МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

КАЛИНИНГРАДСКОЙ ОБЛАСТИ

ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СТРОИТЕЛЬСТВА»

(ГБУ КО ПОО «КИТиС»)

|  |  |  |
| --- | --- | --- |
| Курсовой проект  допущен к защите  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата)  Зам.директора по УМР  \_\_\_\_\_\_\_\_\_\_\_\_\_Павленко Г.Я.  (подпись) |  | Курсовой проект  защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Пояснительная записка к курсовому проекту

по дисциплине: МДК 01.01 Разработка программных модулей

Тема: Разработка мобильного приложения для мерчандайзера

Специальность 09.02.07 «Информационные системы и программирование»

|  |  |
| --- | --- |
| Выполнил: | студент 3 курса,  группы ИСп 19-1  Лучихин Максим Сергеевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель: | Большакова-Стрекалова Анна Викторовна, преподаватель ГБУ КО ПОО КИТиС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Калининград

2022

Содержание

|  |  |
| --- | --- |
| Введение | 4 |
| 1 Описание предметной области | 5 |
| 1.1 Аналоги разрабатываемого приложения | 5 |
| 1.2 Техническое задание | 8 |
| 1.3 Описание структуры приложения | 10 |
| 2 Описание разработки приложения | 11 |
| 2.1 Обоснование средств разработки | 11 |
| 2.2 Разработка интерфейса | 14 |
| 2.3 Разработка логики работы приложения. Схема взаимодействия компонентов проекта | 16 |
| 2.4 Описание переменных, компонентов, классов и подпрограмм | 17 |
| 3 Тестирование и установка приложения | 18 |
| Заключение | 22 |
| Список использованной литературы | 23 |
| Приложение А. Листинг рабочего кода | 24 |
| Приложение Б. Руководство пользователя | 67 |

Рецензии

Протестировав данное приложение, можно сказать, что оно имеет современный дизайн. Меню имеет встроенные анимации, приложение работает быстро.

Из недостатков, фотки не сохраняются если выйти из задания после выбора фотографий.

В итоге можно сказать, что для данного приложения нужны правки и доработки, если это все изменить. То оно приобретет интерес к пользователям.

Ссылка на GitHub - <https://github.com/Makssshow/MerchTriB>

Введение

Данный курсовой проект посвящён разработке мобильного приложения «Мерчандайзер». Для выполнения задание следует определить актуальность темы, её цели и задачи для её достижения.

Работа актуальна, потому что имеет большой потенциал в помощи бизнесу и улучшению качества работы мерчандайзеров.

Целью данной работы является создание мобильного приложения с возможностью отправки фотоотчетов по проделанной работе и создании списка заданий.

Задачи и цели выявить ключевые потребности пользователей, определить целевую аудиторию, разработать красивый минималистичный дизайн и простую навигацию. После составления графического интерфейса необходимо разобрать программу на несколько модулей и скомпилировать проект. После выпуска программного продукта, нужно составить набор тестов и воспроизвести их.

1. Описание предметной области
   1. Аналоги разрабатываемого приложения

Для понимания того, как должно выглядеть разрабатываемое приложение надо просмотреть продукты, которые уже представлены на рынке. Приложение, которое разрабатывается обладает обширным функционалом и поэтому я просмотрю несколько различных приложений с функционалом похожим на данное приложение.

Фотоотчет мерчандайзера

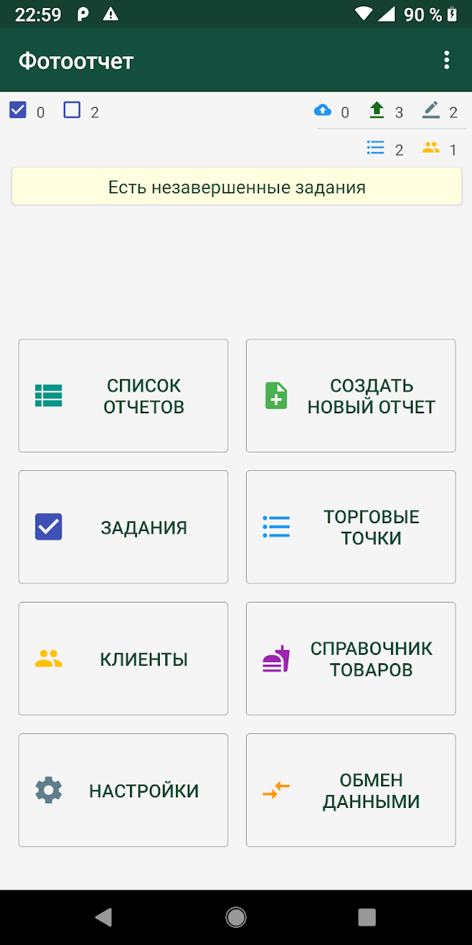


Рисунок 1 – меню приложения

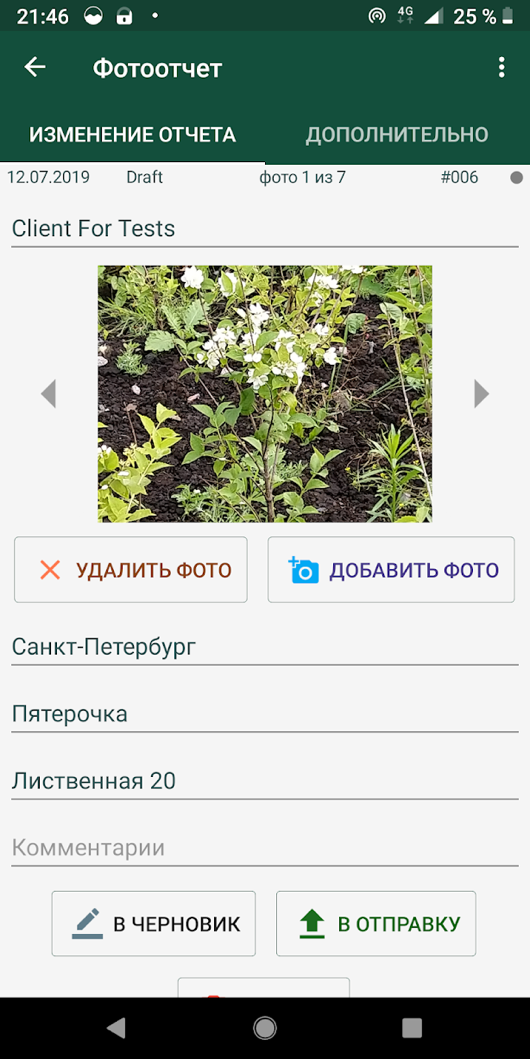


Рисунок 2 – фотоотчет

Минусы:

- Устаревший дизайн

Плюсы:

- Большой функционал

MerchHub Merchandiser

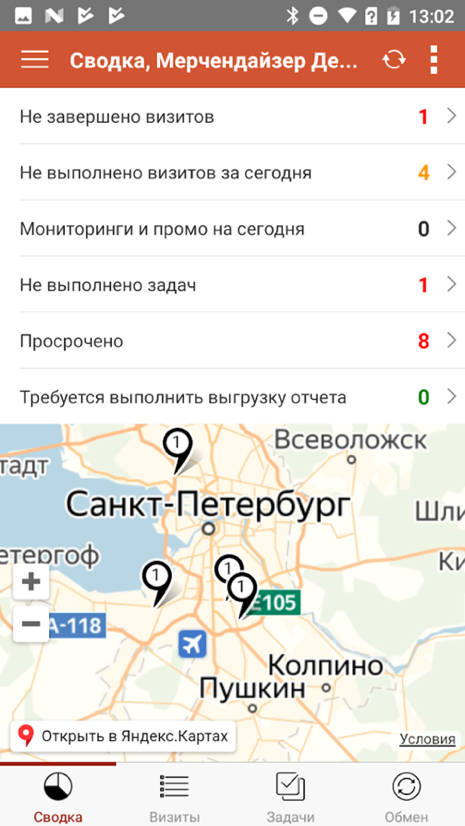


Рисунок 3 – сводка

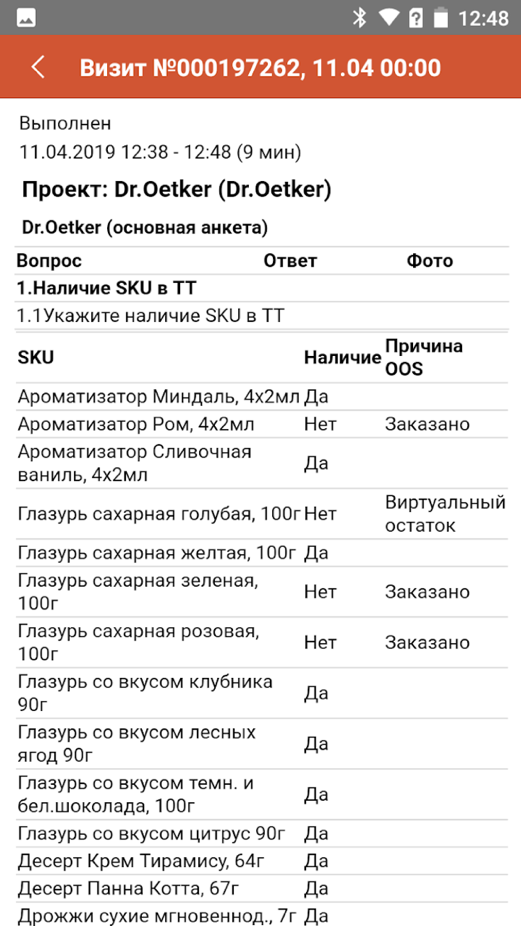


Рисунок 4 – анкета

Минусы:

- Не интуитивный интерфейс;

- Не совсем современный дизайн;

Плюсы:

- Встроенная карта;

- Быстрая работа.

У всех конкурентов устаревший дизайн и не понятный интерфейс, в этом проекте приложение будет иметь современный дизайн и булее интуитивный интерфейс.

1.2. Техническое задание

1. Общие сведения

Требуется разработать мобильное приложение «Мерчандайзер» для устройств под управлением операционной системы Android. Иконка приложения выполнена в векторном стиле и имеет простой и понятный дизайн. Приложение должно сохранять отправленные фотографии и делать их общедоступными среди пользователей одной компании. Приложение нужно предоставить до конца апреля 2022 года с полным набором ниже перечисленных функций, и предоставлением отчётов о тестировании.

При запуске приложения поваляется экран регистрации или если пользователь уже авторизован, то главный экран приложения с меню.

В пункте меню «Задания» выводится список текущих заданий, во вкладке «Отправленные» выводятся выполненные заданий, во вкладке «Пользователи» доступная только админам, выводится список пользователей и возможность их удаления и создания.

1. Требования к системе

Приложение должно быть доступно для устройств под управлением android, начиная с android версии 9.0. Приложение оптимизировано, чтобы оно могло запускаться на устройстве любой мощности. Вес приложения составил 8Мб.

1. Состав и содержание работ по созданию системы

Приложение должно быть разбито на несколько составляющих модулей, для более удобной дальнейшей работы с ним.Каждый этап работы приложения разбит на отдельные модули. Приложение должно иметь понятные комментарии на русском языке для каждого класса и функции.

1. Порядок контроля и приемки системы

После реализации приложения и составления всей документации, проект требуется выложить на репозиторий GitHub для её проверки руководителем. Требуется отправить все исходные файлы, документацию и готовый apk файл приложения.

1. Требования к составу и содержанию работ по подготовке объекта к вводу системы в действие

Требуется составить отчёты тестирования приложения, а также сопроводительную записку программиста и записку пользователя, с указанием принципов работы с приложением.

* 1. Описание структуры приложения

Приложение состоит из нескольких графических и программных модулей.

К графическим модулям относятся 7 activity:

* AddTaskActivity – activity для добавления задания в список;
* AdminActivity – главный экран администратора;
* LoginActivity – экран авторизации;
* MainActivity – главный экран пользователя;
* RegisterCompanyActivity – экран регистрации компании;
* SendingActivity – экран загрузки фото;
* TaskActivity – экран работы с отдельным заданием;

1. Описание разработки приложения

2.1 Обоснование средств разработки

Для выполнения проекта необходимо определить, с помощью каких средств будет разрабатываться приложение. Самими популярными средствами разработки для мобильных являются: Android Studio, Flutter, AppCode. Требуется определить преимущества каждого из средств разработки, и исходя из технических требований выбрать самый подходящий вариант.

Android Studio – это IDE (интегрированная среда разработки) для работы с платформой Android. Она позволяет разрабатывать приложения для смартфонов, планшетов, часов, телевизоров и других устройств на этой ОС.

Ограничения

Отсутствует возможность полностью отключить автосохранение файлов. Это принципиальная безальтернативная позиция разработчиков. Они считают, что такой подход более эффективен, чем тот, который кому-то удобен или привычен.

Преимущества данного средства разработки:

* Удобный редактор кода;
* Официальная среда разработки, которая поддерживает Google;
* Позволяет разрабатывать приложения для палншетов, телефонов, часов и телевизоров;
* Тестирование корректности работы новых игр, утилит, происходит в эмуляторе;
* Перепроектирование кода;
* достаточно большая библиотека с готовыми шаблонами и компонентами для разработки ПО;
* разработка приложений для последних версий операционной системы Android;

Недостатки Android Studio

* для тестирования приложения средствами встроенного Android-эмулятора требуется высокая аппаратная производительность ПК;
* невозможность написания серверных проектов на языке Java для ПК и Android устройств.

## Flutter

Бесплатный и открытый набор средств разработки мобильного пользовательского интерфейса, созданный компанией Google и выпущенный в мае 2017 года. Проще говоря, с помощью Flutter возможно создать собственное мобильное приложение с одним массивом кода. Это означает, что для создания двух приложений (IOS и Android) можно использовать единый язык программирования и одну базу кода.

Плюсы:

* Разработать мобильное приложение с Flutter дешевле, потому что не нужно создавать и — поддерживать два мобильных приложения (одно для IOS и Android).
* Для создания MVP достаточно одного разработчика.
* Это эффективно; невозможно заметить разницу между нативным приложением и приложением Flutter.
* Это красиво; Вы можете легко использовать виджеты, предоставляемые Flutter, и персонализировать их для создания оригинального пользовательского интерфейса для ваших клиентов.

AppCode

Среда разработки от компании JetBrains, полностью совместимая с Xcode и также обладающая большим набором инструментов для разработки.

Достоинства:

* Нет проблем, присущих Xcode: дополнение кода работает, программа не вылетает.
* Есть интеграции с Github, Jira, Redmine и другими популярными такс-трекерами.
* Подробное описание ошибок при компиляции приложений.
* Возможность сборки приложения на девайсе.

Недостатки:

* Xcode все равно потребуется для сборки проекта.
* Отсутствует Inrerface Builder.
* Нет отладчика и песочницы, как в Xcode.
* Запуск приложения будет проходить в эмуляторе Xcode.
* Платная среда разработки.
* Обновление языка происходит с опозданием в отличие от Xcode.

Анализ имеющихся средств разработки позволяет сделать вывод, что лучшим решением для данного проекта является Android Studio, так как именно данное решение больше всего соответствует требования.

Android Studio предоставляет возможность двух разных способов разработки – средством языка программирования Java и Kotlin. Необходимо определить какой способ является наиболее предпочтительным для данного проекта. Для этого нужно определить преимущества и недостатки этих способов.

Преимущества языка программирования Java:

1. С помощью виртуальной машины Java программы на этом языке могут запускаться практически в любой системе. В результате Java получает лидерство среди Android-приложений.
2. Огромное сообщество.
3. Кроссплатформенность.
4. Поскольку Android разработан на Java, этот язык имеет уже готовые библиотеки и SDK для облегчения процесса разработки.

Недостатки Java:

1. Java требует больше памяти и, по сравнению с другими языками, работает намного медленнее.
2. На Java тяжелее писать код.

Преимущества языка программирования Kotlin:

1. Повышенная производительность, достигающийся сокращением строк кода и оптимизацией.
2. Полная совместимость с Java.
3. Официальная поддержка Android Studio.
4. Простота синтаксиса.

Недостатки Kotlin:

1. Низкая скорость компиляции. В среднем компиляция на котлин дольше, чем на джава на 30-40 %. Это связано с тем, что компиляция в виртуальной машине JVM предназначена для джава, поэтому с родным языком она работает быстрее.
2. Мало дополнений. В силу молодости языка у него намного меньше библиотек и фреймворков, чем у Java.
3. Малочисленное сообщество.

Исходя из сравнения, можно сделать вывод, что язык программирования Java более прогнозируемый и стабильный по сравнению с Kotlin, а также имеет большую поддержку со стороны разработчиков. Поэтому в проекте будет использоваться средство разработки Android Studio с языком программирования Java.

2.2 Разработка пользовательского интерфейса.

Разработка интерфейса велась в приложении «Figma». Там были созданные основные компоненты интерфейса и главные цвета.

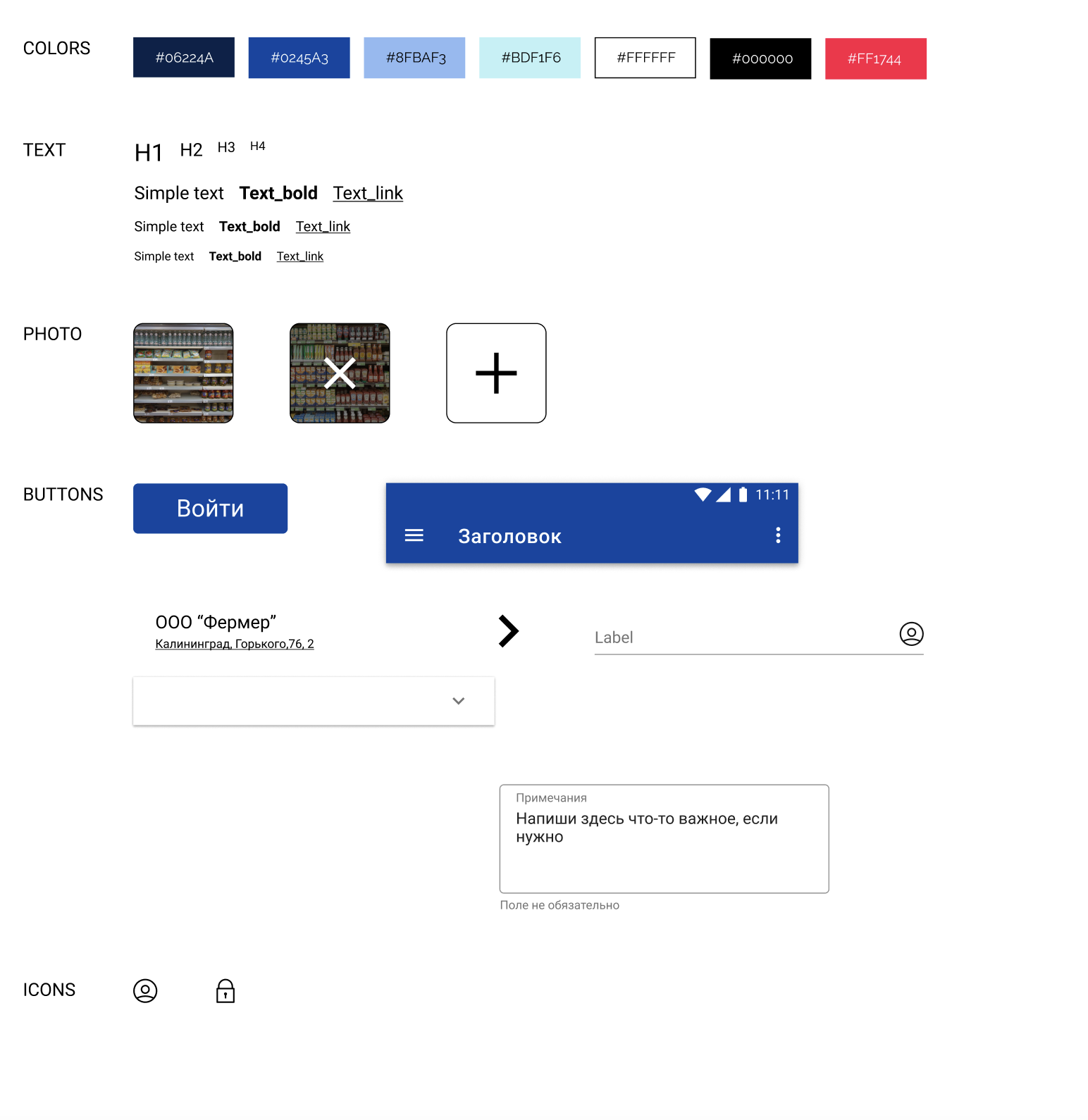


Рисунок 5 – элементы интерфейса

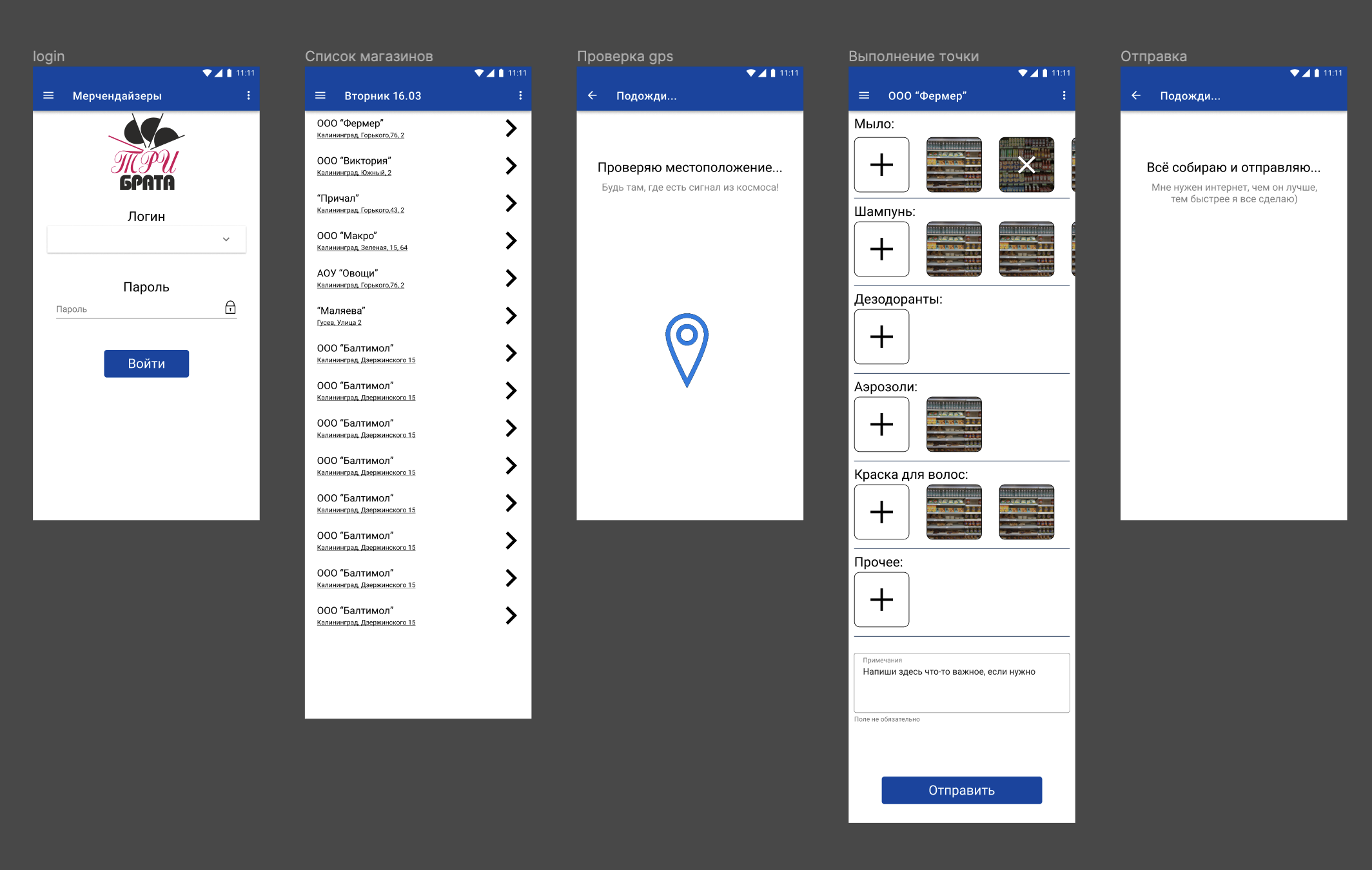


Рисунок 6 – Макеты экранов

* 1. Разработка логики работы приложения. Схема взаимодействия компонентов проекта.

Логика работы приложения была разработана и продумана в приложении «Figma». После регистрации пользователь попадает на главный экран с меню внизу экрана. На главном экране виден список текущих задач, на странице «Отправленные» есть список выполненных заданий. У администраторов доступен пункт меню «Люди», в котором есть список пользователей у компании. При выборе задания открывается экран этого задания, в котором вы видите фотографии и можете их добавить или удалить.

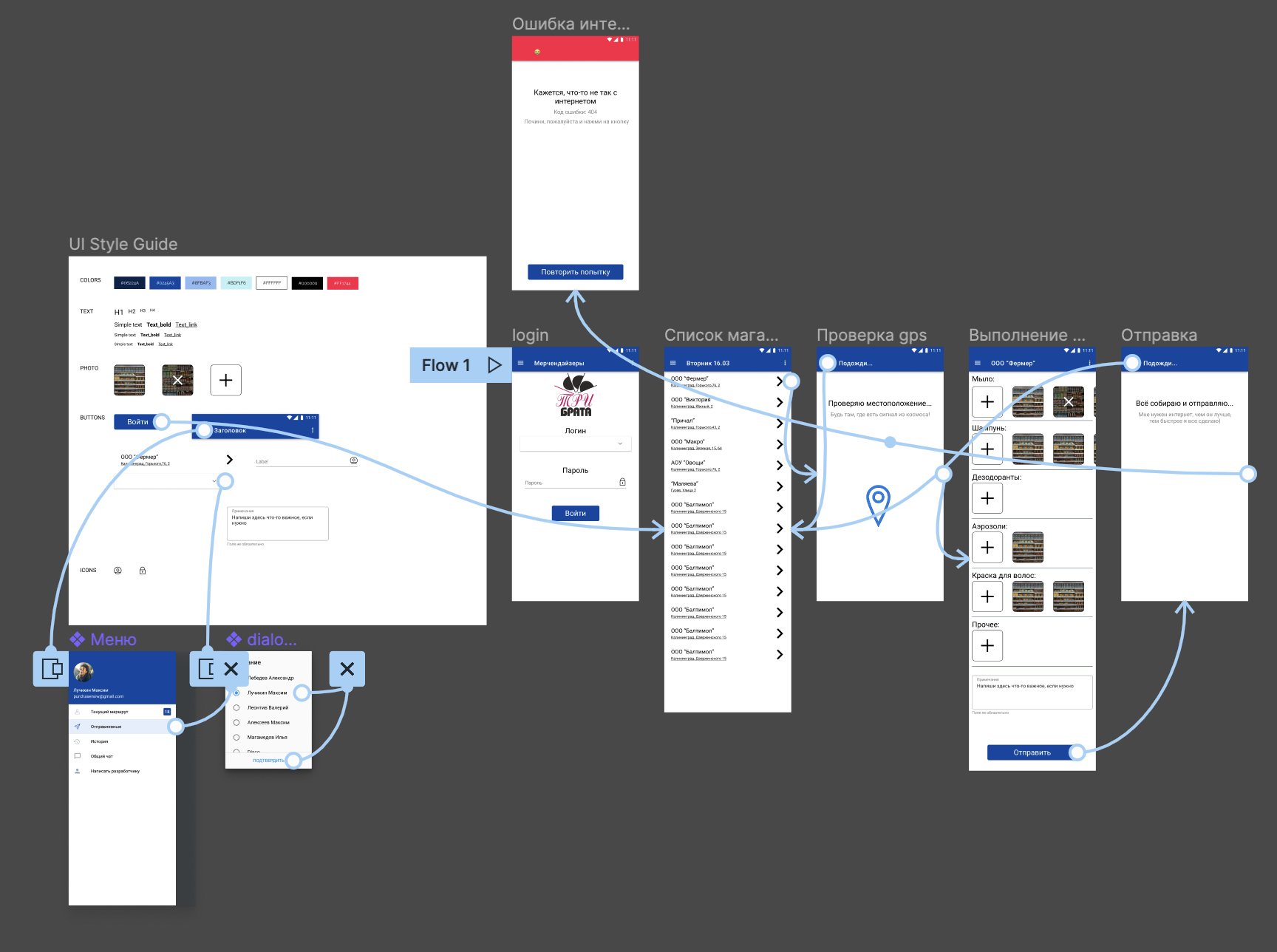


Рисунок 7 – Макет работы приложения

* 1. Описание переменных, компонентов, классов и подпрограмм.

Таблица 1 – Переменные

|  |  |
| --- | --- |
| companyName | название компании |
| name | Название задания |
| address | Название адреса |
| addressLink | Ссылка на адрес |
| user | Пользователь из базы данных |
| id | Идентификатор задания |
| mFirebaseDatabse | База данных |
| email | Поле электронной почты |
| password | Поле пароля |
| toolbar | Панель инструментов |
| galleryAdapter | Адаптер для галереи |
| type | Тип задания |
| preferences | Кэш приложения |
| isAdmin | Является ли админом |
| userEmail | Почта пользователя |
| emptyText | Текст если нет данных |

1. Тестирование и установка приложения

Тестирование производительности

Также известно как нагрузочное тестирование. Это автоматизированное тестирование, которое имитирует работу определенного количества пользователей какого-либо общего ресурса.

Основная цель этого вида тестирования — убедиться в том, что приложение работает приемлемо при определенных требованиях производительности: доступ большому числу пользователей.

Юзабилити-тестирование обычно проводится на пользователях, поскольку только люди могут понять субъективные ощущения других людей, вызываемые тем или иным приложением. Тестирование удобства использования — это метод тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.

В данном приложении кнопки имеют адекватный размер подходит для пользователей с различной толщиной пальцев, так же кнопки находятся в отличной видимости. Весь интерфейс выглядит естественно и одинаково во всех активити. Все кнопки с одинаковыми функциями выглядят одинаково. В приложении поля для ввода проверяют корректность введенных данных.

Для установки приложения непосредственно в среде разработки надо иметь эмулятор. Ниже приведена инструкция как его подключить и запустить на нем проект. Нам нужно найти панельку справа сверху и нажать на список рядом с зеленой стрелочкой (Рисунок 12)



Рисунок 8 – панелька

Далее в списке мы выбираем Device Manager (рисунок 13) и в появившемся окне выбираем create device (Рисунок 14)

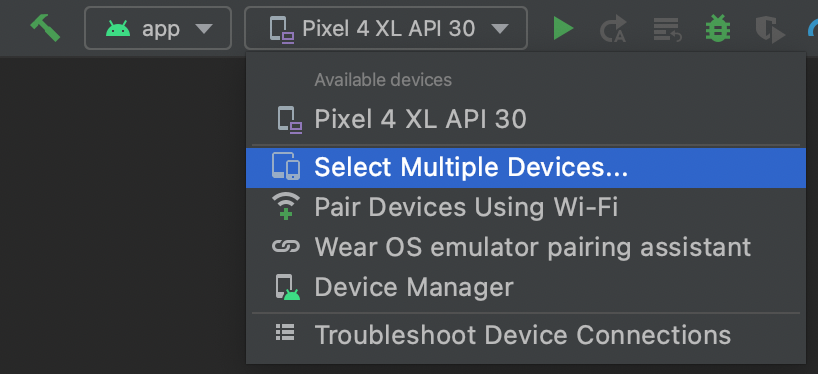


Рисунок 9 - Device Manager

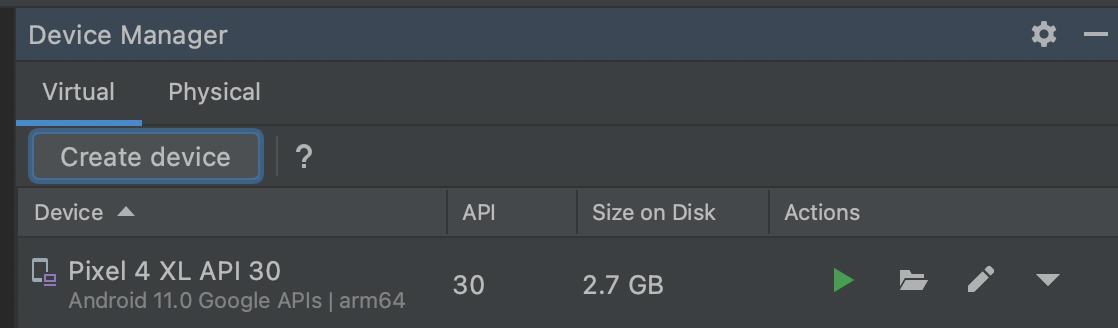


Рисунок 10 - выбираем create device

В появившемся окне (Рисунок 15) выбираем понравившийся девайс нажимает кнопку next. Далее выбираем понравившуюся версию андроида (Рисунок 16) и нажимаем next и finish.

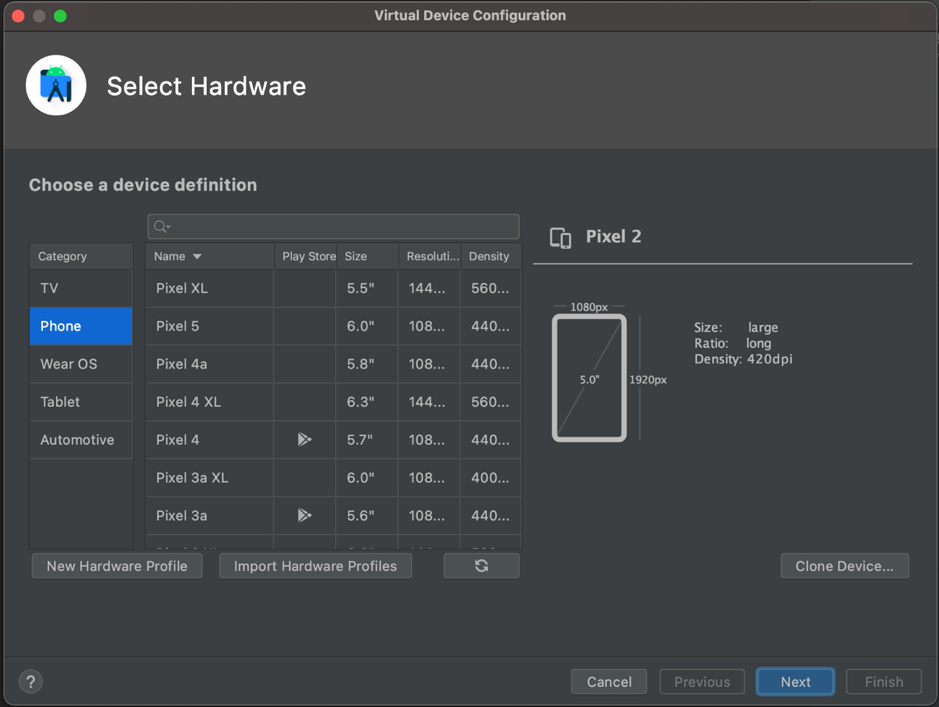


Рисунок 11 - появившееся окно

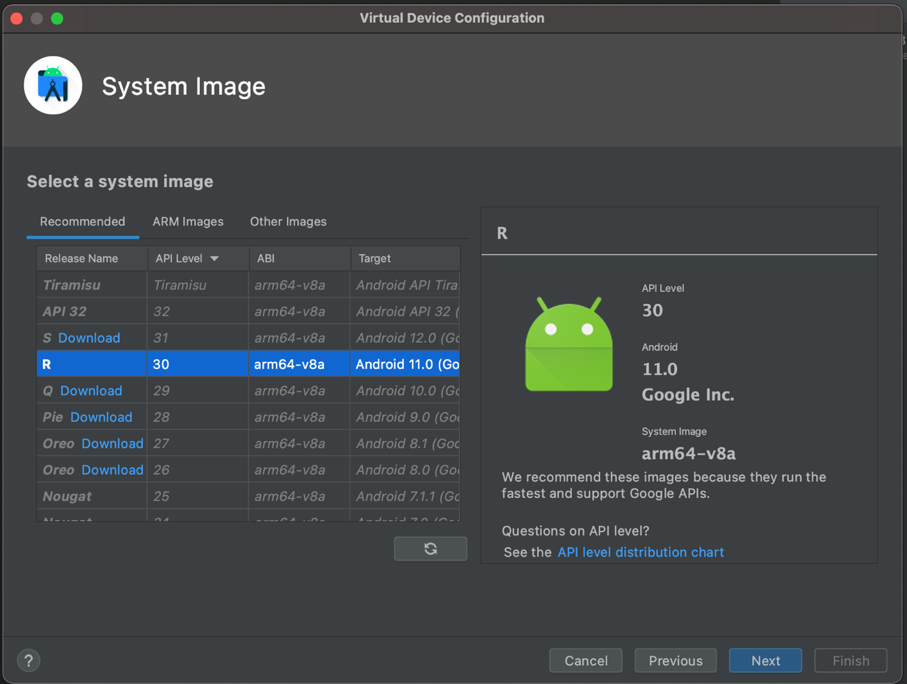


Рисунок 12 - версия андроида

Телефончик появился в нашем списке и теперь можно нажимаю на кнопку пуск и наслаждаться нашим приложением (рисунок 17).



Рисунок 13 - кнопка пуск

Еще один вариант — это подключить телефон по проводу к компьютеру, для этого надо подключить смартфон к компьютеру с помощью кабеля USB.

Включить Отладка через USB на вашем устройстве.

На Android 4.0 и новее, это находится в Настройки> Для разработчиков.

Примечание: на Android 4.2 и новее, «Для разработчиков» по умолчанию скрыт. Чтобы сделать его доступным, перейдите к Настройки> О телефоне и нажмите Номер сборки семь раз. Вернитесь к предыдущему экрану, чтобы найти «Для разработчиков». Откройте один из файлов вашего проекта и нажмите Run apprun на панели инструментов. Или нажмите комбинацию клавиш Shift+F10.

Откроется окно выбора подключенного реального устройства или эмулятора. Android Studio устанавливает приложение на выбранное устройство и запускает его.

Заключение

В этом проекте была задача разработать приложение, проанализировать доступных соперников, сделав это, приложение должно быть не хуже, чем у конкурентов. Дальше нужно было описать техническое задание и этапы разработки приложения. В данном проекте были рассмотрены три среды для разработки приложений, выбрана наиболее лучшая из них. Также рассмотренны этапы разработки индивидуального приложения от разработки до тестирования. Стоит отметить, что процессы разработки и тестирования до того, как приложение запущено в эксплуатацию могут длиться бесконечно долго, до тех пор, пока не будет определено, что дальнейшая разработка не требуется и не требуется дальнейшее тестирование, необходимые ошибки устранены. Однако чаще всего во время эксплуатации приложения пользователями выявляются ошибки, которые не были замечены во время тестирования.

Также рассмотрены платформы для приложения, разработан собственный интерфейс и создано приложение по техническому заданию.

Таким образом, очевидно, что процесс проектирования приложения и его ввод в эксплуатацию представляет собой длительный и сложный процесс.

В работе были выполнены все поставленные во введении задачи, что привело к достижению поставленной цели.

Список использованной литературы

1. <https://developer.android.com/> документация андроида
2. <https://www.fandroid.info/zapusk-vashego-prilozheniya/> Запуск android-приложения
3. <https://zaochnik.ru/blog/standarty-oformlenija-kursovoj-raboty-po-gostu/> госты
4. <https://www.rosdiplom.ru/rd/pubdiplom/view.aspx?id=287> госты
5. <https://play.google.com/store/apps/details?id=net.incona.greenstream.photoreport> фотоотчет мерчандайзер, приложение
6. <https://play.google.com/store/apps/details?id=ru.rassvet.merchant> Merchhub приложение
7. <https://support.google.com/googleplay/answer/113409?hl=ru> Установка приложения
8. <https://github.com/Makssshow/MerchTriB> Репозиторий с приложением
9. <https://vertex-academy.com/tutorials/ru/samouchitel-po-java-s-nulya/> Учебник по Java
10. <https://www.jetbrains.com/objc/> AppCode программа

Приложение А. Листинг рабочего кода

AddTaskActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.objects.Task;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
public class AddTaskActivity extends AppCompatActivity {  
  
 EditText name, address, addressLink;  
 Button addButton;  
  
 String companyName;  
  
 private FirebaseDatabase mFirebaseDatabase;  
 private DatabaseReference mDatabaseReference;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_add\_task);  
  
 mFirebaseDatabase = FirebaseDatabase.getInstance();  
 //получаем ссылку для работы с базой данных  
 companyName = getSharedPreferences("data", MODE\_PRIVATE).getString("companyName", "");  
 mDatabaseReference = mFirebaseDatabase.getReference("companies/" + companyName + "/tasks/current");  
  
  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 if (toolbar != null) {  
 toolbar.setTitle("Добавление задания");  
 setSupportActionBar(toolbar);  
 }  
  
 name = findViewById(R.id.name);  
 address = findViewById(R.id.address);  
 addressLink = findViewById(R.id.addressLink);  
 addButton = findViewById(R.id.addButton);  
  
 }  
  
  
 public void addButtonClickHandler(View view) {  
 // Обрабатываем нажатие на кнопку  
 String nameFinal = name.getText().toString();  
 String id = nameFinal + System.currentTimeMillis();  
 String addressFinal = address.getText().toString();  
 String addressLinkFinal = addressLink.getText().toString();  
  
 if (!nameFinal.isEmpty() && !addressFinal.isEmpty()) {  
 if (!addressLinkFinal.isEmpty()) {  
 mDatabaseReference.child(id).setValue(new Task(id, nameFinal, addressFinal, addressLinkFinal, null));  
 } else {  
 mDatabaseReference.child(id).setValue(new Task(id, nameFinal, addressFinal, null, null));  
 }  
 finish();  
 }  
  
 }  
}

AdminActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.fragments.MainFragment;  
import com.example.merchtrib.ui.fragments.SendTodayFragment;  
import com.example.merchtrib.ui.fragments.UsersFragment;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class AdminActivity extends AppCompatActivity {  
  
 public FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();  
 MainFragment mainFragment = new MainFragment();  
 SendTodayFragment oldFragment = new SendTodayFragment();  
 UsersFragment usersFragment = new UsersFragment();  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_admin);  
  
 if (getSupportFragmentManager().findFragmentById(R.id.fragment\_container\_main) == null) {  
 // start new Activity  
 loadFragment(mainFragment);  
 }  
 BottomNavigationView bottomNavigationView = findViewById(R.id.bottom\_navigation);  
 bottomNavigationView.inflateMenu(R.menu.admin\_bottom\_navigation);  
 bottomNavigationView.setSelectedItemId(R.id.tasks);  
 bottomNavigationView.setOnNavigationItemSelectedListener(item -> {  
 switch (item.getItemId()) {  
 case R.id.tasks:  
 loadFragment(mainFragment);  
 return true;  
 case R.id.users:  
 loadFragment(usersFragment);  
 return true;  
 case R.id.tasks\_old:  
 loadFragment(oldFragment);  
 return true;  
 }  
 return false;  
 });  
  
  
 }  
  
  
 public void loadFragment(Fragment fragment) {  
 FragmentTransaction ft = getSupportFragmentManager().beginTransaction();  
 ft.replace(R.id.fragment\_container\_main, fragment);  
 ft.commit();  
 }  
  
 @Override  
 protected void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 }  
  
  
 @Override  
 public void onBackPressed() {  
 super.onBackPressed();  
 }  
  
}

LoginActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.objects.User;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.regex.Pattern;  
  
public class LoginActivity extends AppCompatActivity {  
  
 EditText email, password;  
 Button loginButton;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_login);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 if (toolbar != null) {  
 toolbar.setTitle("Мерчандайзер");  
 setSupportActionBar(toolbar);  
  
 }  
  
 email = findViewById(R.id.login);  
 password = findViewById(R.id.password);  
 loginButton = findViewById(R.id.loginButton);  
 FirebaseAuth mAuth = FirebaseAuth.getInstance();  
// Нажатие на кнопку регистрации  
 loginButton.setOnClickListener(v -> {  
 if (!email.getText().toString().isEmpty() && !password.getText().toString().isEmpty()) {  
 // если поля не пустые  
 if (isValid(email.getText().toString())) {  
 // Если почта верна  
 if (password.getText().toString().length() >= 6) {  
 //Если пароль больше 6 символов  
 loginButton.setClickable(false);  
 Toast.makeText(getApplicationContext(), "Проверяем...", Toast.LENGTH\_SHORT).show();  
 //Войти  
 mAuth.signInWithEmailAndPassword(email.getText().toString(), password.getText().toString()).addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 startActivity(new Intent(getApplicationContext(), MainActivity.class));  
 finish();  
 } else {  
 //Зарегистрировать  
 mAuth.createUserWithEmailAndPassword(email.getText().toString(), password.getText().toString()).addOnCompleteListener(task2 -> {  
 if (task2.isSuccessful()) {  
 FirebaseUser user = mAuth.getCurrentUser();  
 if (user != null) {  
 FirebaseDatabase.getInstance().getReference("users").child(user.getEmail().replace("@", "").replace(".", "").toLowerCase()).child("company").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.getValue(String.class) == null) {  
 FirebaseDatabase.getInstance().getReference("users").child(user.getEmail().replace("@", "").replace(".", "")).setValue(new User(null, true));  
 }  
 startActivity(new Intent(getApplicationContext(), MainActivity.class));  
 finish();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
  
 }  
 } else {  
 Toast.makeText(getApplicationContext(), "Что-то пошло не так: " + task.getException() + " \n " + task2.getException(), Toast.LENGTH\_LONG).show();  
 email.setText("");  
 password.setText("");  
 }  
 });  
 }  
 });  
  
 loginButton.setClickable(true);  
 } else {  
 Toast.makeText(this, "Пароль должен быть больше 6 символов", Toast.LENGTH\_LONG).show();  
 password.setText("");  
 }  
 } else {  
 Toast.makeText(this, "E-mail не вырный", Toast.LENGTH\_LONG).show();  
 email.setText("");  
 password.setText("");  
 }  
 } else {  
 Toast.makeText(this, "Не все поля заполнены", Toast.LENGTH\_LONG).show();  
 }  
  
 });  
 }  
  
 public static boolean isValid(String email) {  
 String emailRegex = "^[a-zA-Z0-9\_+&\*-]+(?:\\." +  
 "[a-zA-Z0-9\_+&\*-]+)\*@" +  
 "(?:[a-zA-Z0-9-]+\\.)+[a-z" +  
 "A-Z]{2,7}$";  
  
 Pattern pat = Pattern.compile(emailRegex);  
 if (email == null)  
 return false;  
 return pat.matcher(email).matches();  
 }  
  
}

MainActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentTransaction;  
  
  
import android.app.Activity;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Toast;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.activities.AdminActivity;  
import com.example.merchtrib.ui.activities.RegisterCompanyActivity;  
import com.example.merchtrib.ui.activities.LoginActivity;  
import com.example.merchtrib.ui.fragments.MainFragment;  
import com.example.merchtrib.ui.fragments.SendTodayFragment;  
import com.example.merchtrib.ui.objects.User;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
public class MainActivity extends AppCompatActivity {  
  
 public static Activity BackActivity;  
  
 Fragment frag\_main = new MainFragment();  
 Fragment frag\_send\_today = new SendTodayFragment();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();  
 if (user == null) {  
 startActivity(new Intent(this, LoginActivity.class));  
 finish();  
 } else {  
 // Проверить пользователя  
 FirebaseDatabase.getInstance().getReference("users").child(user.getEmail().replace("@", "").replace(".", "").toLowerCase()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.getValue(User.class) != null) {  
  
 User userData = dataSnapshot.getValue(User.class);  
 boolean isCompany = userData.company != null;  
 if (isCompany) {  
 SharedPreferences preferences = getSharedPreferences("data", MODE\_PRIVATE);  
 preferences.edit().putString("companyName", userData.company).putString("userEmailShort", user.getEmail().replace("@", "").replace(".", "").toLowerCase()).putBoolean("isAdmin", userData.admin).apply();  
 FirebaseDatabase.getInstance().getReference("companies/" + userData.company + "/name").addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.getValue(String.class) != null) {  
 preferences.edit().putString("companyNameOriginal", dataSnapshot.getValue(String.class)).apply();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.makeText(getApplicationContext(), "Что-то пошло не так: " + databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
 }  
 if (userData.admin) {  
 if (isCompany) {  
 startActivity(new Intent(getApplicationContext(), AdminActivity.class));  
 } else {  
 startActivity(new Intent(getApplicationContext(), RegisterCompanyActivity.class));  
 }  
 finish();  
 } else {  
 updateUI();  
 }  
 } else {  
 startActivity(new Intent(getApplicationContext(), LoginActivity.class));  
 finish();  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.makeText(getApplicationContext(), "Что-то пошло не так: " + databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
  
 }  
  
  
 }  
  
 private void updateUI() {  
 findViewById(R.id.progress\_bar).setVisibility(View.INVISIBLE);  
 BackActivity = this;  
 if (getSupportFragmentManager().findFragmentById(R.id.fragment\_container\_main) == null) {  
 // start new Activity  
 loadFragment(frag\_main);  
 }  
 BottomNavigationView bottomNavigationView = findViewById(R.id.bottom\_navigation);  
 bottomNavigationView.inflateMenu(R.menu.menu\_bottom\_navigation);  
 bottomNavigationView.setSelectedItemId(R.id.main);  
 bottomNavigationView.setOnNavigationItemSelectedListener(item -> {  
 switch (item.getItemId()) {  
 case R.id.send\_today:  
 loadFragment(frag\_send\_today);  
 return true;  
 case R.id.main:  
 loadFragment(frag\_main);  
 return true;  
 }  
 return false;  
 });  
 }  
  
  
 public void loadFragment(Fragment fragment) {  
 FragmentTransaction ft = getSupportFragmentManager().beginTransaction();  
 ft.replace(R.id.fragment\_container\_main, fragment);  
 ft.commit();  
 }  
  
 @Override  
 protected void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 }  
  
  
 @Override  
 public void onBackPressed() {  
 super.onBackPressed();  
 }  
  
  
}

RegisterCompanyActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.objects.Company;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.HashMap;  
  
public class RegisterCompanyActivity extends AppCompatActivity {  
EditText name;  
Button submit;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_register\_company);  
  
 name = findViewById(R.id.company);  
 submit = findViewById(R.id.submit);  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 if (toolbar != null) {  
 toolbar.setTitle("Регистрация компании");  
 setSupportActionBar(toolbar);  
  
 }  
  
 //Зарегистрировать компанию  
 submit.setOnClickListener(v -> {  
 submit.setClickable(false);  
 String data = name.getText().toString();  
 if (!data.isEmpty() && data.length() >= 2) {  
 String email = FirebaseAuth.getInstance().getCurrentUser().getEmail().replace("@", "").replace(".", "").toLowerCase();  
 FirebaseDatabase mBase = FirebaseDatabase.getInstance();  
 HashMap user = new HashMap();  
 String result = transliterate(data);  
 String companyName = result + System.currentTimeMillis();  
 user.put("company", companyName);  
  
 mBase.getReference("users").child(email).updateChildren(user).addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 mBase.getReference("companies").child(companyName).setValue(new Company(data, null, null)).addOnCompleteListener(t -> {  
 finish();  
 });  
 Intent intent = new Intent(getApplicationContext(), AdminActivity.class);  
 startActivity(intent);  
  
 } else {  
 Toast.makeText(getApplicationContext(), "Что-то пошло не так: " + task.getException(), Toast.LENGTH\_LONG).show();  
 submit.setClickable(true);  
 }  
 });  
 } else {  
 Toast.makeText(getApplicationContext(), "Название должно быть больше двух букв", Toast.LENGTH\_LONG).show();  
 submit.setClickable(true);  
 }  
 });  
 }  
  
  
 public static String transliterate(String message){  
 char[] abcCyr = {' ','а','б','в','г','д','е','ё', 'ж','з','и','й','к','л','м','н','о','п','р','с','т','у','ф','х', 'ц','ч', 'ш','щ','ъ','ы','ь','э', 'ю','я','А','Б','В','Г','Д','Е','Ё', 'Ж','З','И','Й','К','Л','М','Н','О','П','Р','С','Т','У','Ф','Х', 'Ц', 'Ч','Ш', 'Щ','Ъ','Ы','Ь','Э','Ю','Я','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};  
 String[] abcLat = {"","a","b","v","g","d","e","e","zh","z","i","y","k","l","m","n","o","p","r","s","t","u","f","h","ts","ch","sh","sch", "","i", "","e","ju","ja","A","B","V","G","D","E","E","Zh","Z","I","Y","K","L","M","N","O","P","R","S","T","U","F","H","Ts","Ch","Sh","Sch", "","I", "","E","Ju","Ja","a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z","A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"};  
 StringBuilder builder = new StringBuilder();  
 for (int i = 0; i < message.length(); i++) {  
 for (int x = 0; x < abcCyr.length; x++ ) {  
 if (message.charAt(i) == abcCyr[x]) {  
 builder.append(abcLat[x]);  
 }  
 }  
 }  
 return builder.toString();  
 }  
}

SendingActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.ActionMode;  
import android.view.MenuItem;  
  
import com.example.merchtrib.R;  
  
public class SendingActivity extends AppCompatActivity {  
  
 public static Activity activity;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_sending);  
 activity = this;  
  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 if (toolbar != null) {  
 toolbar.setTitle(R.string.wait);  
 setSupportActionBar(toolbar);  
 }  
  
  
 }  
  
}

TaskActivity.java

package com.example.merchtrib.ui.activities;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.constraintlayout.widget.ConstraintLayout;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.Manifest;  
import android.content.ClipData;  
import android.content.ContentResolver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Handler;  
import android.provider.MediaStore;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.webkit.MimeTypeMap;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.adapters.GalleryAdapter;  
import com.example.merchtrib.ui.adapters.TasksAdapter;  
import com.example.merchtrib.ui.objects.Task;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.OnProgressListener;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.google.gson.Gson;  
import com.google.gson.reflect.TypeToken;  
import com.gun0912.tedpermission.PermissionListener;  
import com.gun0912.tedpermission.TedPermission;  
  
import java.lang.reflect.Type;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class TaskActivity extends AppCompatActivity implements GalleryAdapter.OnImageListener {  
  
 private RecyclerView gvGallery;  
 private GalleryAdapter galleryAdapter;  
 private FloatingActionButton addPhotoButton;  
  
 public ArrayList<Uri> mArrayUri = new ArrayList<>();  
 public ArrayList<String> mArrayUriDone = new ArrayList<>();  
 public ArrayList<Uri> mArrayUriAbsolute = new ArrayList<>();  
 public ArrayList<Uri> mArrayLocal = new ArrayList<>();  
 int PICK\_IMAGE\_MULTIPLE = 1;  
  
 private StorageReference mStorageRef;  
 private DatabaseReference mDatabaseRef;  
  
 private ProgressBar progressBar;  
  
 GalleryAdapter.OnImageListener context;  
 Context origContext;  
  
 String title = "Ошибка",  
 idTask = "",  
 type = "current",  
 companyName, userEmail;  
TextView emptyText;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_task);  
 context = this;  
 origContext = this;  
  
 emptyText = findViewById(R.id.emptyText);  
 Intent intent = getIntent();  
 title = intent.getStringExtra("name");  
 idTask = intent.getStringExtra("id");  
 type = intent.getStringExtra("type") == null ? "current" : intent.getStringExtra("type");  
 SharedPreferences preferences = getSharedPreferences("data", MODE\_PRIVATE);  
 companyName = preferences.getString("companyName", "");  
 userEmail = preferences.getString("userEmailShort", "");  
  
 Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
 if (toolbar != null) {  
 toolbar.setTitle(title);  
 setSupportActionBar(toolbar);  
 }  
  
 mStorageRef = FirebaseStorage.getInstance().getReference("uploads/" + idTask);  
 mDatabaseRef = FirebaseDatabase.getInstance().getReference("companies/" + companyName + "/tasks");  
  
 progressBar = findViewById(R.id.progress\_circular);  
 gvGallery = findViewById(R.id.rev\_photo);  
 gvGallery.setLayoutManager(new GridLayoutManager(this, 3));  
 addPhotoButton = findViewById(R.id.floatingActionButton);  
  
 mDatabaseRef.child(type).child(idTask).child("images").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 mArrayUriAbsolute = new ArrayList<>();  
 // Get Post object and use the values to update the UI  
 for (DataSnapshot dataUri : dataSnapshot.getChildren()) {  
 mArrayUriAbsolute.add(Uri.parse(dataUri.getValue(String.class)));  
 }  
 mArrayUri.clear();  
 mArrayUri.addAll(mArrayUriAbsolute);  
 mArrayUri.addAll(mArrayLocal);  
 gvGallery.setAdapter(new GalleryAdapter(origContext, mArrayUri, context));  
 progressBar.setVisibility(View.INVISIBLE);  
 if (mArrayUri.isEmpty()) {  
 emptyText.setVisibility(View.VISIBLE);  
 } else {  
 emptyText.setVisibility(View.INVISIBLE);  
 }  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 // Getting Post failed, log a message  
 Toast.makeText(getApplicationContext(), databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
  
 addPhotoButton.setOnClickListener(view -> requestPermissions(0));  
  
 }  
  
 private void requestPermissions(int i) {  
 PermissionListener permissionlistener = new PermissionListener() {  
 @Override  
 public void onPermissionGranted() {  
 if (i == 0) {  
 selectImagesFromGallery();  
 } else if (i == 1) {  
 addImagesInRecycler(mArrayUri);  
 }  
  
 }  
  
 @Override  
 public void onPermissionDenied(List<String> deniedPermissions) {  
 Toast.makeText(TaskActivity.this, "Permission Denied\n" + deniedPermissions.toString(), Toast.LENGTH\_SHORT).show();  
 }  
  
  
 };  
 TedPermission.with(this)  
 .setPermissionListener(permissionlistener)  
 .setDeniedMessage("Если вы не дадите разрешения, то не сможете пользоваться этим приложением\n\nПожалуйста, дайте разрешения в [Настройки] > [Приложения]")  
 .setPermissions(Manifest.permission.WRITE\_EXTERNAL\_STORAGE, Manifest.permission.READ\_EXTERNAL\_STORAGE, Manifest.permission.CAMERA)  
 .check();  
 }  
  
 private void selectImagesFromGallery() {  
 try {  
  
 // initialising intent  
 Intent intent = new Intent();  
  
 // setting type to select to be image  
 intent.setType("image/\*");  
  
 // allowing multiple image to be selected  
 intent.putExtra(Intent.EXTRA\_ALLOW\_MULTIPLE, true);  
 intent.setAction(Intent.ACTION\_GET\_CONTENT);  
 startActivityForResult(Intent.createChooser(intent, "Выберите фотографии"), PICK\_IMAGE\_MULTIPLE);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 // When an Image is picked  
 if (requestCode == PICK\_IMAGE\_MULTIPLE && resultCode == RESULT\_OK && null != data) {  
 // Get the Image from data  
  
 if (data.getClipData() != null) {  
 ClipData mClipData = data.getClipData();  
 int cout = data.getClipData().getItemCount();  
// mArrayUri = (ArrayList<Uri>) mArrayUriAbsolute.clone();  
 for (int i = 0; i < cout; i++) {  
 Uri imageurl = data.getClipData().getItemAt(i).getUri();  
 mArrayLocal.add(imageurl);  
 }  
 mArrayUri.clear();  
 mArrayUri.addAll(mArrayUriAbsolute);  
 mArrayUri.addAll(mArrayLocal);  
  
 addImagesInRecycler(mArrayUri);  
  
 } else if (data.getData() != null) {  
 mArrayLocal.add(data.getData());  
 mArrayUri.clear();  
 mArrayUri.addAll(mArrayUriAbsolute);  
 mArrayUri.addAll(mArrayLocal);  
 addImagesInRecycler(mArrayUri);  
 } else {  
 Toast.makeText(this, "Вы не выбрали ни одной картинки, ошибка", Toast.LENGTH\_LONG).show();  
 }  
 } else {  
 // show this if no image is selected  
 Toast.makeText(this, "Вы не выбрали ни одной картинки", Toast.LENGTH\_LONG).show();  
 }  
 }  
  
 private void addImagesInRecycler(ArrayList<Uri> mArrayUri) {  
 GalleryAdapter galleryAdapter = new GalleryAdapter(origContext, mArrayUri, this);  
  
 gvGallery.setAdapter(galleryAdapter);  
  
 }  
  
 @Override  
 public void onImageClick(int position) {  
 if (position < mArrayUriAbsolute.size()) {  
 if (mArrayUri.get(position) == mArrayUriAbsolute.get(position)) {  
 mDatabaseRef.child(type).child(idTask).child("images").child(String.valueOf(position)).removeValue();  
 }  
 } else {  
 if (mArrayUri.get(position) == mArrayLocal.get(position - mArrayUriAbsolute.size())) {  
 mArrayLocal.remove(position - mArrayUriAbsolute.size());  
 }  
 mArrayUri.clear();  
 mArrayUri.addAll(mArrayUriAbsolute);  
 mArrayUri.addAll(mArrayLocal);  
 addImagesInRecycler(mArrayUri);  
 }  
 }  
  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.menu\_task, menu);  
 return super.onCreateOptionsMenu(menu);  
 }  
  
  
 // Заглушка, работа с меню  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 if (id == R.id.menu\_task\_clear) {  
 // Toast.makeText(getApplicationContext(), "Добавить примечание", Toast.LENGTH\_SHORT).show();  
// startActivity(new Intent(this, SendingActivity.class));  
 mArrayUri.clear();  
 mArrayLocal.clear();  
 for (Uri imageUri : mArrayUriAbsolute) {  
 FirebaseStorage.getInstance().getReferenceFromUrl(String.valueOf(imageUri)).delete();  
 }  
 mArrayUriAbsolute = new ArrayList<>();  
 mDatabaseRef.child(type).child(idTask).child("images").removeValue();  
 addImagesInRecycler(mArrayUri);  
 } else if (id == R.id.menu\_task\_send) {  
 mArrayUriDone.clear();  
 Intent intentSending = new Intent(this, SendingActivity.class);  
 startActivity(intentSending);  
 for (int i = mArrayUriAbsolute.size(); i < mArrayUri.size(); i++) {  
 uploadFiles(mArrayUri.get(i));  
 }  
 }  
  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 private String getFileExtension(Uri uri) {  
 ContentResolver cR = getContentResolver();  
 MimeTypeMap mime = MimeTypeMap.getSingleton();  
 return mime.getExtensionFromMimeType(cR.getType(uri));  
 }  
  
 private void uploadFiles(Uri uri) {  
 if (mArrayUri != null) {  
 StorageReference fileReference = mStorageRef.child(System.currentTimeMillis() + "." + getFileExtension(uri));  
  
  
 fileReference.putFile(uri)  
 .addOnSuccessListener(taskSnapshot -> {  
 fileReference.getDownloadUrl().addOnSuccessListener(uriAbsolute -> {  
 mArrayUriDone.add(uriAbsolute.toString());  
 mArrayUriAbsolute.add(uriAbsolute);  
 if (mArrayUriAbsolute.size() == mArrayUri.size()) {  
 ArrayList<String> data = new ArrayList<>();  
 for (int i = 0; i < mArrayUriAbsolute.size(); i++) {  
 data.add(String.valueOf(mArrayUriAbsolute.get(i)));  
 }  
 mDatabaseRef.child(type).child(idTask).child("images").setValue(data);  
 if (type.equals("current")) {  
 mDatabaseRef.child(type).child(idTask).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
  
 mDatabaseRef.child("done").child(idTask).setValue(dataSnapshot.getValue());  
 mDatabaseRef.child(type).child(idTask).removeValue();  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.makeText(getApplicationContext(), "Что-то пошло не так: " + databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
 }  
  
 SendingActivity.activity.finish();  
 Toast.makeText(this, "Успешно отправлено", Toast.LENGTH\_LONG).show();  
 this.finish();  
  
 }  
 });  
  
 })  
 .addOnFailureListener(e -> {  
 Toast.makeText(this, "Что-то пошло не так: " + e.getMessage(), Toast.LENGTH\_LONG).show();  
 mArrayUri = new ArrayList<>();  
 addImagesInRecycler(mArrayUri);  
 });  
 } else {  
 Toast.makeText(this, "Файл не выбран", Toast.LENGTH\_LONG).show();  
 }  
 }  
  
}

GalaryAdapter.java

package com.example.merchtrib.ui.adapters;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.net.Uri;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.merchtrib.R;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.storage.FirebaseStorage;  
import com.makeramen.roundedimageview.RoundedImageView;  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
public class GalleryAdapter extends RecyclerView.Adapter<GalleryAdapter.PhotoViewHolder> {  
  
 private Context ctx;  
 private RoundedImageView ivGallery;  
 ArrayList<Uri> mArrayUri;  
 private OnImageListener mOnImageListener;  
  
 public GalleryAdapter(Context ctx, ArrayList<Uri> mArrayUri, OnImageListener mmOnImageListener) {  
 this.ctx = ctx;  
 this.mArrayUri = mArrayUri;  
 this.mOnImageListener = mmOnImageListener;  
 }  
  
  
 @NonNull  
 @Override  
 public PhotoViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.image\_task\_item, parent, false);  
 return new PhotoViewHolder(itemView, mOnImageListener);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull GalleryAdapter.PhotoViewHolder holder, int position) {  
// ivGallery.setImageURI(Uri.parse(mArrayUri.get(position)));  
 Picasso.get().load(mArrayUri.get(position)).error(R.drawable.img\_error).into(ivGallery);  
 ivGallery.setOnLongClickListener(v -> {  
 AlertDialog.Builder builder = new AlertDialog.Builder(ctx);  
 builder.setTitle("Удаление")  
 .setMessage("Вы уверены что хотите удалить это фото?")  
 .setPositiveButton("Да", (dialog, id) -> {  
 // Закрываем окно  
 mOnImageListener.onImageClick(position);  
 dialog.cancel();  
 }).setNegativeButton("Нет", (dialog, id) -> {  
 dialog.cancel();  
 });  
 builder.create().show();  
 return false;  
 });  
 }  
  
  
  
 @Override  
 public int getItemCount() {  
 return mArrayUri.size();  
 }  
  
  
 public class PhotoViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
 OnImageListener mmOnImageListener;  
 public PhotoViewHolder(@NonNull View itemView, OnImageListener OnImageListener) {  
 super(itemView);  
 ivGallery = (RoundedImageView) itemView.findViewById(R.id.photo);  
 mmOnImageListener = OnImageListener;  
  
// ivGallery.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View view) {  
 mOnImageListener.onImageClick(getAdapterPosition());  
 }  
 }  
  
 public interface OnImageListener{  
 void onImageClick(int position);  
 }  
}

TaskAdapter.java

package com.example.merchtrib.ui.adapters;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.net.Uri;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.activities.TaskActivity;  
import com.example.merchtrib.ui.objects.Task;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.makeramen.roundedimageview.RoundedImageView;  
import com.squareup.picasso.Picasso;  
  
import org.w3c.dom.Text;  
  
import java.util.ArrayList;  
  
public class TasksAdapter extends RecyclerView.Adapter<TasksAdapter.TaskViewHolder> {  
  
 private Context ctx;  
 private ArrayList<Task> mArrayTask;  
 private ImageButton imageButton;  
 private TextView name;  
 private TextView address;  
 private LinearLayout parentLayout;  
 private String type;  
 private boolean isAdmin;  
 private String companyName;  
  
 public TasksAdapter(Context ctx, ArrayList<Task> task, String type, boolean isAdmin, String companyName) {  
 this.ctx = ctx;  
 this.mArrayTask = task;  
 this.type = type;  
 this.isAdmin = isAdmin;  
 this.companyName = companyName;  
  
 }  
  
  
 @NonNull  
 @Override  
 public TaskViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.main\_task\_list\_item, parent, false);  
  
 return new TaskViewHolder(itemView);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull TasksAdapter.TaskViewHolder holder, int position) {  
 Task current = mArrayTask.get(position);  
 name.setText(current.getName());  
 address.setText(current.getAddress());  
 if (current.getAddressLink() != null) {  
 imageButton.setOnClickListener(view -> {  
 Uri uri = Uri.parse(current.getAddressLink());  
 Intent intent = new Intent(Intent.ACTION\_VIEW, uri);  
 ctx.startActivity(intent);  
 });  
  
 } else {  
 imageButton.setVisibility(View.INVISIBLE);  
 }  
  
 parentLayout.setOnClickListener(v -> {  
 Intent intent = new Intent(ctx, TaskActivity.class);  
 intent.putExtra("type", type);  
 intent.putExtra("id", current.id);  
 intent.putExtra("name", current.name);  
 intent.putExtra("address", current.address);  
 intent.putExtra("addressLink", current.addressLink);  
 ctx.startActivity(intent);  
 });  
 if (isAdmin) {  
 parentLayout.setOnLongClickListener(v -> {  
 AlertDialog.Builder builder = new AlertDialog.Builder(ctx);  
 builder.setTitle("Удаление")  
 .setMessage("Вы уверены что хотите удалить это задание?")  
 .setPositiveButton("Да", (dialog, id) -> {  
 // Закрываем окно  
 FirebaseDatabase.getInstance().getReference("companies/" + companyName + "/tasks/" + type + "/" + current.id).removeValue();  
 FirebaseStorage.getInstance().getReference("uploads/" + current.id).delete();  
 dialog.cancel();  
 }).setNegativeButton("Отмена", (dialog, id) -> {  
 dialog.cancel();  
 });  
 builder.create().show();  
 return false;  
 });  
 }  
 }  
  
  
 @Override  
 public int getItemCount() {  
 return mArrayTask.size();  
 }  
  
  
 public class TaskViewHolder extends RecyclerView.ViewHolder {  
 public TaskViewHolder(@NonNull View itemView) {  
 super(itemView);  
 imageButton = itemView.findViewById(R.id.task\_item\_button);  
 name = itemView.findViewById(R.id.list\_item\_name);  
 address = itemView.findViewById(R.id.list\_item\_link);  
 parentLayout = itemView.findViewById(R.id.parent);  
  
 }  
  
 }  
}

UsersAdapter.java

package com.example.merchtrib.ui.adapters;  
  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.graphics.Color;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageButton;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.core.content.ContextCompat;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.merchtrib.R;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.collection.LLRBNode;  
import com.google.firebase.storage.FirebaseStorage;  
  
import java.util.ArrayList;  
  
  
public class UsersAdapter extends RecyclerView.Adapter<UsersAdapter.UsersViewHolder> {  
  
 private Context ctx;  
 private ArrayList<String> mData;  
 private ImageButton delete;  
 private TextView name;  
 private String companyName;  
  
 public UsersAdapter(Context ctx, ArrayList<String> data, String companyName) {  
 this.ctx = ctx;  
 this.mData = data;  
 this.companyName = companyName;  
  
 }  
  
  
 @NonNull  
 @Override  
 public UsersViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.user\_item, parent, false);  
  
 return new UsersViewHolder(itemView);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull UsersAdapter.UsersViewHolder holder, int position) {  
 String current = mData.get(position);  
 name.setText(current);  
 delete.setOnClickListener(v -> {  
 AlertDialog.Builder builder = new AlertDialog.Builder(ctx);  
 AlertDialog dialog = builder.setTitle("Удаление")  
 .setMessage("Вы уверены что хотите удалить этого пользователя?")  
 .setPositiveButton("Да", (dialogg, id) -> {  
 // Закрываем окно  
 String email = current.replace("@", "").replace(".", "").toLowerCase();  
 FirebaseDatabase mDataBase = FirebaseDatabase.getInstance();  
 mDataBase.getReference("companies/" + companyName + "/users/" + email).removeValue();  
 mDataBase.getReference("users/" + email).removeValue();  
 dialogg.cancel();  
 }).setNegativeButton("Отмена", (dialogg, id) -> {  
 dialogg.cancel();  
 }).create();  
 dialog.setOnShowListener(arg0 -> {  
 dialog.getButton(AlertDialog.BUTTON\_POSITIVE).setTextColor(ContextCompat.getColor(ctx, R.color.primary));  
 });  
 builder.show();  
 });  
 }  
  
  
 @Override  
 public int getItemCount() {  
 return mData.size();  
 }  
  
  
 public class UsersViewHolder extends RecyclerView.ViewHolder {  
 public UsersViewHolder(@NonNull View itemView) {  
 super(itemView);  
 name = itemView.findViewById(R.id.list\_item\_name);  
 delete = itemView.findViewById(R.id.delete);  
  
 }  
  
 }  
}

MainFragment.java

package com.example.merchtrib.ui.fragments;  
  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ImageButton;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.activities.AddTaskActivity;  
import com.example.merchtrib.ui.activities.LoginActivity;  
import com.example.merchtrib.ui.activities.MainActivity;  
import com.example.merchtrib.ui.activities.TaskActivity;  
import com.example.merchtrib.ui.adapters.TasksAdapter;  
import com.example.merchtrib.ui.objects.Task;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.GenericTypeIndicator;  
import com.google.firebase.database.ValueEventListener;  
  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
import java.util.Locale;  
import java.util.Objects;  
  
public class MainFragment extends Fragment {  
  
 public ArrayList<Task> TASKS = new ArrayList<>();  
 private DatabaseReference mDatabase;  
 SharedPreferences sharedPreferences;  
 String userEmail, companyName, companyNameOriginal;  
 boolean isAdmin;  
 private FirebaseUser user;  
 RecyclerView lv;  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.fragment\_main, container, false);  
  
// user = FirebaseAuth.getInstance().getCurrentUser();  
 sharedPreferences = getActivity().getSharedPreferences("data", Context.MODE\_PRIVATE);  
 userEmail = sharedPreferences.getString("userEmailShort", "");  
 companyName = sharedPreferences.getString("companyName", "");  
 isAdmin = sharedPreferences.getBoolean("isAdmin", false);  
 companyNameOriginal = sharedPreferences.getString("companyNameOriginal", null);  
  
 lv = v.findViewById(R.id.list\_of\_tasks);  
 LinearLayoutManager llm = new LinearLayoutManager(this.getContext());  
 llm.setOrientation(LinearLayoutManager.VERTICAL);  
 DividerItemDecoration dividerItemDecoration = new DividerItemDecoration(lv.getContext(),  
 llm.getOrientation());  
 lv.addItemDecoration(dividerItemDecoration);  
 lv.setLayoutManager(llm);  
  
 TextView emptyText = v.findViewById(R.id.emptyText);  
  
 mDatabase = FirebaseDatabase.getInstance().getReference();  
  
 ProgressBar progressBar = v.findViewById(R.id.progress\_circular);  
 mDatabase.child("companies/" + companyName + "/tasks/current").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 TASKS = new ArrayList<>();  
 // Get Post object and use the values to update the UI  
 for (DataSnapshot dataTask: dataSnapshot.getChildren()) {  
 TASKS.add(dataTask.getValue(Task.class));  
  
 }  
 lv.setAdapter(new TasksAdapter(getContext() , TASKS, "current", isAdmin, companyName));  
 progressBar.setVisibility(View.INVISIBLE);  
 if (!TASKS.isEmpty()) {  
 emptyText.setVisibility(View.INVISIBLE);  
 } else {  
 emptyText.setVisibility(View.VISIBLE);  
 }  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 // Getting Post failed, log a message  
 Toast.makeText(getContext(), databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
 lv.setAdapter(new TasksAdapter(getContext() , TASKS, "current", isAdmin, companyName));  
  
  
  
 SimpleDateFormat formatter = new SimpleDateFormat("EEEE dd.MM", Locale.getDefault());  
 String date = formatter.format(new Date());  
 Toolbar toolbar = (Toolbar) v.findViewById(R.id.toolbar);  
 setHasOptionsMenu(true);  
 AppCompatActivity activity = (AppCompatActivity) getActivity();  
 if (toolbar != null) {  
 if (isAdmin && companyNameOriginal != null) {  
 toolbar.setTitle(companyNameOriginal);  
 } else {  
 toolbar.setTitle(date);  
 }  
 if (activity != null) {  
 activity.setSupportActionBar(toolbar);  
 }  
 }  
  
 return v;  
  
 }  
  
  
 @Override  
 public void onCreateOptionsMenu(@NonNull Menu menu, @NonNull MenuInflater inflater) {  
 inflater.inflate(R.menu.menu\_main, menu);  
 super.onCreateOptionsMenu(menu, inflater);  
 }  
  
 // Заглушка, работа с меню  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
  
 if (id == R.id.menu\_main\_add) {  
 startActivity(new Intent(this.getActivity(), AddTaskActivity.class));  
 } else if (id == R.id.menu\_main\_logOut) {  
 FirebaseAuth.getInstance().signOut();  
 sharedPreferences.getAll().clear();  
 startActivity(new Intent(getActivity(), MainActivity.class));  
 Objects.requireNonNull(getActivity()).finish();  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
  
  
  
 @Override  
 public void onSaveInstanceState(@NonNull Bundle outState) {  
 super.onSaveInstanceState(outState);  
 }  
}

SendTodayFragment.java

package com.example.merchtrib.ui.fragments;  
  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ImageButton;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.activities.AddTaskActivity;  
import com.example.merchtrib.ui.activities.TaskActivity;  
import com.example.merchtrib.ui.adapters.TasksAdapter;  
import com.example.merchtrib.ui.objects.Task;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.GenericTypeIndicator;  
import com.google.firebase.database.ValueEventListener;  
  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
import java.util.Locale;  
import java.util.Objects;  
  
public class SendTodayFragment extends Fragment {  
  
 public ArrayList<Task> TASKS = new ArrayList<Task>();  
 private DatabaseReference mDatabase;  
 SharedPreferences sharedPreferences;  
 String userEmail, companyName, companyNameOriginal;  
 boolean isAdmin;  
 RecyclerView lv;  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.fragment\_send\_today, container, false);  
  
 lv = v.findViewById(R.id.list\_of\_tasks);  
 LinearLayoutManager llm = new LinearLayoutManager(this.getContext());  
 llm.setOrientation(LinearLayoutManager.VERTICAL);  
 DividerItemDecoration dividerItemDecoration = new DividerItemDecoration(lv.getContext(),  
 llm.getOrientation());  
 lv.addItemDecoration(dividerItemDecoration);  
 lv.setLayoutManager(llm);  
 TextView emptyText = v.findViewById(R.id.emptyText);  
  
 sharedPreferences = getActivity().getSharedPreferences("data", Context.MODE\_PRIVATE);  
 userEmail = sharedPreferences.getString("userEmailShort", "");  
 companyName = sharedPreferences.getString("companyName", "");  
 isAdmin = sharedPreferences.getBoolean("isAdmin", false);  
 companyNameOriginal = sharedPreferences.getString("companyNameOriginal", null);  
  
 mDatabase = FirebaseDatabase.getInstance().getReference();  
 ProgressBar progressBar = v.findViewById(R.id.progress\_circular);  
 mDatabase.child("companies/" + companyName + "/tasks/done").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 TASKS = new ArrayList<>();  
 // Get Post object and use the values to update the UI  
 for (DataSnapshot dataTask: dataSnapshot.getChildren()) {  
 TASKS.add(dataTask.getValue(Task.class));  
 }  
 lv.setAdapter(new TasksAdapter(getContext() , TASKS, "done", isAdmin, companyName));  
 progressBar.setVisibility(View.INVISIBLE);  
 if (!TASKS.isEmpty()) {  
 emptyText.setVisibility(View.INVISIBLE);  
 } else {  
 emptyText.setVisibility(View.VISIBLE);  
 }  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 // Getting Post failed, log a message  
 Toast.makeText(getContext(), databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
 lv.setAdapter(new TasksAdapter(getContext() , TASKS, "done", isAdmin, companyName));  
  
  
 Toolbar toolbar = (Toolbar) v.findViewById(R.id.toolbar);  
 AppCompatActivity activity = (AppCompatActivity) getActivity();  
 if (toolbar != null) {  
 toolbar.setTitle("История");  
 if (activity != null) {  
 activity.setSupportActionBar(toolbar);  
 }  
 }  
  
 return v;  
  
 }  
  
  
  
 @Override  
 public void onSaveInstanceState(@NonNull Bundle outState) {  
 super.onSaveInstanceState(outState);  
 }  
}

UsersActivity.java

package com.example.merchtrib.ui.fragments;  
  
import android.content.Context;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DividerItemDecoration;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.merchtrib.R;  
import com.example.merchtrib.ui.adapters.UsersAdapter;  
import com.example.merchtrib.ui.objects.User;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.regex.Pattern;  
  
public class UsersFragment extends Fragment {  
  
 DatabaseReference mDatabase;  
 RecyclerView recyclerView;  
 String companyName;  
 TextView emptyText;  
 ProgressBar progressBar;  
 ArrayList<String> users = new ArrayList<>();  
 Button addButton;  
 EditText email;  
 FirebaseDatabase mAddData;  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.fragment\_users, container, false);  
 companyName = getActivity().getSharedPreferences("data", Context.MODE\_PRIVATE).getString("companyName", "");  
 progressBar = v.findViewById(R.id.progress\_circular);  
 mAddData = FirebaseDatabase.getInstance();  
 mDatabase = mAddData.getReference("companies/" + companyName + "/users");  
 emptyText = v.findViewById(R.id.emptyText);  
 addButton = v.findViewById(R.id.addButton);  
 email = v.findViewById(R.id.email);  
  
  
 mDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 progressBar.setVisibility(View.VISIBLE);  
 emptyText.setVisibility(View.INVISIBLE);  
 users = new ArrayList<>();  
 if (dataSnapshot.getValue() != null) {  
  
 progressBar.setVisibility(View.INVISIBLE);  
 for (DataSnapshot item : dataSnapshot.getChildren()) {  
 users.add(item.getValue(String.class));  
 }  
  
 } else {  
 progressBar.setVisibility(View.INVISIBLE);  
 emptyText.setVisibility(View.VISIBLE);  
 }  
 recyclerView.setAdapter(new UsersAdapter(getContext(), users, companyName));  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 Toast.makeText(getContext(), "Что-то пошло не так: " + databaseError.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
 });  
  
 recyclerView = v.findViewById(R.id.list\_of\_tasks);  
 LinearLayoutManager llm = new LinearLayoutManager(this.getContext());  
 llm.setOrientation(LinearLayoutManager.VERTICAL);  
 DividerItemDecoration dividerItemDecoration = new DividerItemDecoration(recyclerView.getContext(),  
 llm.getOrientation());  
 recyclerView.addItemDecoration(dividerItemDecoration);  
 recyclerView.setLayoutManager(llm);  
  
 Toolbar toolbar = (Toolbar) v.findViewById(R.id.toolbar);  
 AppCompatActivity activity = (AppCompatActivity) getActivity();  
 if (toolbar != null) {  
 toolbar.setTitle("Работники");  
 if (activity != null) {  
 activity.setSupportActionBar(toolbar);  
 }  
 }  
  
 addButton.setOnClickListener(view -> {  
 if (!email.getText().toString().isEmpty() && isValid(email.getText().toString())) {  
 String originalEmail = email.getText().toString();  
 String emailShort = originalEmail.replace("@","").replace(".","").toLowerCase();  
 mAddData.getReference("users").child(emailShort).setValue(new User(companyName, false));  
 mAddData.getReference("companies").child(companyName).child("users").child(emailShort).setValue(originalEmail).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 email.setText("");  
 } else {  
 Toast.makeText(getContext(), "Что-то пошло не так: " + task.getException(), Toast.LENGTH\_LONG).show();  
 }  
 }  
 });  
 } else {  
 Toast.makeText(getContext(), "Поле пустое или введена не электронная почта", Toast.LENGTH\_LONG).show();  
 }  
 });  
  
  
 return v;  
 }  
  
 public static boolean isValid(String email) {  
 String emailRegex = "^[a-zA-Z0-9\_+&\*-]+(?:\\." +  
 "[a-zA-Z0-9\_+&\*-]+)\*@" +  
 "(?:[a-zA-Z0-9-]+\\.)+[a-z" +  
 "A-Z]{2,7}$";  
  
 Pattern pat = Pattern.compile(emailRegex);  
 if (email == null)  
 return false;  
 return pat.matcher(email).matches();  
 }  
}

Company.java

package com.example.merchtrib.ui.objects;  
  
import com.example.merchtrib.ui.objects.Task;  
import com.example.merchtrib.ui.objects.TaskList;  
  
import java.util.ArrayList;  
  
public class Company {  
  
 public String name;  
 public ArrayList<CompanyUser> users;  
 public ArrayList<TaskList> tasks;  
  
 public Company() {};  
  
  
 public Company(String name, ArrayList<CompanyUser> users, ArrayList<TaskList> tasks) {  
 this.name = name;  
 this.users = users;  
 this.tasks = tasks;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public ArrayList<CompanyUser> getUsers() {  
 return users;  
 }  
  
 public void setUsers(ArrayList<CompanyUser> users) {  
 this.users = users;  
 }  
  
 public ArrayList<TaskList> getTasks() {  
 return tasks;  
 }  
  
 public void setTasks(ArrayList<TaskList> tasks) {  
 this.tasks = tasks;  
 }  
}

CompanyUser.java

package com.example.merchtrib.ui.objects;  
  
public class CompanyUser {  
 public String email;  
  
 public CompanyUser(){};  
 public CompanyUser(String email) {  
 this.email = email;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
}

Task.java

package com.example.merchtrib.ui.objects;  
  
import java.util.ArrayList;  
  
public class Task {  
 public String name;  
 public String address;  
 public String id;  
 public String addressLink;  
 public ArrayList<String> images;  
  
 public Task() {};  
  
 public Task(String id, String name, String address, String addressLink, ArrayList<String> images) {  
 this.id = id;  
 this.name = name;  
 this.address = address;  
 this.addressLink = addressLink;  
 this.images = images;  
 }  
  
 public ArrayList<String> getImages() {  
 return images;  
 }  
  
 public void setImages(ArrayList<String> images) {  
 this.images = images;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getAddressLink() {  
 return addressLink;  
 }  
  
 public void setAddressLink(String addressLink) {  
 this.addressLink = addressLink;  
 }  
}

TaskList.java

package com.example.merchtrib.ui.objects;  
  
import java.lang.reflect.Array;  
import java.util.ArrayList;  
  
public class TaskList {  
 public ArrayList<Task> array;  
  
 public TaskList(){};  
  
 public TaskList(ArrayList<Task> array) {  
 this.array = array;  
 }  
  
 public ArrayList<Task> getArray() {  
 return array;  
 }  
  
 public void setArray(ArrayList<Task> array) {  
 this.array = array;  
 }  
}

User.java

package com.example.merchtrib.ui.objects;  
  
public class User {  
  
 public String company;  
 public boolean admin;  
  
 public User() {};  
  
 public User(String company, boolean admin) {  
 this.company = company;  
 this.admin = admin;  
 }  
  
 public String getCompany() {  
 return company;  
 }  
  
 public void setCompany(String company) {  
 this.company = company;  
 }  
  
 public boolean isAdmin() {  
 return admin;  
 }  
  
 public void setAdmin(boolean admin) {  
 this.admin = admin;  
 }  
}

activity\_add\_task.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.AddTaskActivity">  
  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/toolbar"  
 android:layout\_marginTop="16dp"  
 android:orientation="vertical"  
 android:paddingHorizontal="16dp">  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="12dp"  
 >  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="Название"  
 android:inputType="text"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 >  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/address"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="Адрес"  
 android:inputType="text"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 >  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/addressLink"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:hint="Ссылка на карты"  
 android:inputType="text"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:layout\_marginTop="24dp"  
 android:id="@+id/addButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Добавить"  
 android:onClick="addButtonClickHandler"  
 android:background="@color/Primary"  
 />  
  
 </LinearLayout>  
</RelativeLayout>

Activity\_admin.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.AdminActivity">  
  
  
 <FrameLayout  
 android:id="@+id/fragment\_container\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentStart="true"  
 android:layout\_above="@id/bottom\_navigation"  
 />  
  
 <include  
 android:id="@+id/bottom\_navigation"  
 layout="@layout/bottom\_navigation"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 />  
  
</RelativeLayout>

activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.LoginActivity">  
  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:orientation="vertical">  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/login"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginEnd="25dp"  
 android:hint="E-mail"  
 android:inputType="textEmailAddress"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 />  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 app:passwordToggleContentDescription="Показать пароль"  
 app:passwordToggleEnabled="true">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:layout\_width="match\_parent"  
 android:id="@+id/password"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginEnd="25dp"  
 android:hint="Пароль"  
 android:inputType="textPassword"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 android:textColorHighlight="@color/PrimaryLight"  
 android:textColorLink="@color/PrimaryLight" />  
 </com.google.android.material.textfield.TextInputLayout>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Войти"  
 android:id="@+id/loginButton"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/Primary"  
 android:layout\_marginTop="50dp"  
 />  
 </LinearLayout>  
  
</RelativeLayout>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.MainActivity">  
  
  
 <FrameLayout  
 android:id="@+id/fragment\_container\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentStart="true"  
 android:layout\_above="@id/bottom\_navigation"  
 />  
 <ProgressBar  
 android:id="@+id/progress\_bar"  
 style="@style/Widget.AppCompat.ProgressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:indeterminateTint="@color/primary"  
 />  
  
 <include  
 android:id="@+id/bottom\_navigation"  
 layout="@layout/bottom\_navigation"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 />  
  
</RelativeLayout>

Activity\_register\_company.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.RegisterCompanyActivity">  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_below="@+id/toolbar"  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:id="@+id/companyLayout"  
 android:layout\_marginTop="16dp"  
 android:layout\_height="wrap\_content">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/company"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginEnd="25dp"  
 android:hint="Название компании"  
 android:inputType="text"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 />  
 </com.google.android.material.textfield.TextInputLayout>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/companyLayout"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="16dp"  
 android:text="В будущем изменить название будет невозможно"  
 android:id="@+id/warning"  
 android:textColor="@color/Error"  
  
 />  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:paddingHorizontal="16dp"  
 android:layout\_height="wrap\_content"  
 android:text="Применить"  
 android:id="@+id/submit"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/Primary"  
 android:layout\_marginTop="50dp"  
 android:layout\_below="@+id/warning"  
 android:layout\_centerHorizontal="true"  
 />  
  
</RelativeLayout>

Activity\_sending.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.SendingActivity">  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 />  
  
 <TextView  
 android:id="@+id/check\_geo\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/sending\_task\_\_title"  
 android:layout\_marginEnd="@dimen/activity\_horizontal\_margin"  
 android:layout\_marginStart="@dimen/activity\_horizontal\_margin"  
 android:textSize="24sp"  
 android:textAlignment="center"  
 android:textColor="@color/white"  
 android:layout\_marginTop="50dp"  
 android:layout\_below="@+id/toolbar"  
 />  
  
 <TextView  
 android:id="@+id/check\_geo\_second\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/sending\_task\_second\_title"  
 android:layout\_marginEnd="@dimen/activity\_horizontal\_margin"  
 android:layout\_marginStart="@dimen/activity\_horizontal\_margin"  
 android:textSize="18sp"  
 android:textAlignment="center"  
 android:layout\_marginTop="14dp"  
 android:layout\_below="@+id/check\_geo\_title"  
 />  
  
  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:layout\_width="300dp"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/check\_geo\_second\_title"  
 app:lottie\_autoPlay="true"  
 app:lottie\_loop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:scaleType="fitCenter"  
 app:lottie\_url="https://assets8.lottiefiles.com/private\_files/lf30\_jhytdu.json"  
 />  
</RelativeLayout>

Activity\_task.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.activities.TaskActivity">  
  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true" />  
  
 <ProgressBar  
 android:id="@+id/progress\_circular"  
 style="@style/Widget.AppCompat.ProgressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:indeterminateTint="@color/primary"  
 />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/rev\_photo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/toolbar"  
 android:layout\_gravity="center\_horizontal"  
 android:orientation="vertical"  
 android:paddingStart="16dp"  
 android:paddingTop="16dp"  
  
 android:paddingEnd="16dp"  
 app:layoutManager="androidx.recyclerview.widget.GridLayoutManager" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/floatingActionButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:contentDescription="Add photo"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginEnd="32dp"  
 android:layout\_marginBottom="32dp"  
 android:clickable="true"  
 app:backgroundTint="@color/Primary"  
 app:srcCompat="@drawable/ic\_add\_photo" />  
  
  
  
  
</RelativeLayout>

Fragment\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.fragments.MainFragment">  
  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 />  
 <ProgressBar  
 android:id="@+id/progress\_circular"  
 style="@style/Widget.AppCompat.ProgressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:indeterminateTint="@color/primary"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Нет заданий"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:id="@+id/emptyText"  
 android:visibility="invisible"  
 />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/list\_of\_tasks"  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/toolbar"  
 android:layout\_alignParentStart="true"  
 android:transcriptMode="alwaysScroll">  
 </androidx.recyclerview.widget.RecyclerView>  
  
  
</RelativeLayout>

Fragment\_send\_today.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.fragments.SendTodayFragment">  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 />  
 <ProgressBar  
 android:id="@+id/progress\_circular"  
 style="@style/Widget.AppCompat.ProgressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:indeterminateTint="@color/primary"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Нет заданий"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:id="@+id/emptyText"  
 android:visibility="invisible"  
 />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/list\_of\_tasks"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/toolbar"  
 android:layout\_alignParentStart="true"  
 android:transcriptMode="alwaysScroll">  
 </androidx.recyclerview.widget.RecyclerView>  
  
</RelativeLayout>

Fragment\_users.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ui.fragments.UsersFragment">  
 <include  
 android:id="@+id/toolbar"  
 layout="@layout/tool\_bar"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 />  
 <ProgressBar  
 android:id="@+id/progress\_circular"  
 style="@style/Widget.AppCompat.ProgressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:indeterminateTint="@color/primary"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Нет пользователей"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:id="@+id/emptyText"  
 android:visibility="invisible"  
 />  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/editTextLayout"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_below="@+id/toolbar">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginHorizontal="16dp"  
 android:hint="E-mail"  
 android:inputType="textEmailAddress"  
 android:singleLine="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
<Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Добавить"  
 android:background="@color/primary\_dark"  
 android:layout\_below="@id/editTextLayout"  
 android:id="@+id/addButton"  
 android:layout\_marginHorizontal="16dp"  
 android:layout\_marginBottom="16dp"  
 />  
  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/list\_of\_tasks"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/addButton"  
 android:layout\_alignParentStart="true"  
 android:transcriptMode="alwaysScroll">  
 </androidx.recyclerview.widget.RecyclerView>  
  
</RelativeLayout>

Image\_task\_item.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
<com.makeramen.roundedimageview.RoundedImageView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/photo"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_marginBottom="16dp"  
 android:contentDescription="@string/image\_description"  
 android:scaleType="centerCrop"  
 app:riv\_border\_color="@color/md\_grey\_400"  
 app:riv\_border\_width="@dimen/image\_item\_border\_width"  
 app:riv\_corner\_radius="@dimen/image\_item\_corner\_radius"  
 app:riv\_mutate\_background="true"/>  
</LinearLayout>

Main\_task\_List\_item.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/parent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:descendantFocusability="blocksDescendants"  
 android:orientation="horizontal">  
  
 <ImageButton  
 android:id="@+id/task\_item\_button"  
 style="@android:style/Widget.Holo.ImageButton"  
 android:layout\_width="80dp"  
 android:layout\_height="match\_parent"  
 android:background="#00FFFFFF"  
 android:contentDescription="@string/Description\_add\_image"  
 android:src="@drawable/ic\_location" />  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="@dimen/activity\_horizontal\_margin">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_centerVertical="true"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/list\_item\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="OOO@fermer@"  
 android:textColor="@color/white"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/list\_item\_link"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:autoLink="web"  
 android:paddingTop="5dp"  
 android:text="@string/list\_item\_link"  
 android:textStyle="italic" />  
 </LinearLayout>  
  
 <ImageView  
 android:layout\_width="36dp"  
 android:layout\_height="36dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_centerVertical="true"  
 android:contentDescription="@string/Description\_add\_image"  
 android:src="@drawable/ic\_main\_list\_item\_arrow" />  
  
 </RelativeLayout>  
  
  
</LinearLayout>

User\_item.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="16dp"  
 android:gravity="center"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:id="@+id/layout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:orientation="horizontal"  
 >  
  
 <TextView  
 android:id="@+id/list\_item\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="test"  
 android:textColor="@color/white"  
 android:textSize="20sp" />  
 </LinearLayout>  
  
 <ImageButton  
 android:layout\_width="42dp"  
 android:layout\_height="42dp"  
 android:id="@+id/delete"  
 android:background="#00000000"  
 android:src="@drawable/ic\_baseline\_delete\_forever\_24"  
 android:contentDescription="delete" />  
 </LinearLayout>  
</RelativeLayout>

Приложение Б. Руководство пользователя

Руководство пользователя мобильного приложения для устройств на ОС Android

1. Требования к аппаратуре и программному обеспечению

Для устройств на базе ОС Android: Android — версия 9.0 и старше;

2. Установка, обновление, удаление мобильного приложения

Установка мобильного приложения выполняется из магазина приложений Google Play. Обновление мобильного приложения выполняется средствами магазина приложений.

Для удаления мобильного приложения необходимо на мобильном устройстве запустить приложение Google Play и в нем удалить мобильное приложение.

3. Запуск мобильного приложения

Для запуска мобильного приложения необходимо найти нужное приложение в списке и нажать на его картинку. При этом в большинстве случаев будет открыто основное окно мобильного приложения. Однако если с данной программой связано несколько приложений, то будет открыт список этих приложений. В этом случае для запуска приложения следует нажать на нужное приложение в списке.

4. Интерфейс системы

Интерфейс мобильной платформы ориентирован на то, что в один момент времени на экране отображается одна форма. Первой видимой формой обычно является рабочий стол. Команды, открывающие другие формы, собраны в главное меню приложения. Вызов главного меню приложения осуществляется с помощью системной кнопки вызова меню.

* 1. Выход из приложения

Для выхода из него можно свайпнуть экран с нижнего края вверх. И вы окажешь на рабочем экране смартфона.