

EXPLORE AND FIGHT

עבודת גמר תכנון ותכנות מערכות 883589



מגיש: ירין דרורי

ת.ז: 325769438

תיכון מקיף ג' העמית

האפליקציה נקראת בשם "Explore and Fight", היא בעצם משחק אונליין בו 2 שחקנים בודקים את הידע הגיאוגרפי שלהם על מקומות שונים בעולם, כך שהשאלות יהיו בקשר למקומות בעולם, ואזורים מפורסמים. השחקנים יכולים להשיג נקודות שמראות את ההישגים שלהם ולהרוויח מטבעות. במקרה בו שני אנשים רוצים לבדוק מי מכיר יותר טוב את העולם, הם יפתחו את האפליקציה ויתחילו לשחק! בחרתי בנושא זה, כיוון שהנושא מעניין אותי, אני אוהב ללמוד על העולם והייתי רוצה לדעת איך לפתח את זה הלאה .

יעדים

1. ליצור מסך כניסה והרשמה.
2. לעשות מסך רישום firebase .
3. ליצור את כל המסכים של הניווט באפליקציה .
4. להכין את המשחק עצמו .
5. לחבר בין 2 משתמשים וליצור להם משחק .
6. ליצור טבלת ניקוד למשתמשים.
7. להוסיף רמות קושי למשחק ותוספות לVIP .
8. לעצב את האפליקציה ולהוסיף שיפורי גרפיקה .
9. חיפוש באגים באפליקציה ותיקונם .

יכולות המערכת

- אפשרות למשחק של שני שחקנים בין שני מכשירים ע"י תקשורת אינטרנט.
- הצגה ב Google Street View של המקום הנבחר, כמו גם היכולת להסתובב בו.
- אפשרות להדליק ולכבות את המוזיקה בהגדרות.
- אפשרות לשנות את שם המשתמש בהגדרות (במידה והשם פנוי).
- חנות בה אפשר לקנות VIP ועזרה של 50/50 במשחק תמורת מטבעות.
- אפשרות לבדוק את המקום שלך לעומת אחרים בלוח המובילים (לפי נקודות).
- אפשרות ליצור משחק ולהיכנס למשחק לכל משתמש .

תפעול המערכת

תרשים זרימה בין המסכים



* ברוב המסכים יש הפניה לתפריט שמוביל לאודות המתכנת ולאודות הפרויקט.

* מסך "המשחק עצמו" כולל את מסך השאלה שמפנה אליו.

מדריך משתמש

המטרה היא לנצח את היריב שמולך על ידע כללי של מקומות בעולם, להלן המדריך למשחק:

1. במסך Play אפשר לפתוח חדר או להיכנס לחדר – בחרו את מה שתרצו.
 2. לאחר מכן תשתפו את הקוד או תיכנסו אל קוד קיים, והמשחק יתחיל!
 3. כאשר המשחק מתחיל תוכלו להסתובב או ללכת בעזרת החצים במקום בו אתם נמצאים, המטרה שלכם היא לזהות היכן אתם נמצאים – וכאשר הזמן שמופיע בצד המסך מגיע ל 0, תועברו לשאלה לגבי איפה הייתם. בשאלה אפשר לנחש אם לא יודעים, ולהיעזר ב 50/50 (אם יש ברשותכם)
- במשחק אין חשיבות למי עונה ראשון, קחו את הזמן ורק שתהיו בטוחים בתשובה תענו. כמובן יש טיימר אשר מגביל את הזמן – חובה לענות לפני שהוא נגמר!
- יש בסך הכל 2 מקומות ו 2 שאלות, כדי לאפשר משחק קליל קצר וכיף – בסוף המשחק תועברו למסך בו תגלו אם ניצחתם/הפסדתם או שיצא תיקו. במסך זה אפשר גם לראות את כמות הנקודות והמטבעות שהרווחתם במשחק.

מסך פתיחה



מסך זה הוא המסך הראשון שהמשתמש רואה כאשר הוא נכנס לאפליקציה, לחיצה על SIGN IN תעביר אותו להתחברות למשתמש. כמו כן, יש מעבר לתפריט אודות (מצד ימין למעלה).



בתפריט זה, (שמופיע כמעט בכל האפליקציה) יש שתי אפשרויות, באפשרות הראשונה אפשר לעבור למסך שמסביר על הפרויקט ועל כללי המשחק, ובשנייה אפשר לעבור למסך שנותן מידע על המתכנת (עליו).

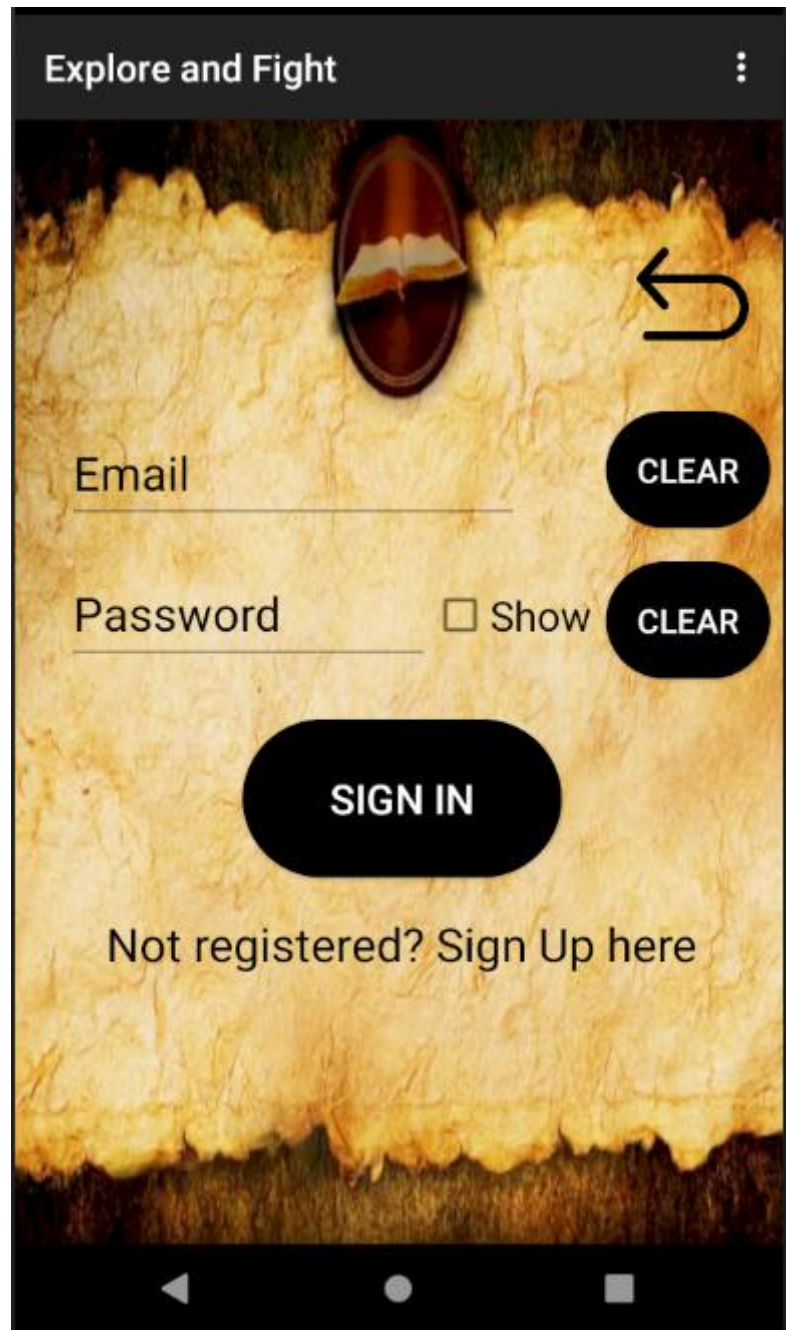


מסך זה (שניתן לגשת אליו רק מן התפריט אודות) מביא רקע על הפרויקט ועל המשחק, לחיצה על הכפתור GO BACK תחזיר את המשתמש אל המסך ממנו הוא נכנס לתפריט.



מסך זה (שניתן לגשת אליו רק מן התפריט אודות) מביא מידע על המתכנת (עליי) לחיצה על הכפתור GO BACK תחזיר את המשתמש אל המסך ממנו הוא נכנס לתפריט.

מסך התחברות



במסך זה, המשתמש יכול להתחבר אל החשבון שלו, לחיצה על הכפתור של החץ בצד ימין למעלה, תחזיר אותו למסך הפתיחה. הצ'ק בוקס ליד הסיסמא, מציג ומחביא את הסיסמא לחלופין לפי בחירה.

2 כפתורי clear - מטרתם היא לנקות את שדה קלט הנתונים של האימייל והסיסמא לפי המיקום.

כפתור ה SIGN IN יוביל לניסיון התחברות לחשבון, אם הפרטים תקינים המשתמש יועבר אל מסך הבית, ותוצג הודעת התחברות בהצלחה ואם לא תוצג הודעת שגיאה. בנוסף, בעת לחיצה על הטקסט למטה המשתמש יועבר לדף ההרשמה.

מסך הרשמה

4:25

Explore and Fight

Email

CLEAR

Password ☐ Show

CLEAR

Username

CLEAR

SIGN UP

Already registered? Sign In here

במסך זה, המשתמש יכול ליצור חשבון משלו, לחיצה על הכפתור של החץ בצד ימין למעלה, תחזיר אותו למסך הפתיחה. הציק בוקס ליד הסיסמא, מציג ומחביא את הסיסמא לחלופין לפי בחירה.

3כפתורי clear - מטרתם היא לנקות את שדה קלט הנתונים של האימייל הסיסמא ושם המשתמש לפי המיקום. כפתור ה SIGN UP יוביל לניסיון הרשמה, אם הפרטים תקינים המשתמש יועבר אל מסך הבית, ותוצג הודעת התחברות בהצלחה ואם לא תוצג הודעת שגיאה. בנוסף, בעת לחיצה על הטקסט למטה המשתמש יועבר לדף ההתחברות.

מסך הבית



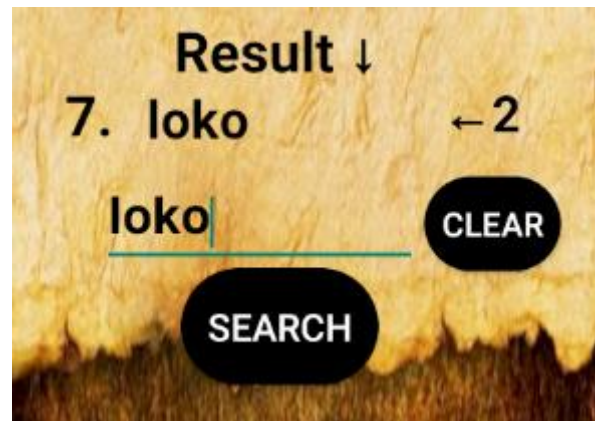
מסך זה הוא מסך הבית, בו מוצגת הודעה שמציגה את שם המשתמש, בתמונה כאן הוא מופיע בצבע זהב עם תג VIP וזאת כיוון שהוא משתמש וי איי פי. במסך הבית יישנם 5 כפתורים שונים:

1. הגדרות – במסך זה יופיעו הגדרות האפליקציה (הסבר בהרחבה במסך הגדרות).
 2. חנות – במסך זה אפשר לקנות עזרה במשחק של 50/50 וגם משתמש VIP.
 3. שחק עכשיו – במסך זה אפשר ליצור משחק חדש או להיכנס למשחק קיים (הסבר בהרחבה במסך שחק עכשיו).
 4. לוח המובילים – במסך זה אפשר לראות את טבלת המובילים (הסבר בהרחבה במסך לוח המובילים).
 5. כפתור LOGOUT – מעביר אותך למסך של "אישור ניתוק". (הסבר בהרחבה במסך אישור ניתוק).
- *לחיצה על התפריט בצד ימין למעלה תוביל לתפריט אודות.

לוח המובילים



במסך זה אפשר לראות את 5 המשתמשים הכי טובים (עם הכי הרבה נקודות), ליד כל שם מופיע ה-Score שהוא בעצם מספר הנקודות, ובאיזה מקום המשתמש נמצא (כולל סימון של YOU במקום בו אתה נמצא). השמות של משתמשי ה-VIP מופיעים בצבע זהב, בעוד שמשתמשים רגילים בצבע שחור רגיל. בנוסף, בתחתית המסך יש אפשרות לחפש משתמש לפי השם שלו, והדבר מציג את הפרטים הבאים:



(את המקום של המשתמש ואת מספר הנקודות שלו)
 בנוסף, אם אתה לא נמצא בטופ 5 משתמשים, אתה תקבל מידע לגבי איפה אתה נמצא ביחס אליהם באופן
 הבא:



אפשר לראות שהמשתמש שמחובר נמצא במקום ה-6 עם 22 נקודות.
 על מנת לחזור אחורה, אפשר ללחוץ על הכפתור חזור בצד שמאל למעלה (שמחזיר אותך למסך הבית).

*לחיצה על התפריט בצד ימין למעלה תוביל לתפריט אודות.

שחק עכשיו

במסך זה אפשר ליצור משחק חדש על ידי לחיצה על CREATE ROOM או להיכנס למשחק קיים על ידי לחיצה על JOIN ROOM.

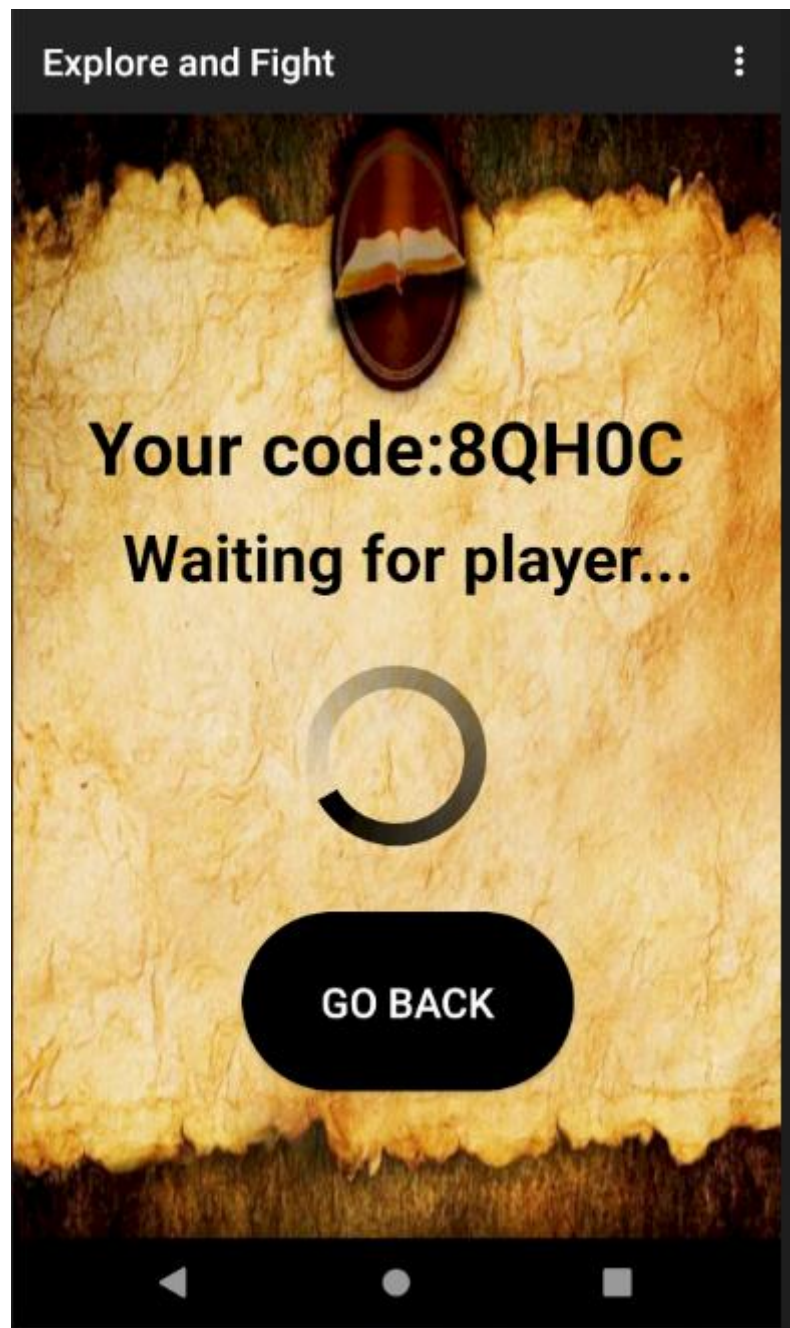
כפתור CREATE ROOM מעביר את המשתמש למסך "צור משחק", בו צריך לחכות עד שמשתמש שני ייכנס.

(הסבר בהרחבה במסך צור משחק)

כפתור JOIN ROOM מעביר את המשתמש למסך "היכנס למשחק" בו אפשר לרשום את הקוד ולהיכנס למשחק שנוצר (הסבר בהרחבה במסך היכנס למשחק)

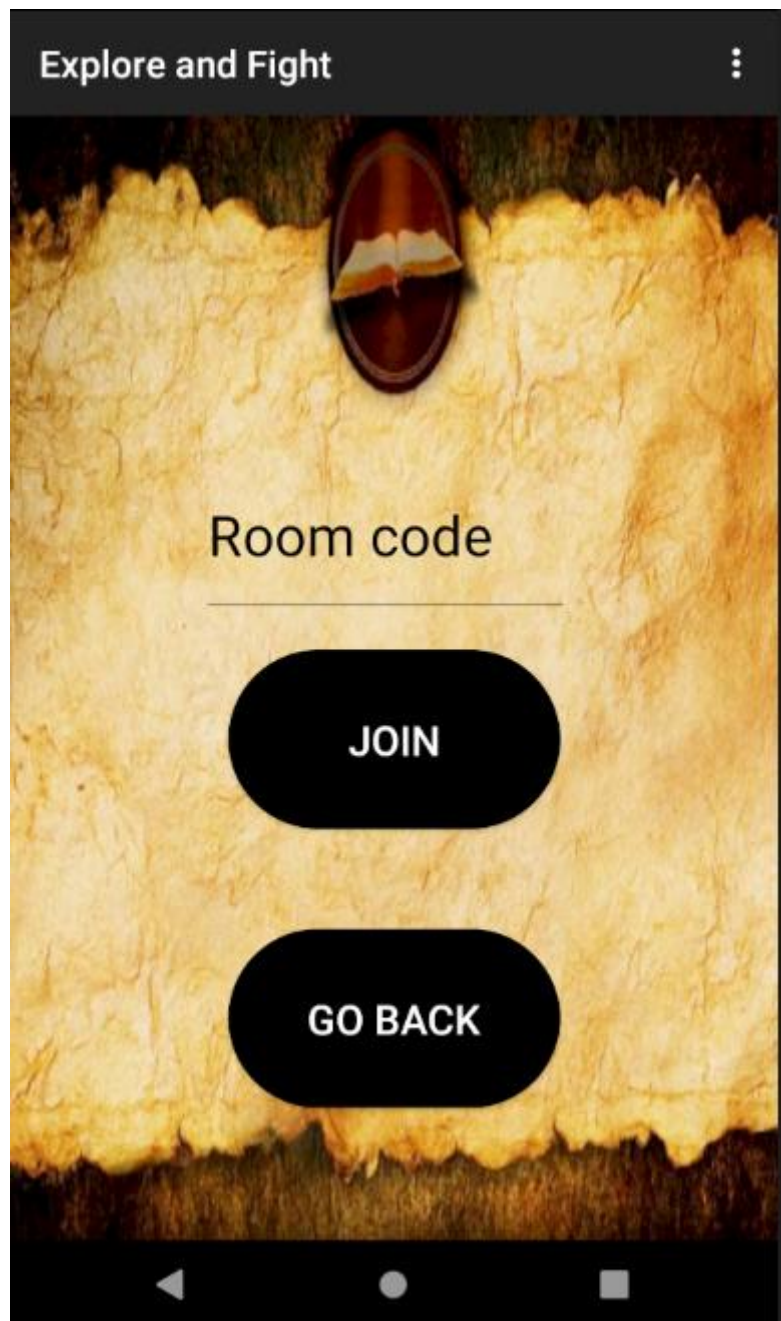
כמו כן, כפתור GO BACK יחזיר את המשתמש למסך הבית.

*לחיצה על התפריט בצד ימין למעלה תוביל לתפריט אודות.

צור משחק

במסך זה, יוצג קוד של 5 ספרות ואותיות באנגלית, אשר מתקבל באופן אקראי וכך משתנה בכל פעם. על המסך יוצג PROGRESS BAR אשר ימשיך להסתובב עד אשר משתמש אחר ייכנס למשחק. כפתור ה GO BACK יחזיר את המשתמש אל מסך הבית.
*לחיצה על התפריט בצד ימין למעלה תוביל לתפריט אודות.

היכנס למשחק



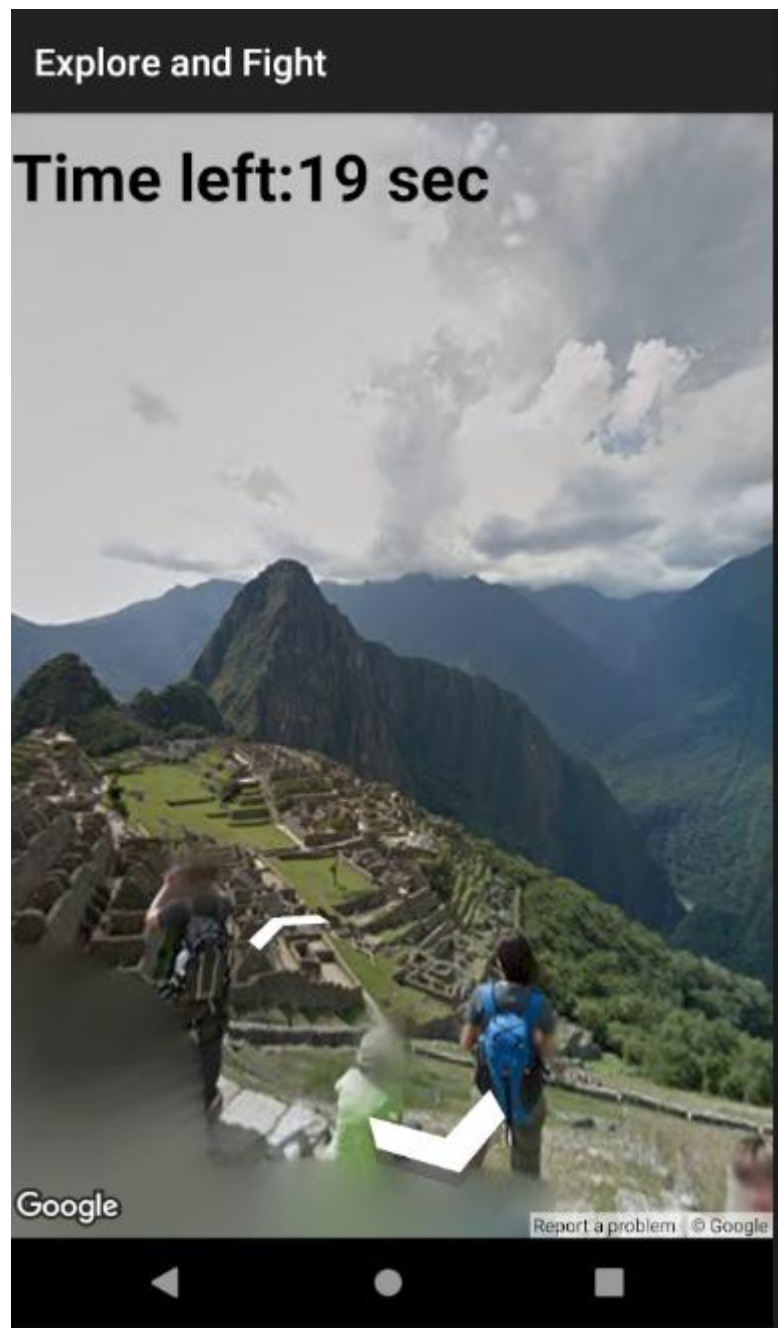
במסך זה, המשתמש יכול להיכנס אל משחק קיים על ידי הקוד בן 5 הספרות והאותיות, אשר המשתמש השני שיצר את החדר יכול לשתף עמו.

כאשר הקוד מוזן ופרטיו נכונים, שני השחקנים (גם היוצר וגם מי שנכנס) יועברו אל מסך המשחק.

כפתור ה GO BACK יחזיר את המשתמש אל מסך הבית.

*לחיצה על התפריט בצד ימין למעלה תוביל לתפריט אודות.

מסך המשחק



במסך זה אפשר לראות מקום אקראי בעולם דרך GOOGLE STREET VIEW כמו כן, אפשר ללכת ולהסתובב במקום. מצד שמאל למעלה מופיע טיימר של 20 שניות, ברגע שהוא נגמר המשתמש מועבר אל מסך השאלה.

מסך השאלה



במסך זה המשתמש יצטרך לבחור תשובה אחת מתוך 4 תשובות אפשריות, רק תשובה אחת נכונה. השאלה היא באיזה מקום הוא היה נמצא במסך המשחק – המשתמש יכול להיעזר בעזרה של 50/50 בצד ימין למעלה – אם ברשותו עזרה כזו. בסוף הזמן שבטיימר, אי אפשר ללחוץ על הכפתורים יותר, התשובה הנכונה תופיע בירוק וכל התשובות השגויות יופיעו באדום. לאחר מכן המשתמש יועבר אל מסך המשחק בפעם נוספת. ואז שייגמר הזמן במסך המשחק הוא יחזור למסך השאלה – כלומר 2 מקומות ו2 שאלות בסך הכל. לאחר השאלה האחרונה יועבר המשתמש אל מסך "מי ניצח" – בו ייקבע תיקו ניצחון והפסד בהתאם לניקוד.

מסך הגדרות



במסך זה אפשר לשנות את ההגדרות כמו להדליק ולכבות את המוזיקה, להדליק ולכבות את מצב "נא לא להפריע" (כאשר פועל האפליקציה תשלח הודעה שאומרת שאתה עסוק כאשר מתקבלת הודעה) כמו כן, אפשר לעבור למסך "שנה את השם" על ידי לחיצה על הטקסט / כפתור מתאים. הכפתור בצד ימין למטה, יחזיר את המשתמש למסך הבית. *לחיצה על התפריט בצד ימין למעלה תוביל לתפריט אודות.

מסך שנה שם

- במסך זה אפשר לערוך את שם המשתמש, ולשנות אותו במידה והוא לא תפוס. לחיצה על כפתור ה CONFIRM תוביל לשינוי שם המשתמש בדאטה בייס במידה והשם תקין ופנוי לשימוש, ותחזיר את המשתמש אל מסך ההגדרות.
- לחיצה על כפתור חזור בצד ימין למטה תחזיר את המשתמש למסך ההגדרות.
- במידה ונעשה שינוי בשם שלא נשמר (המשתמש לדוגמה שכח ללחוץ על כפתור ה CONFIRM) אז ייפתח דיאלוג בוקס:

Are you sure?

Do you want to save changes?

NO

YES

לחיצה על כפתור ה YES תשמור את השם ששונה במידה והוא תקין ופנוי, ותחזיר את המשתמש אל מסך ההגדרות.
לחיצה על כפתור ה NO תחזיר את המשתמש אל מסך ההגדרות מבלי לשמור את השינויים.

מסך חנות

במסך זה המשתמש יכול לראות את כמות המטבעות שלו, ולקנות VIP או 50/50, לחיצה על תמונת ה VIP תוביל את המשתמש אל מסך קניית ה VIP, כמו כן לחיצה על תמונת ה 50/50 תוביל את המשתמש אל מסך קניית 50/50. כפתור ה GO BACK יחזיר את המשתמש אל מסך הבית.

מסך קניית VIP



במסך זה המשתמש יכול לקנות VIP במידה ויש לו מספיק מטבעות, לחיצה על כפתור החזור תחזיר את המשתמש אל מסך החנות.

מסך קניית 50/50

במסך זה המשתמש יכול לקנות 50/50 במידה ויש לו מספיק מטבעות, לחיצה על כפתור החזרה תחזיר את המשתמש אל מסך החנות.



במסך זה אפשר לראות את תוצאת המשחק, במקרה הזה – תיקו. לחזרה למסך הבית אפשר לחוץ על כפתור GO BACK.



במסך זה אפשר לראות את תוצאת המשחק , במקרה הזה – הפסד . לחזרה למסך הבית אפשר לחוץ על כפתור GO BACK.

מסך ניצחון

במסך זה אפשר לראות את תוצאת המשחק , במקרה הזה – ניצחון . לחזרה למסך הבית אפשר לחוץ על כפתור GO BACK.

הרשאות

```
1 android.permission.RECEIVE_SMS
2 android.permission.SEND_SMS
3 android.permission.INTERNET
```

הסבר –

1. הרשאה לקבל הודעות SMS .
2. הרשאה לשלוח הודעות SMS.
3. הרשאה לשימוש באינטרנט על ידי האפליקציה.

דרישות מיוחדות ומגבלות

מגבלה : כדי להתחבר למשתמש ולשחק - דרוש חיבור לאינטרנט .
צריך שיהיה רכיב SIM בתוך המכשיר בכדי לשלוח ולקבל הודעות SMS.

גרסת Android מינימלית

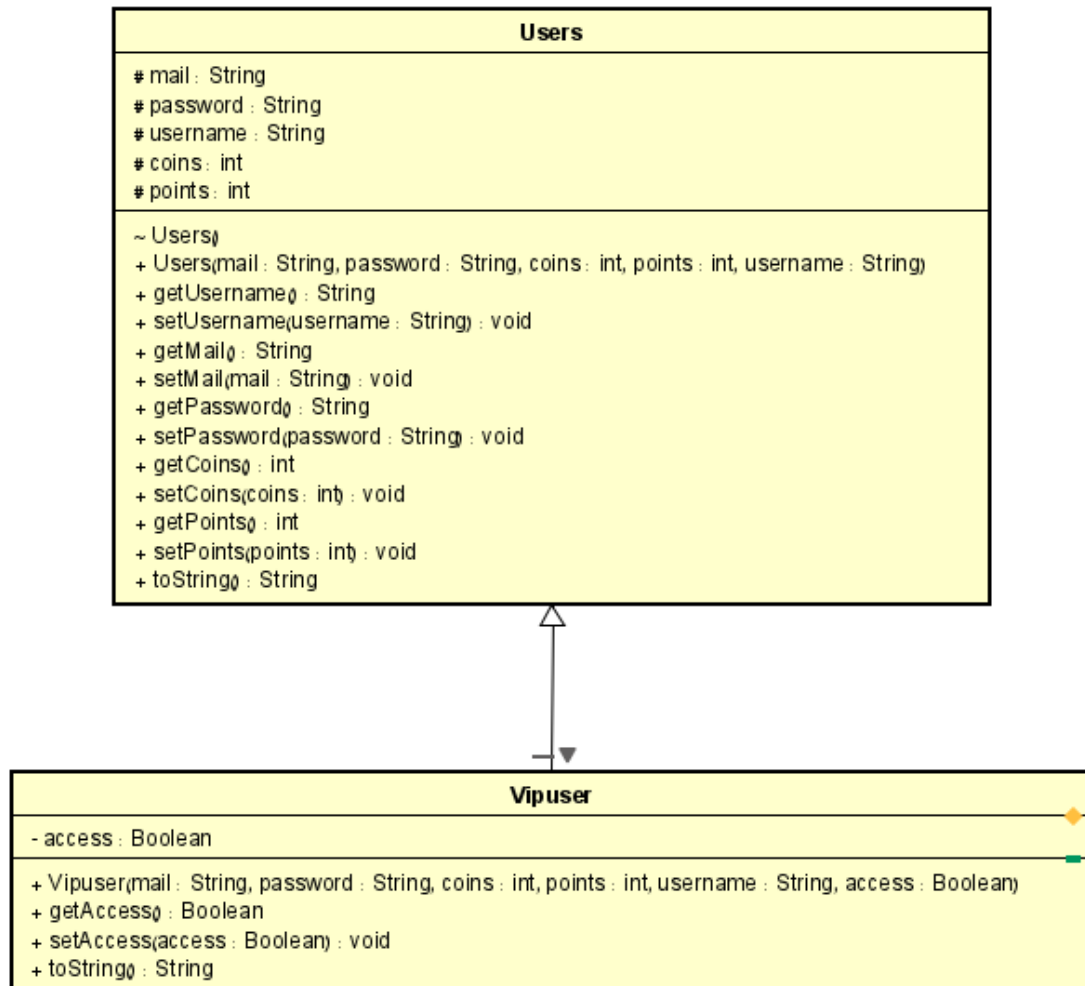
Android 4.4.2 גרסת API 16

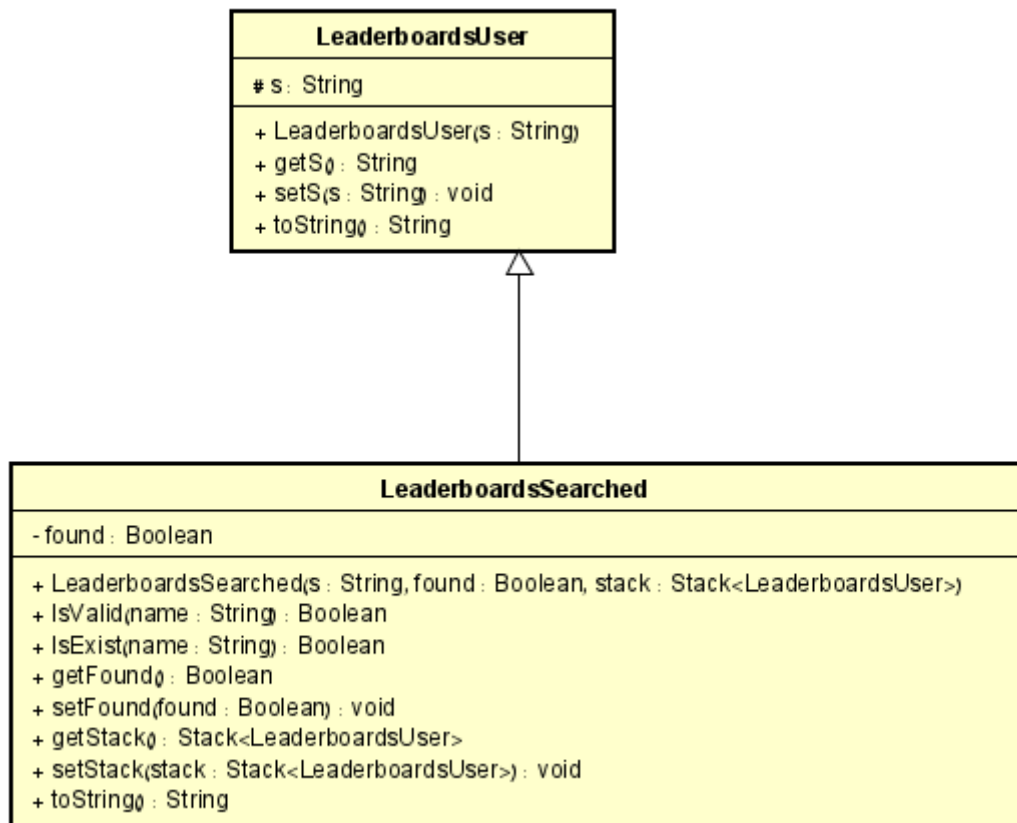
מכשירים עליהם נבדקה המערכת

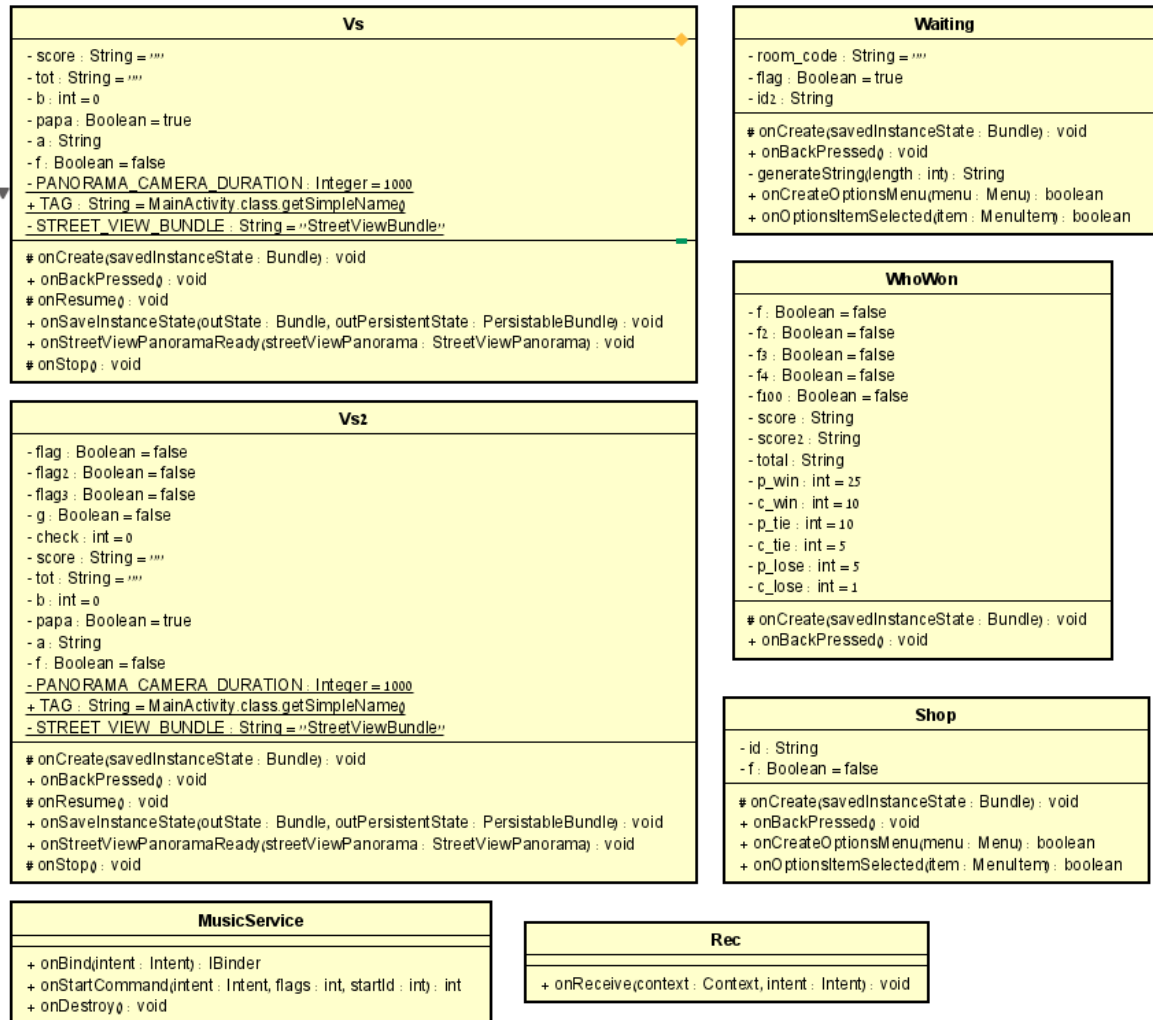
Nexus 5x API 30

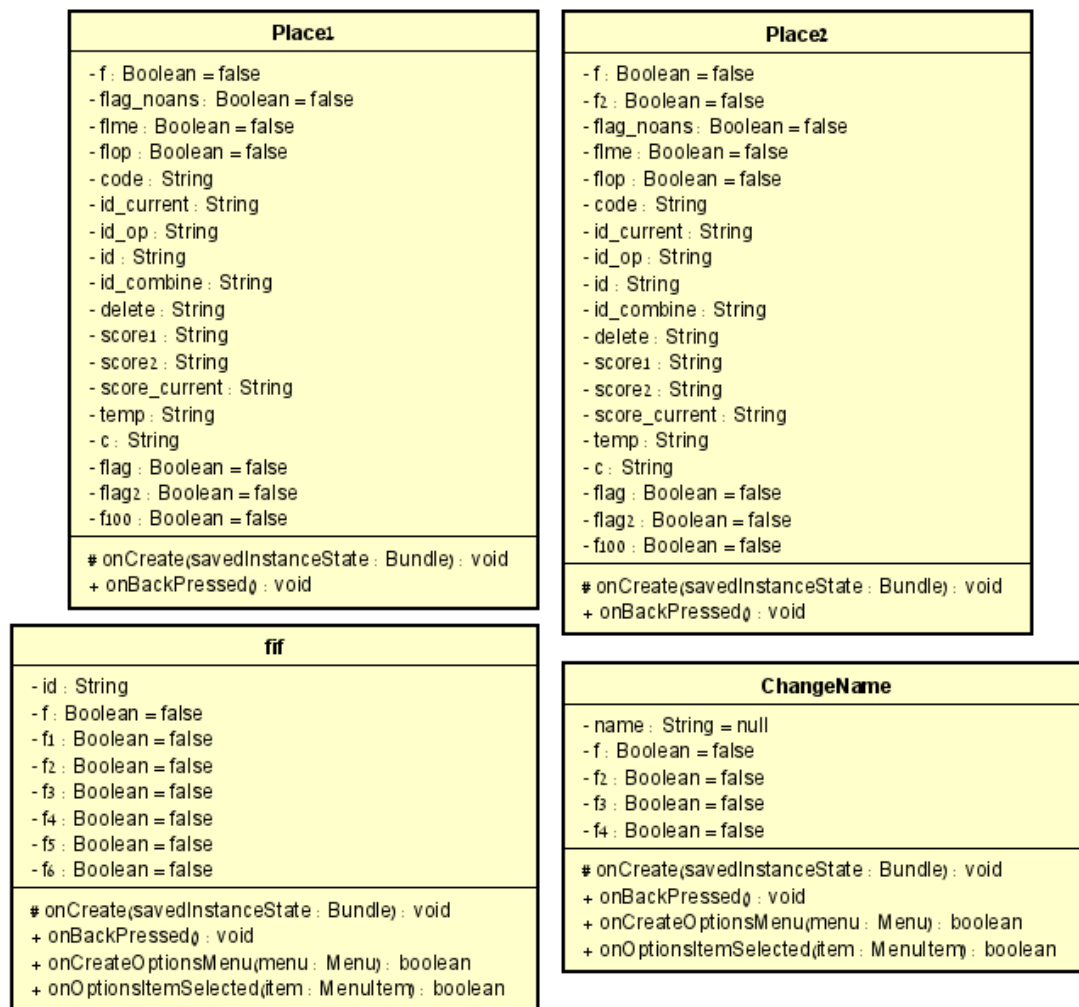
Huawei p30 pro API 30

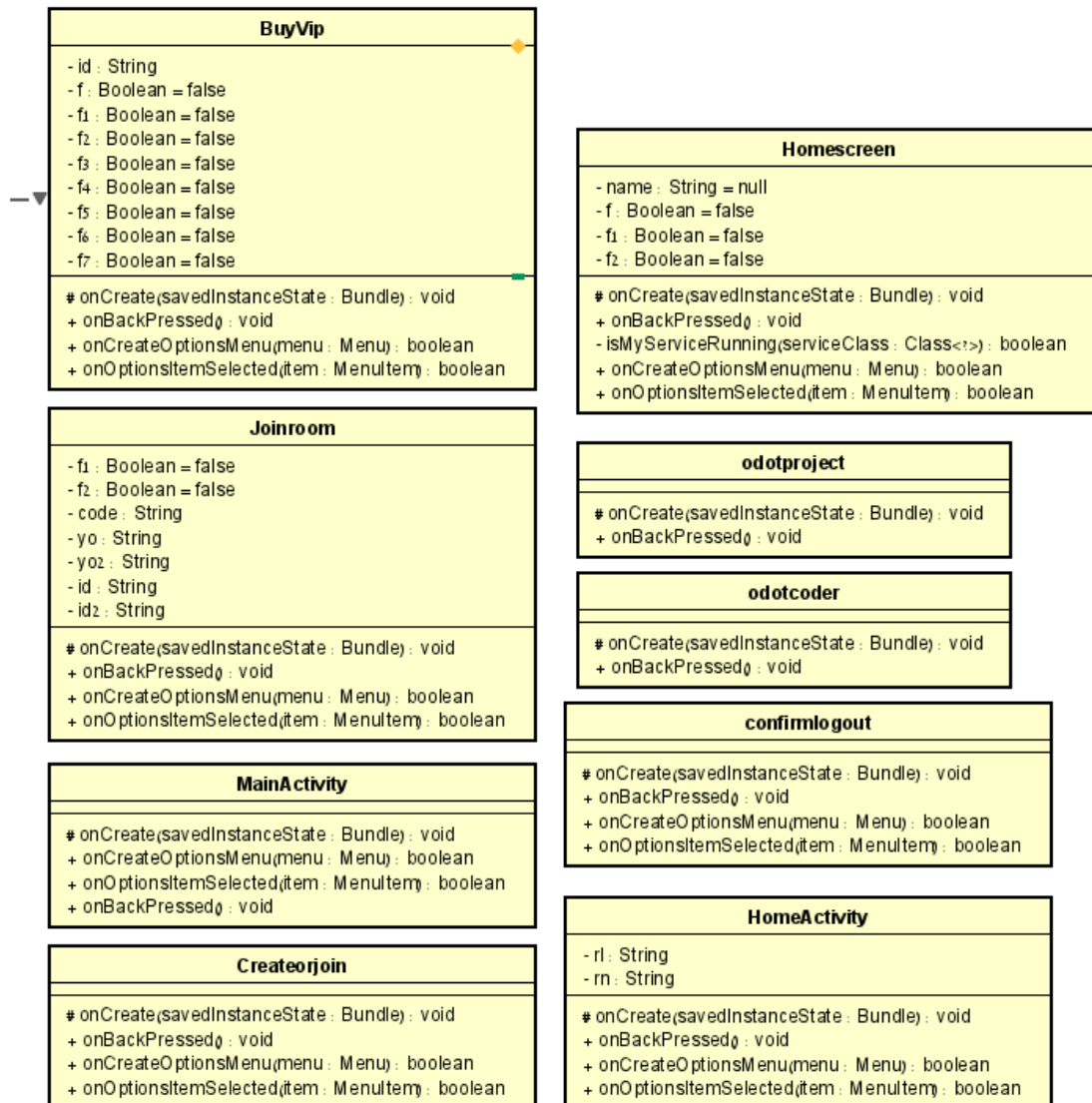
UML











Leaderboards
<ul style="list-style-type: none"> - s : Stack<String> = new Stack<String>() - s2 : Stack<Integer> = new Stack<Integer>() -scopy : Stack<String> -s2copy : Stack<Integer> - stemp : Stack<String> - s2temp : Stack<Integer> - f : Boolean = false
<pre># onCreate(savedInstanceState : Bundle) : void + cloneIntegerStack(s : Stack<Integer>) : Stack<Integer> + cloneStringStack(s : Stack<String>) : Stack<String> + cloneUsersStack(s : Stack<Users>) : Stack<Users> + cloneLeaderboardsUserStack(s : Stack<LeaderboardsUser>) : Stack<LeaderboardsUser> + onBackPressed() : void + onCreateOptionsMenu(menu : Menu) : boolean + onOptionsItemSelected(item : MenuItem) : boolean</pre>



LoginActivity
<ul style="list-style-type: none"> - f : Boolean = false
<pre># onCreate(savedInstanceState : Bundle) : void + onBackPressed() : void + onCreateOptionsMenu(menu : Menu) : boolean + onOptionsItemSelected(item : MenuItem) : boolean</pre>

Settings
<ul style="list-style-type: none"> - id : String - f1 : Boolean = false - f2 : Boolean = false - d : Boolean = true - d2 : Boolean = true
<pre># onCreate(savedInstanceState : Bundle) : void + onBackPressed() : void + isMyServiceRunning(serviceClass : Class<?>) : boolean + onCreateOptionsMenu(menu : Menu) : boolean + onOptionsItemSelected(item : MenuItem) : boolean</pre>

מדריך למתכנת

– מסך פתיחה MAIN ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
SignIn	Button	לעבור למסך התחברות

פעולות :

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מעבירה את המשתמש אל מסך ההתחברות בעת לחיצה -

```
SignIn.setOnClickListener()
```


– LOGIN ACTIVITY מסך התחברות

טבלת משתנים :

שם משתנה	סוג	מטרה
reg	TextView	מעבר למסך הרשמה
login_mail	EditText	הכנסת האימייל על ידי המשתמש
login_pass	EditText	הכנסת הסיסמא על ידי המשתמש
auth2	FirebaseAuth	התחברות לפיירבייס
stateListener	FirebaseAuth.AuthStateListener	בדיקה אם המשתמש מחובר
checkBox	CheckBox	גילוי / הסתרת הסיסמא
imageView	ImageView	חץ בשביל לחזור אחורה
btncsignin	Button	כפתור בשביל להתחבר לחשבון

פעולות :

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מופעלת בלחיצה על הכפתור ומחברת את המשתמש אל החשבון -

```
btncsignin.setOnClickListener()
```

פעולה אשר בודקת אם הצ'ק בוקס מסומן או לא -

```
checkBox.setOnCheckedChangeListener()
```

– מסך הרשמה HOME ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
sign_mail	EditText	הכנסת האימייל על ידי המשתמש
sign_pass	EditText	הכנסת הסיסמא על ידי המשתמש
sign_user	EditText	הכנסת שם המשתמש
btnSign	Button	כפתור בשביל להירשם לחשבון
btnclearMail	Button	כפתור בשביל לנקות את האימייל
btnclearPass	Button	כפתור בשביל לנקות את הסיסמא
btnclearUser	Button	כפתור בשביל לנקות את השם
SignIn	TextView	מעבר למסך התחברות
auth	FirebaseAuth	התחברות לפיירבייס
rl	String	לכאן נשמר השם המשתמש
rn	String	השם עובר שינוי ונשמר כאן
checkBox	CheckBox	גילוי / הסתרת הסיסמא
imageView	ImageView	חץ בשביל לחזור אחורה

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מעבירה את המשתמש אל המסך הקודם בו הוא היה בעת לחיצה -

```
imageView.setOnClickListener()
```

פעולה אשר בודקת אם הצ'ק בוקס מסומן או לא -

```
checkBox.setOnCheckedChangeListener()
```

פעולה אשר רושמת את המשתמש לחשבון בעת לחיצה -

```
btnSign.setOnClickListener()
```

פעולה אשר מנקה את האימייל בעת לחיצה -

```
btnclearMail.setOnClickListener()
```

פעולה אשר מנקה את שם המשתמש בעת לחיצה -

```
btnclearUser.setOnClickListener()
```

פעולה אשר מעבירה את המשתמש אל מסך ההתחברות -

```
SignIn.setOnClickListener()
```

– מסך אודות הפרויקט ODOTPROJECT ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
btn	Button	לחזור אל המסך הקודם

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מחזירה את המשתמש אל המסך הקודם -

```
btn.setOnClickListener()
```

– מסך אודות המתכנת ODOTCODER ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
button	Button	לחזור אל המסך הקודם

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מחזירה את המשתמש אל המסך הקודם -

```
button.setOnClickListener()
```

– מסך הבית HOMESCREEN ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
logout	Button	לנתק את המשתמש
vs	ImageView	מעביר את המשתמש אל צור/היכנס למשחק
ld	ImageView	מעביר את המשתמש ללוח המובילים
shop	ImageView	מעביר את המשתמש למסך החנות
vop	ImageView	מציג תג וי איי פי
settings	ImageView	מעביר את המשתמש אל ההגדרות
auth	FirebaseAuth	התחברות לפירבייס
ref	DatabaseReference	קישור לפירבייס לבדוק אם VIP
tx	TextView	מציג את שם המשתמש
name	String	מאחסן את שם המשתמש
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f1	Boolean	לבדוק אם נכנסו לפעולה פעם 1

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מעבירה את המשתמש אל מסך ההתחברות בעת לחיצה -

```
SignIn.setOnClickListener()
```

פעולה אשר בודקת אם הסרוויס שמתקבל הוא רץ או לא -

```
private boolean isMyServiceRunning(Class<?> serviceClass)
```

פעולה אשר מעבירה את המשתמש למסך צור/היכנס למשחק בעת לחיצה -

```
vs.setOnClickListener()
```

פעולה אשר מעבירה את המשתמש למסך לוח המובילים בעת לחיצה -

```
ld.setOnClickListener()
```

פעולה אשר מעבירה את המשתמש למסך החנות בעת לחיצה -

```
shop.setOnClickListener()
```

פעולה אשר מעבירה את המשתמש למסך ההגדרות בעת לחיצה -

```
settings.setOnClickListener()
```

– מסך צור/היכנס CREATEORJOIN ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
create	Button	מעביר את המשתמש ליצור משחק
join	Button	מעביר את המשתמש להיכנס למשחק
goback	Button	מחזיר את המשתמש למסך הבית

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מעבירה את המשתמש ליצור משחק בעת לחיצה -

```
create.setOnClickListener()
```

פעולה אשר מעבירה את המשתמש להיכנס למשחק בעת לחיצה -

```
join.setOnClickListener()
```

פעולה אשר מחזירה את המשתמש למסך הבית בעת לחיצה -

```
goback.setOnClickListener()
```


– מסך היכנס למשחק JOINROOM ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
f1	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f2	Boolean	לבדוק אם נכנסו לפעולה פעם 1
joincode	Button	מכניס את המשתמש אל הקוד
goback	Button	מחזיר את המשתמש מסך אחורה
code	String	שומר את הקוד שהמשתמש כותב
yo	String	שומר את הערך של מקום 1
yo2	String	שומר את הערך של מקום 2
editText	EditText	כאן המשתמש כותב את הקוד
auth	FirebaseAuth	התחברות לפייירבייס
id	String	אחסון ה ID של המשתמש
tx	TextView	טקסט הוראה
id2	String	שומר את ה ID של המשתמש השני

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מכניסה את המשתמש אל הקוד שהוזן בעת לחיצה -

```
joincode.setOnClickListener()
```

פעולה אשר מחזירה את המשתמש מסך אחורה בעת לחיצה -

```
goback.setOnClickListener()
```

– מסך צור משחק WAITING ACTIVITY

שם משתנה	סוג	מטרה
progressBar	ProgressBar	אנימציה של טעינה - המתנה
imageView	ImageView	מראה סימן של וי אם שחקן שני התחבר
cancel	Button	מבטל את המשחק וחוזר למסך הבית
change	TextView	כותב מתי שנמצא שחקן
code	TextView	מראה על המסך את הקוד
room_code	String	מאחסן את הקוד שנוצר באופן רנדומלי
flag	Boolean	לבדוק אם נכנסו לפעולה פעם 1
ref	DatabaseReference	קישור לפירבייס
auth	FirebaseAuth	התחברות לפירבייס
id2	String	מאחסן את ID של המשתמש השני

טבלת משתנים:

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר יוצרת קוד באורך 5 רנדומלי למשחק -

```
private String generateString(int length)
```

– מסך חנות SHOP ACTIVITY

טבלת משתנים :

שם משתנה	סוג	מטרה
vip	ImageView	מעביר את המשתמש לקניית VIP
fif	ImageView	מעביר את המשתמש לקניית 50/50
coins	TextView	מציג כמה מטבעות יש למשתמש
goback	Button	מחזיר את המשתמש למסך הבית
auth	FirebaseAuth	התחברות לפירבייס
id	String	שומר את ה ID של המשתמש
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1

פעולות :

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה המחזירה את המשתמש למסך הבית -

```
goback.setOnClickListener()
```

– מסך קניית VIP BUYVIP ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
Buy	Button	קונה VIP למשתמש
goback	ImageView	מחזירה את המשתמש לחנות
auth	FirebaseAuth	התחברות לפירבייס
id	String	שומר את ה ID של המשתמש
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f1	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f2	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f3	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f4	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f5	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f6	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f7	Boolean	לבדוק אם נכנסו לפעולה פעם 1

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר קונה למשתמש VIP -

```
Buy.setOnClickListener()
```

מחזירה את המשתמש אל מסך החנות בעת לחיצה -

```
goback.setOnClickListener()
```

מסך קניית FIF ACTIVITY 50/50 –

טבלת משתנים:

שם משתנה	סוג	מטרה
buy50	Button	קונה למשתמש 50/50
goback	ImageView	מחזירה את המשתמש לחנות
auth	FirebaseAuth	התחברות לפיירבייס
tex	TextView	מציג כמה 50/50 יש למשתמש
id	String	מאחסן את ה ID של המשתמש
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f1	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f6	Boolean	לבדוק אם נכנסו לפעולה פעם 1

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר קונה למשתמש את העזרה -

```
buy50.setOnClickListener()
```

מחזירה את המשתמש אל מסך החנות בעת לחיצה -

```
goback.setOnClickListener()
```


– מסך הגדרות SETTINGS ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
change_name	Button	כפתור שמעביר למסך שנה שם
music	Switch	סוויץ' שבדק אם להפעיל או לכבות מוזיקה
busy	Switch	סוויץ' שבדק אם להפעיל או לכבות נא לא להפריע
goback	ImageView	מעביר את המשתמש אל מסך הבית
auth	FirebaseAuth	התחברות לפיירבייס
id	String	מאחסן את ה ID של המשתמש
f1	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f2	Boolean	לבדוק אם נכנסו לפעולה פעם 1
d	Boolean	לבדוק אם נכנסו לפעולה פעם 1
d2	Boolean	לבדוק אם נכנסו לפעולה פעם 1

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה-

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר בודקת אם הסרוויס שהתקבל רץ או לא -

```
private boolean isMyServiceRunning(Class<?> serviceClass)
```

פעולה אשר מעבירה את המשתמש למסך שנה שם -

```
change_name.setOnClickListener()
```

פעולה אשר בודקת אם הסוויץ' של המוזיקה מסומן או לא -

```
music.setOnCheckedChangeListener()
```

פעולה אשר בודקת אם הסוויץ' של העסוק מסומן או לא -

```
busy.setOnCheckedChangeListener()
```

פעולה אשר מחזירה את המשתמש למסך הבית

- ()goback.setOnClickListener

– מסך שנה שם CHANGENAME ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
auth	FirebaseAuth	התחברות לפיירבייס
goback	ImageView	מחזיר את המשתמש להגדרות
confirm	Button	מאשר ומשנה את השם
edit_name	EditText	הצגה ושינוי של השם
name	String	מאחסן את השם
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f2	Boolean	לבדוק אם נכנסו לפעולה פעם 1

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מאשרת את השם ומשנה אותו בפיירבייס -

```
confirm.setOnClickListener()
```

פעולה אשר מחזירה את המשתמש אל מסך ההגדרות -

```
goback.setOnClickListener()
```

– מסך התנתקות CONFIRMLOGOUT ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
confirm_logout	Button	מנתק את המשתמש מהפיירבייס
cancel	Button	מבטל ומחזיר את המשתמש למסך הבית
auth	FirebaseAuth	התחברות לפרייבייס

פעולות:

פעולה היוצרת את המסך, מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מנתקת את המשתמש מהפיירבייס בעת לחיצה -

```
confirm_logout.setOnClickListener()
```

מבטל את ההתנתקות, ומחזיר את המשתמש אל מסך הבית -

```
cancel.setOnClickListener()
```

– מסך התנתקות CONFIRMLOGOUT ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
tex1	TextView	שם של מקום 1
tex2	TextView	שם של מקום 2
tex3	TextView	שם של מקום 3
tex4	TextView	שם של מקום 4
tex5	TextView	שם של מקום 5
tex6	TextView	מציג נקודות על מקום 1
tex7	TextView	מציג נקודות על מקום 2
tex8	TextView	מציג נקודות על מקום 3
tex9	TextView	מציג נקודות על מקום 4
tex10	TextView	מציג נקודות על מקום 5
tex11	TextView	מציג מידע על המשתמש
tex12	TextView	מציג מידע על המשתמש
tex13	TextView	מציג מידע על המשתמש
tex14	TextView	מציג מידע על המשתמש
y1	TextView	אם אתה מקום 1
y2	TextView	אם אתה מקום 2
y3	TextView	אם אתה מקום 3
y4	TextView	אם אתה מקום 4
y5	TextView	אם אתה מקום 5
search_text	EditText	שדה חיפוש משתמש
search	Button	מחפש משתמש ומציג מידע
clear	Button	מנקה את שדה החיפוש
goback	ImageView	מחזיר את המשתמש למסך הבית
auth	FirebaseAuth	התחברות לפיירבייס
s	Stack<String>	מחסנית כל שמות המשתמשים
s2	Stack<Integer>	מחסנית של כל הנקודות
usersStack	Stack<Users>	אחסון העצם בתוך המחסנית
leaderboardsUsers	Stack<LeaderboardsUser>	אחסון העצם בתוך המחסנית
scopy	Stack<String>	שכפול המחסנית של s
s2copy	Stack<Integer>	שכפול המחסנית של s2

usersStack של המחסנית של שכפול	Stack<Users>	usersStackcopy
מחסנית זמנית	Stack<String>	stemp
מחסנית זמנית	Stack<Integer>	s2temp
מחסנית זמנית	Stack<Users>	usersStacktemp
לבדוק אם נכנסו לפעולה פעם 1	Boolean	f

פעולות :

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר מנקה את שדה חיפוש המשתמש בעת לחיצה –

```
clear.setOnClickListener()
```

פעולה אשר מחפשת משתמש בדאטה בייס לפי השם שהוזן בעת לחיצה –

```
search.setOnClickListener()
```

פעולה אשר מחזירה את המשתמש אל מסך הבית בעת לחיצה -

```
goback.setOnClickListener()
```


מסך מי ניצח? – WHOWON ACTIVITY

שם משתנה	סוג	מטרה
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f2	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f3	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f4	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f100	Boolean	לבדוק אם נכנסו לפעולה פעם 1
tex1	TextView	מציג את ה SCORE
tex2	TextView	מציג את ה COINS
whowon	TextView	כותב לך אם ניצחת/הפסדת או תיקו
win	ImageView	מראה לך תמונה שניצחת
lose	ImageView	מראה לך תמונה שהפסדת
draw	ImageView	מראה לך תמונה שיש תיקו
bar	ProgressBar	טוען עד שיתקבלו הנתונים
back	Button	כפתור שמחזיר למסך הבית
auth	FirebaseAuth	התחברות לפיירבייס
score	String	מאחסן את ה SCORE של השחקן השני
score2	String	מאחסן את ה SCORE
total	String	מוסיף נקודות לפיירבייס
p_win	int	כמות נקודות להוספה - ניצחון
c_win	int	כמות מטבעות להוספה - ניצחון
p_tie	int	כמות נקודות להוספה - תיקו
c_tie	int	כמות מטבעות להוספה - תיקו
p_lose	int	כמות נקודות להוספה - הפסד
c_lose	int	כמות מטבעות להוספה - הפסד

טבלת משתנים:

פעולות :

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה שמחזירה את המשתמש אל מסך הבית בעת לחיצה -

```
back.setOnClickListener()
```

– מסך שאלה 1 PLACE1 ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
fif	ImageView	מאפשר להשתמש ב/ 5050
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
flag_noans	Boolean	בודק אם אין תשובה
flme	Boolean	לבדוק אם נכנסו לפעולה פעם 1
flop	Boolean	לבדוק אם נכנסו לפעולה פעם 1
place1	Button	בוחר באפשרות 1
place2	Button	בוחר באפשרות 2
place3	Button	בוחר באפשרות 3
place4	Button	בוחר באפשרות 4
countDownTimer	CountDownTimer	טיימר עד שהזמן נגמר
tex	TextView	מציג את השאלה
count	TextView	מציג את הזמן שנשאר
code	String	מאחסן את הקוד של המשחק
id_current	String	מאחסן את הID של המשתמש הנוכחי
id_op	String	מאחסן את הID של המשתמש השני
id	String	מאחסן את הID של המשתמש
id_combine	String	מאחסן את הID של 2 המשתמשים ביחד

מאחסן את ה ID של 2 המשתמשים ביחד (בסדר הפוך) כדי למחוק	String	delete
מאחסן את הניקוד הראשון	String	score1
מאחסן את הניקוד השני	String	score2
מאחסן את הניקוד הנוכחי	String	score_current
ערך זמני של סטרינג	String	temp
שומר את בחירת המשתמש	String	c
התחברות לפייירבייס	FirebaseAuth	auth
לבדוק אם נכנסו לפעולה פעם 1	Boolean	flag
לבדוק אם נכנסו לפעולה פעם 1	Boolean	flag2
לבדוק אם נכנסו לפעולה פעם 1	Boolean	f100

פעולות :

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר בוחרת את אפשרות 1 בעת לחיצה -

```
place1.setOnClickListener()
```

פעולה אשר בוחרת את אפשרות 2 בעת לחיצה -

```
place2.setOnClickListener()
```

פעולה אשר בוחרת את אפשרות 3 בעת לחיצה -

```
place3.setOnClickListener()
```

פעולה אשר בוחרת את אפשרות 4 בעת לחיצה -

place4.setOnClickListener()

פעולה אשר משתמשת בעזרה של 5050 / בעת לחיצה -

fif.setOnClickListener()

מסך שאלה 2 – PLACE2 ACTIVITY

טבלת משתנים:

שם משתנה	סוג	מטרה
f	Boolean	לבדוק אם נכנסו לפעולה פעם 1
F2	Boolean	לבדוק אם נכנסו לפעולה פעם 1
flag_noans	Boolean	בודק אם אין תשובה
flme	Boolean	לבדוק אם נכנסו לפעולה פעם 1
flop	Boolean	לבדוק אם נכנסו לפעולה פעם 1
place1	Button	בוחר באפשרות 1
place2	Button	בוחר באפשרות 2
place3	Button	בוחר באפשרות 3
place4	Button	בוחר באפשרות 4
countDownTimer	CountDownTimer	טיימר עד שהזמן נגמר
tex	TextView	מציג את השאלה
count	TextView	מציג את הזמן שנשאר
code	String	מאחסן את הקוד של המשחק
id_current	String	מאחסן את הID של המשתמש הנוכחי
id_op	String	מאחסן את הID של המשתמש השני

id	String	מאחסן את ה ID של המשתמש
id_combine	String	מאחסן את ה ID של 2 המשתמשים ביחד
delete	String	מאחסן את ה ID של 2 המשתמשים ביחד (בסדר הפוך) כדי למחוק
score1	String	מאחסן את הניקוד הראשון
score2	String	מאחסן את הניקוד השני
score_current	String	מאחסן את הניקוד הנוכחי
temp	String	ערך זמני של סטרינג
c	String	שומר את בחירת המשתמש
auth	FirebaseAuth	התחברות לפיירבייס
flag	Boolean	לבדוק אם נכנסו לפעולה פעם 1
flag2	Boolean	לבדוק אם נכנסו לפעולה פעם 1
f100	Boolean	לבדוק אם נכנסו לפעולה פעם 1
fif	ImageView	מאפשר להשתמש ב/ 5050

פעולות:

פעולה היוצרת את המסך , מאתחלת את המשתנים ומתחילה את התהליך-

```
protected void onCreate(Bundle savedInstanceState)
```

פעולה המקשרת בין התפריט "אודות" אל התפריט בעמוד הזה -

```
public boolean onCreateOptionsMenu(Menu menu)
```

פעולה אשר מבצעת פעולות בהתאם למה שנלחץ בתפריט "אודות" -

```
public boolean onOptionsItemSelected(MenuItem item)
```

פעולה אשר מונעת מהמשתמש לחזור אחורה על ידי הלחצן בטלפון -

```
public void onBackPressed()
```

פעולה אשר בוחרת את אפשרות 1 בעת לחיצה -

```
place1.setOnClickListener()
```

פעולה אשר בוחרת את אפשרות 2 בעת לחיצה -

```
place2.setOnClickListener()
```

פעולה אשר בוחרת את אפשרות 3 בעת לחיצה -

```
place3.setOnClickListener()
```

פעולה אשר בוחרת את אפשרות 4 בעת לחיצה -

```
place4.setOnClickListener()
```

פעולה אשר משתמשת בעזרה של 5050 / בעת לחיצה -

```
fif.setOnClickListener()
```

מסך סרוויס מוזיקה MUSICSERVICE ACTIVITY –

טבלת משתנים:

שם משתנה	סוג	מטרה
mediaPlayer	MediaPlayer	לנגן מוזיקה מתי שמופעל

פעולות:

הפעולה מחזירה ערך **null** -

```
public IBinder onBind(Intent intent)
```

הפעולה יוצרת ומתחילה את המדיה פלייר ושמה את ההגדרות שלו, מחזירה 1 בסוף -

```
public int onStartCommand(Intent intent, int flags, int startId)
```

הפעולה דורסת ועוצרת את המדיה פלייר -

```
public void onDestroy()
```


עצם LeaderboardsUser –

טבלת משתנים:

שם משתנה	סוג	מטרה
s	String	אחסון ושמירה נוחה על שם המשתמש של כל אחד

פעולות:

בנאי של העצם -

```
public LeaderboardsUser (String s)
```

פעולות גט וסט למשתנה העצם -

```
public String getS ()
```

```
public void setS (String s)
```

פעולה אשר מציגה ב TOAST את ערך המשתנה -

```
public String toString ()
```

עצם LeaderboardsUser –

טבלת משתנים:

שם משתנה	סוג	מטרה
found	Boolean	כדי להחזיר אמת או שקר - בדיקה אם השם המבוקש קיים בדאטה בייס
stack	Stack<LeaderboardsUser>	לעבור על המחסנית ולבדוק אם השם המבוקש קיים במחסנית

פעולות:

בנאי של העצם -

```
public LeaderboardsSearched (String s, Boolean found ,Stack<LeaderboardsUser> stack )
```

פעולה אשר מקבלת שם ובודקת אם השם תקין, במידה וכן מחזירה אמת או שקר אם לא -

```
public Boolean IsValid (String name)
```

פעולה אשר מקבלת שם, בודקת אם השם משתמש קיים וקוראת ל IsValid , לבסוף מחזירה אמת או שקר-

```
public Boolean IsExist (String name)
```

פעולות גט וסט למשתנים של העצם -

```
public Boolean getFound ()
```

```
public void setFound(Boolean found)
```

```
public Stack<LeaderboardsUser> getStack ()
```

```
public void setStack (Stack<LeaderboardsUser> stack)
```

פעולה אשר מציגה ב TOAST את הערכים -

```
public String toString ()
```

עצם Users –

טבלת משתנים:

שם משתנה	סוג	מטרה
mail	String	לאחסן את האימייל של המשתמש
password	String	לאחסן את הסיסמא של המשתמש
username	String	לאחסן את השם של המשתמש
coins	int	לאחסן את המטבעות של המשתמש
points	int	לאחסן את הנקודות של המשתמש

פעולות:

בנאי ריק של העצם -

```
Users ()
```

בנאי של העצם -

```
public Users (String mail, String password, int coins, int points, String username)
```

פעולות גט וסט למשתנים של העצם -

```
public String getUsername ()
public void setUsername (String username)
public String getMail ()
public void setMail (String mail)
public String getPassword ()
public void setPassword (String password)
public int getCoins ()
public void setCoins (int coins)
public int getPoints ()
public void setPoints (int points)
```

פעולה אשר מציגה ב TOAST את הערכים –

```
public String toString ()
```

– עצם Vipuser

טבלת משתנים:

שם משתנה	סוג	מטרה
access	Boolean	כאשר ערך זה true אז יש למשתמש גישה לדברים של VIP, וכאשר false אין .

פעולות:

בנאי של העצם -

```
public Vipuser(String mail, String password,int coins, int points,String username,Boolean acces)
```

פעולות גט וסט למשתנה של העצם -

```
public Boolean getAccess()
```

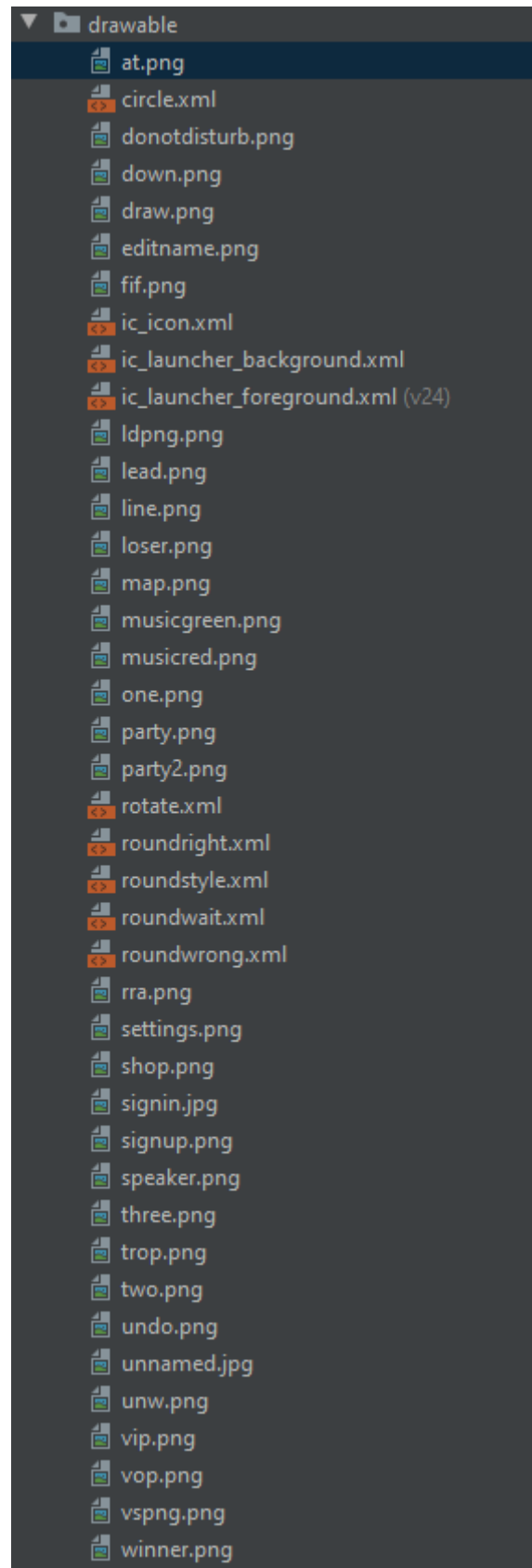
```
public void setAccess(Boolean access)
```

פעולה אשר מציגה ב TOAST את הערכים –

```
public String toString()
```

נספחים

קבצים בפרויקט:



Broadcast:

```
public class Rec extends BroadcastReceiver {
    @Override
```

```

        public void onReceive(Context context,
                               Intent intent) {
            FirebaseAuth auth =
                FirebaseAuth.getInstance();
            String id =
                auth.getCurrentUser().getUid();
            DatabaseReference r =
                FirebaseDatabase.getInstance().getReference("T
                aken");

            r.child(id).child("busy").addValueEventListener
                r(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull
                    DataSnapshot snapshot) {
                        if (snapshot.exists())
                        {
                            try
                            {
                                if
                                (intent.getAction().equals("android.provider.T
                                elephony.SMS_RECEIVED"))
                                {
                                    Bundle bundle =
                                        intent.getExtras();
                                    if (bundle !=
                                        null)
                                    {
                                        String num =
                                            "";
                                        String msg =
                                            "";
                                        Object[]
                                        pdusObj = (Object[]) bundle.get("pdus");

```

```

final
    SmsMessage[] messages = new
    SmsMessage[pduObj.length];
    for (int i =
0; i < pduObj.length; i++) {
        if
            (Build.VERSION.SDK_INT
            >=Build.VERSION_CODES.M)
            {
                String
                format = bundle.getString("format");

                messages[i] =
                SmsMessage.createFromPdu((byte[]) pduObj[i],
                format);
            }
            else
            {

                messages[i] =
                SmsMessage.createFromPdu((byte[]) pduObj[i]);
            }
            num =
            messages[i].getOriginatingAddress();
            msg =
            messages[i].getMessageBody();
        }

        Toast.makeText(context, "New SMS from: " +num
        + "-" + "\n"+msg, Toast.LENGTH_LONG).show();
        SmsManager
        smsManager = SmsManager.getDefault();

        smsManager.sendTextMessage(num, null, "Please

```



```

do not disturb! I'm playing a game!", null,
                                null);
                                }
                                }
                                }
                                catch (Exception e)
                                {
                                }
                                }
                                }
                                @Override
                                public void onCancelled(@NonNull
                                DatabaseError error) {
                                }
                                });
                                }
                                }

```

Service:

```

public class MusicService extends Service {
    private MediaPlayer mediaPlayer;

```

```

        @Nullable
        @Override
        public IBinder onBind(Intent intent) {
            return null;
        }

        @Override
        public int onStartCommand(Intent intent,
                                   int flags, int startId) {
            mediaPlayer = MediaPlayer.create(this,
                                             R.raw.soundd);
            mediaPlayer.setLooping(true);
            mediaPlayer.setVolume(100,100);
            mediaPlayer.start();
            return START_STICKY;
        }

        @Override
        public void onDestroy() {
            super.onDestroy();
            mediaPlayer.stop();
        }
    }
}

```