



שם התלמיד - נדב אלטבט

כיתה - יב' 2, מקיף ה' אמירים

נושא הפרויקט - אפליקציה לניהול סניף בתנועת הנוער "הנוער הלאומי בית"ר"

ת"ז - 214167207

שם המורה - יוסף ברון

## תוכן עניינים:

מבוא לאפליקציה: 3

תרשים זרימה ומפת אפליקציה: 4

Access: 5

תכנות מונחה עצמים: 6-7

אוספים: 8

חריגות: 8

פעילות: 9

ממשק משתמש: 9-10

תכנות מונחה אירועים: 11

מסרים: 11

רכיבים מתקדמים: 12-14

משאבי האפליקציה: 15

אחסון נתונים: 16-17

Threads & Processes: 18

תכנון וניתוח אפליקציות: 18-19

מקלט שידורים: 20

טלפוניה: 20

אנימציה: 21

תיאור מחלקות: 21-25

מדריך משתמש: 26

ביבליוגרפיה: 27

## NLB - FINAL PROJECT

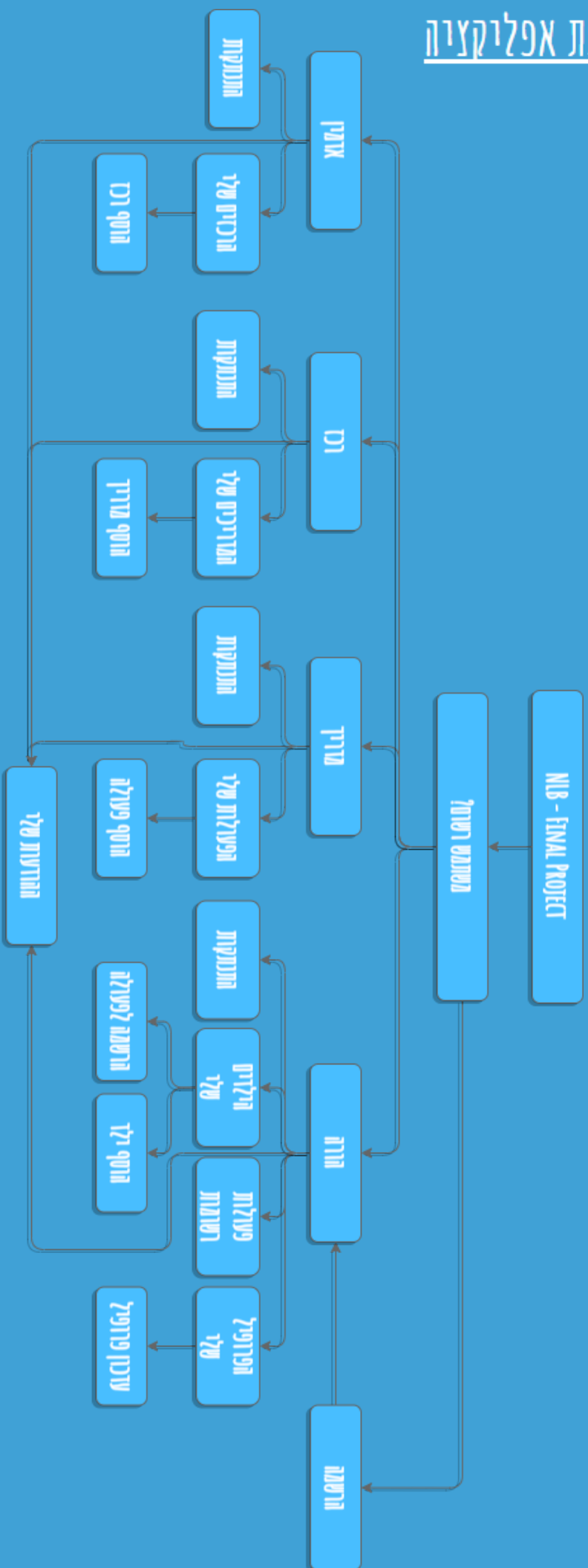
האפליקציה "NLB" על שם תנועת הנוער "הנוער הלאומי בית"ר" היא אפליקציה חדשנית שנוצרה על מנת לעזור לרכזי הסניפים של תנועת הנוער בניהול טפסי הפעולות של המדריכים בכל סניף, שמירת נתוני החניכים והרשומים במסד נתונים מסודר ותפעול יעיל של המתרחש בכל הסניפים ברחבי הארץ.

תנועת הנוער הלאומי בית"ר הינה תנועת נוער ציונית חברתית. התנועה מחנכת את חבריה על פי משנתו החברתית של זאב ז'בוטינסקי: פיתוח מנהיגות צעירה, אהבת הארץ, קליטת עלייה, אינטגרציה, 5 המ"מים, רוח ההדר ורוח התגר.

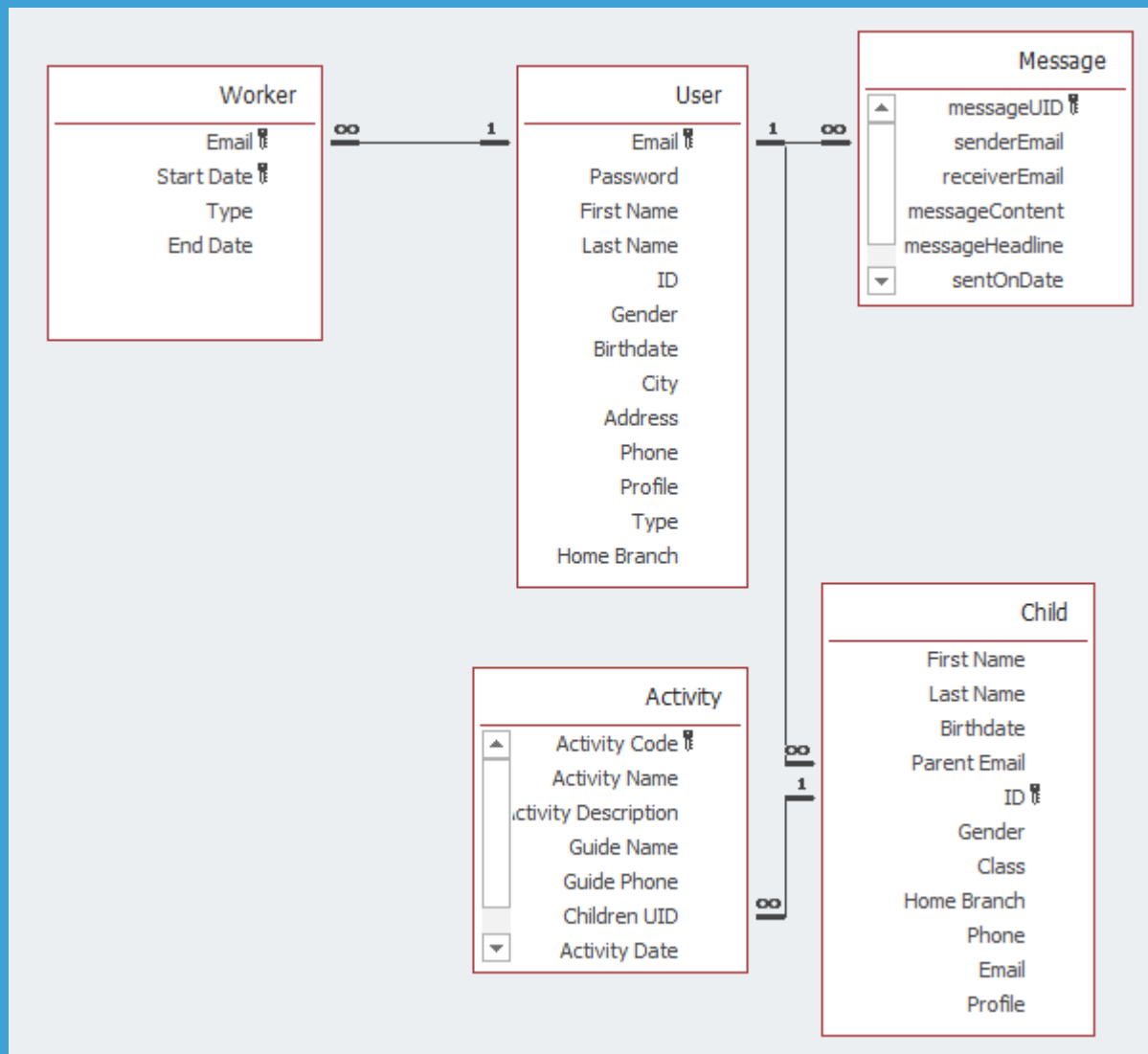
בתור חניך ומדריך בתנועה מזה 5 שנים אני מודע למגוון הבעיות שיש בכל נושא שמירת הנתונים בכל הקשור לפעולות ולרישומים לסניפים, אם זה בגלל התחלפות של רכזי הסניפים ואיתם גם נעלמים הנתונים, או בגלל שכל הפעולות נכתבות על דפים והם נעלמים לאט לאט בין כל הקלסרים המסורבלים ועוד מגוון של בעיות. לכן, החלטתי להרים את הכפפה וליצור אפליקציה שתעזור להנהלת התנועה לעקוב בצורה מסודרת ודיגיטלית אחר כל הנתונים השונים של כלל סניפי התנועה ברחבי הארץ.

האפליקציה מאפשרת להורים לרשום את ילדיהם לתנועת הנוער ולפעולות שונות במהלך השבוע בסניפים, בנוסף לך מאפשרת האפליקציה למדריכים להעלות את טפסי הפעולות שלהם באופן דיגיטלי ובכך להיות מסוגלים לשלוף בכל עת כל פעולה שיצטרכו ובקלות. ולבסוף, האפליקציה תאפשר לרכזי הסניפים לעקוב אחר המדריכים הרשומים בכל סניף, להוסיף ולהסיר בכל רגע נתון את מדריכיהם בסניפים השונים.

## תרשים זרימה ומפת אפליקציה



## ACCESS:



## תכנות מונחה עצמים:

דוגמא לתכנות מונחה עצמים בפרוייקט ניתן לראות במחלקת CHILD\_CRADLSVIEW\_ADAP.  
מטרת המחלקה היא להמיר אובייקט מסוג CHILD לעיצוב מותאם אישית בתוך LISTVIEW:

כותרת המחלקה -

```
public class child_cardlsview_adap extends BaseAdapter {
```

פעולות המחלקה -

פעולה בונה - יוצרת את הממיר (האדפטר/ADAPTER):

```
public child_cardlsview_adap(ArrayList<Child> arrChild, ArrayList<String> arrChildUID, Activity ctx, ProgressDialog prg) {  
    this.arrChild = arrChild;  
    this.arrChildUID = arrChildUID;  
    this.ctx = ctx;  
    this.prg = prg;  
}
```

פעולת GETCOUNT - מחזירה את אורך הרשימה של האובייקטים המתקבלים לצורך המרתה לאיבר ברשימה:

```
@Override  
public int getCount() {  
    return this.arrChild.size();  
}
```

פעולת OBJECT:

```
@Override  
public Object getItem(int i) {  
    return null;  
}
```

פעולת GETITEMID:

```
@Override  
public long getItemId(int i) {  
    return 0;  
}
```

## פעולת getView - אחראית על בניית האיבר הנוכחי מהרשימה. הפעולה מקבלת את האובייקט במקום position ברשימת האובייקטים וממירה את האובייקט לאיבר ברשימה:

```
@Override
public View getView(int position, View view, ViewGroup viewGroup) {

    this.prg.show();
    LayoutInflater inflater =ctx.getLayoutInflater();

    final View myrow= inflater.inflate(R.layout.child_listview,null,true);

    TextView child_name = myrow.findViewById(R.id.child_name);
    child_name.setText(this.arrChild.get(position).getChild_first_name() + " " + this.arrChild.get(position).getChild_last_name());

    TextView child_ID=myrow.findViewById(R.id.child_ID);
    child_ID.setText(this.arrChild.get(position).getChild_ID());

    TextView child_gender=myrow.findViewById(R.id.child_gender);
    child_gender.setText(this.arrChild.get(position).getChild_gender());

    TextView child_birthdate=myrow.findViewById(R.id.child_birthdate);
    child_birthdate.setText(this.arrChild.get(position).getChild_birthdate().getDay() + "/" +
        this.arrChild.get(position).getChild_birthdate().getMonth() + "/" + this.arrChild.get(position).getChild_birthdate().getYear());

    TextView child_class=myrow.findViewById(R.id.child_class);
    child_class.setText(this.arrChild.get(position).getChild_class_letter() + " " + this.arrChild.get(position).getChild_class_number());

    TextView child_school=myrow.findViewById(R.id.child_school);
    child_school.setText(this.arrChild.get(position).getChild_school());

    TextView child_branch=myrow.findViewById(R.id.child_branch);
    child_branch.setText(this.arrChild.get(position).getChild_branch());

    TextView child_email=myrow.findViewById(R.id.child_email);
    child_email.setText(this.arrChild.get(position).getChild_email());

    TextView child_phone=myrow.findViewById(R.id.child_phone);
    child_phone.setText(this.arrChild.get(position).getChild_phone());
```

.....

## אוספים:

דוגמא לשימוש באוספים ניתן למצוא בWORKER\_CARDLSVIEW\_ADAP. האוסף הוא ArrayList מסוג אובייקט Worker והוא מחזיק בתוכו את כל העובדים השמורים בענן:

```
private ArrayList<Worker> arrWorkers;
```

```
final String[] branchName = new String[1];
```

## חריגות:

דוגמא לשימוש בכלי למניעת חריגות (TRY & CATCH) ניתן למצוא בMY\_MESSAGES. נשתמש בTRY & CATCH כדי להימנע מקריסת האפליקציה במקומות שבהם ישנה אפשרות לגיטימית לקריסה, כמו בדוגמא הזו שבה נבדוק האם הפעולה תקרוס בניסיון השמת רשימה שאמורה להחזיק הודעות שקיבל המשתמש (במקרה שהמשתמש לא קיבל שום הודעה הפעולה אמורה לקרוס אך נימנע מקריסה בשימוש בTRY & CATCH):

```
try {  
    final message_cardlsview_adap adap =  
        new message_cardlsview_adap(messageArray, usersArray, ctx: nadav.altabet.nlb.my_messages.this);  
    my_messages.setAdapter(adap);  
} catch (Exception e) {  
    Toast.makeText( context: my_messages.this, text: "אין לך הודעות חדשות", Toast.LENGTH_LONG).show();  
}
```



## פעילות :ACTIVITY

דוגמא לשימוש בפעילות ניתן למצוא במחלקת MAIN\_ACTIVITY. הפעילות מייצגת מסך ממשק משתמש גרפי יחיד מהאפליקציה:  
כותרת המחלקה -

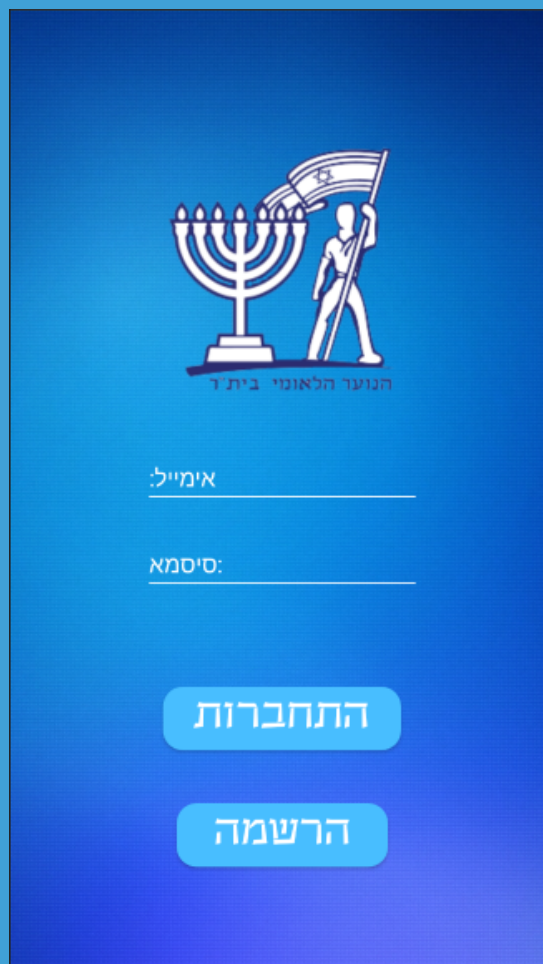
```
public class main_activity extends AppCompatActivity {
```

פעולות onCreate - הפעולה מופעלת בעת פתיחת דף הפעילות:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);  
    getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_HIDE_NAVIGATION);  
    super.onCreate(savedInstanceState);
```

## ממשק משתמש / UI INTERFACE

ממשק משתמש ניתן למצוא בהמון מקומות באפליקציה, לדוגמא WELCOME\_SCREEN.XML. הממשק הוא למעשה הכלי שמאפשר למשתמש לבצע פעולות שונות באפליקציה בצורה נוחה וידידותית:



## WIDGETS, טכנולוגיות, ספריות חיצוניות:

WIDGETS:

SPINNER	TEXTVIEW
CARDVIEW	IMAGEVIEW
IMAGEBUTTON	SCROLLVIEW
LISTVIEW	EDITTEXT
FLOATING ACTION BUTTON	RADIOBUTTON
PROGRESS BAR	BUTTON
	SPINNER

LAYOUTS:

CONSTRAINTLAYOUT  
LINEARLAYOUT

ספריות חיצוניות:

ספריות חיצוניות לצורך קישוריות האפליקציה לענן האחסון FIREBASE:

```
implementation 'com.google.firebase:firebase-auth:16.0.5'  
implementation 'com.google.firebase:firebase-database:16.0.4'  
implementation 'com.google.firebase:firebase-storage:16.0.4'
```

ספרייה חיצונית לאובייקט עיצובי בצורת קלף:

```
implementation "androidx.cardview:cardview:1.0.0"
```

## תכנות מונחה אירועים

דוגמא לתכנות מונחה אירועים ניתן לראות בREGISTER. תכנות מונחה אירועים מאפשר לנו להפעיל פעולות רק תחת מצבים מסוימים או "אירועים" מסוימים, כמו לחיצת כפתור לצורך העניין:

```
btnPick.setOnClickListener((v) → {  
    startActivityForResult(new Intent(Intent.ACTION_PICK,  
        MediaStore.Images.Media.EXTERNAL_CONTENT_URI), requestCode: 1);  
});
```

## מסרים INTENTS:

דוגמא לשימוש בINTENT בפרויקט ניתן למצוא במחלקת WELCOME\_SCREEN. מעבר בין הדפים השונים באפליקציה -

```
startActivity(new Intent(welcome_screen.this, nadav.altabet.nlb.register.class));
```

פעולה שפותחת תיבת הודעה ושואלת האם להעלות תמונה מתוך גלריית הטלפון או מתוך המצלמה. לכל מקרה יש INTENT ייחודי שפועל בסנכרון עם פעולת onActivityResult שפועלת בהתאם לקוד הבקשה שהיא מקבלת -

```
AlertDialog.Builder builder = new AlertDialog.Builder(context: add_child.this);  
builder.setTitle("העלאת תמונה לאפליקציה").  
setMessage("?צילום תמונה או בחירה מהגלריה").  
setPositiveButton(text: "תמונה מהגלריה", (dialogInterface, i) → {  
    startActivityForResult(new Intent(Intent.ACTION_PICK,  
        MediaStore.Images.Media.EXTERNAL_CONTENT_URI), requestCode: 1);  
    dialogInterface.cancel();  
}).setNegativeButton(text: "צילום תמונה", (dialogInterface, i) → {  
    startActivityForResult(new Intent(MediaStore.ACTION_IMAGE_CAPTURE), requestCode: 2);  
    dialogInterface.cancel();  
}).setCancelable(false);  
AlertDialog dialog = builder.create();  
dialog.setTitle("העלאת תמונה לאפליקציה");  
dialog.show();
```

## רכיבים מתקדמים:

תפריט MENU - דוגמא לשימוש בתפריט ניתן למצוא במחלקת PARENTS\_CHILDREN.  
התפריט למעשה מאפשר לנו לנווט בין הדפים השונים באפליקציה בעיצוב נוח למשתמש  
ונגיש מכל דף:

JAVA:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.parents_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.menu_parents_profile:
            startActivity(new Intent(this, parents_profile.class));
            return true;
        case R.id.menu_parents_children:
            startActivity(new Intent(this, parents_children.class));
            return true;
        case R.id.menu_parents_logout:
            Client.setCurrentUser(null);
            startActivity(new Intent(this, welcome_screen.class));
            return true;
        case R.id.menu_parents_homepage:
            startActivity(new Intent(this, parents_hub.class));
            return true;
        case R.id.menu_parents_online_store:
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

## XML:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      >
    <item
        android:id="@+id/menu__parents_homepage"
        android:icon="@drawable/homepage_100px"
        android:title="@string/דף_בית"
        app:showAsAction="always"
        />
    <item
        android:id="@+id/menu__parents_profile"
        android:icon="@drawable/profile_100px"
        android:title="@string/פרופיל"
        app:showAsAction="always"
        />
    <item
        android:id="@+id/menu_parents_online_store"
        android:icon="@drawable/online_store_100px"
        android:title="@string/חנויות"
        app:showAsAction="always"
        />
    <item
        android:id="@+id/menu_parents_children"
        android:icon="@drawable/children_100px"
        android:title="@string/ילדים"
        app:showAsAction="always"
        />
    <item
        android:id="@+id/menu_parents_logout"
        android:icon="@drawable/logout_100px"
        android:title="@string/התנתקות"
        app:showAsAction="always"
        />
</menu>
```

תיבת דו שיח - דוגמא לשימוש בתיבת דו שיח ניתן למצוא במחלקת REGISTER. המשתמש נעזר בכלי זה בזמן ההרשמה לאפליקציה כאשר בוחר את תאריך הלידה שלו:

```
btnDate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DatePickerDialog datePickerDialog = new DatePickerDialog(register.this, new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
                btnDate.setText(dayOfMonth + "/" + (month+1) + "/" + year);
                chosenDay = dayOfMonth;
                chosenMonth = month+1;
                chosenYear = year;
            }
        },year,month,day);
        datePickerDialog.show();
    }
});
```

מצלמה - דוגמא לשימוש במצלמה ניתן למצוא במחלקת ADD\_CHILD. המשתמש יכול לבחור להעלות תמונה של הילד מהגלריה או לחילופין לצלם תמונה באמצעות המצלמה:

```
btnPickChild.setOnClickListener((v) -> {
    AlertDialog.Builder builder = new AlertDialog.Builder( context: add_child.this);
    builder.setTitle("העלאת תמונה לאפליקציה").
    setMessage("צילום תמונה או בחירה מהגלריה?").
    setPositiveButton( text: "תמונה מהגלריה", (dialogInterface, i) -> {
        startActivityResult(new Intent(Intent.ACTION_PICK,
            MediaStore.Images.Media.EXTERNAL_CONTENT_URI), requestCode: 1);
        dialogInterface.cancel();
    }).setNegativeButton( text: "צילום תמונה", (dialogInterface, i) -> {
        startActivityResult(new Intent(MediaStore.ACTION_IMAGE_CAPTURE), requestCode: 2);
        dialogInterface.cancel();
    }).setCancelable(false);
    AlertDialog dialog = builder.create();
    dialog.setTitle("העלאת תמונה לאפליקציה");
    dialog.show();
});
```

```
else if(requestCode == 2 && data!=null)
{
    chosen=true;
    selectedPic = getImageUri( inContext: add_child.this,
        (Bitmap) data.getExtras().get("data"));
    profileChild.setImageURI(selectedPic);

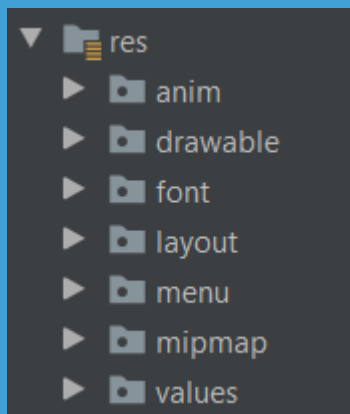
    String[] filepathcolumn={MediaStore.Images.Media.DATA};
    Cursor cursor=getContentResolver().query(selectedPic,filepathcolumn,
        selection: null, selectionArgs: null, sortOrder: null);
    cursor.moveToFirst();
    int columnIndex=cursor.getColumnIndex(filepathcolumn[0]);

    picName=cursor.getString(columnIndex);

    picName=picName.substring(picName.lastIndexOf( str: "/")+1);
}
```

## משאבי האפליקציה:

לאפליקציה תיקויה ייחודית (RESOURCES) שבה היא מאחסנת את כל משאביה - תמונות, דפי ממשק (XML), קבצי מדיה, תפריטים וכו':



ניתן לראות מס' דוגמאות באפליקציה לשימוש במידע הנשאב מתיקיות משאבי האפליקציה כמו למשל ב MainActivity כאשר נשלפת אנימציה מתיקיות ANIM שבתוך תיקיות משאבי האפליקציה RES:

```
anim = AnimationUtils.loadAnimation( context: this,R.anim.top_animation);
```

פעולה המקבלת אובייקט של תפריט ובודק את המס' הייחודי שלו במשאבי האפליקציה:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.menu__parents_profile:
            startActivity(new Intent( packageContext: this, parents_profile.class));
            return true;
        case R.id.menu__parents_children:
            startActivity(new Intent( packageContext: this, parents_children.class));
            return true;
        case R.id.menu__parents_logout:
            Client.setCurrentUser(null);
            startActivity(new Intent( packageContext: this, welcome_screen.class));
            return true;
        case R.id.menu__parents_homepage:
            startActivity(new Intent( packageContext: this, parents_hub.class));
            return true;
        case R.id.menu__parents_online_store:
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

## FIREBASE - נתונים


במהלך השימוש באפליקציה מתבצעת תקשורת עם הענן הייעודי שלה שעליו נשמרים כל הנתונים החשובים לחיבור המשתמש. הענן הייעודי של האפליקציה הוא למעשה ענן של FIREBASE שבעזרתו ניתן לנהל את כל נושא רישום המשתמשים ואחסון הנתונים שלהם כגון תמונות פרופיל, שמות, ילדים רשומים וכו'.

### טבלת AUTHENTICATION:

Search by email address, phone number, or user UID					Add user	↺	⋮
Identifier	Providers	Created	Signed In	User UID	↑		
test2@mail.com	✉	Apr 8, 2021	Apr 8, 2021	251lvxWNe0cLsriZJCzKEbAFih2			
a@a.com	✉	Jan 15, 2021	Jan 15, 2021	9SpuHAZ7rHSRUuT5YhStklrhAyW2			
test@gmail.com	✉	Nov 27, 2020	Apr 11, 2021	T5KQX8bH0wXGFGmEJ7K14NxT...			
guide@mail.com	✉	Jan 14, 2021	Apr 8, 2021	WWLatnuUJgSf0ZnVEyhmHnmj5q...			
coordinator@mail.com	✉	Jan 11, 2021	Apr 8, 2021	YIBTBi6v3lcWIFRXv9xjl9CwXpA2			
adminlb@gmail.com	✉	Nov 27, 2020	Apr 8, 2021	jAEwhaEAO9fCErtH89PIP7mEqtt1			
test@mail.com	✉	Jan 14, 2021	Jan 18, 2021	yv3Hx7T3Q0fq0pLuTOZvrDPCHSx1			
Rows per page: 50					1 - 7 of 7	Activate Window Go to Settings to a	

### מסד הנתונים בענן - REALTIME DATABASE:

<https://nlb-project-2287b.firebaseio.com/>

 Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

[Learn more](#) [Dismiss](#)

nlb-project-2287b

- Activities
- Branches
- Children
- Classes
- Messages
- Parents
- Roles
- Workers

Database location: United States (us-central1)

Activate Window  
Go to Settings to a



## אחסון הקבצים בענן - STORAGE:

gs://nlb-project-2287b.appspot.com					Upload file		
<input type="checkbox"/>	Name	Size	Type	Last modified			
<input type="checkbox"/>	Activities/	—	Folder	—			
<input type="checkbox"/>	Child Profiles/	—	Folder	—			
<input type="checkbox"/>	Profiles/	—	Folder	—			
<input type="checkbox"/>	סופס כתיבת פעולה.docx	5.6 MB	application/vnd.openofficedocument.word	Jan 10, 2021			

שורות קוד אלו ממחלקת REGISTER למעשה אחראיות על רישום המשתמש והוספתו גם למסד הנתונים וגם לטבלת AUTHENTICATION שבה מוחזקים המפתחות היעודיים של כל משתמש ומקשרים בין הסיסמא השמורה תחת הגנה לשאר נתוני המשתמש במסד הנתונים:

```
firebaseAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener( activity: register.this, (task) → {  
    if (task.isSuccessful()) {  
        FirebaseUser currentUser = firebaseAuth.getCurrentUser();  
        final String UID = currentUser.getId();  
        if (dateOfBirth.checkForAbove18(new Date(year, month, day)) > 18) {  
            databaseReference.child(UID).setValue(user).addOnCompleteListener( activity: register.this, (task) → {
```

הפעולה הבאה עוברת על כל ה"ילדים" בטבלת CHILDREN במסד הנתונים ובודקת האם האימייל השמור תחת כל ילד שמקשר בינו לבין ההורה שלו הוא אכן האימייל של המשתמש המחובר כעת, במקרה ואותו אימייל נמצא גם תחת נתוני הילד וגם אצל נתוני המשתמש נתוני הילד מתווספים לרשימה מיוחדת שנועדה להחזיק את כל נתוני הילדים של אותו הורה:

```
databaseReference.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
        Iterable<DataSnapshot> children = dataSnapshot.getChildren();  
        for (DataSnapshot childDatabase: children)  
        {  
            Child child = childDatabase.getValue(Child.class);  
            if (child.getParent_UID().equals(Client.getUID()))  
            {  
                arrChild.add(child);  
                arrChildUID.add(childDatabase.getKey());  
            }  
        }  
    }  
})
```

## THREADS & PROCESSES

תהליכון או Thread הוא רצף של פעולות שמתבצעות באופן עצמאי במטרה לבצע משימה מסוימת, וזאת במקביל לתהליכונים אחרים. ניתן למצוא דוגמא לRunnable במחלקת MainActivity. Runnable משמש להרצת הקוד לאחר פרק זמן מוגדר מראש:

```
new Handler().postDelayed(new Runnable() {  
    @Override  
    public void run() {  
        startActivity(new Intent(main_activity.this, welcome_screen.class));  
        finish();  
    }  
}, SPLASH_SCREEN);
```

## תכנון וניתוח אפליקציות

בפרק זה מוצגים דוגמאות למס' דפים שונים באפליקציה לשם תיאור המסכים של האפליקציה. כל דף שמוצג כאן הוא למעשה דף הבית של כל משתמש בהתאם לסוג אותו משתמש, ישנם 4 סוגי משתמשים בפרוייקט זה - הורה, מדריך, רכז, אדמין:



דף הבית למשתמשים מסוג הורה. מסוגלים לראות ולהוסיף את ילדיהם כחניכי התנועה בסניפים השונים, לרשום את ילדיהם לפעולות בסניפי הבית של כל ילד (וכמובן גם לבטל כל פעולה רשומה), לצפות ולעדכן בפרופיל שלהם, ולהתכתב עם משתמשים שונים באפליקציה

## ברוך הבא



התנתקות



הפעולות שלי



ההודעות שלי

דף הבית למשתמשים מסוג רכז.

דרכו ניתן לנווט לדף "המדריכים שלי" ודף "ההודעות שלי".

בדף "המדריכים שלי" יכול הרכז להוסיף מדריכים חדשים לסניפו בכך שיאשר מדריכים שממתינים לאישור מרכזם. בדף "ההודעות שלי" יכול המדריך להתכתב עם שאר משתמשי האפליקציה.

## ברוך הבא



העובדים שלי



הפרופיל שלי



התנתקות



ההודעות שלי

דף הבית למשתמשים מסוג מדריך.

דרכו ניתן לנווט לדף "הפעולות שלי" ודף "ההודעות שלי". בדף "הפעולות שלי" מסוגל המדריך לקבל תבנית לכתיבת פעולה, להעלות פעולה חדשה ולעדכן פעולה שכבר הועלתה. בדף "ההודעות שלי" יכול המדריך להתכתב עם שאר משתמשי האפליקציה.

## ברוך הבא



התנתקות



המדריכים שלי



ההודעות שלי

דף הבית למשתמש מסוג אדמין.

משתמש זה נוסף באופן ידני למערכת ואינו יכול להירשם לבד באפליקציה. הוא מסוגל להוסיף רכזים חדשים ולהסיר רכזים קיימים, לצפות בפרופיל האישי שלו, ולהתכתב עם משתמשים אחרים באפליקציה.

## מקלט שידורים BROADCAST RECEIVER

מקלט שידורים הוא מחלקה באפליקציה, שתפקידו להאזין למסרים שמשודרים. כל מקלט כזה נרשם לקליטת מסר או קבוצת מסרים, והינו מיועד ע"י מערכת האנדרואיד והיישום ששייך לו במסרים שמשודרים, כלומר מתפקד כמאזין למסרי היישום והמערכת לפי הרשאות ויעוד. דוגמא למקלט שידורים ניתן למצוא במחלקת WELCOME\_SCREEN למקלט שמאזין לאחוז הסוללה של המכשיר:

```
private class BatteryBroadcastReceiver extends BroadcastReceiver
{
    private final static String BATTERY_LEVEL = "level";
    @Override
    public void onReceive(Context context, Intent intent)
    {
        int level = intent.getIntExtra(BATTERY_LEVEL, defaultValue: 0);
        mBatteryLevelText.setText("Battery level:" + " " + level);
        mBatteryLevelProgress.setProgress(level);
    }
}
```

## טלפוניה

ניתן למצוא דוגמא לשליחת מסרון באמצעות SMSMANAGER במחלקת REGISTER\_TO\_ACTIVITY. ההורה יכול לרשום חניך לפעולה ובעת ההרשמה נשלחת הודעה למדריך הפעולה:

```
String phone = arrActivities.get(position).getGuidePhone();
String Message = parents_children.arrChild.get(position).getChild_first_name() + " " +
    parents_children.arrChild.get(position).getChild_last_name() + " נרשם לפעולה " + "'" +
    arrActivities.get(position).getActivity_name() + "'";
SmsManager sm = SmsManager.getDefault();
sm.sendTextMessage(phone, scAddress: null, Message, sentIntent: null, deliveryIntent: null);
Toast.makeText(context: register_to_activity.this, text: "הודעה על ההרשמה נשלחה למדריך!", Toast.LENGTH_SHORT).show();
```

## אנימציה

במסך הפתיחה של האפליקציה מוצגת אנימציה מסוג `VIEWANIMATION` שמשתמשת בקובץ `XML` שנמצא בתיקיית `ANIM` שבמשאבי האפליקציה ולפי אותו קובץ האובייקט שעליו מופעלת האנימציה (במקרה זה `ImageView`) "יודע" אילו פעולות לבצע (כמו סיבוב, דעיכה, תזוזה ועוד מגוון סוגי אנימציה):

יישום האנימציה על אובייקט `ImageView` -

```
anim = AnimationUtils.loadAnimation( context: this,R.anim.top_animation);  
appLogo.setAnimation(anim);
```

תיאור האנימציה בקובץ ה-`XML` -

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
  
    <translate  
        android:fromXDelta="0%"  
        android:fromYDelta="-70%"  
        android:duration="2000"  
    />  
  
    <alpha  
        android:fromAlpha="0.1"  
        android:toAlpha="1.0"  
        android:duration="1500"  
    />  
</set>
```

## תיאור מחלקות

ACTIVITY -

מחלקה לייצוג אובייקט מסוג ACTIVITY. האובייקט מחזיק נתונים של הפעולות השונות של המדריכים.

ACTIVITY\_CARDLSVIEW\_ADAP -

מחלקה המקבלת רשימת אובייקטים מסוג ACTIVITY וממירה אותם לתצוגה בתוך LISTVIEW.

ADD\_CHILD -

מחלקה המאפשרת הוספת ילד לאפליקציה ע"י ההורה שלו.

ADD\_GUIDE -

מחלקה המאפשרת הוספת מדריך לאפליקציה ע"י הרכז שלו.

ADD\_WORKER -

מחלקה המאפשרת הוספת רכז ע"י המנהל (אדמין) של האפליקציה.

ADMINS\_HUB -

המחלקה של דף הבית למשתמשים מסוג אדמין

ADMINS\_WORKERS -

מחלקה המאפשרת תצוגה של הרכזים לכל סניף

CHILD -

מחלקה לייצוג אובייקט מסוג CHILD. האובייקט מחזיק נתונים של הילד(ים) של הורה המשתמש באפליקציה.

CHILD\_LSVIEW\_ADAP -

מחלקה המקבלת רשימת אובייקטים מסוג CHILD\_LSVIEW\_ADAP וממירה אותם לתצוגה בתוך LISTVIEW.

CHILDERN\_UPDATE -

מחלקה המאפשרת עדכון של אחד מילדי ההורה המשתמש באפליקציה.







REGISTER\_TO\_ACTIVITY -

מחלקה המאפשרת רישום של אחד הילדים של ההורה לפעולה של מדריך המתקיימת בסניף.

USER -

מחלקה לייצוג אובייקט מסוג User. אובייקט User מחזיק נתונים של משתמשים כגון שם מלא, אימייל משתמש, תאריך לידה וכו'.

WELCOME\_SCREEN -

מחלקה המאפשרת התחברות או הרשמה לאפליקציה לכל סוגי המשתמשים בהתאם לנתוני ההתחברות (לכל משתמש סוג אחר ולכן יתחבר לדף אחר בהתאם לסוג המשתמש).

WORKER -

מחלקה לייצוג אובייקט מסוג Worker. אובייקט זה מאפשר קישור בין נתוני המשתמש לנתוני העבודה של רכז מסויים המשתמש באפליקציה.

WORKER\_CARDLSVIEW\_ADAP -

מחלקה המקבלת רשימת אובייקטים מסוג Worker וממירה אותם לתצוגה בתוך ListView

WORKERS\_UPDATE -

מחלקה המאפשרת עדכון של נתוני העבודה של רכז המשתמש באפליקציה.

## מדריך משתמש

לאפליקציה 4 סוגי משתמשים - הורה, מדריך, רכז ואדמין:

-משתמש מסוג הורה ניתן ליצור בקלות בדף ההרשמה באפליקציה בתנאי שהמשתמש מעל גיל 18 (נעשית בדיקה באמצעות פעולה שבודקת את תאריך הלידה).

\*אימייל למשתמש מסוג הורה: [TEST@GMAIL.COM](mailto:TEST@GMAIL.COM)

\*סיסמא למשתמש מסוג הורה: 111111

-במקרה והמשתמש מתחת לגיל 18 מוצאת לו האפשרות ליצור משתמש מסוג מדריך ולאחר מכן צריך לבקש אותו מדריך מרכז הסניף שלו שיאשר אותו דרך המשתמש של הרכז.

\*אימייל למשתמש מסוג מדריך: [GUIDE@MAIL.COM](mailto:GUIDE@MAIL.COM)

\*סיסמא למשתמש מסוג מדריך: 111111

-על מנת לקבל משתמש מסוג רכז רק משתמש מסוג אדמין יכול להפוך משתמש רגיל לרכז ולכן יש ליצור קשר עם אחד האדמינים של האפליקציה על מנת לקבל אישור להפוך את המשתמש מסוג הורה של אותו רכז למשתמש מסוג רכז.

\*אימייל למשתמש מסוג רכז: [COORDINATOR@MAIL.COM](mailto:COORDINATOR@MAIL.COM)

\*סיסמא למשתמש מסוג רכז: 111111

-ולבסוף, על מנת ליצור משתמש מסוג אדמין צריך לשנות ידנית בענן את סוג המשתמש לאדמין או ליצור ידנית לגמרי בענן את המשתמש ובכך רק מס' אנשים בעלי הגישה לענן יכולים להתחבר כאדמינים של האפליקציה.

\*אימייל למשתמש מסוג אדמין: [ADMINLB@MAIL.COM](mailto:ADMINLB@MAIL.COM)

\*סיסמא למשתמש מסוג אדמין: 111111

## ביבליוגרפיה

- חומר הלימוד בכיתה
- מדריך ליצירת ALERTDIALOG משולב עיצוב XML:  
(51) ANDROID CUSTOM DIALOG - CREATE ANDROID ALERTDIALOG WITH A CUSTOM LAYOUT - YOUTUBE
- מדריך להעלאת קבצים לאחסון בענן בFIREBASE:  
(51) HOW TO UPLOAD PDF / ANY FILE TO FIREBASE | ANDROID | IN JAVA - YOUTUBE
- מדריך לשליפת קבצים מאחסון הענן בFIREBASE:  
(51) HOW TO FETCH UPLOADED FILE(S) / PDF(S) FROM FIREBASE | ANDROID | IN JAVA - YOUTUBE
- מדריך ליצירת SPLASH SCREEN:  
(51) ANDROID SPLASH SCREEN WITH ANIMATIONS IN ANDROID STUDIO | SPLASH SCREEN ANDROID STUDIO - YOUTUBE
- ANDROID DEVELOPERS: ANDROID DEVELOPERS
- FIREBASE DOCUMENTATION: DOCUMENTATION | FIREBASE (GOOGLE.COM)