

## “Unipi Audio Stories”

Η εφαρμογή Unipi Audio Stories είναι μία Android εφαρμογή που επιτρέπει στους χρήστες να ακούσουν παιδικές ιστορίες και να παρακολουθούν τα στατιστικά της ακρόασης αυτών των ιστοριών. Η εφαρμογή συνδέεται με την Firebase Realtime Database για την αποθήκευση και τη διαχείριση των δεδομένων.

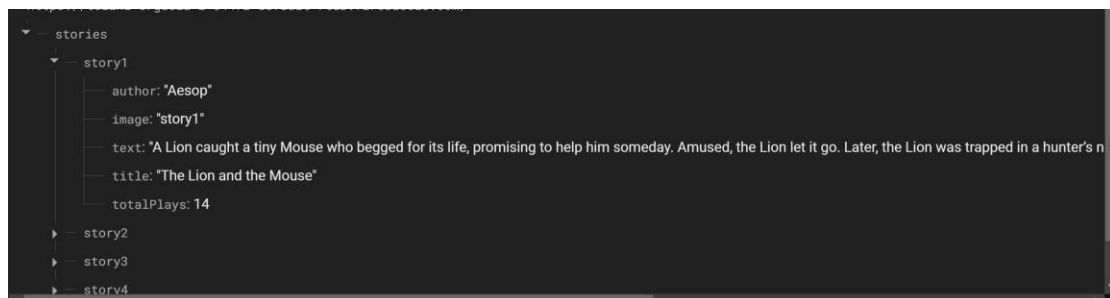
### Σύνδεση με τη Βάση Δεδομένων Firebase

Το πρώτο βήμα που κάναμε αφού δημιουργήσαμε το project της εφαρμογής είναι να το συνδέσουμε με το Firebase Realtime Database για την αποθήκευση των ιστοριών. Το συνδέσαμε μέσω του Android Studio.

Ο σύνδεσμος για πρόσβαση στην βάση που δημιουργήσαμε είναι ο εξής:

<https://console.firebase.google.com/project/teliki-ergasia-2-64472/database/teliki-ergasia-2-64472-default-rtdb/data/~2F>

Η δομή της βάσης είναι η ακόλουθη:



Κάθε ιστορία είναι αποθηκευμένη ως αντικείμενο με τα εξής πεδία:

- title: Ο τίτλος της ιστορίας.
- author: Ο συγγραφέας της ιστορίας.
- text: Το κείμενο της ιστορίας.
- image: Το όνομα της εικόνας της ιστορίας.
- totalPlays: Ο αριθμός των αναπαραγωγών της ιστορίας.

Τα δεδομένα αυτά ανακτώνται και προβάλλονται στην εφαρμογή μέσω του RecyclerView, το οποίο επιτρέπει την εμφάνιση λιστών και αντικειμένων. Κάθε ιστορία αναπαριστάται από το αντικείμενο Story, το οποίο περιλαμβάνει όλα τα παραπάνω πεδία. Δημιουργήσαμε και την κλάση StoryAdapter, η οποία συνδέει τα δεδομένα της λίστας (storyList) από τη βάση με το RecyclerView, ώστε κάθε ιστορία να εμφανίζεται σωστά στη λίστα.

## Λειτουργίες της Εφαρμογής με κώδικα και screenshots από την εκτέλεση

### 1. MainActivity:

Η MainActivity είναι η αρχική οθόνη της εφαρμογής, όπου ο χρήστης καλωσορίζεται και μπορεί να επιλέξει τη γλώσσα της εφαρμογής. Υπάρχουν τρεις διαθέσιμες γλώσσες (Αγγλικά, Ελληνικά, Γερμανικά), και ο χρήστης μπορεί να επιλέξει οποιαδήποτε από αυτές.

Η κύρια λειτουργία της MainActivity είναι να παρέχει δύο βασικές επιλογές:

1. Μετάβαση στην οθόνη με τις ιστορίες (StoriesActivity) μέσω του κουμπιού Start.

Αυτή η μετάβαση υλοποιείται με μέθοδο ορισμένη από το onclick του button στο layout.

```
// Μέθοδος μετάβασης στις ιστορίες
2 usages
public void start(View view){
    Intent intent = new Intent( packageContext: this, StoriesActivity.class);
    startActivity(intent);
}
```

2. Μετάβαση στην οθόνη με τα στατιστικά (StatisticsActivity) μέσω του κουμπιού View Statistics.

Αυτή η μετάβαση γίνεται μέσω onClickListener.

```
//κουμπί για μετάβαση στη σελίδα στατιστικών
Button statisticsButton = findViewById(R.id.button_statistics);
statisticsButton.setOnClickListener(v -> {
    Intent intent = new Intent( packageContext: MainActivity.this, StatisticsActivity.class);
    startActivity(intent);
});
```

Η παρακάτω μέθοδος επιτρέπει την αλλαγή της γλώσσας της εφαρμογής δυναμικά. Χρησιμοποιούμε την κλάση Locale για να ορίσουμε τη γλώσσα και να ενημερώσουμε τις ρυθμίσεις της εφαρμογής με την νέα γλώσσα. Καλούμε αυτήν τη μέθοδο όταν ο χρήστης επιλέξει μια γλώσσα από τις διαθέσιμες επιλογές.

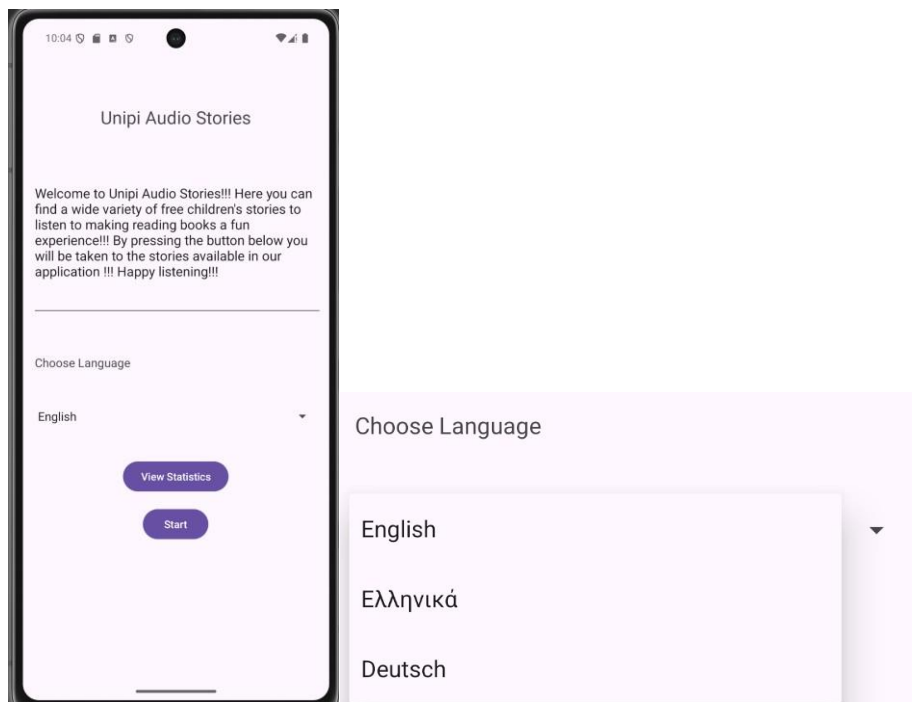
```
// Μέθοδος αλλαγής γλώσσας
2 usages
private void setLocale(String langCode, boolean refresh) {
    Locale locale = new Locale(langCode);
    Locale.setDefault(locale);

    Resources resources = getResources();
    Configuration config = resources.getConfiguration();
    config.setLocale(locale);
    resources.updateConfiguration(config, resources.getDisplayMetrics());

    // Αποθήκευση της επιλεγμένης γλώσσας
    getSharedPreferences("AppSettings", MODE_PRIVATE).edit().putString("Language", langCode).apply();

    // Ανανέωση Activity μόνο αν είναι απαραίτητο
    if (refresh) {
        Intent refreshIntent = new Intent(packageContext, this, MainActivity.class);
        startActivity(refreshIntent);
        finish();
    }
}
```

## Screenshot από την εκτέλεση της εφαρμογής:



## 2. StoriesActivity:

Η **StoriesActivity** είναι υπεύθυνη για την εμφάνιση των ιστοριών που ανακτώνται από τη βάση δεδομένων Firebase. Κάθε ιστορία εμφανίζεται με την εικόνα, τον τίτλο της και τον συγγραφέα της σε μία λίστα (RecyclerView).

Παρακάτω φαίνεται ότι το RecyclerView χρησιμοποιείται για την εμφάνιση της λίστας ιστοριών με κατακόρυφη διάταξη:

```
recyclerView = findViewById(R.id.recyclerView);
recyclerView.setHasFixedSize(true); // Δηλώνουμε ότι το μέγεθος του RecyclerView είναι σταθερό για καλύτερη απόδοση
recyclerView.setLayoutManager(new LinearLayoutManager(context, this)); // Ορίζουμε την διάταξη για την εμφάνιση των ιστοριών σε κάθετη λίστα
```

Για να συνδεθούν τα δεδομένα της λίστας (storyList) από τη βάση με το RecyclerView, ώστε κάθε ιστορία να εμφανίζεται σωστά στη λίστα χρησιμοποιούμε την κλάση StoryAdapter:

```
// Δημιουργία της λίστας ιστοριών και του adapter για το RecyclerView
storiesList = new ArrayList<>();
adapter = new StoryAdapter( context: this, storiesList); // Adapter για την εμφάνιση των ιστοριών
recyclerView.setAdapter(adapter); // Ρύθμιση του adapter στο RecyclerView
```

Στον StoryAdapter, η StoryViewHolder αναπαριστά κάθε στοιχείο της λίστας (τίτλος, συγγραφέας, εικόνα):

```
// ViewHolder για τη διαχείριση των views κάθε στοιχείου
4 usages
public static class StoryViewHolder extends RecyclerView.ViewHolder {
    3 usages
    TextView storyTitle, storyAuthor;
    4 usages
    ImageView storyImage;

    1 usage
    public StoryViewHolder(@NonNull View itemView) {
        super(itemView);
        storyTitle = itemView.findViewById(R.id.storyTitle);
        storyAuthor = itemView.findViewById(R.id.storyAuthor);
        storyImage = itemView.findViewById(R.id.storyImage);
    }
}
```

Οι χρήστες μπορούν να επιλέξουν μία ιστορία από τη λίστα και να μεταβούν στην **AudioStoryActivity**, όπου θα ακούσουν την ιστορία μέσω της λειτουργίας TextToSpeech (TTS).

Η παρακάτω μέθοδος ενεργοποιεί την επιλογή ιστορίας ώστε να μεταφερθούν τα δεδομένα στο AudioStoryActivity μέσω Intent:

```
@Override
public void onBindViewHolder(@NonNull StoryViewHolder holder, int position) {
    Story story = storiesList.get(position); // Παίρνει το αντικείμενο Story της τρέχουσας θέσης
    holder.storyTitle.setText(story.getTitle()); // Εμφανίζει τον τίτλο της ιστορίας
    holder.storyAuthor.setText("Author: " + story.getAuthor()); // Εμφανίζει τον συγγραφέα της ιστορίας

    // Ρύθμιση click listener για το στοιχείο
    View.OnClickListener clickListener = v -> {
        // Δημιουργία intent για να ανοίξει το AudioStoryActivity
        Intent intent = new Intent(context, AudioStoryActivity.class);
        intent.putExtra( name: "title", story.getTitle());
        intent.putExtra( name: "author", story.getAuthor());
        intent.putExtra( name: "text", story.getText());
        intent.putExtra( name: "image", story.getImage());
        context.startActivity(intent);
    };

    // Εγγραφή του click listener στα στοιχεία του ViewHolder
    holder.storyTitle.setOnClickListener(clickListener);
    holder.storyAuthor.setOnClickListener(clickListener);
    holder.storyImage.setOnClickListener(clickListener);

    // Φόρτωση εικόνας από το drawable με βάση το όνομα που υπάρχει στη Firebase
    int imageId = context.getResources().getIdentifier(story.getImage(), "drawable", context.getPackageName());
    if (imageId != 0) {
        holder.storyImage.setImageResource(imageId); // Φόρτωση της εικόνας
    } else {
        // Εάν η εικόνα δεν βρεθεί, εμφανίζεται μια προεπιλεγμένη εικόνα
        holder.storyImage.setImageResource(R.drawable.audio_stories);
    }
}
```

Η ανάκτηση των δεδομένων γίνεται από τη βάση μέσω του Firebase Realtime Database, και τα δεδομένα προβάλλονται μέσω του RecyclerView, το οποίο επιτρέπει την ομαλή εμφάνιση της λίστας των ιστοριών.

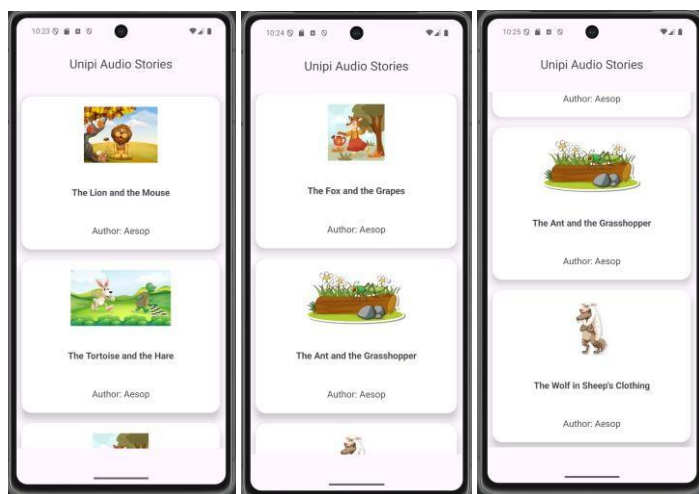
```
// Δημιουργία της λίστας ιστοριών και του adapter για το RecyclerView
storiesList = new ArrayList<>();
adapter = new StoryAdapter( context: this, storiesList); // Adapter για την εμφάνιση των ιστοριών
recyclerView.setAdapter(adapter); // Ρύθμιση του adapter στο RecyclerView

// Σύνδεση με τη βάση δεδομένων Firebase, στην περιοχή "stories"
databaseReference = FirebaseDatabase.getInstance().getReference( path: "stories");

// Φόρτωση δεδομένων από Firebase όταν αλλάξει κάτι
databaseReference.addValueEventListener(new ValueEventListener() {
    2 usages
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        // Καθορισμός της λίστας πριν την προσθήκη νέων δεδομένων
        storiesList.clear();
        // Διατρέχουμε όλα τα δεδομένα που έχουμε από τη βάση
        for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
            Story story = dataSnapshot.getValue(Story.class);
            // Προσθήκη της ιστορίας στη λίστα
            storiesList.add(story);
        }
        // Ενημέρωση του adapter για να ανανεώσει την εμφάνιση του RecyclerView
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        // Χειρισμός σφαλμάτων που μπορεί να προκύψουν κατά τη φόρτωση των δεδομένων
    }
});
```

Screenshot από την εκτέλεση της εφαρμογής:



### 3. AudioStoryActivity:

Η **AudioStoryActivity** είναι η οθόνη όπου ο χρήστης μπορεί να ακούσει την ιστορία που επέλεξε, χρησιμοποιώντας τη λειτουργία **TextToSpeech (TTS)**. Στην οθόνη εμφανίζεται επίσης ο τίτλος της ιστορίας, ο συγγραφέας της και η εικόνα της, η οποία φορτώνεται δυναμικά από τα resources του Android.

Η λήψη των δεδομένων γίνεται μέσω του Intent:

```
// Λήψη δεδομένων από το Intent
Intent intent = getIntent();
String title = intent.getStringExtra( name: "title");
String author = intent.getStringExtra( name: "author");
String text = intent.getStringExtra( name: "text");
String image = intent.getStringExtra( name: "image");
```

Όταν ο χρήστης πατήσει το button με το ηχείο, ενεργοποιείται η λειτουργία Text to Speech από τον παρακάτω κώδικα:

```
// Ρύθμιση του TextToSpeech
textToSpeech = new TextToSpeech( context: this, listener: this);

// Ρύθμιση του click listener για το κουμπί αναπαραγωγής
speakerButton.setOnClickListener(v -> {
    // Λήψη του κειμένου της ιστορίας και αναπαραγωγή του μέσω TTS
    String storyTextToRead = storyText.getText().toString();
    textToSpeech.speak(storyTextToRead, TextToSpeech.QUEUE_FLUSH, params: null, utteranceId: null);
    // Ενημέρωση του totalPlays στο Firebase
    updateTotalPlays(title);
});
```

Κάθε φορά που ο χρήστης ακούει μία ιστορία, η εφαρμογή καταγράφει την ακρόαση αυξάνοντας τον μετρητή totalPlays για τη συγκεκριμένη ιστορία στη βάση δεδομένων Firebase. Αυτό λειτουργεί από τον παρακάτω κώδικα:

```
private void updateTotalPlays(String title) {
    // Έλεγχος αν ο τίτλος της ιστορίας είναι άδειος
    if (title == null || title.isEmpty()) {
        Toast.makeText( context: this, text: "Invalid story title!", Toast.LENGTH_SHORT).show();
        return;
    }

    // Προσθήκη listener για ανδρύων δεδομένων από το Firebase
    databaseReference.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            boolean storyFound = false;

            // Αναζήτηση ιστορίας με τον αντίστοιχο τίτλο
            for (DataSnapshot storySnapshot : snapshot.getChildren()) {
                String storyTitle = storySnapshot.child( path: "title").getValue(String.class);

                if (storyTitle != null && storyTitle.equals(title)) {
                    storyFound = true;
                    // Αναζήτηση του αριθμού των προηγούμενων εκπομπών
                    Integer currentPlays = storySnapshot.child( path: "totalPlays").getValue(Integer.class);

                    if (currentPlays == null) {
                        currentPlays = 0; // Εξασφάλιση μη null τιμής
                    }

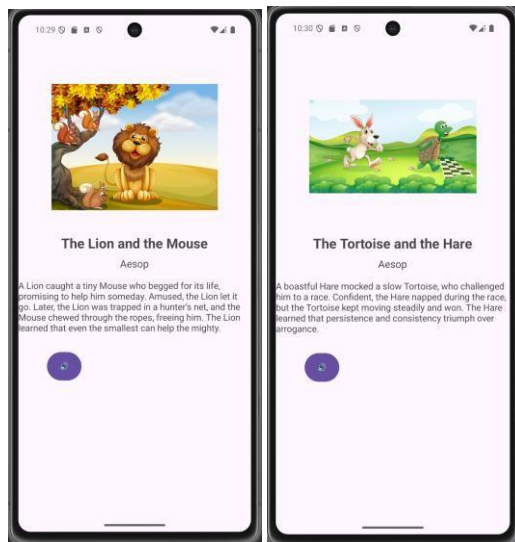
                    // Ενημέρωση του totalPlays με την νέα τιμή
                    int newTotalPlays = currentPlays + 1;
                    storySnapshot.getRef().child( path: "totalPlays").setValue(newTotalPlays);
                }
            }
        }
    });
}
```

```
// Ενημέρωση του totalPlays με την νέα τιμή
int newTotalPlays = currentPlays + 1;
storySnapshot.getRef().child( path: "totalPlays").setValue(newTotalPlays)
    .addOnSuccessListener(avoid -> Log.d( tag: "AudioStoryActivity", msg: "Plays updated successfully to: " + newTotalPlays))
    .addOnFailureListener(avoid -> Toast.makeText( context: AudioStoryActivity.this, text: "Failed to update plays: " + e.getMessage(), Toast.LENGTH_SHORT).show());
break;
}

if (storyFound) {
    // Ενημέρωση αν η ιστορία δεν βρέθηκε
    Log.d( tag: "AudioStoryActivity", msg: "Story not found in database: " + title);
    Toast.makeText( context: AudioStoryActivity.this, text: "Story not found in database!", Toast.LENGTH_SHORT).show();
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    // Αναζήτηση αποτυχημένη από το Firebase
    Toast.makeText( context: AudioStoryActivity.this, text: "Failed to fetch story: " + error.getMessage(), Toast.LENGTH_SHORT).show();
}
});
}
```

Screenshot από την εκτέλεση της εφαρμογής:



#### 4. StatisticsActivity:

Η **StatisticsActivity** επιτρέπει στους χρήστες να δουν τα στατιστικά τους, όπως τον αριθμό των ακροάσεων για κάθε ιστορία και τις τρεις πιο αγαπημένες τους ιστορίες, με βάση τις ακροάσεις.

Η βάση δεδομένων Firebase καταγράφει τις ακροάσεις και αποθηκεύει τα στατιστικά για κάθε ιστορία. Ο χρήστης μπορεί να βλέπει σε πραγματικό χρόνο την ενημέρωση αυτών των στατιστικών.

Ο κώδικας για την εμφάνιση των στατιστικών ακρόασης ανά ιστορία είναι ο εξής:

```
// Κλήση της μεθόδου για να φορτώσουμε τα στατιστικά  
fetchStatistics();
```



```

private void fetchStatistics() {
    // Listener για να παρακολουθεί τις αλλαγές στη βάση δεδομένων
    databaseReference.addValueEventListener(new ValueEventListener() {
        2 usages
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            int count = 0;
            // Διατρέχουμε όλα τα δεδομένα που έχουμε από τη βάση
            for (DataSnapshot storySnapshot : snapshot.getChildren()) {
                if (count >= 5) break;

                // Ανάκτηση του τίτλου και των συνολικών plays για κάθε ιστορία
                String title = storySnapshot.child( path: "title").getValue(String.class);
                int totalPlays = storySnapshot.child( path: "totalPlays").getValue(Integer.class);

                // Ενημέρωση των TextViews με τα δεδομένα της κάθε ιστορίας
                switch (count) {
                    case 0:
                        story1Title.setText(title);
                        story1Plays.setText(String.valueOf(totalPlays));
                        break;
                    case 1:
                        story2Title.setText(title);
                        story2Plays.setText(String.valueOf(totalPlays));
                        break;
                    case 2:
                        story3Title.setText(title);
                        story3Plays.setText(String.valueOf(totalPlays));
                        break;
                    case 3:
                        story4Title.setText(title);
                        story4Plays.setText(String.valueOf(totalPlays));
                        break;
                    case 4:
                        story5Title.setText(title);
                        story5Plays.setText(String.valueOf(totalPlays));
                        break;
                }
                count++;
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                // Εμφάνιση μηνύματος λάθους αν αποτύχει η ανάκτηση των δεδομένων
                Toast.makeText( context: StatisticsActivity.this, text: "Failed to fetch data: " + error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    });
}

```

Ο κώδικας για την εμφάνιση των 3 κορυφαίων ιστοριών με βάση τις ακροάσεις είναι ο εξής:



```

private void loadTopStories() {
    // Κλήση για ανάγνωση όλων των ιστοριών από τη βάση δεδομένων
    databaseReference.addListenerForSingleValueEvent(new ValueEventListener() {
        2 usages
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            List<Story> allStories = new ArrayList<>();

            // Ανάκτηση όλων των ιστοριών από τη βάση δεδομένων
            for (DataSnapshot storySnapshot : snapshot.getChildren()) {
                String title = storySnapshot.child( path: "title").getValue(String.class);
                Integer totalPlays = storySnapshot.child( path: "totalPlays").getValue(Integer.class);

                // Φιλτράρισμα ιστοριών με totalPlays > 0
                if (title != null && totalPlays != null && totalPlays > 0) {
                    allStories.add(new Story(title, totalPlays));
                }
            }

            // Ταξινόηση σε φθίνουσα σειρά με βάση τα totalPlays
            Collections.sort(allStories, (story1, story2) -> story2.getTotalPlays() - story1.getTotalPlays());

            // Εμφάνιση δυναμικά των 3 κορυφαίων ιστοριών ή όσων υπάρχουν
            linearLayoutFavorites.removeAllViews(); // Αφαίρεση προηγούμενων Views
            int count = Math.min(3, allStories.size());

            for (int i = 0; i < count; i++) {
                Story story = allStories.get(i);

                // Δημιουργία TextView για την αριθμηση
                TextView numberTextView = new TextView( context: StatisticsActivity.this);
                numberTextView.setText((i + 1) + ". ");
                numberTextView.setTextSize(18);
                numberTextView.setGravity(Gravity.START);
                numberTextView.setPadding( left: 16, top: 8, right: 16, bottom: 8);

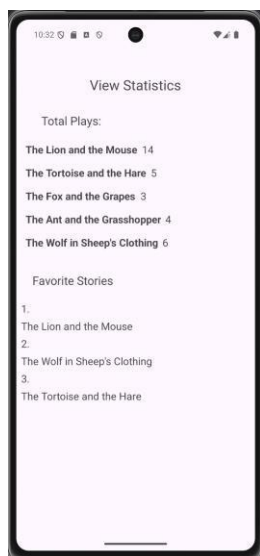
                // Δημιουργία TextView για τον τίτλο
                TextView titleTextView = new TextView( context: StatisticsActivity.this);
                titleTextView.setText(story.getTitle());
                titleTextView.setTextSize(18);
                titleTextView.setGravity(Gravity.START);
                titleTextView.setPadding( left: 16, top: 8, right: 16, bottom: 8);

                // Προσθήκη του TextView με την αριθμηση και του τίτλου στο LinearLayout
                linearLayoutFavorites.addView(numberTextView);
                linearLayoutFavorites.addView(titleTextView);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                // Εμφάνιση μηνύματος λάθους αν αποτύχει η φόρτωση των κορυφαίων ιστοριών
                Toast.makeText( context: StatisticsActivity.this, text: "Failed to load statistics: " + error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    });
}

```

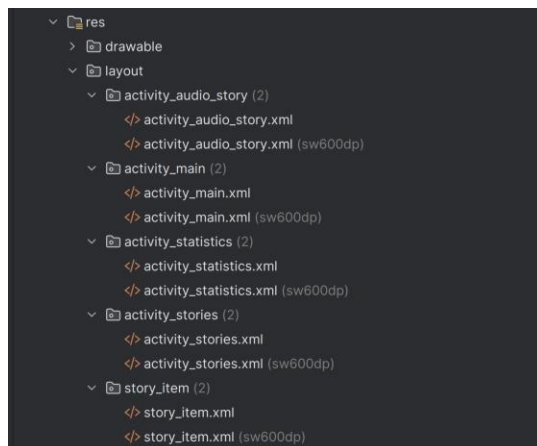
## Screenshot από την εκτέλεση της εφαρμογής:



## Υποστήριξη Responsive Design για Tablets

Η εφαρμογή έχει βελτιστοποιηθεί για μεγαλύτερες συσκευές, όπως tablets, μέσω της χρήσης του φακέλου layout-sw600dp. Ο φάκελος αυτός περιλαμβάνει προσαρμοσμένες εκδόσεις των αρχείων διάταξης, οι οποίες διασφαλίζουν την καλύτερη εμφάνιση του περιεχομένου σε οθόνες με πλάτος τουλάχιστον **600dp**.

Παρακάτω φαίνεται ότι τα layouts είναι προσαρμοσμένα και για smartphone και για tablets:



Screenshots από την εκτέλεση σε tablet:

