

# ATID PLUS

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

## 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, ומה השתמשתי ולמה החלטתי לעשות זאת. בנוסף, אצהר צילומי מסך הנ' משלצתי על ידי, והן מהאפליקציה עצמה.

### רקע לפרויקט

האפליקציה המשמשת כדلت וירטואלית ואישית, באפליקציה תוכלו לראות אילו שיעורים נוספים יש לכם שייהיו 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 ואני הוא ייראה, הגעתו לעיצוב ואני הוא יעבד מאפס. בנוסף, בדקתי גם איך נראית אפליקציית היום האלקטרוני הפופולרי ביותר של בית ספר חכם. בסופה של דבר פיתחתי יומן אלקטרוני, שהוא מאד נוח ופשוט

## **אתגרים מרכזיים**

### **סיבות לבחירת הפרויקט**

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

היום שלי קל לשימוש ונגיש יותר, וגם מוחבר לעולם הטלפונים המודרני. באופן אישי, כמה פעמים היה לי רצון במהלך הלימודים תהיה אפליקציה כזו שאפשרה לי לראות את לוח הזמנים שלי להיום.

### **CRCים שעליים עונה הפרויקט**

הפרויקט הופך את לוח הזמנים של שיעורים נוספים לזמן. היום הווירטואלי מקל על לחיצה וצפיה בפעולות נוספות בנוסף לכך את החיים לקלים ככל האפשר לתלמיד תיכון. הרি כולם והלכים לכל מקום עם הטלפון שלו, אבל עוד לא שמעתי על מישחו שהוא שולץ בכל מקום עם היום האלקטרוני שלו עם פעילויות נוספות. כך למעשה הפרויקט הופך את לוח הזמנים לזמן. בנוסף, לתלמיד בית ספר יש עניינים אישיים מסוימים בשגרה, וכך לא לשכוה איזה שיעור יש לו היום, הוא יכול פשוט לפתוח את הקשה שלו.

### **בעיה/דילמה בפיתוח הפרויקט**

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

### **פתרונות שנבחנו כדי לפתור את הבעיה**

היתה בעיה כיצד מורים או האחראים לשיעורים נוספים מודיעים לתלמידים על השיעור שלהם וקובעים עבורם לוח זמנים. לשם כך יצרתי זכויות גישה רק למנחים, כך שהם יכולים עצמם 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 המ██ים שהמשתמש רואה ברגע שהוא משתמש באפליקציה.

- **Activities** תכונות שנמצואו מאחוריו המ██ים. עברו כל מסך פרישה בפועל, מסך איפוס סיטה. יש קוד שימוש בנתונים שנקבעו על ידי המנהל (שיעורים). לדוגמה, בפעולות, אתה לוקח את המידע שהזונן למנהל, מאוחסן במסד הנתונים, ואז צזה התראה שיעור על מסך התלמיד.

**המ██ים העיקריים של תלמיד המ:** מסך התחברות, מסך הרשמה, מסך השיעורים של היום, מסך משמש בו הוא יכול לראות את רמת הלמידה שלו ובמקרה של שינויים יפנה למנהל לשינוי הנתונים

**מ██ים למנהל מערכת:** רשימה של כל התלמידים, מסנן תלמידים לפי בית ספר, רמת המתמטיקה והאנגלית שלהם ודרך להעביר שיעור לתלמידים. עברו כך מסך זה יש אקטיביטי. באקטיביטי יש את הפעולות השונות הקשורות ברגע שלוחצים על כפתורים או מכנים מידע. לדוגמה, כאשר אתה מגדר שיעור אתה לוקח את כל הנתונים שהזוננו וברגע שלוחצים על כפתור "אישור" מגדרים את השיעור ושולחים הודעה לחת-הקבוצה.

- **Intents** אפשרים מעבר בין מ██ים ואקטיביטים שונים על מנת להעביר מידע באפליקציה.

## – (שמירת מידע) DB

מסד נתונים הוא כלי המשמש לארגון נתונים במחשב לצורך אחסון ואחיזור בעת הצורך. בתוך המאגר נתונים מאוחסנים באוסףם בהם מאוחסנים נתונים תלמידים. לכן, נדרש לכל לכל האלמנטים המאוחסנים באותו מסד נתונים יש אותן תוכנות.

## המחלקות הראשיות שלנו:

**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

מבנה נתונים שלי:

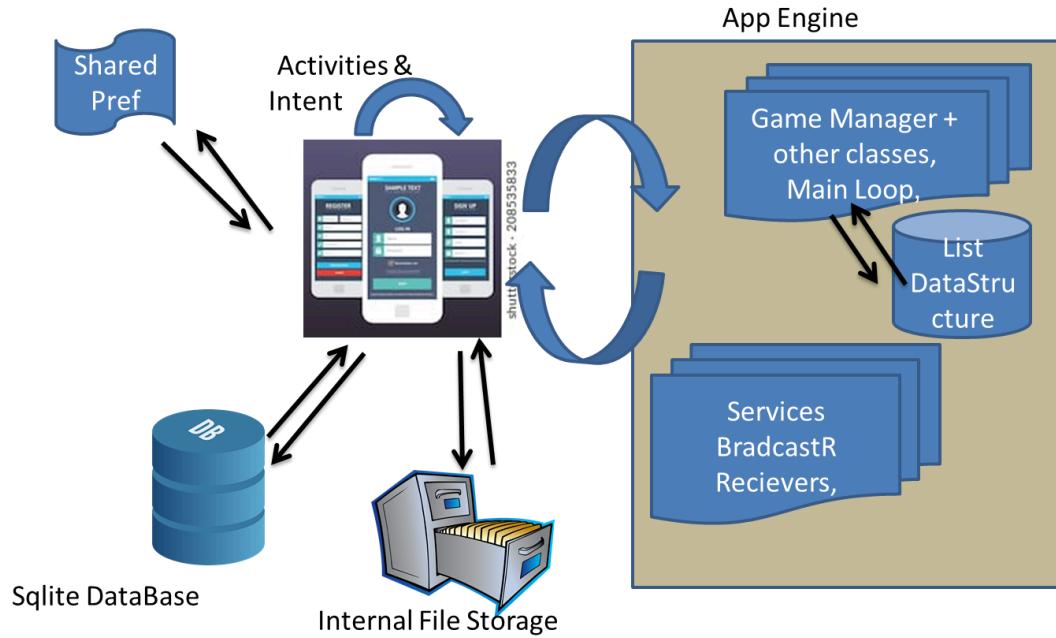
. אוסף זה מזוהה את כל המשתמשים ומידע על רמת ההדרכה שלהם בענין. Firebase Firestore - user collection

## מנגוני אנדרואיד

- Service – מאפשרים ליצור ולנהל תהליכי הקשרים למה שקרה ברקע האפליקציה, כמו למשל השמעת מזיקה רקע במהלך ריצת האפליקציה.

## لتצוגת רשימות – Adapters

אדפטור מתאם בין DB ל-`list` . כמובן הוא מתאים בין הרשימה המוצגת באחד המרכיבים לבין המידע מהבסיס נתונים שורצים להכנס לרשימה. השימושי רק באדפטר אחד: עבור משתמשים והכנסתם לרשימה

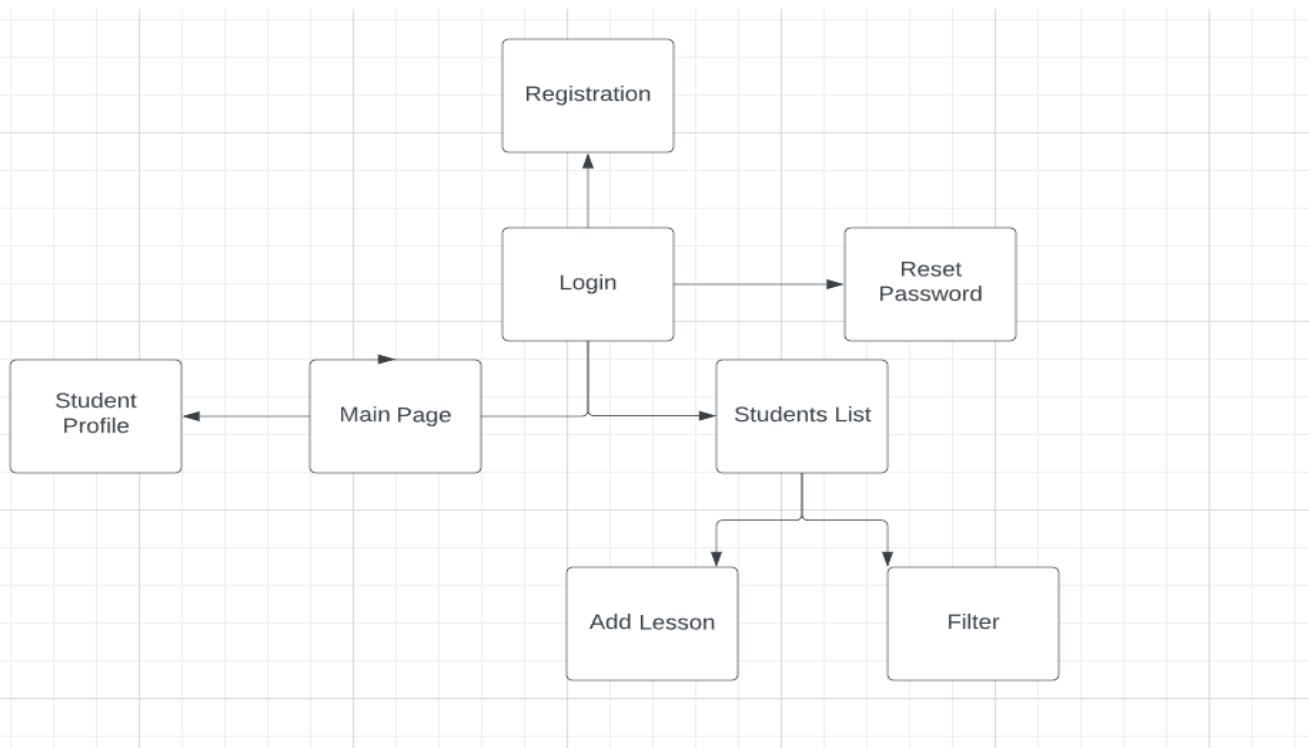


## מדריך למשתמש

### מטרת המערכת

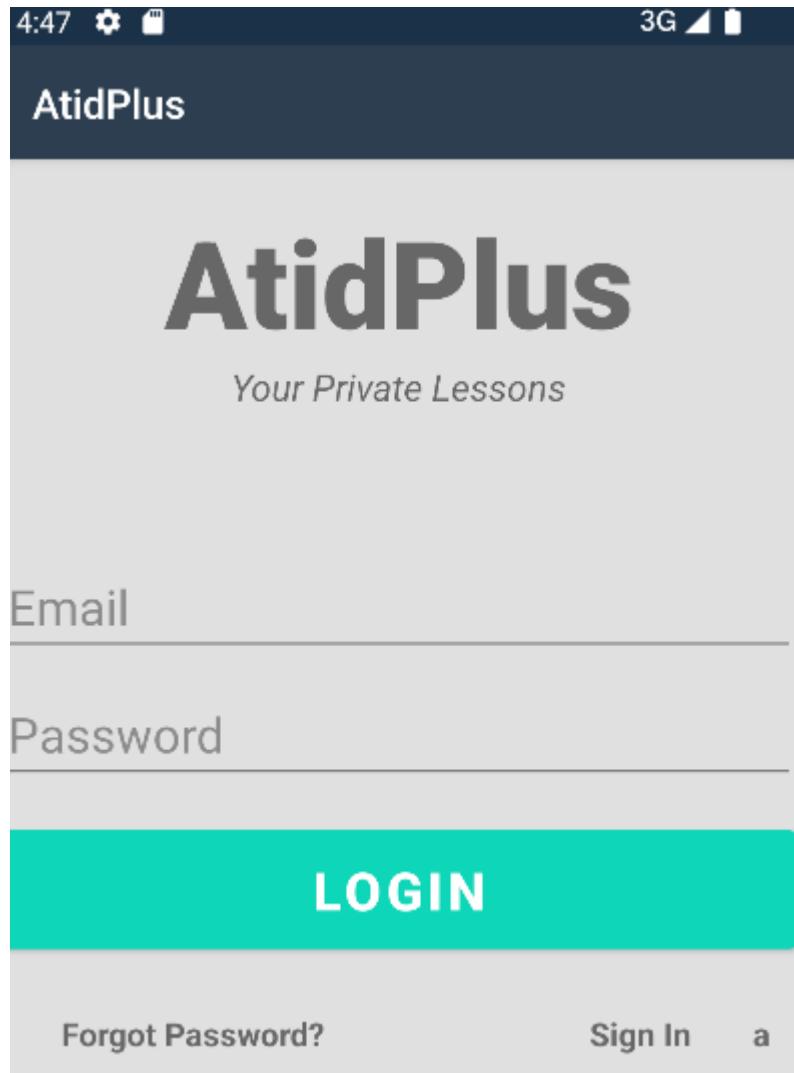
באמצעות האפליקציה יכולים המשתמשים להכנס למכשיר סמארטפון אישר אתلوح השיעורים הנוספים, וכך ליצור יומן אלקטרוני אישי של תgesories. האפליקציה עוזרת לך להציג בנותה ובקלות את השיעורים שלך בכל מקום ובכל זמן מבליל לשכוח אותם. לאפליקציה שני סוגי שיעורים. מתמטיקה ואנגלית. המנהל או האחראי על העברת שיעורים נוספים יכולים לחתם לך שיעורים בשני נושאים אלו.

## תרשים זרימה בין מסכים -



## תפועל המערכת

:Login in



תיאור מסך כניסה והיכן מוצג

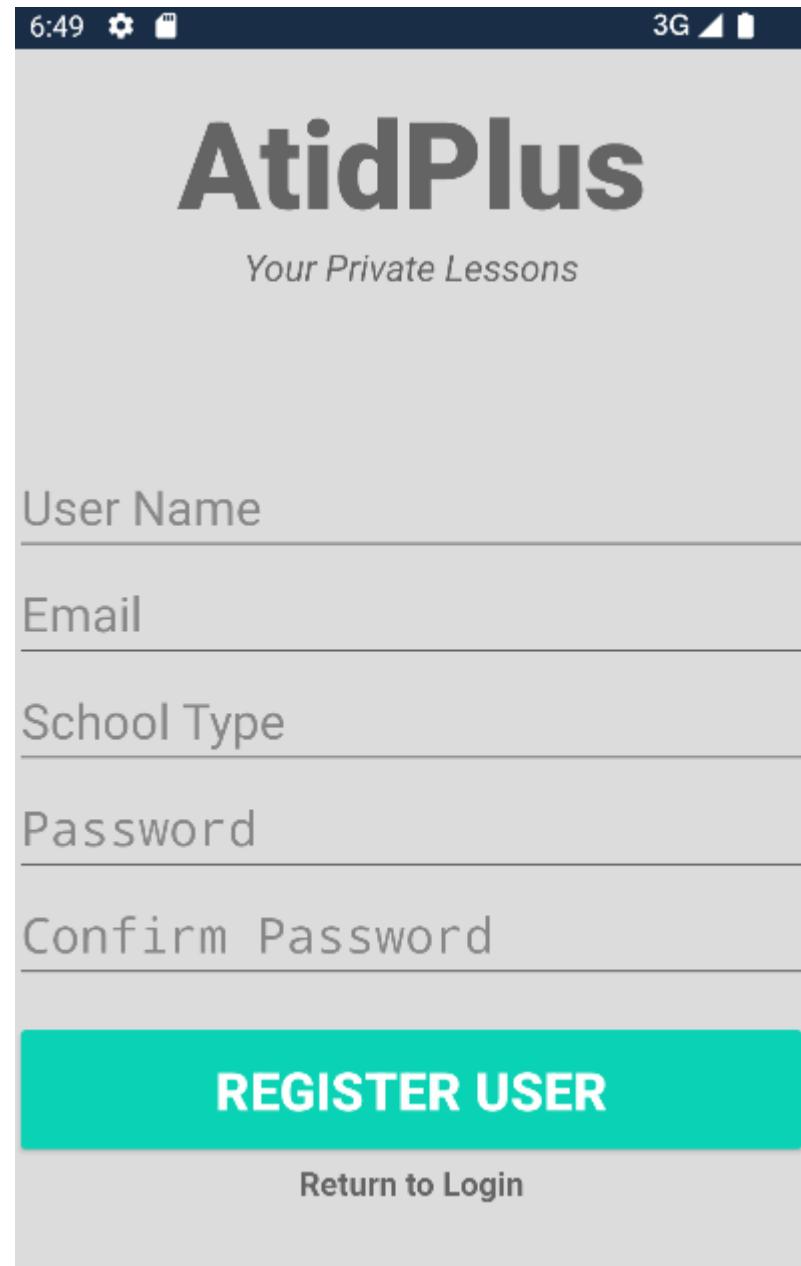
לאחר הכניסה, המרכיבים של המנהל והמשתמש יהיו שונים  
כניסה כולל בתוכו מעבר למסכים הבאים, וכן במסכים האלה כניסה מופיע:

מה שcoliום רואים:

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

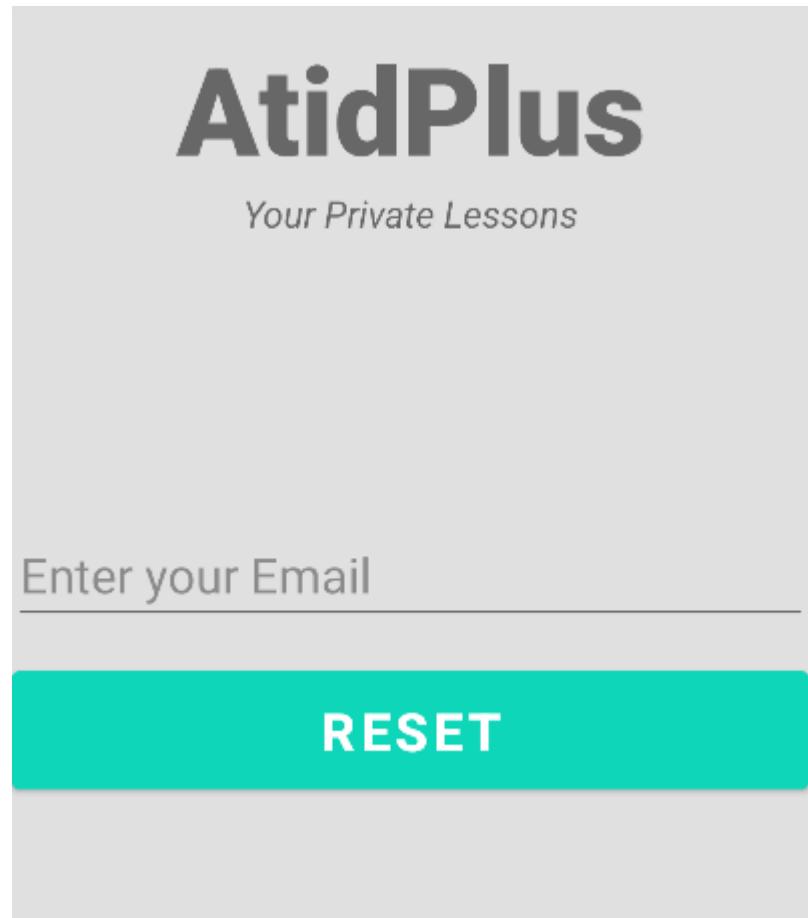
מקדים:

הרשמה(Registration.Xml)



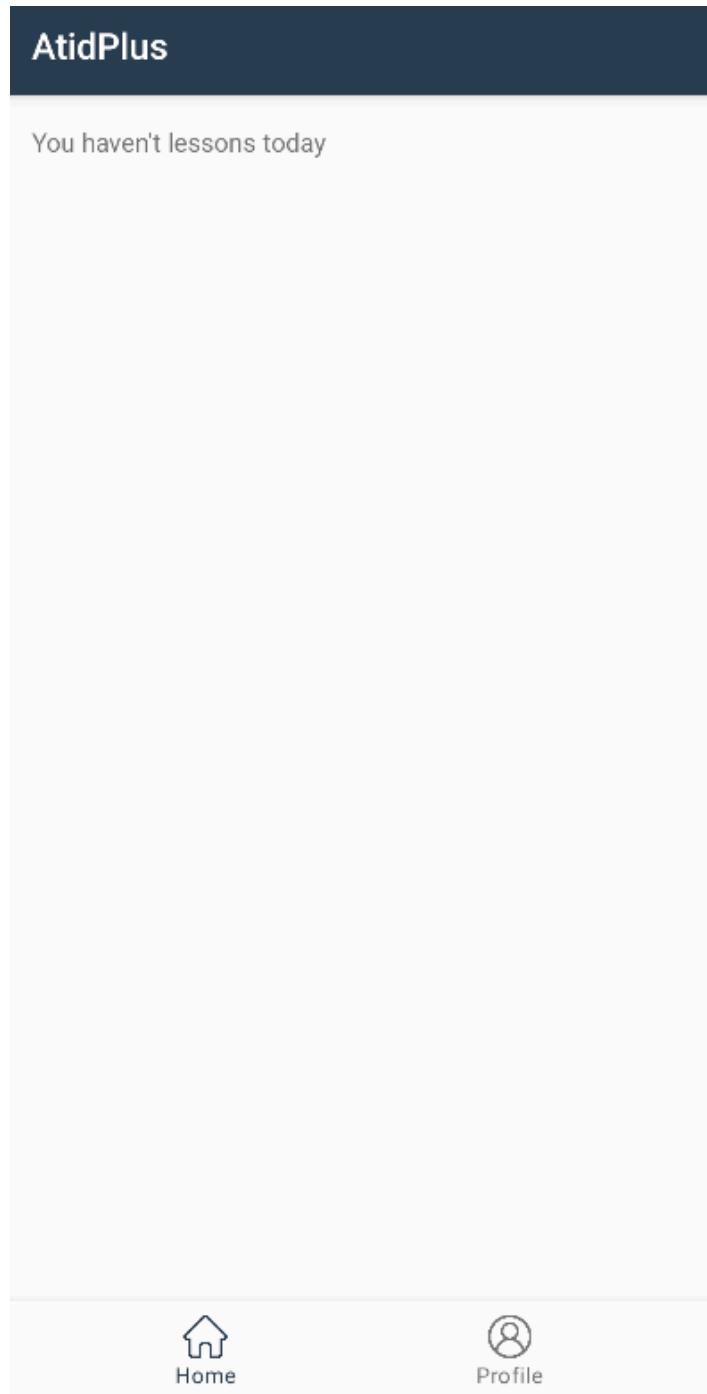
מסך רישום, בו התלמיד יוצר חשבון

**אייפוס סיסמה (ForgetPasswordEmail.xml)**



כאן המשתמש יכול להזין כתובת מייל ואם הוא רשומה בסיסד הנתונים, היא תישלח אליו מייל לאיפוס הסיסמה

**מסך הבית של התלמיד (MainActivity.xml)**



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

**נתוני התלמיד (User.xml)**

**Math level: 5**  
**English level: 5**

**LOGIN OUT**



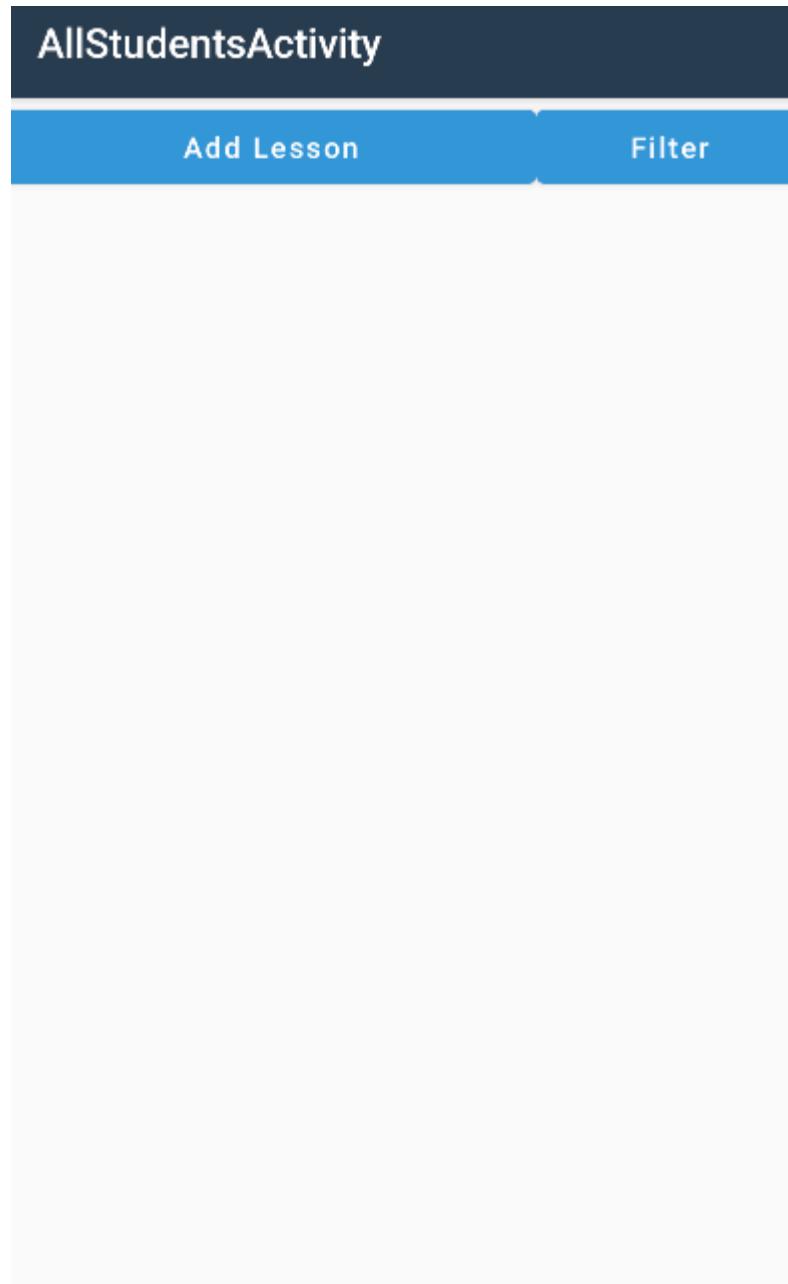
Home



Profile

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 להציג מידע על רמת ההשכלה שלו

**מסך הגדירות 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

The image shows a user interface for a mobile application. At the top, there is a header with the text ':ListFilet (FilterList.xml'. Below the header, there is a search bar. The search bar contains three filter options: 'School', 'Math', and 'English'. Each option is preceded by a radio button. The 'School' radio button is selected, indicated by a blue circle with a white dot. To the right of each option is a horizontal input field. Below the search bar is a large, solid blue button with the word 'Search' written in white. The entire interface is contained within a light gray rectangular frame.

במסך זה אני יכול לسانן תלמידים לפי פרמטרים שצווינו

**מסך הגדרת שיעור (AddLessons.xml)**

The screenshot displays a user interface for selecting lesson types. At the top, there is a header in Hebrew: "מסך הגדרת שיעור (AddLessons.xml)". Below this, there are three radio button options: "Additional Info" (selected), "Math", and "English". Each option has a horizontal input field to its right. At the bottom, there is a large blue button with the text "Set Lesson".

Lesson Type	Input Field
Additional Info	Selected
Math	Unselected
English	Unselected

בשיעור זה אני מצין מי יקבל את השיעור כמנהל ומוסיף מידע על המיקום ועל רזאה החשבונו

## **הרשאות – פירוט ההרשאות הנדרשות**

*AdminPermission = true/false(בפ"רבי"ס)*

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

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

### **דרישות תוכנה/חומרה**

#### **גירסת Android מינימלית**

API 19 גירסת Android 4.4.2

#### **מכשירים עליהם נבדקה המערכת (גירסת המכשיר/ים שעלייהם הרצתם)**

Xiaomi Mi A2 lite – android 9.0

(Visual Studio – Android\_Accelerate\_X86\_Oreo(android 8.1 – Api 27

## **בסיס הנתונים**

### **שימוש ב- Firestore**

**טבלה של משתמשים**

The screenshot shows the Google Cloud Firestore console interface. At the top, there's a navigation bar with 'AtidPlus' and other tabs like 'Data', 'Rules', 'Indexes', 'Usage', and 'Extensions'. Below the navigation is a toolbar with 'Panel view' and 'Query builder' buttons. The main area displays a 'users' collection under the 'atidplus-66ac5' database. A single document named 'voS6icHgp7P3CsWBOYxWOWWHIL2' is selected. The document contains the following fields and their values:

Field	Type	Value
admin	boolean	false
email	string	"tester@gmail.com"
english	number	5
information	string	" "
math	number	5
password	string	"test123"
schooltype	string	"Kfar Galim"
username	string	"tester3"

## **שימוש ב- קבצים**

באפליקציה יש השימוש בקבצי נתונים.

בעת ההרשמה, המשמש בוורاء מזיהה בית ספר הוא מגיע ולאחורי 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 מפרק הכנסה.

דרכו ננכדים לאפליקציה בפרק 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 FirebaseAuth;
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).AddOnSuccessListener(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, confirmPassword, 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;
        confirmPassword = 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 != confirmPassword)
        {
            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.ResettingPasswordEmail);
            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 יוכל 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.MainActivitys;
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;
        SharedPreferences 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.SetOnNavigationItemSelectedListener(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 מהמחשבון

```
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.MainActivity
{
    [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;
}
```

## לאחראים על שיעורים נוספים Activities

### **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.IOnClickListener, 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);
```



## **FilterListActivity.cs – Activities**

כאן המורה יכול לسانן את הרשימה לפי קритריונים 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.IOnClickListener, 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

למתקן יש רשימה של תלמידים. זה לוקח את כל התלמידים מהרשימה ומציג אותם לפי השדרות הבאים:  
עבור כל מתכוון, זה מראה את רמת המתמטיקה והאנגלית שלו. אז בעצם המתאים יוצר רשימה של תלמידים שמצויה עבור כל תלמיד נתוניים מסוימים שנשלפו מסד הנתונים.  
פעולה `GetView` מציגה את מחוوظ המתכוון בהתאם לננתוניים שלו. תפקידו של המתאים הוא בהתאם בין מסד הנתוניים 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 layoutInflater;
    if (filter)
        layoutInflater = ((FilterListActivity)context).LayoutInflater;
    else
        layoutInflater = ((AllStudentsListActivity)context).LayoutInflater;

    View view = layoutInflater.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.engLevel);
    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 העוזר 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 FirebaseAuth;
using FirebaseDatabase;
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

- המסך הראשי, הוא מציג את השיעורים Activity\_main

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

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

אתה יכול להתחבר כאן LoginIn

- הקישור. כאן אתה יכול ליצור חשבון Registration

- כביכול לאפס את הסיסמה שלך על ידי הונת כתובות הדוא"ל ResetingPasswordEmail

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

- רשימה של כל התלמידים. יוצג כאן Students\_List

- כך יוצגו האנשים ברשימה, שם ורמת הידע שלהם StudentsListRow

- איזור אישי. כאן התלמיד יכול להסתכל על רמת הידע שלו ולהתנק מהחשבון User

## Menus

- Navigation.xml

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

## Drawables

- התמונה של המסך הראשי שמוצגת בתפריט Home.png

- תמונה של חשבון אישי שמוצג בתפריט User.png

## **סיכום אישי – רפלקציה**

הניתי לעשות את הפרויקט הזה. היה לי מאד קשה לעבוד עם מסדי נתונים ולגרום להכל לעבוד בלי בגים. ניסיתי לעשות אפליקציה שמתאימה לתלמידי תיכון עתיד פלוס. כדי שיהיה להם כל יותר ללמידה.

אני מקווה שבעחיד אצליח להביא את האפליקציה זו לרמה מסחרית.

היה קשה להתמודד עם הבוא זהה וזכיות הגישה. במשך זמן רב מאוד לא הצליחי למצוא מידע באינטרנט הקשור ל-*xamarin android firebase firestore*.

היה מעניין גם לבנות את מבנה האפליקציה. אהבתך שהעבודה הזו היא כמו פרויקט רציני שאני עושה בלבד. אני גם רוצה להזכיר את *xamarin android*. אחת מסביבות הפיתוח הנgrossות ביותר של אפליקציות אנדרואיד. זה היה הניסיון הראשון והאחרון שלי עם המסגרת הזו. אם הייתה הזדמנות לעשות את זה ב-*studio*, הייתי משתמש בה. בתחילת התהליך היה מעוניין, סביבת פיתוח חדשה והכל חדש. אבל בסופו של דבר הייתה מאוד לא מרווחה, חמורה של בגים שנגרמו בגל סביבת הפיתוח, אני כל הזמן צריך למחוק את התיקיות *bin* ו-*object* רק כי זה נחוץ והבאג לא עובד בלוויו.