

# ATID PLUS

שם התלמיד:	דניס מרצ'נקו
ת.ז.:	PRIVATE INFO
שם המנחה:	יניב
בית ספר:	כפר גלים

## מבוא

### תיאור תוכן הספר

ספר הפרויקט יציג אפליקציה לשיעורים נוספים במתמטיקה ואנגלית, בהתאם לרמת התלמיד. בספר הפרויקט אסביר מה יצירתי של חלקי התוכנית Visual Studio-בפרויקט, במה השתמשתי ולמה החלטתי לעשות זאת. בנוסף, אצרך צילומי מסך הן מ שנכתבו על ידי, והן מהאפליקציה עצמה

### רקע לפרויקט

האפליקציה המשמשת כדלת וירטואלית ואישית, באפליקציה תוכלו לראות אילו שיעורים נוספים יש לכם שיהיו היום בסך הכל יש 2 שיעורים אפשריים: מתמטיקה ואנגלית. לכל שיעור רמת למידה, 3,4, 5 יחידות (כמו בתוכנית הלימודים הישראלית).

הפרויקט מיועד לכל תלמיד תיכון

### תהליך מחקר

תהליך מחקר

הפרויקט שלי מחקה יומן אלקטרוני. כדי להחליט איך יראה הפרויקט ואיך הוא ייראה, הגעתי לעיצוב ואיך הוא יעבוד מאפס. בנוסף, בדקתי גם איך נראית אפליקציית היומן האלקטרוני הפופולרי ביותר של בית ספר חכם. בסופו של דבר פיתחתי יומן אלקטרוני, שהוא מאוד נוח ופשוט

## אתגרים מרכזיים

### סיבות לבחירת הפרויקט

רציתי להנגיש את היומן האלקטרוני. בעולם המודרני, בו כבר ניתן לעשות הכל בטלפונים ניידים, ראוי שתהיה אפליקציה המאפשרת לראות אילו שיעורים נוספים יש לכם היום, ניתן לפתוח את היומן ולהשתמש בו בכל מקום ובכל זמן.

היומן שלי קל לשימוש ונגיש יותר, וגם מחובר לעולם הטלפונים המודרני.

באופן אישי, כמה פעמים היה לי רצון במהלך הלימודים תהיה אפליקציה כזו שאפשרה לי לראות את לוח הזמנים שלי להיום.

### צרכים שעליהם עונה הפרויקט

הפרויקט הופך את לוח הזמנים של שיעורים נוספים לזמין. היומן הוירטואלי מקל על לחיצה וצפייה בפעילויות נוספות שלך להיום כדי להפוך את החיים לקלים ככל האפשר לתלמיד תיכון. הרי כולם הולכים לכל מקום עם הטלפון שלו, אבל עוד לא שמעתי על מיישהו שהולך לכל מקום עם היומן האלקטרוני שלו עם פעילויות נוספות. כך למעשה הפרויקט הופך את לוח הזמנים לזמין. בנוסף, לתלמיד בית ספר יש עניינים אישיים משלו בשגרה, וכדי לא לשכוח איזה שיעור יש לו היום, הוא יכול פשוט לפתוח את הבקשה שלי.

### בעיה/דילמה בפיתוח הפרויקט

בהתחלה תכננתי ליצור יומן וירטואלי שיכול להיות בו רק בית ספר אחד. זה כפר גלים. בשלב מתקדם הבנתי שאולי הפרויקט שלי לא מספיק מורכב וניסיתי להבין איך לשפר אותו.

### פתרונות שנבחנו כדי לפתור את הבעיה

הייתה בעיה כיצד מורים או האחראים לשיעורים נוספים מודיעים לתלמידים על השיעור שלהם וקובעים עבורם לוח זמנים. לשם כך יצרתי זכויות גישה רק למנהלים, כך שהם בעצמם קובעים את השיעורים.

## מבנה הפרויקט

מחלקות הפרויקט והאינטראקציה ביניהן מאורגנות בשיטת MVC:

Model –classes , List

View –activity,layout

Controller – Adapters,dbclass,handlers

## - Layout, activities -

**-Layouts** - מסכי האפליקציה שכוללים את כל העיצובים והם כוללים את המסכים שהמשתמש רואה ברגע שהוא משתמש באפליקציה.

**-Activities** - תכנות שנמצא מאחורי המסכים. עבור כל מסך פריסה בפעולה, מסך איפוס סיסמה. יש קוד שמשתמש בנתונים שנקבעו על ידי המנהל (שיעורים). לדוגמה, בפעילות, אתה לוקח את המידע שהוזן למנהל, מאוחסן במסד הנתונים, ואז צצה התראת שיעור על מסך התלמיד.

המסכים העיקריים שלי לתלמיד הם: מסך התחברות, מסך הרשמה, מסך השיעורים של היום, מסך משתמש בו הוא יכול לראות את רמת הלמידה שלו ובמקרה של שינויים יפנה למנהל לשינוי הנתונים

מסכים למנהלי מערכת: רשימה של כל התלמידים, מסנן שמסנן תלמידים לפי בית ספר, רמת המתמטיקה והאנגלית שלהם ודרך להעביר שיעור לתלמידים. עבור כך מסך כזה יש אקטיביטי. באקטיביטי יש את הפעולות השונות שקורות ברגע שלוחצים על כפתורים או מכניסים מידע. לדוגמה, כאשר אתה מגדיר שיעור אתה לוקח את כל הנתונים שהוזנו וברגע שלוחצים על כפתור "אישור" מגדירים את השיעור ושולחים הודעות לתת-הקבוצה.

-Intents - מאפשרים מעבר בין מסכים ואקטיביטים שונים על מנת להעביר מידע באפליקציה.

## DB (שמירת מידע) -

מסד נתונים הוא כלי המשמש לארגון נתונים במחשב לצורך אחסון ואחזור בעת הצורך. בתוך המאגר נתונים מאוחסנים באוספים בהם מאוחסנים נתוני תלמידים. לכן, בדרך כלל לכל האלמנטים המאוחסנים באותו מסד נתונים יש אותן תכונות.

## המחלקות הראשיות שלי:

**Student**: משמש לאחסון נתוני תלמידים ולאחר מכן להשתמש בהם כדי לקבוע את רמת הידע שלו ולהקצאה נוספת של שיעורים

**AppDataHelper**: מכיל מידע והגדרת מסד נתונים וכן פונקציות מסייעות

מבני נתונים שלי:

user collection - Firebase Firestore. אוסף זה מזהה את כל המשתמשים ומידע על רמת ההדרכה שלהם בענן.

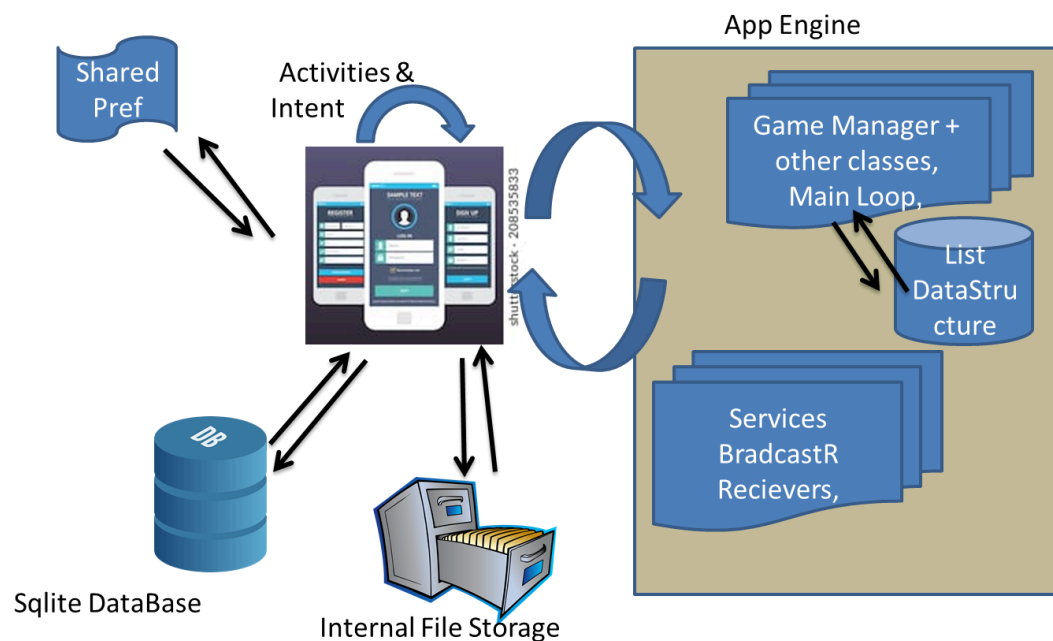
## מנגנוני אנדרואיד

Service - מאפשרים ליצור ולנהל תהליכים הקשורים למה שקורה ברקע האפליקציה, כמו למשל השמעת מוזיקת רקע במהלך ריצת האפליקציה.

## Adapters – לתצוגת רשימות

אדפטור מתאם בין הDB לבין ה list . כלומר הוא מתאם בין הרשימה המוצגת באחד המסכים לבין המידע מהבסיס נתונים שרוצים להכניס לרשימה.

השתמשתי רק באדפטור אחד: עבור משתמשים והכנסתם לרשימה

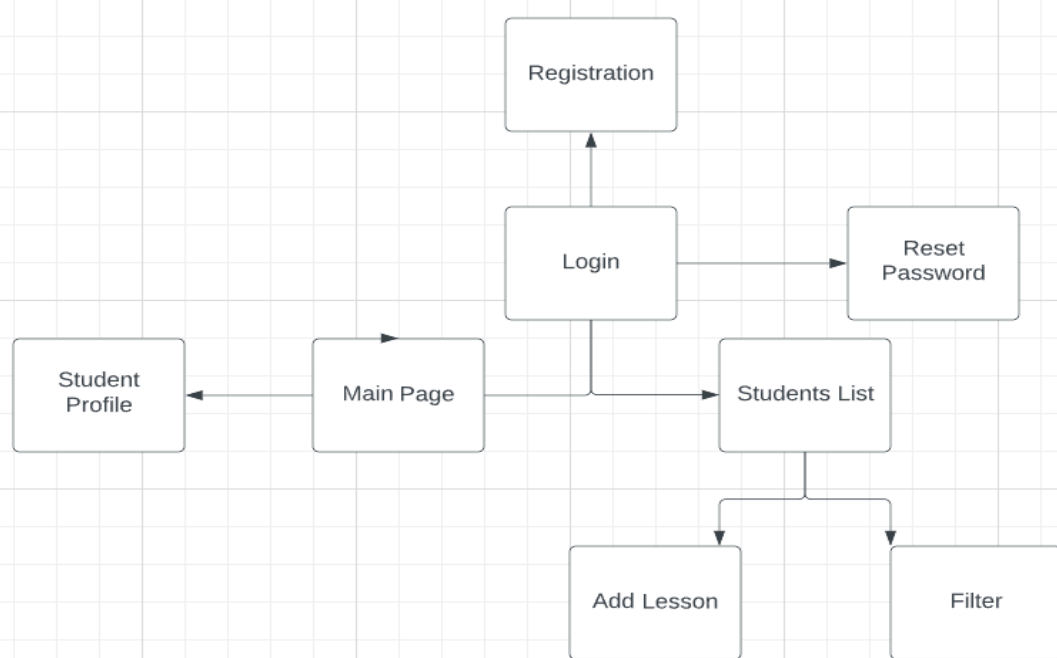


## מדריך למשתמש

### מטרת המערכת

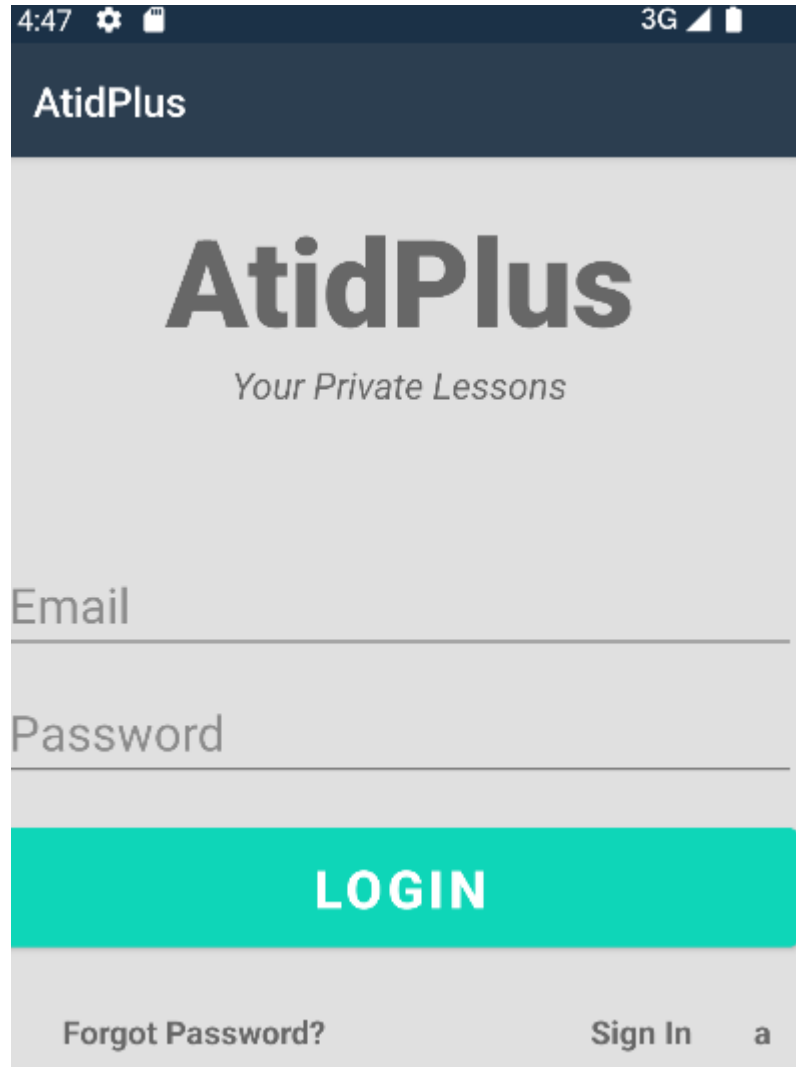
באמצעות האפליקציה יכולים המשתמשים להכניס למכשיר סמארטפון אישי את לוח השיעורים הנוספים, וכך ליצור יומן אלקטרוני אישי של תגבורים. האפליקציה עוזרת לך להציג בנוחות ובקלות את השיעורים שלך בכל מקום ובכל זמן מבלי לחשוש לשכוח אותם. לאפליקציה שני סוגי שיעורים. מתמטיקה ואנגלית. המנהל או האחראי על העברת שיעורים נוספים יכולים לתת לך שיעורים בשני נושאים אלו.

## תרשים זרימה בין מסכים –



## תפעול המערכת

### :Login in



AtidPlus

AtidPlus

Your Private Lessons

Email

Password

**LOGIN**

Forgot Password? Sign In a

תיאור מסך כניסה והיכן מוצג

לאחר הכניסה, המסכים של המנהל והמשתמש יהיו שונים  
כניסה כולל בתוכו מעבר למסכים הבאים, וכן במסכים האלה כניסה מופיע:

מה שכולם רואים:

- (1) מסך כניסה – מסך כניסה (LoginIn.xml)
- (2) מסך הקטגוריות המכיל את כל המתכונים (ResetingPasswordEmail.xml)
- (3) מסך הרשמה ((Registration. xml) **מסך בית (MainActivity.xml):**



מסכים:

הרשמה (Registration.Xml)

6:49

3G

AtidPlus

Your Private Lessons

User Name

Email

School Type

Password

Confirm Password

REGISTER USER

Return to Login

מסך רישום, בו התלמיד יוצר חשבון

איפוס סיסמה (ForgetPasswordEmail.xml)

# AtidPlus

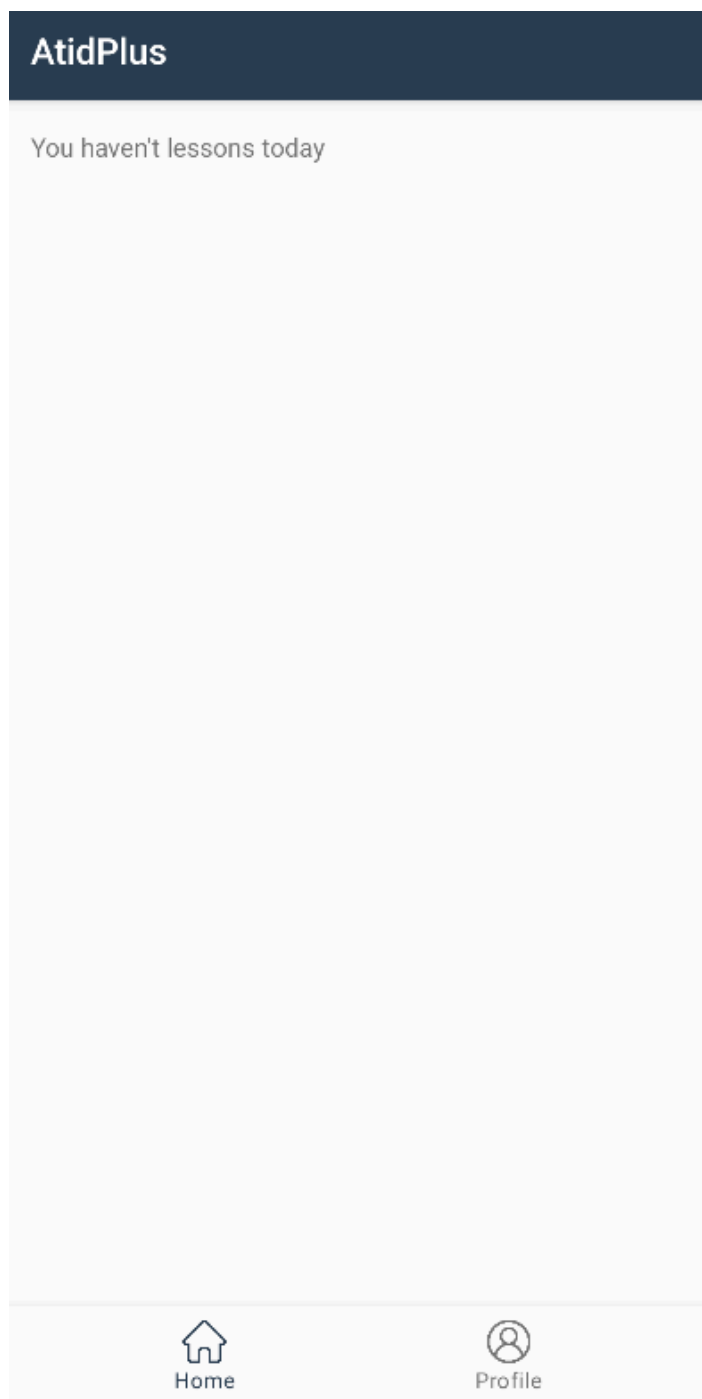
*Your Private Lessons*

Enter your Email

**RESET**

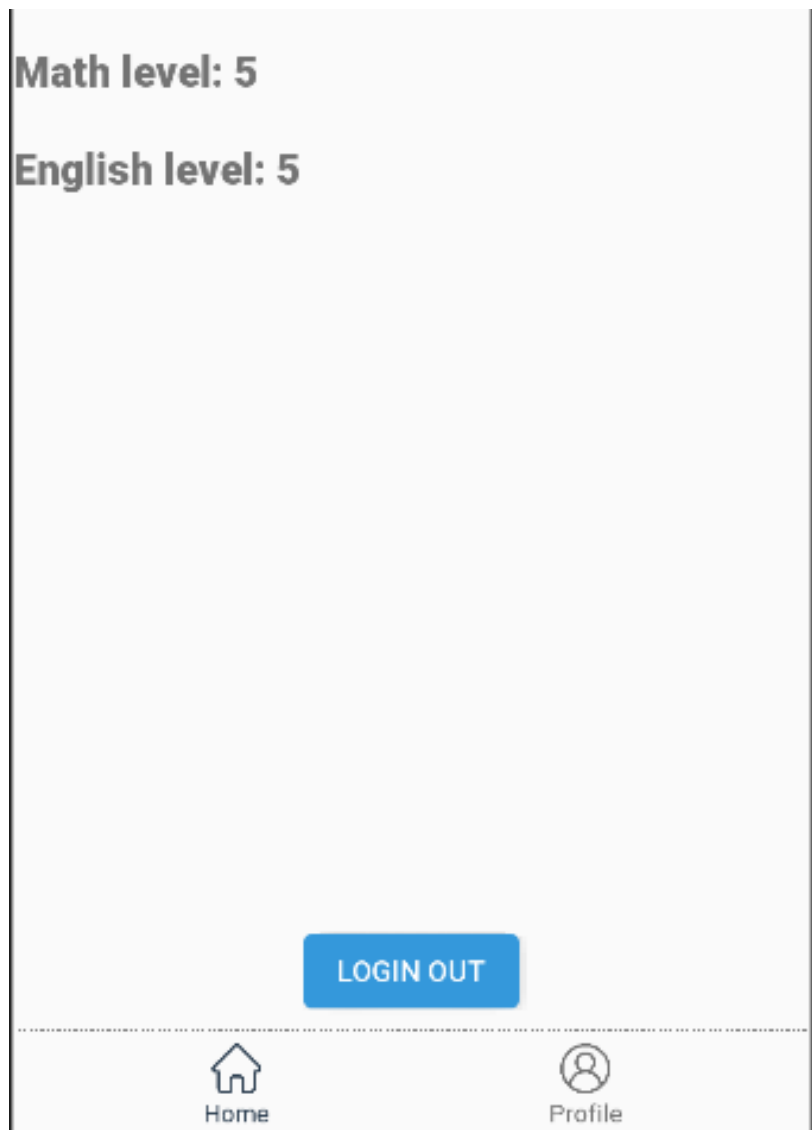
כאן המשתמש יכול להזין כתובת מייל ואם היא רשומה במסד הנתונים, היא תישלח אליו במייל לאיפוס הסיסמה

## מסך הבית של התלמיד (MainActivity.xml)



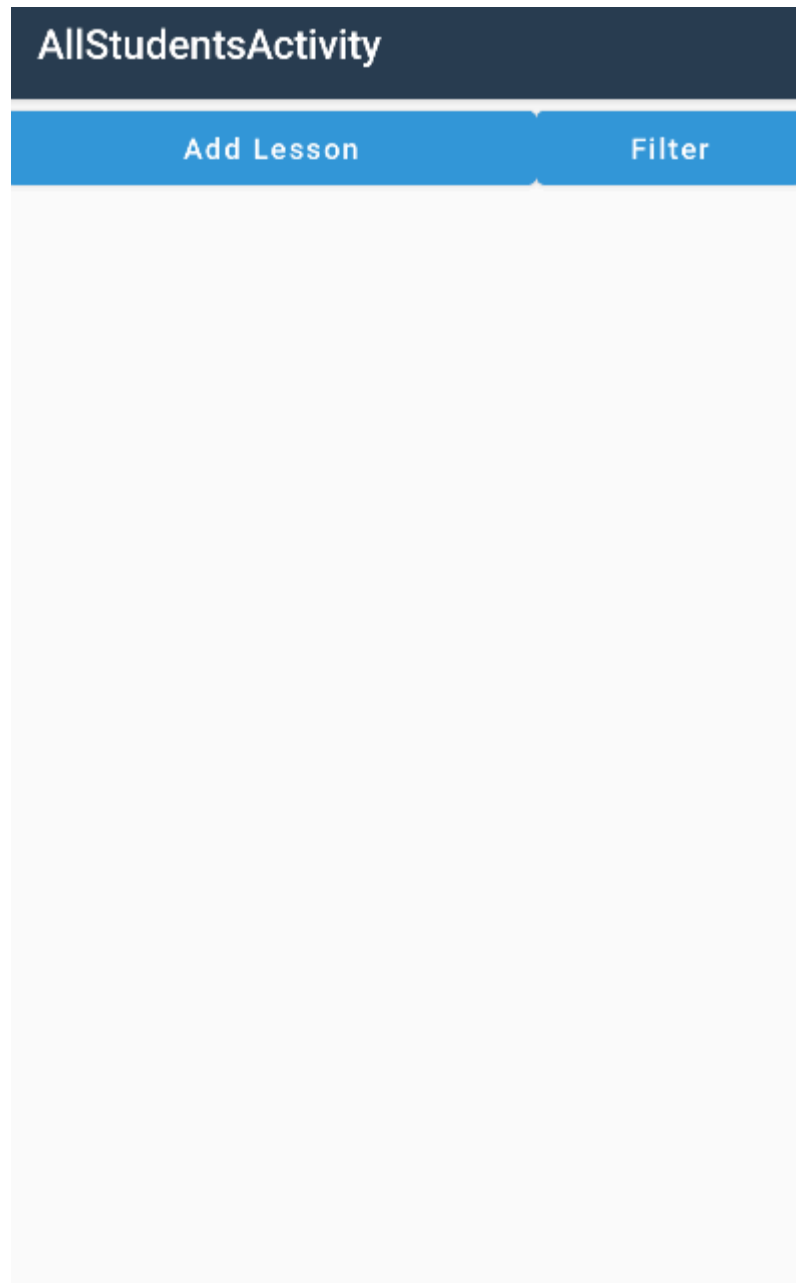
במסך זה, התלמיד יכול לראות אילו שיעורים נוספים יש לו היום, היכן ובאיזו שעה

## נתוני התלמיד (User.xml)



כאן התלמיד יכול לראות את רמת הידע שלו בשיעורים כגון מתמטיקה ואנגלית, ובמקרה של שינויים, הוא ייצור קשר עם האחראי על תגבורים כדי לשנות את המידע.

## מסך רשימת תלמידים למנהלים (Students\_List.xml)



כאן המנהל או האחראי לשיעורים נוספים יכולים לראות את רשימת התלמידים. ועל ידי לחיצה על תלמיד, הוא יכול להגדיר מידע על רמת ההשכלה שלו

## מסך הגדרות רמת הידע (SetPoints.xml)

School: Kfar Galim

Math

English

Cancel

Change

במסך זה, המנהל יכול להגדיר את רמת הידע של התלמידים, ובמקרה זה לשנות אותם

**:ListFilet (FilterList.xml**

☒ School

☐ Math

☐ English

Search

במסך זה אני יכול לסנן תלמידים לפי פרמטרים שצוינו

מסך הגדרת שיעור (AddLessons.xml)

☒ Additional Info

☐ Math

☐ English

Set Lesson

בשיעור זה אני מציין מי יקבל את השיעור כמנהל ומוסיף מידע על המקום ועל רואה החשבון



## הרשאות – פירוט ההרשאות הנדרשות

*AdminPermission = true/false* (בפירבייס)

### דרישות מיוחדות ומגבלות (מיקרופון, מצלמה, תקשורת וכו', זכרון פנוי...)

לאפליקציה יש פונקציונליות שונה בהתאם למי שמשתמש בה. לשם כך, ב-firebase, אני משתמש בפרמטר Admin Permission. אם הפרמטר הוא "שקר" אז זה תלמיד. אם "נכון" אז זה מנהל או אדם שאחראי לשיעורים נוספים

### דרישות תוכנה/חומרה

#### **גירסת Android מינימלית**

Android 4.4.2 גירסת API 19

#### **מכשירים עליהם נבדקה המערכת (גרסת המכשיר/ים שעליהם הרצתם)**

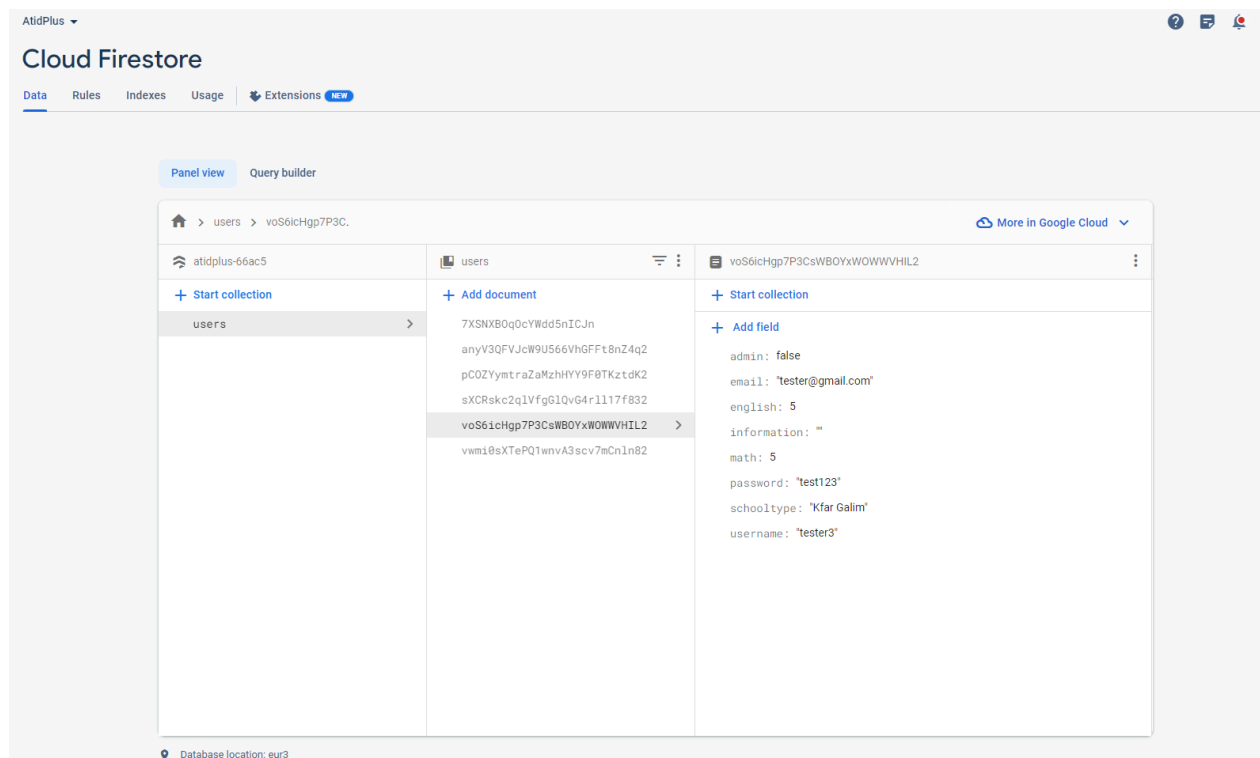
Xiaomi Mi A2 lite – android 9.0

אמולטור 27 Api – android 8.1 (Oreo\_X86\_Accelerate\_Android – Visual Studio)

# בסיס הנתונים

## שימוש ב-FirebaseFirestore

טבלה של משתמשים



## **שימוש ב- קבצים**

באפליקציה יש השימוש בקבצי נתונים.

בעת ההרשמה, המשתמש בוחר מאיזה בית ספר הוא מגיע ולאחריו מנהל המערכת קובע את רמת הידע שלו. זה נעשה יותר פרטיות. כל השינויים שנעשו על ידי האחראי לשיעורים נוספים נשמרים במסד הנתונים ומוצגים למשתמש בעתיד

# מדריך למפתח

## תרשים מחלקות

תרשים UML של כל ה-activities



תרשים UML של כל המחלקות שלכם כולל ירושות – אין

# Activities

## **MainActivity.cs – Activities**

פעילות מסך הכניסה.

דרכו נכנסים לאפליקציה במסך זה תוכלו ללחוץ על הרשמה, להיכנס לחשבון ולאפס את הסיסמה

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using AndroidX.AppCompat.App;
using App1.Admin;
using App1.EventListeners;
using App1.Helpers;
using App1.LoginAndRegistration;
using Firebase.Auth;
using System;

namespace App1
{
    [Activity(Label = "@string/app_name", Theme = "@style/AppTheme", MainLauncher = true)]
    public class Login : AppCompatActivity, View.OnClickListener
    {
        TextView signIn, forgotPass, adminBut;
        Button login;
        EditText email,password;
        ISharedPreferences sp;
        Intent intent;
        FirebaseAuth mAuth;
        TaskCompletionListeners taskCompletionListeners = new TaskCompletionListeners();
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.onCreate(savedInstanceState);
            Xamarin.Essentials.Platform.Init(this, savedInstanceState);
            SetContentView(Resource.Layout.LoginIn);
            InitViews();
        }
        public void OnSuccess(Java.Lang.Object result)
        {
            Toast.MakeText(this, "Succes", ToastLength.Short).Show();
            // StartActivity(typeof(MainActivity));
        }
        public void OnFailure(Java.Lang.Exception e)
        {
            Toast.MakeText(this, "Failed: " + e.Message, ToastLength.Short).Show();
        }
    }
}
```

```

    }

    public void InitViews (){
        login = findViewById<Button>(Resource.Id.Login);
        login.Click += LoginClick;
        email = findViewById<EditText>(Resource.Id.email_editText);
        password = findViewById<EditText>(Resource.Id.password_editText);
        email.SetOnClickListener(this);
        password.SetOnClickListener(this);
        signIn = findViewById<TextView>(Resource.Id.SignIn);
        signIn.SetOnClickListener(this);
        forgotPass = findViewById<TextView>(Resource.Id.forgotPassword);
        forgotPass.SetOnClickListener(this);
        sp = this.GetSharedPreferences("details", FileCreationMode.Private);
        intent = new Intent(this,typeof(Registration));
        mAuth = AppDataHelper.GetFirebaseAuth();
        adminBut = findViewById<TextView>(Resource.Id.forAdmin);
        adminBut.SetOnClickListener(this);
    }

    public void LoginClick(object sender, EventArgs e)
    {
        string emailText = email.Text;
        string passwordText = password.Text;

        if (!emailText.Contains("@"))
        {
            Toast.MakeText(this, "Please enter a valid email", ToastLength.Short).Show();
            return;
        }

        else if (passwordText.Length < 5)
        {
            Toast.MakeText(this, "Please enter a valid password", ToastLength.Short).Show();
            return;
        }

        mAuth.SignInWithEmailAndPassword(emailText,
passwordText).AddSuccessListener(taskCompletionListeners)
            .AddOnFailureListener(taskCompletionListeners);

        taskCompletionListeners.Succes += (succes, args) =>
        {
            var intent = new Intent(this, typeof(MainActivity));
            intent.PutExtra("userId", mAuth.CurrentUser.Uid);

            //поставить проверку как-то связав с файрбейс чтобы узнать, есть ли права админа или нет
            StartActivity(intent);

            Toast.MakeText(this, "Login Was successful", ToastLength.Short).Show();
        };

        taskCompletionListeners.Failure += (failure, args) =>
        {
            Toast.MakeText(this, "Login failed: " + args.Cause, ToastLength.Short).Show();
        };
    }

```

```

    }
    public void OnClick(View v)
    {
        if(v == signIn)
            StartActivityForResult(intent,0);

        if (v == forgotPass)
            StartActivity(typeof(ForgetPassword_FirstStage));
        if (v == adminBut)
            StartActivity(typeof(AllStudentsListActivity));
    }

    protected override void OnActivityResult(int requestCode, [GeneratedEnum] Result resultCode, Intent data)
    {
        base.OnActivityResult(requestCode, resultCode, data);

        if(requestCode == 0)
        {
            if(resultCode == Result.Ok)
            {
                Intent intent = new Intent();
                email.Text = intent.GetStringExtra("mail");
                password.Text = intent.GetStringExtra("pass");
            }
        }
    }
}

```

## Registration.cs – Activities

כאן המשתמש יכול להירשם

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Widget;
using App1.EventListeners;
using App1.Helpers;
using Firebase.Auth;
using Firebase.Firestore;
using Java.Util;
using System;
using System.Threading;

namespace App1
{
    [Activity(Label = "Activity1")]
    public class Registration : Activity
    {
        Button register;
        string user, email, password, confirmedPassword, school;
        EditText email_edit, password_edit, userName_edit, confirmPassword, schoolName_edit;
        ISharedPreferences sp;
        FirebaseAuth mAuth;
        TextView clickToLogin;
        TaskCompletionListeners taskCompletionListeners = new TaskCompletionListeners();
        FirebaseFirestore database;
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.onCreate(savedInstanceState);
            SetContentView(Resource.Layout.Registration);
            InitViews();
            SetClickLogics();
        }

        public void InitViews()
        {
            userName_edit = FindViewById<EditText>(Resource.Id.UserName);
            email_edit = FindViewById<EditText>(Resource.Id.email_editText);
            //school = FindViewById<EditText>(Resource.Id.yourSchool);
            database = AppDataHelper.GetFirestore();
            password_edit = FindViewById<EditText>(Resource.Id.password_editText);
            register = FindViewById<Button>(Resource.Id.Register);
            clickToLogin = FindViewById<TextView>(Resource.Id.returtToLogin);
            confirmPassword = FindViewById<EditText>(Resource.Id.confirmPas);
            mAuth = AppDataHelper.GetFirebaseAuth();
            schoolName_edit = FindViewById<EditText>(Resource.Id.schoolType);
        }
    }
}
```



```

    }
    public void SetClickLogics()
    {
        register.Click += RegistrationButton_ClickLogic;
        clickToLogin.Click += (sender, e) =>
        {
            BackToLogin();
        };
    }
    private void BackToLogin()
    {
        Intent intent = new Intent();
        intent.PutExtra("mail",email);
        intent.PutExtra("pass", password);
        SetResult(Result.Ok);
        Finish();
    }
    private void RegistrationButton_ClickLogic (object sender, EventArgs e)
    {
        user = userName_edit.Text;
        email = email_edit.Text;
        password = password_edit.Text;
        confirmedPassword = confirmPassword.Text;
        school = schoolName_edit.Text;

        //Validations

        if(user.Length < 4|| string.IsNullOrEmpty(user))
        {
            Toast.MakeText(this,"Please enter a valid name",ToastLength.Short).Show();
            return;
        }
        else if (!email.Contains("@"))
        {
            Toast.MakeText(this, "Please enter a valid email", ToastLength.Short).Show();
            return;
        }
        else if (password.Length<5||string.IsNullOrEmpty(password))
        {
            Toast.MakeText(this, "Please enter a valid password", ToastLength.Short).Show();
            return;
        }
        else if (password != confirmedPassword)
        {
            Toast.MakeText(this, "Password doesn't match", ToastLength.Short).Show();
            return;
        }
        else if(string.IsNullOrEmpty(school))
        {
            Toast.MakeText(this, "Incorrect School Name", ToastLength.Short).Show();
            return;
        }

        //Sending all information to database after validation
        mAuth.CreateUserWithEmailAndPassword(email,
        password).AddOnSuccessListener(this,taskCompletionListeners)

```

```

        .AddOnFailureListener(this,taskCompletionListeners);

taskCompletionListeners.Succes += (succes, args) =>
{
    HashMap userMap = new HashMap();
    userMap.Put("email", email);
    userMap.Put("username", user);
    userMap.Put("password", password);
    userMap.Put("schooltype", school);
    userMap.Put("math", 3);
    userMap.Put("english", 3);
    userMap.Put("math", 3);
    userMap.Put("information", "");
    userMap.Put("admin", false);
    DocumentReference userReference = database.Collection("users").Document(user);
    Intent.PutExtra("userName", userReference.Id);
    userReference.Set(userMap);
    Toast.MakeText(this, "Registration was Successfull, confirm email", ToastLength.Long).Show();
};

taskCompletionListeners.Failure += (failure, args) =>
{
    Toast.MakeText(this, "Registration Failed" + args.Cause, ToastLength.Short).Show();
};

Thread.Sleep(5000);
BackToLogin();
}
}
}

```

## **ForgetPassword\_FirstStage.cs – Activities**

כאן המשתמש יכול לאפס את הסיסמה. ישלח אליו מייל עם איפוס סיסמה.

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using AndroidX.AppCompat.App;
using App1.Helpers;
using Firebase.Auth;
using Firebase.Firestore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;

namespace App1.LoginAndRegistration
{
    [Activity(Label = "Activity1")]
    public class ForgetPassword_FirstStage : AppCompatActivity
    {
        Button reset;
        EditText email_editText;
        FirebaseAuth mAuth;
        FirebaseFirestore database;
        protected override void onCreate(Bundle savedInstanceState)
        {
            base.onCreate(savedInstanceState);
            SetContentView(Resource.Layout.ResetingPasswordEmail);
            InitViews();
        }

        public void InitViews()
        {
            reset = FindViewById<Button>(Resource.Id.sendVerification);
            email_editText = FindViewById<EditText>(Resource.Id.email_forReseting);
            mAuth = AppDataHelper.GetFirebaseAuth();
            reset.Click += Reset_Click;
            database = AppDataHelper.GetFirestore();
        }

        private void Reset_Click(object sender, EventArgs e)
        {
            string email = email_editText.Text;

            if (string.IsNullOrEmpty(email))
            {
                Toast.MakeText(this, "Email's field is empty", ToastLength.Short).Show();
                return;
            }
        }
    }
}
```

```
}  
else if (!email.Contains("@"))  
{  
    Toast.MakeText(this, "Please enter a valid email", ToastLength.Short).Show();  
    return;  
}  
  
 mAuth.SendPasswordResetEmailAsync(email);  
 Toast.MakeText(this, "Email Confirmation was sended", ToastLength.Short).Show();  
 Thread.Sleep(4000);  
 StartActivity(typeof(Login));  
}  
}  
}
```

## MainActivity.cs – Activities

כאן המשתמש יכול לראות אילו  
שיעורים זמינים היום

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Util;
using Android.Views;
using Android.Widget;
using AndroidX.AppCompat.App;
using App1.Helpers;
using App1.MainActivities;
using Firebase.Auth;
using Firebase.Database;
using Firebase.Firestore;
using Google.Android.Material.BottomNavigation;
using Java.Util;
using System;
using IEventListener = Java.Util.IEventListener;

namespace App1
{
    [Activity(Label = "@string/app_name", Theme = "@style/AppTheme", MainLauncher = false)]
    public class MainActivity : AppCompatActivity, BottomNavigationView.IOnNavigationItemSelectedListener,
        Firebase.Firestore.IEventListener
    {
        TextView textMessage;
        FirebaseFirestore database;
        EditText editText;
        Button button;
        ISharedPreferences sp;
        CollectionReference cf;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.activity_main);
            InitViews();
        }

        public void InitViews()
        {
            BottomNavigationView navigation = FindViewById<BottomNavigationView>(Resource.Id.navigation);
            navigation.SetNavigationItemSelectedListener(this);
            textMessage = FindViewById<TextView>(Resource.Id.message);
            database = AppDataHelper.GetFirestore();
        }
    }
}
```

```

//AppDataHelper appDataHelper = new AppDataHelper();
//appDataHelper.AddCollectionSnapshotListener(this, "users");
var MyId = Intent.GetStringExtra("userId");
database.Collection("users").Document(MyId).AddSnapshotListener(this);
}

```

```

public bool AddNewDocument(string str)
{
    try {
        HashMap hashMap = new HashMap();
        hashMap.Put("value", str);
        DocumentReference document = database.Collection("TestColection").Document();
        document.Set(hashMap);

        Toast.MakeText(this, "true", ToastLength.Short);

    } catch (Exception ex)
    {
        return false;
    }
    return true;
}

```

```

public override void OnRequestPermissionsResult(int requestCode, string[] permissions, [GeneratedEnum]
Android.Content.PM.Permission[] grantResults)
{
    Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

    base.OnRequestPermissionsResult(requestCode, permissions, grantResults);
}
public bool OnNavigationItemSelected(IMenuItem item)
{
    switch (item.ItemId)
    {
        //case Resource.Id.navigation_home:
        //var intent = new Intent(this, typeof(MainActivity));
        //StartActivity(intent);
        //return true;

        case Resource.Id.profile_icon:
            var intent2 = new Intent(this, typeof(User));
            StartActivity(intent2);
            return true;
    }
    return false;
}

```

```
public void OnEvent(Java.Lang.Object obj, FirebaseFirestoreException error)
{
    Console.WriteLine($"obj -> {obj}");
    DocumentSnapshot snapshot = (DocumentSnapshot)obj;
    var x = snapshot.Get("username");
    Console.WriteLine(x);
}
}
```

## **User.cs – Activities**

כאן התלמיד יכול לראות איזו רמת ידע יש לו ולהתנתק מהחשבון

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using AndroidX.AppCompat.App;
using Firebase.Firestore.Model;
using Firebase.Firestore;
using Google.Android.Material.BottomNavigation;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Java.Lang;

namespace App1.MainActivities
{
    [Activity(Label = "Activity1")]
    public class User : AppCompatActivity, BottomNavigationView.IOnNavigationItemSelectedListener
    {
        EditText math, english;
        FirebaseFirestore database;
        Button editButton;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.User);

            BottomNavigationView navigation = FindViewById<BottomNavigationView>(Resource.Id.navigation);
            navigation.SetOnNavigationItemSelectedListener(this);
        }
        public void InitViews()
        {
        }

    }

    public override void OnRequestPermissionsResult(int requestCode, string[] permissions, [GeneratedEnum]
    Android.Content.PM.Permission[] grantResults)
    {
        Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

        base.OnRequestPermissionsResult(requestCode, permissions, grantResults);
    }
}
```



```
public bool OnNavigationItemSelected(IMenuItem item)
{
    switch (item.ItemId)
    {
        case Resource.Id.navigation_home:
            var intent = new Intent(this, typeof(MainActivity));
            StartActivity(intent);
            return true;

            //case Resource.Id.profile_icon:
            //    var intent2 = new Intent(this, typeof(User));
            //    StartActivity(intent2);
            //    return true;
    }
    return false;
}
```

## **AllStudentsListActivity.cs – Activities**

כאן האדם האחראי יכול לראות את רשימת התלמידים ואת רמת הידע שלהם ולשנות אותה על ידי לחיצה

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using AndroidX.AppCompat.App;
using App1.Helpers;
using Firebase.Firestore;
using System;
using System.Collections.Generic;
using static Android.Widget.AdapterView;

namespace App1.Admin
{
    [Activity(Label = "AllStudentsActivity")]
    public class AllStudentsListActivity : AppCompatActivity, View.OnClickListener, IEventListener,
        ListView.IOnItemLongClickListener, IOnItemClickListener
    {
        AppDataHelper datahelper;
        Button btnAddLesson, btnFilter;
        ListView lvStudents;
        List<Student> lstStudents;
        StudentAdapter studentsAdapter;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.Students_List);
        }

        public void InitObject()
        {
            datahelper = new AppDataHelper();
            datahelper.AddCollectionSnapshotListener(this, "users");
        }

        public void InitViews()
        {
            btnAddLesson = FindViewById<Button>(Resource.Id.btnAddLesson);
```

```

        btnFilter = findViewById<Button>(Resource.Id.btnFilter);
        lvStudents = findViewById<ListView>(Resource.Id.lvStudents);
        btnAddLesson.setOnClickListener(this);
        btnFilter.setOnClickListener(this);
        lvStudents.setOnItemClickListener = this;
        lvStudents.OnItemLongClickListener = this;
    }

    public void OnEvent(Java.Lang.Object obj, FirebaseFirestoreException error)
    {

        GetStudents((QuerySnapshot)obj);
    }

    private void GetStudents(QuerySnapshot snapshot)
    {
        lstStudents = new List<Student>();

        foreach(DocumentSnapshot item in snapshot.Documents)
        {
            Student student = new Student
            {
                School = item.Get("schoolType").ToString(),
                MathLevel = int.Parse(item.Get("math").ToString()),
                EnglishLevel = int.Parse(item.Get("english").ToString()),
            };
            lstStudents.Add(student);
        }

        studentsAdapter = new StudentAdapter(this, lstStudents, false);
        lvStudents.Adapter=studentsAdapter;

    }

    public void OnItemClick(AdapterView parent, View view, int position, long id)
    {
        throw new NotImplementedException();
    }

    public bool OnItemLongClick(AdapterView parent, View view, int position, long id)
    {
        throw new NotImplementedException();
    }

    public void OnClick(View v)
    {
        }
    }
}

```

## **FilterListActivity.cs – Activities**

כאן המורה יכול לסנן את הרשימה לפי קריטריונים כמו מתמטיקה ואנגלית

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using AndroidX.AppCompat.App;
using App1.Helpers;
using Firebase.Firestore;
using System;
using System.Collections.Generic;
using static Android.Widget.AdapterView;

namespace App1.Admin
{
    [Activity(Label = "AllStudentsActivity")]
    public class AllStudentsListActivity : AppCompatActivity, View.OnClickListener, IEventListener,
        ListView.IOnItemLongClickListener, IOnItemClickListener
    {
        AppDataHelper datahelper;
        Button btnAddLesson, btnFilter;
        ListView lvStudents;
        List<Student> lstStudents;
        StudentAdapter studentsAdapter;

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);
            SetContentView(Resource.Layout.Students_List);
        }

        public void InitObject()
        {
            datahelper = new AppDataHelper();
            datahelper.AddCollectionSnapshotListener(this, "users");
        }

        public void InitViews()
        {
            btnAddLesson = FindViewById<Button>(Resource.Id.btnAddLesson);
            btnFilter = FindViewById<Button>(Resource.Id.btnFilter);
            lvStudents = FindViewById<ListView>(Resource.Id.lvStudents);
            btnAddLesson.SetOnClickListener(this);
            btnFilter.SetOnClickListener(this);
        }
    }
}
```

```

        lvStudents.OnItemClickListener = this;
        lvStudents.OnItemLongClickListener = this;
    }

    public void OnEvent(Java.Lang.Object obj, FirebaseFirestoreException error)
    {

        GetStudents((QuerySnapshot)obj);
    }

    private void GetStudents(QuerySnapshot snapshot)
    {
        lstStudents = new List<Student>();

        foreach(DocumentSnapshot item in snapshot.Documents)
        {
            Student student = new Student
            {
                School = item.Get("schoolType").ToString(),
                MathLevel = int.Parse(item.Get("math").ToString()),
                EnglishLevel = int.Parse(item.Get("english").ToString()),
            };
            lstStudents.Add(student);
        }

        studentsAdapter = new StudentAdapter(this, lstStudents, false);
        lvStudents.Adapter=studentsAdapter;

    }

    public void OnItemClick(AdapterView parent, View view, int position, long id)
    {
        throw new NotImplementedException();
    }

    public bool OnItemLongClick(AdapterView parent, View view, int position, long id)
    {
        throw new NotImplementedException();
    }

    public void OnClick(View v)
    {

    }

}
}

```

## **AddLessonActivity.cs – Activities**

כאן המורה יכול להעביר שיעור לתלמידים בהתאם לרמת הידע שלהם ולהוסיף מידע על המקום, המורה והזמן

# Adapters

למתאם יש רשימה של תלמידים. זה לוקח את כל התלמידים מהרשימה ומציג אותם לפי השדות הבאים:  
עבור כל מתכון, זה מראה את רמת המתמטיקה והאנגלית שלו. אז בעצם המתאם יוצר רשימה של תלמידים שמציגה עבור כל תלמיד נתונים מסוימים שנשלפו ממסד הנתונים.  
פעולת `GetView` מציגה את מחרוזת המתכון בהתאם לנתונים שלה. תפקידו של המתאם הוא לתאם בין מסד הנתונים לרשימה.

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using App1.Admin;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace App1
{
    internal class StudentAdapter : BaseAdapter
    {
        readonly Context context;
        readonly List<Student> students;
        readonly bool filter;

        public StudentAdapter(Context context, List<Student> students, bool filter)
        {
            this.context = context;
            this.students = students;
            this.filter = filter;
        }

        public override Java.Lang.Object GetItem(int position)
        {
            return position;
        }

        public override long GetItemId(int position)
        {
            return position;
        }

        public override View GetView(int position, View convertView, ViewGroup parent)
```

```

{
    LayoutInflater inflater;
    if (filter)
        inflater = ((FilterListActivity)context).LayoutInflater;
    else
        inflater = ((AllStudentsListActivity)context).LayoutInflater;

    View view = inflater.inflate(Resource.Layout.StudentsListRow, parent, false);
    TextView tvName = view.findViewById<TextView>(Resource.Id.tvName);
    TextView mthLevel = view.findViewById<TextView>(Resource.Id.mthLevel);
    TextView englishLevel = view.findViewById<TextView>(Resource.Id.englLevel);
    TextView tvSchool = view.findViewById<TextView>(Resource.Id.tvSchool);
    Student student = students[position];

    if (student != null)
    {
        tvName.Text = student.Name;
        mthLevel.Text = student.MathLevel.ToString();
        englishLevel.Text = student.EnglishLevel.ToString();
        tvSchool.Text = student.School;
    }
    return view;
}

//Fill in count here, currently 0
public override int Count
{
    get
    {
        return 0;
    }
}

}

internal class StudentAdapterViewHolder : Java.Lang.Object
{
    //Your adapter views to re-use
    //public TextView Title { get; set; }
}
}

```



## **Services**

# מחלקות

## **Classes – Student.cs**

כלס לנוחות בו נשמרים הפרמטרים העיקריים של התלמיד. מתמטיקה, אנגלית, שמו ובית הספר

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace App1
{
    public class Student
    {
        public string Name { get; set; }
        public string School { get; set; }
        public int MathLevel { get; set; }
        public int EnglishLevel { get; set; }

        public Student() {
            this.Name = string.Empty;
            this.School = string.Empty;
            this.EnglishLevel = 0;
            this.MathLevel = 0;
        }

        public Student (string id, string name, string school, int mathLevel, int englishLevel)
        {
            this.Name = name;
            this.School = school;
            this.MathLevel = mathLevel;
            this.EnglishLevel = englishLevel;
        }
    }
}
```

## **Classes – AppDataHelper.cs**

הגדרות מסד הנתונים ופונקציות העזר מאוחסנות כאן

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Firebase;
using Firebase.Firestore;
using Firebase.Auth;
using Firebase.Database;
using Java.Nio.FileNio;
using Java.Util;
using Android.Gms.Tasks;

namespace App1.Helpers
{
    public class AppDataHelper
    {
        private readonly FirebaseFirestore firestore;

        public static FirebaseAuth GetFirebaseAuth()
        {
            var app = FirebaseApp.InitializeApp(Application.Context);
            FirebaseAuth mAuth;

            if(app == null)
            {
                var options = new FirebaseOptions.Builder()
                    .SetProjectId("atidplus-66ac5")
                    .SetApplicationId("atidplus-66ac5")
                    .SetApiKey("AIzaSyBrWkxWCRQ-NtC62xmIKP2R_TWMY9a7ITc")
                    .SetDatabaseUrl("atidplus-66ac5.firebaseio.com")
                    .SetStorageBucket("atidplus-66ac5.appspot.com")
                    .Build();

                app = FirebaseApp.InitializeApp(Application.Context,options);
                mAuth = FirebaseAuth.Instance;
            }
            else
            {
                mAuth = FirebaseAuth.Instance;
            }
            return mAuth;
        }
    }
}
```

```

public static FirebaseFirestore GetFirestore()
{
    var app = FirebaseApp.initializeApp(Application.Context);
    FirebaseFirestore database;

    if (app == null)
    {
        var options = new FirebaseOptions.Builder()
            .SetProjectId("atidplus-66ac5")
            .SetApplicationId("atidplus-66ac5")
            .SetApiKey("AIzaSyBrWkxWCRQ-NtC62xmIKP2R_TWMY9a7ITc")
            .SetDatabaseUrl("atidplus-66ac5.firebaseio.com")
            .SetStorageBucket("atidplus-66ac5.appspot.com")
            .Build();

        app = FirebaseApp.initializeApp(Application.Context, options);
        database = FirebaseFirestore.Instance;
    }
    else
    {
        database = FirebaseFirestore.Instance;
    }
    return database;
}

public void AddCollectionSnapshotListener(Activity activity, string cName)
{
    firestore.Collection(cName).AddSnapshotListener((Firebase.Firestore.IEventListener)activity);
}

public Task SetFsDocument(string cName, string id, HashMap hmFields, out string newId)
{
    DocumentReference dr;
    if (id == string.Empty)
        dr = firestore.Collection(cName).Document();
    else
        dr = firestore.Collection(cName).Document(id);
    newId = dr.Id;

    return dr.Set(hmFields);
}

public Task GetEqualCollection(string fName, string fValue)
{
    return firestore.Collection("users").WhereEqualTo(fName, fValue).Get();
}

public Task GetEqualSubject(string fName, int fValue)
{
    return firestore.Collection("users").WhereEqualTo(fName, fValue).Get();
}

}
}

```

# Resources

## Layouts

Activity\_main - המסך הראשי, הוא מציג את השיעורים

AddLesson - מסך ניהול. כאן האדם האחראי יכול לקבוע את השיעורים

Filter\_layout - לְסַנֵּן. כאן אתה יכול למיין תלמידים לפי רמת הידע של מתמטיקה ואנגלית

LoginIn אתה יכול להתחבר כאן

Registration- הִרְשָׁמָה. כאן אתה יכול ליצור חשבון

ResettingPasswordEmail - כאן תוכל לאפס את הסיסמה שלך על ידי הזנת כתובת הדוא"ל שלך.

SetPoints - במסך זה, האחראי קובע את רמת הידע של התלמיד

Students\_List - רשימה של כל התלמידים. יוצג כאן

StudentsListRow - כך יוצגו האנשים ברשימה, שמם ורמת הידע שלהם

User - אזור אישי. כאן התלמיד יכול להסתכל על רמת הידע שלו ולהתנתק מהחשבון

## Menus

- Navigation.xml

תפריט שיוצג בתחתית מסך התלמיד. הוא יכול לעבור מהחשבון האישי שלו לעמוד הראשי ובחזרה

## Drawables

Home.png - התמונה של המסך הראשי שמוצגת בתפריט

User.png - תמונה של חשבון אישי שמוצג בתפריט

## סיכום אישי – רפלקציה

נהניתי לעשות את הפרויקט הזה. היה לי מאוד קשה לעבוד עם מסדי נתונים ולגרום להכל לעבוד בלי באגים. ניסיתי לעשות אפליקציה שמתאימה לתלמידי תכנית עתיד פלוס. כדי שיהיה להם קל יותר ללמוד. אני מקווה שבעתיד אצליח להביא את האפליקציה הזו לרמה מסחרית. היה קשה להתמודד עם הבאז הזה וזכויות הגישה. במשך זמן רב מאוד לא הצלחתי למצוא מידע באינטרנט הקשור ל-`firebase firestore` ו-`xamarin android`. היה מעניין גם לבנות את מבנה האפליקציה. אהבתי שהעבודה הזו היא כמו פרויקט רציני שאני עושה לבד. אני גם רוצה להזכיר את `xamarin`. אחת מסביבות הפיתוח הגרועות ביותר של אפליקציות אנדרואיד. זה היה הניסיון הראשון והאחרון שלי עם המסגרת הזו. אם הייתה הזדמנות לעשות את זה ב-`android studio`, הייתי משתמש בה. בתחילת התהליך היה מעניין, סביבת פיתוח חדשה והכל חדש. אבל בסופו של דבר הייתי מאוד לא מרוצה, חבורה של באגים שנגרמו בגלל סביבת הפיתוח, אני כל הזמן צריך למחוק את התיקיות `bin` ו-`object` רק כי זה נחוץ והבאג לא יעבוד בלעדיו.