****

Tic Tac Toe

**בית ספר: קריית החינוך למדעים רחובות**

**Tic Tac Toe שם הפרויקט:**

**שם התלמיד: טל רמון**

**מספר תעודת זהות: 325181519**

**שם המנחה: זהבה לביא**

**תאריך הגשה:**

תוכן עניינים

[מבוא 3](#_Toc74001615)

[מבנה וUML 4](#_Toc74001616)

[מדריך למשתמש 6](#_Toc74001617)

[הוראות התקנה 6](#_Toc74001618)

[פירוט מסכים 6](#_Toc74001619)

[בסיס נתונים – פיירבייס 13](#_Toc74001632)

[מדריך למפתח 15](#_Toc74001633)

[Activities 15](#_Toc74001634)

[Classes 38](#_Toc74001635)

[Adapters 45](#_Toc74001639)

[Broadcast receiver 47](#_Toc74001640)

[Menu 48](#_Toc74001641)

[service 49](#_Toc74001642)

[Drawable 51](#_Toc74001643)

[Layout 51](#_Toc74001644)

[Raw 52](#_Toc74001646)

[Manu 52](#_Toc74001647)

[mipmap 52](#_Toc74001648)

[סיכום אישי 53](#_Toc74001649)

מבוא

אפליקציית Tic Tac Toe היא הפתרון המושלם לכל האנשים שאוהבים את המשחק האגדי איקס עיגול וגם אוהבים להיות מקום ראשון בכל תחרות. המשחק מאפשר לכלל המשתתפים לשחק נגד אלגוריתם (או נגד חבר כדי להתאמן) ולשבור שיא ברצף ניצחונות, לנצח במשחק איקס עיגול זו לו המשימה הכי קשה בעולם, אבל לנצח בהכי הרבה משחקים ברצף זה כבר אתגר ברמה אחרת לגמרי. האפליקציה נוצרה לאחר הבנת הצורך בקרב חברים, למשחק קל, פשוט, קצר אך תחרותי מאוד, שכל מטרתו היא להיות יותר טוב מהחברים שלך, כי בכולנו יש את יצר התחרותיות. מחקר של השוק מראה כי קיים טרנד רחב מאוד של אפליקציות תחרותיות שעיקר המשחק בהן הינו פשוט מאוד. מאפשרת לשחקנים שונים ברחבי העולם, לשחק ולהתחרות על המיקום הראשון בטבלת השיאים. לכל משתמש קיים משתמש אישי אשר אליו נכנס בעזרת שם משתמש וסיסמא. כל משתמש יכול להכנס למשתמש של עצמו מכל מכל בעולם ומכל מכשיר אנדרואיד.

האפליקציה מבוססת על מסד נתונים SQLDataBase וגם על מסד נתונים למשתמשים של FireBase המאפשר לכל משתמש להתחבר לאפליקציה מכל מקום בעולם ולגשת למשתמש שלו. במידה והמשתמש איבד את הסיסמא שהגדיר בתהליך ההרשמה, הוא יכול לשחזר אותה בעזרת מסך ייעודי לאיפוס סיסמא.

האפליקציה שמה דגש גדול על חווית משתמש מרבית, בעזרת עיצוב מודרני, השלמה אוטומטית, אנימציות, מוזיקת רקע והתראה ייעודית על כך שנגמר התור שלך לשחק (בעזרת BroadcasrReceiver).

האפליקציה מומלצת מאוד לכל אחד ואחת בכל גיל.

את האפליקציה בחרתי ליצור בעקבות כך שאני מאוד תחרותי, יצא לי בעצמי לשחק המון איקס עיגול בעבר וחשתי תחושת נוסטלגיה גדולה בעת יצירת האפליקציה.

האתגר המרכזי בכתיבת האפליקציה היתה בניהול הנתונים עבור כל משתמש שיוכל להתחבר למשתמש שלהם, ושיוכל למחוק באופן מיידי משתמש ששיחק בפלאפון שלהם מרשימת השיאים (במידה וחבר כבר לא משתתף בתחרות). החיבור לפיירבייס היה ללא ספק האתגר המרכזי בפרויקט.

בנוסף, בניית האלגוריתם למשחק המחשב היה מאתגר.

מבנה

שם משתמש וסיסמא

שם משתמש וסיסמא

טעינה

בחירה מהתפריט

SQL-DB

FireBase

FireBase

FireBase

משחק נגד חבר

משחק נגד מחשב

רשימת שיאים

מסך סיום משחק

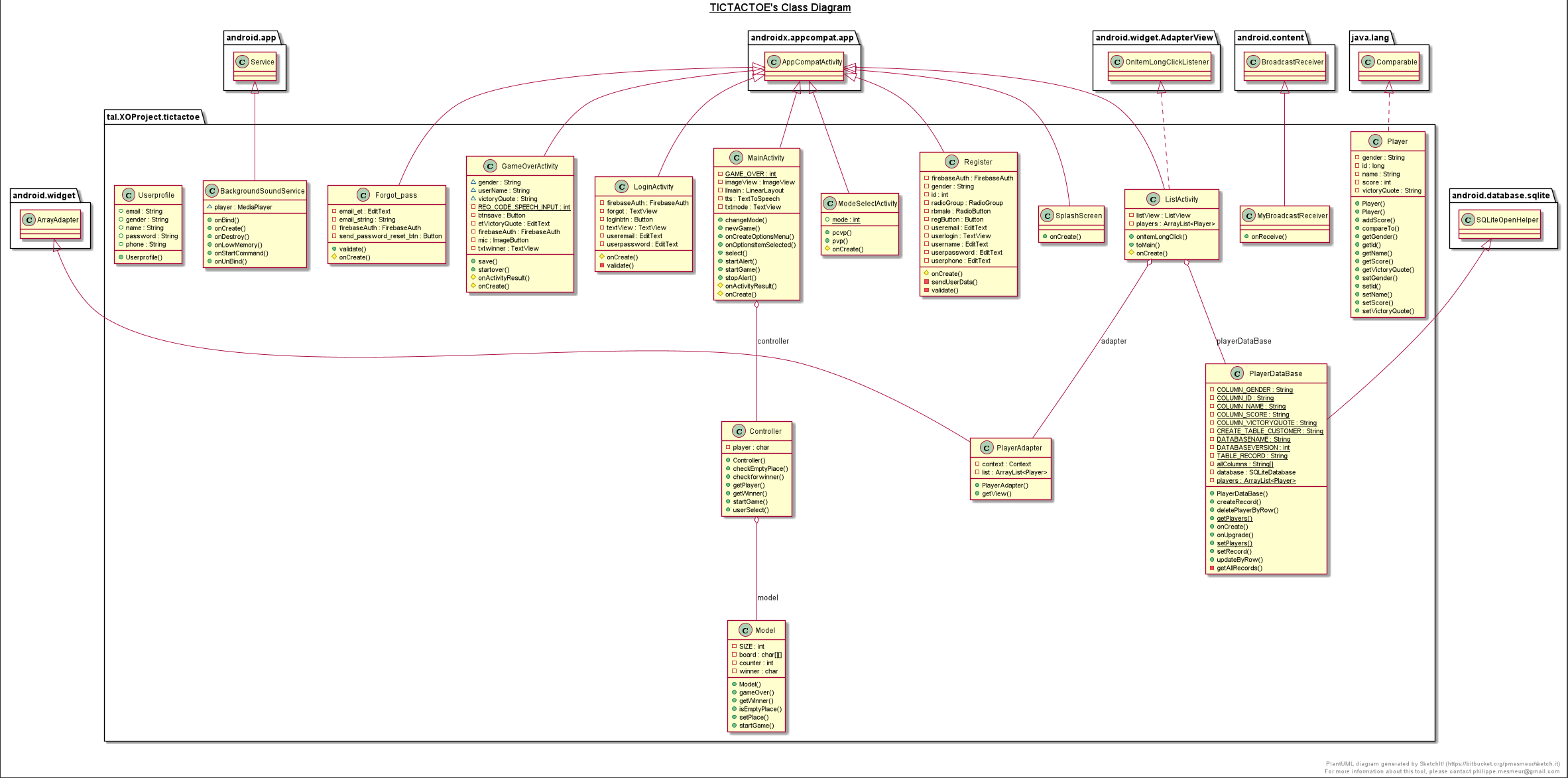
משחק

מסך בחירה

התחברות

שכחתי סיסמה

הרשמה



UML

מדריך למשתמש

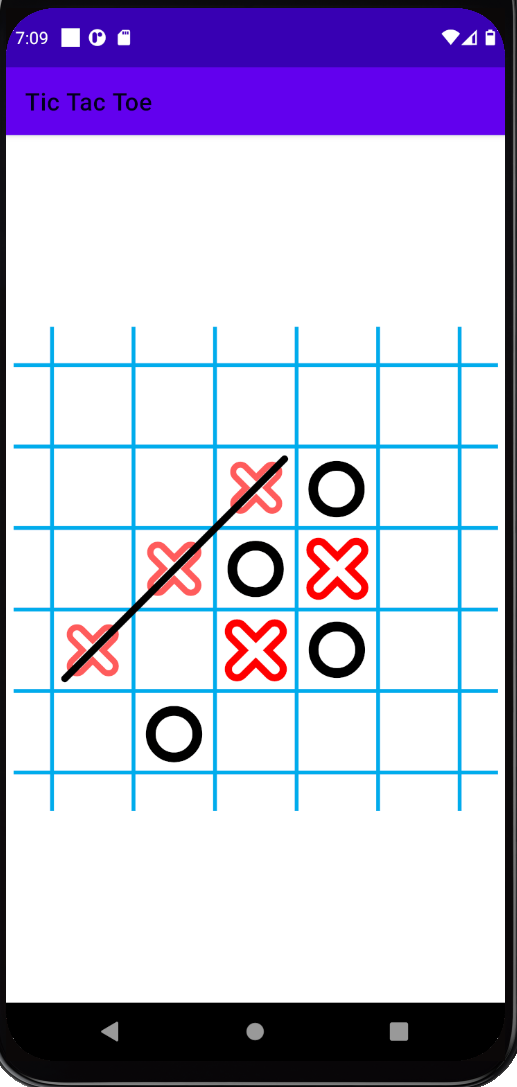
הוראות התקנה –

האפליקציה מותקנת על פלאפוני אנדרואיד גרסה 16 ומעלה, כאשר ההתקנה מתבצעת על ידי קובץ   
APK   
הנשלח ישירות למכשיר על ידי המפתח.

פירוט מסכים**-**

מסך טעינה- מסך אנימציה מספריית   
LOTTIE

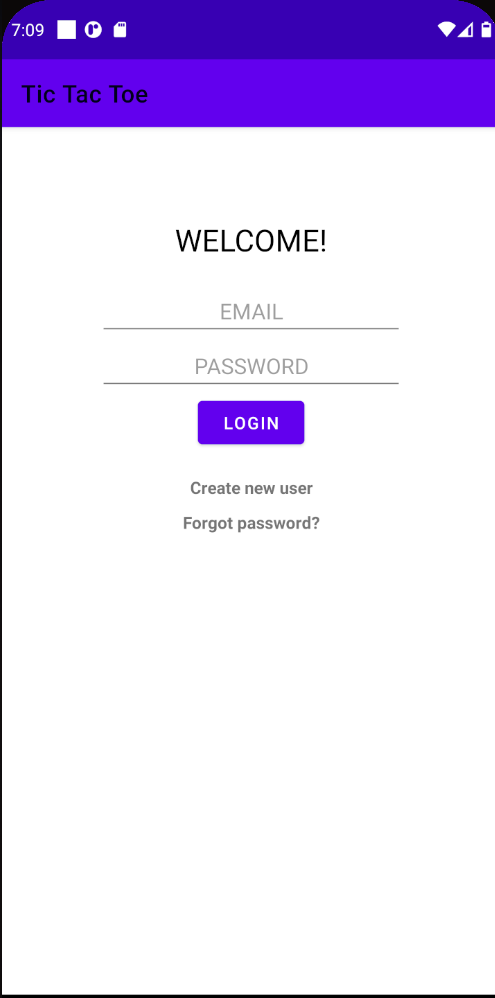
המאפשרת למשתמש להכנס בצורה חלקה יותר לאפליקציה, המאפשרת למשתמש מעט זמן לפני שהאפליקציה נפתחת.



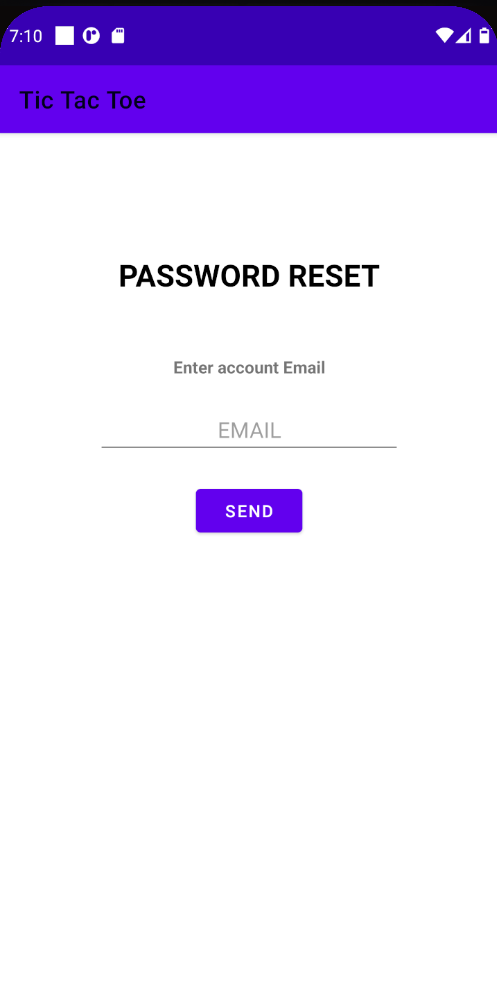
מסך התחברות- במסך זה המשתמש מתבקש להזין שם משתמש וסיסמא, במידה והשם לא קיים או הסיסמא לא נכונה תוצג למשתמש הודעה מתאימה.

במידה והמשתמש שכח את הסיסמא הוא יכול ללחוץ על איפוס סיסמא.

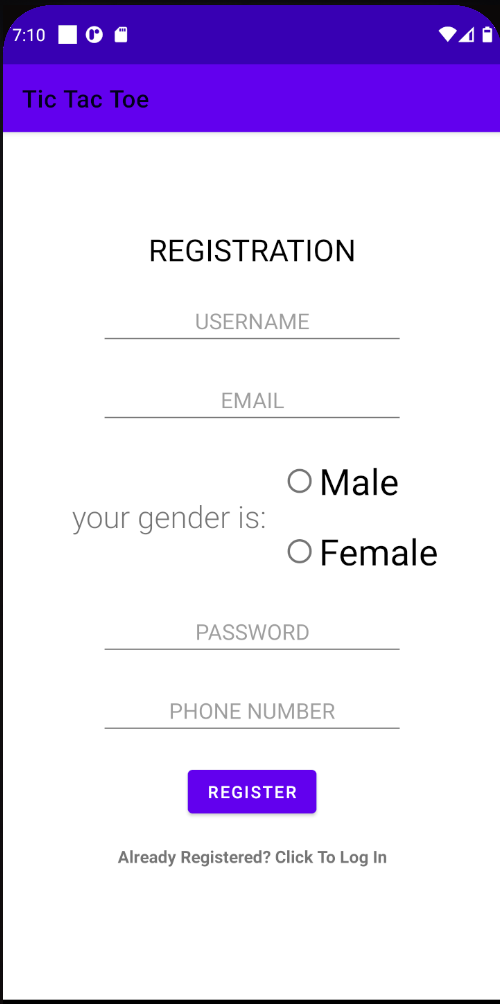
במידה והמשתמש חדש באפליקציה הוא יכול ללחות על כפתור ההרשמה.



מסך איפוס סיסמא -במידה והמשתמש שכח את סיסמתו, הוא יכול במסך זה להזין את שם המשתמש שלו והסיסמא לאיפוס תישלח אליו לאימייל. מסך זה משתמש בשירותים של פיירבייס. במידה והמשתמש מזין אימייל שלא קיים במערכת, תוצג למשתמש הודעה מתאימה.



מסך הרשמה- במסך זה המשתמש נרשם לאפליקציה, הוא מזין שם משתמש, סיסמא ופרטים נוספים והמערכת תיצור למשתמש, משתמש במערכת הפיירבייס ותכניס אותו לאפליקציה. במידה והמשתמש מנסה להזין אימייל קיים, או אימייל לא חוקי, או סיסמא לא חוקית תוצג למשתמש הודעה מתאימה. בדיקת תקינות הקלט מתבצעת על ידי פיירבייס. במידה והמשתמש לא מזין כלום תוצג לו הודעה מתאימה.



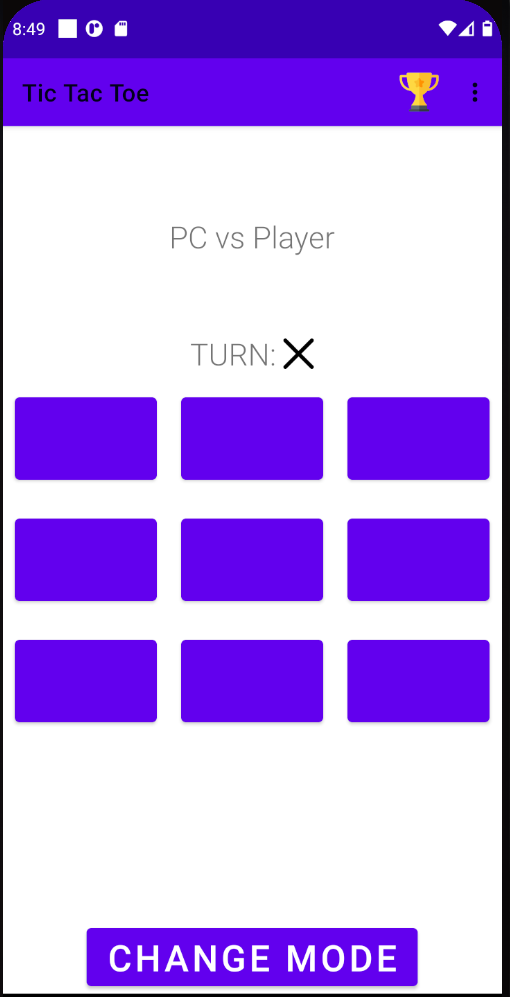
מסך הבחירה- במסך זה המשתמש יכול לבחור מה הוא רוצה לעשות, האם לשחק נגד שחקן (אדם) אחר כדי להתאמן או האם לשחק נגד המחשב כדי לשבור שיא ברצף נצחונות.

תמונה שמכילה טקסט

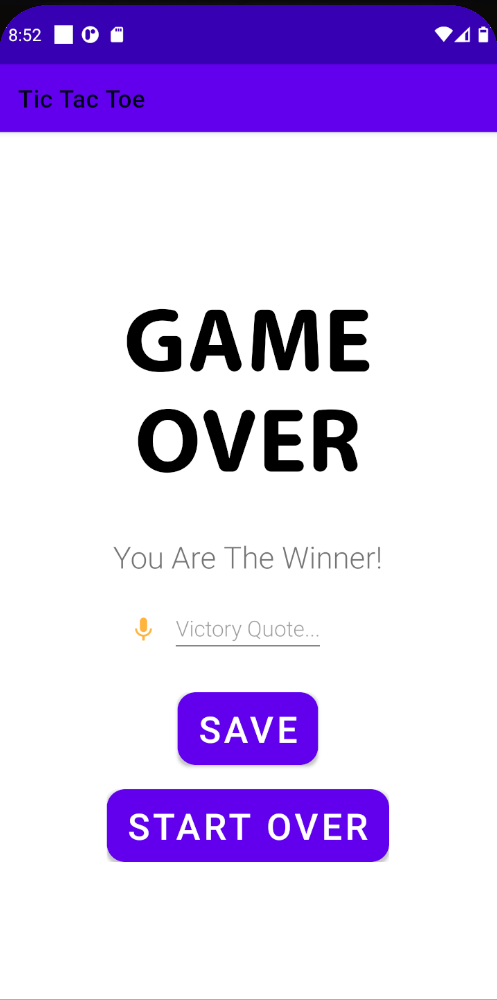
התיאור נוצר באופן אוטומטי

מסך המשחק- במסך זה המשתמש משחק במשחק איקס עיגול, נגד אדם כדי להתאמן או נגד המשחק במשחק צובר שיא, יש לו אפשרות ללכת ישירות לטבלת השיאים בלחיצה על הגביע למעלה או לשתף את האפליקציה דרך התפריט. במידה ומעוניין להחליף מוד (כדי לשחק נגד אדם או נגד מחשב) הוא יכול ללחוץ על הכפתור למטה. בראש המסך מוצג המוד הנוכחי והתור הנוכחי, במידה והתור הוא של השחקן המשחק האפליקציה מחכה שהמשתמש יבחר איפה הוא רוצה לשים את התור שלו. בנוסף, יש כאן מוזיקת רקע של המשחק בעזרת סרביס וטיימר של 30 שניות בעזרת ברודקאסט רסיבר שמתריע למשתמש כשהתור שלו נגמר.

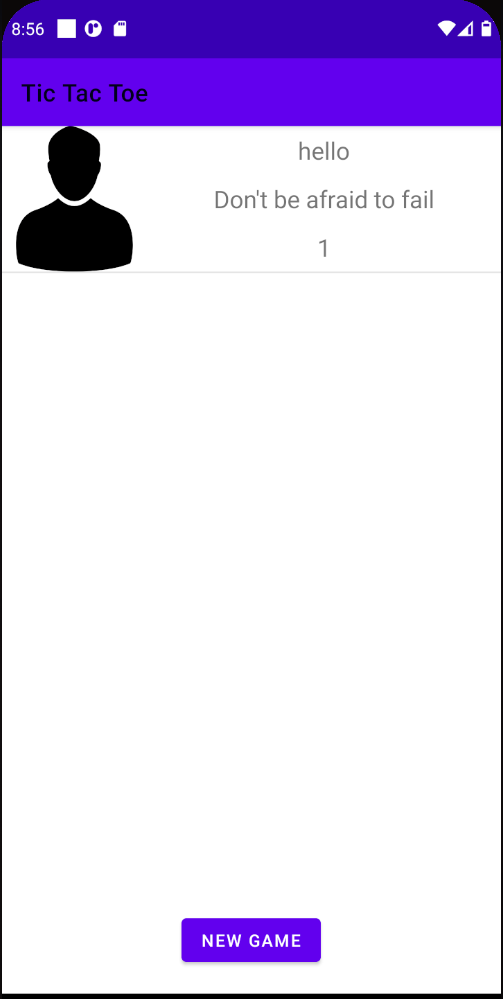
תמונה שמכילה ריבוע

התיאור נוצר באופן אוטומטי

מסך סוף משחק – במסך זה מוצג למשתמש מי ניצח, האם הוא או היריב, האפליקציה אומרת בקול רם באנגלית "אתה המנצח!" במידה ואכן המשתמש ניצח, המשתמש יכול להקליד או להקליט בעזרת כפתור המיקרופון ואלגוריתם דיבור לטקסט, את ציטוט הנצחון שלו עבור המשחק הנוכחי וללחוץ על כפתור השמירה כדי לתעד את הניצחון בדאטאבייס, או להתחיל ישירות במשחק חדש.



מסך השיאים- במסך זה המשתמש יכול לראות את כל השחקנים ואת כל השיאים ששיחקו בפלאפון הזה ישירות מהדאטבייס (אס קיו אל). הרשימה מציגה את שם המשתמש של השחקן, הציטוט של הניצחון האחרון שלו, מספר הנצחונות שלו ואייקון של זכר/נקבה בהתאם למין המשתמש. הרשימה מסודרת מהגדול לקטן, כאשר משתמש מנצח מספר פעמים, האיבר ברשימה מתעדכן, כאשר משתמש חדש משחק ומנצח איבר חדש עם משתמש חדש נוסף לרשימה.



# הרשאות-

*<*uses-permission android:name="android.permission.VIBRATE" */>*

הרשאת גישה לבצע ויברציה בטלפון.

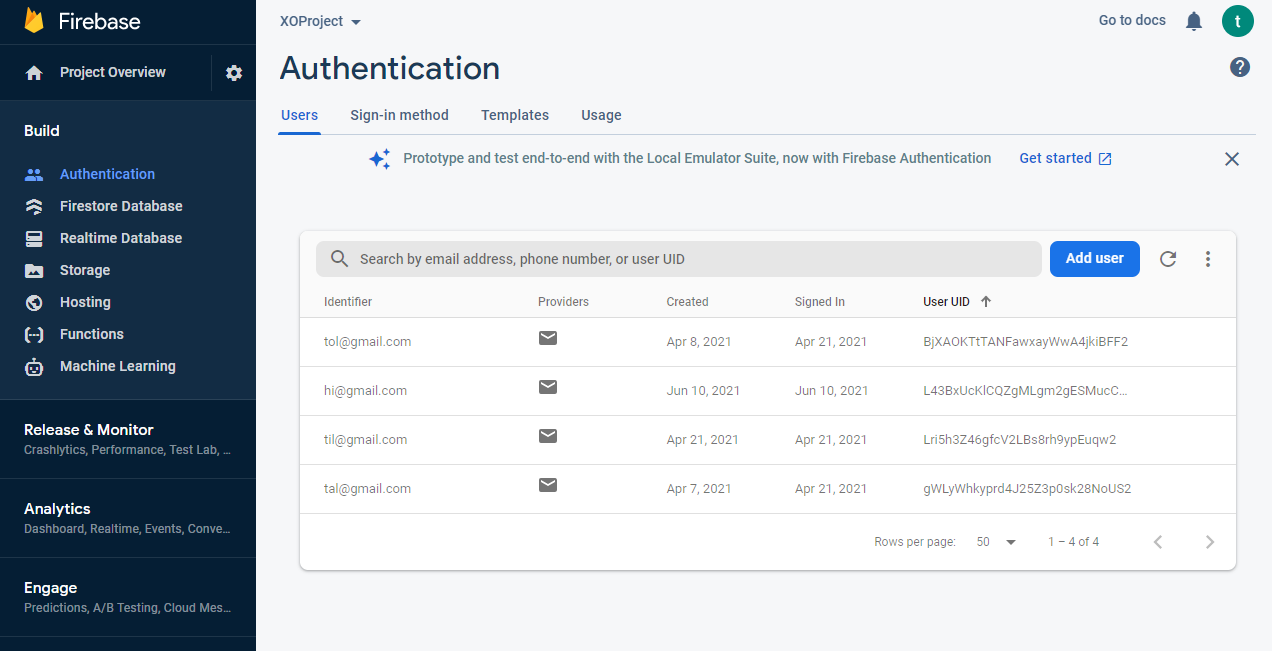
## דרישות מיוחדות ומגבלות- גישה לרשת האינטרנט

# גירסאות עליהן נבדקה המערכת-



בסיס נתונים – פיירבייס

האפליקציה משתמש בבסיס הנתונים פיירביס לשמירת המשתמשים:



כמו כן, האפליקציה משתמשת בדאטבייס מסוג אס קיו אל כדי לשמור עבור כל משתמש (לפי הקוד הייחודי של כל משתמש) את הפרטים האישיים שלו שישמו אותנו ברשימת השיאים (שם ומין) שלו:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

על מסד הנתונים הפנימי (אס קיו אל) נרחיב בהמשך במחלקות האפליקציה.

מדריך למפתח

## Activities

**SplashScreen**

מסך הטעינה אשר מחכה שש שניות ועובר למסך הכניסה, במקביל מפעיל את הסרביס של מוזיקה.

package tal.XOProject.tictactoe;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
public class SplashScreen extends AppCompatActivity {  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_splash\_screen);  
 Intent svc=new Intent(this, BackgroundSoundService.class);  
 startService(svc);  
 new Handler().postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 Intent intent = new Intent(SplashScreen.this, LoginActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 }, 6000);  
  
 }  
  
}

**LoginActivity**

מסך הכניסה, מכיל את כלל הפרמטרים של הלייאוט, סטרינג לאימייל וגישה לפיירבייס, בלחיצה על לוגין מופעל אלגוריתם כניסה של גוגל הבודק את המשתמש בשרת, במידה והכל תקין עובר למסך הראשי. יש אופציה לעבור למסכים איפוס סיסמא והרשמה

package tal.XOProject.tictactoe;  
  
import android.app.AlarmManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Vibrator;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class LoginActivity extends AppCompatActivity {  
 private TextView textView,forgot;  
 private Button loginbtn;  
 private EditText useremail,userpassword;  
 private FirebaseAuth firebaseAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_login);  
  
 textView=(TextView)findViewById(R.id.reg);  
 textView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(getApplicationContext(), Register.class));  
 }  
 });  
  
 forgot=(TextView)findViewById(R.id.forgetpassword);  
 forgot.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(getApplicationContext(), Forgot\_pass.class));  
 }  
 });  
 useremail= (EditText) findViewById(R.id.email1);  
 userpassword=(EditText)findViewById(R.id.password1);  
 firebaseAuth= FirebaseAuth.getInstance();  
 loginbtn=(Button) findViewById(R.id.loginbtn);  
 loginbtn.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View view) {  
 if (validate()) {  
 final String user\_email=useremail.getText().toString().trim();  
 final String user\_password=userpassword.getText().toString().trim();  
 try {  
 firebaseAuth.signInWithEmailAndPassword(user\_email, user\_password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 Toast.makeText(getApplicationContext(), "Successfully Logged In", Toast.LENGTH\_LONG).show();  
 startActivity(new Intent(getApplicationContext(), ModeSelectActivity.class));  
 } else {  
 Toast.makeText(getApplicationContext(), "Log In Failed", Toast.LENGTH\_LONG).show();  
 }  
 }  
 });  
 }  
 catch (IllegalArgumentException e)  
 {  
 Toast.makeText(getApplicationContext(), "Illegal Arguments", Toast.LENGTH\_LONG).show();  
 }  
 }  
 }  
 });  
 }  
 private boolean validate(){  
 return true;  
 }  
  
}

}  
  
 @Override  
 public void onClick(View view) {  
 if (view==btnLogIn){  
 login();  
 }  
 if (view==btnForgotPass){  
 Intent intent = new Intent(LoginActivity.this, ForgotPasswordActivity.class);  
 startActivity(intent);  
 }  
 if (view==btnRegister){  
 Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);  
 startActivity(intent);  
 }  
 }  
}

**Forgot\_pass**

מסך איפוס סיסמה מכיל את כלל הפרמטרים של הלייאוט ועוד סטרינג אימייל וגישה לפיירבייס, מכיל את האלגוריתם של גוגל לאיפוס סיסמא משתמש פיירבייס

package tal.XOProject.tictactoe;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class Forgot\_pass extends AppCompatActivity {  
  
 private EditText email\_et;  
 private Button send\_password\_reset\_btn;  
 private FirebaseAuth firebaseAuth;  
 private String email\_string;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_forgot\_pass);  
  
 firebaseAuth= FirebaseAuth.getInstance();  
 email\_et=findViewById(R.id.emailet);  
 send\_password\_reset\_btn=findViewById(R.id.send\_password\_reset\_btn);  
 send\_password\_reset\_btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 email\_string=email\_et.getText().toString().trim();  
 if (validate())  
 {  
 try {  
 firebaseAuth.sendPasswordResetEmail(email\_string).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 Toast.makeText(getApplicationContext(), "Password reset email was sent", Toast.LENGTH\_LONG).show();  
 finish();  
 } else {  
 Toast.makeText(getApplicationContext(), "Password reset failed", Toast.LENGTH\_LONG).show();  
 }  
 }  
 });  
 }  
 catch (Exception e)  
 {  
 Toast.makeText(getApplicationContext(), "Invalid/Empty Email", Toast.LENGTH\_LONG).show();  
 }  
 }  
 }  
 });  
 }  
  
 public boolean validate()  
 {  
 if (email\_string.isEmpty())  
 {  
 Toast.makeText(getApplicationContext(),"Email cant be empty",Toast.LENGTH\_LONG).show();  
 return false;  
 }  
 return true;  
 }  
}

**Register**

מסך הרשמה מכיל את הפרמטרים מהלייאוט וגישה לפיירבייס, מכיל את האלגוריתם של גוגל להרשמת משתמש חדש ויצירת עצם חדש מטיפוס יוזר אשר נשלח לפיירבייס בעת יצירת משתמש חדש.

package tal.XOProject.tictactoe;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.Objects;  
  
  
public class Register extends AppCompatActivity {  
 private TextView userlogin;  
 private EditText username,userpassword,useremail,userphone;  
 private Button regButton;  
 private FirebaseAuth firebaseAuth;  
 private RadioButton rbmale;  
 private RadioGroup radioGroup;  
 private String gender="";  
 private int id=1;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_register);  
 radioGroup = findViewById(R.id.rb);  
 rbmale = findViewById(R.id.maleBt);  
 rbmale.setId(id);  
 username=(EditText)findViewById(R.id.user2);  
 userpassword=(EditText)findViewById(R.id.password2);  
 useremail=(EditText)findViewById(R.id.email2);  
 regButton=(Button) findViewById(R.id.send);  
 userlogin=(TextView) findViewById(R.id.reg2);  
 userphone=(EditText)findViewById(R.id.phone2);  
  
 firebaseAuth= FirebaseAuth.getInstance();  
 regButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if (validate()) {  
 final String user\_name = username.getText().toString().trim();  
 final String user\_email = useremail.getText().toString().trim();  
 final String user\_password = userpassword.getText().toString().trim();  
 final String user\_phone = userphone.getText().toString().trim();  
 int genderId = radioGroup.getCheckedRadioButtonId();  
 if (genderId==id){gender="male";}  
 else {gender = "female";}  
 try  
 {  
 firebaseAuth.createUserWithEmailAndPassword(user\_email, user\_password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 sendUserData(user\_name, user\_email,gender, user\_password, user\_phone);  
 Toast.makeText(getApplicationContext(), "Successfully Registered", Toast.LENGTH\_LONG).show();  
 startActivity(new Intent(tal.XOProject.tictactoe.Register.this, ModeSelectActivity.class));  
 } else {  
 Toast.makeText(getApplicationContext(), "Registration Failed", Toast.LENGTH\_LONG).show();  
 }  
 }  
 });  
 }  
 catch (IllegalArgumentException e)  
 {  
 Toast.makeText(getApplicationContext(), "Illegal Arguments", Toast.LENGTH\_LONG).show();  
 }  
  
 }  
 }  
 });  
  
 userlogin.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 finish();  
 }  
 });  
 }  
  
 private void sendUserData(String username,String email, String gender, String password,String phone) {  
 FirebaseDatabase firebaseDatabase=FirebaseDatabase.getInstance();  
 firebaseAuth= FirebaseAuth.getInstance();  
 DatabaseReference myRef = firebaseDatabase.getReference("Users/"+Objects.requireNonNull(firebaseAuth.getUid())+"/UserData");  
 Userprofile userprofile=new Userprofile(phone,email,gender,username,password);  
 myRef.setValue(userprofile);  
 }  
 private boolean validate(){  
 return true;  
 }  
}

**ModeSelectActivity**

מסך הבחירה הראשי, מכיל משתנה גלובאלי של מצב המשחק 1 לשחקן נגד שחקן ו2 לשחקן נגד מחשב.

package tal.XOProject.tictactoe;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.tts.TextToSpeech;  
import android.view.View;  
import android.widget.Button;  
  
public class ModeSelectActivity extends AppCompatActivity {  
 public static int mode =0;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_mode\_select);  
 }  
 public void pvp(View v) {  
 mode=1;  
 Intent intent= new Intent(this,MainActivity.class);  
 startActivity(intent);  
 }  
 public void pcvp(View v) {  
 mode=2;  
 Intent intent= new Intent(this,MainActivity.class);  
 startActivity(intent);  
 }  
  
}

**MainActivity**

מסך המשחק עצמו, מכיל את האלמנטים מהליאוט, משתנה גלובאלי שמייצג את הקוד תגובה לסיום המשחק (1), אלמנט טקסט טו ספיץ' וקונרולר לשילטה במשחק, במסך הזה קורה עיקר המשחק, אלגוריתם המשחק האוטומטי של המחשב, מכיל את האלגוריתם לאתחול סטופר לזמן תור של שחקן, את הקוד של התפריט (שליחה למסך התוצאות ושיתוף הפאליקציה).

package tal.XOProject.tictactoe;  
  
import android.app.AlarmManager;  
import android.app.PendingIntent;  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.tts.TextToSpeech;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.Locale;  
import java.util.Random;  
  
import static tal.XOProject.tictactoe.ModeSelectActivity.mode;  
  
  
public class MainActivity extends AppCompatActivity {  
 private static final int GAME\_OVER = 1;  
 private Controller controller;  
 private LinearLayout llmain;  
 private ImageView imageView;  
 private TextToSpeech tts;  
 private TextView txtmode;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 controller = new Controller();  
 llmain = findViewById(R.id.mainBoard);  
 imageView = findViewById(R.id.imgPlayer);  
 txtmode = findViewById(R.id.txtmode);  
 if(mode==1){  
 txtmode.setText("Player vs Player");  
 }  
 else{  
 txtmode.setText("PC vs Player");  
 }  
 controller.startGame();  
  
 tts = new TextToSpeech(this, new TextToSpeech.OnInitListener() {  
 @Override  
 public void onInit(int status) {  
 if (status!= TextToSpeech.ERROR) {tts.setLanguage(Locale.US);}  
 }  
 });  
 }  
  
 public void select(View v) {  
  
 // להעביר את זהות הכפתור הנלחץ לבקר  
 //0-8  
 int loc=0;  
 char player = controller.getPlayer();  
 Button b = findViewById(R.id.btn0);  
 Random rand = new Random();  
 if(mode==1||player=='x'){  
 stopAlert();  
 startAlert();  
 loc = Integer.parseInt(v.getTag().toString());  
 b = (Button) v;  
 }  
 else{  
 while(!controller.checkEmptyPlace(loc)){  
 loc=rand.nextInt(9);  
 }  
 switch (loc){  
 case 0:  
 b=findViewById(R.id.btn0);  
 break;  
 case 1:  
 b=findViewById(R.id.btn1);  
 break;  
 case 2:  
 b=findViewById(R.id.btn2);  
 break;  
 case 3:  
 b=findViewById(R.id.btn3);  
 break;  
 case 4:  
 b=findViewById(R.id.btn4);  
 break;  
 case 5:  
 b=findViewById(R.id.btn5);  
 break;  
 case 6:  
 b=findViewById(R.id.btn6);  
 break;  
 case 7:  
 b=findViewById(R.id.btn7);  
 break;  
 case 8:  
 b=findViewById(R.id.btn8);  
 break;  
 }  
 }  
 //geting last player and send location  
 player = controller.userSelect(loc);  
 // משנה את התמונה על הכפתור  
 if (player == 'x') {b.setBackgroundResource(R.drawable.x);}  
 else b.setBackgroundResource(R.drawable.o);  
  
 //לנטרל את הכפתור  
 b.setEnabled(false);  
  
 //presenting the player  
 char corent = controller.getPlayer();  
 if (corent=='x'){imageView.setImageResource(R.drawable.x);}  
 else imageView.setImageResource(R.drawable.o);  
 if (controller.checkforwinner()){  
 tts.speak("game over", TextToSpeech.QUEUE\_FLUSH,null);  
 stopAlert();  
 Intent intent= new Intent(this,GameOverActivity.class);  
 intent.putExtra("winner",controller.getWinner());  
 startActivityForResult(intent,GAME\_OVER);  
 }  
 else if(mode==2&&player=='x'){  
 select(v);  
 }  
 }  
 public void changeMode(View view) {  
 Intent intent = new Intent(MainActivity.this, ModeSelectActivity.class);  
 startActivity(intent);  
 }  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == GAME\_OVER && resultCode == RESULT\_OK) {  
 String answer = data.getStringExtra("victoryQuote") ;  
 Toast.makeText(this, answer, Toast.LENGTH\_SHORT).show();  
 newGame();  
 }  
 }  
  
 public void startGame(View view) {  
 newGame();  
 }  
  
 public void newGame(){  
  
 for (int i=0;i<9;i++){  
 Button b = llmain.findViewWithTag(String.valueOf(i));  
 b.setBackgroundResource(android.R.drawable.btn\_default);  
 b.setEnabled(true);  
 }  
 controller.startGame();  
 }  
 public void startAlert() {  
 int i = 30;  
 Intent intent = new Intent(this, MyBroadcastReceiver.class);  
 PendingIntent pendingIntent = PendingIntent.getBroadcast(  
 this.getApplicationContext(), 234324243, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);  
 AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);  
 assert alarmManager != null;  
 alarmManager.set(AlarmManager.RTC\_WAKEUP, System.currentTimeMillis()  
 + (i \* 1000), pendingIntent);  
 //Toast.makeText(this, "your turn end in " + i + " seconds",Toast.LENGTH\_LONG).show();  
 }  
 public void stopAlert() {  
 Intent intent = new Intent(this, MyBroadcastReceiver.class);  
 PendingIntent pendingIntent = PendingIntent.getBroadcast(  
 this.getApplicationContext(), 234324243, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);  
 AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);  
 assert alarmManager != null;  
 alarmManager.cancel(pendingIntent);  
 }  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.main\_manu, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 if (item.getItemId()==R.id.share\_item){  
 Toast.makeText(this, "share item chosen", Toast.LENGTH\_SHORT).show();  
 Intent sendIntent = new Intent();  
 sendIntent.setAction(Intent.ACTION\_SEND);  
 sendIntent.putExtra(Intent.EXTRA\_TEXT, "I'm playing tic tac toe");  
 sendIntent.setType("text/plain");  
 startActivity(sendIntent);  
 }  
 else if (item.getItemId()==R.id.list\_item){  
 Intent intent = new Intent(this, ListActivity.class);  
 startActivity(intent);  
 }  
 return true;  
 }  
}

**GameOverActivity**

מסך סיום המשחק, מכיל אלמנטים מהלייאוט, משתנה גלובלי לתגובת הטקסט טו ספיץ'(2) וסטרינגים לשמירת נתוני המשתמש המחובר לפיירבייס כדי לשמור אותם באס קיו אל. הקוד מציג מי ניצח לפי המידע שעבר ממשחק המשחק למסך סיום המשחק, מכיל את אלגוריתם הזיהוי קול לטקסט, הבאת נתונים מהפיירבייס לפי המשתמש המחובר ושמירת התוצאה במסד הנתונים אס קיו אל.

package tal.XOProject.tictactoe;  
  
import android.content.ActivityNotFoundException;  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.RecognizerIntent;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.firebase.FirebaseError;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Objects;  
  
import static tal.XOProject.tictactoe.ModeSelectActivity.mode;  
  
  
public class GameOverActivity extends AppCompatActivity {  
  
 private EditText etVictoryQuote;  
 private TextView txtwinner;  
 private ImageButton mic;  
 private Button btnsave;  
 private static final int REQ\_CODE\_SPEECH\_INPUT = 2;  
 private FirebaseAuth firebaseAuth;  
 String gender = "";  
 String userName = "";  
 String victoryQuote ="";  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_game\_over);  
 mic=findViewById(R.id.btnmic);  
 btnsave=findViewById(R.id.btn2);  
 etVictoryQuote = findViewById(R.id.etVictoryQuote);  
 txtwinner=findViewById(R.id.txtwinner);  
 btnsave.setVisibility(View.INVISIBLE);  
 Bundle extras = getIntent().getExtras();  
 char winner = extras.getChar("winner");  
 if(mode==2){  
 if(winner=='o'){  
 txtwinner.setText("The Winner Is: PC");  
 etVictoryQuote.setHint("Inspiration Quote..");  
 }  
 else if(winner=='d'){  
 txtwinner.setText("Its a draw!");  
 etVictoryQuote.setHint("Inspiration Quote..");  
 }  
 else{  
 txtwinner.setText("You Are The Winner!");  
 btnsave.setVisibility(View.VISIBLE);  
 }  
 }  
 else{  
 if(winner=='d'){  
 txtwinner.setText("Its a draw!");  
 etVictoryQuote.setHint("Inspiration Quote..");  
 }  
 else{  
 txtwinner.setText("The Winner Is: "+winner);  
 }  
 }  
 mic.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);  
 intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL, RecognizerIntent.LANGUAGE\_MODEL\_FREE\_FORM);  
 intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE, "en");  
 intent.putExtra(RecognizerIntent.EXTRA\_PROMPT,  
 getString(R.string.speech));  
 try {  
 startActivityForResult(intent, REQ\_CODE\_SPEECH\_INPUT);  
 } catch (ActivityNotFoundException a) {  
 Toast.makeText(getApplicationContext(),  
 getString(R.string.speech\_not\_supported),  
 Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == REQ\_CODE\_SPEECH\_INPUT) {  
 if (resultCode == RESULT\_OK && null != data) {  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA\_RESULTS);  
 assert result != null;  
 etVictoryQuote.setText(result.get(0));  
 }  
 }  
 }  
 public void startover(View view) {  
 victoryQuote = etVictoryQuote.getText().toString();  
 if (victoryQuote.isEmpty()){victoryQuote = "No Victory Quote";}  
 Intent intent1= new Intent();  
 intent1.putExtra("victoryQuote", victoryQuote);  
 setResult(RESULT\_OK, intent1);  
 finish();  
 }  
  
 public void save(View view) {  
 // creation of Player instance!  
 victoryQuote = etVictoryQuote.getText().toString();  
 if (victoryQuote.isEmpty()){victoryQuote = "No Victory Quote";}  
 FirebaseDatabase firebaseDatabase=FirebaseDatabase.getInstance();  
 firebaseAuth= FirebaseAuth.getInstance();  
 DatabaseReference myRef = firebaseDatabase.getReference("Users/"+Objects.requireNonNull(firebaseAuth.getUid())+"/UserData");  
 myRef.addValueEventListener(new ValueEventListener(){  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 for(DataSnapshot datas: dataSnapshot.getChildren()){  
 if(datas.getKey().equals("gender"))  
 gender = datas.getValue().toString();  
 if(datas.getKey().equals("name"))  
 userName = datas.getValue().toString();  
 if(datas.getKey().equals("phone")){  
 Player player = new Player(userName, victoryQuote, 1 , gender);  
  
 // create DB (Table)  
 PlayerDataBase playerDataBase = new PlayerDataBase(GameOverActivity.this);  
  
 // upload the data---> save!  
 playerDataBase.setRecord(player);  
  
 // move to ListActivity  
 Intent intent = new Intent(GameOverActivity.this, ListActivity.class);  
 startActivity(intent);  
 }  
  
 }  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 Toast.makeText(GameOverActivity.this, "error saving data", Toast.LENGTH\_SHORT).show();  
 Intent intent = new Intent(GameOverActivity.this, MainActivity.class);  
 startActivity(intent);  
 }  
 });  
 }  
}

**ListActivity**

מסך רשימת המשחקים, מכיל את הליסטויו, רשימת שחקנים מטיפוס שחקן, אדאפטר מטיפוס פלייאר אדאפטר. המסך מביא את כל השחקנים השמורים לו בדאטבייס ומציג אותם ברשימה, בלחיצה ארוכה על איבר ברשימה יופיע דיאלוג בוקס למחיקת המשתמש (רק התוצאה), במידה והמשתמש יבחר כן, האיבר ימחק מהדאטבייס

package tal.XOProject.tictactoe;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ListView;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class ListActivity extends AppCompatActivity implements AdapterView.OnItemLongClickListener {  
  
 private ListView listView;  
 private ArrayList<Player> players;  
 private PlayerAdapter adapter;  
 private PlayerDataBase playerDataBase;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_list\_view);  
  
 listView = findViewById(R.id.listView);  
  
 //create DB instance;  
 playerDataBase = new PlayerDataBase(this);  
 //pull data to Array  
 players = PlayerDataBase.getPlayers();  
  
 //players = playerDataBase.getAllRecords();  
 adapter = new PlayerAdapter(this, players);  
  
 listView.setAdapter(adapter);  
  
 listView.setOnItemLongClickListener(this);  
  
 }  
 public void toMain(View view) {  
 Intent intent = new Intent(ListActivity.this, MainActivity.class);  
 startActivity(intent);  
 }  
 @Override  
 public boolean onItemLongClick(AdapterView<?> adapterView, View view, final int position, long l) {  
 AlertDialog.Builder alert = new AlertDialog.Builder(this);  
 alert.setTitle("delete player");  
 alert.setMessage("are you sure you want to delete player?");  
 alert.setIcon(R.drawable.x);  
 alert.setCancelable(false);  
  
 alert.setPositiveButton("delete", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int which) {  
 playerDataBase.deletePlayerByRow(players.get(position).getId());  
 PlayerDataBase.getPlayers().remove(position);  
 adapter.notifyDataSetChanged();  
 dialogInterface.dismiss();  
 }  
 });  
  
 alert.setNegativeButton("cancel", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 dialogInterface.dismiss();  
 }  
 });  
 alert.create();  
 alert.show();  
 adapter.notifyDataSetChanged();  
 return true;  
 }  
}

## Classes

**מחלקת Controller**

המחלקה אחראית על שליטת השחקן במשחק, כלומר בפעולה הבונה להתחיל משחק חדש, לאתחל את השחקן לאיקס, להודיע למחלקת מודל על תחילת משחק. בפעולת בחירת שחקן הפעולה מקבלת את מיקום הבחירה והפעולה תסמן עבור השחקן את המיקום שבחר ותעביר את התור לשחקן השני, ותחזיר מי השחקן ששיחק. המחלקה מכילה גם בדיקות כמו האם יש מנצח או האם מיקום מסויים ריק או לא.

package tal.XOProject.tictactoe;  
  
public class Controller {  
 private Model model;  
 private char player;  
  
 public Controller(){  
 model = new Model();  
 startGame();  
 }  
  
 public void startGame() {  
 player='x';  
 model.startGame();  
 }  
 //בקבלת מיקום שמשתמש בחר, יש החלפת שחקן והחזרת השחקן הקודם  
 public char userSelect(int loc){  
 char prev = player;  
 model.setPlace(loc,player);  
 if (player=='x') {player='o';}  
 else player='x';  
 return prev;  
 }  
 //check if gameover  
 public boolean checkforwinner(){  
 return model.gameOver();  
 }  
  
 public boolean checkEmptyPlace(int loc){  
 return model.isEmptyPlace(loc);  
 }  
  
 public char getPlayer() {  
 return player;  
 }  
  
 public char getWinner() {  
 return model.getWinner();  
 }  
}

## מחלקת Model

המחלקה לניהול לוח המשחק, מכילה טבלה דו מימדית של הלוח מסוג תו, את תו המנצח וסופר תורות. המחלקה מכילה פעולות של התחלת משחק ואיפוס לוח (כל התווים הם רווח). לבדוק האם מיקום הוא פנוי, הצבת סמן לשחקן במיקום מסויים, בדיקה האם המשחק נגמר ומישהו ניצח.

package tal.XOProject.tictactoe;  
  
public class Model {  
 private char[][] board;  
 private final int SIZE = 3;  
 private char winner;//'x' or 'o' or 'd'  
 private int counter;  
  
 public Model(){  
 board = new char[SIZE][SIZE];  
 }  
  
 public void startGame() {  
 //איפוס לוח  
 //איפוס של המנצח  
 counter = 0;  
 winner = ' ';  
 for (int i=0;i<SIZE;i++){  
 for (int k=0;k<SIZE;k++){  
 board[i][k]=' ';  
 }  
 }  
 }  
 public boolean isEmptyPlace(int loc){  
 int row = loc/SIZE; //0/1/2  
 int col = loc%SIZE; //0/1/2  
 if(board[row][col]=='x'||board[row][col]=='o'){  
 return false;  
 }  
 return true;  
 }  
 // מקבל מיקום בלוח ותתרגם אותו למיקום במערך בצורת מיקום במערך דו מימדי ותציב לי משתמש במיקום  
 public void setPlace(int loc,char player){  
 //loc=0-8 ==>0/1/2 player=x/o  
 int row = loc/SIZE; //0/1/2  
 int col = loc%SIZE; //0/1/2  
 //הצבת שחקן במערך  
 board[row][col]= player;  
 //קידום טור  
 counter++;  
 }  
  
 //winner check ==> returns true if gameover  
 public boolean gameOver()  
 {  
 if (counter<5)  
 {return false;}  
 // 5-9  
 for (int i = 0;i<SIZE;i++)  
 {  
 if (board[i][0]!=' '&& board[i][0]==board[i][1]&& board[i][1]==board[i][2])  
 {  
 winner = board[i][0];  
 return true;  
 }  
 }  
 for (int i = 0;i<SIZE && winner==' ';i++)  
 {  
 if (board[0][i]!=' '&& board[0][i]==board[1][i]&& board[1][i]==board[2][i])  
 {  
 winner = board[0][i];  
 return true;  
 }  
 }  
 if (board[0][0]!=' '&& board[0][0]==board [1][1]&& board[1][1]==board[2][2])  
 {  
 winner=board[0][0];  
 return true;  
 }  
 if (board[0][2]!=' '&& board[0][2]==board[1][1]&& board[1][1]==board[2][0])  
 {  
 winner=board[2][0];  
 return true;  
 }  
 if (counter==9)  
 {  
 winner = 'd';  
 return true;  
 }  
 return false;  
 }  
  
 public char getWinner() {  
 return winner;  
 }  
}

## מחלקת Userprofile

מחלקה לשמירה זמנית של הנתונים של המשתמש ושליחתם לפיירבייס

package tal.XOProject.tictactoe;  
  
public class Userprofile {  
 public String phone;  
 public String email;  
 public String name;  
 public String password;  
 public String gender;  
  
 public Userprofile(String phone1, String email1, String gender1, String name1, String password1)  
 {  
 this.phone=phone1;  
 this.email=email1;  
 this.name=name1;  
 this.gender=gender1;  
 this.password=password1;  
 }  
  
}

## מחלקת Player

## המחלקה מכילה את כל הנתונים שיש לשחקן – שם, ציטוט נצחון אחרון, שיא, מין ומספר מזהה, למחלקה כלל הפעולות הסטנדרטיות של מחלקה.

package tal.XOProject.tictactoe;  
  
// class modeling!!!!  
public class Player implements Comparable {  
 // fillds  
 private String name;  
 private String victoryQuote;  
 private int score;  
 private String gender;  
 private long id;  
  
 // constructors  
 public Player (String name, String victoryQuote, int score, String gender) {  
 this.name = name;  
 this.victoryQuote = victoryQuote;  
 this.score = score;  
 this.gender = gender;  
 }  
  
 public Player(String name, String victoryQuote, int score, String gender, long id) {  
 this.name = name;  
 this.victoryQuote = victoryQuote;  
 this.score = score;  
 this.gender = gender;  
 this.id = id;  
 }  
  
 //getters and setters  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getVictoryQuote() {  
 return victoryQuote;  
 }  
  
 public void setVictoryQuote(String victoryQuote) {  
 this.victoryQuote = victoryQuote;  
 }  
  
 public int getScore() {  
 return score;  
 }  
  
 public void setScore(int score) {  
 this.score = score;  
 }  
  
 public String getGender() {  
 return gender;  
 }  
  
 public void setGender(String gender) {  
 this.gender = gender;  
 }  
  
 public long getId() {  
 return id;  
 }  
  
 public void setId(long id) {  
 this.id = id;  
 }  
  
 //methods  
 public void addScore(){  
 score++;  
 }  
  
 @Override  
 public int compareTo(Object o) {  
 if (o instanceof Player) {  
 Player other = (Player) o;  
 return name.compareTo(other.name);  
 }//names comparison  
 //return other.score - this.score;}  
 return 0;  
 }  
}

## PlayerDataBase

הדאטבייס מסוג אס קיו אל אשר שומר את תוצאותיו של משתמש לאחר משחק, מכיל את כלל הפעולות הסטנדרטיות של מסד נתונים כמו יצירה, עדכון ומחיקה של רשומות.

package tal.XOProject.tictactoe;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import java.util.ArrayList;  
import java.util.Collections;  
  
public class PlayerDataBase extends SQLiteOpenHelper {  
 private static final String DATABASENAME = "score.db";  
 private static final String TABLE\_RECORD = "tblscore";  
 private static final int DATABASEVERSION = 1;  
 private static final String COLUMN\_ID = "Id";  
 private static final String COLUMN\_NAME = "Name";  
 private static final String COLUMN\_VICTORYQUOTE = "VictoryQuote";  
 private static final String COLUMN\_SCORE = "Score";  
 private static final String COLUMN\_GENDER = "Gender";  
 private static final String[] allColumns = {COLUMN\_ID, COLUMN\_NAME, COLUMN\_VICTORYQUOTE, COLUMN\_SCORE, COLUMN\_GENDER};  
 private static ArrayList<Player> players;  
  
 private static final String CREATE\_TABLE\_CUSTOMER = "CREATE TABLE IF NOT EXISTS " +  
 TABLE\_RECORD + "(" +  
 COLUMN\_ID + " INTEGER PRIMARY KEY," +  
 COLUMN\_NAME + " TEXT," +  
 COLUMN\_VICTORYQUOTE + " TEXT," +  
 COLUMN\_GENDER + " TEXT," +  
 COLUMN\_SCORE + " INTEGER );";  
  
 private SQLiteDatabase database;  
  
 public PlayerDataBase(Context context) {   
 super(context, DATABASENAME, null, DATABASEVERSION);  
 getAllRecords();  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase sqLiteDatabase) {  
 sqLiteDatabase.execSQL(CREATE\_TABLE\_CUSTOMER);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {  
 sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE\_RECORD);  
 onCreate(sqLiteDatabase);  
 }  
  
 public void setRecord(Player player) {  
 Player current = null;  
 for (Player p: players)  
 {  
 if(p.getName().equalsIgnoreCase(player.getName()))  
 {  
 current=p;  
 break;  
 }  
 }  
 if(current==null) {  
 players.add(player);  
 createRecord(player);  
 }  
 else {  
 current.addScore();  
 current.setVictoryQuote(player.getVictoryQuote());  
 updateByRow(current);  
 }  
 }  
  
// write record  
 public Player createRecord(Player record) {  
 database = getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(COLUMN\_NAME, record.getName());  
 values.put(COLUMN\_VICTORYQUOTE, record.getVictoryQuote());  
 values.put(COLUMN\_GENDER, record.getGender());  
 values.put(COLUMN\_SCORE, record.getScore());  
 long id = database.insert(TABLE\_RECORD, null, values);  
 record.setId(id);  
 database.close();  
 return record;  
 }  
// read records  
 private ArrayList<Player> getAllRecords() {  
 database = getReadableDatabase();  
 players = new ArrayList<>();  
 String sortOrder = COLUMN\_SCORE + " DESC"; // sorting by score  
 Cursor cursor = database.query(TABLE\_RECORD, allColumns, null, null, null, null, sortOrder);  
 if (cursor.getCount() > 0) {  
 while (cursor.moveToNext()) {  
 String name = cursor.getString(cursor.getColumnIndex(COLUMN\_NAME));  
 String victoryQuote = cursor.getString(cursor.getColumnIndex(COLUMN\_VICTORYQUOTE));  
 String gender = cursor.getString(cursor.getColumnIndex(COLUMN\_GENDER));  
 int score = cursor.getInt(cursor.getColumnIndex(COLUMN\_SCORE));  
 long id = cursor.getLong(cursor.getColumnIndex(COLUMN\_ID));  
 Player record = new Player(name,victoryQuote, score, gender, id);  
 players.add(record);  
 }  
 }  
 database.close();  
 return players;  
 }  
  
//delete record  
 public void deletePlayerByRow(long id) {  
 database = getWritableDatabase();  
 database.delete(TABLE\_RECORD, COLUMN\_ID + " = " + id, null);  
 database.close();  
 }  
  
//update record  
 public void updateByRow(Player player) {  
 database = getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(COLUMN\_ID, player.getId());  
 values.put(COLUMN\_NAME, player.getName());  
 values.put(COLUMN\_VICTORYQUOTE, player.getVictoryQuote());  
 values.put(COLUMN\_GENDER, player.getGender());  
 values.put(COLUMN\_SCORE, player.getScore());  
 database.update(TABLE\_RECORD, values, COLUMN\_ID + "=" + player.getId(), null);  
 database.close();  
 }  
  
 // setters and getters  
 public static ArrayList<Player> getPlayers() {  
 return players;  
 }  
  
 public static void setPlayers(ArrayList<Player> players) {  
 PlayerDataBase.players = players;  
 }  
}

## Adapters

האדפטר מקשר בין כל אחד מהאיברים ברשימה לרשימה עצמה, הוא יודע להחליף לתמונה של זכר או נקבה בהתאם למין המשתמש.

package tal.XOProject.tictactoe;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import java.util.ArrayList;  
  
public class PlayerAdapter extends ArrayAdapter<Player> {  
  
 private Context context;  
 private ArrayList<Player> list;  
  
 public PlayerAdapter(@NonNull Context context, ArrayList<Player> list) {  
 super(context, R.layout.item\_player ,list);  
 this.context = context;  
 this.list = list;  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater inflater=(LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);  
  
 View rowView= inflater.inflate(R.layout.item\_player,parent,false);  
 //UserName  
 TextView tvUserName = rowView.findViewById(R.id.tvPlayerUN);  
 tvUserName.setText(list.get(position).getName());  
 //Victory Quote  
 TextView tvVictoryQuote = rowView.findViewById(R.id.tvPlayerVQ);  
 tvVictoryQuote.setText(""+list.get(position).getVictoryQuote());  
 //score  
 TextView tvScore = rowView.findViewById(R.id.tvScore);  
 tvScore.setText(""+list.get(position).getScore());  
 //pic  
 ImageView imageView=rowView.findViewById(R.id.imgPlayer);  
 if (list.get(position).getGender().equalsIgnoreCase("male"))  
 {imageView.setImageResource(R.drawable.male);}  
 else {imageView.setImageResource(R.drawable.female);}  
  
 return rowView;  
 }  
}

## Broadcast receiver

המשתמש יקבל הודעה כאשר נגמר לו התור (חלפו 30 שניות) והפלאפון ירטוט למשך 2 שניות

public void startAlert() {  
 int i = 30;  
 Intent intent = new Intent(this, MyBroadcastReceiver.class);  
 PendingIntent pendingIntent = PendingIntent.getBroadcast(  
 this.getApplicationContext(), 234324243, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);  
 AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);  
 assert alarmManager != null;  
 alarmManager.set(AlarmManager.RTC\_WAKEUP, System.currentTimeMillis()  
 + (i \* 1000), pendingIntent);  
 //Toast.makeText(this, "your turn end in " + i + " seconds",Toast.LENGTH\_LONG).show();  
}  
public void stopAlert() {  
 Intent intent = new Intent(this, MyBroadcastReceiver.class);  
 PendingIntent pendingIntent = PendingIntent.getBroadcast(  
 this.getApplicationContext(), 234324243, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);  
 AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);  
 assert alarmManager != null;  
 alarmManager.cancel(pendingIntent);  
}

package tal.XOProject.tictactoe;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Vibrator;  
import android.widget.Toast;  
  
public class MyBroadcastReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 Toast.makeText(context, "your time is up!",  
 Toast.LENGTH\_LONG).show();  
 // Vibrate the mobile phone  
 Vibrator vibrator = (Vibrator) context.getSystemService(Context.VIBRATOR\_SERVICE);  
 assert vibrator != null;  
 vibrator.vibrate(2000);  
 }  
  
}

## Menu

התפריט מוצג אך ורק במסך המשחק, שם הוא יכול לבחור לשתף את המשחק (שכן לא כדאי שיעשה את זה לפני ששיחק בעצמו) ואפשרות ללכת לטבלת השיאים.

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.main\_manu, menu);  
 return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 if (item.getItemId()==R.id.share\_item){  
 Toast.makeText(this, "share item chosen", Toast.LENGTH\_SHORT).show();  
 Intent sendIntent = new Intent();  
 sendIntent.setAction(Intent.ACTION\_SEND);  
 sendIntent.putExtra(Intent.EXTRA\_TEXT, "I'm playing tic tac toe");  
 sendIntent.setType("text/plain");  
 startActivity(sendIntent);  
 }  
 else if (item.getItemId()==R.id.list\_item){  
 Intent intent = new Intent(this, ListActivity.class);  
 startActivity(intent);  
 }  
 return true;  
}

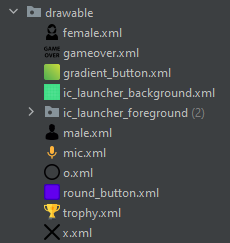
## service

הסרוויס משמש להשמעת מוזיקת רקע בלופים מיד בכניסה לאפליקציה ועד סגירת האפליקציה.

package tal.XOProject.tictactoe;  
  
import android.app.Service;  
import android.content.Intent;  
import android.media.MediaPlayer;  
import android.os.IBinder;  
  
public class BackgroundSoundService extends Service {  
 MediaPlayer player;  
 public IBinder onBind(Intent arg0) {  
 return null;  
 }  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 player = MediaPlayer.create(this, R.raw.backmusic);  
 player.setLooping(true); // Set looping  
 player.setVolume(100,100);  
  
 }  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 player.start();  
 return START\_STICKY;  
 }  
   
 public IBinder onUnBind(Intent arg0) {  
 // TO DO Auto-generated method  
 return null;  
 }  
  
 @Override  
 public void onDestroy() {  
 player.stop();  
 player.release();  
 }  
  
 @Override  
 public void onLowMemory() {  
  
 }  
}

## Drawable

ברשימת התמונות של האפליקציה יש אלמנטים עיצוביים של האפליקציה, אייקונים ותמונות של רכיבים באפליקציה.



## Layout

רשימת המסכים של האפליקציה בתוספת עיצוב עצם אחד של שחקן ברשימת השיאים

## תמונה שמכילה טקסט התיאור נוצר באופן אוטומטי

## Raw

מכיל אנימציה מסוג ג'ייסון ומוזיקת הרקע מסוג אם פי 3

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

## Manu

התפריט

## mipmap

אייקון הפאליקציה

סיכום אישי

הפרויקט היה עבורה חוויה לא פשוטה אבל בהחלט נהניתי מכל רגע, בעיקר נהניתי ללמוד על עולם חדש ומרתק של פיתוח אפליקציות ובמיוחד בגלל שזה נושא קרוב מאוד לליבי, נושא המשחקים הפשוטים לפלאפון בעולם ומשחקים נוסטלגיים. אז לא רק שנהניתי מהתהליך והתוצר הסופי אלה זה היה חשוב לי אישית כי עכשיו גם אני אוכל לשחק באפליקציה הזו עם חבריי.

אני מרגיש עכשיו שאוכל לפתח אפליקציות נוספות שישמשו אותי בעתיד בעזרת הידע שרכשתי במהלך הפרויקט, אני מרגיש שאני מכיר טוב את תהליך פיתוח של אפליקציה, איך נבנה אלגוריתם ואיך מתכנתים בשפות קוד שונות. גם ג'אווה וגם אקס אם אל לעיצוב, בנוסף לעקרונות עיצוב גרפי ונוחות למשתמש, ויש לי את הכלים להתקדם בתחום ההייטק.

אני מודה שהיה לי קשה בהצלחה להיכנס לעולם הקוד, זו הפעם הראשונה שאני עובד באמת עם קוד בכלל ועם אנדרואיד סטודיו בפרט, זו גם הפעם הראשונה שאני חווה פרויקט בסדר גודל שכזה, זה פרויקט שלם ולקח לי זמן להבין מאיפה מתחילים ואיך הדברים עובדים זה עם זה, לאחר המון שעות סרטוני הדרכה וקריאת מדריכים, יחד עם החומר שלמדתי בבית הספר, אני מרגיש שצלחתי את הפרויקט.

אין לי ספק שבעתיד אמשיך לעסוק בעולם התכנות, הפיתוח והיצירה.