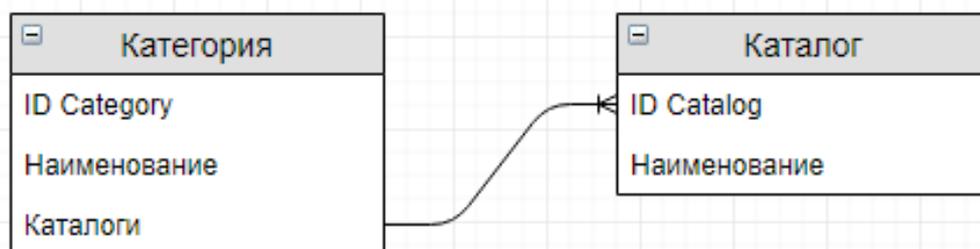


Рисунок 4.3 – Отношение сущностей Category и Catalog

Из особенностей класса Category можно отнести поле catalogs. Ему присвоен тип данных ArrayList <Catalog> т.е. поле catalogs класса Category содержит бесконечный массив объектов класса Catalog. ArrayList – это распространённая структура данных в java, которая пользуется преимуществами массивов и связанных списков. Из него можно быстро получить элемент по индексу и вставить новый.

## 4.2 Создание представлений

В качестве представлений в android studio используются документы в формате XML, расположенные в папке layout. Эти файлы описаны языком расширяемой разметки – XML. Здесь расположены все элементы графического интерфейса.

Почти для каждой Активности или Фрагмента создается свой XML-документ. Разрабатывались представления на основе прототипов UI в интегрированной среде разработки android studio.

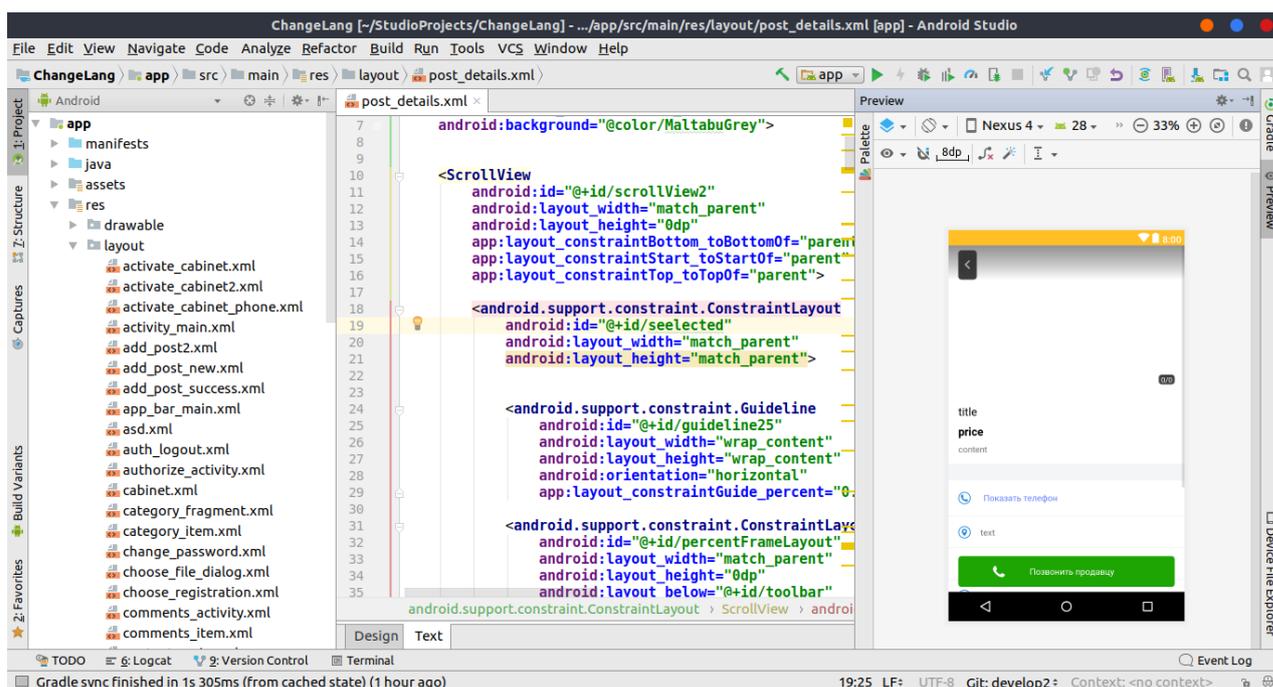


Рисунок 4.4 – Окно IDE Android Studio

При создании макетов представлений важно выбрать правильные идентификаторы элементов т.к. именно через них будет происходить взаимодействие с бизнес-логикой.

При проектировании использовались: ConstraintLayout, ScrollView, TextView, EditText, RecyclerView, Button и др. Чаще всего, как основу для макета, выбирался ConstraintView. ConstraintView – это контейнер для элементов, который обладает специальными свойствами–ограничениями благодаря которым легче компоновать элементы внутри контейнера.

### 4.2.1 Создание главного экрана

В прототипе приложения было определено наличие бокового меню. Оно реализовывалось посредством Navigation Drawer.

В верхней части меню находятся логотип и иконка поиска. При нажатии на логотип происходит замена фрагмента главного экрана на раздел Горячие. При нажатии на иконку поиска происходит замена фрагмента главного экрана на фрагмент с поиском.

В нижней части располагаются следующие разделы:

- Личный кабинет;
- Категории объявлений;
- Кнопка смены языка.

После нажатия кнопки «личный кабинет» приложение меняет текущую Активность на Активность личного кабинета (если пользователь авторизован) либо на Активность авторизации (если пользователь не авторизован).

При выборе категорий фрагмент главного экрана меняется на фрагмент выбранной категории.

Кнопка смены языка служит для смены языка интерфейса. Располагающаяся рядом иконка флага меняется в соответствии с выбранным языком. Приложение реализовано на казахском и русском языках.

#### 4.2.2 Создание экрана категории

Для экономии места на экране, было решено разметить каталоги, которые находятся внутри категории, на горизонтальном слайдере и реализовать список объявлений по каталогам в виде фрагментов. Чтобы решить эту задачу была использована связка элементов TabLayout и ViewPager. В первом элементе размещены иконки и названия каталогов, а во втором список объявлений.

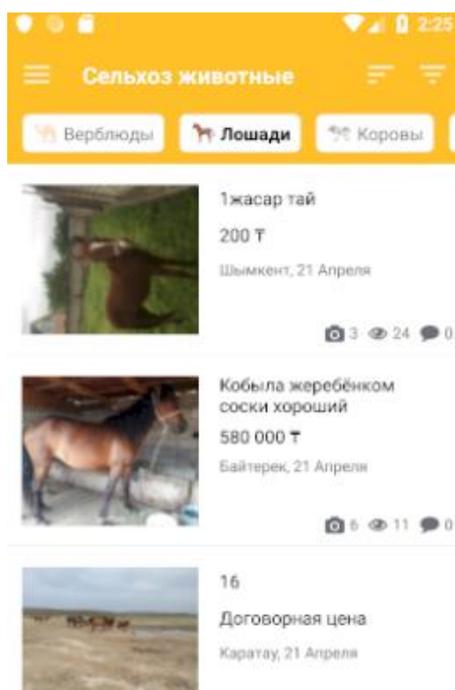


Рисунок 4.5 – Фрагмент Категория

Список объявлений представляет собой набор элементов, управляемый специально написанным адаптером и отображаемый в элементе RecyclerView. RecyclerView был выбран из-за плавности прокрутки, которая достигается благодаря статическому классу – ViewHolder, который наследуется от RecyclerView.ViewHolder.

### 4.2.3 Создание экрана PostDetails

Это представление создавалось на основе экрана подробной информации объявления в мобильной версии сайта.

Здесь представлена вся информация:

- Заголовок;
- Дата создания;
- Описание (если имеется);
- Фотографии (если имеются);
- Номер телефона;
- Цена.

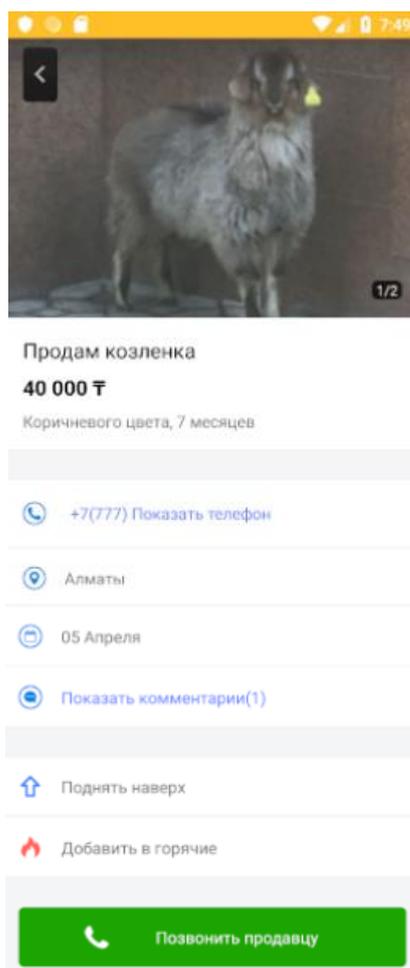


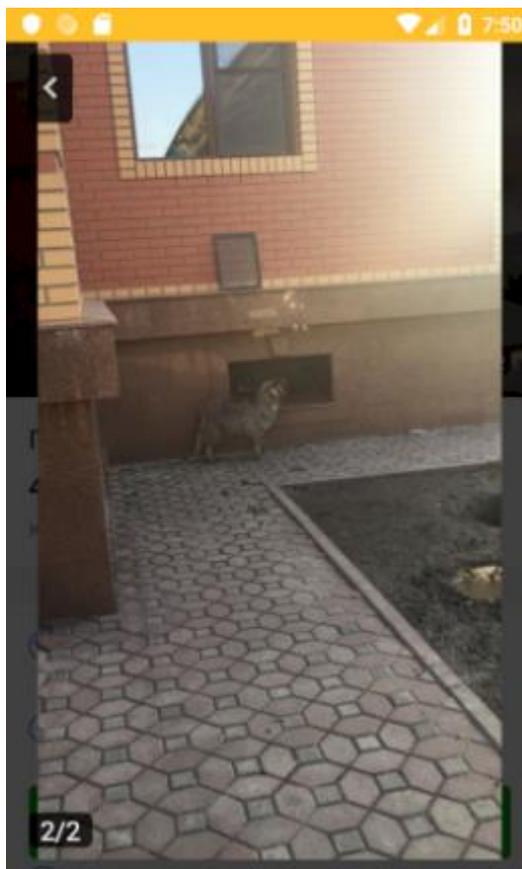
Рисунок 4.6 – Экран подробной информации

Кнопка «Показать телефон» используется для подсчетов пользователей, которые заинтересовались объявлением. Кнопка «Позвонить» предназначена для осуществления звонка по указанному, в объявлении, номеру.

Авторизованные пользователи могут оставить отзыв если перейдут на Активность комментариев кликнув по полю «Написать комментарий».

Путь к фотографиям указан в объектах класса Post. На сервере имеется 4 размера одной фотографии. Это сделано для быстрой загрузки изображений. При загрузке больших списков сообщений загружаются фотографии с меньшим разрешением, а при открытии подробного описания - фотографии с большим. Для загрузки фотографий с сервера используется библиотека Picasso. В ней указывается URL фотографии, параметры обрезки изображения и путь к фотографии которая будет отображена по умолчанию пока закончится загрузка фото.

При нажатии на фотографию открывается Активность с массивом фотографий. Здесь есть возможность увеличить фотографии для подробного просмотра.



**Рисунок 4.7 – Подробный просмотр фотографий**

Эта Активность создана с помощью библиотеки SildableActivity, которая дает возможность закрыть Активность свайпом вверх т.е. провести пальцем по экрану от нижнего края к верхнему. Этот подход к завершению Активности часто используется в мобильных приложениях. Также эту активность можно завершить нажав иконку, в виде стрелки, расположенной в верхнем левом углу.

Фотографии переключаются горизонтально. Фон этой Активности – полупрозрачный т.к. предыдущая Активность остановлена, но не удалена из оперативной памяти.

### **4.3 Функциональность приложения**

В данном приложении реализована следующая функциональность:

- Подача объявления;
- Регистрация пользователя;
- Авторизация в системе;
- Восстановление пароля;
- Продвижение объявления;
- Поиск объявления;
- Фильтрация объявлений.
- Смена языка интерфейса;

#### **4.3.1 Подача объявления**

Подача объявления в сервис Maltabu.kz, через мобильное приложение, осуществляется посредством http-post запроса. Для работы с http-запросами была использована Библиотека Okhttp3.

В тело запроса входит:

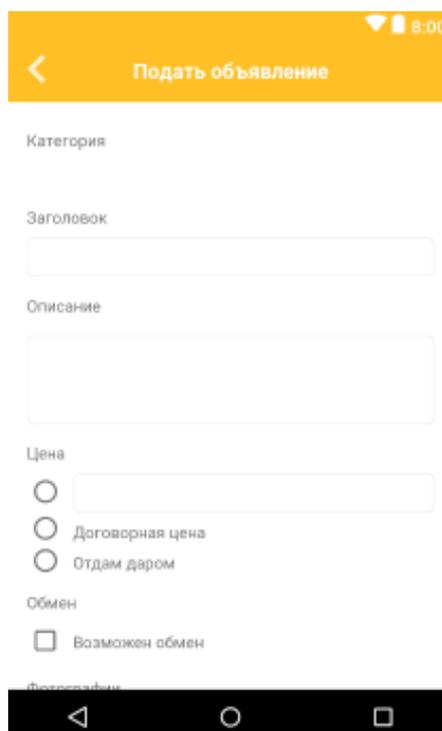
- Заголовок;
- Описание (необязательное поле);
- Цена;
- Фотографии (если пользователь решил добавить);
- Номер контактного телефона (максимальное количество – 5);
- Адрес (необязательное поле);
- Адрес электронной почты.

В заголовок запроса входит: токен (если пользователь авторизован в системе).

В приложении, для подачи объявления необходимо, в главном меню, выбрать пункт «Добавить объявление», после чего открывается Активность с выбором категории и каталога публикуемого объявления. Как только пользователь выберет каталог, открывается Активность с формой подачи объявления.

Прежде чем выполнится запрос, отправляемые данные проходят валидацию. Для этой задачи был написан специальный класс – ValidationHelper. ValidationHelper это java класс в котором созданы функции предназначенные, в основном, для работы со строковыми типами данных. Они помогают определить является данная строка адресом электронной почты, номером телефона или

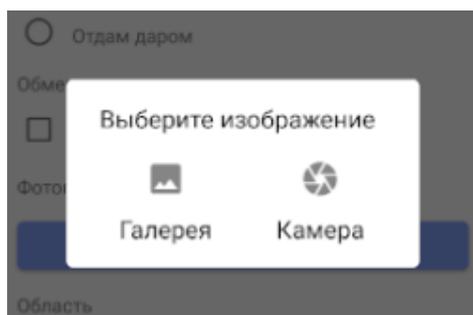
чем-то иным. Реализация происходит с использованием регулярных выражений.



**Рисунок 4.8 – Форма подачи объявления**

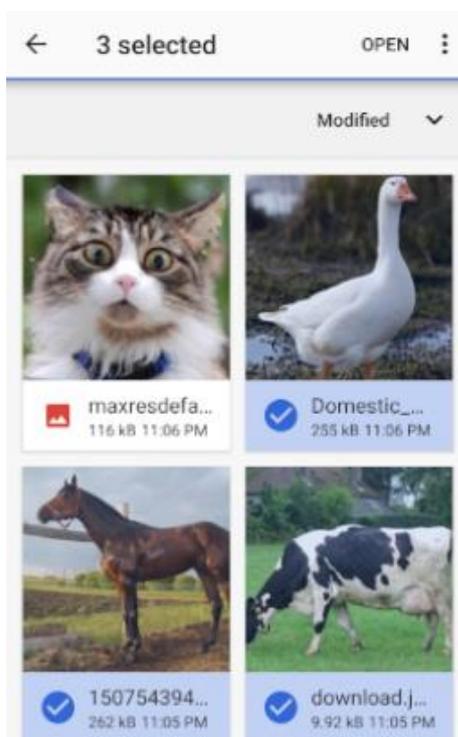
Если данные не проходят валидацию запрос на сервер не осуществляется и на дисплее показывается сообщение с указанием невалидных элементов запроса.

Для добавления фотографий пользователю нужно нажать кнопку «Добавить фотографии». После чего появляется диалоговое окно, где пользователю необходимо выбрать между галереей и камерой. Для выбора второго требуется специальное разрешение. При нежелании пользователя предоставлять доступ к ресурсам мобильного устройства, появляется кнопка перехода в настройки разрешений приложения. Если разрешение получено, пользователь может сделать снимок. Сделав снимок, приложение возвращается на форму подачи объявления и, в зависимости от ориентации камеры, поворачивает изображение.



**Рисунок 4.9 – Выбор источника фотографий**

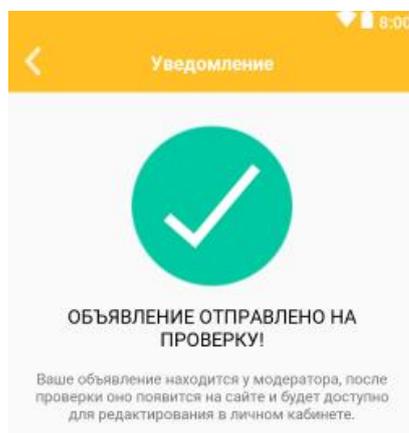
Для выбора фотографий из Галерей требуется настройка проводника файлов – fileprovider. Пользователь имеет возможность выбрать несколько фотографий. Максимальным количеством загружаемых фотографий является 7 фото.



**Рисунок 4.10 – Галерея**

Некоторые фотографии имеют слишком большое разрешение и большинство смартфонов не обладают достаточным количеством свободной памяти, поэтому было решено изменить разрешение больших изображений.

При успешном прохождении валидации осуществляется вышеуказанный запрос и Активность меняет представление в котором указано сообщение об успешной подаче объявления на модерацию. После чего объявление проверяется администратором системы.



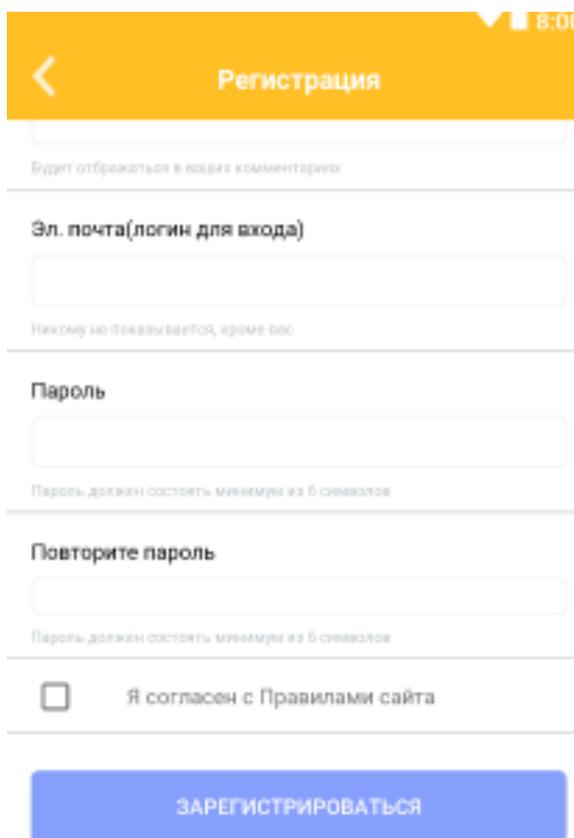
**Рисунок 4.11 – Уведомление об успешной подаче объявления**

### 4.3.2 Регистрация и авторизация пользователя. Восстановление пароля

У незарегистрированных пользователей есть возможность создать личный кабинет, с помощью которого можно продвигать сообщения и оставлять комментарии. Регистрация может проходить через номер телефона или адрес электронной почты. Это необходимо для восстановления пароля, если пользователь забыл пароль.

Регистрация в приложении происходит по тому же сценарию, что и подача объявления:

- Заполнение формы;
- Валидация;
- Формирование http-запроса;
- Уведомление об успешной регистрации.



The screenshot shows a mobile application registration screen. At the top, there is a yellow header with a back arrow and the title 'Регистрация'. Below the header, there is a text input field with a placeholder 'Будет отображаться в виде комментария'. This is followed by three form fields: 'Эл. почта(логин для входа)', 'Пароль', and 'Повторите пароль'. Each field has a small note below it: 'Никому не показывается, кроме вас', 'Пароль должен состоять минимум из 6 символов', and 'Пароль должен состоять минимум из 6 символов'. Below the password fields is a checkbox with the text 'Я согласен с Правилами сайта'. At the bottom of the form is a large blue button with the text 'ЗАРЕГИСТРИРОВАТЬСЯ'. The top right corner of the screen shows a status bar with the time '8:00'.

Рисунок 4.12 – Форма регистрации пользователя

Для авторизации в приложении используются следующие http-запросы:

- Post запрос для получения токена;
- Post запрос для получения информации о клиенте.

В тело первого запроса входит логин и пароль. После валидации с сервера поступает ответ, в виде токена, если запрос удачный. Токен сохраняется в приложении в постоянную память. Эту задачу решает класс FileHelper.

FileHelper предназначен для работы с файлами.

Сразу же после первого запроса, осуществляется следующий. В нем в тело запроса помещается токен, полученный ранее. В результате запроса возвращается информация о пользователе:

- Количество единиц на счету;
- Номер личного кабинета;
- Адрес электронной почты или номер телефона.

Номер личного кабинета нужен для пополнения счета. Эта информация должна храниться в постоянной памяти, чтобы пользователю не приходилось выполнять авторизацию каждый раз когда он открывает приложение.



**Рисунок 4.13 – Окно авторизации**

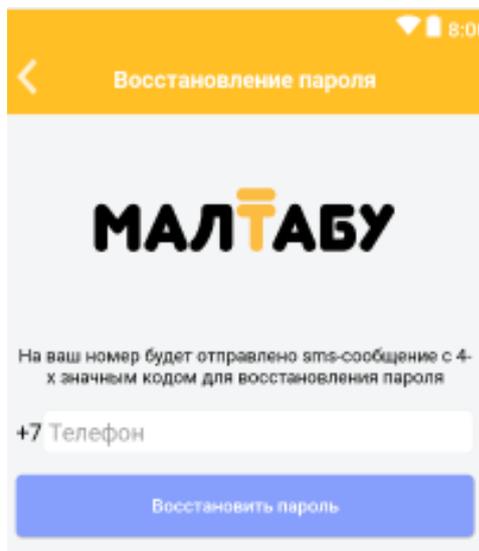
Каждый раз когда происходит авторизация, на сервере генерируется новый токен поэтому нужно его проверять каждый раз когда пользователь заходит в приложение. В случае неподходящего токена вся информация о пользователе, хранящаяся в постоянной памяти мобильного устройства, стирается.

Когда пользователь авторизовался, в главном меню надпись «Личный кабинет» заменяется на имя пользователя, что напоминает ему о том что он авторизован в системе.

Приложение переходит в окно авторизации в нескольких случаях:

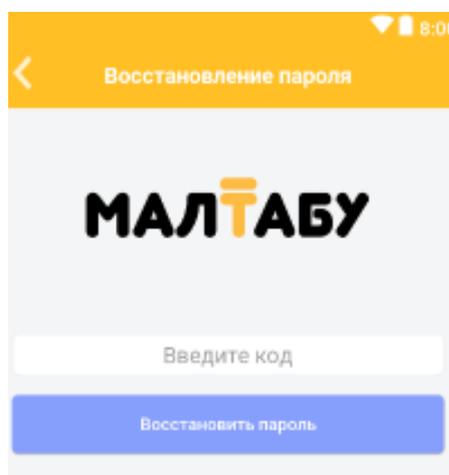
- Когда неавторизованный пользователь пытается продвинуть объявление;
- Когда неавторизованный пользователь пытается оставить комментарий.

Если пользователь забыл пароль от личного кабинета, то он может его восстановить. Для восстановления пароля, в окне авторизации, прожимается кнопка «Забыл пароль».



**Рисунок 4.14 – Окно восстановления пароля**

После чего пользователь вводит свой логин для входа, на который приходит четырехзначный код. При успешном вводе кода восстановления пользователю дается возможность придумать новый пароль.

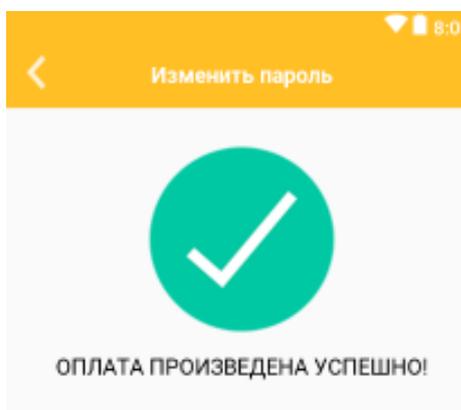


**Рисунок 4.15 – Окно восстановления пароля**

### **4.3.3 Продвижение объявления**

Продвижение объявления в приложении возможно только при наличии у пользователя необходимого количества средств на счету. Эта информация получается при авторизации пользователя.

Реализованы эти функции с помощью http-запросов. В заголовке указан токен, по которому сервер определяет пользователя. В тело запроса указывается номер объявления. При успешном проведении продвижения объявления на экране появляется уведомление об успешно выполненной операции.

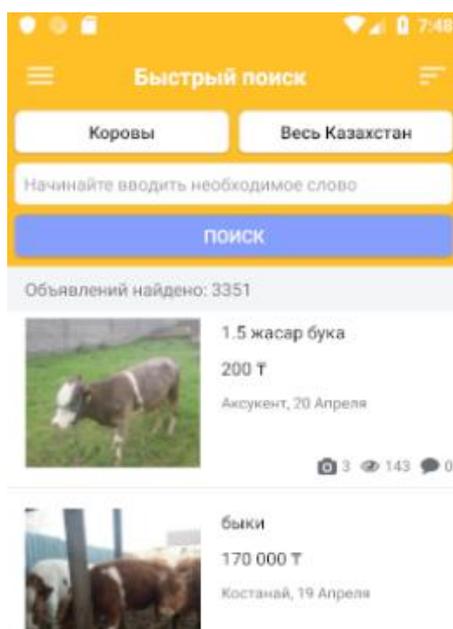


**Рисунок 4.16 – Уведомление об успешном продвижении объявления**

#### **4.3.4 Поиск объявления**

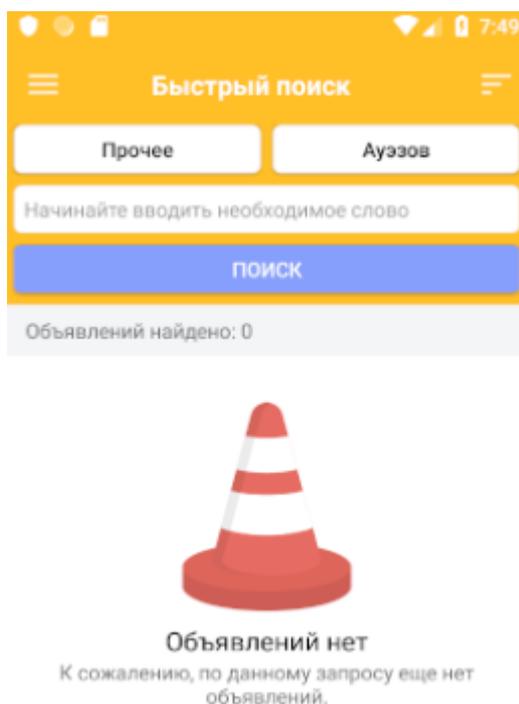
При нажатии иконки поиска в меню приложения фрагмент главного экрана меняется на фрагмент поиска, на котором расположены кнопка выбора региона или города, кнопка выбора категории или каталога и поле ввода текста.

Поиск выполняется по http-запросу, в теле которого указывается необходимые параметры. Успешное завершение запроса возвращало JSON с объявлениями, которые отображаются в нижней части фрагмента экрана.



**Рисунок 4.17 – Фрагмент Поиска**

В случае отсутствия объявлений, по сделанному запросу, на фрагменте будет показано уведомление об отсутствии объявлений.



**Рисунок 4.18 – Уведомление при отсутствии объявлений**

#### **4.3.5 Фильтрация объявлений**

Для более эффективного поиска необходимого товара, пользователь приложения может фильтровать объявления.

Критерии фильтра:

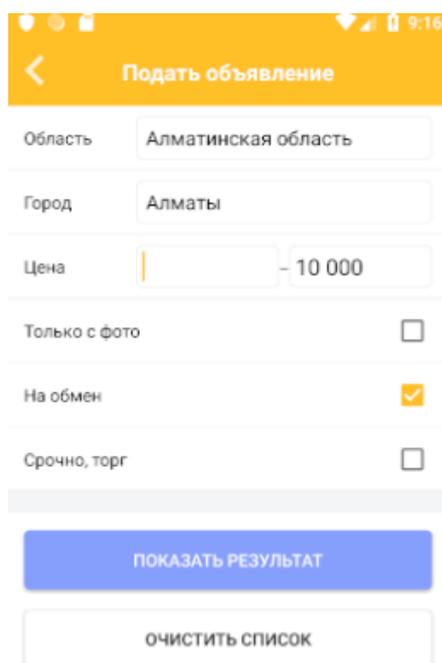
- Область;
- Город;
- Начальная цена;
- Конечная цена;
- Наличие фото;
- Возможность обмена;
- Возможность торга.

Ни один из критериев не является обязательным.

Иконка фильтра находится в правом верхнем углу фрагмента Каталога. После нажатия на иконку переключает главный экран мобильного устройства на активность фильтра. В верхней части активности расположены 2 спиннера. Эти элементы представляют собой выпадающие списки областей и городов Казахстана. Выбрав область определяется список городов.

Под спиннерами располагается диапазон цен, под которым расположено 3 кнопки отметив которые можно указать наличие фото, обмена или торга.

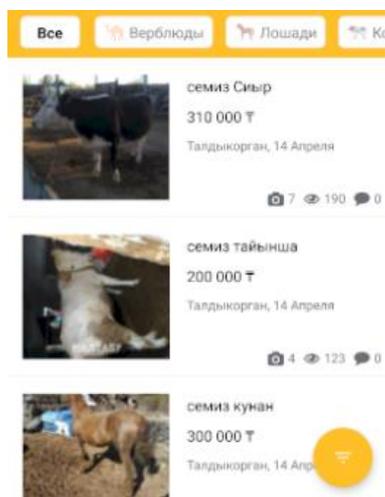
В нижней части экрана расположены 2 кнопки «Показать результат» и «Очистить список».



**Рисунок 4.19 – Окно фильтрации объявлений**

При нажатии на первую кнопку приложение возвращается на фрагмент с каталогом, из которого происходил переход на активность фильтра. При этом создается объект класса `Filter`. Этот `java` класс реализует `Parcelable`, который позволяет передавать объекты в другие активности. В этом фрагменте проверяется наличие переданного фильтра. Если объект класса `Filter` был передан то он применяется при `http`-запросе. После применения фильтра, отображаются отфильтрованные объявления и в нижнем правом углу отображается иконка фильтра, обозначающая применение фильтра.

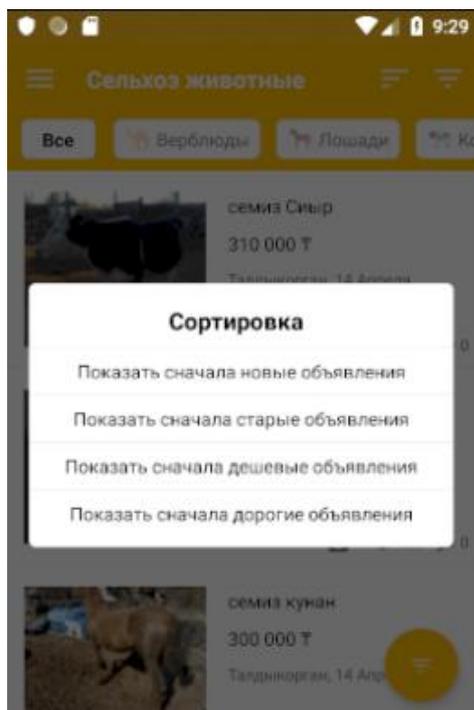
Иконку можно увидеть на Рисунке 4.18



**Рисунок 4.20 – Результат применения фильтра**

Вторая кнопка приводит фильтр в первоначальное состояние и после возвращении на фрагмент каталога, действие фильтра отменяется.

Объявления также модно отсортировать.



**Рисунок 4.21 – Сортировка объявлений**

Варианты сортировки объявлений:

- Сначала новые объявления;
- Сначала старые объявления;
- Сначала дешевые объявления;
- Сначала дорогие объявления.

По умолчанию отображаются сначала новые объявления. Пользователь может выбрать другой вид сортировки. После выбора фрагмент каталога обновляется в соответствии с указанной сортировкой.

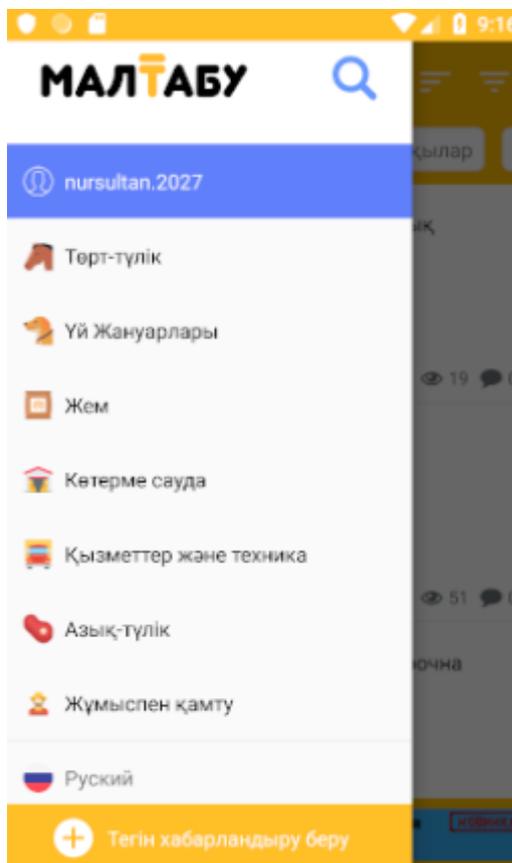
#### **4.3.6 Смена языка интерфейса**

Одной из важнейших функций данного приложения является смена языка интерфейса. Каждый строковый ресурс, в данном приложении, хранится в папке values/strings. Существует по 2 варианта этих ресурсов – казахский и русский. Используемый язык определяется согласно настройкам локализации смартфона.

Предпочитаемый язык сохраняется в Sharedpreferences. Sharedpreferences – это инструмент android разработки для сохранения данных в виде чисел, строк и символов и обращения к нему с помощью текстового ключа.

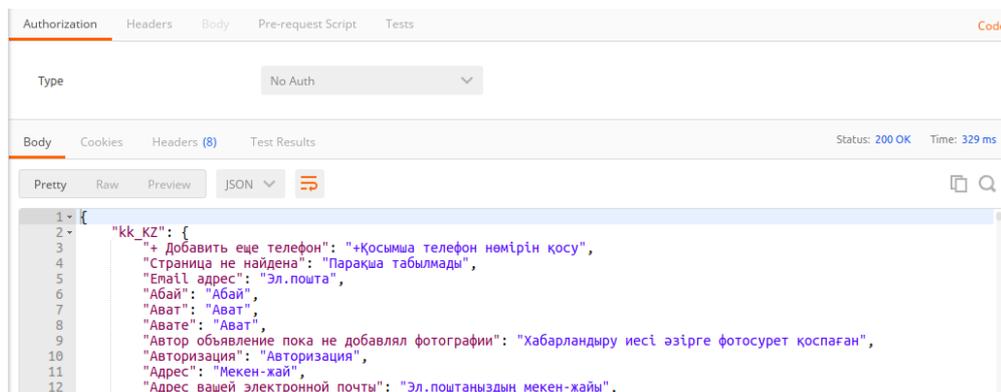
Для работы с этим инструментом была выбрана библиотека PaperDb. При смене активности элементы интерфейса, содержащие текст, отрисовываются в соответствии с текущим языком.

Сменить язык интерфейса можно в меню приложения. В нижней части расположена кнопка с указанием языка на который можно переключиться.



**Рисунок 4.22 – Меню на казахском**

При первом открытии приложения выполняется http-запрос возвращающий JSON с переводом ключевых слов, используемых в проекте. Эти данные сохраняются в постоянной памяти смартфона и используются при переводе названий категорий, каталогов, городов и областей.



**Рисунок 4.23 – Результат запроса в окне программы Postman**

## ЗАКЛЮЧЕНИЕ

В результате выполнения дипломного проекта было создано мобильное приложение для агропромышленной площадки на платформе андроид. Целевой аудиторией данного приложения являются частные лица, желающие продать или купить сельскохозяйственных животных и продукты фермерства, а также крупные предприятия сельскохозяйственного производства. Основная задача данной работы – создание удобного приложения, с помощью которого пользователи могут просматривать, добавлять и комментировать объявления.

Исследование предметной области показало острую необходимость создания приложения для мобильных устройств. Большинство потенциальных пользователей сервиса не имеют персонального компьютера или ноутбука, а мобильная версия сайта – не самое оптимальное решение т.к. постоянно заходить на сайт через браузер мобильного устройства создает слишком много переходов. Мобильные приложения, как правило, помогают увеличить количество пользователей и собирают, в основном, целевую аудиторию.

Разработанное приложение является законченным продуктом и размещено в интернет магазине Google Play Market. В будущем планируется расширить возможности приложения. Главным направлением станет монетизация. Сегодня существует множество способов заработать на мобильном приложении. Планируются рекламные интеграции с использованием таких технологий как Google Admob, Facebook SDK и Яндекс Директ. Эти инструменты используют нейронные сети для определения предпочтений пользователей и, в результате, интегрируют необходимую рекламу. Также будут изменены правила продвижения объявлений. Они будут ориентированы на более жесткую конкуренцию среди подающих объявления.

Новые функции, которые будут добавлены в ближайшее время:

- создание раздела «Избранные объявления»;
- добавление чата для пользователей;
- добавление push уведомлений;
- регистрация в системе через сервисы Facebook и Google.

Программный код проекта писался с возможностью доработки и хранится в удаленных репозиториях для безопасности данных. В процессе разработки активно использовалась система контроля версий – git.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Введение в платформу Android // Электронная версия на сайте <https://metanit.com/java/android/1.1.php>.

2 Учебник по Android // Электронная версия на сайте <https://startandroid.ru/ru/uroki.html>.

3 Гриффитс Д., Гриффитс Д. Head first программирование для Android // Электронная версия на сайте <http://www.intuit.ru/studies/courses/3439/681/lecture/14019>.

4 Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android программирование для профессионалов // Электронная версия на сайте <http://webnotes.by/docs/sql/259>.

5 Разработка под Android. Немного о быстрой работе со списками // Электронная версия на сайте <https://habr.com/ru/post/424129>.

6 Шилдт Г. Java 8. Полное руководство 9-е издание // Электронная версия на сайте <http://webnotes.by/docs/sql/270>.

7 Введение в Java // Электронная версия на сайте <https://metanit.com/java/tutorial/1.1.php>.

## **Приложение А**

### **Техническое задание**

#### **А.1.1 Техническое задание на разработку мобильного приложения**

Техническое задание распространяется на разработку мобильной онлайн доски сельскохозяйственных объявлений, предназначенной для просмотра, подачи и продвижения объявлений. Потенциальными пользователями являются частные лица, покупающие или продающие сельскохозяйственных животных и продукты фермерства. Мобильное приложение избавит продавцов и покупателей от посредников и физического пребывания на рынке, а также позволит увеличить количество пользователей сервисом.

##### **А.1.1.1 Основание для разработки**

Система разрабатывается на основании дипломной работы.

##### **А.1.1.2 Назначение**

Приложение предназначено для просмотра, поиска, подачи и продвижения сельскохозяйственных объявлений со всего Казахстана.

##### **А.1.1.3 Требования к функциональным характеристикам**

Список функций, которые приложение должно обеспечивать:

- отображение списка объявлений;
- отображение подробной информации об объявлении (с возможностью увеличения фотографий);
- регистрации в системе;
- авторизации в системе;
- хранение данных авторизованного пользователя;
- поиск объявлений (по тексту, городу или категории);
- фильтрация объявлений;
- сортировка объявлений (по времени или стоимости);
- подача объявления;
- восстановление пароля;

## **Продолжение приложения А**

- делать фотографии, используя камеру мобильного устройства;
- получение данных из внутреннего хранилища.

### **А.1.1.4 Требования к надежности**

Предусмотреть валидацию вводимых данных. Предусмотреть сбои и другие исключительные ситуации при работе пользователя с приложением. Обеспечить целостность данных.

### **А.1.1.5 Требования к составу и параметрам технических средств**

Минимальные системные требования: 48 Мб свободной памяти; объем ОЗУ – 256 Мб и выше, Поддержка 4G – рекомендуется.

### **А.1.1.6 Требования к информационной и программной совместимости**

Система предназначена для работы на мобильных устройствах под управлением ОС Android, начиная с 5.0 версии.

### **А.1.1.7 Требования к программной документации**

Тексты программ должны быть комментированы.

## Приложение Б

### Текст программы

```
PACKAGE KZ.MALTABU.APP.MALTABUKZ.ACTIVITIES;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import kz.maltabu.app.maltabukz.R;
import kz.maltabu.app.maltabukz.helpers.FileHelper;
import kz.maltabu.app.maltabukz.helpers.Maltabu;

import org.json.JSONException;
import org.json.JSONObject;

import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

public class MainActivity extends AppCompatActivity{
    private FileHelper fileHelper;
    private Dialog epicDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.asd);
        fileHelper = new FileHelper(this);
        epicDialog = new Dialog(this);
        if (isConnected())
        {
            GetVersion();
        }
    }
}
```

## Продолжение приложения Б

```
} else {
    setContentView(R.layout.no_internet_connection);
    TextView refresh = (TextView) findViewById(R.id.textView35);
    refresh.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(isConnected()){
                noRecord();
            }
        }
    });
}
}

public void GetCategories() {
    final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
        .url("http://maltabu.kz/v1/api/clients/data")
        .get()
        .addHeader("isAuthorized", "false")
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response2 = client.newCall(request2).execute();
                if (!response2.isSuccessful()) {
                    return null;
                }
                return response2.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
    }

    @Override
    protected void onPostExecute(String s1) {
        super.onPostExecute(s1);
        if (s1 != null) {
            fileHelper.writeDataFile(s1);
        }
    }
}
```

## Продолжение приложения Б

```
};
    asyncTask1.execute();
}
public boolean isConnected() {
    boolean connected = false;
    try {
        ConnectivityManager cm =
(ConnectivityManager)getContext().getSystemService(Context.CONN
ECTIVITY_SERVICE);
        NetworkInfo nInfo = cm.getActiveNetworkInfo();
        connected = nInfo != null && nInfo.isAvailable() &&
nInfo.isConnected();
        return connected;
    } catch (Exception e) {
    }
    return connected;
}
public void GetDictionary() {
    final OkHttpClient client = new OkHttpClient();
    final Request request = new Request.Builder()
        .url("http://maltabu.kz/dist/translations/kk_KZ.json")
        .get()
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response = client.newCall(request).execute();
                if (!response.isSuccessful()) {
                    return null;
                }
                return response.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        if (s != null) {
```

## Продолжение приложения Б

```
try {
    JSONObject dict = new
JSONObject(s).getJSONObject("kk_KZ");
    fileHelper.writeDictionary(dict.toString());
    if(!fileHelper.readToken().isEmpty()){
        Maltabu.token=fileHelper.readToken();
        Maltabu.isAuth="true";
    }
    startActivity(new Intent(MainActivity.this,
MainActivity2.class));
    finish();
} catch (JSONException e) {
    e.printStackTrace();
}
};
asyncTask1.execute();
}
```

```
public void GetVersion()
{
    final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
        .url("http://maltabu.kz/v1/api/clients/data/apps")
        .get()
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response2 = client.newCall(request2).execute();
                if (!response2.isSuccessful()) {
                    noRecord();
                    return null;
                }
                return response2.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
    }
```

## Продолжение приложения Б

```
}

@Override
protected void onPostExecute(String s1) {
    super.onPostExecute(s1);
    if (s1 != null) {
        try {
            JSONObject jsonObject = new JSONObject(s1);
            if (jsonObject.has("android_version")) {
                if
(Maltabu.version.equals(jsonObject.getString("android_version"))) {
                    noRecord();
                }
            } else{
                if(jsonObject.has("android_link"))
                {
                    String url = jsonObject.getString("android_link");
                    sDialog(url);
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

};
asyncTask1.execute();
}

protected void sDialog(final String uri) {
    epicDialog setContentView(R.layout.auth_logout);
    final Button asd = (Button)
epicDialog.findViewById(R.id.buttonCancel);
    final Button asd3 = (Button)
epicDialog.findViewById(R.id.buttonOkok);
    asd.setTextColor(Color.parseColor("#69aef3"));
    final TextView asd2 = (TextView)
epicDialog.findViewById(R.id.changeText);
    asd2.setText("Обновить приложение");
    asd.setText("Обновить");
    asd3.setText("Потом");
}
```

## Продолжение приложения Б

```
asd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent browserIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse(uri));
        startActivity(browserIntent);
    }
});
asd3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (epicDialog != null && epicDialog.isShowing()) {
            epicDialog.dismiss();
        }
        noRecord();
    }
});
epicDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
epicDialog.show();
}

public void noRecord(){
    GetCategories();
    if(fileHelper.readDictionary().isEmpty())
        GetDictionary();
    else {
        if(!fileHelper.readToken().isEmpty()) {
            Maltabu.token = fileHelper.readToken();
            Maltabu.isAuth = "true";
        }
        startActivity(new Intent(MainActivity.this, MainActivity2.class));
        finish();
    }
}
}

package kz.maltabu.app.maltabukz.activities;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.content.res.Resources;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
```

## Продолжение приложения Б

```
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.constraint.ConstraintLayout;
import android.support.design.widget.AppBarLayout;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import kz.maltabu.app.maltabukz.R;
import kz.maltabu.app.maltabukz.fragments.SearchFragment;
import kz.maltabu.app.maltabukz.helpers.FileHelper;
import kz.maltabu.app.maltabukz.helpers.LocaleHelper;
import kz.maltabu.app.maltabukz.helpers.Maltabu;
import kz.maltabu.app.maltabukz.fragments.CatalogFragment;
import kz.maltabu.app.maltabukz.fragments.HotFragment;
import org.json.JSONException;
import org.json.JSONObject;
import io.paperdb.Paper;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
public class MainActivity2 extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {
    private
        TextView
        menu1,
menu2,menu3,menu4,menu6,menu5,menu7,menu8, menu82, cab, lange, hottitle;
    private ConstraintLayout c11, m1, m2, m3, m4, m5, m6, m7,cab1, cLnag,
search;
    private ImageView filter, flag,sort, menuLogo;
    private JSONObject object;
    private FileHelper fileHelper;
    private DrawerLayout drawer;
    private Dialog epicDialog, sortDialog;
    private Intent filterIntent;
```

## Продолжение приложения Б

```
private static long back_pressed;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    fileHelper = new FileHelper(this);
    SetActivityView();
    epicDialog = new Dialog(this);
    sortDialog = new Dialog(this);
    openCurrentFragment(Maltabu.fragmentNumb);
}
private void SetActivityView(){
    setContentView(R.layout.activity_main);
    Resources res = this.getResources();
    filter = (ImageView) findViewById(R.id.filter);
    sort = (ImageView) findViewById(R.id.sort);
    filterIntent = new Intent(this, FilterActivity.class);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    AppBarLayout appBar = (AppBarLayout)
findViewById(R.id.appBarLayout2);
    setSupportActionBar(toolbar);
    appBar.setOutlineProvider(null);
    getSupportActionBar().setDisplayHomeAsUpEnabled(false);
    Paper.init(this);
    String lgg = Paper.book().read("language");
    if(lgg==null){
        Paper.book().write("language", "kk");
    } else {
        Maltabu.lang = lgg;
    }
    drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
    drawer.addDrawerListener(toggle);
    toggle.syncState();
    getSupportActionBar().setHomeButtonEnabled(true);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setHomeAsUpIndicator(R.drawable.ic_action_toggle);
    NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);
    View view = navigationView.getHeaderView(0);
```

## Продолжение приложения Б

```
cab = (TextView) view.findViewById(R.id.cabinet);
menu1 = (TextView) view.findViewById(R.id.menu1);
menu2 = (TextView) view.findViewById(R.id.menu2);
menu3 = (TextView) view.findViewById(R.id.menu3);
menu4 = (TextView) view.findViewById(R.id.menu4);
menu5 = (TextView) view.findViewById(R.id.menu5);
menu6 = (TextView) view.findViewById(R.id.menu6);
menu7 = (TextView) view.findViewById(R.id.menu7);
menuLogo = (ImageView) view.findViewById(R.id.imageView37);
hottitle = (TextView) findViewById(R.id.hottitle);
lange = (TextView) view.findViewById(R.id.langText);
cLnag = (ConstraintLayout)
view.findViewById(R.id.constraintLayout29);
flag = (ImageView) view.findViewById(R.id.imageView30);
cab1 = (ConstraintLayout) view.findViewById(R.id.constraintLayout);
m1 = (ConstraintLayout) view.findViewById(R.id.constraintLayout2);
m2 = (ConstraintLayout) view.findViewById(R.id.constraintLayout3);
m3 = (ConstraintLayout) view.findViewById(R.id.constraintLayout4);
m4 = (ConstraintLayout) view.findViewById(R.id.constraintLayout5);
m5 = (ConstraintLayout) view.findViewById(R.id.constraintLayout6);
m6 = (ConstraintLayout) view.findViewById(R.id.constraintLayout9);
m7 = (ConstraintLayout)
view.findViewById(R.id.constraintLayout10);
search = (ConstraintLayout)
view.findViewById(R.id.constraintLayout228);
menu8 = (TextView) findViewById(R.id.menu8);
menu82 = (TextView) findViewById(R.id.menu82);
cl1 = (ConstraintLayout) findViewById(R.id.constraintLayout7);
initListeners();
getWindow().setSoftInputMode(
WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN
);
try {
    object = new JSONObject(fileHelper.readUserFile());
} catch (JSONException e) {
    e.printStackTrace();
}
updateView((String) Paper.book().read("language"));
}
private void initListeners() {
    cLnag.setOnClickListener(new View.OnClickListener() {
        @Override
```

## Продолжение приложения Б

```
public void onClick(View v) {
    if(Maltabu.lang.equals("kk")){
        Paper.book().write("language", "ru");
        Maltabu.lang = "ru";
        flag.setImageResource(R.drawable.kz);
        updateView((String)Paper.book().read("language"));
        opentCurrentFragment(Maltabu.fragmentNumb);
    } else {
        Paper.book().write("language", "kk");
        Maltabu.lang = "kk";
        flag.setImageResource(R.drawable.ru);
        updateView((String)Paper.book().read("language"));
        opentCurrentFragment(Maltabu.fragmentNumb);
    }
}
});
menuLogo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        fragmentMain();
        Maltabu.selectedFragment = 0;
    }
});
cab1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            if(Maltabu.isAuth.equals("false")) {
                startActivity(new Intent(MainActivity2.this,
AuthAvtivity.class));
                finish();
            } else {
                sDialog();
                getPosting();
            }
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
sort.setOnClickListener(new View.OnClickListener() {
    @Override
```

## Продолжение приложения Б

```
public void onClick(View v) {
    soDialog();
}
});
search.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            fragmentSearch();
            Maltabu.selectedFragment = 0;
            sort.setVisibility(View.VISIBLE);
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
filter.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            sDialog();
            SecThread secThread = new SecThread();
            secThread.start();
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
m1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            try {
                fragment1();
                Maltabu.selectedFragment = 0;
                sort.setVisibility(View.VISIBLE);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        } else {
            startActivity(new Intent(MainActivity2.this,
```

## Продолжение приложения Б

```
NoConnection.class));
    }
}
});
m2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            try {
                fragment2();
                Maltabu.selectedFragment = 0;
                sort.setVisibility(View.VISIBLE);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
m3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            try {
                fragment3();
                Maltabu.selectedFragment = 0;
                sort.setVisibility(View.VISIBLE);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
m4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            try {
```

## Продолжение приложения Б

```
        fragment4();
        Maltabu.selectedFragment = 0;
        sort.setVisibility(View.VISIBLE);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    } else {
        startActivity(new Intent(MainActivity2.this,
NoConnection.class));
    }
    });
m5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            try {
                fragment5();
                Maltabu.selectedFragment = 0;
                sort.setVisibility(View.VISIBLE);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
m6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            fragment6();
            Maltabu.selectedFragment = 0;
            sort.setVisibility(View.VISIBLE);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
});
m7.setOnClickListener(new View.OnClickListener() {
    @Override
```

## Продолжение приложения Б

```
public void onClick(View v) {
    if(isConnected()) {
        try {
            fragment7();
            Maltabu.selectedFragment = 0;
            sort.setVisibility(View.VISIBLE);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    } else {
        startActivity(new Intent(MainActivity2.this,
NoConnection.class));
    }
});
c11.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(isConnected()) {
            startActivity(new Intent(MainActivity2.this,
AddPostActivity.class));
            finish();
        } else {
            startActivity(new Intent(MainActivity2.this,
NoConnection.class));
        }
    }
});
}
private void updateView(String lang) {
    Context context = LocaleHelper.setLocale(this, lang);
    Maltabu.lang = lang;
    if (lang.equals("ru"))
        flag.setImageResource(R.drawable.kz);
    else
        flag.setImageResource(R.drawable.ru);
    Resources resources = context.getResources();
    setTitle("");
    if(Maltabu.isAuth.equals("false")) {
        cab.setText(resources.getString(R.string.Cabinet));
    } else {
        try {
            cab.setText(object.getString("name"));
        }
    }
}
```

## Продолжение приложения Б

```
} catch (JSONException e) {
    e.printStackTrace();
}
}
menu1.setText(resources.getString(R.string.menu1));
menu2.setText(resources.getString(R.string.menu2));
menu3.setText(resources.getString(R.string.menu3));
menu4.setText(resources.getString(R.string.menu4));
menu5.setText(resources.getString(R.string.menu5));
menu6.setText(resources.getString(R.string.menu6));
menu7.setText(resources.getString(R.string.menu7));
menu8.setText(resources.getString(R.string.menu8));
menu82.setText(resources.getString(R.string.menu82));
lange.setText(resources.getString(R.string.other_lang));
}
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        if (Maltabu.fragmentNumb==0) {
            if (back_pressed + 2000 > System.currentTimeMillis()) {
                super.onBackPressed();
            } else {
                if (Maltabu.lang.equals("ru"))
                    Toast.makeText(getApplicationContext(), "Нажмите еще раз для
выхода", Toast.LENGTH_SHORT).show();
                else
                    Toast.makeText(getApplicationContext(), "Шығу үшін тағы
басыңыз", Toast.LENGTH_SHORT).show();
            }
            back_pressed = System.currentTimeMillis();
        } else {
            Maltabu.filterModel = null;
            Maltabu.s1= null;
            Maltabu.s3= null;
            Maltabu.s5= null;
            Maltabu.s6= null;
            Maltabu.s4= null;
            Maltabu.s2= null;
            fragmentMain();
        }
    }
}
```

## Продолжение приложения Б

```
    }
    }
}
@Override
protected void attachBaseContext(Context newBase) {
    super.attachBaseContext(LocaleHelper.onAttach(newBase,"ru"));
}
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    int id = item.getItemId();
//    displaySelectedScreen(id);
    return true;
}
private void fragmentMain(){
    filter.setVisibility(View.GONE);
    sort.setVisibility(View.GONE);
    Context context = LocaleHelper.setLocale(this, Maltabu.lang);
    hottitle.setText(context.getResources().getString(R.string.hotTitle));
    android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    HotFragment fragment = new HotFragment();
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 0;
    Maltabu.byTime = true;
    Maltabu.increment = true;
    Maltabu.s1= null;
    Maltabu.s3= null;
    Maltabu.s5= null;
    Maltabu.s6= null;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.text = null;
}
private void fragment1() throws JSONException {
    filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
```

## Продолжение приложения Б

```
fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
fileHelper.getCategoriesFromFile().get(1));
    String [] asd = {"5ab672c9559d5e049c25a62c",
        "5ab672c9559d5e049c25a62d", "5ab672c9559d5e049c25a62e",
        "5ab672c9559d5e049c25a62f", "5ab672c9559d5e049c25a630",
        "5ab672c9559d5e049c25a631", "5ab672c9559d5e049c25a632"};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5ab672c9559d5e049c25a62b");
    fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 1;
    Maltabu.s1= null;
    Maltabu.s3= null;
    Maltabu.s5= null;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.text = null;
    Maltabu.s6= null;
}
private void fragment2() throws JSONException {
android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
fileHelper.getCategoriesFromFile().get(4));
    String [] asd =
{"5ab672c9559d5e049c25a645", "5ab672c9559d5e049c25a646",
        "5ab672c9559d5e049c25a647", "5ab672c9559d5e049c25a648",
        "5ab672c9559d5e049c25a649", "5ab672c9559d5e049c25a64a"};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5ab672c9559d5e049c25a644");
    fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
```

## Продолжение приложения Б

```
drawer.closeDrawer(GravityCompat.START);
Maltabu.fragmentNumb = 2;
Maltabu.s1= null;
Maltabu.text = null;
Maltabu.s4= null;
Maltabu.s2= null;
Maltabu.s3= null;
Maltabu.s5= null;
Maltabu.s6= null;
}
private void fragment3() throws JSONException {
    android.support.v4.app.FragmentManager fragmentManager =
    getSupportFragmentManager();
    filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentTransaction fragmentTransaction =
    fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
    fileHelper.getCategoriesFromFile().get(2));
    String [] asd =
    {"5ab672c9559d5e049c25a634","5ab672c9559d5e049c25a635",
    "5ab672c9559d5e049c25a636","5ab672c9559d5e049c25a637","5ab672c9559d5e0
    49c25a638"};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5ab672c9559d5e049c25a633");
    fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 3;
    Maltabu.s1= null;
    Maltabu.text = null;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.s3= null;
    Maltabu.s5= null;
    Maltabu.s6= null;
}
private void fragment4() throws JSONException {
    android.support.v4.app.FragmentManager fragmentManager =
    getSupportFragmentManager();
```

## Продолжение приложения Б

```
filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
fileHelper.getCategoriesFromFile().get(0));
    String [] asd = {"5ab672c9559d5e049c25a63a",
"5ab672c9559d5e049c25a63b"};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5ab672c9559d5e049c25a639");
    fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 4;
    Maltabu.text = null;
    Maltabu.s1= null;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.s3= null;
    Maltabu.s5= null;
    Maltabu.s6= null;
}
private void fragment5() throws JSONException {
    android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
fileHelper.getCategoriesFromFile().get(3));
    String [] asd =
{"5ab672c9559d5e049c25a63f","5ab672c9559d5e049c25a640",
"5ab672c9559d5e049c25a641","5ab672c9559d5e049c25a642",
"5ab672c9559d5e049c25a643","5b0bffe2530c6256285a1933","5b0bffe2530c6256
285a19b1"
};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5ab672c9559d5e049c25a63e");
```

## Продолжение приложения Б

```
fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    Maltabu.text = null;
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 5;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.s1= null;
    Maltabu.s3= null;
    Maltabu.s5= null;
    Maltabu.s6= null;
}
private void fragment6() throws JSONException {
    filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
fileHelper.getCategoriesFromFile().get(5));
    String [] asd =
{"5ab672c9559d5e049c25a64c","5ab672c9559d5e049c25a64d",
    "5ab672c9559d5e049c25a64e","5ab672c9559d5e049c25a64f",
    "5ab672c9559d5e049c25a650"};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5ab672c9559d5e049c25a64b");
    fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 6;
    Maltabu.s1= null;
    Maltabu.text = null;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.s3= null;
    Maltabu.s5= null;
    Maltabu.s6= null;
}
private void fragment7() throws JSONException {
```

## Продолжение приложения Б

```
filter.setVisibility(View.VISIBLE);
    android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    CatalogFragment fragment = new CatalogFragment();
    Bundle bundle1 = new Bundle();
    bundle1.putParcelable("categ",
fileHelper.getCategoriesFromFile().get(6));
    String [] asd =
{"5afeb741d151e32d5cc245c4","5afeb741d151e32d5cc245c5"};
    bundle1.putStringArray("str", asd);
    bundle1.putString("categId", "5afeb741d151e32d5cc245c3");
    fragment.setArguments(bundle1);
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 7;
    Maltabu.s1= null;
    Maltabu.text = null;
    Maltabu.s3= null;
    Maltabu.s4= null;
    Maltabu.s2= null;
    Maltabu.s5= null;
    Maltabu.s6= null;
}
private void fragmentSearch() {
    filter.setVisibility(View.GONE);
    Context context = LocaleHelper.setLocale(this, Maltabu.lang);
    hottitle.setText(context.getResources().getString(R.string.SearchText));
    android.support.v4.app.FragmentManager fragmentManager =
getSupportFragmentManager();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    SearchFragment fragment = new SearchFragment();
    fragmentTransaction.replace(R.id.main, fragment);
    fragmentTransaction.commit();
    drawer.closeDrawer(GravityCompat.START);
    Maltabu.fragmentNumb = 8;
    Maltabu.byTime = true;
    Maltabu.increment = true;
}
public void getPosting(){
```

## Продолжение приложения Б

```
final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
        .url("http://maltabu.kz/v1/api/clients/cabinet/posting")
        .get()
        .addHeader("isAuthorized", Maltabu.isAuth)
        .addHeader("token", Maltabu.token)
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response2 = client.newCall(request2).execute();
                if (!response2.isSuccessful()) {
                    Maltabu.token = null;
                    Maltabu.isAuth = "false";
                    fileHelper.writeUserFile("");
                    fileHelper.writeToken("");
                    startActivity(new Intent(MainActivity2.this,
AuthActivity.class));
                    finish();
                    return null;
                }
                return response2.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
        @Override
        protected void onPostExecute(String s1) {
            super.onPostExecute(s1);
            if (s1 != null) {
                fileHelper.writePostingFile(s1);
                getUser();
            }
        }
    };
    asyncTask1.execute();
}
public void getUser(){
    final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
```

## Продолжение приложения Б

```
.url("http://maltabu.kz/v1/api/login/clients")
.get()
.addHeader("isAuthorized", Maltabu.isAuth)
.addHeader("token", Maltabu.token)
.build();
AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
    @Override
    protected String doInBackground(Void... params) {
        try {
            Response response2 = client.newCall(request2).execute();
            if (!response2.isSuccessful()) {
                return null;
            }
            return response2.body().string();
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
    @Override
    protected void onPostExecute(String s1) {
        super.onPostExecute(s1);
        if (s1 != null) {
            fileHelper.writeUserFile(s1);
            if (epicDialog != null && epicDialog.isShowing()) {
                epicDialog.dismiss();
            }
            startActivity(new Intent(MainActivity2.this,
CabinetActivity.class));
            finish();
        }
    }
};
asyncTask1.execute();
}
protected void sDialog() {
    epicDialog.setContentView(R.layout.progress_dialog);
    epicDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
    epicDialog.show();
}
protected void soDialog() {
```

## Продолжение приложения Б

```
sortDialog.setContentView(R.layout.sort_dialog);
Resources resources = LocaleHelper.setLocale(this,
Maltabu.lang).getResources();
final TextView txt1 = (TextView)
sortDialog.findViewById(R.id.textView57);
final TextView txt2 = (TextView)
sortDialog.findViewById(R.id.textView59);
final TextView txt3 = (TextView)
sortDialog.findViewById(R.id.textView60);
final TextView txt4 = (TextView)
sortDialog.findViewById(R.id.textView61);
txt1.setText(resources.getString(R.string.sort1));
txt2.setText(resources.getString(R.string.sort2));
txt3.setText(resources.getString(R.string.sort3));
txt4.setText(resources.getString(R.string.sort4));
txt1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Maltabu.byTime = true;
        Maltabu.increment = true;
        opentCurrentFragment(Maltabu.fragmentNumb);
        if (sortDialog != null && sortDialog.isShowing()) {
            sortDialog.dismiss();
        }
    }
});
txt2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Maltabu.byTime = true;
        Maltabu.increment = false;
        opentCurrentFragment(Maltabu.fragmentNumb);
        if (sortDialog != null && sortDialog.isShowing()) {
            sortDialog.dismiss();
        }
    }
});
txt3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Maltabu.byTime = false;
        Maltabu.increment = false;
        opentCurrentFragment(Maltabu.fragmentNumb);
```

## Продолжение приложения Б

```
if (sortDialog != null && sortDialog.isShowing()) {
    sortDialog.dismiss();
}
});
txt4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Maltabu.byTime = false;
        Maltabu.increment = true;
        opentCurrentFragment(Maltabu.fragmentNumb);
        if (sortDialog != null && sortDialog.isShowing()) {
            sortDialog.dismiss();
        }
    }
});
sortDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
sortDialog.show();
}
public boolean isConnected() {
    boolean connected = false;
    try {
        ConnectivityManager cm =
(ConnectivityManager)getApplicationContext().getSystemService(Context.CONN
ECTIVITY_SERVICE);
        NetworkInfo nInfo = cm.getActiveNetworkInfo();
        connected = nInfo != null && nInfo.isAvailable() &&
nInfo.isConnected();
        return connected;
    } catch (Exception e) {
    }
    return connected;
}
public void opentCurrentFragment(int numb){
    switch (numb) {
        case 0:
            fragmentMain();
            break;
        case 1:
            try {
                fragment1();
            } catch (JSONException e) {
```

## Продолжение приложения Б

```
        e.printStackTrace();
    }
    break;
case 2:
    try {
        fragment2();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    break;
case 3:
    try {
        fragment3();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    break;
case 4:
    try {
        fragment4();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    break;
case 5:
    try {
        fragment5();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    break;
case 6:
    try {
        fragment6();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    break;
case 7:
    try {
        fragment7();
    } catch (JSONException e) {
        e.printStackTrace();
    }
```

## Продолжение приложения Б

```
}
    break;
case 8:
    fragmentSearch();
    break;
}
}
public class SecThread extends Thread{
    SecThread(){ };
    @Override
    public void run() {
        startActivity(filterIntent);
        finish();
    }
}
}
package kz.maltabu.app.maltabukz.activities;
import android.Manifest;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.res.Resources;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.provider.Settings;
import android.support.annotation.NonNull;
import android.support.constraint.ConstraintLayout;
import android.support.design.widget.Snackbar;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
```

## Продолжение приложения Б

```
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.google.gson.Gson;
import com.r0adkll.slidr.Slidr;
import com.r0adkll.slidr.model.SlidrConfig;
import com.r0adkll.slidr.model.SlidrListener;
import com.r0adkll.slidr.model.SlidrPosition;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import kz.maltabu.app.maltabukz.R;
import kz.maltabu.app.maltabukz.helpers.FileHelper;
import kz.maltabu.app.maltabukz.helpers.LocaleHelper;
import kz.maltabu.app.maltabukz.helpers.Maltabu;
import kz.maltabu.app.maltabukz.fragments.ImageFragment;
import kz.maltabu.app.maltabukz.models.Comment;
import kz.maltabu.app.maltabukz.models.Image;
import kz.maltabu.app.maltabukz.models.Post;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
public class ShowDetails extends AppCompatActivity {
    private TextView callPhoneText, title, content, price, phone, location, date,
photos, commen;
    private ImageView img;
    private ConstraintLayout cs1, hot, top, comments;
    private Post post;
    private String currentNumber;
    private FileHelper fileHelper;
    private LinearLayout layout;
    private int PAGE_COUNT;
    private ViewPager pager;
    private PagerAdapter pagerAdapter;
    private Dialog epicDialog;
    private int selectedImg, phonecount=1;;
    private Intent imagesIntent;
    private static final int PERMISSION_REQUEST_CODE = 123;
    private GestureDetector tapGestureDetector;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

## Продолжение приложения Б

```
super.onCreate(savedInstanceState);
fileHelper = new FileHelper(this);
epicDialog = new Dialog(this);
imagesIntent = new Intent(this, ShowDetailsImages.class);
closeLeftSwipe();
sDialog();
getPost(getIntent().getStringExtra("postNumb"));
}
private void closeLeftSwipe() {
    SlidrConfig config = new SlidrConfig.Builder()
        .position(SlidrPosition.LEFT)
        .sensitivity(1f)
        .scrimColor(Color.BLACK)
        .scrimStartAlpha(0.8f)
        .scrimEndAlpha(0f)
        .velocityThreshold(2400)
        .distanceThreshold(0.5f)
        .edge(true|false)
        .edgeSize(0.18f) // The % of the screen that counts as the edge,
default 18%
        .listener(new SlidrListener(){
            @Override
            public void onSlideStateChanged(int state) {
            }
            @Override
            public void onSlideChange(float percent) {
            }
            @Override
            public void onSlideOpened() {
            }
            @Override
            public void onSlideClosed() {
            }
        })
        .build();
    Slidr.attach(this, config);
}
protected void sDialog() {
    epicDialog.setContentView(R.layout.progress_dialog);
    epicDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
    epicDialog.show();
}
```

## Продолжение приложения Б

```
private void getPost(String numb) {
    final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
        .url("http://maltabu.kz/v1/api/clients/posts/"+numb)
        .get()
        .addHeader("isAuthorized", "false")
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response2 = client.newCall(request2).execute();
                if (!response2.isSuccessful()) {
                    return null;
                }
                return response2.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
        @Override
        protected void onPostExecute(String s1) {
            super.onPostExecute(s1);
            Gson googleJson = new Gson();
            if (s1 != null) {
                try {
                    JSONObject postObject = new JSONObject(s1);
                    JSONArray arr = postObject.getJSONArray("images");
                    ArrayList<Image> imagesArrayList = new ArrayList<>();
                    ArrayList<Image> imgObjList =
googleJson.fromJson(String.valueOf(arr), ArrayList.class);
                    Image image = null;
                    Comment com = null;
                    for (int j = 0; j < imgObjList.size(); j++) {
                        JSONObject imgJson = arr.getJSONObject(j);
                        image = new Image(
                            imgJson.getString("extra_small"),
                            imgJson.getString("small"),
                            imgJson.getString("medium"),
                            imgJson.getString("big"));
                        imagesArrayList.add(image);
                    }
                }
            }
        }
    };
}
```

## Продолжение приложения Б

```
    }
    JSONArray commentsArr =
postObject.getJSONArray("comments");
    ArrayList<Comment> commentsArrayList = new
ArrayList<>();
    ArrayList commObjList =
googleJson.fromJson(String.valueOf(commentsArr), ArrayList.class);
    for (int k = 0; k < commObjList.size(); k++) {
        JSONObject imgJson = commentsArr.getJSONObject(k);
        com = new Comment(
            imgJson.getString("content"),
            imgJson.getString("createdAt"),
            imgJson.getString("name"),
            imgJson.getString("mail"));
        commentsArrayList.add(com);
    }
    String phones = "";
    JSONArray arr2 = postObject.getJSONArray("phones");
    ArrayList ObjList =
googleJson.fromJson(String.valueOf(arr2), ArrayList.class);
    for (int k=0; k<ObjList.size();k++){
        phones+=arr2.getString(k)+" ";
    }
    int visitors =
postObject.getJSONObject("stat").getInt("visitors");
    String createdAt = postObject.getString("createdAt");
    String cityID =
postObject.getJSONObject("cityID").getString("name");
    int number = postObject.getInt("number");
    String title = postObject.getString("title");
    String price =
postObject.getJSONObject("price").getString("kind");
    if (price.equals("value")) {
        price =
uiPrice(String.valueOf(postObject.getJSONObject("price").getInt("value")))+" ₸";
    } else {
        if (price.equals("trade")) {
            if (Maltabu.lang.toLowerCase().equals("ru")) {
                price = "Договорная цена";
            } else {
                String kazName =
fileHelper.diction().getString("Договорная цена");
                price = kazName;
            }
        }
    }
}
```

## Продолжение приложения Б

```
    }
    } else {
        if (price.equals("free")) {
            if (Maltabu.lang.toLowerCase().equals("ru")) {
                price = "Отдам даром";
            } else {
                String kazName =
fileHelper.diction().getString("Отдам даром");
                price = kazName;
            }
        }
    }
}
if (postObject.getBoolean("hasContent")) {
    String content = postObject.getString("content");
    Post post2 = new Post(visitors, getDate(createdat), title,
content, cityID, price, String.valueOf(number),
imagesArrayList,commentsArrayList);
    post2.setPhones(phones);
    post = post2;
    setContentView(R.layout.post_details);
    initView();
    setInfo();
    if(epicDialog.isShowing()) {
        epicDialog.dismiss();
    }
} else {
    Post post2 = new Post(visitors, getDate(createdat), title,
cityID, price, String.valueOf(number), imagesArrayList,commentsArrayList);
    post2.setPhones(phones);
    post = post2;
    setContentView(R.layout.post_details);
    initView();
    setInfo();
    if(epicDialog.isShowing()) {
        epicDialog.dismiss();
    }
}
} catch (JSONException e) {
    e.printStackTrace();
}
}
```

## Продолжение приложения Б

```
};
asyncTask1.execute();
}

private void setInfo() {
    title.setText(post.getTitle());
    if (post.getContent()!=null) {
        content.setText(post.getContent());
    }
    price.setText(post.getPrice());
    location.setText(post.getCityID());
    String dates [] = post.getCreatedAt().split(",");
    if (Maltabu.lang.equals("ru"))
        date.setText(dates[0]+ " "+dates[1]);
    else
        date.setText(dates[0]+ " "+dates[2]);
    PAGE_COUNT = post.getImages().size();
    if(post.getImages().size()>0)
        photos.setText(String.valueOf("1/"+post.getImages().size()));
    img.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });
    pager = (ViewPager) findViewById(R.id.pager);
    pagerAdapter = new MyPagerAdapter(getSupportFragmentManager());
    pager.setAdapter(pagerAdapter);
    pager.setCurrentItem(0);
    phonePlaceholder();
    cs1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            makeCall();
        }
    });
    tapGestureDetector = new TapGestureDetector(this, new TapGestureListener());
    pager.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            tapGestureDetector.onTouchEvent(event);
        }
    });
}
```

## Продолжение приложения Б

```
return false;
    }
    });
    pager.setOnPageChangeListener(new
ViewPager.OnPageChangeListener() {
    @Override
    public void onPageSelected(int position) {
        selectedImg = position;
        if(post.getImages().size(>0)

photos.setText(String.valueOf(position+1+"/"+post.getImages().size()));
    }
    @Override
    public void onPageScrolled(int position, float positionOffset,
        int positionOffsetPixels) {

    }
    @Override
    public void onPageScrollStateChanged(int state) {
    }
    });
    phone.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String [] gg = phonesArr(post.getPhones());
        currentNumber = gg[0];
        showPhoneNumber();
    }
    });
    hot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(Maltabu.isAuth.equals("false")){
            startActivity(new Intent(ShowDetails.this, AuthAvtivity.class));
        } else {
            getPosting();
        }
    }
    });
    top.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(Maltabu.isAuth.equals("false")){
            startActivity(new Intent(ShowDetails.this, AuthAvtivity.class));
```

## Продолжение приложения Б

```
        } else {
            getPosting();
        }
    }
});
comments.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent comIntent = new Intent>ShowDetails.this,
CommentsActivity.class);
        comIntent.putExtra("post", post);
        startActivity(comIntent);
        finish();
    }
});
}
private void showPhoneNumber(){
    final String [] gg = phonesArr(post.getPhones());

    phone.setText("+7("+gg[0].substring(0,3)+")"+gg[0].substring(3,gg[0].length()));
    Toast.makeText>ShowDetails.this, currentNumber,
Toast.LENGTH_SHORT).show();
    if(phonecount<gg.length) {
        if (gg.length > 1) {
            for (int j = 1; j < gg.length; j++) {
                TextView phone2 = new TextView>ShowDetails.this);
                phone2.setPadding(5,5,5,5);
                phone2.setText("+7(" + gg[j].substring(0, 3) + ")" +
gg[j].substring(3, gg[j].length()));
                final int finalJ = j;
                phone2.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        currentNumber = gg[finalJ];
                        Toast.makeText>ShowDetails.this, gg[finalJ],
Toast.LENGTH_SHORT).show();
                    }
                });
                phonecount++;
                if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
                    phone2.setTextColor(getColor(R.color.MaltabuBlue));
                }
            }
        }
    }
}
```

## Продолжение приложения Б

```
        layout.addView(phone2);
        }
    }
    showPhone(post.getNumber());
}
}
private void initView() {
    cs1 = (ConstraintLayout) findViewById(R.id.callPhone);
    callPhoneText = (TextView) findViewById(R.id.textView32);
    commen = (TextView) findViewById(R.id.commenttxt);
    img = (ImageView) findViewById(R.id.finish);
    comments = (ConstraintLayout)
findViewById(R.id.constraintLayout19);
    layout = (LinearLayout) findViewById(R.id.lin);
    title = (TextView) findViewById(R.id.textView2);
    content = (TextView) findViewById(R.id.textView777);
    price = (TextView) findViewById(R.id.textView6);
    phone = (TextView) findViewById(R.id.phoneTxt);
    location = (TextView) findViewById(R.id.locationTxt);
    date = (TextView) findViewById(R.id.dateTxt);
    photos = (TextView) findViewById(R.id.photos);
    hot = (ConstraintLayout) findViewById(R.id.constraintLayout21);
    top = (ConstraintLayout) findViewById(R.id.constraintLayout30);
}
private class MyFragmentPagerAdapter extends FragmentPagerAdapter {
    public MyFragmentPagerAdapter(FragmentManager fm) {
        super(fm);
    }
    @Override
    public Fragment getItem(int position) {
        return ImageFragment.newInstance(position,
post.getImages().get(position).getSmall());
    }
    @Override
    public int getCount() {
        return PAGE_COUNT;
    }
}
class TapGestureListener extends
GestureDetector.SimpleOnGestureListener{
    @Override
    public boolean onSingleTapConfirmed(MotionEvent e) {
        imagesIntent.putExtra("post", post);
    }
}
```

## Продолжение приложения Б

```
imagesIntent.putExtra("pageCount", PAGE_COUNT);
    imagesIntent.putExtra("select", selectedImg);
    startActivity(imagesIntent);
    return false;
}
}
public void showPhone(String numb){
    final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
        .url("http://maltabu.kz/v1/api/clients/posts/"+numb+"/phone")
        .get()
        .addHeader("isAuthorized", "false")
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response2 = client.newCall(request2).execute();
                if (!response2.isSuccessful()) {
                    return null;
                }
                return response2.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
        @Override
        protected void onPostExecute(String s1) {
            super.onPostExecute(s1);
            if (s1 != null) {
            }
        }
    };
    asyncTask1.execute();
}
public String [] phonesArr(String s){
    String [] spl = s.split(";");
    for(int i=0; i<spl.length;i++){
        spl[i].replace(";", "");
    }
    return spl;
}
```

## Продолжение приложения Б

```
}
public void phonePlaceholder(){
    String [] gg = phonesArr(post.getPhones());
    currentNumber = gg[0];
    String placeHplder="+7("+gg[0].substring(0,3);
    Context context = LocaleHelper.setLocale(this, Maltabu.lang);
    Resources resources = context.getResources();
    callPhoneText.setText(resources.getString(R.string.CallPhone));
    if(post.getComments().size()>0)

    commen.setText(resources.getString(R.string.showComments)+"("+String.valueOf
(post.getComments().size())+"");
        else
            commen.setText(resources.getString(R.string.comments2));
            phone.setText(placeHplder+"
"+resources.getString(R.string.showPhone));
    }
    public void makeCall(){
        if (hasPermissions())
        {
            Intent callIntent = new Intent(Intent.ACTION_DIAL);
            callIntent.setData(Uri.parse("tel:+7"+currentNumber));
            showPhoneNumber();
            startActivity(callIntent);
        }
        else {
            requestPermissionWithRationale();
        }
    }
    private boolean hasPermissions(){
        int res = 0;
        String[] permissions = new
String[]{Manifest.permission.CALL_PHONE};
        for (String perms : permissions){
            res = checkCallingOrSelfPermission(perms);
            if (!(res == PackageManager.PERMISSION_GRANTED)){
                return false;
            }
        }
        return true;
    }
    public void requestPermissionWithRationale() {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
```

## Продолжение приложения Б

```
Manifest.permission.READ_EXTERNAL_STORAGE)) {
    final String message = "Storage permission is needed to show files
count";
    Snackbar.make(ShowDetails.this.findViewById(R.id.seelected),
message, Snackbar.LENGTH_LONG)
        .setAction("GRANT", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                requestPerms();
            }
        })
        .show();
    } else {
        requestPerms();
    }
}
private void requestPerms(){
    String[] permissions = new
String[]{Manifest.permission.CALL_PHONE};
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
        requestPermissions(permissions,PERMISSION_REQUEST_CODE);
    }
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    boolean allowed = true;
    switch (requestCode) {
        case PERMISSION_REQUEST_CODE:
            for (int res : grantResults) {
                allowed = allowed && (res ==
PackageManager.PERMISSION_GRANTED);
            }
            break;
        default:
            allowed = false;
            break;
    }
    if (allowed) {
        String [] gg= phonesArr(post.getPhones());
        Intent callIntent = new Intent(Intent.ACTION_DIAL);
        callIntent.setData(Uri.parse("tel:+7"+gg[0]));
        startActivity(callIntent);
    }
}
```

## Продолжение приложения Б

```
    } else {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if
(shouldShowRequestPermissionRationale(Manifest.permission.CALL_PHONE)) {
                Toast.makeText(this, "Storage Permissions denied.",
                    Toast.LENGTH_SHORT).show();
            } else {
                showNoStoragePermissionSnackbar();
            }
        }
    }
}

public void showNoStoragePermissionSnackbar() {
    Snackbar.make(ShowDetails.this.findViewById(R.id.selected),
        "Storage permission isn't granted" , Snackbar.LENGTH_LONG)
        .setAction("SETTINGS", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openApplicationSettings();
                Toast.makeText(getApplicationContext(),
                    "Open Permissions and grant the Storage permission",
                    Toast.LENGTH_SHORT)
                    .show();
            }
        })
        .show();
}

public void openApplicationSettings() {
    Intent appSettingsIntent = new
Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS,
        Uri.parse("package:" + getPackageName()));
    startActivityForResult(appSettingsIntent,
PERMISSION_REQUEST_CODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    if (requestCode == PERMISSION_REQUEST_CODE) {
        String [] gg= phonesArr(post.getPhones());
        Intent callIntent = new Intent(Intent.ACTION_DIAL);
        callIntent.setData(Uri.parse("tel:+7"+gg[0]));
        startActivity(callIntent);
        return;
    }
}
```

## Продолжение приложения Б

```
    }
    super.onActivityResult(requestCode, resultCode, data);
}
public void getPosting(){
    final OkHttpClient client = new OkHttpClient();
    final Request request2 = new Request.Builder()
        .url("http://maltabu.kz/v1/api/clients/cabinet/posting")
        .get()
        .addHeader("isAuthorized", Maltabu.isAuth)
        .addHeader("token", Maltabu.token)
        .build();
    AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            try {
                Response response2 = client.newCall(request2).execute();
                if (!response2.isSuccessful()) {
                    Maltabu.token = null;
                    Maltabu.isAuth = "false";
                    fileHelper.writeUserFile("");
                    fileHelper.writeToken("");
                    startActivity(new Intent(ShowDetails.this,
AuthActivity.class));
                    finish();
                    return null;
                }
                return response2.body().string();
            } catch (Exception e) {
                e.printStackTrace();
                return null;
            }
        }
        @Override
        protected void onPostExecute(String s1) {
            super.onPostExecute(s1);
            if (s1 != null) {
                fileHelper.writePostingFile(s1);
                getUser();
            }
        }
    };
    asyncTask1.execute();
}
```

## Продолжение приложения Б

```
}
    public void getUser(){
        final OkHttpClient client = new OkHttpClient();
        final Request request2 = new Request.Builder()
            .url("http://maltabu.kz/v1/api/login/clients")
            .get()
            .addHeader("isAuthorized", Maltabu.isAuth)
            .addHeader("token", Maltabu.token)
            .build();
        AsyncTask<Void, Void, String> asyncTask1 = new AsyncTask<Void,
Void, String>() {
            @Override
            protected String doInBackground(Void... params) {
                try {
                    Response response2 = client.newCall(request2).execute();
                    if (!response2.isSuccessful()) {
                        return null;
                    }
                    return response2.body().string();
                } catch (Exception e) {
                    e.printStackTrace();
                    return null;
                }
            }
            @Override
            protected void onPostExecute(String s1) {
                super.onPostExecute(s1);
                if (s1 != null) {
                    Intent hotIntent = new Intent(ShowDetails.this,
TopHotActivity.class);
                    hotIntent.putExtra("number", post.getNumber());
                    hotIntent.putExtra("rrr", post.getTitle());
                    startActivity(hotIntent);
                }
            }
        };
        asyncTask1.execute();
    }
}
package kz.maltabu.app.maltabukz.adapters;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
```

## Продолжение приложения Б

```
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.AsyncTask;
import android.support.constraint.ConstraintLayout;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.google.gson.Gson;
import com.squareup.picasso.Picasso;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import kz.maltabu.app.maltabukz.R;
import kz.maltabu.app.maltabukz.activities.ShowDetails;
import kz.maltabu.app.maltabukz.helpers.FileHelper;
import kz.maltabu.app.maltabukz.helpers.Maltabu;
import kz.maltabu.app.maltabukz.models.Comment;
import kz.maltabu.app.maltabukz.models.Image;
import kz.maltabu.app.maltabukz.models.Post;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
public class PostRecycleAdapterNew extends
RecyclerView.Adapter<PostRecycleAdapterNew.ViewHolder>{
    private ArrayList<Post> posts;
    private Context context;
    private FileHelper fileHelper;
    private JSONObject object;
    public PostRecycleAdapterNew(ArrayList<Post> posts, Context context) {
        this.posts = posts;
        this.context = context;
        this.fileHelper = new FileHelper(context);
        try {
            object = fileHelper.diction();
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    @Override
```

## Продолжение приложения Б

```
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
{
    View v = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.item_item,parent,false);
    return new ViewHolder(v);
}
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    final Post post = posts.get(position);
    holder.nameView.setText(post.getTitle());
    holder.visitors.setText(post.getVisitors());
    holder.nameView2.setText(post.getPrice());

    holder.commCount.setText(String.valueOf(post.getComments().size()));
    String dates [] = post.getCreatedAt().split(",");
    if (Maltabu.lang.equals("ru")) {
        holder.nameView3.setText(post.getCityID()+",    "+dates[0]+    "
"+dates[1]);
    } else {
        try {
            holder.nameView3.setText(object.getString(post.getCityID())
                +", "+dates[0]+ " "+dates[2]);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    if(post.getImages().size()>0) {
        Picasso.with(context).load("http://maltabu.kz/"
+post.getImages().get(0).getExtra_small()).placeholder(R.drawable.listempty).cent
erCrop().fit().into(holder.img);
        holder.photoCount.setText(String.valueOf(post.getImages().size()));
    } else {

    holder.img.setImageDrawable(context.getDrawable(R.drawable.listempty));
    holder.photoCount.setText(String.valueOf(0));
    }
    holder.selected.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            SecondThread thread = new SecondThread(post.getNumber());
            thread.start();
        }
    });
}
```

## Продолжение приложения Б

```
});
}
@Override
public int getItemCount() {
    return posts.size();
}
public class ViewHolder extends RecyclerView.ViewHolder{
    public TextView nameView ;
    public TextView nameView2;
    public TextView nameView3;
    public TextView photoCount;
    public TextView commCount;
    public TextView visitors;
    public ConstraintLayout selected;
    public ImageView img;
    public ViewHolder(View itemView) {
        super(itemView);
        selected = (ConstraintLayout)
itemView.findViewById(R.id.selectedPost);
        nameView = (TextView) itemView.findViewById(R.id.textView3);
        nameView2 = (TextView) itemView.findViewById(R.id.textView4);
        nameView3 = (TextView) itemView.findViewById(R.id.textView5);
        commCount = (TextView) itemView.findViewById(R.id.textView9);
        photoCount = (TextView)
itemView.findViewById(R.id.textView11);
        visitors = (TextView) itemView.findViewById(R.id.textView10);
        img = (ImageView) itemView.findViewById(R.id.imageView17);
    }
}

public void getPost(String numb){
    Intent details = new Intent(context, ShowDetails.class);
    details.putExtra("postNumb", numb);
    context.startActivity(details);
}
public class SecondThread extends Thread{
    String numb;
    SecondThread (String numb){
        this.numb = numb;
    }
    @Override
    public void run() {
        getPost(numb);
    }
}
```

## Продолжение приложения Б

```
}  
  }  
}  
package kz.maltabu.app.maltabukz.fragments;  
import android.app.Dialog;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.design.widget.FloatingActionButton;  
import android.support.v4.app.Fragment;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import com.google.gson.Gson;  
import kz.maltabu.app.maltabukz.R;  
import kz.maltabu.app.maltabukz.activities.FilterActivity;  
import kz.maltabu.app.maltabukz.adapters.PostRecycleAdapterNew;  
import kz.maltabu.app.maltabukz.helpers.EndlessListener;  
import kz.maltabu.app.maltabukz.helpers.FileHelper;  
import kz.maltabu.app.maltabukz.helpers.Maltabu;  
import kz.maltabu.app.maltabukz.models.Comment;  
import kz.maltabu.app.maltabukz.models.FilterModel;  
import kz.maltabu.app.maltabukz.models.Image;  
import kz.maltabu.app.maltabukz.models.Post;  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
import java.io.BufferedInputStream;  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;
```

## Продолжение приложения Б

```
import java.net.URL;
import java.util.ArrayList;
import java.util.concurrent.TimeUnit;
public class CategoryFragment extends Fragment {
    private JSONArray resObj;
    private ImageView imageView36;
    private TextView textView56,noPostsText;
    private String catalog;
    private int page;
    private ArrayList<Post> posts;
    private boolean isCatalog;
    private RecyclerView lst;
    private JSONObject object;
    private FloatingActionButton filterButton;
    private View view;
    private Dialog epicDialog;
    private ProgressBar button;
    private boolean can = true;
    private FileHelper fileHelper;
    private EndlessListener listener;
    private PostRecycleAdapterNew adapter;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        posts = new ArrayList<>();
        Bundle bundle = this.getArguments();
        epicDialog = new Dialog(getActivity());
        fileHelper = new FileHelper(getActivity());
        isCatalog = bundle.getBoolean("isCatalog");
        catalog = bundle.getString("catalog");
        page = 1;
        sDialog();
        view=inflater.inflate(R.layout.category_fragment, container, false);
        load(view);
        SecondThread thread = new SecondThread();
        thread.start();
        return view;
    }
    public View load(View view){
```

## Продолжение приложения Б

```
button = (ProgressBar) view.findViewById(R.id.button);
lst = (RecyclerView) view.findViewById(R.id.prodss);
filterButton = (FloatingActionButton)
view.findViewById(R.id.filterButton);
adapter = new PostRecycleAdapterNew(posts, getActivity());
imageView36 = (ImageView) view.findViewById(R.id.imageView36);
textView56 = (TextView) view.findViewById(R.id.textView56);
noPostsText = (TextView) view.findViewById(R.id.noPostsText);
LinearLayoutManager manager = new
LinearLayoutManager(getActivity());
lst.setHasFixedSize(true);
lst.setAdapter(adapter);
lst.setLayoutManager(manager);
if(Maltabu.filterModel!=null){
    filterButton.setVisibility(View.VISIBLE);
    filterButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            getActivity().startActivity(new Intent(getActivity(),
FilterActivity.class));
            getActivity().finish();
        }
    });
}
else {
    filterButton.setVisibility(View.GONE);
}
listener = new EndlessListener(manager) {
    @Override
    public void onLoadMore(int page, int totalItemsCount, RecyclerView
view) {
        button.setVisibility(View.VISIBLE);
        button.setIndeterminate(true);
        can = false;
        SecondThread thread = new SecondThread();
        thread.start();
    }
};
lst.addOnScrollListener(listener);
return view;
}
private void post() {
    AsyncTask<String, Void, String> task = new AsyncTask<String,
```

## Продолжение приложения Б

```
Void, String>() {
    @Override
    protected String doInBackground(String... urls) {
        try {
            try {
                return HttpPost2("http://maltabu.kz/v1/api/clients/posts");
            } catch (JSONException e) {
                e.printStackTrace();
                return "Error!";
            }
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    @Override
    protected void onPostExecute(String result) {
        try {
            JSONObject Obj = new JSONObject(result);
            if(Obj.getInt("count")==0)
            {
                imageView36.setVisibility(View.VISIBLE);
                textView56.setVisibility(View.VISIBLE);
                noPostsText.setVisibility(View.VISIBLE);
                if (epicDialog != null && epicDialog.isShowing()) {
                    epicDialog.dismiss();
                }
            }
            else {
                imageView36.setVisibility(View.GONE);
                textView56.setVisibility(View.GONE);
                noPostsText.setVisibility(View.GONE);
                resObj = Obj.getJSONArray("posts");
                catalogList();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
};
task.execute();
}
private void setPostRequestContent(URLConnection conn,
JSONObject jsonObject) throws IOException {
    OutputStream os = conn.getOutputStream();
```

## Продолжение приложения Б

```
BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os,
"UTF-8"));
    writer.write(jsonObject.toString());
    writer.flush();
    writer.close();
    os.close();
}
private JSONObject buildJsonObject2() throws JSONException {
    JSONObject jsonObject = new JSONObject();
    if (this.isCatalog) {
        jsonObject.accumulate("catalogID", catalog);
    } else {
        jsonObject.accumulate("categoryID", catalog);
    }
    jsonObject.accumulate("byTime", Maltabu.byTime);
    jsonObject.accumulate("increment", Maltabu.increment);
    jsonObject.accumulate("countPosts", true);
    jsonObject.accumulate("page", this.page);
    if(Maltabu.filterModel!=null){
        FilterModel filter = Maltabu.filterModel;
        if(filter.getRegId()!=null){
            jsonObject.accumulate("regionID", filter.getRegId());
        }
        if(filter.getCityId()!=null){
            jsonObject.accumulate("cityID", filter.getCityId());
        }
        if(filter.getPrice1()!=null){
            jsonObject.accumulate("fromPrice", filter.getPrice1());
        }
        if(filter.getPrice2()!=null){
            jsonObject.accumulate("toPrice", filter.getPrice2());
        }
        jsonObject.accumulate("onlyImages", filter.isWithPhoto());
        jsonObject.accumulate("onlyExchange", filter.isBarter());
        jsonObject.accumulate("onlyEmergency", filter.isBargain());
    }
    page++;
    return jsonObject;
}
private String HttpPost2(String myUrl) throws IOException,
JSONException {
    StringBuilder res = new StringBuilder();
    URL url = new URL(myUrl);
```

## Продолжение приложения Б

```
URLConnection conn = (URLConnection) url.openConnection();
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Content-Type", "application/json;
charset=utf-8");
    conn.setRequestProperty("isAuthorized", "false");
    conn.setDoOutput(true);
    JSONObject jsonObject = buildJsonObject2();
    setPostRequestContent(conn, jsonObject);
    conn.connect();
    InputStream in = new BufferedInputStream(conn.getInputStream());
    BufferedReader reader = new BufferedReader(new
InputStreamReader(in));
    String line;
    while ((line = reader.readLine()) != null) {
        res.append(line);
    }
    return res.toString();
}
private void catalogList() throws JSONException {
    Gson googleJson = new Gson();
    object = fileHelper.diction();
    ArrayList postObjList = googleJson.fromJson(String.valueOf(resObj),
ArrayList.class);
    JSONObject postObject = new JSONObject();
    Image image = new Image();
    Comment com = new Comment();
    Post post = new Post();
    JSONObject imgJson = new JSONObject();
    for (int i = 0; i < postObjList.size(); i++) {
        postObject = resObj.getJSONObject(i);
        ArrayList<Image> imagesArrayList = new ArrayList<>();
        JSONArray arr = postObject.getJSONArray("images");
        ArrayList imgObjList = googleJson.fromJson(String.valueOf(arr),
ArrayList.class);
        for (int j = 0; j < imgObjList.size(); j++) {
            imgJson = arr.getJSONObject(j);
            image = new Image(
                imgJson.getString("extra_small"),
                imgJson.getString("small"),
                imgJson.getString("medium"),
                imgJson.getString("big"));
            imagesArrayList.add(image);
        }
    }
}
```

## Продолжение приложения Б

```
JSONArray commentsArr = postObject.getJSONArray("comments");
    ArrayList<Comment> commentsArrayList = new ArrayList<>();
    ArrayList          commObjList          =
googleJson.fromJson(String.valueOf(commentsArr), ArrayList.class);
    for (int k = 0; k < commObjList.size(); k++) {
        com = new Comment();
        commentsArrayList.add(com);
    }
    int visitors = postObject.getJSONObject("stat").getInt("visitors");
    String createdAt = postObject.getString("createdAt");
    String          cityID          =
postObject.getJSONObject("cityID").getString("name");
    int number = postObject.getInt("number");
    String title = postObject.getString("title");
    String price = postObject.getJSONObject("price").getString("kind");
    if (price.equals("value")) {
        price          =
uiPrice(String.valueOf(postObject.getJSONObject("price").getInt("value"))+" ₸");
    } else {
        if (price.equals("trade")) {
            if (Maltabu.lang.toLowerCase().equals("ru")) {
                price = "Договорная цена";
            } else {
                String kazName = object.getString("Договорная цена");
                price = kazName;
            }
        } else {
            if (price.equals("free")) {
                if (Maltabu.lang.toLowerCase().equals("ru")) {
                    price = "Отдам даром";
                } else {
                    String kazName = object.getString("Отдам даром");
                    price = kazName;
                }
            }
        }
    }
    }
    if (postObject.getBoolean("hasContent")) {
        String content = postObject.getString("content");
        post = new Post(visitors, getDate(createdAt), title, content, cityID,
price, String.valueOf(number), imagesArrayList,commentsArrayList);
        posts.add(post);
    } else {
```

## Продолжение приложения Б

```
        post = new Post(visitors, getDate(createdAt), title, cityID, price,
String.valueOf(number), imagesArrayList,commentsArrayList);
        posts.add(post);
    }
}
try {
    lst.invalidate();
    adapter.notifyDataSetChanged();
    lst.refreshDrawableState();
} catch (Exception e) {}
if (!can) {
    can = true;
    try {
        TimeUnit.MILLISECONDS.sleep(1000);
        button.setVisibility(View.INVISIBLE);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
} else {
    if (epicDialog != null && epicDialog.isShowing()) {
        epicDialog.dismiss();
    }
}
}
protected void sDialog() {
    epicDialog.setContent View(R.layout.progress_dialog);
    epicDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
    epicDialog.show();
}
public class SecondThread extends Thread{
    SecondThread(){ }
    @Override
    public void run() {
        post();
    }
}
}
package kz.maltabu.app.maltabukz.helpers;
import android.app.Activity;
import android.content.Context;
import android.util.Patterns;
import android.view.View;
```

## Продолжение приложения Б

```
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
public class InputValidation {
    private Context context;
    public InputValidation(Context context) {
        this.context = context;
    }
    public boolean isInputEditTextEmail(EditText EditText) {
        String value = EditText.getText().toString().trim();
        if (value.isEmpty() ||
!android.util.Patterns.EMAIL_ADDRESS.matcher(value).matches()) {
            hideKeyboardFrom(EditText);
            return false;
        }
        return true;
    }
    public boolean isInputEditTextFilled(EditText EditText) {
        String value = EditText.getText().toString().trim();
        if (value.isEmpty()) {
            hideKeyboardFrom(EditText);
            return false;
        }
        return true;
    }
    private void hideKeyboardFrom(View view) {
        InputMethodManager imm = (InputMethodManager)
context.getSystemService(Activity.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(view.getWindowToken(),
WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
    }
    public boolean validatePhoneNumber(EditText EditText) {
        String value = EditText.getText().toString().trim();
        if (value.isEmpty() || !Patterns.PHONE.matcher(value).matches()
||value.length()!=10) {
            return false;
        }
        return true;
    }
}
package kz.maltabu.app.maltabukz.models;
import android.os.Parcel;
import android.os.Parcelable;
```

## Продолжение приложения Б

```
import java.util.ArrayList;
public class Post implements Parcelable{
    private String createdAt;
    private String phones;
    private String updatedAt;
    private String title;
    private String visitors;
    private String content;
    private String catalogID;
    private String categoryID;
    private String regionID;
    private String cityID;
    private String price;
    private String priceValue;
    private String address;
    private String number;
    private ArrayList<Comment> comments;
    private ArrayList<Image> images;
    private boolean exchange;
    protected Post(Parcel in) {
        createdAt = in.readString();
        phones = in.readString();
        updatedAt = in.readString();
        title = in.readString();
        visitors = in.readString();
        content = in.readString();
        catalogID = in.readString();
        categoryID = in.readString();
        regionID = in.readString();
        cityID = in.readString();
        price = in.readString();
        priceValue = in.readString();
        address = in.readString();
        number = in.readString();
        comments = in.createTypedArrayList(Comment.CREATOR);
        images = in.createTypedArrayList(Image.CREATOR);
        exchange = in.readByte() != 0;
    }
    public static final Creator<Post> CREATOR = new Creator<Post>() {
        @Override
        public Post createFromParcel(Parcel in) {
            return new Post(in);
        }
    }
}
```

## Продолжение приложения Б

```
@Override
public Post[] newArray(int size) {
    return new Post[size];
}
};
public ArrayList<Comment> getComments() {
    return comments;
}
public void setComments(ArrayList<Comment> comments) {
    this.comments = comments;
}
public String getPhones() {
    return phones;
}
public void setPhones(String phones) {
    this.phones = phones;
}
public String getCreatedAt() {
    return createdAt;
}
public void setCreatedAt(String createdAt) {
    this.createdAt = createdAt;
}
public String getUpdatedAt() {
    return updatedAt;
}
public void setUpdatedAt(String updatedAt) {
    this.updatedAt = updatedAt;
}
public String getVisitors() {
    return visitors;
}
public void setVisitors(String visitors) {
    this.visitors = visitors;
}
public String getTitle() {
    return title;
}
public void setTitle(String title) {
    this.title = title;
}
public String getContent() {
    return content;
}
```

## Продолжение приложения Б

```
}  
    public void setContent(String content) {  
        this.content = content;  
    }  
    public String getCatalogID() {  
        return catalogID;  
    }  
    public void setCatalogID(String catalogID) {  
        this.catalogID = catalogID;  
    }  
    public String getCategoryID() {  
        return categoryID;  
    }  
    public void setCategoryID(String categoryID) {  
        this.categoryID = categoryID;  
    }  
    public String getRegionID() {  
        return regionID;  
    }  
    public void setRegionID(String regionID) {  
        this.regionID = regionID;  
    }  
    public String getCityID() {  
        return cityID;  
    }  
    public void setCityID(String cityID) {  
        this.cityID = cityID;  
    }  
    public String getPrice() {  
        return price;  
    }  
    public void setPrice(String price) {  
        this.price = price;  
    }  
    public String getPriceValue() {  
        return priceValue;  
    }  
    public void setPriceValue(String priceValue) {  
        this.priceValue = priceValue;  
    }  
    public String getAddress() {  
        return address;  
    }  
}
```

## Продолжение приложения Б

```
public void setAddress(String address) {
    this.address = address;
}
public String getNumber() {
    return number;
}
public void setNumber(String number) {
    this.number = number;
}
public ArrayList<Image> getImages() {
    return images;
}
public void setImages(ArrayList<Image> images) {
    this.images = images;
}
public Post (){}
public Post(String cityID, String price, String number, ArrayList<Image>
images){
    this.cityID = cityID;
    this.price = price;
    this.number = number;
    this.images = images;
}
public Post(int visitors, String createdAt, String title, String content, String
cityID, String price, String number, ArrayList<Image> images) {
    this.visitors = String.valueOf(visitors);
    this.createdAt = createdAt;
    this.title = title;
    this.content = content;
    this.cityID = cityID;
    this.price = price;
    this.number = number;
    this.images = images;
}
public Post(int visitors, String createdAt, String title, String content, String
cityID, String price, String number, ArrayList<Image> images,
ArrayList<Comment> comments) {
    this.visitors = String.valueOf(visitors);
    this.createdAt = createdAt;
    this.title = title;
    this.content = content;
    this.cityID = cityID;
    this.price = price;
```

## Продолжение приложения Б

```
this.number = number;
    this.images = images;
    this.comments = comments;
}
public Post(int visitors, String createdAt, String title, String cityID, String
price, String number, ArrayList<Image> images) {
    this.createdAt = createdAt;
    this.title = title;
    this.visitors = String.valueOf(visitors);
    this.cityID = cityID;
    this.price = price;
    this.number = number;
    this.images = images;
}
public Post(int visitors, String createdAt, String title, String cityID, String
price, String number, ArrayList<Image> images, ArrayList<Comment> comments)
{
    this.createdAt = createdAt;
    this.title = title;
    this.visitors = String.valueOf(visitors);
    this.cityID = cityID;
    this.price = price;
    this.number = number;
    this.images = images;
    this.comments = comments;
}
@Override
public int describeContents() {
    return 0;
}
@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(createdAt);
    dest.writeString(phones);
    dest.writeString(updatedAt);
    dest.writeString(title);
    dest.writeString(visitors);
    dest.writeString(content);
    dest.writeString(catalogID);
    dest.writeString(categoryID);
    dest.writeString(regionID);
    dest.writeString(cityID);
    dest.writeString(price);
}
```

## Продолжение приложения Б

```
dest.writeString(priceValue);
dest.writeString(address);
dest.writeString(number);
dest.writeTypedList(comments);
dest.writeTypedList(images);
dest.writeByte((byte) (exchange ? 1 : 0));
    }
}
package kz.maltabu.app.maltabukz;
import android.app.Application;
import android.content.Context;
import kz.maltabu.app.maltabukz.helpers.LocaleHelper;
public class MainApplication extends Application{
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(LocaleHelper.onAttach(base,"kk"));
    }
}
```

## АКТ ВНЕДРЕНИЯ мобильного приложения “Maltabu.kz”

Разработанное студентом 4-го курса Satbayev University Чипатаевым Нурсултаном Келисұлы мобильное приложение для ОС Android “Maltabu.kz” было передано в эксплуатацию в ИП “Maltabu.kz” в феврале 2019 года.

Назначение программы:

- Просмотр, подача и поиск объявлений;
- Регистрация и авторизация в системе;
- Фильтрация и сортировка объявлений;
- Обеспечение возможности смены языка интерфейса.

Для обеспечения работы программы необходимо мобильное устройство под управлением ОС Android версии 5.0 и выше, 40 Мб свободного места в памяти устройства и не менее 128 Мб ОЗУ.

Показатели экономической и социальной эффективности:

- Увеличение количества пользователей;
- Увеличение количества объявлений в день;
- Количество скачиваний в Google Play Market – больше 100 000;
- Сокращение времени на подачу объявления.

Представители ИП «Maltabu.kz»:  
Руководитель



Абдраимов Е.А.

Исполнитель:



Чипатаев Н.К.



13.05.2019

