Приложение Д. Текст программы

АННОТАЦИЯ

В данном программном документе приведен текст программы мобильного приложения «MAU».

Текст программы включает в себя все необходимые компоненты для полноценного функционирования системы.

СОДЕРЖАНИЕ

[1. ТЕКСТ ПРОГРАММЫ 4](#_Toc167269593)

[1.1. Наименование программы 4](#_Toc167269594)

[1.2. Область применения программы 4](#_Toc167269595)

[1.3. Модули 4](#_Toc167269596)

[1.4. Код программы 9](#_Toc167269597)

# 1. ТЕКСТ ПРОГРАММЫ

## Наименование программы

Наименование программы – «MAU».

## Область применения программы

Приложение предназначено для использования в области ментального здоровья в повседневной жизни.

## Модули

| Модуль | Описание | Количество строк кода | Размер (в Кбайтах) |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| AndroidManifest.xml | Содержит информацию о приложении, его компонентах, разрешениях, минимальных и целевых версиях SDK, а также конфигурациях устройств и дополнительных метаданных | 76 | 3 КБ |
| build.gradle.kts(Project) | Содержит настройки для всего проекта, включая репозитории, общие зависимости, и плагины | 11 | 1 КБ |
| build.gradle.kts(App) | Содержит зависимости конкретного модуля, настройки компиляции, конфигурацию плагинов и параметры сборки | 56 | 2 КБ |
| Articles/AddArticleActivity.java | Активити для добавления новой статьи | 119 | 6 КБ |
| Articles/ArticleAdapter.java | Адаптер для отображения списка статей в RecyclerView | 123 | 5 КБ |
| Articles/ArticleDetailsActivity.java | Активити для отображения подробной информации о выбранной статье | 46 | 2 КБ |
| Articles/ArticlesFragment.java | Фрагмент, который отображает список статей, поддерживающий взаимодействие с пользователем и управление элементами списка (добавление, изменение, удаление) | 161 | 7 КБ |
| Articles/ArticlesManager.java | Класс для управления статьями, включающий методы для добавления, редактирования, удаления и получения статей из базы данных | 89 | 4 КБ |
| Articles/EditArticleActivity.java | Активити для редактирования существующей статьи | 145 | 7 КБ |
| models/Articles.java | Представляет собой модель статьи, содержит конструктор, методы доступа для получения и установки значений полей | 62 | 2 КБ |
| models/Note.java | Представляет собой модель заметки, содержит конструктор, методы доступа для получения и установки значений полей | 49 | 2 КБ |
| models/Song.java | Представляет собой модель мелодии, содержит конструктор, методы доступа для получения и установки значений полей | 36 | 2 КБ |
| models/User.java | Представляет собой модель пользователя, содержит конструктор, методы доступа для получения и установки значений полей | 48 | 1 КБ |
| Notes/AddNoteActivity.java | Активити для добавления новой заметки | 115 | 4 КБ |
| Notes/EditNoteActivity.java | Активити для редактирования существующей заметки | 110 | 4 КБ |
| Notes/NoteAdapter.java | Адаптер для отображения списка заметок в RecyclerView | 113 | 4 КБ |
| Notes/NoteDetailsActivity.java | Активити для отображения подробной информации о выбранной заметке | 42 | 2 КБ |
| Notes/NoteManager.java | Класс для управления заметками, включающий методы для добавления, редактирования, удаления и получения статей из базы данных | 65 | 3 КБ |
| Notes/ProfileFragment.java | Фрагмент, который отображает список заметок, поддерживающий взаимодействие с пользователем и управление элементами списка (добавление, изменение, удаление) | 158 | 7 КБ |
| Notification/NotificationSubscriptionManager.java | Класс отвечает за управление подпиской на push-уведомления | 43 | 2 КБ |
| Notification/PushService.java | Служба для обработки и отправки push-уведомлений | 71 | 3 КБ |
| Player/AddSongActivity.java | Активити для добавления новой мелодии | 184 | 8 КБ |
| Player/HomeFragment.java | Фрагмент, который отображает список мелодий, поддерживающий взаимодействие с пользователем и управление элементами списка (добавление, удаление) | 138 | 6 КБ |
| Player/PlayerActivity.java | Активити аудиоплеера, для прослушивания выбранной мелодии | 178 | 7 КБ |
| Player/SongAdapter.java | Адаптер для отображения списка мелодий в RecyclerView | 107 | 4 КБ |
| LoginManager.java | Класс, отвечающий за управление процессом входа пользователя в приложение | 114 | 4 КБ |
| MainActivity.java | Класс, отвечающий за управление пользовательским интерфейсом, навигацию между фрагментами приложения | 66 | 3 КБ |
| RegistrationManager.java | Класс, отвечающий за управление процессом регистрации пользователя в системе | 140 | 8 КБ |
| SettingsFragment.java | Фрагмент для управления настройками уведомлений и профиля пользователя | 413 | 19 КБ |
| WelcomeActivity.java | Активити для отображения приветственного сообщение после регистрации пользователя | 29 | 1 КБ |
| activity\_add\_article.xml | Макет отображения формы добавления новой статьи | 180 | 7 КБ |
| activity\_add\_note.xml | Макет отображения формы добавления новой заметки | 97 | 4 КБ |
| activity\_add\_song.xml | Макет отображения формы добавления новой песни | 112 | 5 КБ |
| activity\_article\_details.xml | Макет отображения деталей статьи | 62 | 3 КБ |
| activity\_edit\_article.xml | Макет отображения формы редактирования статьи | 181 | 7 КБ |
| activity\_edit\_note.xml | Макет отображения формы редактирования заметки | 97 | 4 КБ |
| activity\_login.xml | Макет отображения экрана входа в систему | 166 | 7 КБ |
| activity\_main.xml | Макет главного экрана приложения | 31 | 2 КБ |
| activity\_note\_details.xml | Макет отображения деталей заметки | 51 | 2 КБ |
| activity\_player.xml | Макет отображения экрана проигрывателя музыки | 79 | 3 КБ |
| activity\_registration.xml | Макет отображения формы регистрации нового пользователя | 187 | 8 КБ |
| activity\_welcome.xml | Макет отображения экрана приветствия приложения | 121 | 5 КБ |
| circle\_progress\_with\_text.xml | Макет отображения круглого прогресс-бара с текстовым значением прогресса | 29 | 2 КБ |
| fragment\_articles.xml | Макет отображения списка статей в фрагменте | 92 | 4 КБ |
| fragment\_home.xml | Макет главного фрагмента приложения | 103 | 5 КБ |
| fragment\_profile.xml | Макет отображения профиля пользователя во фрагменте | 126 | 6 КБ |
| fragment\_settings.xml | Макет отображения настроек приложения во фрагменте | 308 | 13 КБ |
| list\_item\_article.xml | Макет элемента списка для отображения одной статьи | 63 | 3 КБ |
| list\_item\_note.xml | Макет элемента списка для отображения одной заметки | 48 | 2 КБ |
| list\_item\_song.xml | Макет элемента списка для отображения одной песни | 78 | 3 КБ |
| add\_note\_button\_background.xml | Макет для кнопки добавления заметки с закругленными углами и синим фоном | 9 | 1 КБ |
| auth\_button\_back.xml | Макет для кнопки аутентификации с закругленными углами и черным фоном | 10 | 1 КБ |
| begin\_button\_back.xml | Макет для кнопки начала с закругленными углами и светло-серым фоном | 9 | 1 КБ |
| circle\_progress\_bar.xml | Макет для круглого прогресс-бара с двумя слоями: фоновым и прогрессом | 38 | 2 КБ |
| date\_and\_description\_back.xml | Макет для заднего фона с закругленными углами для отображения даты и описания | 5 | 1 КБ |
| edit\_text\_back.xml | Макет для заднего фона текстового поля с белым фоном и закругленными углами | 10 | 1 КБ |
| header\_profile\_settings.xml | Макет для заднего фона заголовка профиля и настроек с голубым цветом | 4 | 1 КБ |
| round\_background.xml | Макет для заднего фона с закругленными углами и белым цветом | 5 | 1 КБ |
| settings\_card\_background.xml | Макет для фона карточки настроек с серым цветом и закругленными углами | 6 | 1 КБ |
| bottom\_navigation\_bar.xml | Макет для нижней панели навигации. | 21 | 1 КБ |
| context\_menu\_delete.xml | Макет для контекстного меню, предназначенного для удаления элементов | 6 | 1 КБ |
| context\_menu\_update\_delete.xml | Макет для контекстного меню, предназначенного для удаления и редактирования элементов | 12 | 1 КБ |
| themes.xml | Файл ресурсов, определяющий тему и стили приложения | 36 | 2 КБ |

## Код программы

1) AddArticleActivity.java;

package com.example.MAU.Articles;

import android.content.Intent;

import android.os.Bundle;

import android.text.Editable;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.MAU.MainActivity;

import com.example.MAU.R;

import com.example.MAU.models.Articles;

import com.google.firebase.auth.FirebaseAuth;

public class AddArticleActivity extends AppCompatActivity {

private EditText titleEditText, descriptionEditText, photoURLEditText, articleTextEditText;

private Button addArticleButton;

private ImageButton backButton, boldButton, italicButton, paragraph, h1Button, h2Button, h3Button;

private ArticlesManager articlesManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setTheme(R.style.AppTheme);

setContentView(R.layout.activity\_add\_article);

FirebaseAuth auth = FirebaseAuth.getInstance();

titleEditText = findViewById(R.id.titleEditText);

descriptionEditText = findViewById(R.id.descriptionEditText);

articleTextEditText = findViewById(R.id.articleTextEditText);

photoURLEditText = findViewById(R.id.photoURLEditText);

addArticleButton = findViewById(R.id.addArticleButton);

backButton = findViewById(R.id.backButton);

boldButton = findViewById(R.id.boldButton);

italicButton = findViewById(R.id.italicButton);

paragraph = findViewById(R.id.paragraph);

h1Button = findViewById(R.id.h1Button);

h2Button = findViewById(R.id.h2Button);

h3Button = findViewById(R.id.h3Button);

articlesManager = new ArticlesManager();

View.OnClickListener formatButtonClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

String tag = (String) v.getTag();

if (tag != null) {

int start = articleTextEditText.getSelectionStart();

int end = articleTextEditText.getSelectionEnd();

if (start == end){

Toast.makeText(AddArticleActivity.this, "Выделите текст", Toast.LENGTH\_SHORT).show();

} else {

String selectedText = articleTextEditText.getText().toString().substring(start, end);

String htmlText = String.format("<%s>%s</%s>", tag, selectedText, tag);

appendFormattedText(htmlText);

}

}

}

};

boldButton.setOnClickListener(formatButtonClickListener);

italicButton.setOnClickListener(formatButtonClickListener);

paragraph.setOnClickListener(formatButtonClickListener);

h1Button.setOnClickListener(formatButtonClickListener);

h2Button.setOnClickListener(formatButtonClickListener);

h3Button.setOnClickListener(formatButtonClickListener);

addArticleButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String title = titleEditText.getText().toString().trim();

String description = descriptionEditText.getText().toString().trim();

String articleText = articleTextEditText.getText().toString().trim();

String photo\_URL = photoURLEditText.getText().toString().trim();

if (!title.isEmpty() && !description.isEmpty() && !articleText.isEmpty()) {

if (photo\_URL.isEmpty()){

photo\_URL = "<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS9PJscu_d_qoxDIWQnCEU-KF-6WpzI0hbMS-TmuUeaWw&s>";

}

Articles article = new Articles(title, description, articleText, photo\_URL);

articlesManager.addArticle(article);

startActivity(new Intent(getApplicationContext(), MainActivity.class));

finish();

} else {

Toast.makeText(AddArticleActivity.this, "Заполните все поля", Toast.LENGTH\_SHORT).show();

}

}

});

backButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(getApplicationContext(), MainActivity.class));

finish();

}

});

}

private void appendFormattedText(String formattedText) {

Editable editable = articleTextEditText.getText();

int start = articleTextEditText.getSelectionStart();

int end = articleTextEditText.getSelectionEnd();

editable.replace(start, end, formattedText);

articleTextEditText.setSelection(start + formattedText.length());

}

@Override

public void onBackPressed() {

super.onBackPressed();

Intent intent = new Intent(this, MainActivity.class);

intent.putExtra("fragment", "articles");

intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_SINGLE\_TOP);

startActivity(intent);

finish();

}

}

2) ArticleAdapter.java;

package com.example.MAU.Articles;

import android.content.Context;

import android.content.Intent;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.example.MAU.R;

import com.example.MAU.models.Articles;

import com.example.MAU.models.Note;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.squareup.picasso.Picasso;

import java.util.List;

import java.util.Objects;

public class ArticleAdapter extends RecyclerView.Adapter<ArticleAdapter.ArticleViewHolder> {

private Context context;

private List<Articles> articles;

public interface OnArticleClickListener {

void onArticleClick(int position);

void onArticleLongClick(int position);

}

private OnArticleClickListener mListener;

public void setOnArticleClickListener(OnArticleClickListener listener) {

mListener = listener;

}

public ArticleAdapter(Context context, List<Articles> articles) {

this.context = context;

this.articles = articles;

}

public List<Articles> getNotes() {

return articles;

}

@NonNull

@Override

public ArticleViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(context).inflate(R.layout.list\_item\_article, parent, false);

return new ArticleViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ArticleViewHolder holder, int position) {

Articles currentArticle = articles.get(position);

holder.textViewTitle.setText(currentArticle.getTitle());

holder.textViewDescription.setText(currentArticle.getDescription());

Picasso.get().load(currentArticle.getPhoto\_URL()).into(holder.imageViewPhoto);

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (mListener != null) {

mListener.onArticleClick(position);

Intent intent = new Intent(context, ArticleDetailsActivity.class);

intent.putExtra("title", currentArticle.getTitle());

intent.putExtra("description", currentArticle.getDescription());

intent.putExtra("articleText", currentArticle.getArticleText());

intent.putExtra("photo\_URL", currentArticle.getPhoto\_URL());

context.startActivity(intent);

}

}

});

FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();

if(Objects.equals(currentUser.getDisplayName(), "Administrator")){

holder.itemView.setOnLongClickListener(new View.OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

if (mListener != null) {

mListener.onArticleLongClick(position);

return true;

}

return false;

}

});

}

}

@Override

public int getItemCount() {

return articles.size();

}

public class ArticleViewHolder extends RecyclerView.ViewHolder {

public ImageView imageViewPhoto;

TextView textViewTitle;

TextView textViewDescription;

public ArticleViewHolder(@NonNull View itemView) {

super(itemView);

textViewTitle = itemView.findViewById(R.id.textViewTitle);

textViewDescription = itemView.findViewById(R.id.textViewDescription);

imageViewPhoto = itemView.findViewById(R.id.imageViewPhoto);

}

}

public void setArticles(List<Articles> articles) {

this.articles = articles;

notifyDataSetChanged();

}

public List<Articles> getArticles() {

return articles;

}

}

3) ArticleDetailsActivity.java;

package com.example.MAU.Articles;

import android.os.Build;

import android.os.Bundle;

import android.text.Html;

import android.view.View;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.text.HtmlCompat;

import com.example.MAU.R;

public class ArticleDetailsActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_article\_details);

// Получение данных о статье из Intent

String title = getIntent().getStringExtra("title");

String description = getIntent().getStringExtra("description");

String articleText = getIntent().getStringExtra("articleText");

// Настройка TextView для отображения данных о статье

TextView titleTextView = findViewById(R.id.titleTextView);

TextView descriptionTextView = findViewById(R.id.descriptionTextView);

TextView articleTextView = findViewById(R.id.articleTextTextView);

titleTextView.setText(title);

descriptionTextView.setText(description);

articleTextView.setText(HtmlCompat.fromHtml(articleText, HtmlCompat.FROM\_HTML\_MODE\_LEGACY));

ImageButton backButton = findViewById(R.id.backButton);

backButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

}

}

4) ArticlesFragment.java;

package com.example.MAU.Articles;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.ContextMenu;

import android.view.Gravity;

import android.view.LayoutInflater;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageButton;

import android.widget.PopupMenu;

import android.widget.ProgressBar;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.example.MAU.Notes.AddNoteActivity;

import com.example.MAU.Notes.EditNoteActivity;

import com.example.MAU.Notes.NoteManager;

import com.example.MAU.R;

import com.example.MAU.models.Articles;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

import java.util.List;

import java.util.Objects;

public class ArticlesFragment extends Fragment implements ArticleAdapter.OnArticleClickListener {

ImageButton addArticleButton;

private FirebaseFirestore db;

private RecyclerView recyclerViewArticles;

private ArticleAdapter articleAdapter;

private ProgressBar progressBarArticles;

private ArticlesManager articleManager;

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

articleAdapter.setOnArticleClickListener(this);

}

@Override

public void onArticleClick(int position) {

}

@Override

public void onArticleLongClick(int position) {

View itemView = recyclerViewArticles.findViewHolderForAdapterPosition(position).itemView;

showPopupMenu(itemView, position);

}

private void showPopupMenu(View view, int position) {

PopupMenu popupMenu = new PopupMenu(requireContext(), view);

popupMenu.inflate(R.menu.context\_menu\_update\_delete);

popupMenu.setOnMenuItemClickListener(item -> {

if (item.getItemId() == R.id.action\_edit) {

Articles currentArticleEdit = articleAdapter.getArticles().get(position);

Intent intent = new Intent(getActivity(), EditArticleActivity.class);

intent.putExtra("article\_id", currentArticleEdit.getArticle\_id());

intent.putExtra("title", currentArticleEdit.getTitle());

intent.putExtra("description", currentArticleEdit.getDescription());

intent.putExtra("articleText", currentArticleEdit.getArticleText());

intent.putExtra("photo\_URL", currentArticleEdit.getPhoto\_URL());

startActivity(intent);

return true;

} else if (item.getItemId() == R.id.action\_delete) {

Articles currentArticleDelete = articleAdapter.getArticles().get(position);

articleManager.deleteArticle(currentArticleDelete.getArticle\_id());

loadArticles();

return true;

} else {

return false;

}

});

popupMenu.setGravity(Gravity.RIGHT);

popupMenu.show();

}

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,

@Nullable Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_articles, container, false);

db = FirebaseFirestore.getInstance();

recyclerViewArticles = view.findViewById(R.id.recyclerViewArticles);

addArticleButton = view.findViewById(R.id.addArticleButton);

recyclerViewArticles.setLayoutManager(new LinearLayoutManager(getContext()));

progressBarArticles = view.findViewById(R.id.progressBarArticles);

articleAdapter = new ArticleAdapter(getContext(), new ArrayList<>());

recyclerViewArticles.setAdapter(articleAdapter);

articleManager = new ArticlesManager();

loadArticles();

FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();

if(Objects.equals(currentUser.getDisplayName(), "Administrator")){

addArticleButton.setVisibility(View.VISIBLE);

addArticleButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(getActivity(), AddArticleActivity.class));

getActivity().finish();

}

});

} else addArticleButton.setVisibility(View.GONE);

return view;

}

private void loadArticles() {

db.collection("articles")

.get()

.addOnCompleteListener(task -> {

if (task.isSuccessful()) {

List<Articles> articlesList = new ArrayList<>();

for (DocumentSnapshot document : task.getResult()) {

progressBarArticles.setVisibility(View.VISIBLE);

Articles article = document.toObject(Articles.class);

if (article != null) {

article.setArticle\_id(document.getId());

articlesList.add(article);

progressBarArticles.setVisibility(View.VISIBLE);

}

}

progressBarArticles.setVisibility(View.GONE);

articleAdapter.setArticles(articlesList);

} else {

progressBarArticles.setVisibility(View.GONE);

Toast.makeText(getContext(), "Ошибка загрузки статей", Toast.LENGTH\_SHORT).show();

}

});

}

@Override

public void onResume() {

super.onResume();

loadArticles();

}

}

5) ArticlesManager.java;

package com.example.MAU.Articles;

import android.util.Log;

import com.example.MAU.models.Articles;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

import java.util.List;

public class ArticlesManager {

private final String COLLECTION\_NAME = "articles";

private final FirebaseFirestore db = FirebaseFirestore.getInstance();

private final CollectionReference articlesCollection = db.collection(COLLECTION\_NAME);

// Получение всех статей

public void getAllArticles(final OnDataRetrievedListener listener) {

List<Articles> articlesList = new ArrayList<>();

articlesCollection

.get()

.addOnSuccessListener(queryDocumentSnapshots -> {

for (DocumentSnapshot snapshot : queryDocumentSnapshots) {

Articles article = snapshot.toObject(Articles.class);

article.setArticle\_id(snapshot.getId());

articlesList.add(article);

}

listener.onArticlesRetrieved(articlesList);

})

.addOnFailureListener(e -> {

Log.e("ArticlesManager", "Error getting documents: " + e.getMessage());

listener.onError(e.getMessage());

});

}

// Добавление новой статьи

public void addArticle(Articles article) {

articlesCollection.document()

.set(article)

.addOnSuccessListener(aVoid -> Log.d("ArticlesManager", "Article added successfully"))

.addOnFailureListener(e -> Log.e("ArticlesManager", "Error adding article: " + e.getMessage()));

}

// Обновление существующей статьи

public void updateArticle(Articles article) {

articlesCollection.document(article.getArticle\_id())

.set(article)

.addOnSuccessListener(aVoid -> Log.d("ArticlesManager", "Article updated successfully"))

.addOnFailureListener(e -> Log.e("ArticlesManager", "Error updating article: " + e.getMessage()));

}

// Удаление статьи

public void deleteArticle(String articleId) {

articlesCollection.document(articleId)

.delete()

.addOnSuccessListener(aVoid -> Log.d("ArticlesManager", "Article deleted successfully"))

.addOnFailureListener(e -> Log.e("ArticlesManager", "Error deleting article: " + e.getMessage()));

}

public interface OnDataRetrievedListener {

void onArticlesRetrieved(List<Articles> articles);

void onError(String errorMessage);

}

public void getCurrentArticle(String articleId, final OnArticleRetrievedListener listener) {

articlesCollection.document(articleId)

.get()

.addOnSuccessListener(documentSnapshot -> {

Articles article = documentSnapshot.toObject(Articles.class);

if (article != null) {

article.setArticle\_id(documentSnapshot.getId());

listener.onArticleRetrieved(article);

} else {

listener.onError("Article not found");

}

})

.addOnFailureListener(e -> {

Log.e("ArticlesManager", "Error getting current article: " + e.getMessage());

listener.onError(e.getMessage());

});

}

public interface OnArticleRetrievedListener {

void onArticleRetrieved(Articles article);

void onError(String errorMessage);

}

}

6) EditArticleActivity.java;

package com.example.MAU.Articles;

import android.content.Intent;

import android.os.Bundle;

import android.text.Editable;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.EditText;

import android.widget.Toast;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.Fragment;

import com.example.MAU.MainActivity;

import com.example.MAU.R;

import com.example.MAU.models.Articles;

public class EditArticleActivity extends AppCompatActivity {

private EditText titleEditText, descriptionEditText, articleTextEditText, photoURLEditText;

private Button editArticleButton;

private ImageButton backButton, boldButton, italicButton, paragraph, h1Button, h2Button, h3Button;

private Articles currentArticle;

private ArticlesManager articleManager;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_edit\_article);

// Инициализация полей EditText и кнопки

titleEditText = findViewById(R.id.titleEditText);

descriptionEditText = findViewById(R.id.descriptionEditText);

articleTextEditText = findViewById(R.id.articleTextEditText);

photoURLEditText = findViewById(R.id.photoURLEditText);

editArticleButton = findViewById(R.id.editArticleButton);

backButton = findViewById(R.id.backButton);

boldButton = findViewById(R.id.boldButton);

italicButton = findViewById(R.id.italicButton);

paragraph = findViewById(R.id.paragraph);

h1Button = findViewById(R.id.h1Button);

h2Button = findViewById(R.id.h2Button);

h3Button = findViewById(R.id.h3Button);

View.OnClickListener formatButtonClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

String tag = (String) v.getTag();

if (tag != null) {

int start = articleTextEditText.getSelectionStart();

int end = articleTextEditText.getSelectionEnd();

if (start == end){

Toast.makeText(EditArticleActivity.this, "Выделите текст", Toast.LENGTH\_SHORT).show();

} else {

String selectedText = articleTextEditText.getText().toString().substring(start, end);

String htmlText = String.format("<%s>%s</%s>", tag, selectedText, tag);

appendFormattedText(htmlText);

}

}

}

};

boldButton.setOnClickListener(formatButtonClickListener);

italicButton.setOnClickListener(formatButtonClickListener);

paragraph.setOnClickListener(formatButtonClickListener);

h1Button.setOnClickListener(formatButtonClickListener);

h2Button.setOnClickListener(formatButtonClickListener);

h3Button.setOnClickListener(formatButtonClickListener);

articleManager = new ArticlesManager();

String articleId = getIntent().getStringExtra("article\_id");

articleManager.getCurrentArticle(articleId, new ArticlesManager.OnArticleRetrievedListener() {

@Override

public void onArticleRetrieved(Articles article) {

currentArticle = article;

titleEditText.setText(currentArticle.getTitle());

descriptionEditText.setText(currentArticle.getDescription());

articleTextEditText.setText(currentArticle.getArticleText());

photoURLEditText.setText(currentArticle.getPhoto\_URL());

}

@Override

public void onError(String errorMessage) {

Log.e("EditArticleActivity", "Ошибка получения текущей статьи " + errorMessage);

Toast.makeText(EditArticleActivity.this, "Ошибка: " + errorMessage, Toast.LENGTH\_SHORT).show();

}

});

editArticleButton.setOnClickListener(v -> {

String editedTitle = titleEditText.getText().toString().trim();

String editedDescription = descriptionEditText.getText().toString().trim();

String editedArticleText = articleTextEditText.getText().toString().trim();

String editedPhotoURL = photoURLEditText.getText().toString().trim();

if (!editedTitle.isEmpty() && !editedDescription.isEmpty() && !editedArticleText.isEmpty()){

if (currentArticle != null) {

if (editedPhotoURL.isEmpty()){

editedPhotoURL = "<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS9PJscu_d_qoxDIWQnCEU-KF-6WpzI0hbMS-TmuUeaWw&s>";

}

currentArticle.setTitle(editedTitle);

currentArticle.setDescription(editedDescription);

currentArticle.setArticleText(editedArticleText);

currentArticle.setPhoto\_URL(editedPhotoURL);

articleManager.updateArticle(currentArticle);

}

} else {

Toast.makeText(EditArticleActivity.this, "Заполните все поля", Toast.LENGTH\_SHORT).show();

return;

}

finish();

});

backButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(getApplicationContext(), MainActivity.class));

finish();

}

});

}

private void appendFormattedText(String formattedText) {

Editable editable = articleTextEditText.getText();

int start = articleTextEditText.getSelectionStart();

int end = articleTextEditText.getSelectionEnd();

editable.replace(start, end, formattedText);

articleTextEditText.setSelection(start + formattedText.length());

}

public void onBackPressed() {

super.onBackPressed();

Intent intent = new Intent(this, MainActivity.class);

intent.putExtra("fragment", "articles");

intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_SINGLE\_TOP);

startActivity(intent);

finish();

}

}

7) Articles.java;

package com.example.MAU.models;

import java.io.Serializable;

import java.util.Date;

public class Articles implements Serializable{

public Articles() {}

private String article\_id;

private String title;

private String description;

public Articles(String title, String description, String articleText, String photo\_URL) {

this.title = title;

this.description = description;

this.articleText = articleText;

this.photo\_URL = photo\_URL;

}

public String getArticleText() {

return articleText;

}

public void setArticleText(String articleText) {

this.articleText = articleText;

}

private String articleText;

private String photo\_URL;

public String getArticle\_id() {

return article\_id;

}

public void setArticle\_id(String article\_id) {

this.article\_id = article\_id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getPhoto\_URL() {

return photo\_URL;

}

public void setPhoto\_URL(String photo\_URL) {

this.photo\_URL = photo\_URL;

}

}

8) Note.java;

package com.example.MAU.models;

import java.io.Serializable;

import java.util.Date;

public class Note implements Serializable {

private String note\_id;

private Date date;

private String description;

private String user\_id;

public Note() {}

public Note(Date date, String description, String user\_id) {

this.date = date;

this.description = description;

this.user\_id = user\_id;

}

public String getNote\_id() {

return note\_id;

}

public void setNote\_id(String note\_id) {

this.note\_id = note\_id;

}

public void setDate(Date date) {this.date = date;}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getUser\_id() {

return user\_id;

}

public void setUser\_id(String user\_id) {

this.user\_id = user\_id;

}

public Date getDate() {

return date;

}

}

9) Song.java;

package com.example.MAU.models;

public class Song {

private String song\_id;

private String title;

private String description;

private String image\_url;

private String song\_url;

public Song(){}

public Song(String title, String description, String image\_url, String song\_url) {

this.title = title;

this.description = description;

this.image\_url = image\_url;

this.song\_url = song\_url;

}

public String getSong\_id() {return song\_id;}

public void setSong\_id(String song\_id) {this.song\_id = song\_id;}

public String getTitle() {return title;}

public void setTitle(String title) {this.title = title;}

public String getDescription() {return description;}

public void setDescription(String description) {this.description = description;}

public String getImage\_url() {return image\_url;}

public void setImage\_url(String image\_url) {this.image\_url = image\_url;}

public String getSong\_url() {return song\_url;}

public void setSong\_url(String song\_url) {this.song\_url = song\_url;}

}

10) User.java;

package com.example.MAU.models;

public class User {

private String email;

private String uid;

private String displayName;

public User() {

}

public User(String email, String uid, String displayName) {

this.email = email;

this.uid = uid;

this.displayName = displayName;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getUid() {

return uid;

}

public void setUid(String uid) {

this.uid = uid;

}

public String getDisplayName() {

return displayName;

}

public void setDisplayName(String displayName) {

this.displayName = displayName;

}

}

11) AddNoteActivity.java;

package com.example.MAU.Notes;

import android.app.DatePickerDialog;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.EditText;

import android.widget.ImageButton;

import androidx.appcompat.app.AppCompatActivity;

import com.example.MAU.MainActivity;

import com.example.MAU.R;

import com.example.MAU.models.Note;

import com.google.firebase.auth.FirebaseAuth;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.Locale;

public class AddNoteActivity extends AppCompatActivity {

private EditText dateEditText;

private EditText descriptionEditText;

private Button addNoteButton;

private ImageButton backButton;

private NoteManager noteManager;

private Calendar calendar;

private DatePickerDialog.OnDateSetListener dateSetListener;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_add\_note);

FirebaseAuth auth = FirebaseAuth.getInstance();

dateEditText = findViewById(R.id.dateEditText);

descriptionEditText = findViewById(R.id.descriptionEditText);

addNoteButton = findViewById(R.id.addNoteButton);

backButton = findViewById(R.id.backButton);

noteManager = new NoteManager();

// Инициализация календаря

calendar = Calendar.getInstance();

// Установка слушателя на выбор даты

dateEditText.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new DatePickerDialog(AddNoteActivity.this, dateSetListener,

calendar.get(Calendar.YEAR), calendar.get(Calendar.MONTH),

calendar.get(Calendar.DAY\_OF\_MONTH)).show();

}

});

dateSetListener = new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {

calendar.set(Calendar.YEAR, year);

calendar.set(Calendar.MONTH, month);

calendar.set(Calendar.DAY\_OF\_MONTH, dayOfMonth);

updateDateEditText();

}

};

updateDateEditText();

addNoteButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String description = descriptionEditText.getText().toString();

Date date = calendar.getTime();

if(description.isEmpty()){

descriptionEditText.setError("Заполните это поле");

}

else {

Note note = new Note(date, description, auth.getCurrentUser().getUid());

noteManager.addNote(note);

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

startActivity(intent);

finish();

}

}

});

backButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

startActivity(intent);

finish();

}

});

}

private void updateDateEditText() {

String myFormat = "dd.MM.yyyy";

SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.getDefault());

dateEditText.setText(sdf.format(calendar.getTime()));

}

@Override

public void onBackPressed() {

super.onBackPressed();

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

startActivity(intent);

finish();

}

}

12) EditNoteActivity.java;

package com.example.MAU.Notes;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.EditText;

import android.widget.Toast;

import com.example.MAU.R;

import com.example.MAU.models.Note;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.Locale;

public class EditNoteActivity extends AppCompatActivity {

private NoteManager noteManager;

private EditText editTextDescription, editTextDate;

private Button buttonUpdate;

private Note currentNote;

private Calendar calendar;

private DatePickerDialog.OnDateSetListener dateSetListener;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_edit\_note);

noteManager = new NoteManager();

editTextDescription = findViewById(R.id.descriptionEditText);

editTextDate = findViewById(R.id.dateEditText);

buttonUpdate = findViewById(R.id.editNoteButton);

calendar = Calendar.getInstance();

editTextDate.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

new DatePickerDialog(EditNoteActivity.this, dateSetListener,

calendar.get(Calendar.YEAR), calendar.get(Calendar.MONTH),

calendar.get(Calendar.DAY\_OF\_MONTH)).show();

}

});

dateSetListener = new DatePickerDialog.OnDateSetListener() {

@Override

public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {

calendar.set(Calendar.YEAR, year);

calendar.set(Calendar.MONTH, month);

calendar.set(Calendar.DAY\_OF\_MONTH, dayOfMonth);

updateDateEditText();

}

};

Intent intent = getIntent();

if (intent != null && intent.hasExtra("note")) {

currentNote = (Note) intent.getSerializableExtra("note");

if (currentNote != null) {

editTextDescription.setText(currentNote.getDescription());

String formattedDate = intent.getStringExtra("formatted\_date");

editTextDate.setText(formattedDate);

}

}

buttonUpdate.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (currentNote != null) {

String updatedDescription = editTextDescription.getText().toString();

currentNote.setDescription(updatedDescription);

if (updatedDescription.isEmpty()) {

editTextDescription.setError("Заполните это поле");

return;

}

String updatedData = editTextDate.getText().toString();

SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy", Locale.getDefault());

try {

Date date = sdf.parse(updatedData);

currentNote.setDate(date);

} catch (ParseException e) {

e.printStackTrace();

}

noteManager.updateNote(currentNote);

finish();

}

}

});

}

private void updateDateEditText() {

String myFormat = "dd.MM.yyyy";

SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.getDefault());

editTextDate.setText(sdf.format(calendar.getTime()));

}

}

13) NoteAdapter.java;

package com.example.MAU.Notes;

import android.content.Context;

import android.content.Intent;

import android.view.ContextMenu;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.example.MAU.R;

import com.example.MAU.models.Note;

import java.text.SimpleDateFormat;

import java.util.List;

import java.util.Locale;

public class NoteAdapter extends RecyclerView.Adapter<NoteAdapter.NoteViewHolder> {

private Context context;

private List<Note> notes;

private int selectedPosition = -1;

public NoteAdapter(Context context, List<Note> notes) {

this.context = context;

this.notes = notes;

}

public List<Note> getNotes() {

return notes;

}

@NonNull

@Override

public NoteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(context).inflate(R.layout.list\_item\_note, parent, false);

return new NoteViewHolder(view);

}

public int getSelectedPosition() {

return selectedPosition;

}

@Override

public void onBindViewHolder(@NonNull NoteViewHolder holder, int position) {

Note currentNote = notes.get(position);

holder.textViewDescription.setText(currentNote.getDescription());

// Форматирование даты

SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM yyyy", Locale.getDefault());

String formattedDate = sdf.format(currentNote.getDate());

holder.textViewDate.setText(formattedDate);

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(context, NoteDetailsActivity.class);

intent.putExtra("note\_description", currentNote.getDescription());

intent.putExtra("note\_date", formattedDate);

context.startActivity(intent);

}

});

holder.itemView.setOnLongClickListener(new View.OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

selectedPosition = holder.getAdapterPosition();

return false;

}

});

holder.itemView.setOnCreateContextMenuListener((menu, v, menuInfo) -> {

MenuInflater inflater = new MenuInflater(context);

inflater.inflate(R.menu.context\_menu\_update\_delete, menu);

});

}

@Override

public int getItemCount() {

return notes.size();

}

public void setNotes(List<Note> notes) {

this.notes = notes;

notifyDataSetChanged();

}

public class NoteViewHolder extends RecyclerView.ViewHolder implements View.OnCreateContextMenuListener {

TextView textViewDate;

TextView textViewDescription;

public NoteViewHolder(@NonNull View itemView) {

super(itemView);

textViewDate = itemView.findViewById(R.id.textViewDate);

textViewDescription = itemView.findViewById(R.id.textViewDescription);

itemView.setOnCreateContextMenuListener(this);

}

@Override

public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {

menu.add(Menu.NONE, R.id.action\_edit, Menu.NONE, "Изменить");

menu.add(Menu.NONE, R.id.action\_delete, Menu.NONE, "Удалить");

}

}

}

14) NoteDetailsActivity.java;

package com.example.MAU.Notes;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageButton;

import android.widget.TextView;

import com.example.MAU.R;

public class NoteDetailsActivity extends AppCompatActivity {

ImageButton backButton;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_note\_details);

backButton = findViewById(R.id.backButton);

// Получение информации о заметке из Intent

Intent intent = getIntent();

String description = intent.getStringExtra("note\_description");

String date = intent.getStringExtra("note\_date");

// Установка инфы из карточки

TextView descriptionTextView = findViewById(R.id.textViewDescription);

TextView dateTextView = findViewById(R.id.textViewDate);

descriptionTextView.setText(description);

dateTextView.setText(date);

backButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

}

}

15) NoteManager.java;

package com.example.MAU.Notes;

import android.util.Log;

import com.example.MAU.models.Note;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

import java.util.List;

public class NoteManager {

private final String COLLECTION\_NAME = "notes";

private final FirebaseFirestore db = FirebaseFirestore.getInstance();

private final CollectionReference notesCollection = db.collection(COLLECTION\_NAME);

public void getNotesByUserID(String user\_id, final OnDataRetrievedListener listener) {

List<Note> notesList = new ArrayList<>();

notesCollection.whereEqualTo("user\_id", user\_id)

.get()

.addOnSuccessListener(queryDocumentSnapshots -> {

for (DocumentSnapshot snapshot : queryDocumentSnapshots) {

Note note = snapshot.toObject(Note.class);

note.setNote\_id(snapshot.getId());

notesList.add(note);

}

listener.onNotesRetrieved(notesList);

})

.addOnFailureListener(e -> {

Log.e("NoteManager", "Error getting documents: " + e.getMessage());

listener.onError(e.getMessage());

});

}

public void addNote(Note note) {

notesCollection.document()

.set(note)

.addOnSuccessListener(aVoid -> Log.d("NoteManager", "Note added successfully"))

.addOnFailureListener(e -> Log.e("NoteManager", "Error adding note: " + e.getMessage()));

}

public void updateNote(Note note) {

notesCollection.document(note.getNote\_id())

.set(note)

.addOnSuccessListener(aVoid -> Log.d("NoteManager", "Note updated successfully"))

.addOnFailureListener(e -> Log.e("NoteManager", "Error updating note: " + e.getMessage()));

}

public void deleteNote(String noteId) {

notesCollection.document(noteId)

.delete()

.addOnSuccessListener(aVoid -> Log.d("NoteManager", "Note deleted successfully"))

.addOnFailureListener(e -> Log.e("NoteManager", "Error deleting note: " + e.getMessage()));

}

public interface OnDataRetrievedListener {

void onNotesRetrieved(List<Note> notes);

void onError(String errorMessage);

}

public interface OnNoteClickListener {

void onNoteClick(Note note);

}

}

16) ProfileFragment.java;

package com.example.MAU.Notes;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.ContextMenu;

import android.view.LayoutInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.example.MAU.R;

import com.example.MAU.models.Note;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import java.util.Locale;

public class ProfileFragment extends Fragment {

private NoteManager noteManager;

private FirebaseAuth firebaseAuth;

private FirebaseFirestore db;

private RecyclerView recyclerViewNotes;

private NoteAdapter noteAdapter;

private ProgressBar progressBar;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,

@Nullable Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_profile, container, false);

firebaseAuth = FirebaseAuth.getInstance();

db = FirebaseFirestore.getInstance();

noteManager = new NoteManager();

// Настройка RecyclerView

recyclerViewNotes = view.findViewById(R.id.recyclerViewNotes);

recyclerViewNotes.setLayoutManager(new LinearLayoutManager(getContext()));

// Создание и установка адаптера

noteAdapter = new NoteAdapter(getContext(), new ArrayList<>());

recyclerViewNotes.setAdapter(noteAdapter);

progressBar = view.findViewById(R.id.progressBar);

// Получение текущего пользователя

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

TextView userEmail = view.findViewById(R.id.userEmail);

TextView userLogin = view.findViewById(R.id.userLogin);

userEmail.setText(user.getEmail());

userLogin.setText(user.getDisplayName());

loadNotes(user.getUid());

}

ImageButton addNoteButton = view.findViewById(R.id.addNoteButton);

addNoteButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(getActivity(), AddNoteActivity.class));

getActivity().finish();

}

});

return view;

}

// Создание контекстного меню

@Override

public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {

super.onCreateContextMenu(menu, v, menuInfo);

getActivity().getMenuInflater().inflate(R.menu.context\_menu\_update\_delete, menu);

}

// Обработка нажатий на элементы контекстного меню

@Override

public boolean onContextItemSelected(MenuItem item) {

int position = noteAdapter.getSelectedPosition();

if (position != RecyclerView.NO\_POSITION) {

if (item.getItemId() == R.id.action\_edit) {

Note currentNote = noteAdapter.getNotes().get(position);

Intent intent = new Intent(getContext(), EditNoteActivity.class);

intent.putExtra("note", currentNote);

SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy", Locale.getDefault());

String formattedDate = sdf.format(currentNote.getDate());

intent.putExtra("formatted\_date", formattedDate);

startActivity(intent);

return true;

} else if (item.getItemId() == R.id.action\_delete) {

Note currentNoteDelete = noteAdapter.getNotes().get(position);

noteManager.deleteNote(currentNoteDelete.getNote\_id());

loadNotes(firebaseAuth.getCurrentUser().getUid());

return true;

}

}

return super.onContextItemSelected(item);

}

// Метод для подгрузки заметок каждый раз когда открывается фрагмент

@Override

public void onResume() {

super.onResume();

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

loadNotes(user.getUid());

}

}

// Метод для подгрузки заметок

private void loadNotes(String userId) {

db.collection("notes")

.whereEqualTo("user\_id", userId)

.get()

.addOnCompleteListener(task -> {

if (task.isSuccessful()) {

progressBar.setVisibility(View.VISIBLE);

List<Note> notesList = new ArrayList<>();

for (DocumentSnapshot document : task.getResult()) {

Note note = document.toObject(Note.class);

if (note != null) {

note.setNote\_id(document.getId());

notesList.add(note);

}

}

progressBar.setVisibility(View.GONE);

noteAdapter.setNotes(notesList);

} else {

progressBar.setVisibility(View.GONE);

Toast.makeText(getContext(), "Ошибка загрузки заметок", Toast.LENGTH\_SHORT).show();

Log.d("ProfileFragment", "Error getting notes: ", task.getException());

}

});

}

}

17) NotificationSubscriptionManager.java;

package com.example.MAU.Notification;

import android.content.Context;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.messaging.FirebaseMessaging;

public class NotificationSubscriptionManager {

public static void subscribeToNotifications(Context context) {

FirebaseMessaging.getInstance().subscribeToTopic("Meditation")

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(Task<Void> task) {

if (task.isSuccessful()) {

Toast.makeText(context, "Вы успешно подписались на уведомления", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(context, "Ошибка подписки на уведомления", Toast.LENGTH\_SHORT).show();

}

}

});

}

public static void unsubscribeFromNotifications(Context context) {

FirebaseMessaging.getInstance().unsubscribeFromTopic("Meditation")

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(Task<Void> task) {

if (task.isSuccessful()) {

Toast.makeText(context, "Вы успешно отписались от уведомлений ", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(context, "Ошибка отписки от уведомлений", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

18) PushService.java;

package com.example.MAU.Notification;

import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Context;

import android.content.Intent;

import android.media.RingtoneManager;

import android.net.Uri;

import android.os.Build;

import android.util.Log;

import androidx.annotation.NonNull;

import androidx.core.app.NotificationCompat;

import com.example.MAU.MainActivity;

import com.example.MAU.R;

import com.google.firebase.messaging.FirebaseMessagingService;

import com.google.firebase.messaging.RemoteMessage;

public class PushService extends FirebaseMessagingService {

private static final String TAG = "PushService";

@Override

public void onNewToken(@NonNull String token) {

Log.d(TAG, "Refreshed token: " + token);

}

@Override

public void onMessageReceived(RemoteMessage remoteMessage) {

// TODO(developer): Handle FCM messages here.

if (remoteMessage.getNotification()!=null){

String notificationBody = remoteMessage.getNotification().getBody();

String notificationTitle = remoteMessage.getNotification().getTitle();

sendNotification (notificationTitle, notificationBody);

}

}

private void sendNotification(String title, String body) {

Intent intent = new Intent(this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

PendingIntent pendingIntent = PendingIntent.getActivity(this, 0 , intent,

PendingIntent.FLAG\_IMMUTABLE);

String channelId = "fcm\_default\_channel";

Uri defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

NotificationCompat.Builder notificationBuilder =

new NotificationCompat.Builder(this, channelId)

.setSmallIcon(R.drawable.app\_icon)

.setContentTitle(title)

.setContentText(body)

.setAutoCancel(true)

.setSound(defaultSoundUri)

.setContentIntent(pendingIntent);

NotificationManager notificationManager =

(NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

NotificationChannel channel = new NotificationChannel(channelId,

"Channel human readable title",

NotificationManager.IMPORTANCE\_DEFAULT);

notificationManager.createNotificationChannel(channel);

}

notificationManager.notify(0 , notificationBuilder.build());

}

}

19) AddSongActivity.java;

package com.example.MAU.Player;

import android.app.Activity;

import android.content.Intent;

import android.database.Cursor;

import android.net.Uri;

import android.os.Bundle;

import android.provider.OpenableColumns;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.example.MAU.MainActivity;

import com.example.MAU.R;

import com.example.MAU.models.Song;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.Query;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;

import java.util.Arrays;

import java.util.List;

import java.util.UUID;

public class AddSongActivity extends AppCompatActivity {

private static final int PICK\_AUDIO\_REQUEST = 1;

private static final List<String> ALLOWED\_MIME\_TYPES = Arrays.asList("audio/mpeg", "audio/aac", "audio/wav", "audio/ogg");

private Button buttonChooseFile, addSongButton;

private ImageButton backButton;

private EditText titleEditText, descriptionEditText, photoUrlEditText;

private ProgressBar uploadProgressBar;

private Uri audioUri;

private FirebaseStorage storage;

private FirebaseFirestore firestore;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_add\_song);

storage = FirebaseStorage.getInstance();

firestore = FirebaseFirestore.getInstance();

buttonChooseFile = findViewById(R.id.buttonChooseFile);

addSongButton = findViewById(R.id.addSongButton);

backButton = findViewById(R.id.backButton);

titleEditText = findViewById(R.id.titleEditText);

descriptionEditText = findViewById(R.id.descriptionEditText);

photoUrlEditText = findViewById(R.id.photoUrlEditText);

uploadProgressBar = findViewById(R.id.uploadProgressBar);

buttonChooseFile.setOnClickListener(v -> openFileChooser());

addSongButton.setOnClickListener(v -> uploadFile());

backButton.setOnClickListener(v -> onBackPressed());

}

private void openFileChooser() {

Intent intent = new Intent(Intent.ACTION\_GET\_CONTENT);

intent.setType("audio/\*");

startActivityForResult(intent, PICK\_AUDIO\_REQUEST);

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == PICK\_AUDIO\_REQUEST && resultCode == Activity.RESULT\_OK && data != null && data.getData() != null) {

audioUri = data.getData();

if (!isAllowedFileType(audioUri)) {

Toast.makeText(this, "Выбранный файл имеет недопустимый формат", Toast.LENGTH\_SHORT).show();

audioUri = null; // Reset the URI if the file format is not allowed

} else {

String fileName = getFileName(audioUri);

buttonChooseFile.setText(fileName);

}

}

}

private boolean isAllowedFileType(Uri uri) {

String mimeType = getContentResolver().getType(uri);

return ALLOWED\_MIME\_TYPES.contains(mimeType);

}

private String getFileName(Uri uri) {

String fileName = null;

Cursor cursor = getContentResolver().query(uri, null, null, null, null);

if (cursor != null && cursor.moveToFirst()) {

int nameIndex = cursor.getColumnIndex(OpenableColumns.DISPLAY\_NAME);

if (nameIndex != -1) {

fileName = cursor.getString(nameIndex);

}

cursor.close();

}

return fileName;

}

private void uploadFile() {

String title = titleEditText.getText().toString().trim();

String description = descriptionEditText.getText().toString().trim();

String imageUrl = photoUrlEditText.getText().toString().trim();

if (title.isEmpty() || description.isEmpty()) {

Toast.makeText(this, "Пожалуйста, заполните все поля", Toast.LENGTH\_SHORT).show();

return;

}

if (audioUri == null) {

Toast.makeText(this, "Пожалуйста, выберите файл", Toast.LENGTH\_SHORT).show();

return;

}

checkIfTitleExists(title, () -> {

uploadProgressBar.setVisibility(ProgressBar.VISIBLE);

String fileName = UUID.randomUUID().toString();

StorageReference storageReference = storage.getReference().child("songs/" + fileName);

storageReference.putFile(audioUri)

.addOnSuccessListener(taskSnapshot -> storageReference.getDownloadUrl().addOnSuccessListener(uri -> {

uploadProgressBar.setVisibility(ProgressBar.GONE);

String songUrl = uri.toString();

saveSongToFirestore(title, description, imageUrl, songUrl);

}))

.addOnFailureListener(e -> {

uploadProgressBar.setVisibility(ProgressBar.GONE);

Toast.makeText(AddSongActivity.this, "Ошибка загрузки файла", Toast.LENGTH\_SHORT).show();

});

});

}

private void checkIfTitleExists(String title, Runnable onSuccess) {

Query titleQuery = firestore.collection("songs")

.whereEqualTo("title", title)

.limit(1);

titleQuery.get().addOnSuccessListener(queryDocumentSnapshots -> {

if (!queryDocumentSnapshots.isEmpty()) {

Toast.makeText(AddSongActivity.this, "Мелодия с таким названием уже существует", Toast.LENGTH\_SHORT).show();

} else {

onSuccess.run();

}

}).addOnFailureListener(e -> {

Toast.makeText(AddSongActivity.this, "Ошибка проверки существования названия: " + e.getMessage(), Toast.LENGTH\_SHORT).show();

});

}

private void saveSongToFirestore(String title, String description, String imageUrl, String songUrl) {

if (imageUrl.isEmpty()) {

imageUrl = "<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS9PJscu_d_qoxDIWQnCEU-KF-6WpzI0hbMS-TmuUeaWw&s>";

}

Song song = new Song(title, description, imageUrl, songUrl);

firestore.collection("songs").add(song)

.addOnCompleteListener(task -> {

if (task.isSuccessful()) {

Toast.makeText(AddSongActivity.this, "Мелодия успешно добавлена", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(getApplicationContext(), MainActivity.class));

finish();

} else {

Toast.makeText(AddSongActivity.this, "Ошибка добавления мелодии", Toast.LENGTH\_SHORT).show();

}

})

.addOnFailureListener(e -> Toast.makeText(AddSongActivity.this, "Ошибка: " + e.getMessage(), Toast.LENGTH\_SHORT).show());

}

@Override

public void onBackPressed() {

super.onBackPressed();

Intent intent = new Intent(this, MainActivity.class);

intent.putExtra("fragment", "songs");

intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_SINGLE\_TOP);

startActivity(intent);

finish();

}

}

20) HomeFragment.java;

package com.example.MAU.Player;// HomeFragment.java

import android.content.Intent;

import android.os.Bundle;

import android.view.Gravity;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageButton;

import android.widget.PopupMenu;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.example.MAU.R;

import com.example.MAU.models.Song;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import java.util.ArrayList;

import java.util.List;

public class HomeFragment extends Fragment {

private FirebaseAuth firebaseAuth;

private FirebaseFirestore db;

private FirebaseStorage storage;

private RecyclerView recyclerViewSongs;

private SongAdapter songAdapter;

private ProgressBar progressBar;

private ImageButton addSongButton;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,

@Nullable Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_home, container, false);

firebaseAuth = FirebaseAuth.getInstance();

storage = FirebaseStorage.getInstance();

db = FirebaseFirestore.getInstance();

addSongButton = view.findViewById(R.id.addSongButton);

progressBar = view.findViewById(R.id.progressBar);

recyclerViewSongs = view.findViewById(R.id.recyclerViewSongs);

recyclerViewSongs.setLayoutManager(new LinearLayoutManager(getContext()));

songAdapter = new SongAdapter(getContext(), new ArrayList<>());

recyclerViewSongs.setAdapter(songAdapter);

loadSongs();

songAdapter.setOnItemLongClickListener(this::showPopupMenu);

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

TextView loginTextView = view.findViewById(R.id.loginTextView);

loginTextView.setText(user.getDisplayName());

}

if (user.getDisplayName().equals("Administrator")){

addSongButton.setVisibility(View.VISIBLE);

addSongButton.setOnClickListener(v -> {

startActivity(new Intent(getActivity(), AddSongActivity.class));

getActivity().finish();

});

} else {

addSongButton.setVisibility(View.GONE);

}

return view;

}

private void showPopupMenu(View view, int position) {

PopupMenu popupMenu = new PopupMenu(requireContext(), view);

popupMenu.inflate(R.menu.context\_menu\_delete);

popupMenu.setGravity(Gravity.RIGHT);

popupMenu.setOnMenuItemClickListener(item -> {

if (item.getItemId() == R.id.action\_delete) {

deleteSong(position);

return true;

} else {

return false;

}

});

popupMenu.show();

}

private void deleteSong(int position) {

Song song = songAdapter.getSongs().get(position);

StorageReference songRef = storage.getReferenceFromUrl(song.getSong\_url());

songRef.delete().addOnSuccessListener(aVoid -> {

db.collection("songs").document(song.getSong\_id())

.delete()

.addOnSuccessListener(aVoid1 -> {

Toast.makeText(getContext(), "Песня удалена", Toast.LENGTH\_SHORT).show();

songAdapter.getSongs().remove(position);

songAdapter.notifyItemRemoved(position);

})

.addOnFailureListener(e -> Toast.makeText(getContext(), "Ошибка удаления трека из базы данных", Toast.LENGTH\_SHORT).show());

}).addOnFailureListener(e -> Toast.makeText(getContext(), "Ошибка удаления трека из хранилища", Toast.LENGTH\_SHORT).show());

}

private void loadSongs() {

db.collection("songs")

.get()

.addOnCompleteListener(task -> {

if (task.isSuccessful()) {

List<Song> songList = new ArrayList<>();

for (DocumentSnapshot document : task.getResult()) {

progressBar.setVisibility(View.VISIBLE);

Song song = document.toObject(Song.class);

if (song != null) {

song.setSong\_id(document.getId());

songList.add(song);

}

}

progressBar.setVisibility(View.GONE);

songAdapter.setSongs(songList);

} else {

progressBar.setVisibility(View.GONE);

Toast.makeText(getContext(), "Ошибка загрузки треков", Toast.LENGTH\_SHORT).show();

}

});

}

}

21) PlayerActivity.java;

package com.example.MAU.Player;

import android.content.DialogInterface;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.ProgressBar;

import android.widget.TextView;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import com.example.MAU.R;

import com.squareup.picasso.Picasso;

public class PlayerActivity extends AppCompatActivity {

private MediaPlayer mediaPlayer;

private TextView progressTextView, songTitle, songDescription, finishNowLink;

private ProgressBar circleProgressBar;

private ImageButton playPauseButton;

private ImageView songImageView;

private boolean isPlaying = true;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_player);

String songUrl = getIntent().getStringExtra("songUrl");

String title = getIntent().getStringExtra("title");

String description = getIntent().getStringExtra("description");

String image\_url = getIntent().getStringExtra("image\_url");

progressTextView = findViewById(R.id.tvProgressText);

circleProgressBar = findViewById(R.id.circleProgressBar);

songTitle = findViewById(R.id.songTitle);

songDescription = findViewById(R.id.songDescription);

songImageView = findViewById(R.id.songImageView);

finishNowLink = findViewById(R.id.finishNowLink);

playPauseButton = findViewById(R.id.playPauseButton);

playPauseButton.setImageResource(R.drawable.pause\_button\_ic);

songTitle.setText(title);

songDescription.setText(description);

Picasso.get()

.load(image\_url)

.into(songImageView);

mediaPlayer = new MediaPlayer();

try {

mediaPlayer.setDataSource(songUrl);

mediaPlayer.prepare();

mediaPlayer.start();

} catch (Exception e) {

e.printStackTrace();

}

startProgressUpdate();

playPauseButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (isPlaying) {

mediaPlayer.pause();

playPauseButton.setImageResource(R.drawable.play\_button\_for\_player\_ic);

} else {

mediaPlayer.start();

playPauseButton.setImageResource(R.drawable.pause\_button\_ic);

}

isPlaying = !isPlaying;

}

});

finishNowLink.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

AlertDialog.Builder builder = new AlertDialog.Builder(PlayerActivity.this);

builder.setTitle("Завершить сейчас?");

builder.setMessage("Вы уверены, что хотите завершить воспроизведение и вернуться на главный экран?");

builder.setPositiveButton("Да", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

mediaPlayer.stop();

mediaPlayer.release();

mediaPlayer = null;

isPlaying = false;

finish();

}

});

builder.setNegativeButton("Отмена", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

}

});

AlertDialog dialog = builder.create();

dialog.show();

}

});

mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {

@Override

public void onCompletion(MediaPlayer mp) {

showFinishDialog();

}

});

}

private void showFinishDialog() {

AlertDialog.Builder builder = new AlertDialog.Builder(PlayerActivity.this);

builder.setTitle("Медитация завершена");

builder.setMessage("Уделяйте больше времени своему ментальному здоровью. Мы будем напоминать вам об этом");

builder.setPositiveButton("Выйти на главный экран", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

mediaPlayer.stop();

mediaPlayer.release();

mediaPlayer = null;

isPlaying = false;

finish();

}

});

AlertDialog dialog = builder.create();

dialog.show();

}

private void startProgressUpdate() {

new Thread(() -> {

while (mediaPlayer != null) {

if (mediaPlayer.isPlaying()) {

runOnUiThread(() -> {

int currentPosition = mediaPlayer.getCurrentPosition();

int totalDuration = mediaPlayer.getDuration();

int progress = (int) (currentPosition \* 100.0 / totalDuration);

int secondsRemaining = (totalDuration - currentPosition) / 1000;

String timeRemaining = String.format("%02d:%02d", secondsRemaining / 60, secondsRemaining % 60);

circleProgressBar.setProgress(progress);

progressTextView.setText(timeRemaining);

});

}

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}).start();

}

@Override

protected void onDestroy() {

super.onDestroy();

if (mediaPlayer != null) {

mediaPlayer.release();

mediaPlayer = null;

}

}

@Override

public void onBackPressed() {

super.onBackPressed();

mediaPlayer.stop();

mediaPlayer.release();

mediaPlayer = null;

isPlaying = false;

finish();

}

}

22) SongAdapter.java;

package com.example.MAU.Player;

import android.content.Context;

import android.content.Intent;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.PopupMenu;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.example.MAU.R;

import com.example.MAU.models.Song;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.squareup.picasso.Picasso;

import java.util.List;

import java.util.Objects;

public class SongAdapter extends RecyclerView.Adapter<SongAdapter.SongViewHolder> {

private Context context;

private List<Song> songList;

private OnItemLongClickListener onItemLongClickListener;

// Interface for long-click events

public interface OnItemLongClickListener {

void onItemLongClick(View view, int position);

}

public void setOnItemLongClickListener(OnItemLongClickListener listener) {

this.onItemLongClickListener = listener;

}

public SongAdapter(Context context, List<Song> songList) {

this.context = context;

this.songList = songList;

}

@NonNull

@Override

public SongViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(context).inflate(R.layout.list\_item\_song, parent, false);

return new SongViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull SongViewHolder holder, int position) {

Song currentSong = songList.get(position);

holder.titleTextView.setText(currentSong.getTitle());

holder.descriptionTextView.setText(currentSong.getDescription());

Picasso.get().load(currentSong.getImage\_url()).into(holder.imageViewPhoto);

holder.itemView.setOnClickListener(v -> {

Intent intent = new Intent(context, PlayerActivity.class);

intent.putExtra("title", currentSong.getTitle());

intent.putExtra("description", currentSong.getDescription());

intent.putExtra("image\_url", currentSong.getImage\_url());

intent.putExtra("songUrl", currentSong.getSong\_url());

context.startActivity(intent);

});

FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();

if(Objects.equals(currentUser.getDisplayName(), "Administrator")){

holder.itemView.setOnLongClickListener(v -> {

if (onItemLongClickListener != null) {

onItemLongClickListener.onItemLongClick(holder.itemView, position);

}

return true;

});

}

}

@Override

public int getItemCount() {

return songList.size();

}

public void setSongs(List<Song> songList) {

this.songList = songList;

notifyDataSetChanged();

}

public List<Song> getSongs() {

return songList;

}

public static class SongViewHolder extends RecyclerView.ViewHolder {

TextView titleTextView, descriptionTextView;

ImageView imageViewPhoto;

public SongViewHolder(@NonNull View itemView) {

super(itemView);

titleTextView = itemView.findViewById(R.id.textViewTitle);

descriptionTextView = itemView.findViewById(R.id.textViewDescription);

imageViewPhoto = itemView.findViewById(R.id.imageViewPhoto);

}

}

}

23) LoginManager.java;

package com.example.MAU;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.util.Patterns;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.FirebaseApp;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class LoginManager extends AppCompatActivity {

EditText emailEditText, passwordEditText;

Button loginButton;

ProgressBar progressBar;

TextView clickToReg2;

FirebaseAuth firebaseAuth;

@Override

public void onStart() {

super.onStart();

setContentView(R.layout.activity\_login);

SharedPreferences sharedPreferences = getSharedPreferences("appPreferences", MODE\_PRIVATE);

boolean isFirstStart = sharedPreferences.getBoolean("isFirstStart", true);

if (isFirstStart) {

FirebaseAuth.getInstance().signOut();

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putBoolean("isFirstStart", false);

editor.apply();

}

if (isUserLoggedIn()) {

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

startActivity(intent);

finish();

}

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

emailEditText = findViewById(R.id.emailEditText);

passwordEditText = findViewById(R.id.passwordEditText);

progressBar = findViewById(R.id.progressBar);

loginButton = findViewById(R.id.loginButton);

clickToReg2 = findViewById(R.id.clickToReg2);

firebaseAuth = FirebaseAuth.getInstance();

clickToReg2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), RegistrationManager.class);

startActivity(intent);

finish();

}

});

loginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

progressBar.setVisibility(View.VISIBLE);

String email = emailEditText.getText().toString().trim();

String password = passwordEditText.getText().toString().trim();

loginUser(email, password);

}

});

}

private void loginUser(String email, String password) {

firebaseAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

startActivity(intent);

finish();

progressBar.setVisibility(View.GONE);

} else {

emailEditText.setError("Неверный адрес электронной почты или пароль");

progressBar.setVisibility(View.GONE);

}

}

});

}

private boolean isUserLoggedIn() {

FirebaseAuth auth = FirebaseAuth.getInstance();

FirebaseUser user = auth.getCurrentUser();

if (user != null) {

return true;

} else {

return false;

}

}

}

24) MainActivity.java;

package com.example.MAU;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.Fragment;

import com.example.MAU.Articles.ArticlesFragment;

import com.example.MAU.Notes.ProfileFragment;

import com.example.MAU.Player.HomeFragment;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {

BottomNavigationView bottomNavigationView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

bottomNavigationView = findViewById(R.id.bottomNavbar);

bottomNavigationView.setSelectedItemId(R.id.profile\_nav);

SetFragment(new ProfileFragment());

if (getIntent().hasExtra("fragment")) {

String fragmentName = getIntent().getStringExtra("fragment");

if (fragmentName.equals("articles")) {

getSupportFragmentManager().beginTransaction().replace(R.id.fragment\_container, new ArticlesFragment()).commit();

} else if (fragmentName.equals("songs")) {

getSupportFragmentManager().beginTransaction().replace(R.id.fragment\_container, new HomeFragment()).commit();

}

} else {

}

bottomNavigationView.setOnNavigationItemSelectedListener(item -> {

int itemId = item.getItemId();

if (itemId == R.id.profile\_nav) {

SetFragment(new ProfileFragment());

return true;

} else if (itemId == R.id.settings\_nav) {

SetFragment(new SettingsFragment());

return true;

} else if (itemId == R.id.articles\_nav) {

SetFragment(new ArticlesFragment());

return true;

}

else if (itemId == R.id.home\_nav) {

SetFragment(new HomeFragment());

return true;

}

return false;

});

}

public void SetFragment(Fragment fragment) {

getSupportFragmentManager().beginTransaction()

.replace(R.id.fragment\_container, fragment, null)

.commit();

}

}

25) RegistrationManager.java;

package com.example.MAU;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Patterns;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import com.example.MAU.models.User;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.auth.UserProfileChangeRequest;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.regex.Pattern;

public class RegistrationManager extends AppCompatActivity {

EditText emailEditText, passwordEditText, loginEditText;

Button authButton, chooseImageButton;

ProgressBar progressBar;

TextView clickToLogin;

FirebaseAuth firebaseAuth;

FirebaseFirestore db = FirebaseFirestore.getInstance();

CollectionReference userRef = db.collection("users");

private static final Pattern LOGIN\_PATTERN = Pattern.compile("^(?=.\*[A-Z])[A-Za-z]{6,18}$");

private static final Pattern PASSWORD\_PATTERN = Pattern.compile("^(?=.\*[0-9])(?=.\*[!@#$%^&\*])[A-Za-z0-9!@#$%^&\*]{8,}$");

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_registration);

emailEditText = findViewById(R.id.emailEditText);

passwordEditText = findViewById(R.id.passwordEditText);

loginEditText = findViewById(R.id.loginEditText);

progressBar = findViewById(R.id.progressBar);

authButton = findViewById(R.id.authButton);

clickToLogin = findViewById(R.id.clickToLogin);

firebaseAuth = FirebaseAuth.getInstance();

clickToLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), LoginManager.class);

startActivity(intent);

finish();

}

});

authButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

progressBar.setVisibility(View.VISIBLE);

String email = emailEditText.getText().toString().trim();

String password = passwordEditText.getText().toString().trim();

String login = loginEditText.getText().toString().trim();

if (!Patterns.EMAIL\_ADDRESS.matcher(email).matches()) {

emailEditText.setError("Неверный формат адреса электронной почты");

progressBar.setVisibility(View.GONE);

} else if (!PASSWORD\_PATTERN.matcher(password).matches()) {

passwordEditText.setError("Пароль должен быть не менее 8 символов, содержать хотя бы одну цифру и один спецсимвол");

progressBar.setVisibility(View.GONE);

} else if (!LOGIN\_PATTERN.matcher(login).matches()) {

loginEditText.setError("Логин должен быть не менее 6 и не более 18 символов, содержать только латинские буквы и хотя бы одну заглавную букву");

progressBar.setVisibility(View.GONE);

} else {

registrationUser(email, password, login);

}

}

});

}

private void registrationUser(String email, String password, String displayName) {

firebaseAuth.createUserWithEmailAndPassword(email, password)

.addOnSuccessListener(new OnSuccessListener<AuthResult>() {

@Override

public void onSuccess(AuthResult authResult) {

FirebaseUser user = firebaseAuth.getCurrentUser();

UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()

.setDisplayName(displayName)

.build();

user.updateProfile(profileUpdates)

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

User newUser = new User(user.getEmail(), user.getUid(), displayName);

userRef.add(newUser).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

progressBar.setVisibility(View.GONE);

Toast.makeText(RegistrationManager.this, "Пользователь " + user.getEmail() + " зарегистрирован!", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(getApplicationContext(), WelcomeActivity.class);

startActivity(intent);

finish();

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

progressBar.setVisibility(View.GONE);

Toast.makeText(RegistrationManager.this, e.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

} else {

Toast.makeText(RegistrationManager.this, "Ошибка установки имени пользователя", Toast.LENGTH\_SHORT).show();

}

}

});

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

progressBar.setVisibility(View.GONE);

Toast.makeText(RegistrationManager.this, e.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

}

}

26) SettingsFragment.java;

package com.example.MAU;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import androidx.annotation.NonNull;

import androidx.core.app.NotificationManagerCompat;

import androidx.fragment.app.Fragment;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.CompoundButton;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.LinearLayout;

import android.widget.Switch;

import android.widget.TextView;

import android.widget.Toast;

import com.example.MAU.Notification.NotificationSubscriptionManager;

import com.example.MAU.models.User;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.auth.UserProfileChangeRequest;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.firebase.messaging.FirebaseMessaging;

import java.util.Objects;

public class SettingsFragment extends Fragment {

private static final String SWITCH\_STATE\_KEY = "switch\_state";

private static final String TAG = "PushService";

private FirebaseAuth firebaseAuth;

private FirebaseFirestore db;

private LinearLayout deleteAccountLayout, resetLoginLayout;

private View deleteAccountSeparator, resetLoginSeparator;

private ImageButton logoutButton, deleteAccountButton, resetPasswordButton, resetLoginButton;

private SharedPreferences sharedPreferences;

private Switch noticeSwitch;

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_settings, container, false);

firebaseAuth = FirebaseAuth.getInstance();

logoutButton = view.findViewById(R.id.logoutButton);

db = FirebaseFirestore.getInstance();

FirebaseUser currentUser = firebaseAuth.getCurrentUser();

if (currentUser == null) {

Intent intent = new Intent(getContext(), LoginManager.class);

startActivity(intent);

getActivity().finish();

return view;

}

deleteAccountButton = view.findViewById(R.id.deleteAccountButton);

deleteAccountLayout = view.findViewById(R.id.deleteAccountLayout);

deleteAccountSeparator = view.findViewById(R.id.deleteAccountSeparator);

resetPasswordButton = view.findViewById(R.id.resetPasswordButton);

resetLoginButton = view.findViewById(R.id.resetLoginButton);

resetLoginLayout = view.findViewById(R.id.resetLoginLayout);

resetLoginSeparator = view.findViewById(R.id.resetLoginSeparator);

noticeSwitch = view.findViewById(R.id.noticeSwitch);

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

TextView userEmail = view.findViewById(R.id.userEmail);

TextView userLogin = view.findViewById(R.id.userLogin);

userEmail.setText(user.getEmail());

userLogin.setText(user.getDisplayName());

}

if (Objects.equals(user.getDisplayName(), "Administrator")) {

deleteAccountLayout.setVisibility(View.GONE);

resetLoginLayout.setVisibility(View.GONE);

deleteAccountSeparator.setVisibility(View.GONE);

resetLoginSeparator.setVisibility(View.GONE);

}

// Выход из системы

logoutButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

firebaseAuth.signOut();

Intent intent = new Intent(getContext(), LoginManager.class);

startActivity(intent);

getActivity().finish();

}

});

// Удаление аккаунта (кнопка скрыта у администратора)

deleteAccountButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Модальное окно удаления аккаунта и всех данных связанных с ним

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

builder.setTitle("Подтвердите удаление аккаунта");

builder.setMessage("Вы уверены, что хотите удалить свой аккаунт? Это действие нельзя отменить.");

builder.setPositiveButton("Удалить", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

if (user != null) {

deleteUserAndData(user);

}

}

});

builder.setNegativeButton("Отмена", null);

builder.show();

}

});

// Изменение имени пользователя

resetLoginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

showChangeUsernameDialog();

}

});

// Изменение пароля пользователя

resetPasswordButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

showChangePasswordDialog();

}

});

sharedPreferences = requireActivity().getSharedPreferences("MyPrefs", Context.MODE\_PRIVATE);

noticeSwitch.setChecked(sharedPreferences.getBoolean(SWITCH\_STATE\_KEY, false));

noticeSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {

boolean notificationsEnabled = areNotificationsEnabled(getContext());

if (notificationsEnabled) {

if (isChecked) {

NotificationSubscriptionManager.subscribeToNotifications(getContext());

} else {

NotificationSubscriptionManager.unsubscribeFromNotifications(getContext());

}

} else {

noticeSwitch.setChecked(false);

Toast.makeText(getContext(), "Включите уведомления в настройках вашего устройства", Toast.LENGTH\_SHORT).show();

}

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putBoolean(SWITCH\_STATE\_KEY, isChecked);

editor.apply();

}

});

return view;

}

private boolean areNotificationsEnabled(Context context) {

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(context);

return notificationManager.areNotificationsEnabled();

}

private void deleteUserAndData(FirebaseUser user) {

user.delete().addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void unused) {

firebaseAuth.signOut();

deleteUserDataFromFirestore(user.getUid());

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Log.e("err", e.getMessage());

}

});

}

private void deleteUserDataFromFirestore(String userId) {

FirebaseFirestore db = FirebaseFirestore.getInstance();

db.collection("users")

.whereEqualTo("uid", userId)

.get()

.addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {

@Override

public void onSuccess(QuerySnapshot queryDocumentSnapshots) {

for (DocumentSnapshot documentSnapshot : queryDocumentSnapshots) {

deleteUserFb(documentSnapshot);

}

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

}

});

}

private void deleteUserFb(DocumentSnapshot doc) {

User delUser = doc.toObject(User.class);

db.collection("users")

.document(doc.getId())

.delete()

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

deleteNotesByUserId(delUser.getUid());

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Log.e("SettingsFragment", "Ошибка удаления документа пользователя: " + e.getMessage());

}

});

}

private void deleteNotesByUserId(String uid) {

db.collection("notes").whereEqualTo("user\_id", uid)

.get()

.addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {

@Override

public void onSuccess(QuerySnapshot queryDocumentSnapshots) {

for (DocumentSnapshot doc : queryDocumentSnapshots) {

delAllNotes(doc);

}

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

}

});

Intent intent = new Intent(getContext(), LoginManager.class);

startActivity(intent);

getActivity().finish();

}

private void delAllNotes(DocumentSnapshot doc) {

db.collection("notes")

.document(doc.getId())

.delete()

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void unused) {

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(getContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

}

private void showChangeUsernameDialog() {

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity(), R.style.CustomAlertDialog);

final EditText input = new EditText(getContext());

input.setHint("Введите новое имя пользователя");

builder.setView(input);

builder.setPositiveButton("Сохранить", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

String newUsername = input.getText().toString().trim();

if (!newUsername.isEmpty()) {

updateUsername(newUsername);

}

}

});

builder.setNegativeButton("Отмена", null);

builder.show();

}

private void updateUsername(final String newUsername) {

if (!validateUsername(newUsername)) {

Toast.makeText(getContext(), "Логин должен быть больше 6 символов, только латинские буквы и содержать хотя бы одну заглавную букву", Toast.LENGTH\_SHORT).show();

return;

}

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()

.setDisplayName(newUsername)

.build();

user.updateProfile(profileUpdates)

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

Log.d("SettingsFragment", "User profile updated.");

// Обновление Firestore

updateUsernameInFirestore(user.getUid(), newUsername);

} else {

Log.e("SettingsFragment", "Error updating user profile: " + task.getException());

Toast.makeText(getContext(), "Ошибка обновления имени пользователя", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

private void updateUsernameInFirestore(String userId, final String newUsername) {

db.collection("users").whereEqualTo("uid", userId).get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

for (DocumentSnapshot document : task.getResult()) {

db.collection("users").document(document.getId())

.update("displayName", newUsername)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Log.d("SettingsFragment", "Username updated in Firestore.");

Toast.makeText(getContext(), "Имя пользователя обновлено", Toast.LENGTH\_SHORT).show();

// Обновление отображения

TextView userLogin = getView().findViewById(R.id.userLogin);

userLogin.setText(newUsername);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Log.e("SettingsFragment", "Error updating username in Firestore: " + e.getMessage());

Toast.makeText(getContext(), "Ошибка обновления имени пользователя в базе данных", Toast.LENGTH\_SHORT).show();

}

});

}

} else {

Log.e("SettingsFragment", "Error getting user documents: " + task.getException());

}

}

});

}

private void showChangePasswordDialog() {

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity(), R.style.CustomAlertDialog);

final EditText input = new EditText(getContext());

input.setHint("Введите новый пароль");

builder.setView(input);

builder.setPositiveButton("Сохранить", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

String newPassword = input.getText().toString().trim();

if (!newPassword.isEmpty()) {

if (validatePassword(newPassword)) {

updatePassword(newPassword);

} else {

Toast.makeText(getContext(), "Пароль должен быть не менее 8 символов, содержать хотя бы одну цифру и специальный символ", Toast.LENGTH\_SHORT).show();

}

}

}

});

builder.setNegativeButton("Отмена", null);

builder.show();

}

private void updatePassword(String newPassword) {

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

user.updatePassword(newPassword).addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

Toast.makeText(getContext(), "Пароль обновлен", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(getContext(), "Ошибка обновления пароля", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

private boolean validateUsername(String username) {

return username.length() > 6 && username.matches(".\*[A-Z].\*") && username.matches("[a-zA-Z]+");

}

private boolean validatePassword(String password) {

return password.length() >= 8 && password.matches(".\*[\\d.\*](file:///\\d.*)") && password.matches(".\*[!@#$%^&\*+=?-].\*");

}

}

27) WelcomeActivity.java;

package com.example.MAU;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class WelcomeActivity extends AppCompatActivity {

Button beginButton;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_welcome);

beginButton = findViewById(R.id.beginButton);

beginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

startActivity(intent);

finish();

}

});

}

}

28) AndroidManifest.xml;

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools">

<uses-permission-sdk-23 android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme"

tools:targetApi="31">

<activity

android:name=".Player.PlayerActivity"

android:exported="false" />

<activity

android:name=".Player.AddSongActivity"

android:exported="false" />

<activity

android:name=".Player.AddTrack"

android:exported="false" />

<service

android:name=".Notification.PushService"

android:exported="false">

<intent-filter>

<action android:name="com.google.firebase.MESSAGING\_EVENT" />

</intent-filter>

</service>

<activity

android:name=".Articles.EditArticleActivity"

android:exported="false" />

<activity

android:name=".Articles.ArticleDetailsActivity"

android:exported="false" />

<activity

android:name=".Articles.AddArticleActivity"

android:exported="false" />

<activity

android:name=".Notes.NoteDetailsActivity"

android:exported="false" />

<activity

android:name=".Notes.EditNoteActivity"

android:exported="false" />

<activity

android:name=".Notes.AddNoteActivity"

android:exported="false" />

<activity

android:name=".WelcomeActivity"

android:exported="false" />

<activity

android:name=".MainActivity"

android:exported="false" />

<activity

android:name=".RegistrationManager"

android:exported="false" />

<activity

android:name=".LoginManager"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

29) activity\_add\_article.xml;

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

android:padding="16dp"

android:background="#F4F4F4"

tools:context=".Articles.AddArticleActivity">

<ImageButton

android:id="@+id/backButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

android:layout\_marginBottom="50dp"/>

<TextView

android:layout\_marginLeft="10dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Создание статьи"

android:textSize="24sp"

android:textColor="#18556D"

android:fontFamily="@font/ubuntu\_bold" />

<TextView

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="25dp"

android:id="@+id/subtitleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Для ввода форматированного текста статьи, используйте html-разметку"

android:textSize="18sp"

android:textColor="#18556D"

android:layout\_marginTop="8dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/titleEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/date\_and\_description\_back"

android:hint="Заголовок"

android:inputType="none"

android:maxLines="1"

android:padding="15dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/descriptionEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Описание"

android:inputType="textMultiLine"

android:maxLines="4"

android:padding="15dp"/>

<LinearLayout

android:id="@+id/rte\_toolbar\_container"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_gravity="center"

android:layout\_height="wrap\_content">

<EditText

android:id="@+id/articleTextEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Текст статьи"

android:maxLines="10"

android:minLines="4"

android:padding="15dp"/>

<HorizontalScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingHorizontal="10dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<ImageButton

android:id="@+id/boldButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/bold\_ic"

android:backgroundTint="#fff"

android:text="Bold"

android:tag="b"/>

<ImageButton

android:id="@+id/italicButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/italic\_ic"

android:backgroundTint="#fff"

android:text="Italic"

android:tag="i"/>

<ImageButton

android:id="@+id/paragraph"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/paragraph\_ic"

android:backgroundTint="#fff"

android:text="paragraph"

android:tag="p"/>

<ImageButton

android:id="@+id/h1Button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/h1\_ic"

android:backgroundTint="#fff"

android:text="h1"

android:tag="h1"/>

<ImageButton

android:id="@+id/h2Button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/h2\_ic"

android:backgroundTint="#fff"

android:text="h2"

android:tag="h2"/>

<ImageButton

android:id="@+id/h3Button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/h3\_ic"

android:backgroundTint="#fff"

android:text="h3"

android:tag="h3"/>

</LinearLayout>

</HorizontalScrollView>

</LinearLayout>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/photoURLEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="URL фото статьи"

android:inputType="textMultiLine"

android:maxLines="3"

android:padding="15dp"/>

<Button

android:layout\_marginHorizontal="10dp"

android:id="@+id/addArticleButton"

android:layout\_width="157dp"

android:layout\_height="40dp"

android:layout\_marginTop="16dp"

android:textAllCaps="false"

android:text="Добавить статью"

android:textColor="#fff"

android:background="@drawable/add\_note\_button\_background"/>

</LinearLayout>

30) activity\_add\_note.xml;

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

android:padding="16dp"

android:background="#F4F4F4"

tools:context=".Notes.AddNoteActivity">

<ImageButton

android:layout\_marginLeft="5dp"

android:id="@+id/backButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

android:layout\_marginBottom="50dp"/>

<TextView

android:layout\_marginLeft="10dp"

android:id="@+id/titleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Опишите свои впечатления"

android:textSize="24sp"

android:textColor="#18556D"

android:fontFamily="@font/ubuntu\_bold"/>

<TextView

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="25dp"

android:id="@+id/subtitleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Как вы себя чувствуете?"

android:textSize="18sp"

android:textColor="#18556D"

android:layout\_marginTop="8dp"/>

<LinearLayout

android:layout\_marginHorizontal="10dp"

android:layout\_marginVertical="8dp"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:orientation="horizontal"

android:layout\_marginTop="8dp"

android:padding="12dp"

android:gravity="center\_vertical"

android:background="@drawable/date\_and\_description\_back">

<EditText

android:id="@+id/dateEditText"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:hint="Дата"

android:focusable="false"

android:clickable="true"

android:inputType="none"

android:background="@android:color/transparent"/>

<ImageView

android:id="@+id/calendarIcon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/calendar\_ic"

android:contentDescription="Календарь"/>

</LinearLayout>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/descriptionEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Ваши впечатления"

android:inputType="textMultiLine"

android:minLines="4"

android:maxLines="10"

android:padding="12dp" />

<Button

android:layout\_marginHorizontal="10dp"

android:id="@+id/addNoteButton"

android:layout\_width="157dp"

android:layout\_height="40dp"

android:layout\_marginTop="16dp"

android:textAllCaps="false"

android:text="Добавить заметку"

android:textColor="#fff"

android:background="@drawable/add\_note\_button\_background"/>

</LinearLayout>

31) activity\_add\_song.xml;

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

android:padding="16dp"

android:background="#F4F4F4"

tools:context=".Articles.AddArticleActivity">

<ImageButton

android:id="@+id/backButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

android:layout\_marginBottom="50dp"/>

<TextView

android:layout\_marginLeft="10dp"

android:id="@+id/titleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Загрузка мелодий"

android:textSize="24sp"

android:textColor="#18556D"

android:fontFamily="@font/ubuntu\_bold" />

<TextView

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="25dp"

android:id="@+id/subtitleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Выберите файл с устройства в формате .mp3/.aac/.wav/.ogg"

android:textSize="18sp"

android:textColor="#18556D"

android:layout\_marginTop="8dp"/>

<Button

android:id="@+id/buttonChooseFile"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Нажмите чтобы выбрать файл"

android:textSize="18dp"

android:textAlignment="textStart"

android:textAllCaps="false"

android:background="@drawable/edit\_text\_back"

android:layout\_marginTop="10dp"

android:layout\_marginHorizontal="10dp"

android:padding="15dp"

android:layout\_marginBottom="10dp"

android:drawableEnd="@drawable/audio\_file\_ic"

android:drawablePadding="8dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/titleEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/date\_and\_description\_back"

android:hint="Название мелодии"

android:inputType="none"

android:maxLines="1"

android:padding="15dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/descriptionEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Описание"

android:inputType="textMultiLine"

android:maxLines="4"

android:padding="15dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/photoUrlEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="URL картинки"

android:inputType="textMultiLine"

android:maxLines="1"

android:padding="15dp"/>

<ProgressBar

android:id="@+id/uploadProgressBar"

android:layout\_marginHorizontal="10dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:visibility="gone" />

<Button

android:layout\_marginHorizontal="10dp"

android:id="@+id/addSongButton"

android:layout\_width="157dp"

android:layout\_height="40dp"

android:layout\_marginTop="16dp"

android:textAllCaps="false"

android:text="Добавить мелодию"

android:textColor="#fff"

android:background="@drawable/add\_note\_button\_background"/>

</LinearLayout>

32) activity\_article\_details.xml;

<?xml version="1.0" encoding="utf-8"?>

<ScrollView 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"

android:scrollbars="none"

android:padding="16dp"

android:layout\_marginBottom="20dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<ImageButton

android:id="@+id/backButton"

android:layout\_marginTop="10dp"

android:layout\_marginHorizontal="15dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

/>

<TextView

android:id="@+id/titleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="70dp"

android:layout\_marginHorizontal="20dp"

android:layout\_marginBottom="16dp"

android:ellipsize="end"

android:fontFamily="@font/ubuntu\_bold"

android:text="Заголовок"

android:textColor="#18556D"

android:textSize="30sp" />

<TextView

android:id="@+id/descriptionTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:ellipsize="end"

android:layout\_marginHorizontal="20dp"

android:text="Описание статьи"

android:textColor="#1E647F"

android:textSize="16sp"

android:layout\_marginBottom="15dp"/>

<TextView

android:id="@+id/articleTextTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:ellipsize="end"

android:layout\_marginHorizontal="20dp"

android:text="Текст статьи"

android:textColor="#1E647F"

android:textSize="16sp"/>

</LinearLayout>

</ScrollView>

33) activity\_edit\_article.xml;

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

android:padding="16dp"

android:background="#F4F4F4"

tools:context=".Articles.AddArticleActivity">

<ImageButton

android:id="@+id/backButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

android:layout\_marginBottom="50dp"/>

<TextView

android:layout\_marginLeft="10dp"

android:id="@+id/titleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Редактирование статьи"

android:textSize="24sp"

android:textColor="#18556D"

android:fontFamily="@font/ubuntu\_bold" />

<TextView

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="25dp"

android:id="@+id/subtitleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Для ввода форматированного текста статьи, используйте html-разметку"

android:textSize="18sp"

android:textColor="#18556D"

android:layout\_marginTop="8dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/titleEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/date\_and\_description\_back"

android:hint="Заголовок"

android:inputType="none"

android:maxLines="1"

android:padding="15dp"/>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/descriptionEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Описание"

android:inputType="textMultiLine"

android:maxLines="4"

android:padding="15dp"/>

<LinearLayout

android:id="@+id/rte\_toolbar\_container"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_gravity="center"

android:layout\_height="wrap\_content">

<EditText

android:id="@+id/articleTextEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="10dp"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Текст статьи"

android:maxLines="10"

android:minLines="4"

android:padding="15dp"/>

<HorizontalScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingHorizontal="10dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<ImageButton

android:id="@+id/boldButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/bold\_ic"

android:backgroundTint="#fff"

android:text="Bold"

android:tag="b"/>

<ImageButton

android:id="@+id/italicButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/italic\_ic"

android:backgroundTint="#fff"

android:text="Italic"

android:tag="i"/>

<ImageButton

android:id="@+id/paragraph"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/paragraph\_ic"

android:backgroundTint="#fff"

android:text="paragraph"

android:tag="p"/>

<ImageButton

android:id="@+id/h1Button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/h1\_ic"

android:backgroundTint="#fff"

android:text="h1"

android:tag="h1"/>

<ImageButton

android:id="@+id/h2Button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/h2\_ic"

android:backgroundTint="#fff"

android:text="h2"

android:tag="h2"/>

<ImageButton

android:id="@+id/h3Button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/h3\_ic"

android:backgroundTint="#fff"

android:text="h3"

android:tag="h3"/>

</LinearLayout>

</HorizontalScrollView>

</LinearLayout>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/photoURLEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="URL фото статьи"

android:inputType="textMultiLine"

android:maxLines="3"

android:padding="15dp"/>

<Button

android:layout\_marginHorizontal="10dp"

android:id="@+id/editArticleButton"

android:layout\_width="157dp"

android:layout\_height="40dp"

android:layout\_marginTop="16dp"

android:textAllCaps="false"

android:text="Изменить статью"

android:textColor="#fff"

android:background="@drawable/add\_note\_button\_background"/>

</LinearLayout>

34) activity\_edit\_note.xml;

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

android:padding="16dp"

android:background="#F4F4F4"

tools:context=".Notes.AddNoteActivity">

<ImageButton

android:layout\_marginLeft="5dp"

android:id="@+id/backButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

android:layout\_marginBottom="50dp"/>

<TextView

android:layout\_marginLeft="10dp"

android:id="@+id/titleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Изменение заметки"

android:textSize="24sp"

android:textColor="#18556D"

android:fontFamily="@font/ubuntu\_bold"/>

<TextView

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="25dp"

android:id="@+id/subtitleTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Измените дату или описание"

android:textSize="18sp"

android:textColor="#18556D"

android:layout\_marginTop="8dp"/>

<LinearLayout

android:layout\_marginHorizontal="10dp"

android:layout\_marginVertical="8dp"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:orientation="horizontal"

android:layout\_marginTop="8dp"

android:padding="12dp"

android:gravity="center\_vertical"

android:background="@drawable/date\_and\_description\_back">

<EditText

android:id="@+id/dateEditText"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:hint="Дата"

android:focusable="false"

android:clickable="true"

android:inputType="none"

android:background="@android:color/transparent"/>

<ImageView

android:id="@+id/calendarIcon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/calendar\_ic"

android:contentDescription="Календарь"/>

</LinearLayout>

<EditText

android:layout\_marginHorizontal="10dp"

android:id="@+id/descriptionEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/date\_and\_description\_back"

android:gravity="top"

android:hint="Ваши впечатления"

android:inputType="textMultiLine"

android:minLines="4"

android:maxLines="10"

android:padding="12dp" />

<Button

android:layout\_marginHorizontal="10dp"

android:id="@+id/editNoteButton"

android:layout\_width="157dp"

android:layout\_height="40dp"

android:layout\_marginTop="16dp"

android:textAllCaps="false"

android:text="Изменить заметку"

android:textColor="#fff"

android:background="@drawable/add\_note\_button\_background"/>

</LinearLayout>

35) activity\_login.xml;

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

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"

android:background="#1E647F"

tools:context=".MainActivity">

<TextView

android:id="@+id/loginText2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:fontFamily="@font/ubuntu\_regular"

android:gravity="center"

android:text="Войдите для продолжения"

android:textColor="#F4F4F4"

android:textSize="20dp"

app:layout\_constraintTop\_toBottomOf="@+id/loginText" />

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/star"

app:layout\_constraintBottom\_toTopOf="@+id/loginText"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.18"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="24dp"

android:layout\_height="31dp"

android:src="@drawable/star"

app:layout\_constraintBottom\_toBottomOf="@+id/loginText2"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.108"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.944" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="30dp"

android:layout\_height="43dp"

android:layout\_marginTop="12dp"

android:src="@drawable/star"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.908"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/loginText" />

<EditText

android:id="@+id/emailEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="55dp"

android:layout\_marginTop="56dp"

android:layout\_marginRight="55dp"

android:background="@drawable/edit\_text\_back"

android:hint="Введите email"

android:padding="12dp"

android:textSize="16dp"

android:inputType="text"

android:imeOptions="actionDone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

android:maxLines="1"

app:layout\_constraintTop\_toBottomOf="@+id/loginText2" />

<TextView

android:id="@+id/loginText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="264dp"

android:fontFamily="@font/ubuntu\_bold"

android:gravity="center"

android:text="Здравствуйте!"

android:textColor="#F4F4F4"

android:textSize="25dp"

android:textStyle="bold"

app:layout\_constraintTop\_toTopOf="parent"

tools:layout\_editor\_absoluteX="0dp" />

<EditText

android:id="@+id/passwordEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="55dp"

android:layout\_marginTop="12dp"

android:layout\_marginRight="55dp"

android:background="@drawable/edit\_text\_back"

android:hint="Введите пароль"

android:inputType="textPassword"

android:imeOptions="actionDone"

android:padding="12dp"

android:textSize="16dp"

android:maxLines="1"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/emailEditText" />

<ProgressBar

android:id="@+id/progressBar"

android:layout\_width="32dp"

android:layout\_height="28dp"

android:visibility="gone"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.753" />

<Button

android:id="@+id/loginButton"

android:layout\_width="160dp"

android:layout\_height="40dp"

android:layout\_marginTop="20dp"

android:background="@drawable/auth\_button\_back"

android:elevation="0dp"

android:text="Войти"

android:textAllCaps="false"

android:textColor="#f4f4f4"

android:fontFamily="@font/ubuntu\_bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.497"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/passwordEditText" />

<LinearLayout

android:id="@+id/registrationLayout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

android:layout\_marginBottom="16dp">

<TextView

android:id="@+id/clickToReg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_regular"

android:text="Еще нет аккаунта?"

android:textColor="#2D2926"

android:textSize="16dp" />

<TextView

android:id="@+id/clickToReg2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="4dp"

android:fontFamily="@font/ubuntu\_bold"

android:text="Регистрация"

android:textColor="#F4F4F4"

android:textSize="16dp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

36) activity\_main.xml;

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout 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=".MainActivity">

<FrameLayout

android:id="@+id/fragment\_container"

android:layout\_width="0dp"

android:layout\_height="0dp"

app:layout\_constraintBottom\_toTopOf="@id/bottomNavbar"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottomNavbar"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:menu="@menu/bottom\_navigation\_bar"

app:labelVisibilityMode="unlabeled"

app:itemIconTint="@drawable/bottom\_nav\_icon\_color"

app:itemIconSize="25dp"

app:itemBackground="@color/white"/>

</androidx.constraintlayout.widget.ConstraintLayout>

37) activity\_note\_details.xml;

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

android:padding="16dp">

<ImageButton

android:id="@+id/backButton"

android:layout\_marginTop="10dp"

android:layout\_marginHorizontal="15dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/back\_arrow"

android:background="@android:color/transparent"

/>

<TextView

android:id="@+id/textViewDate"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="70dp"

android:layout\_marginHorizontal="20dp"

android:layout\_marginBottom="16dp"

android:ellipsize="end"

android:fontFamily="@font/ubuntu\_bold"

android:text="Дата"

android:textColor="#18556D"

android:textSize="30sp" />

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:scrollbars="none">

<TextView

android:id="@+id/textViewDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:ellipsize="end"

android:layout\_marginHorizontal="20dp"

android:text="Описание заметки"

android:textColor="#949494"

android:textSize="16sp" />

</ScrollView>

</LinearLayout>

38) activity\_player.xml;

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout 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:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".Player.PlayerActivity">

<FrameLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageView

android:id="@+id/songImageView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:src="@drawable/feduk\_2022"

android:scaleType="centerCrop"

/>

<View

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#B0000000" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:id="@+id/songTitle"

android:layout\_width="300dp"

android:textAlignment="center"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="220dp"

android:fontFamily="@font/ubuntu\_bold"

android:text="Song title"

android:textColor="#fff"

android:textSize="35dp"

android:layout\_marginBottom="5dp"/>

<TextView

android:id="@+id/songDescription"

android:layout\_width="300dp"

android:textAlignment="center"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="Song description"

android:fontFamily="@font/ubuntu\_light"

android:textSize="16dp"

android:layout\_marginBottom="40dp"/>

<include layout="@layout/circle\_progress\_with\_text" />

<ImageButton

android:id="@+id/playPauseButton"

android:layout\_width="70dp"

android:layout\_height="70dp"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="30dp"

android:background="@android:color/transparent"

android:src="@drawable/play\_button\_for\_player\_ic" />

<TextView

android:id="@+id/finishNowLink"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|bottom"

android:layout\_marginTop="20dp"

android:text="Завершить сейчас"/>

</LinearLayout>

</FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

39) activity\_registration.xml;

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout 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=".MainActivity"

android:background="#1E647F">

<TextView

android:id="@+id/loginText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="256dp"

android:fontFamily="@font/ubuntu\_bold"

android:gravity="center"

android:text="Добро пожаловать!"

android:textColor="#F4F4F4"

android:textSize="25dp"

android:textStyle="bold"

app:layout\_constraintTop\_toTopOf="parent"

tools:layout\_editor\_absoluteX="0dp" />

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/star"

app:layout\_constraintBottom\_toTopOf="@+id/loginText"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.129"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.966" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="24dp"

android:layout\_height="31dp"

android:src="@drawable/star"

app:layout\_constraintBottom\_toBottomOf="@+id/loginText2"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.08"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.941" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="30dp"

android:layout\_height="43dp"

android:layout\_marginTop="4dp"

android:src="@drawable/star"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.9"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/loginText" />

<EditText

android:id="@+id/loginEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="55dp"

android:layout\_marginTop="52dp"

android:layout\_marginRight="55dp"

android:background="@drawable/edit\_text\_back"

android:hint="Введите логин"

android:padding="12dp"

android:textSize="16dp"

android:maxLines="1"

android:inputType="text"

android:imeOptions="actionDone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.495"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/loginText2" />

<TextView

android:id="@+id/loginText2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:fontFamily="@font/ubuntu\_regular"

android:gravity="center"

android:text="Создайте аккаунт"

android:textColor="#F4F4F4"

android:textSize="20dp"

app:layout\_constraintTop\_toBottomOf="@+id/loginText"

tools:layout\_editor\_absoluteX="0dp" />

<EditText

android:id="@+id/emailEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="55dp"

android:layout\_marginTop="12dp"

android:layout\_marginRight="55dp"

android:background="@drawable/edit\_text\_back"

android:maxLines="1"

android:hint="Введите email"

android:padding="12dp"

android:textSize="16dp"

android:inputType="text"

android:imeOptions="actionDone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/loginEditText" />

<EditText

android:id="@+id/passwordEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="55dp"

android:layout\_marginTop="12dp"

android:layout\_marginRight="55dp"

android:background="@drawable/edit\_text\_back"

android:hint="Введите пароль"

android:maxLines="1"

android:inputType="textPassword"

android:padding="12dp"

android:textSize="16dp"

android:imeOptions="actionDone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/emailEditText" />

<ProgressBar

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/authButton"

android:visibility="gone"

android:id="@+id/progressBar"

tools:ignore="MissingConstraints" />

<Button

android:id="@+id/authButton"

android:layout\_width="160dp"

android:layout\_height="40dp"

android:layout\_marginTop="20dp"

android:background="@drawable/auth\_button\_back"

android:elevation="0dp"

android:text="Регистрация"

android:textAllCaps="false"

android:textColor="#f4f4f4"

android:fontFamily="@font/ubuntu\_bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.497"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/passwordEditText" />

<LinearLayout

android:id="@+id/registrationLayout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

android:layout\_marginBottom="16dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_regular"

android:text="Уже есть аккаунт?"

android:textColor="#2D2926"

android:textSize="16dp" />

<TextView

android:id="@+id/clickToLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="4dp"

android:fontFamily="@font/ubuntu\_bold"

android:text="Войти"

android:textColor="#F4F4F4"

android:textSize="16dp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

40) activity\_welcome.xml;

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout 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=".WelcomeActivity"

android:background="#00744B">

<ImageView

android:id="@+id/initialForm"

android:layout\_width="300dp"

android:layout\_height="100dp"

android:layout\_marginTop="272dp"

android:src="@drawable/welcome\_activity\_ic"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/description"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:width="280dp"

android:fontFamily="@font/ubuntu\_light"

android:text="Откройте внутренний покой с нашим приложением для медитаций – ваш путь к гармонии каждый день."

android:textAlignment="center"

android:textColor="#F4F4F4"

android:textSize="16dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.496"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/descriptionHeader" />

<TextView

android:id="@+id/descriptionHeader"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="76dp"

android:text="Добро пожаловать в MAU"

android:textColor="#F4F4F4"

android:textSize="20dp"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/initialForm" />

<ImageView

android:layout\_width="26dp"

android:layout\_height="23dp"

android:layout\_marginBottom="16dp"

android:background="#00744B"

android:src="@drawable/green\_star"

app:layout\_constraintBottom\_toTopOf="@+id/descriptionHeader"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.194"

app:layout\_constraintStart\_toStartOf="parent" />

<ImageView

android:layout\_width="34dp"

android:layout\_height="35dp"

android:background="#00744B"

android:src="@drawable/green\_star"

app:layout\_constraintBottom\_toTopOf="@+id/description"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.084"

app:layout\_constraintStart\_toStartOf="parent" />

<ImageView

android:layout\_width="35dp"

android:layout\_height="34dp"

android:layout\_marginBottom="4dp"

android:background="#00744B"

android:src="@drawable/green\_star"

app:layout\_constraintBottom\_toTopOf="@+id/descriptionHeader"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.825"

app:layout\_constraintStart\_toStartOf="parent" />

<ImageView

android:layout\_width="26dp"

android:layout\_height="23dp"

android:layout\_marginEnd="20dp"

android:background="#00744B"

android:src="@drawable/green\_star"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.421"

app:layout\_constraintStart\_toEndOf="@+id/description"

tools:layout\_editor\_absoluteY="417dp" />

<ImageView

android:layout\_width="52dp"

android:layout\_height="53dp"

android:layout\_marginTop="36dp"

android:layout\_marginEnd="120dp"

android:background="#00744B"

android:elevation="-1dp"

android:src="@drawable/green\_star"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="@+id/description" />

<Button

android:id="@+id/beginButton"

android:layout\_width="180dp"

android:layout\_height="41dp"

android:layout\_marginTop="56dp"

android:background="@drawable/begin\_button\_back"

android:fontFamily="@font/ubuntu\_bold"

android:text="Начать"

android:textAllCaps="false"

android:textColor="#016A45"

android:textSize="17dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.497"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/description" />

</androidx.constraintlayout.widget.ConstraintLayout>

41) circle\_progress\_with\_text.xml;

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:orientation="vertical">

<ProgressBar

android:id="@+id/circleProgressBar"

android:layout\_width="200dp"

android:layout\_height="200dp"

android:indeterminate="false"

android:progress="50"

style="@style/Widget.AppCompat.ProgressBar.Horizontal"

android:progressDrawable="@drawable/circle\_progress\_bar"

android:layout\_centerInParent="true"/>

<TextView

android:id="@+id/tvProgressText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:gravity="center"

android:text="0:00"

android:textColor="#FFFFFF"

android:textSize="25dp"

android:fontFamily="@font/ubuntu\_bold" />

</RelativeLayout>

42) fragment\_articles.xml;

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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=".Articles.ArticlesFragment">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="35dp"

android:layout\_marginTop="50dp"

android:gravity="start"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_bold"

android:text="Статьи MAU"

android:textColor="#18556D"

android:textSize="30dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_regular"

android:paddingVertical="10dp"

android:text="Здесь собраны полезные материалы для

улучшения качества ваших медитаций"

android:textColor="#0E3341"

android:textSize="17dp" />

<View

android:layout\_width="270dp"

android:layout\_height="1dp"

android:layout\_gravity="center"

android:layout\_marginTop="15dp"

android:background="#D9D9D9" />

</LinearLayout>

<ProgressBar

android:id="@+id/progressBarArticles"

android:layout\_width="32dp"

android:layout\_height="28dp"

android:layout\_marginBottom="420dp"

android:visibility="visible"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/linearLayout" />

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerViewArticles"

android:layout\_width="380dp"

android:layout\_height="0dp"

android:layout\_gravity="top"

android:layout\_marginTop="20dp"

android:scrollbars="none"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.516"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/linearLayout"

app:layout\_constraintVertical\_bias="0.0" />

<ImageButton

android:id="@+id/addArticleButton"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_gravity="bottom|end"

android:layout\_marginHorizontal="20dp"

android:layout\_marginBottom="15dp"

android:background="@drawable/add\_note\_button\_background"

android:src="@drawable/add\_note\_ic"

android:textAllCaps="false"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</FrameLayout>

43) fragment\_home.xml;

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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=".Player.HomeFragment">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="35dp"

android:layout\_marginTop="40dp"

android:gravity="start"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_bold"

android:text="Добро пожаловать, "

android:textColor="#18556D"

android:textSize="30dp" />

<TextView

android:id="@+id/loginTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_bold"

android:text="Login"

android:textColor="#18556D"

android:textSize="30dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_regular"

android:paddingVertical="10dp"

android:text="Выберите трек для сопровождения"

android:textColor="#0E3341"

android:textSize="17dp" />

<View

android:layout\_width="270dp"

android:layout\_height="1dp"

android:layout\_gravity="center"

android:layout\_marginTop="15dp"

android:background="#D9D9D9" />

</LinearLayout>

<ProgressBar

android:id="@+id/progressBar"

android:layout\_width="32dp"

android:layout\_height="28dp"

android:layout\_marginBottom="420dp"

android:visibility="visible"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

/>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerViewSongs"

android:layout\_width="370dp"

android:layout\_height="0dp"

android:layout\_gravity="top"

android:scrollbars="none"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.512"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/linearLayout2"

app:layout\_constraintVertical\_bias="0.0"

android:layout\_marginTop="20dp"

/>

<ImageButton

android:id="@+id/addSongButton"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_gravity="bottom|end"

android:layout\_marginBottom="16dp"

android:background="@drawable/add\_note\_button\_background"

android:src="@drawable/add\_note\_ic"

android:textAllCaps="false"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="@+id/recyclerViewSongs" />

</androidx.constraintlayout.widget.ConstraintLayout>

</FrameLayout>

44) fragment\_profile.xml;

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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=".Notes.ProfileFragment">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="121dp"

android:src="@drawable/header\_profile\_settings"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="0dp"

tools:layout\_editor\_absoluteY="-3dp" />

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="80dp"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

android:paddingHorizontal="10dp">

<TextView

android:id="@+id/userLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingHorizontal="20dp"

android:text="Login"

android:textColor="#fff"

android:textSize="35dp"

android:layout\_marginBottom="5dp"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintStart\_toStartOf="@+id/imageView"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.371" />

<TextView

android:id="@+id/userEmail"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_light"

android:paddingHorizontal="20dp"

android:text="Email"

android:textColor="#fff"

android:textSize="18dp"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="@+id/userLogin"

app:layout\_constraintTop\_toBottomOf="@+id/userLogin"

app:layout\_constraintVertical\_bias="0.071" />

</LinearLayout>

<androidx.constraintlayout.widget.ConstraintLayout

android:id="@+id/constraintLayout"

android:layout\_width="match\_parent"

android:layout\_height="80dp"

android:layout\_marginTop="8dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView">

<TextView

android:layout\_width="350dp"

android:layout\_height="wrap\_content"

android:text="Ваш личный дневник медитаций. Запишите свои мысли, наблюдения и впечатления."

android:textColor="#18556D"

android:textSize="18sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

<ProgressBar

android:id="@+id/progressBar"

android:layout\_width="32dp"

android:layout\_height="28dp"

android:layout\_marginBottom="420dp"

android:visibility="visible"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout" />

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerViewNotes"

android:layout\_width="370dp"

android:layout\_height="0dp"

android:layout\_gravity="top"

android:scrollbars="none"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout"

app:layout\_constraintVertical\_bias="0.0" />

<ImageButton

android:id="@+id/addNoteButton"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_gravity="bottom|end"

android:background="@drawable/add\_note\_button\_background"

android:src="@drawable/add\_note\_ic"

android:textAllCaps="false"

app:layout\_constraintBottom\_toBottomOf="parent"

android:layout\_marginBottom="15dp"

android:layout\_marginHorizontal="20dp"

app:layout\_constraintEnd\_toEndOf="@+id/imageView" />

</androidx.constraintlayout.widget.ConstraintLayout>

</FrameLayout>

45) fragment\_settings.xml;

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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=".SettingsFragment">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="121dp"

android:src="@drawable/header\_profile\_settings"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="0dp"

tools:layout\_editor\_absoluteY="-3dp" />

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="80dp"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

android:paddingHorizontal="10dp">

<TextView

android:id="@+id/userLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingHorizontal="20dp"

android:text="Login"

android:textColor="#fff"

android:textSize="35dp"

android:layout\_marginBottom="5dp"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintStart\_toStartOf="@+id/imageView"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.371" />

<TextView

android:id="@+id/userEmail"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_light"

android:paddingHorizontal="20dp"

android:text="Email"

android:textColor="#fff"

android:textSize="18dp"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="@+id/userLogin"

app:layout\_constraintTop\_toBottomOf="@+id/userLogin"

app:layout\_constraintVertical\_bias="0.071" />

</LinearLayout>

<LinearLayout

android:id="@+id/noticeSettingsCard"

android:layout\_width="340dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:background="@drawable/settings\_card\_background"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView">

<TextView

android:id="@+id/noticeSettingsTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_bold"

android:paddingHorizontal="26dp"

android:paddingVertical="15dp"

android:text="Уведомления"

android:textColor="#0E3341"

android:textSize="18dp"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:paddingHorizontal="10dp"

android:paddingVertical="16dp">

<ImageView

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_gravity="center"

android:layout\_marginLeft="18dp"

android:src="@drawable/notice\_ic" />

<TextView

android:layout\_width="225dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:fontFamily="@font/ubuntu\_regular"

android:paddingLeft="15dp"

android:text="Подписаться на уведомления"

android:textColor="#0E3341"

android:textSize="16dp" />

<Switch

android:id="@+id/noticeSwitch"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center" />

</LinearLayout>

</LinearLayout>

<LinearLayout

android:id="@+id/profileSettings"

android:layout\_width="340dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="25dp"

android:background="@drawable/settings\_card\_background"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/noticeSettingsCard">

<TextView

android:id="@+id/cardTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_bold"

android:paddingHorizontal="26dp"

android:paddingVertical="15dp"

android:text="Профиль"

android:textColor="#0E3341"

android:textSize="18dp"/>

<LinearLayout

android:id="@+id/resetLoginLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:paddingHorizontal="10dp"

android:paddingVertical="14dp">

<ImageView

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_gravity="center"

android:layout\_marginLeft="18dp"

android:src="@drawable/profile\_ic" />

<TextView

android:id="@+id/firstLine1"

android:layout\_width="244dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:fontFamily="@font/ubuntu\_regular"

android:paddingLeft="15dp"

android:text="Имя пользователя"

android:textColor="#0E3341"

android:textSize="16dp" />

<ImageButton

android:id="@+id/resetLoginButton"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_gravity="center"

android:background="@android:color/transparent"

android:src="@drawable/gray\_arrow\_ic" />

</LinearLayout>

<View

android:id="@+id/resetLoginSeparator"

android:layout\_width="290dp"

android:layout\_height="1dp"

android:layout\_gravity="center"

android:background="#D9D9D9" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:paddingHorizontal="10dp"

android:paddingVertical="14dp">

<ImageView

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_gravity="center"

android:layout\_marginLeft="18dp"

android:src="@drawable/password\_ic" />

<TextView

android:id="@+id/secondLine"

android:layout\_width="245dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:fontFamily="@font/ubuntu\_regular"

android:paddingLeft="15dp"

android:text="Пароль"

android:textColor="#0E3341"

android:textSize="16dp" />

<ImageButton

android:id="@+id/resetPasswordButton"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_gravity="center"

android:background="@android:color/transparent"

android:src="@drawable/gray\_arrow\_ic" />

</LinearLayout>

<View

android:layout\_width="290dp"

android:layout\_height="1dp"

android:layout\_gravity="center"

android:background="#D9D9D9" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:paddingHorizontal="10dp"

android:paddingVertical="14dp">

<ImageView

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_gravity="center"

android:layout\_marginLeft="18dp"

android:src="@drawable/exit\_ic" />

<TextView

android:id="@+id/logoutTextView"

android:layout\_width="245dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:fontFamily="@font/ubuntu\_regular"

android:paddingLeft="15dp"

android:text="Выйти"

android:textColor="#0E3341"

android:textSize="16dp" />

<ImageButton

android:id="@+id/logoutButton"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_gravity="center"

android:background="@android:color/transparent"

android:src="@drawable/gray\_arrow\_ic" />

</LinearLayout>

<View

android:id="@+id/deleteAccountSeparator"

android:layout\_width="290dp"

android:layout\_height="1dp"

android:layout\_gravity="center"

android:background="#D9D9D9" />

<LinearLayout

android:id="@+id/deleteAccountLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:paddingHorizontal="10dp"

android:paddingVertical="14dp">

<ImageView

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_gravity="center"

android:layout\_marginLeft="18dp"

android:src="@drawable/delete\_profile\_ic" />

<TextView

android:layout\_width="245dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:fontFamily="@font/ubuntu\_regular"

android:paddingLeft="15dp"

android:text="Удалить профиль"

android:textColor="#0E3341"

android:textSize="16dp" />

<ImageButton

android:id="@+id/deleteAccountButton"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_gravity="center"

android:background="@android:color/transparent"

android:src="@drawable/gray\_arrow\_ic"/>

</LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

</FrameLayout>

46) list\_item\_article.xml;

<androidx.cardview.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="130dp"

android:layout\_margin="8dp"

android:background="#F4F4F4"

app:cardCornerRadius="10dp"

android:elevation="2dp"

android:layout\_gravity="top|center">

<LinearLayout

android:id="@+id/cardView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal"

android:padding="10dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:paddingVertical="10dp"

android:paddingLeft="5dp"

android:orientation="vertical">

<TextView

android:id="@+id/textViewTitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:ellipsize="end"

android:fontFamily="@font/ubuntu\_bold"

android:maxLines="1"

android:text="Заголовок статьи"

android:textColor="#18556D"

android:textSize="20sp" />

<TextView

android:id="@+id/textViewDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="20dp"

android:ellipsize="end"

android:maxLines="3"

android:text="Начинать советуют с пяти минут, из которых самой медитации нужно посвятить только две."

android:textColor="#949494"

android:textSize="14sp" />

</LinearLayout>

<ImageView

android:id="@+id/imageViewPhoto"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="@drawable/round\_background"

android:scaleType="centerCrop"

android:layout\_gravity="center"

android:src="@drawable/placeholder\_img"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

47) list\_item\_note.xml;

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="8dp"

android:background="#F4F4F4"

app:cardCornerRadius="8dp"

android:elevation="2dp"

android:layout\_gravity="top|center">

<LinearLayout

android:id="@+id/cardView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:padding="16dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_weight="1">

<TextView

android:id="@+id/textViewDate"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="20sp"

android:fontFamily="@font/ubuntu\_bold"

android:textColor="#18556D"

android:layout\_marginBottom="10dp"

android:ellipsize="end"/>

<TextView

android:id="@+id/textViewDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="14sp"

android:textColor="#949494"

android:maxLines="2"

android:layout\_marginBottom="4dp"

android:ellipsize="end"/>

</LinearLayout>

</LinearLayout>

</androidx.cardview.widget.CardView>

48) list\_item\_song.xml;

<androidx.cardview.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="80dp"

android:layout\_margin="5dp"

android:background="#F4F4F4"

app:cardCornerRadius="10dp"

android:elevation="2dp"

android:layout\_gravity="top|center">

<LinearLayout

android:id="@+id/cardView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal"

android:padding="10dp">

<FrameLayout

android:layout\_width="55dp"

android:layout\_height="55dp"

android:layout\_gravity="center"

android:layout\_marginHorizontal="10dp">

<ImageView

android:id="@+id/imageViewPhoto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/round\_background"

android:scaleType="centerCrop"

android:src="@drawable/placeholder\_img" />

<ImageButton

android:id="@+id/playButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:background="@android:color/transparent"

android:src="@drawable/play\_button\_ic"/>

</FrameLayout>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="vertical"

android:paddingVertical="10dp"

android:paddingLeft="5dp">

<TextView

android:id="@+id/textViewTitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:ellipsize="end"

android:fontFamily="@font/ubuntu\_bold"

android:maxLines="1"

android:text="Название трека"

android:textColor="#18556D"

android:textSize="16sp" />

<TextView

android:id="@+id/textViewDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="20dp"

android:ellipsize="end"

android:maxLines="1"

android:text="Описание трека ssjslv sdvas dfbadba aedvadv vaeva aevaw aevawev WEV"

android:textColor="#949494"

android:textSize="12sp" />

</LinearLayout>

</LinearLayout>

</androidx.cardview.widget.CardView>

49) bottom\_navigation\_bar.xml;

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/settings\_nav"

android:title="settings"

android:icon="@drawable/settings\_ic" />

<item

android:id="@+id/profile\_nav"

android:title="profile"

android:icon="@drawable/profile\_ic" />

<item

android:id="@+id/articles\_nav"

android:title="articles"

android:icon="@drawable/articles\_ic" />

<item

android:id="@+id/home\_nav"

android:title="home"

android:icon="@drawable/home\_ic" />

</menu>

50) context\_menu\_delete.xml;

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/action\_delete"

android:title="Удалить"

/>

</menu>

51) context\_menu\_update\_delete.xml;

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/action\_edit"

android:title="Изменить"

/>

<item

android:id="@+id/action\_delete"

android:title="Удалить"

/>

</menu>

52) themes.xml;

<resources xmlns:tools="http://schemas.android.com/tools">

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

<!-- Основные цвета -->

<item name="colorPrimary">#FFFFFF</item>

<item name="colorOnPrimary">#23708E</item>

<item name="colorPrimaryDark">#1E647F</item>

<item name="colorAccent">#2D2926</item>

<item name="android:colorButtonNormal">#18556D</item>

<!-- Цвета для текста -->

<item name="android:textColorPrimary">#0E3341</item>

<item name="android:textColorTertiary">#1E647F</item>

<item name="android:textColor">#BDBDBD</item>

<item name="android:textColorHint">#BDBDBD</item>

<!-- Цвет для карточек -->

<item name="cardBackgroundColor">#F4F4F4</item>

<!-- Шрифт -->

<item name="android:fontFamily">@font/ubuntu\_regular</item>

</style>

<style name="CustomAlertDialog" parent="Theme.AppCompat.Light.Dialog.Alert">

<item name="colorAccent">#1E647F</item>

<item name="android:textColorPrimary">#0E3341</item>

<item name="android:background">@drawable/round\_background</item>

<item name="android:textColor">#0E3341</item>

<item name="android:paddingLeft">5dp</item>

<item name="android:paddingRight">10dp</item>

<item name="android:paddingTop">10dp</item>

<item name="android:paddingBottom">5dp</item>

</style>

</resources>

53) add\_note\_button\_background.xml;

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#18556D"/>

<corners

android:bottomRightRadius="50dp"

android:bottomLeftRadius="50dp"

android:topLeftRadius="50dp"

android:topRightRadius="50dp"/>

</shape>

54) auth\_button\_back.xml;

<?xml version="1.0" encoding="utf-8"?>

<!-- res/drawable/rounded\_edittext.xml -->

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#2D2926"/>

<corners

android:bottomRightRadius="50dp"

android:bottomLeftRadius="50dp"

android:topLeftRadius="50dp"

android:topRightRadius="50dp"/>

</shape>

55) begin\_button\_back.xml;

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#F4F4F4"/>

<corners

android:bottomRightRadius="50dp"

android:bottomLeftRadius="50dp"

android:topLeftRadius="50dp"

android:topRightRadius="50dp"/>

</shape>

56) circle\_progress\_bar.xml;

<rotate xmlns:android="http://schemas.android.com/apk/res/android"

android:fromDegrees="0"

android:pivotX="50%"

android:pivotY="50%"

android:toDegrees="360">

<layer-list>

<item android:id="@android:id/background">

<shape

android:innerRadiusRatio="3"

android:shape="ring"

android:thicknessRatio="10"

android:useLevel="true">

<corners android:radius="50dp"/>

<gradient

android:startColor="#979797"

android:endColor="#979797"

android:type="sweep" />

</shape>

</item>

<item android:id="@android:id/progress">

<shape

android:innerRadiusRatio="3"

android:shape="ring"

android:thicknessRatio="10"

android:useLevel="true">

<corners android:radius="50dp"/>

<gradient

android:startColor="#F4F4F4"

android:endColor="#FFF"

android:type="sweep" />

</shape>

</item>

</layer-list>

</rotate>

57) date\_and\_description\_back.xml;

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#FFFFFF" />

<corners android:radius="10dp" />

</shape>

58) edit\_text\_back.xml;

<?xml version="1.0" encoding="utf-8"?>

<!-- res/drawable/rounded\_edittext.xml -->

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#FFFFFF"/>

<corners

android:bottomRightRadius="10dp"

android:bottomLeftRadius="10dp"

android:topLeftRadius="10dp"

android:topRightRadius="10dp"/>

</shape>

59) header\_profile\_settings.xml;

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#1E647F"/>

</shape>

60) round\_background.xml;

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<corners android:radius="30dp" />

<solid android:color="#FFFFFF" />

</shape>

61) settings\_card\_background.xml;

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#f4f4f4"/>

<corners android:radius="6dp"/>

</shape>

62) build.gradle.kts;

import org.gradle.internal.impldep.bsh.commands.dir

import org.gradle.internal.impldep.org.eclipse.jgit.lib.ObjectChecker.tree

plugins {

id("com.android.application")

id("com.google.gms.google-services")

}

android {

namespace = "com.example.MAU"

compileSdk = 34

defaultConfig {

applicationId = "com.example.MAU"

minSdk = 24

targetSdk = 34

versionCode = 1

versionName = "1.0"

testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

isMinifyEnabled = false

proguardFiles(

getDefaultProguardFile("proguard-android-optimize.txt"),

"proguard-rules.pro"

)

}

}

compileOptions {

sourceCompatibility = JavaVersion.VERSION\_1\_8

targetCompatibility = JavaVersion.VERSION\_1\_8

}

}

dependencies {

implementation("androidx.appcompat:appcompat:1.6.1")

implementation("com.google.android.material:material:1.11.0")

implementation("androidx.constraintlayout:constraintlayout:2.1.4")

implementation(platform("com.google.firebase:firebase-bom:29.0.3"))

implementation("com.google.firebase:firebase-auth")

implementation("com.google.firebase:firebase-firestore")

implementation("com.google.firebase:firebase-messaging")

implementation("com.google.firebase:firebase-analytics")

implementation("com.google.firebase:firebase-database")

implementation("com.google.firebase:firebase-storage")

implementation("androidx.activity:activity:1.9.0")

testImplementation("junit:junit:4.13.2")

androidTestImplementation("androidx.test.ext:junit:1.1.5")

androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

implementation ("com.squareup.picasso:picasso:2.71828")

}

63) build.gradle.kts;

buildscript {

dependencies {

classpath("com.google.gms:google-services:4.4.0")

}

}

// Top-level build file where you can add configuration options common to all sub-projects/modules.

plugins {

id("com.android.application") version "8.2.1" apply false

id("com.google.gms.google-services") version "4.4.1" apply false

}