

## תיק פרויקט – ReFood



נושא : אפליקציה לתרומות

שם בית ספר : בליך

שם פרויקט : ReFood

שם התלמיד : ים בנג'י

שם המורה : יוסי ברון

ת"ז : 325385656

שנה : 2021

## תוכן עניינים

3	הקדמה.....
4	מפת האפליקציה.....
5	תרשים זרימה.....
6	ניתוח מערכות (Access).....
7	Layout, Widgets.....
8	ספריות חיצוניות.....
9	טכנולוגיות.....
11	טכנולוגיה מתקדמת.....
12	תהליכים ותהליכונים Processes& Threads.....
13	אוספים.....
14	מסרים- intents.....
15	שימוש במשאבי האפליקציה.....
16	תכנות מונחה אירועים ושימוש במחלקה פנימית.....
17	רכיבים מתקדמים : תיבות שיח, תפריט, הודעות.....
19	טלפוניה.....
20	חריגות.....
21	תכנות מונחה עצמים.....
22	שימוש בענן firebase.....
23	תמונות הטבלאות.....
24	רישום מחלקות.....
28	מסכי דוגמה מהאפליקציה.....
32	מדריך משתמש.....
33	ביבליוגרפיה.....

## ReFood

האפליקציה "ReFood" היא אפליקציה המאגדת את הפעולות של ארגון תרומות אוכל לנזקקים.

לפעמים, קשה לעקוב אחר כל האירועים שארגון תרומות עושה, כמו מקום בו ניתן לתרום אוכל/ משחקי קופסא/ בגדים ומקום בו ניתן לאסוף אוכל/ משחקי קופסא/ בגדים, ולכך נוצר מצב שבו הרבה אנשים שרוצים לתרום ולעזור לא יכולים/ לא יודעים איך לעשות את זה. ואנשים נזקקים שרוצים לקבל אוכל לא יודעים איך הם יכולים לקבל.

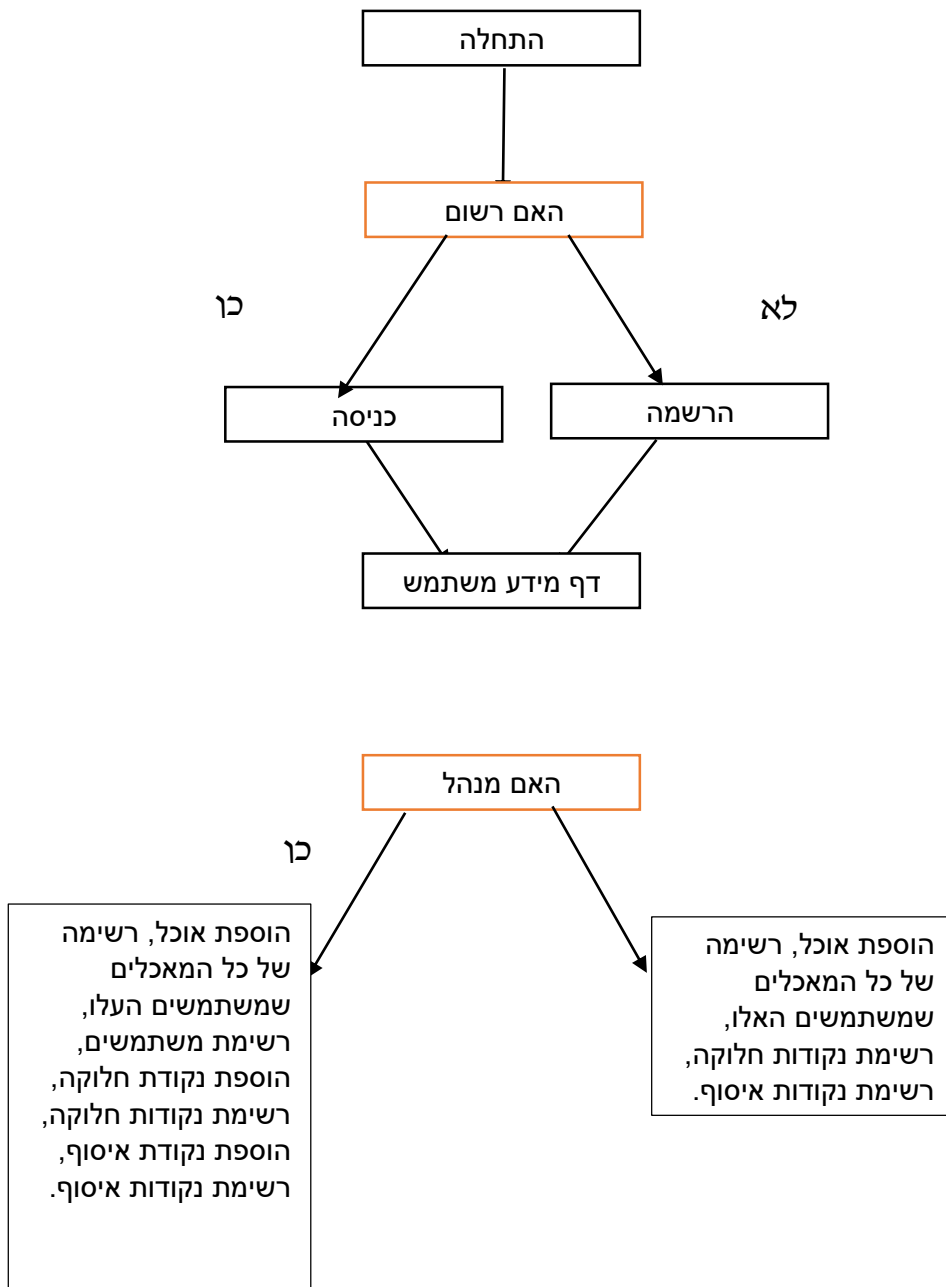
האפליקציה, מציעה למשתמש להירשם לנקודות חלוקה במקרה והוא רוצה לתרום, או להירשם לנקודות חלוקה במקרה והוא רוצה לקבל תרומה.

בנוסף, האפליקציה נותנת שירות נוסף שבו אתה יכול לעלות אוכל שנשאר לך בבית, כלומר במקרה בו הכנת יותר מידי אוכל ונשאר לך ממנו אתה יכול פשוט לעלות אותו לאפליקציה ומישהו כבר יבוא לקחת אותו ממך במקום לזרוק לפח.

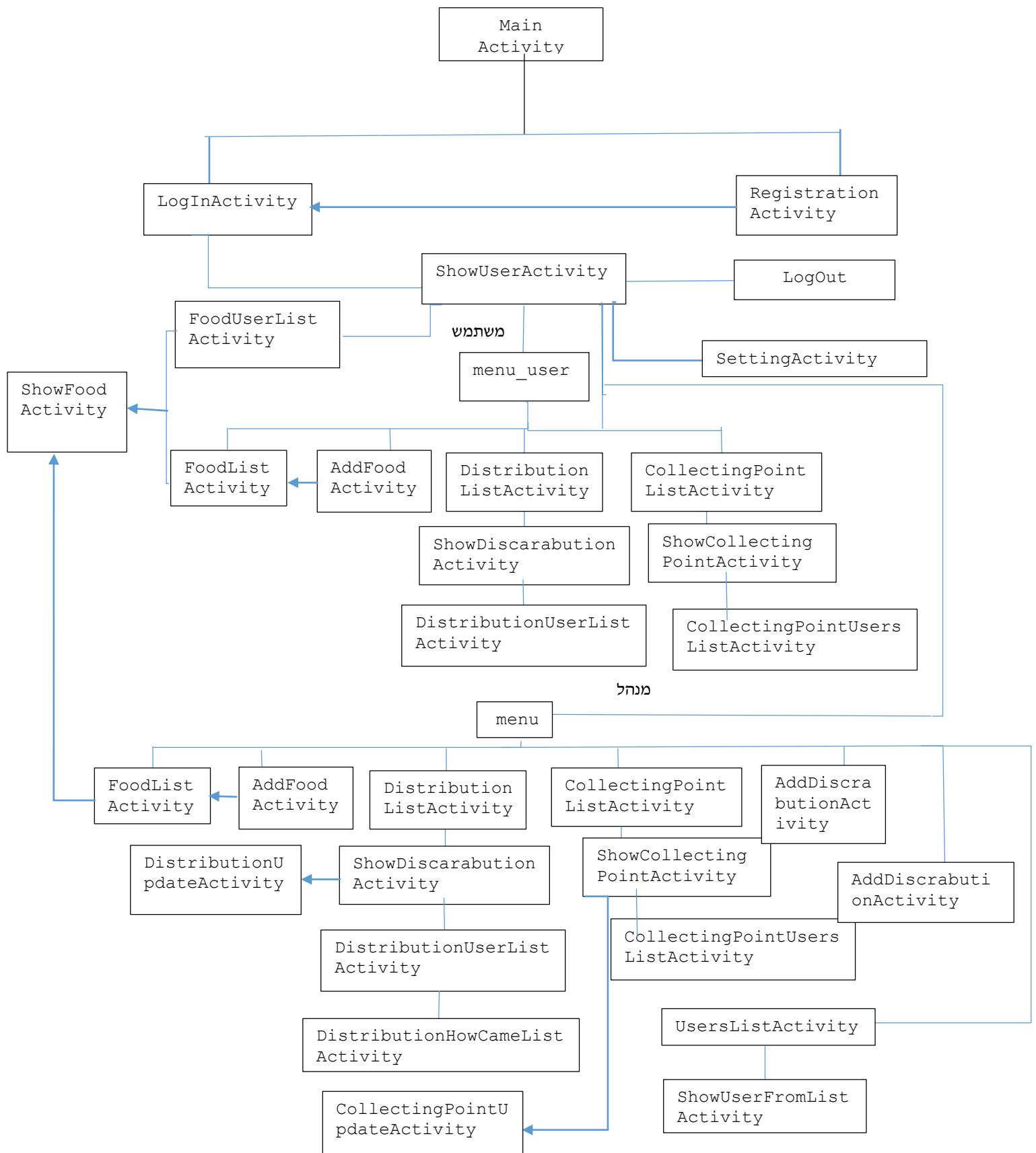
האפליקציה נועדה כדי לתרום לקהילה, כדי למזער את זריקת האוכל לפח, ואת תרבות הצריכה.

האפליקציה "ReFood" בעוצבת בעיצוב מינימלי בצע ירוק ולבן, הצבע הירוק מסמן את הסביבה, מחזור של דברים, שזה מה שהאפליקציה באה לייצג תרומה לקהילה על יד כך שמאחדים אותה ותורמים לאיכות הסביבה.

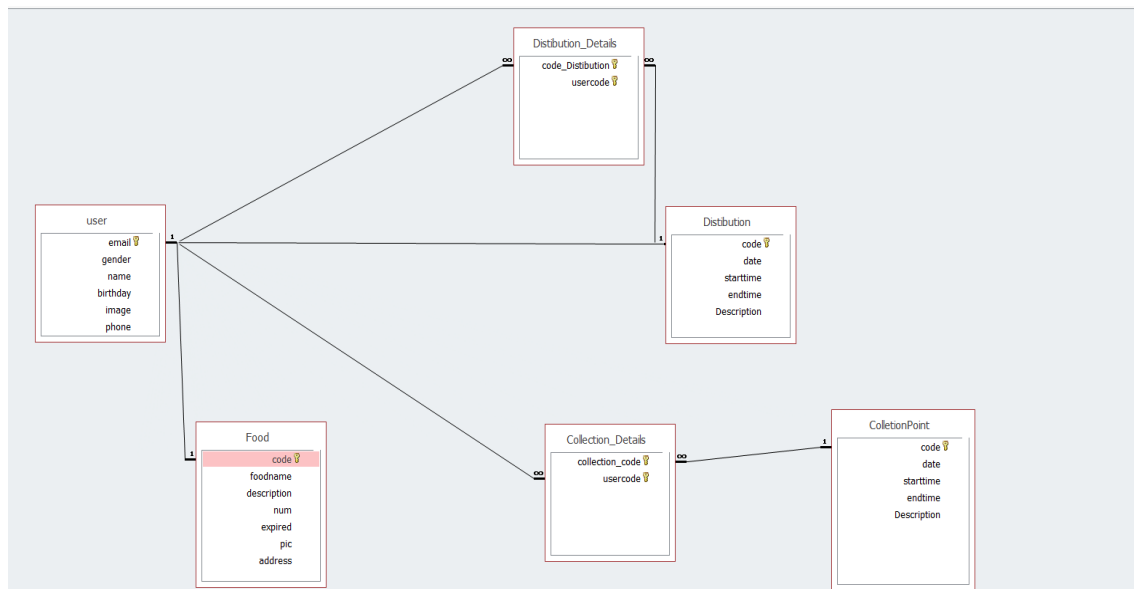
## תרשים זרימה:



## מפת אפליקציה:



## ניתוח מערכות (Access):



## טכנולוגיות, ספריות חיצוניות, Widgets :

### Widgets

- TextView
- ImageView
- Button
- ListView
- EditText
- TimePicker
- DatePicker
- CardView
- ImageButton
- CircleImageView
- Radio Button
- CheckBox

### Layouts

- LinearLayout
- RelativeLayout
- ConstraintLayout

## ספריות חיצוניות:

- **FireBase:**

שרותי אחסון בענן המוצעים על ידי גוגל בהם השתמשתי במהלך הפרויקט.  
דוגמא לפעולה שפונה לשירותי הענן:

- ```
private FirebaseStorage  
firebaseStorage=FirebaseStorage.getInstance();
```

- **CircleImageView:**

דוגמא לשימוש באפליקציה:

```
<de.hdodenhof.circleimageview.CircleImageView  
    android:id="@+id/imgcrl"  
    android:layout_width="120dp"  
    android:layout_height="120dp"  
    android:layout_rowSpan="2"  
    android:layout_columnSpan="2"  
    android:layout_alignParentStart="true"  
    android:layout_alignParentTop="true"  
    android:layout_alignParentEnd="true"  
    android:layout_marginStart="145dp"  
    android:layout_marginLeft="15sp"  
    android:layout_marginTop="29dp"  
    android:layout_marginEnd="146dp"  
    android:src="@drawable/profile"  
    app:civ_border_color="#FF000000"  
    app:civ_border_width="2dp" />
```



## טכנולוגיות:

- BroadcastReceiver:

מערכת האנדרואיד פולטת כל מיני הודעות לדוגמא כשהטלפון נכנס למצב הטענה, על מנת שהאפליקציות יוכלו לדעת דבר כזה קיים ה-Broadcast Receiver אשר מאזין להודעות מסוג זה הנשלחות ממערכת האנדרואיד לכלל האפליקציות הפעילו. באפליקציה שלי ניתן לראות את מצב הסוללה הקיים בטלפון.

דוגמא לשימוש באפליקציה:

```
public class BatteryReceiver extends BroadcastReceiver {  
  
    TextView tv;  
    BatteryReceiver(TextView tv){  
        this.tv=tv;  
    }  
    @Override  
    public void onReceive(Context context, Intent intent) {  
  
        int percentage= intent.getIntExtra("level",0);  
        if (percentage!=0)  
        {  
            tv.setText (percentage+"%");  
        }  
    }  
}
```

- Animation

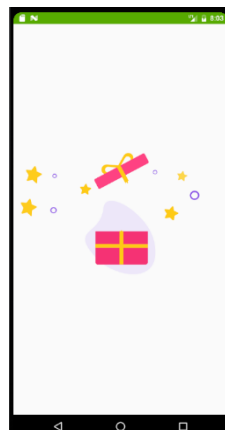
נמצא ב SplashScreen

```
<com.airbnb.lottie.LottieAnimationView  
    android:id="@+id/lottie"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:lottie_autoPlay="true"  
    app:lottie_loop="true"  
    app:lottie_rawRes="@raw/gift"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"/>  
</androidx.constraintlayout.widget.ConstraintLayout>
```

בהתחלה



בסוף



## טכנולוגיה מתקדמת:

- GPS: מציאת מיקום נוכחי

בדף showfoodActivity משתמש יכול לראות היכן מי שהעלה את האוכל נמצא, ולקבל חיבור ישיר לגוגל מפות מהכתובת הנוכחית לכתובת של מי שהעלה את האוכל

```
navigate.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // Create a Uri from an intent string. Use the result to  
        create an Intent.  
        Uri googleMapsIntentUri = Uri.parse("google.navigation:q=" +  
address.getText().toString());  
        Intent mapIntent = new Intent(Intent.ACTION_VIEW,  
(googleMapsIntentUri));  
        mapIntent.setPackage("com.google.android.apps.maps");  
        startActivity(mapIntent);  
    }  
});
```

## תהליכים ותהליכונים Processes & Threads

תהליכון (Thread) הוא רצף של פעולות שמתבצעות באופן עצמאי במטרה לבצע משימה מסוימת, וזאת במקביל לתהליכונים אחרים. במערכת האנדרואיד מוקצה תהליך יחיד ותהליכון עבור כל יישום שמבקש לרוץ.

דוגמא לתהליכון : מסך ספלאש בעת הכניסה לאפליקציה שמוביל באופן אוטומטי לדף הראשי.

```
new Handler().postDelayed(new Runnable() {  
    @Override  
    public void run() {  
  
        Intent it =new Intent(Splashscreen.this, MainActivity.class);  
        startActivity(it);  
        finish();  
  
    }  
}, splashtime);
```

## אוספים:

- ArrayList<E>

```
private ArrayList<User> finallist = new ArrayList<User>();
```

- HashMap<K,v>

```
mUserDatabase.addListenerForSingleValueEvent(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        if (dataSnapshot.exists() && dataSnapshot.getChildrenCount() > 0) {  
            Map<String, Object> map = (Map<String, Object>) dataSnapshot.getValue();  
            if (map.get("name") != null) {  
                name = map.get("name").toString();  
                mNameField.setText(name);  
            }  
            if (map.get("phone") != null) {  
                phone = map.get("phone").toString();  
                mPhoneField.setText(phone);  
            }  
            if (map.get("gender") != null) {  
                userSex = map.get("gender").toString();  
                gender.setText(userSex);  
            }  
            if (map.get("birthDay") != null) {  
                birthday = map.get("birthDay").toString();  
                bday.setText(birthday);  
            }  
            if (map.get("pic") != null) {  
                profileImageUrl = map.get("pic").toString();  
                ShowImage(profileImageUrl);  
                //StorageReference storageRef =  
            }  
            if (map.get("email") != null) {  
                email = map.get("email").toString();  
                memail.setText(email);  
            }  
        }  
    }  
});  
  
@Override  
public void onCancelled(DatabaseError databaseError) {  
}  
});
```

## מסרים - Intents

במהלך האפליקציה השתמשתי במספר רב של מסרים שונים:

- Intent שמחליף את הדף המוצג בדף אחר:

```
Intent it = new Intent(UsersListActivity.this, ShowUser.class);
startActivity(it);
```

- Intent שמפעיל את פעולת ההתקשרות בטלפון (מוגדר גם בתור טלפוניה):

```
startActivity(new Intent(Intent.ACTION_CALL, Uri.parse(dial)));
```

- Intent להחלפת קוד של תמונה:

```
Intent it = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
```

### שימוש במשאבי האפליקציה:

משאבי האפליקציה הם צרכים חיצוניים שנצטרך ביישום, כגון תמונות, מחרוזות, צבעים, סגנונות ועוד. כאמור משאבים אלו אמורים להימצא בתיקה המתאימה תחת תיקיית המשאבים בפרויקט. res/ שיטה זו מאפשרת להתאים את היישום המפותח לפי סוג המכשיר, שפה ומצב מסך המכשיר.

דוגמא לשימוש באפליקציה :

- הגדרת אובייקט לפי זיהוי מהמחלקה R.id :

```
button=findViewById(R.id.updetdis);
```

- הגדרת תצוגה לפי תמונה במחלקה drawable :

```
app:srcCompat="@drawable/ic_settings"
```

- Uri : נותן את האפשרות לבחור תמונה ממאגר התמונות שבטלפון ומציג את התמונה שנבחרה באפליקציה

```
private static Uri selectedImage=null;
```

```
imgcrl.setImageURI(selectedImage);
```

```
Intent it = new Intent(Intent.ACTION_PICK,  
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
```

### תכנות מונחה אירועים ושימוש במחלקה הפנימית:

כל אינטראקציה המתבצעת עם המסך והאפליקציה נקלטת דרך "מאזינים" (המהווים חלק מהView), מערכת האנדרואיד מדווחת ליישומים ולרכיבים הגרפיים על אירועים שמתרחשים והקשורים לאותו

- חיפוש משתמש באפליקציה באמצעות פרמטרים שונים שמכניסים לתיבת

```
isSearched = true;
namestr = name.getText().toString();
if (namestr.equals(""))
    Toast.makeText(UsersListActivity.this, "Choose name or username",
        Toast.LENGTH_SHORT).show();

else {

    for (int i = 0; i < userslist.size(); i++) {
        String fname = userslist.get(i).getName();
        //String lname = userslist.get(i).getLName();
        String username = userslist.get(i).getEmail();
        if (fname.equals(namestr) || /* lname.equals(namestr) || */
            username.equals(namestr))
        {
            userssearched.add(userslist.get(i));
            uids.add(uid.get(i));
        }
    }

    if (userssearched.size() == 0)
        Toast.makeText(UsersListActivity.this, "There are no users in this
name", Toast.LENGTH_SHORT).show();

    else {
        UsersListAdapter adapter3 = new
UsersListAdapter(UsersListActivity.this, userssearched, picpath);
        listView.setAdapter(null);
        listView.setAdapter(adapter3);}
    }
```

- לחיצה על כפתור הדף משתמשים מעבירה לדף

```
startActivity(new Intent(Login.this,
    ShowUser.class));
```



## רכיבים מתקדמים: תפריט, תיבת דו שיח, הודעות:

ישנם תפריטים, תיבות דו שיח והודעות טוסט (Toast) שמוצגות על ידי האפליקציה בסיום, התרחשות ולפני ביצוע פעולות בכדי לדווח למשתמש מידע חשוב על התהליך.

- ProgressDialog:

```
private ProgressDialog prg =null;

prg=new ProgressDialog(this);
prg.setTitle("RegistrationActivity Data");
prg.setMessage("Saving Data");
prg.setProgressStyle(ProgressDialog.STYLE_SPINNER);
prg.setCancelable(false);
```

- AlertDialog:

```
new AlertDialog.Builder(FoodUserListActivity.this)
    .setTitle("AreYou Sure?")
    .setMessage("Do you want to delete this item")
    .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            mAdapter.deleteItem(position);
            mAdapter.notifyDataSetChanged();
            ref.child(foodid.get(position)).removeValue();
            refFood.child(foodid.get(position)).removeValue();
        }
    })
    .setNegativeButton("No",null)
    .show();
```

- Toast:

```
Toast.makeText(LoginActivity.this, "You are in",
Toast.LENGTH_SHORT).show();
```

- Menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

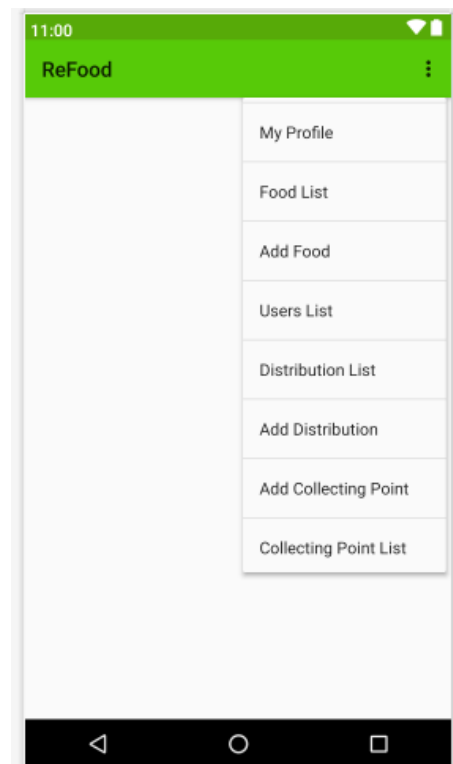
    <item
        android:id="@+id/item2"
        android:title="My Profile"
        app:showAsAction="never" />
    <item
        android:id="@+id/item3"
        android:title="Food List"
        app:showAsAction="never"/>
</item>
```

```

        android:id="@+id/item4"
        android:title="Add Food"
        app:showAsAction="never"/>
<item
    android:id="@+id/item5"
    android:title="Users List"
    app:showAsAction="never"/>
<item
    android:id="@+id/item6"
    android:title="Distribution List"
    app:showAsAction="never"/>
<item
    android:id="@+id/item7"
    android:title="Add Distribution"
    app:showAsAction="never"/>
<item
    android:id="@+id/item8"
    android:title="Add Collecting Point"
    app:showAsAction="never"/>
<item
    android:id="@+id/item9"
    android:title="Collecting Point List"
    app:showAsAction="never"/>

</menu>

```



## טלפוניה:

באפליקציה כאשר רוצים לתקשר אם בן אדם הרוצה למסור אוכל ניתן להתקשר עליו.

```
public String findPhoneUser (String idUser) {
    DatabaseReference refUsers =
    FirebaseDatabase.getInstance().getReference("Users");

    refUsers.child(idUser).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            String phoneOfUser =
            dataSnapshot.child("phone").getValue().toString();

            if (phoneOfUser.trim().length() > 0) {
                if (ContextCompat.checkSelfPermission>ShowFood.this,
                    Manifest.permission.CALL_PHONE) !=
                PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions>ShowFood.this,
                        new String[]{Manifest.permission.CALL_PHONE},
                REQUEST_CALL);
                } else {
                    String dial = "tel:" + phoneOfUser;
                    startActivity(new Intent(Intent.ACTION_CALL,
                Uri.parse(dial)));
                }
            } else {
                Toast.makeText>ShowFood.this, "Enter Phone Number",
                Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
    return idUser;
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    if (requestCode == REQUEST_CALL) {
        if (grantResults.length > 0 && grantResults[0] ==
        PackageManager.PERMISSION_GRANTED) {
            makePhoneCall();
        } else {
            Toast.makeText(this, "Permission DENIED",
            Toast.LENGTH_SHORT).show();
        }
    }
}
```

### חריגות:

לפעמים בעת כתיבת הקוד אנו נאלצים להתאים אותו במקרים של מצבים חריגים כדי שהאפליקציה לא תקרוס ותמשיך לרוץ. אנו מתמודדים עם מצבים כאלו בעזרת קוד try and catch.  
דוגמא לחריגה באפליקציה:

```
try
{
    final File localfile=File.createTempFile(picname,suffix);

    storageRef.getFile(localfile).addOnSuccessListener(new
    OnSuccessListener<FileDownloadTask.TaskSnapshot>()
    {
        @Override
        public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot)
        {
            Bitmap bitmap =
            BitmapFactory.decodeFile(localfile.getAbsolutePath());
            // set image based on selected text
            imageView.setImageBitmap(bitmap);
        }
    });
}
catch (IOException e)
{
    e.printStackTrace();
}
```

## תכנות מונחה עצמים:

בפרויקט השתמשתי בסוגי תכנות כמו הורשה ופולימורפיזם שהם סוג של תכנות מונחה עצמים.

- שימוש בהורשה:

```
public class FoodListAdapter extends BaseAdapter  
public class BatteryReceiver extends BroadcastReceiver
```

- שימוש בפולימורפיזם (המרה כלפי מטה):

```
email = (TextView) findViewById(R.id.useremail);  
mProfileImage = (ImageView) findViewById(R.id.userimg);
```

## שימוש בענן Firebase:

לאורך כל האפליקציה ישנו שימוש בענן חיצוני – Firebase , בין אם זה שימוש של לקוח, מנהל , שמירת פריטים ועוד .להלן מספר שימושים שונים של טבלאות מהענן שבאות לידי ביטוי באפליקציה.

- Users:

טבלת ה- user מעניקה למשתמש את היכולת להתחבר לאפליקציה, הפעולות העיקריות של הענן דרך טבלה זו הן התחברות והרשמה.

- CollectingPoint:

טבלת ה- CollectingPoint מכילה בתוכה את כל המידע הקשור לנקודת האיסוף (זמן, מקום, תאריך), כל מי שאמר שיגיע.

- Distribution:

טבלת ה- Distribution מכילה בתוכה את כל המידע הקשור לנקודת האיסוף (זמן, מקום, תאריך), כל מי שאמר שיגיע.

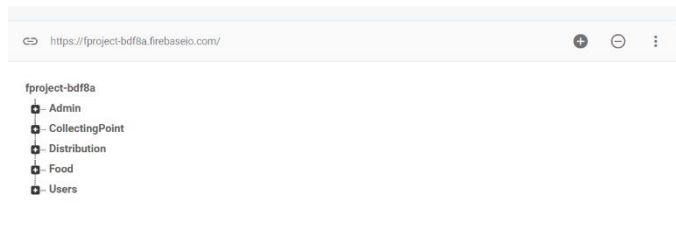
- Food:

טבלת ה- Food מכילה בתוכה את כל הנתונים הזקוקים לאוכל שמשתמש מעלה כגון : תאריך תפוגה, כמות האוכל, סוג האוכל, תמונה, המשתמש שהעלה את אותו אוכל.

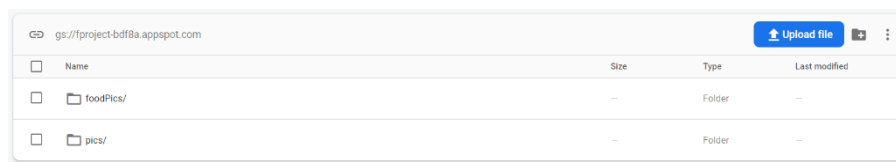
שם הענן: fproject

## תמונות הטבלאות:

- טבלאות המכילות נתונים בסיסיים ומידע בRealTime :



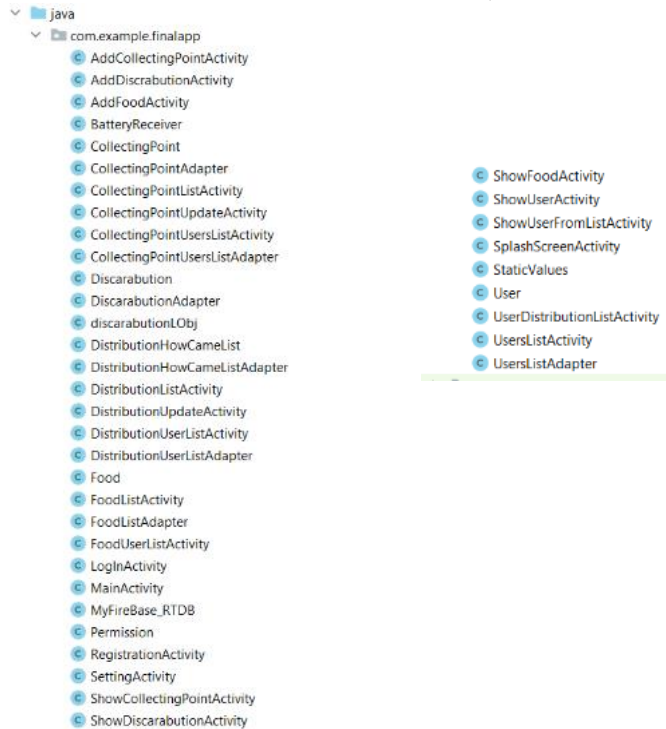
- תיקיות המכילות קבצים (תמונות של משתמשים ואוכל)



- טבלאות המכילות את המידע הדרוש להתחברות בתור משתמש – Authentication :



## רישום מחלקות:



## מחלקות בונות אובייקטים:

- `public class User`

מחלקה המייצגת משתמש. בעלת תכונות כדוגמת, שם משתמש, סיסמא, תמונת פרופיל, אימייל

- `public class Food`

מחלקה המייצגת אוכל שמשתמש העלה. בעלת תכונות כמו, שם האוכל, תאריך תפוגה, תמונה.

- `public class Discarbution`

מחלקה המייצגת נקודת חלוקה של תרומות. בעלת תכונות כמו, מקום, זמן התחלה, זמן סיום, תאריך.

- `public class CollectingPoint`

מחלקה המייצגת נקודת איסוף של תרומות. בעלת תכונות כמו, מקום, זמן התחלה, זמן סיום, תאריך.



## דפי JAVA הקשורים לדפי Layout

- `public class CollectingPointListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_collecting\_point\_list.xml

- `public class CollectingPointUpdateActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_collecting\_point\_update.xml

- `public class CollectingPointUsersListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_collecting\_point\_users\_list.xml

- `public class DistributionHowCameListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_distribution\_how\_came\_list.xml

- `public class DistributionListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_distribution\_list.xml

- `public class DistributionUpdateActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_distribution\_update.xml

- `public class DistributionUserListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_distribution\_user\_list.xml

- `public class FoodListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_food\_list.xml

- `public class FoodUserListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_food\_user\_list.xml

- `public class LogInActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_log\_in.xml

- `public class MainActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_main.xml
- `public class RegistrationActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_registration.xml
- `public class SettingActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_setting.xml
- `public class ShowCollectingPointActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_show\_collecting\_point.xml
- `public class ShowDiscarabutionActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_show\_discarabution.xml
- `public class ShowFoodActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_show\_food.xml
- `public class ShowUserActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_show\_user.xml
- `public class SplashScreenActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_splash\_screen.xml
- `public class UserDistributionListActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_user\_distribution\_list.xml
- `public class UsersListActivity`  
המחלקה המכילה פעולות הקשורות לדף  
activity\_users\_list.xml

- `public class AddCollectingPointActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_add\_collecting\_point.xml

- `public class AddDiscrabutionActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_add\_discrabution.xml

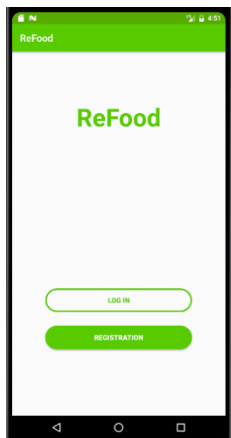
- `public class AddFoodActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_add\_food.xml

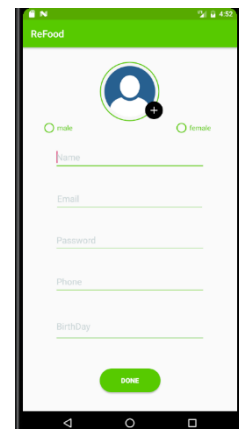
- `public class ShowUserFromListActivity`

המחלקה המכילה פעולות הקשורות לדף  
activity\_show\_user\_from\_list.xml

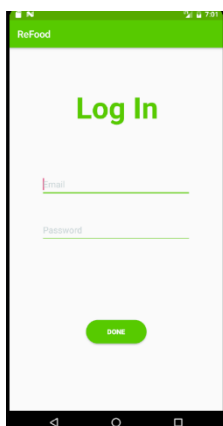
## מסכי דוגמה מהאפליקציה:

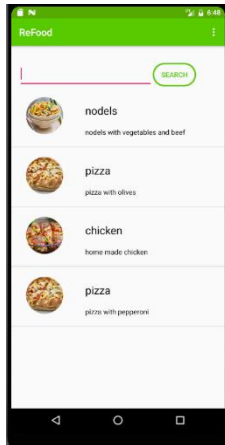


### עמוד הרשמה

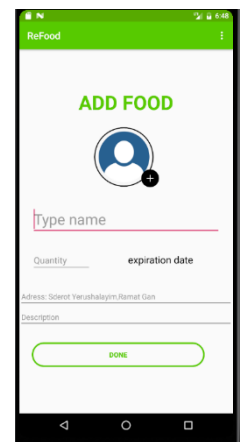


### עמוד התחברות

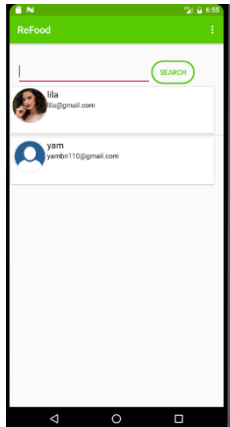




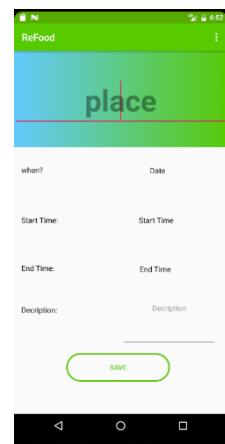
עמוד שמראה  
רשימה של אוכל



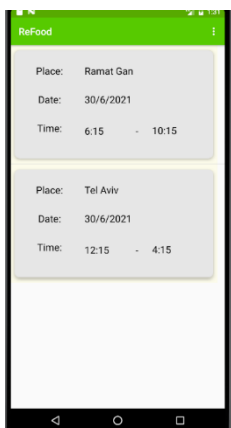
עמוד להוספת  
אוכל



עמוד שמראה  
רשימה של

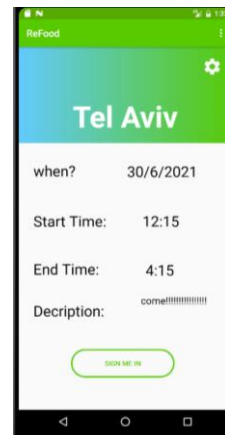


עמוד להוספת נקודות  
חלוקה



עמוד רשימת  
נקודות חלוקה

עמוד שמראה  
נקודת חלוקה



## מדריך משתמש:

- מנהל:  
אימייל: [yambn110@gmail.com](mailto:yambn110@gmail.com)  
סיסמה: 123456
- משתמש:  
אימייל: [lila@gmail.com](mailto:lila@gmail.com)  
סיסמה: lila123



### ביבליוגרפיה:

- StackOverflow
- חומר הלימוד בכיתה
- <https://stackoverflow.com/questions/10407159/how-to-manage-startactivityforresult-on-android>
- <https://developer.android.com/guide/topics/ui/dialogs>
- <https://developer.android.com/guide/topics/ui/menus>