

פרויקט EduCare

עבודת גמר תכנון ותכנות מערכות 883589

חלופה: מערכות טלפונים ניידים

מגיש: ראובנס יהונתן אליעזר

ת.ז. 327932679

מנחה: יחזקאל אנג'ל

תיכון קרית נוער ירושלים



תאריך הגשה: ביוני 2023 19

תוכן העניינים

1.....	תוכן העניינים.....
5.....	מענה לדרישות משרד החינוך לפרויקט 5 יח"ל.....
7.....	מבוא.....
7.....	הרקע לפרויקט.....
7.....	מטרת המערכת.....
7.....	יכולות המערכת.....
8.....	תהליך המחקר.....
8.....	אתגרים מרכזיים.....
9.....	מדריך למשתמש.....
9.....	תרשים מסכים.....
10.....	תיאור המסכים.....
10.....	מסך כניסה.....
11.....	מסך הירשמות.....
12.....	מסך הבית.....
13.....	מסך השיעור למורה.....
14.....	מסך שיעורי הבית.....
15.....	מסך המבחנים.....
16.....	מסך השיעורים לתלמיד.....
17.....	מסך האירועים.....
18.....	מסך השיעורים למורה.....
20.....	מסך הוספת מבחן.....
21.....	מסך השיעור.....
22.....	התפריט הראשי של האפליקציה.....
23.....	הרשאות.....
23.....	דרישות מיוחדות ומגבלות.....
23.....	גירסת Android מינימלית.....
23.....	מכשירים עליהם נבדקה המערכת.....
24.....	תיאור תחום הידע.....
24.....	אובייקטים נחוצים.....
24.....	סוגי נתונים.....
25.....	מבנה/ארכיטקטורה.....
25.....	מחלקות הפרויקט וקשרים ביניהם.....
25.....	תרשים מחלקות.....
29.....	מחלקות.....
30.....	מחלקה AddClassFragment.....
31.....	מחלקה lesson.....
32.....	מחלקה AddHomeworkFragment.....
33.....	מחלקה AddLessonFragment.....
34.....	מחלקה AddMenuActivity.....
35.....	מחלקה AddStudentToClassFragment.....

36.....	StudentsToClassAdapter	מחלקה
37.....	StudentsToClassViewHolder	מחלקה
38.....	AddTestActivity	מחלקה
39.....	TestInput	מחלקה
40.....	StudentTestListAdapter	מחלקה
41.....	StudentTestListViewHolder	מחלקה
42.....	ClassDetailsActivity	מחלקה
43.....	StudentsListInClassAdapter	מחלקה
44.....	StudentsListInClassViewHolder	מחלקה
45.....	Lesson	מחלקה
47.....	LessonsListAdapter	מחלקה
48.....	LessonsListViewHolder	מחלקה
49.....	ClassesTActivity	מחלקה
50.....	ClassAdapter	מחלקה
51.....	ClassViewHolder	מחלקה
52.....	HomePageActivity	מחלקה
53.....	Lesson	מחלקה
54.....	LessonAdapter	מחלקה
55.....	LessonViewHolder	מחלקה
56.....	LessonSFragment	מחלקה
57.....	LessonTActivity	מחלקה
58.....	Student	מחלקה
59.....	StudentsListAdapter	מחלקה
60.....	StudentsListViewHolder	מחלקה
61.....	MainActivity	מחלקה
62.....	NotificationReceiver	מחלקה
63.....	ShowDisturbanceActivity	מחלקה
64.....	DisturbanceListAdapter	מחלקה
65.....	DisturbanceListViewHolder	מחלקה
66.....	SignInActivity	מחלקה
67.....	SignUpActivity	מחלקה
68.....	User	מחלקה
69.....	StudentClassesActivity	מחלקה
70.....	classesAdapter	מחלקה
71.....	classesViewHolder	מחלקה
72.....	StudentHomeworksActivity	מחלקה
73.....	Homework	מחלקה
74.....	HomeworksAdapter	מחלקה
75.....	HomeworksViewHolder	מחלקה
76.....	StudentTestsActivity	מחלקה
77.....	TestOutput	מחלקה
78.....	TestAdapter	מחלקה
79.....	TestViewHolder	מחלקה

80.....	מחלקה Test
81.....	מחלקה Time
82.....	מחלקה UserLocation
83.....	שמירת נתונים
83.....	קבצים
83.....	Shared Preferences
84.....	סכמת Database
89.....	רפלקציה אישית
90.....	ביבליוגרפיה
91.....	נספחים
91.....	קוד המקור (source code))
91.....	מחלקה AddClassFragment
99.....	מחלקה AddHomeworkFragment
104.....	מחלקה AddLessonFragment
107.....	מחלקה AddMenuActivity
109.....	מחלקה AddStudentToClassFragment
113.....	מחלקה AddTestActivity
119.....	מחלקה ClassDetailsActivity
129.....	מחלקה ClassesTActivity
134.....	מחלקה HomePageActivity
142.....	מחלקה HomePageActivity
145.....	מחלקה LessonTActivity
156.....	מחלקה MainActivity
157.....	מחלקה NotificationReceiver
158.....	מחלקה ShowDisturbanceActivity
162.....	מחלקה SignInActivity
169.....	מחלקה SignUpActivity
174.....	מחלקה StudentClassesActivity
178.....	מחלקה StudentHomeworksActivity
182.....	מחלקה StudentTestsActivity
186.....	מחלקה Test
187.....	מחלקה Time
189.....	מחלקה UserLocation

מענה לדרישות משרד החינוך לפרויקט 5 יח"ל

מחלקה/ות בהן מומש הנושא	נושאים שמומשו	פרק במחווך
NotificationReceiver	AlarmManager + Notification •	6
AddClassFragment AddHomeworkFragment AddStudentToClassFragment AddTestActivity ClassDetailsActivity ClassesTActivity HomePageActivity LessonSFragment LessonTActivity ShowDisturbanceActivity SignInActivity SignUpActivity StudentClassesActivity StudentHomeworksActivity StudentTestsActivity	אחסון וטיפול בנתונים (חובה שמירה ושליפה)	7
	אפליקציית רב משתתפים •	9
AddClassFragment AddHomeworkFragment AddLessonFragment AddStudentToClassFragment LessonSFragment	Fragment •	
AddClassFragment	Threads •	
AddClassFragment AddHomeworkFragment AddStudentToClassFragment AddTestActivity ClassDetailsActivity ClassesTActivity HomePageActivity LessonSFragment LessonTActivity ShowDisturbanceActivity SignInActivity SignUpActivity StudentClassesActivity StudentHomeworksActivity StudentTestsActivity	Firebase / Firestore •	
AddTestActivity	Shared Preference •	10

ClassDetailsActivity ClassesTActivity HomePageActivity LessonSFragment LessonTActivity ShowDisturbanceActivity SignInActivity SignUpActivity StudentClassesActivity StudentTestsActivity		
UserLocation SignInActivity LessonTActivity	GPS •	

מבוא

הרקע לפרויקט

הפרויקט שלי הוא EduCare. זו אפליקציה לתפעול מערכות למידה קטנות. (לדוגמא קביעת שיעורים, מתן ציונים וכו...). קהל היעד של האפליקציה הוא תלמידים ומורים אשר לומדים/מלמדים במערכות למידה קטנות וזקוקים למערכת שתאפשר להם לבצע זאת בצורה טובה ויעילה. בחרתי בנושא זה מכיוון והוא קשור אלי אישית, אני בזמני הפנוי מעביר שיעורים פרטיים ולעיתים מתקשה לעקוב אחרי השעות והזמנים של השיעורים, מגוון האפליקציות הקיימות בשוק פונות לקהל יעד של מערכות גדולות ולא התאימו לצרכי האישיים. ניסיתי להשתמש באפליקציות לניהול זמנים כללים אך הרגשתי שזה לא מספיק ורציתי משהו שיבנה בצורה יעודית לצרכי. בנוסף אחותי עובדת בכפר נוער ולכן התייעצתי איתה והרגשתי שהאפליקציה במידה ותהיה מפותחת מספיק תוכל להוות פיתרון לכלל מערכות למידה.

מטרת המערכת

המערכת מאפשרת למורה לקבוע שיעורים עם תלמידים. המערכת כוללת ממשק למורה וממשק לתלמיד שכל אחד מהם מבצע דברים אחרים אך קשורים. הממשק לתלמיד בנוי כך שהוא יכול לראות את כלל המידע שהוזן על ידי המורה (כגון ציונים, נוכחות, שיעורי בית וכו...) ולעקוב אחר השיעורים שלו. הממשק למורה מתנהל כך שהמורה בעיקרו מזין מידע ובנוסף יכול להשתמש במידע שהזין, לדוגמא המורה יכול לפתוח שיעור ולקשר אליו תלמידים ולאחר מכן לעבור על נתוני ההתנהגות של התלמידים בשיעורים.

יכולות המערכת

- אפשרות לשימוש בשני מכשירים שונים
- אפשרות למורה לעדכן מידע אצל מספר תלמידים בו זמנית
- אפשרות להזנת נוכחות אוטומטית על פי מיקום GPS
- אפשרות להוספת התראות לפני שיעור
- הצגה של פרטי התלמיד

תהליך המחקר

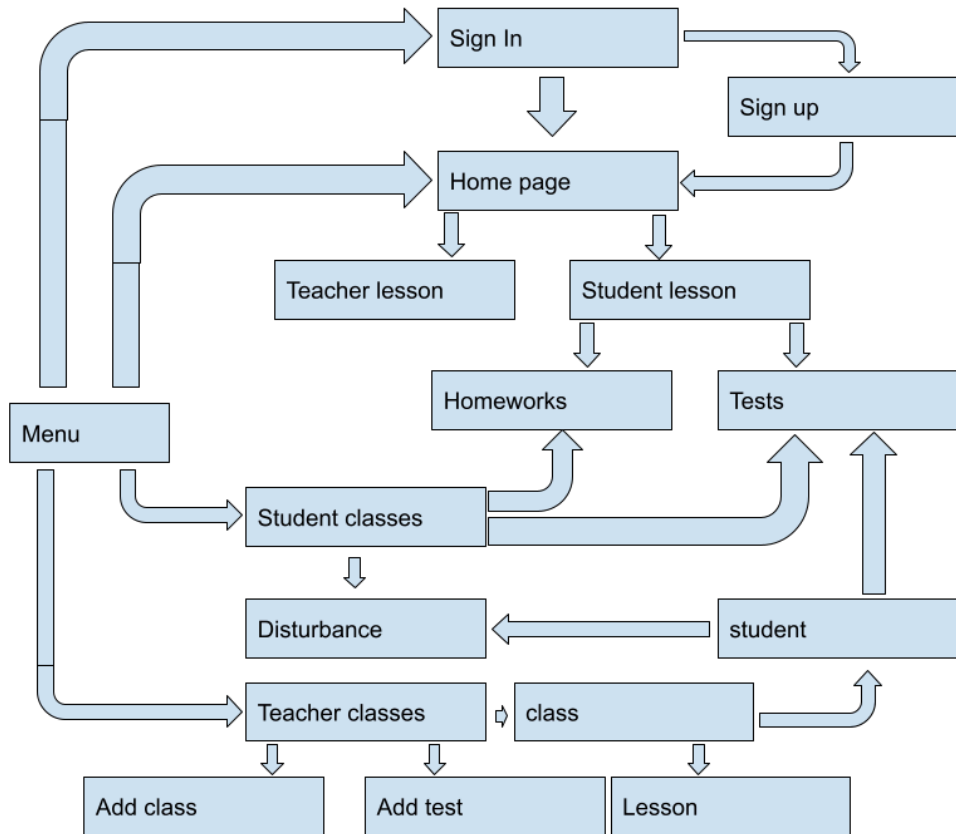
1. מחקר על תחום הידע: מערכות למידה קטנות הן מערכות המיועדות לשימוש במסגרת קהילות או קבוצות קטנות של אנשים. הן מספקות פלטפורמה טכנולוגית שמאפשרת למשתמשים ללמוד, לשתף ולתקשר עם זהים מחברתיים ומורים בדרך אינטראקטיבית. מערכות כאלו דורשות.
 - גישה ניידת: מערכות למידה קטנות מתאימות למשתמשים שמחפשים גישה ניידת לתוכן הלמידה. סמארטפונים שנמצאים בשימוש נרחב ומאפשרים למשתמשים ללמוד בכל מקום ובכל זמן שהם רוצים.
 - מעקב ואיתור התקדמות: מערכות למידה קטנות מאפשרות מעקב ואיתור התקדמות ברמה אישית וקבוצתית. המשתמשים יכולים לראות את הציונים שלהם.
2. עבודת שטח: כיום בשוק ישנם אפליקציות רבות העושות פעולות דומות אך למערכות למידה גדולות (כדוגמת משוב או smartSchool), מה שיוצר סרבול רב בהתנהלות עם האפליקציה. בנוסף ישנם אפליקציות שאפשר לנצל אותם לפעולות כאלו כגון Whatsapp אבל אלו לא נוחות בתפעולם לשימושים כאלו (לדוגמא ריכוז ציונים).
3. פירוט טכנולוגיות שהשתמשו שאינן חלק מתכנית הלימודים: שימוש במחלקות פנימיות, שימוש בתפריט, העברת נתונים בין מערכות.

אתגרים מרכזיים

1. הבעיה איתה התמודד התלמיד במהלך פיתוח הפרוייקט: במהלך הפרוייקט התמודדתי בעיקר עם הבעיה של תיאום בין הממשק למורה לבין הממשק לתלמיד. הדרישה לתיאום יוצרת בעיות רבות שהרי מידע שנכתב ברגע מסויים ניתן לבצע עליו מניפולציה מאותו רגע הן המורה והן התלמיד.
2. על איזה צורך הפרויקט עונה? הפרויקט עונה על צורך של מערכות למידה קטנות לממשק נוח ויעיל בו כל המידע על התלמיד מוצג בצורה מסודרת.
3. הצגת פתרונות לבעיה הפתרונות שנבחנו במסגרת המחקר המקדים: אפליקציות למערכות מידע גדולות, אפליקציות לרשימת נתונים, שימוש בשיטה הישנה (דפים), שימוש באפליקציה שעוצה במיוחד לשם הבעיה הזו.

מדריך למשתמש

תרשים מסכים



תיאור המסכים

מסך כניסה

המסך הראשון שמשתמש לא מחובר רואה בו ניתן להתחבר למשתמש רשום במערכת

אלמנטים:

- תיבת טקסט שבו המשתמש מזין את כתובת המייל הרשומה
- תיבת טקסט שבו המשתמש מזין את הסיסמה שלו לחשבון
- כפתור שכשהמשתמש לוחץ עליו הוא מתחבר לחשבון ומופנה למסך הבית
- כפתור שבו משתמש לא רשום יכול להירשם

מסך הירשמות

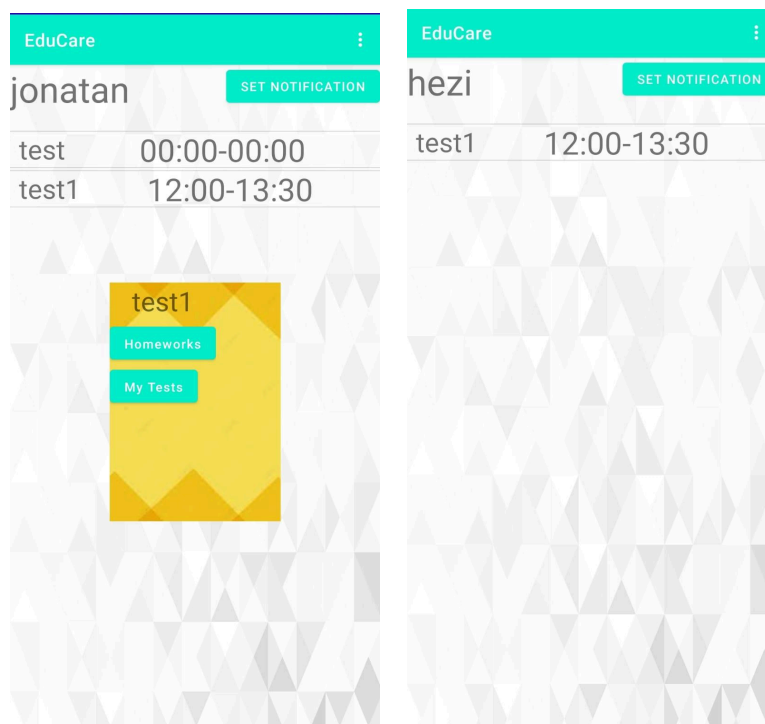
המסך בו המשתמש יכול להירשם לאפליקציה

אלמנטים:

- תיבת טקסט בה המשתמש מזין את שם המשתמש
- תיבת טקסט בה המשתמש מזין את הסיסמה שהוא מעוניין שתהיה לו
- תיבת טקסט בה המשתמש מאמת את סיסמתו
- תיבת טקסט בה המשתמש מזין את כתובת המייל שבה הוא רוצה להתחבר
- תיבת טקסט בה המשתמש מזין את הארגון אליו הוא שייך (במידה והוא יזין ארגון שלא קיים במערכת התוכנה תרשום גם את הארגון החדש)
- מתג בו המשתמש בוחר את סוגו (תלמיד / מורה)
- כפתור אשר רושם את המשתמש ומפנה אותו למסך הבית של האפליקציה

מסך הבית

המסך הראשון אותו רואה משתמש רשום

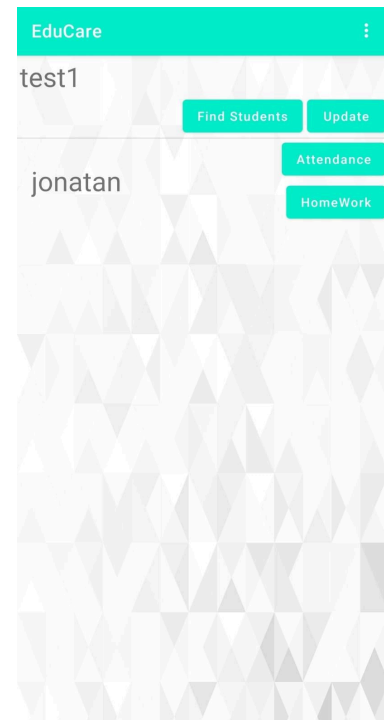


אלמנטים:

- טקסט הכולל את שם המשתמש
- רשימת שיעורים המתרחשים באותו היום יחד עם השעות בו הם מתקיימים (לחיצה על שיעור תוביל את המשתמש למסך השיעור בהתאם לתנאים) הללו :
- במידה והמשתמש הוא מורה הוא יופנה ל"מסך השיעור למורה"
- במידה והמשתמש הוא תלמיד יפתח מסך קטן כמראה בתמונה שבו
 - טקסט המציין את שם השיעור
 - כפתור שמפנה את המשתמש ל"מסך שיעורי הבית" באותו נושא
 - כפתור שמפנה את המשתמש ל"מסך המבחנים" בו מוצגים המבחנים שהיו לו באותו נושא
- כפתור המאפשר למשתמש להגדיר התראה לכלל השיעורים שלו באותו יום (ההתראה תופעל 5 דקות לפני השיעור)
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך השיעור למורה

המסך אותו יכול לראות המורה בנוגע לשיעור ספציפי ובו להזין נתונים על התלמידים



אלמנטים:

- טקסט שבו מופיע שם השיעור
- רשימת תלמידים הכוללת על כל תלמיד
- 1. טקסט בו מופיע שם התלמיד
- 2. כפתור בו ניתן לסמן האם התלמיד נוכח בשיעור
- 3. כפתור בו ניתן לסמן האם התלמיד עשה את שיעורי הבית
- כפתור המאפשר למורה למצוא את התלמידים ולהזין להם נוכחות באופן אוטומטי
- כפתור שבו המשתמש מאשר שסיים להזין מידע (הכפתור מעלה את המידע ומחזיר את המשתמש למסך הבית)
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך שיעורי הבית

המסך בו רואה המשתמש בו מוזנים שיעורי הבית שניתנו לו מנושא ספציפי



אלמנטים:

- טקסט בו מופיע שם התלמיד
- טקסט בו מופיע הכיתוב Homeworks
- רשימה של שיעורי הבית שניתנו בנושא שנבחר הכוללת:
 - טקסט שבו מופיעים פרטי שיעורי הבית
 - טקסט בו מופיע תאריך ההגשה של שיעורי הבית
 - תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך המבחנים

המסך איליו מופנה המשמש שבו הוא יכול לראות את הציונים של תלמיד מסויים בנושא ספציפי

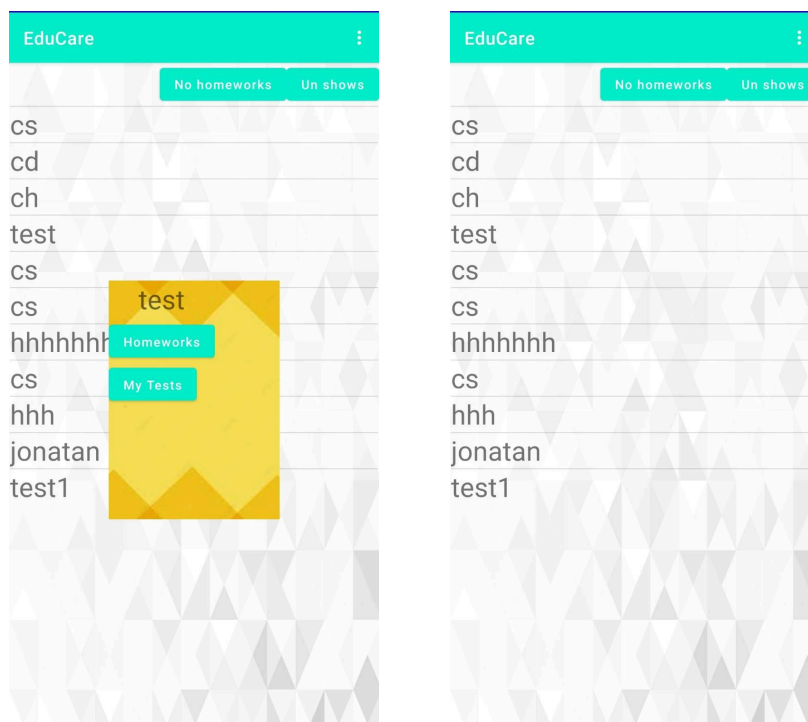
EduCare	
jonatan	
Tests	
test1	jonatan
100	19/06/2023
	test1

אלמנטים:

- טקסט בו רשום שם התלמיד
- טקסט בו רשום הכיתוב Tests
- רשימה של מבחנים שבכל מבחן מופיע
- טקסט בו רשום שם המבחן
- טקסט בו רשום שם התלמיד הנבחן
- טקסט בו רשום הציון במבחן
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך השיעורים לתלמיד

המסך אילו מופנה התלמיד דרך התפריט בו הוא יכול לצפות בכלל השיעורים בהם הוא משתתף

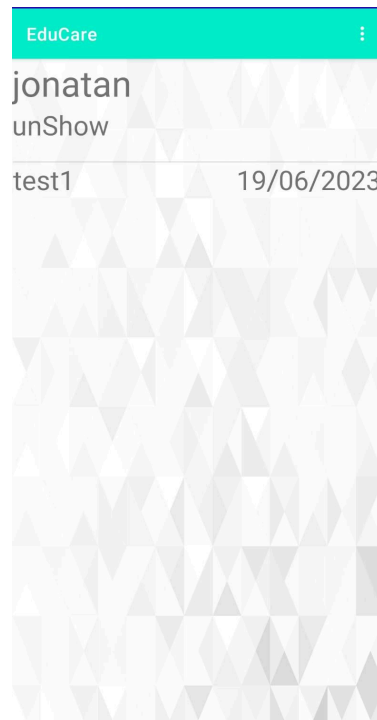


אלמנטים:

- כפתור המפנה את המשתמש אל "מסך האירועים" בנושא שיעורי הבית בכלל השיעורים בהם התלמיד משתתף
- כפתור המפנה את המשתמש אל "מסך האירועים" בנושא היעדרויות משיעורים בכלל השיעורים בהם התלמיד משתתף
- רשימה של כלל הכיתות אליהם התלמיד משויך במידה והמשתמש לוחץ על אחד השיעורים יופיע מסך קטן ובו
 - טקסט המציין את שם השיעור
 - כפתור שמפנה את המשתמש ל"מסך שיעורי הבית" באותו נושא
 - כפתור שמפנה את המשתמש ל"מסך המבחנים" בו מוצגים המבחנים שהיו לו באותו נושא
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך האירועים

המסך אילו מופנה המשתמש בנוגע לסוג אירוע מסויים של תלמיד בו הוא יצפה בהיסטורית האירועים שלו באותו נושא

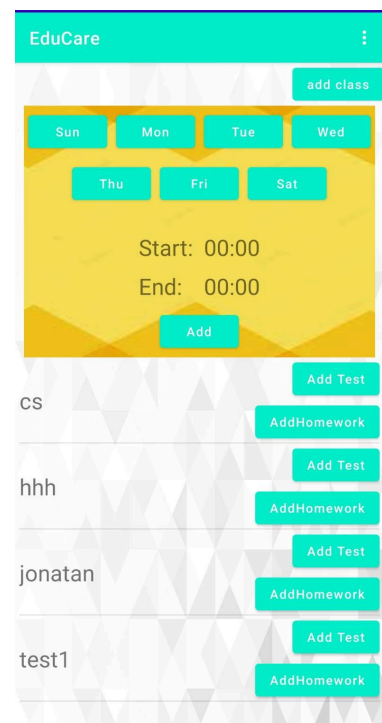
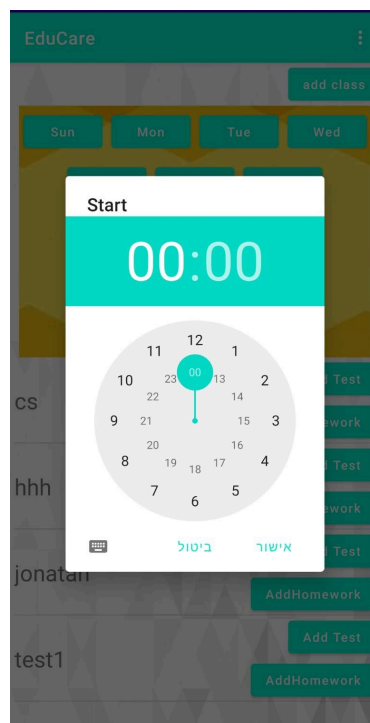
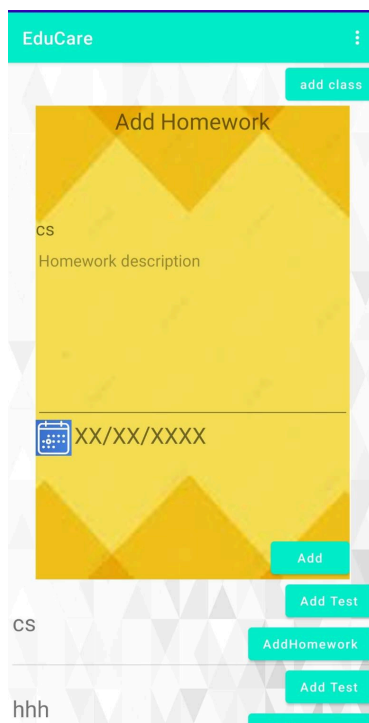
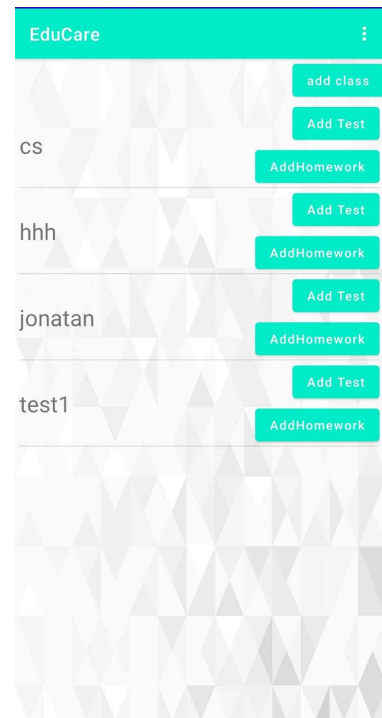
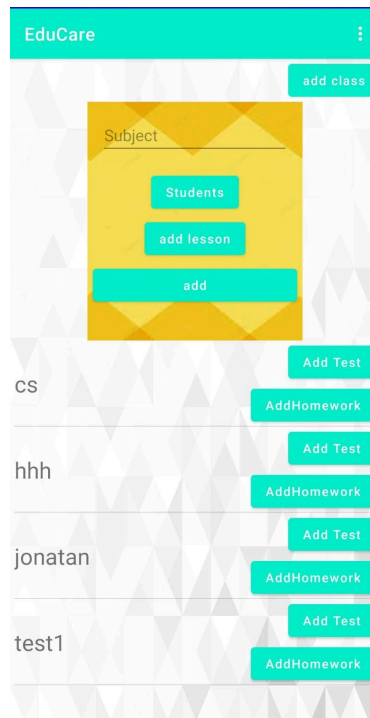


אלמנטים:

- טקסט בו רשום שם התלמיד
- טקסט בו רשום סוג האירוע (אי עשיית שיעורי בית / היעדרות משיעור)
- רשימה של כל האירועים בנושא הנבחר שבו מצויין
 1. טקסט ובו הנושא בו בוצע האירוע
 2. טקסט ובו מצויין התאריך בו התרחש האירוע
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך השיעורים למורה

המסך אילו מופנה המורה ובו מוצגים כלל השיעורים אותם הוא מלמד ובו הוא יכול לבצע פעולות על כל שיעור



אלמנטים:

- כפתור המאפשר למשתמש להוסיף כיתה חדשה למקצוע הכפתור פותח מסך קטן ובו
 - תיבת טקסט בה המשתמש מזין את שם המקצוע
 - כפתור המפנה למסך קטן בו המשתמש בוחר מי יהיו התלמידים בכיתה שכולל:
 - רשימת התלמידים בארגון (לחיצה על תלמיד תוסיף אותו לכיתה)
 - כפתור הוספה שמאשר את הבחירה של המשתמש
 - כפתור המפנה את המשתמש למסך קטן בו הוא מזין את השיעורים וכולל:
 - כפתורים המציינים את ימי השבוע לחיצה על כפתור כזה יבחר את היום לשיעור
 - טקסט ובו מצויין שעת ההתחלה לחיצה על הטקסט תאפשר למשתמש לשנות את השעה
 - טקסט ובו מצויין שעת הסיום לחיצה על הטקסט תאפשר למשתמש לשנות את השעה
 - כפתור הוספה שמאשר את הבחירה של המשתמש
 - כפתור המאשר את בחירת המשתמש
 - רשימת השיעורים אותם מעביר המשתמש ובהם:
 - כפתור המפנה את המשתמש אל "מסך הוספת מבחן"
 - כפתור המאפשר למורה להוסיף שיעורי בית בנושא מסויים ופותח מסך קטן לשם כך ובו:
 - Add Homework
 - טקסט ובו הכיתוב
 - טקסט ובו מצויין שם השיעור
 - תיבת טקסט בה המורה מזין את פרטי שיעורי הבית
 - טקסט ובו מצויין התאריך להגשה (לחיצה על האייקון ליד הטקסט מאפשרת שינוי התאריך)
 - כפתור הוספה שמאשר את הבחירה של המשתמש
 - לחיצה על שיעור תפנה את המשתמש אל "מסך השיעור"
 - תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך הוספת מבחן

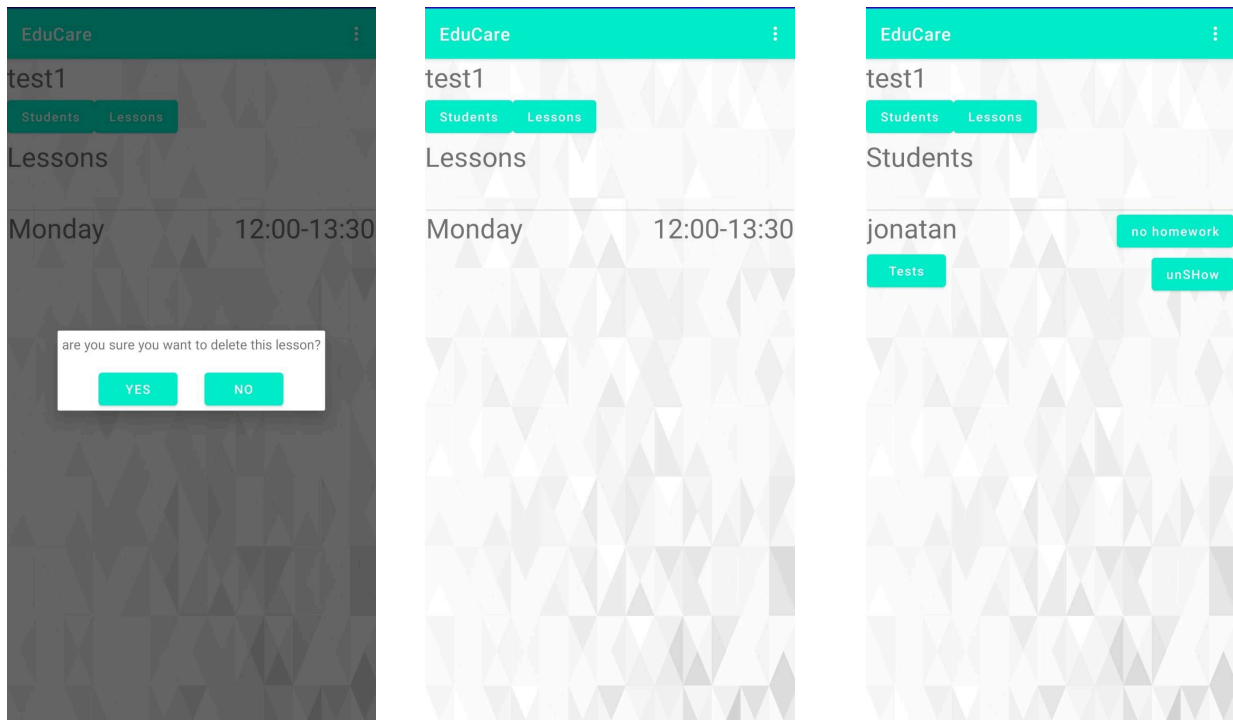
המסך אילו מופנה המורה בו הוא מוסיף מבחן ומזין לתלמידים ציונים

אלמנטים:

- טקסט בו כתוב Add Test
- טקסט בו מצויין שם השיעור
- תיבת טקסט בה המשתמש מזין את שם המבחן
- רשימת התלמידים שנמצאים בכיתה ועל כל תלמיד יש:
 1. טקסט ובו מופיע שם התלמיד
 2. תיבת טקסט בה המורה מציין את ציון התלמיד במבחן
- כפתור הוספה שמאשר את הבחירה של המשתמש
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

מסך השיעור

המסך אילו מופנה מורה ובו הוא צופה בפרטים על השיעור



אלמנטים:

- טקסט ובו מצויין שם השיעור
- כפתור הפותח את רשימת התלמידים הרשומים בשיעור ועל כל אחד
- טקסט המציין את שם התלמיד
- כפתור המפנה אל מסך המבחנים של התלמיד באותו שיעור
- כפתור שמפנה את המשתמש ל"מסך שיעורי הבית" באותו נושא
- כפתור שמפנה את המשתמש ל"מסך המבחנים" בו מוצגים המבחנים שהיו לו באותו נושא
- כפתור הפותח את רשימת השיעורים באותו מקצוע ובה מופיע:
- טקסט המציין את היום והשעה בה מתרחש כל שיעור
- לחיצה ארוכה על שיעור תחזק את השיעור לא לפני שתוודא שהמשתמש אכן מעוניין בכך
- טקסט המציין את סוג הרשימה הנפתח כעת
- תפריט (התפריט הרגיל של האפליקציה מורחב בהמשך)

התפריט הראשי של האפליקציה

התפריט המאפשר ניווט באפליקציה לכל משתמש מחובר

Home

My Classes

Log out

אלמנטים:

- כפתור המפנה אל מסך הבית של האפליקציה
- כפתור המפנה את המשתמש אל רשימת השיעורים בהתאם
 1. משתמש מסוג מורה יופנה אל "מסך השיעורים למורה"
 2. משתמש מסוג תלמיד יופנה אל "מסך השיעורים לתלמיד"
- כפתור התנתקות המוחק את פרטי המשתמש ומחזיר את המשתמש אל "מסך כניסה"

הרשאות

android.permission.SET_ALARM - הרשאה זו מאפשרת לאפליקציה להגדיר את השעות של התרעות (הבנתי שזו התרעה על התרחשות אירוע מסוים). זו הרשאה מתן לאפליקציה ליצור ולבטל את התרעות השעון.

android.permission.WAKE_LOCK - הרשאה זו מאפשרת לאפליקציה למנוע מהמסך של מכשיר Android להיכנס למצב "שינה" (sleep mode) ולהשאיר אותו פעיל. זה יכול לשמש למטרות כמו להפעיל את המסך במקום כדי להציג מידע למשתמש גם כאשר המכשיר מונע מלהיכנס למצב שינה.

android.permission.ACCESS_COARSE_LOCATION - הרשאה זו מאפשרת לאפליקציה לגשת למידע כתובות IP מרוחקות כדי לקבוע את מיקומו הגולמי של המכשיר בצורה יחסית גסה. זו הרשאה יחסית פחות מדויקת ויכולה לשמש לאפליקציות שמתעסקות עם שירותים מבוססי מיקום כלליים.

android.permission.ACCESS_FINE_LOCATION - הרשאה זו מאפשרת לאפליקציה לגשת למידע מדויק יותר על מיקום המכשיר. זו הרשאה יותר מדויקת מההרשאה ACCESS_COARSE_LOCATION ומאפשרת לאפליקציה לקבוע את מיקומו הגאוגרפי של המכשיר בצורה מדויקת.

דרישות מיוחדות ומגבלות

GPS, תקשורת Internet

גירסת Android מינימלית

Android 4.3.14 גירסת API 31

מכשירים עליהם נבדקה המערכת

Sumsung galaxy A9

Resizable (Experimental) API 33

Pixel_3a_API_33_x86_64

תיאור תחום הידע

אובייקטים נחוצים

שיעור
מבחן
תלמיד
משתמש
שיעורי בית
זמן
מיקום

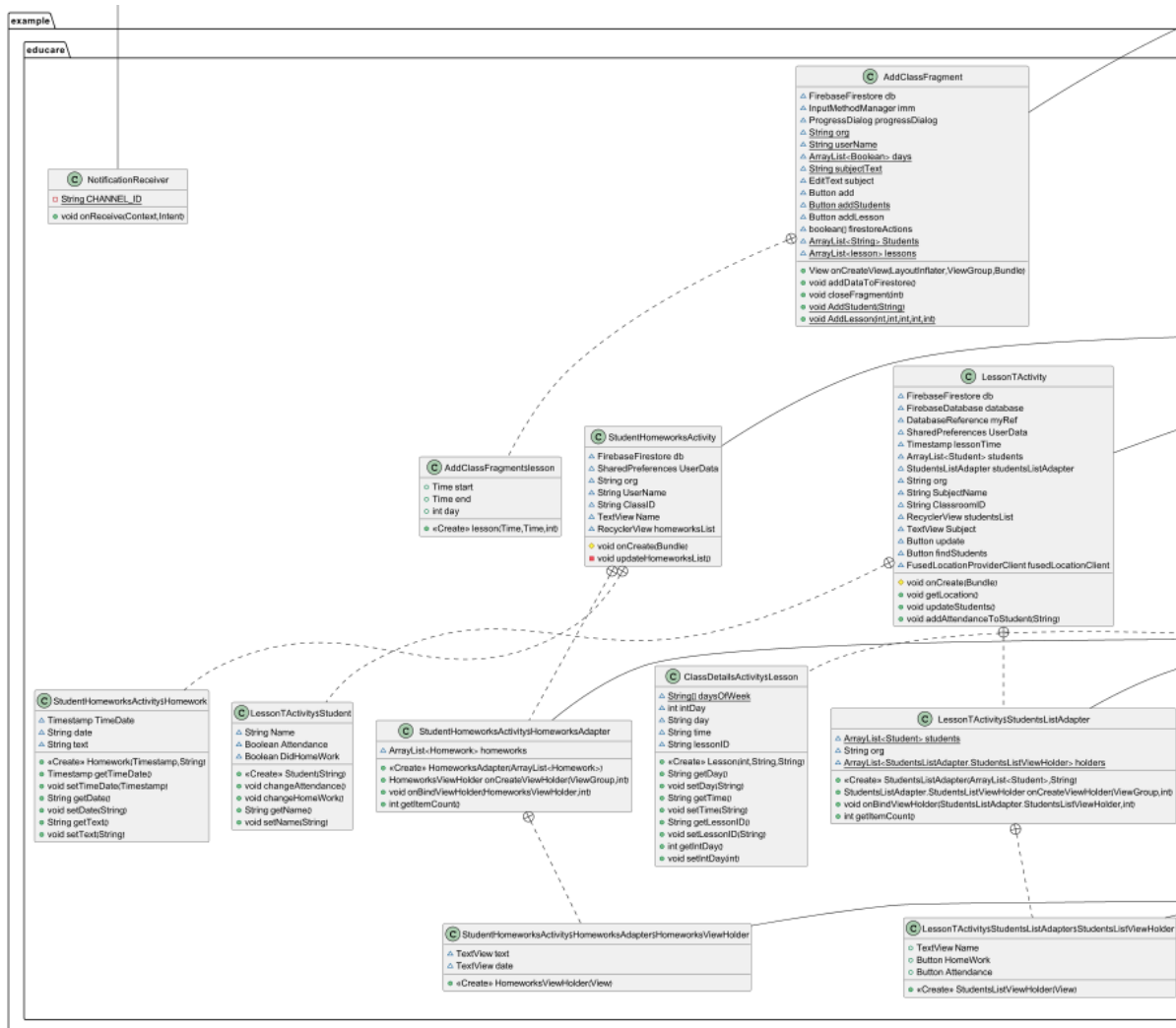
סוגי נתונים

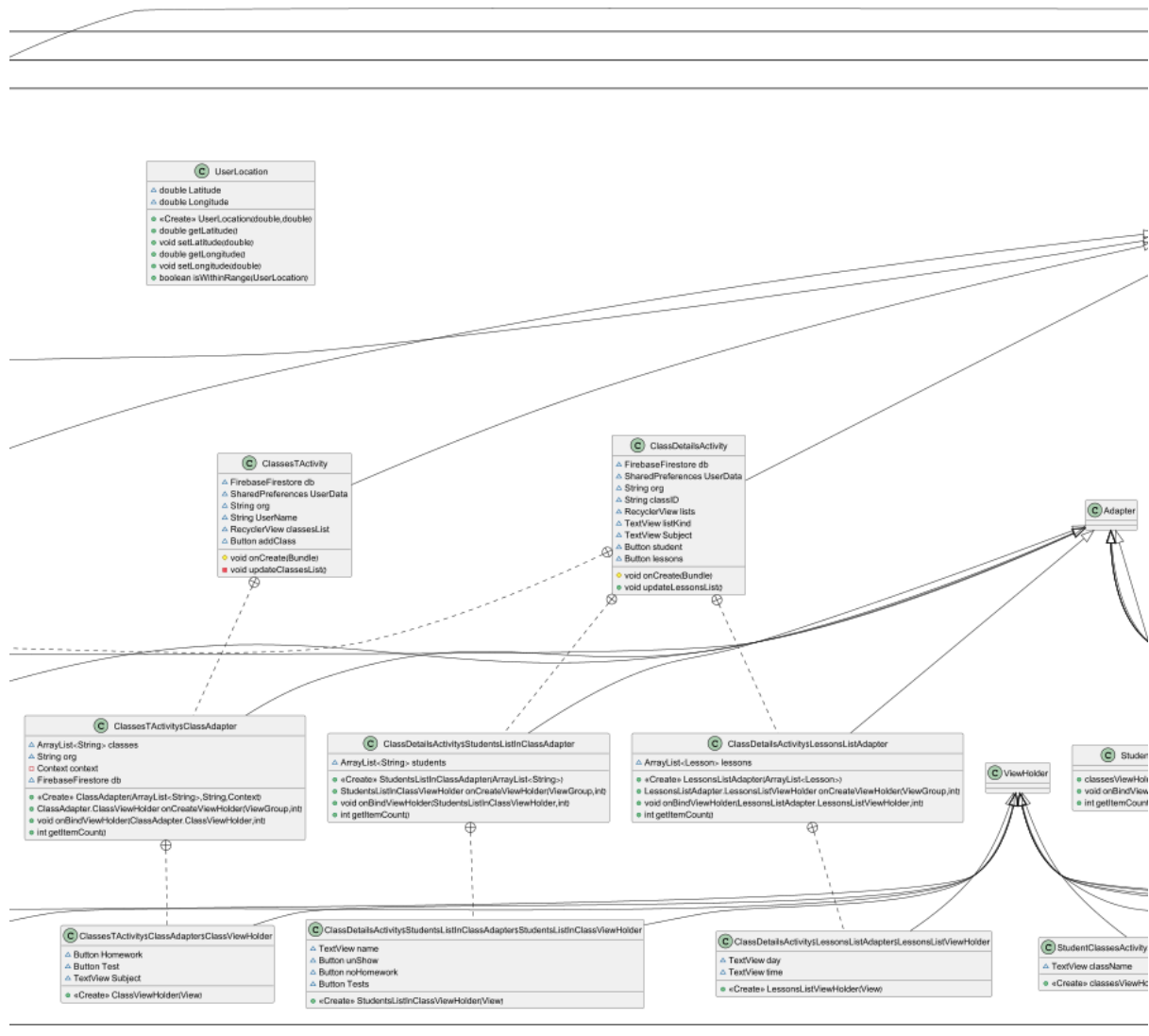
שיעור - זמן התחלה, זמן סיום, יום
מבחן - שם מבחן, ציון, נושא, שם המורה הבוחן, תאריך
תלמיד - שם, נוכחות, עשיית ש"ב
משתמש - Email, שם משתמש, סוג משתמש
שיעורי בית - תאריך הגשה, פרטי שיעורי הבית
זמן - שעה, דקה
מיקום - מיקום צפון-דרום, מיקום מזרח-מערב

מבנה/ארכיטקטורה

מחלקות הפרויקט וקשרים ביניהם

תרשים מחלקות









Holder

מחלקות

עבור כל מחלקה הקיימת בפרויקט (לא רק אקטיביטי, אלא כולל מחלקות העצמים, מחלקות עזר, מחלקות סרוויס וכו') יש להציג תיאור כל מחלקה בדף נפרד:

- שם המחלקה
- תפקיד המחלקה
- הסבר עבור כל תכונה ושימושה (כולל תכונות סטטיות)
- הסבר עבור כל פעולה ותפקידה (למעט get/set).
- את הפעולות הלוגיות העיקריות יש להסביר בהרחבה ולצרף קוד.
- כאשר מתבצע זימון פעולות סדרתי, יש לצרף תרשים זרימה.

מחלקות אשר עוסקות במסדי נתונים (כדוגמת: רשימת פריטים, משתמשים...) יש להציג תיאור של המידע כפי שנשמר בקובץ / טבלה.

מחלקה AddClassFragment

```

FirebaseFirestore db = FirebaseFirestore.getInstance(); - קישור לפיירסטור
InputMethodManager imm; - fragments קישור למקלדת כדי שיהיה ניתן לסגור אותה בזמן שעוברים בין -
ProgressDialog progressDialog; - dialog של טעינה
static String org; - הארגון אילו משויך המשתמש -
static String userName; - שם המשתמש -
static ArrayList<Boolean> days = new ArrayList<>(Collections.nCopies(7, false)); - מערך עזר המציין -
באילו ימים נוספו שיעורים
static String subjectText; - משתנה השומר את שם השיעור -
EditText subject; - תיבת טקסט -
Button add; - כפתור המשמש להוספה של הכיתה -
static Button addStudents; - כדי לאפשר הוספת תלמידים fragments כפתור המעביר בין -
Button addLesson; - כדי לאפשר הוספת שיעורים fragments כפתור המעביר בין -
static ArrayList<String> Students = new ArrayList<>(); - מערך השומר את התלמידים בכיתה -
static ArrayList<Lesson> lessons = new ArrayList<>(); - מערך השומר את ימי השיעורים -

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
    פעולה הטוענת את הfragment

public void addToFirestore() - פעולה אשר מעלה את המידע לפיירסטור

    boolean[] firestoreActions = new boolean[]{false, false, false};
    /**
     * 0 - add class to the teacher
     * 1 - add class to the students
     * 2 - add lesson to firebase
     */
    public void closeFragment(int code)
        פעולה ומערך אשר מצינים איזה מידע עלה לפיירסטור וברגע שכל המידע עלה סוגרים את הfragment
    public static void AddStudent(String student) מ
    public static void AddLesson(int day, int startHour, int startMinute, int endHour, int endMinute)
        פעולה המאפשרת הוספה של שיעורים מ fragments אחרים

```

מחלקה lesson

(מחלקה פנימית בAddClassFragment)

משתנה השומר את זמן תחילת השיעור - `public Time start;`
משתנה השומר את זמן סוף השיעור - `public Time end;`
משתנה השומר את יום השיעור - `public int day;`

AddHomeworkFragment מחלקה

FirebaseFirestore db = FirebaseFirestore.getInstance(); - קישור לפיירסטור
 int mDate, mMonth, mYear; - משתנים השומרים את התאריך הנבחר
 ArrayList<String> Students; - מערך השומר את רשימת התלמידים ביתה
 String org; - משתנה השומר את שם הארגון של המשתמש
 String ClassID; - משתנה השומר את אמצעי הזיהוי בפיירבייס של הכיתה
 String subject; - משתנה השומר את שם השיעור
 TextView Subject; - טקסט
 TextView dateTXT; - טקסט של התאריך
 ImageView Cal; - אייקון של לוח שנה לבחירת תאריך
 EditText description; - תיבת טקסט להזנת פרטים
 Button Add; - כפתור המשמש להוספת שיעור
 private void ImportStudentsListFromFirestore()
 פעולה אשר מוצאת בפיירסטור את רשימת התלמידים הרשומים בכיתה
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
 פעולה אשר מקימה את הfragment לפי הדרישות
 public void addDataToFirestore() - פעולה המוסיפה את שיעורי הבית לתלמידים

AddLessonFragment מחלקה

int day = 0; - משתנה השומר את היום של השיעור
 int startHour; - משתנה השומר את שעת ההתחלה של השיעור
 int startMinute; - משתנה השומר את דקות ההתחלה של השיעור
 int endHour; - משתנה השומר את שעת הסיום של השיעור
 int endMinute; - משתנה השומר את דקות הסיום של השיעור
 TextView Start; - טקסט שמשמש ככפתור ומציג את זמן ההתחלה של השיעור
 TextView End; - טקסט המשמש ככפתור ומציג את זמן הסיום של השיעור
 Button Add; - כפתור להוספת השיעור וחזרה ל - AddClassFragment
 Button[] days = new Button[7]; - שבעה כפתורים המייצגים את שבעת הימים בשבוע ומשמשים לבחירת יום
 OnClickListener timeOnClick - מאזין המשמש לקביעת זמן התחלה וסיום
 OnClickListener dayOnClick - מאזין המשמש לקביעת יום לשיעור
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
 פעולה אשר מגדיר את כלל המשתנים ויוצרת את הfragment

AddMenuActivity מחלקה

Intent i; - Activity משתנה המאפשר מעבר בין -

SharedPreferences UserData; - מאפשר גישה למידע השמור במכשיר של המשתמש -

public boolean onCreateOptionsMenu(Menu menu)

פעולה המתרחשת כאשר ה-Activity נוצר ויוצרת את התפריט

public boolean onOptionsItemSelected(@NonNull MenuItem item)

פעולה שמופעלת כאשר אחד מן הפריטים נבחר ומבצעת את הפעולה הנדרשת על פי הפריט

מחלקה AddStudentToClassFragment

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`
 הממשש לטעינת רשימת התלמידים `StudentsToClassAdapter studentsToClassAdapter; - Adapter`
 רשימת התלמידים בארגון - `ArrayList<String> StudentsList = new ArrayList<>();`
 שם הארגון של המשתמש - `String org;`
 רשימה שמשמשת להצגת התלמידים - `RecyclerView students;`
`AddClassFragment` וחזרה ל `fragment` כפתור המשמ/ש ליציאה מה - `Button add;`
`public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)`
 פעולה אשר יוצרת את ה `fragment` עצמו
`public void findStudentsInFirestore()` - כלל התלמידים בארגון - מוסיפה לרשימה את כלל התלמידים בארגון
`updateStudentsList()` - פעולה אשר מעדכנת את רשימת התלמידים המוצגת למשתמש -

מחלקה StudentsToClassAdapter

(מחלקה פנימית ב-AddStudentToClassFragment)

public StudentsToClassViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) - באופן אינדיבידואלי view פעולה אשר יוצרת את ה -

public void onBindViewHolder(@NonNull StudentsToClassViewHolder holder, int position)

פעולה אשר מגדירה את כל הפריטים בתוך הview

public int getItemCount() - פעולה המחזירה את כמות האובייקטים ברשימה -

מחלקה StudentsToClassViewHolder

(מחלקה פנימית ב-StudentsToClassAdapter)

משתנה השומר את התיבה בה ירשם שם התלמיד - TextView name;

public StudentsToClassViewHolder(@NonNull View itemView)

פעולה אשר מגדירה את הפריטים במחלקה

AddTestActivity מחלקה

Firestore db = FirebaseFirestore.getInstance(); - קישור לפיירסטור
SharedPreferences UserData; - קישור למידע של המשתמש
ArrayList<String> students = new ArrayList<>(); - רשימה של התלמידים
StudentTestListAdapter studentsTestListAdapter; - Adapter של רשימת התלמידים
String org; - הארגון אליו שייך המשתמש
String ClassroomID; - מזהה הכיתה בפיירסטור
String teachers_name; - משתנה השומר את שם המורה
TextView Subject; - תיבת הטקסט בה נשמר נושא המבחן
RecyclerView studentList; - התיבה בה נופיע רשימת התלמידים
Button add; - הכפתור שמאשר את ציוני התלמידים
EditText testName; - תיבת טקסט השומרת את שם המבחן

protected void onCreate(Bundle savedInstanceState) - Activity פעולה הממשת את ה-
private void createStudentsList() - פעולה היוצרת את רשימת התלמידים

מחלקה TestInput

(יורשת Testn, מחלקה פנימית בAddTestActivity)

משתנה השומר את הזיהוי של הכיתה בפירסטור - String ClassID;

משתנה השומר את התאריך שבו בוצע המבחן - Timestamp date;

מחלקה StudentTestListAdapter

(מחלקה פנימית ב-AddTestActivity)

ArrayList<String> students; - רשימה השומרת את התלמידים -
 String org; - משתנה שהומר את הארגון של המשתמש -
 public static int[] grades; - מערך השומר את הציונים של כל תלמיד -
 public StudentTestListAdapter(StudentTestListViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) - בצורה אינבדואלית ברשימה view פעולה אשר יוצרת כל -
 public void onBindViewHolder(@androidx.annotation.NonNull StudentTestListAdapter.StudentTestListViewHolder holder, int position) - פעולה אשר מגדירה כל view לפי הנתונים שלו
 public int getItemCount() - פעולה המחזירה את כמות הפריטים ברשימה -

מחלקה StudentTestListViewHolder

(מחלקה פנימית ב-StudentTestListAdapter)

TextView name; - תיבת טקסט השומרת את שם התלמיד -

EditText grade; - תיבת טקסט השומרת את ציון התלמיד -

public StudentTestListViewHolder(@NonNull View itemView) - פעולה המגדירה את משתני המחלקה -

מחלקה ClassDetailsActivity

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`

קישור למידע על המשתמש - `SharedPreferences UserData;`

הארגון אילו משויך המשתמש - `String org;`

מזהה הכיתה בפיירסטור - `String classID;`

מקשר לרשימה שמופיעה למשתמש - `RecyclerView lists;`

טקסט שיזהה את סוג הרשימה - `TextView listKind;`

טקסט שיזהה את המקצוע - `TextView Subject;`

כפתור שמשמש להצגת רשימת התלמידים - `Button student;`

כפתור המשמש להצגת רשימת השיעורים - `Button lessons;`

ומגדירה את המשתנים Activity פעולה היוצרת את ה - `protected void onCreate(Bundle savedInstanceState)`

פעולה אשר מעדכנת את רשימת השיעורים במידה ובשימוש - `public void updateLessonsList()`

מחלקה StudentsListInClassAdapter

(מחלקה פנימית ב-ClassDetailsActivity)

רשימת ה- `ArrayList<String> students;`

`public StudentsListInClassViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)` - בצורה אינדיבדואלית ברשימה `view` פעולה אשר יוצרת כל

`public void onBindViewHolder(@NonNull StudentsListInClassViewHolder holder, int position)`

לפי הנתונים שלו `view` פעולה אשר מגדירה כל

`public int getItemCount()` - פעולה המחזירה את כמות הפריטים ברשימה -

מחלקה StudentsListInClassViewHolder

(מחלקה פנימית ב-StudentsListInClassAdapter)

TextView name; - טקסט שישמש לשם התלמיד -

Button unShow; - כפתור המשמש לצפיה בהיעדרויות של התלמיד -

Button noHomework; - כפתור המשמש לצפייה באירועי האי הבאת ש"ב של התלמיד -

Button Tests; - כפתור המשמש לצפייה במבחנים של התלמיד -

פעולה המגדירה את משתני המחלקה - public StudentsListInClassViewHolder(@NonNull View itemView)

מחלקה Lesson

(מחלקה פנימית ב-ClassDetailsActivity)

```
static String[] daysOfWeek = {  
    "Sunday",  
    "Monday",  
    "Tuesday",  
    "Wednesday",  
    "Thursday",  
    "Friday",  
    "Saturday"
```

```
{ - משתנה סטטי ששומר את ימי השבוע במערך
```

```
int intDay; - יום השיעור במספר  
String day; - יום השיעור במילה  
String time; - זמן השיעור בטקסט  
String lessonID; - זיהוי של הכיתה בפירסטר
```


מחלקה LessonsListAdapter

(מחלקה פנימית ב-ClassDetailsActivity)

רשימה של השיעורים באותה כיתה - `ArrayList<Lesson> lessons;`
`public LessonsListAdapter.LessonsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)` - בצורה אינדיבדואלית ברשימה view פעולה אשר יוצרת כל
`public void onBindViewHolder(@NonNull LessonsListAdapter.LessonsViewHolder holder, int position)`
לפי הנתונים שלו view פעולה אשר מגדירה כל
`public int getItemCount()` - פעולה המחזירה את כמות הפריטים ברשימה

מחלקה LessonsListViewHolder

(מחלקה פנימית ב-LessonsListAdapter)

טקסט של יום השיעור - TextView day;

טקסט של זמן השיעור - TextView time;

פעולה המגדירה את משתני המחלקה - public StudentsListInClassViewHolder(@NonNull View itemView)

מחלקה ClassesTActivity

קישור לפיירסטור - `Firestore db = Firestore.getInstance();`

קישור למידע על המשתמש - `SharedPreferences UserData;`

הארגון אילו משוייך המשתמש - `String org;`

שם המשתמש - `String UserName;`

רשימת הכיתות עם האלמנטים - `RecyclerView classesList;`

כפתור המפנה להוספת כיתה - `Button addClass;`

ומגדירה את המשתנים Activity פעולה היוצרת את ה - `protected void onCreate(Bundle savedInstanceState)`

פעולה המעדכנת את רשימת השיעורים - `private void updateClassesList()`

מחלקה ClassAdapter

(מחלקה פנימית בClassesTActivity)

```
ArrayList<String> classes; - המזהים של הכיתות אילן המורה משויך -  
String org; - הארגון אילו משויך המשתמש -  
private Context context; - הcontext שממנו נשלח ה Adapter  
public ClassAdapter.ClassViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int  
viewType) - בצורה אינבדואלית ברשימה view פעולה אשר יוצרת כל -  
public void onBindViewHolder(@NonNull ClassAdapter.ClassViewHolder holder, int position)  
לפי הנתונים שלו view פעולה אשר מגדירה כל  
public int getItemCount() - פעולה המחזירה את כמות הפריטים ברשימה -
```

מחלקה ClassViewHolder

(מחלקה פנימית ב-ClassAdapter)

Button Homework; - כפתור להוספת שיעורי בית -

Button Test; - כפתור להזנת נתונים על מבחן -

TextView Subject; - טקסט שבו מצויין נושא השיעור -

פעולה המגדירה את משתני המחלקה - `public ClassViewHolder(@NonNull View itemView)`

מחלקה HomePageActivity

FirebaseFirestore db = FirebaseFirestore.getInstance(); - קישור לפיירסטור
 SharedPreferences UserData; - קישור לפרטי התלמיד
 int day; - מספר המייצג את היום
 Calendar calendar = Calendar.getInstance(); - קישור ליומן המובנה למציאת היום
 AlarmManager alarmManager; - כדי ליצור התראה
 ArrayList<String> Classes = new ArrayList<>(); - רשימת הכיתות בהם המשתמש רשום
 ArrayList<Lesson> lessons = new ArrayList<>(); - רשימת השיעורים המתרחשים באותו יום
 String org; - הארגון אליו משויך המשתמש
 String tORs; - סוג המשתמש
 String UserName; - שם המשתמש
 TextView Name; - תיבת טקסט שתכיל את שם המשתמש
 RecyclerView timetable; - תיבה בה ירשמו כל השיעורים שיש לתלמיד באותו היום
 Button SetNotification; - כפתור להגדרת התראות
 protected void onCreate(Bundle savedInstanceState) - ומגדירה את המשתנים Activity פעולה היוצרת את ה-
 private void setAlarm() - פעולה המגדירה את ההתראות לכל השיעורים
 public void updateTimeTable() - פעולה המעדכנת את לוח השעות באותו יום

מחלקה Lesson

(מחלקה פנימית בHomePageActivity)

נושא השיעור - String subject;

זמן תחילת השיעור - Time Start;

זמן סוף השיעור - Time End;

מזהה הכיתה בפירבייס - String ClassroomID;

מחלקה LessonAdapter

(מחלקה פנימית בHomePageActivity)

רשימת השיעורים - `ArrayList<Lesson> lessons;`

סוג המשתמש - `String tORs;`

פעולה - `public LessonViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)` -
בצורה אינדיקטוראלית ברשימה view אשר יוצרת כל

`public void onBindViewHolder(@NonNull LessonViewHolder holder, int position)`

לפי הנתונים שלו view פעולה אשר מגדירה כל

פעולה המחזירה את כמות הפריטים ברשימה - `public int getItemCount()`

מחלקה LessonViewHolder

(מחלקה פנימית בLessonAdapter)

```
public TextView className; משתנה השומר את שם השיעור  
public TextView Time; משתנה השומר את זמן השיעור -  
public LessonViewHolder(@NonNull View itemView) - פעולה המגדירה את משתני המחלקה -
```

מחלקה LessonSFragment

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`
קישור למידע על המשתמש - `SharedPreferences UserData;`
הארגון אליו משוייך המשתמש - `String org;`
מזהה הכיתה בפיירסטור - `String ClassID;`
תיבת טקסט בה מופיע שם השיעור - `TextView TVSubject;`
כפתור המפנה לרשימת המבחנים - `Button tests;`
כפתור המפנה לרשימת שיעורי הבית - `Button Homeworks;`
`public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)`
פעולה אשר מקימה את הfragment לפי הדרישות

מחלקה LessonTActivity

```

FirebaseFirestore db = FirebaseFirestore.getInstance(); - קישור לפיירסטור
FirebaseDatabase database = FirebaseDatabase.getInstance(); - קישור לrealtimeDatabase
DatabaseReference myRef; - realtimeDatabase רפרנס ב
SharedPreferences UserData; - קישור למידע על המשתמש
Timestamp lessonTime; - היום בו מתרחש השיעור
ArrayList<Student> students = new ArrayList<>(); - רשימת התלמידים הרשומים בכיתה
StudentsListAdapter studentsListAdapter; - Adapter בשביל רשימת התלמידים
String org; - הארגון אילו משויך המשתמש
String SubjectName; - המקצוע בו מתרחש השיעור
String ClassroomID; - (וגם ב - realtimeDatabase) הכיתה בפיירסטור
RecyclerView studentsList; - תיבה בה תוצג רשימת התלמידים
TextView Subject; - תיבה בה יוצג הנושא
Button update; - כפתור לאישור המידע
Button findStudents; - כפתור להזנת נוכחות אוטומטית של התלמידים
protected void onCreate(Bundle savedInstanceState) - Activity פעולה היוצרת את ה
/* This method find the teachers location and send it to all of the students by
that the user find the distance between him and the teacher and returns if the
student is in reasonable distance from the teacher and if it is the teacher set
the student as Attendance in lesson */
FusedLocationProviderClient fusedLocationClient;
public void getLocation()
public void updateStudents() - פעולה המעדכנת את רשימת התלמידים בכיתה
public void addAttendanceToStudent(String student) - פעולה אשר מקבלת תלמיד ומעדכנת אותו כנוכח בשיעור

```

מחלקה Student

(מחלקה פנימית בLessonTActivity)

שם התלמיד - String Name;

משתנה בולאני האם התלמיד נוכח בשיעור - Boolean Attendance;

משתנה בולאני האם התלמיד עשה ש"ב Boolean DidHomeWork;

מחלקה StudentsListAdapter

(מחלקה פנימית ב-LessonTActivity)

```
static ArrayList<Student> students; - רשימה הכוללת את התלמידים בכיתה -  
String org; - הארגון אילו משוייכים התלמידים -  
static ArrayList<StudentsListAdapter.StudentsListViewHolder> holders  
רשימה הכוללת בתוכה את הView של התלמידים (מכיוון והרשימה סטטית זה מאפשר שינוי של View לאחר שהוא נבנה)  
public StudentsListAdapter.StudentsListViewHolder onCreateView(@NonNull ViewGroup  
parent, int viewType) - בצורה אינדיבדואלית ברשימה view פעולה אשר יוצרת כל -  
public void onBindViewHolder(@NonNull StudentsListAdapter.StudentsListViewHolder holder, int  
position)  
לפי הנתונים שלו view פעולה אשר מגדירה כל  
public int getItemCount() - פעולה המחזירה את כמות הפריטים ברשימה -
```

מחלקה StudentsListViewHolder

(מחלקה פנימית ב StudentsListAdapter)

```
public TextView Name; - שם התלמיד  
public Button Homework; - כפתור המעדכן שיעורי בית  
public Button Attendance; - כפתור המעדכן נוכחות  
public StudentsListViewHolder(@NonNull View itemView) - פעולה המגדירה את משתני המחלקה
```

מחלקה MainActivity

פעולה הסוגרת את האפליקציה ברגע שהמשתמש חוזר למסך - `protected void onResume()`

פעולה המפנה את המשתמש אל מסך הכניסה של - `protected void onCreate(Bundle savedInstanceState)`
האפליקציה

מחלקה NotificationReceiver

משתנה השומר את המקום בו מופיעה ההתראה - `private static final String CHANNEL_ID = "my_channel";`
פעולה אשר מעלה התראה - `public void onReceive(Context context, Intent intent)`

מחלקה ShowDisturbanceActivity

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`
קישור למידע על המשתמש - `SharedPreferences UserData;`
הארגון אילו שייך המשתמש - `String org;`
סוג ההפרעה המבוקשת - `String disturbance;`
שם התלמיד שעליו מוצגים האירועים - `String name;`
מזהה הכיתה בפיירסטור - `String classID;`
תיבת הטקסט בו מוצג שם התלמיד - `TextView TVName;`
תיבת הטקסט בו מוצג סוג האירוע - `TextView TVDisturbance;`
התיבה בה יוצגו האירועים - `RecyclerView disturbanceList;`
ומגדירה את המשתנים Activity פעולה היוצרת את ה - `protected void onCreate(Bundle savedInstanceState)`
פעולה המעדכנת את רשימת ההפרעות - `private void updateDisturbanceList()`

מחלקה DisturbanceListAdapter

(מחלקה פנימית ב-ShowDisturbanceActivity)

`ArrayList<String> subjects = new ArrayList<>();` - רשימה הכוללת את הנושאים בהם בוצעה הפרעה -

`ArrayList<String> dates = new ArrayList<>();` - רשימה הכוללת את התאריכים בהם בוצעה ההפרעה -

(שני הרשימות מתואמות במיקומן)

`public DisturbanceViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)` - בצורה אינדיקטורית ברשימה view פעולה אשר יוצרת כל -

`public void onBindViewHolder(@NonNull DisturbanceViewHolder holder, int position)`

לפי הנתונים שלו view פעולה אשר מגדירה כל

`public int getItemCount()` - פעולה המחזירה את כמות הפריטים ברשימה -

מחלקה DisturbanceViewHolder

(מחלקה פנימית ב-DisturbanceListAdapter)

טקסט הכולל את הנושא בו הייתה הפרעה - `TextView subject;`

טקסט הכולל את התאריך בו התרחשה ההפרעה - `TextView date;`

פעולה המגדירה את משתני המחלקה - `public StudentsViewHolder(@NonNull View itemView)`

מחלקה SignInActivity

FirebaseFirestore db = FirebaseFirestore.getInstance(); - קישור לפיירסטור
 FirebaseAuth mAuth = FirebaseAuth.getInstance(); - קישור ל-Authentication
 SharedPreferences savedUserData; - קישור למידע על המשתמש
 SharedPreferences.Editor UserData; - קישור לעורך המידע על המשתמש
 String UID; - Authentication המזהה שניתן למשתמש על ידי ה-
 String org; - הארגון אילו משוייך המשתמש
 String tORs; - סוג המשתמש
 String Name; - שם המשתמש
 TextView Email; - תיבת הטקסט בה מוזן המייל
 TextView Password; - תיבת הטקסט בה תוזן הסיסמה
 Button enter; - כפתור האמשר את פרטי הכניסה
 Button signUp; - כפתו לפתיחת חשבון
 ומגדירה את המשתנים Activity פעולה היוצרת את ה- onCreate(Bundle savedInstanceState)

מחלקה SignUpActivity

קישור לפיירסטור - `FirebaseFirestore db = FirebaseFirestore.getInstance();`
קישור ל- Authentication `FirebaseAuth mAuth = FirebaseAuth.getInstance();`
קישור לעורך המידע על המשתמש - `SharedPreferences.Editor UserData;`
מזהה המשתמש שניתן על ידי Authentication - `String userID;`
כפתור המאשר את הפרטים וההרשמה - `Button SignUp;`
סוויצ' המחליט על סוג המשתמש - `Switch Teacher_Student;`
תיבת טקסט בו המשתמש יזין את שם המשתמש שלו - `EditText Name;`
תיבת הטקסט בה המשתמש יזין את המייל שלו - `EditText ETEmail;`
תיבת הטקסט בו המשתמש יזין את הסיסמה שלו - `EditText Password1;`
תיבת הטקסט בה המשתמש יאמת את הסיסמה שלו - `EditText Password2;`
תיבת הטקסט שבה במשתמש יזין את הארגון אליו הוא משויך - `EditText Organization;`
ומגדירה את המשתנים Activity פעולה היוצרת את ה- `protected void onCreate(Bundle savedInstanceState)`

מחלקה User

(מחלקה פנימית בSignUpActivity)

משתנה השומר את האררגון של המשתמש - `public String email;`
משתנה השומר את שם המשתמש - `public String name;`
משתנה השומר את סוג המשתנה - `public String stu_teach;`

מחלקה StudentClassesActivity

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`
 קישור למידע על המשתמש - `SharedPreferences UserData;`
 רשימת הכיתות אליהן משויך המשתמש - `ArrayList<String> classes;`
 הארגון אילו משויך המשתמש - `String org;`
 שם המשתמש - `String UserName;`
 תיבה התציג את הכיתות שהמשתמש משויך אליהן - `RecyclerView classesList;`
 כפתור המפנה אל רשימת החיסורים הכללית - `Button unShow;`
 כפתור המפנה אל רשימת אי הבאת שיעורי בית כללית - `Button noHomework;`
`View.OnClickListener disturbanceOnClick`
`ShowDisturbanceActivity` ומעביר `noHomework` מאזין לכפתורים, `unShow`
`private void updateClassesList()` - פעולה המעדכנת את רשימת הכיתות
`protected void onCreate(Bundle savedInstanceState)` - ומגדירה את המשתנים `Activity` פעולה היוצרת את ה-

מחלקה classesAdapter

(מחלקה פנימית ב-StudentClassesActivity)

פעולה - onCreateViewHolder(@NonNull ViewGroup parent, int viewType) ב-classesViewHolder

בצורה אינדוקטואלית ברשימה view אשר יוצרת כל

public void onBindViewHolder(@NonNull classesViewHolder holder, int position)

לפי הנתונים שלו view פעולה אשר מגדירה כל

פעולה המחזירה את כמות הפריטים ברשימה - getItemCount() public int

מחלקה ViewHolder

(מחלקה פנימית בclassesAdapter)

תיבה הבה רשום שם הכיתה - `TextView className;`
פעולה המגדירה את משתני המחלקה - `public ViewHolder(@NonNull View itemView)`

מחלקה StudentHomeworksActivity

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`
קישור למידע על המשתמש - `SharedPreferences UserData;`
קישור לארגון אילו משוייך המשתמש - `String org;`
שם המשתמש - `String UserName;`
מזהה בכיתה בפיירסטור - `String ClassID;`
תיבה בה יוצג שם המשתמש - `TextView Name;`
תיבה בה תוצג רשימת שיעורי הבית של התלמיד - `RecyclerView homeworksList;`
פעולה אשר מעדכנת את רשימת שיעורי הבית לתלמיד - `private void updateHomeworksList()`
ומגדירה את המשתנים Activity פעולה היוצרת את ה - `protected void onCreate(Bundle savedInstanceState)`

מחלקה Homework

(מחלקה פנימית בStudentHomeworksActivity)

זמן ההגשה של הש"ב - Timestamp TimeDate;

זמן ההגשה של הש"ב בטקסט - String date;

הפרטים על שיעורי הבית - String text;

מחלקה HomeworksAdapter

(מחלקה פנימית ב-StudentHomeworksActivity)

רישמת שיעורי הבית שיש להציג - `ArrayList<Homework> homeworks;`
- `public HomeworkViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)` -
בצורה אינדבדואלית ברשימה view פעולה אשר יוצרת כל
`public void onBindViewHolder(@NonNull HomeworkViewHolder holder, int position)`
לפי הנתונים שלו view פעולה אשר מגדירה כל
`public int getItemCount()` - פעולה המחזירה את כמות הפריטים ברשימה

מחלקה HomeworksViewHolder

(מחלקה פנימית ב-HomeworksAdapter)

TextView text; - תיבת טקסט בה תוצג הפרטים על שיעורי הבית -

TextView date; - תיבת הטקסט בה יוצג התאריך להגשה -

public class ViewHolder(@NonNull View itemView) - פעולה המגדירה את משתני המחלקה -

מחלקה StudentTestsActivity

קישור לפיירסטור - `Firestore db = FirebaseFirestore.getInstance();`

קישור למידע על המשתמש - `SharedPreferences UserData;`

הארגון אילו משוייך המשתמש - `String org;`

שם התלמיד עליו מוצגים הנתונים - `String UserName;`

מזהה הכיתה בפיירסטור - `String ClassID;`

תיבה בה יוצג שם התלמיד - `TextView Name;`

תיבה בה יוצגו כלל המבחנים של התלמיד באותו נושא - `RecyclerView testsList;`

פעולה המעדכנת את רשימת המבחנים - `private void updateTestsList()`

ומגדירה את המשתנים Activity פעולה היוצרת את ה - `protected void onCreate(Bundle savedInstanceState)`

מחלקה TestOutput

(יורשת Test, מחלקה פנימית בStudentTestsActivity)

התאריך של המבחן במחרוזת - String date;

מחלקה TestAdapter

(מחלקה פנימית ב-StudentTestsActivity)

רשימת המבחנים שיש להציג - `ArrayList<TestOutput> tests;`
פעולה - `public TestViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)` -
בצורה אינדיבדואלית ברשימה `view` אשר יוצרת כל
`public void onBindViewHolder(@NonNull TestViewHolder holder, int position)`
לפי הנתונים שלו `view` פעולה אשר מגדירה כל
פעולה המחזירה את כמות הפריטים ברשימה - `public int getItemCount()`

מחלקה ViewHolder

(מחלקה פנימית ב-TestAdapter)

```
public TextView testName; - שם המבחן  
public TextView Grade; - הציון במבחן  
public TextView subject; - נושא המבחן  
public TextView date; - התאריך של המבחן  
public TextView teachersName; - שם המורה הבוחן  
public class ViewHolder(@NonNull View itemView) - פעולה המגדירה את משתני המחלקה
```

מחלקה Test

String TestName; - שם המבחן -

int grade; - הציון במבחן -

String Subject; - נושא המבחן -

String teachers_name; - שם המורה הבוחן -

מחלקה Time

שעה - `int Hour;`

דקות - `int Minutes;`

`public String toString()` - פעולה המחזירה את השעה בפורמט תקין (xx:xx)

`public Calendar subtractMinutes(int minutes)` - בהתאם `Calendar` פעולה המחסירה מספר דקות ומחזירה -

מחלקה UserLocation

double Latitude; - מיקום Latitude

double Longitude; - מיקום Longitude

public boolean isWithinRange(UserLocation location) -

פעולה המחזירה האם המרחק בין המיקום הנ"ל והמיקום שהתקבל קטן מ-15 מטרים

שמירת נתונים

קבצים

Shared Preferences

UserData - שומר את הנתונים על המשתמש משמש בשביל כניסה אוטומטית וגישה לנתונים בFireBase

- org - שומר את הארגון אליו משוייך המשתמש
- Name - שומר את שם המשתמש
- tORs - שומר את סוג המשתמש (Teacher / Student)

סכמת Database

🏠 > Users > AXsYDxqPUXUR More in Google Cloud		
educarebase	Users	AXsYDxqPUXURHV2KBxMFRrNJMF2
+ Start collection	+ Add document	+ Start collection
Users >	AXsYDxqPUXURHV2KBxMFRrNJMF2 >	+ Add field
organizations	nWtNX8YmVbSgfg0WzeTLQyBVSSr2	Name: "hezi" org: "kn" tORs: "Teacher"

🏠 > organizations > kn More in Google Cloud		
educarebase	organizations	kn
+ Start collection	+ Add document	+ Start collection
Users	kn >	Classes Lessons Student Teacher
organizations >		+ Add field
		This document does not exist, it will not appear in queries or snapshots Learn more

Home > organizations > kn > Classes > A3WGk4eFJJT2...

More in Google Cloud

kn	Classes	A3WGk4eFJJT2DLhFVbE1
+ Start collection Classes Lessons Student Teacher + Add field	+ Add document A3WGk4eFJJT2DLhFVbE1 AKiCZ6JSWGq155UIHgie JTeh0zfz2Cm0cojwEGge K0UG2fi1G68ywnr6PxWn SouYGZKLEr0S6IT2JIKC SvpXgsyQ5nMauuCDJqQ4 dimd0DgAaF0EuWpa4nn1 e0cx0p0sQ0uHYTrN3SBr eRAmCVgNWpz6pLR8Kbo9 m0lNEw0mHOKfVGioGuvr pmbUms06E5Hm7cQycrSJ sq51Xy0dzLRw1q3B5jbc	+ Start collection Lessons + Add field Days: [true, false, false, false...] Students 0 "jonatan" Subject: "hhh" teacher: "hezi"

This document does not exist, it will not appear in queries or snapshots
[Learn more](#)

Home > ... > Classes > A3WGk4eFJJT2... > Lessons > 1

More in Google Cloud

A3WGk4eFJJT2DLhFVbE1	Lessons	1
+ Start collection Lessons + Add field Days 0 true 1 false 2 false 3 false 4 false 5 false 6 false Students 0 "jonatan"	+ Add document 1	+ Start collection + Add field lessonsID 0 "TDenni7eT0dtieCslXVP"

Home > organizations > kn > Lessons > 6oGEx3FuG9mP More in Google Cloud		
<div>kn</div> <div> + Start collection </div> <div> Classes </div> <div> Lessons </div> <div> Student </div> <div> Teacher </div> <div> + Add field </div> <div> This document does not exist, it will not appear in queries or snapshots Learn more </div>	<div>Lessons</div> <div> + Add document </div> <div> 6oGEx3FuG9mPDIDLX1L1 </div> <div> 9b90y4cj6v1tZocf0jJv </div> <div> GX4B41cIBY1reXNI9N5r </div> <div> LLewH3iTdEJJxWj9gEzu </div> <div> TDennI7eT0dtieCsIXVP </div> <div> V5I2tMcIkfKB1SW0zKoK </div> <div> Zvb6CajKE0pkv1VREyxd </div> <div> cGVbb2X0G05dPgL17sZ8 </div> <div> gVXzfnJaQhySHLzG6MPj </div> <div> gkvPEPF1JOHd3sqatIsw </div> <div> uAmihFeCPM9eEWuEUJkR </div> <div> uju5rCBsUUGmAsvcimPR </div> <div> zaA1jj7VEFSG10F1jvp0 </div>	<div>6oGEx3FuG9mPDIDLX1L1</div> <div> + Start collection </div> <div> + Add field </div> <div> endHour: 20 </div> <div> endMinute: 10 </div> <div> startHour: 3 </div> <div> startMinute: 15 </div>

Home > organizations > kn > Student > jonatan More in Google Cloud		
<div>kn</div> <div> + Start collection </div> <div> Classes </div> <div> Lessons </div> <div> Student </div> <div> Teacher </div> <div> + Add field </div> <div> This document does not exist, it will not appear in queries or snapshots Learn more </div>	<div>Student</div> <div> + Add document </div> <div> jonatan </div>	<div>jonatan</div> <div> + Start collection </div> <div> + Add field </div> <div> Tests </div> <div> disturbance </div> <div> Classes: ["sq51Xy0dzLRw1q3B5jbc", "..."] </div> <div> email: "jonatan@gmail.com" </div> <div> name: "jonatan" </div> <div> stu_teach: "Student" </div>

Home > ... > Student > jonatan > Tests > first test [More in Google Cloud](#)

jonatan	Tests	first test
+ Start collection	+ Add document	+ Start collection
Tests > disturbance	first test > ghh test1	+ Add field
+ Add field		classID: "sq51Xy0dzLRw1q3B5jbc" (string) date: June 14, 2023 at 3:00:00 AM UTC+3 grade: 33 subject: "cs" teachers_name: "hezi" testName: "first test"

Classes

- 0 "sq51Xy0dzLRw1q3B5jbc"
- 1 "JTehOzfz2Cm0cojwEGge"
- 2 "SvpxgsyQ5nMauuCDJqQ4"
- 3 "SouYGZKLEr0S6IT2JIKC"
- 4 "mOlNEw0mHOKfVGioGuvr"
- 5 "eOcxOp0sQOuHYTrN3SBr"
- 6 "KOUg2fi1G68ywnr6PxWn"
- 7 "eRAmCVgNWpz6pLR8Kbo9"
- 8 "A3WGk4eFJT2DLhFVbE1"

Home > ... > Student > jonatan > disturbance > noHomeWork [More in Google Cloud](#)

jonatan	disturbance	noHomeWork
+ Start collection	+ Add document	+ Start collection
Tests disturbance >	noHomeWork >	+ Add field
+ Add field		Class 0 "JTehOzfz2Cm0cojwEGge" date 0 June 14, 2023 at 3:30:00 PM UTC+3

Classes

- 0 "sq51Xy0dzLRw1q3B5jbc"
- 1 "JTehOzfz2Cm0cojwEGge"
- 2 "SvpxgsyQ5nMauuCDJqQ4"
- 3 "SouYGZKLEr0S6IT2JIKC"
- 4 "mOlNEw0mHOKfVGioGuvr"
- 5 "eOcxOp0sQOuHYTrN3SBr"
- 6 "KOUg2fi1G68ywnr6PxWn"
- 7 "eRAmCVgNWpz6pLR8Kbo9"
- 8 "A3WGk4eFJT2DLhFVbE1"

🏠 > organizations > kn > Teacher > hezi
[More in Google Cloud](#)

kn	Teacher	hezi
+ Start collection Classes Lessons Student Teacher	+ Add document hezi	+ Start collection + Add field ▼ Classes <pre> 0 "eRamCVgNWpz6pLR8Kbo9" 1 "A3WGk4eFJt2DLhFVbE1" 2 "AKICZ6JSWqI55UIHgle" email: "hezi@gmail.com" name: "hezi" stu_teach: "Teacher"</pre>
<p>This document does not exist, it will not appear in queries or snapshots Learn more</p>		

Search by email address, phone number, or user UID
 Add user

Identifier	Providers	Created ↓	Signed In	User UID
jonatan@gmail.com		Jun 14, 2023	Jun 18, 2023	nWtNX0YmnbSgfgOWzeTLQyBVS...
hezi@gmail.com		Jun 14, 2023	Jun 18, 2023	AXsYDxqpUXURHV2KBxMFRrNJ...

Rows per page: 50 1 – 2 of 2

<https://educarebase-default-rtadb.firebaseio.com>

```

https://educarebase-default-rtadb.firebaseio.com/
├── kn
│   └── JTeh0zfz2Cm0cojwEGge
│       ├── Students
│       │   └── jonatan: "jonatan"
│       └── Teacher
│           ├── latitude: 0
│           └── longitude: 0
```


רפלקציה אישית

במהלך הפרוייקט למדתי המון על עקרונות תכנות למדתי בעיקר איך ללמוד פעילויות של מתודות לבד ולנצל את השימוש בהם בצורה הטובה ביותר.

במהלך הפרוייקט התמודדתי עם המון אתגרים בעיקר בשימוש בספריות חדשות שאינני מכיר וזה גרם לי ללמוד יותר על דרכי פיתרון מקוריות. וכתבת שאילתות בצורה ברורה ומובנת

אני כנראה אקח להמשך את היכולת שלי "לעבוד עם האינטרנט" שהוא חברי הטוב ביותר מאז שהתחלתי את הפרוייקט ואני מאוד מרוצה מהתוצר

בנוסף אני חושב שאני ארגיש יותר בנוח לבקש עזרה מאנשים שהתמודדו עם אותם בעיות כמוני כמו עמיתי לכיתה ומוריי או מתכנתים רגילים שאני מכיר.

בראיה לאחור הייתי מנסה לעבוד יותר על הפן העיצובי בפרוייקט ובנוסף לכתוב את הקוד מלכתכילה מסודר (לא כמו שעשיתי שכתבתי מחלקה ואז סידרתי אותה)

אם היה לי עוד עזרה מעמיתים כנראה הייתי משקיע יותר בעיצוב הפרוייקט ומנסה לקבל יותר חוות דעת מאנשים בתחום בנוגע לצרכים שלהם שניתן לפתור בצורה טכנולוגית.

אני חושב שהעבודה על הפרוייקט גרמה לי להרגיש יותר בנוח עם פלטפורמות תכנות נוספות ומאז שסיימתי אותו הספקתי לקחת על עצמי פרוייקטים תכנותיים נוספים שהרגשתי בהם צורך.

אני חושב שהפרוייקט כלל בתוכו המון אלמנטים קטנים שאני חושב שיוכלו להועיל לי בעתיד וגרמו לי להכיר שימוש באמצעים שלא ידעתי גם אם בסוף לא נכנסו לפרוייקט עצמו.

אני מרגיש שהעבודה ע הפרוייקט והדרישות גרמו לי לפתח דרך חשיבה שונה בנושאים תכנותיים רבים ורגמו לי להיחשף ליכולת שלי להשתמש בספריות נוספות.

ביבליוגרפיה

app school - <https://appschool.co.il/>

אנדרואיד לבגרות - <https://www.youtube.com/channel/UCyuP9YBpsg4Qq7fMF-IOTJQ?app=desktop>

ChatGPT - <https://chat.openai.com/>

Coding in Flow - <https://www.youtube.com/@codinginflow>

Codetutor - <https://www.youtube.com/@Codetutor-DemystifyCoding>

stack overflow - <https://stackoverflow.com/>

נספחים**קוד המקור (source code)****מחלקה AddClassFragment**

```

public class AddClassFragment extends Fragment {

                                //General                data                variables
*****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();
    //keyboard
    InputMethodManager imm;
    //ProgressDialog variables
    ProgressDialog progressDialog;

    //User variables
    static String org;
    static String userName;

                                //General                data                variables
*****

    static ArrayList<Boolean> days = new ArrayList<>(Collections.nCopies(7,
false));
    static String subjectText;

    //Views
    EditText subject;
    Button add;
    static Button addStudents;
    Button addLesson;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_add_class, container,
false);

                                //getting                general                vars
*****

                                imm                =                (InputMethodManager)
getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
        // getting data from bundle
        Bundle args = getArguments();
        if (args != null) {

```

```

// the fragment opened to add class not to add data
org = args.getString("org");
userName = args.getString("user name");

//delete all the lists
days = new ArrayList<>(Collections.nCopies(7, false));
lessons = new ArrayList<>();
Students = new ArrayList<>();
subjectText = null;
addStudents = null;
}

progressDialog = new ProgressDialog(this.getContext());
progressDialog.setTitle("Loading");
progressDialog.setMessage("adding the class");
progressDialog.setCancelable(false);

//getting general vars
*****

//Find views
subject = view.findViewById(R.id.FRETSsubject);
addStudents = view.findViewById(R.id.FRBTNStudents);
addLesson = view.findViewById(R.id.FRBTNLesson);
add = view.findViewById(R.id.FRBTNAddClass);

//Sets views
if (subjectText != null) subject.setText(subjectText);
subject.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int
count, int after) {}
    @Override
    public void onTextChanged(CharSequence s, int start, int before,
int count) {}
    @Override
    public void afterTextChanged(Editable s) {
        subjectText = s.toString();
    }
});

addStudents.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Close the keyboard
        imm.hideSoftInputFromWindow(subject.getWindowToken(), 0);

        //set button enabled
        addStudents.setEnabled(false);

```

```

        //Open the AddStudentToClassFragment
        AppCompatActivity activity = (AppCompatActivity)
view.getContext();

        AddStudentToClassFragment fragment = new
AddStudentToClassFragment();
        activity.getSupportFragmentManager().beginTransaction()
            .replace(R.id.FContainerClassesT, fragment)
            .addToBackStack(null).commit();
    }
});

addLesson.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Close the keyboard
        imm.hideSoftInputFromWindow(subject.getWindowToken(), 0);

        //Open the AddLessonFragment
        AppCompatActivity activity = (AppCompatActivity)
view.getContext();
        AddLessonFragment fragment = new AddLessonFragment();
        activity.getSupportFragmentManager().beginTransaction()
            .replace(R.id.FContainerClassesT, fragment)
            .addToBackStack(null).commit();
    }
});

add.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Close the keyboard
        imm.hideSoftInputFromWindow(subject.getWindowToken(), 0);

        //check for valid input
        if (subject.getText().toString().equals("")) {
            Toast.makeText(getActivity(), "need to enter subject",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (Students.size() == 0) {
            Toast.makeText(getActivity(), "you must have student in
class", Toast.LENGTH_SHORT).show();
            return;
        }

        addDataToFirestore();
    }
}

```

```

    });

    return view;
}

//Adding all the needed data to firestore
public void addDataToFirestore(){
    progressDialog.show();

    Map<String, Object> classData = new HashMap<>();
    classData.put("teacher", userName);
    classData.put("Students", Students);
    classData.put("Subject", subject.getText().toString());
    classData.put("Days", days);

    db.collection("organizations").document(org).collection("Classes")
        .add(classData)
        .addOnSuccessListener(new
    OnSuccessListener<DocumentReference>() {
        @Override
        public void onSuccess(DocumentReference documentReference)
    {
        String documentId = documentReference.getId();

        //add class to the teacher
        db.collection("organizations").document(org)

        .collection("Teacher").document(userName).get()
        .addOnSuccessListener(new
    OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot
    documentSnapshot) {
        ArrayList<String> classes =
    (ArrayList<String>) documentSnapshot.get("Classes");
        if (classes == null)
            classes = new ArrayList<>();
        classes.add(documentId);

        db.collection("organizations").document(org)

        .collection("Teacher").document(userName)
        .update("Classes", classes);
        closeFragment(0);
        }
    });
}

```

```

        //add class to the students
        CollectionReference studentsColRef =
db.collection("organizations").document(org).collection("Student");
        for (int i = 0; i < Students.size(); i++) {
            int finalI = i;
            studentsColRef.document(Students.get(i)).get()
                .addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                @Override
                public void onSuccess(DocumentSnapshot
documentSnapshot) {
                    ArrayList<String> classes =
(ArrayList<String>) documentSnapshot.get("Classes");
                    if (classes == null)
                        classes = new ArrayList<>();
                    classes.add(documentId);

studentsColRef.document(Students.get(finalI)) //crash in the 'get' command
                    .update("Classes",
classes);

                    if (finalI == Students.size() -
1)

                        closeFragment(1);

                }
            });
        }

        //add lesson to firebase
        /*need to use thread to prevent from case when the
fragment is closed before all data has been uploaded*/
        new Thread(new Runnable() {
            @Override
            public void run() {
                //add lesson col to firebase
                CollectionReference LessonsColRef =
documentReference.collection("Lessons");
                for (int i = 0; i < 7; i++) {
                    if (days.get(i)) {
                        LessonsColRef
                            .document((i + 1) + "")
                            .set(new HashMap<>());
                    }
                }
                final boolean[] j = {false};
                for (int i = 0; i < lessons.size(); i++) {
                    while (j[0]) ;
                    j[0] = true;

```

```

Map<String, Object> lessonData = new
HashMap<>();
        lessonData.put("endHour",
lessons.get(i).end.getHour());
        lessonData.put("endMinute",
lessons.get(i).end.getMinutes());
        lessonData.put("startHour",
lessons.get(i).start.getHour());
        lessonData.put("startMinute",
lessons.get(i).start.getMinutes());

        int finalI = i;

db.collection("organizations").document(org).collection("Lessons")
        .add(lessonData)
        .addOnSuccessListener(new
OnSuccessListener<DocumentReference>() {
            @Override
                public void
onSuccess(DocumentReference documentReference) {
                    String documentId =
documentReference.getId();

LessonsColRef.document(lessons.get(finalI).day + "").get()

        .addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
                public void
onSuccess(DocumentSnapshot documentSnapshot) {

ArrayList<String>          lessonsID          =          (ArrayList<String>)
documentSnapshot.get("lessonsID");
                                                                    if
(lessonsID == null)

lessonsID = new ArrayList<>();

lessonsID.add(documentId);

LessonsColRef.document(lessons.get(finalI).day + "")

        .update("lessonsID", lessonsID);
                                                                    if (finalI
== lessons.size() - 1)

closeFragment(2);

```



```

false;
j[0] =

}
});
}
});
}
}
}).start();
}
});
}

//To reset the static variables when fragment is closed;
boolean[] firestoreActions = new boolean[]{false, false, false};
/**
 * 0 - add class to the teacher
 * 1 - add class to the students
 * 2 - add lesson to firebase
 **/
public void closeFragment(int code) {
    firestoreActions[code] = true;
    for (int i = 0; i < firestoreActions.length; i++) {
        if (!firestoreActions[i]) return;
    }

    progressDialog.dismiss();

    days = new ArrayList<>(Collections.nCopies(7, false));
    lessons = new ArrayList<>();
    Students = new ArrayList<>();
    subjectText = null;
    userName = null;
    org = null;
    addStudents = null;

    FragmentManager fragmentManager =
requireActivity().getSupportFragmentManager();

fragmentManager.beginTransaction().remove(AddClassFragment.this).commit();
}

//Used to add students from AddStudentToClassFragment
static ArrayList<String> Students = new ArrayList<>();

public static void AddStudent(String student) {

```

```
        Students.add(student);
    }

    //Used to add lesson from AddLessonFragment
    static ArrayList<lesson> lessons = new ArrayList<>();

    public static void AddLesson(int day, int startHour, int startMinute, int
endHour, int endMinute) {
        lessons.add(new lesson(new Time(startHour, startMinute), new
Time(endHour, endMinute), day));
    }

    private static class lesson {
        public Time start;
        public Time end;
        public int day;

        public lesson(Time start, Time end, int day) {
            this.start = start;
            this.end = end;
            this.day = day;
            days.set(day - 1, true);
        }
    }
}
```

מחלקה AddHomeworkFragment

```

package com.example.educare;

import android.app.DatePickerDialog;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.Timestamp;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class AddHomeworkFragment extends Fragment {

                                //General                                data                                variables
    *****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //Activity variables
    int mDate, mMonth, mYear;
    ArrayList<String> Students;

    //User variables

```

```

String org;
String ClassID;
String subject;

//General data variables
*****

//Views
TextView Subject;
TextView dateTXT;
ImageView Cal;
EditText description;
Button Add;

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_add_homework,
container,false);

//getting general vars
*****

//getting data from bundle
Bundle args = getArguments();
if (args != null) {
    subject = args.getString("Subject");
    ClassID = args.getString("ClassID");
    org = args.getString("org");
}

ImportStudentsListFromFirestore();

//getting general vars
*****

//Find views
dateTXT = view.findViewById(R.id.FRTVDate);
Cal = view.findViewById(R.id.FRIVCalendar);
Subject = view.findViewById(R.id.FRTVAddHomeworkSubject);
description = view.findViewById(R.id.FRETHomeworkDescription);
Add = view.findViewById(R.id.FRBTNAddHomework);

//Sets views

Subject.setText(subject);
Cal.setOnClickListener(new View.OnClickListener() {
    //change the date
    @Override

```

```

        public void onClick(View view) {
            final Calendar cal = Calendar.getInstance();
            mDate = cal.get(Calendar.DATE);
            mMonth = cal.get(Calendar.MONTH);
            mYear = cal.get(Calendar.YEAR);

            DatePickerDialog datePickerDialog = new
DatePickerDialog(getContext(), android.R.style.Theme_DeviceDefault_Dialog,
new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker datePicker, int year, int
month, int date){
                dateTXT.setText(date+"/"+month+"/"+year);
                },mYear,mMonth,mDate);
            datePickerDialog.show();
        }
    });

    Add.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //check if all data has been given
            if(description.getText().toString().equals("")){
                Toast.makeText(getActivity(), "you need to add
description", Toast.LENGTH_SHORT).show();
                return;
            }
            if(dateTXT.getText().toString().equals("XX/XX/XXXX")) {
                Toast.makeText(getActivity(), "you need to add submission
date", Toast.LENGTH_SHORT).show();
                return;
            }

            addDataToFirestore();
        }
    });

    return view;
}

//Adding all the needed data to firestore
public void addDataToFirestore(){
    CollectionReference colRef =
db.collection("organizations").document(org)
    .collection("Student");
    for (int i = 0; i< Students.size(); i++){

```

```

        DatabaseReference docRef =
colRef.document(Students.get(i)).collection("Homeworks").document(ClassID);
        docRef.get().addOnCompleteListener(new
OnCompleteListener<DocumentSnapshot>() {
            @Override
            public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                DocumentSnapshot document = task.getResult();
                //importing data from firebase
                ArrayList<String> text = (ArrayList<String>)
document.get("text");
                if (text == null) text = new ArrayList<>();
                ArrayList<Timestamp> date = (ArrayList<Timestamp>)
document.get("date");
                if (date == null) date = new ArrayList<>();

                Calendar calendar = Calendar.getInstance();
                calendar.set(mYear, mMonth, mDate);
                date.add(new Timestamp(calendar.getTime()));
                text.add(description.getText().toString());

                Map<String, Object> updates = new HashMap<>();
                updates.put("text", text);
                updates.put("date", date);

                docRef.set(updates);

                FragmentManager fragmentManager =
requireActivity().getSupportFragmentManager();

fragmentManager.beginTransaction().remove(AddHomeworkFragment.this).commit();
            }
        });
    }

    private void ImportStudentsListFromFirestore() {
        db.collection("organizations").document(org).collection("Classes")
            .document(ClassID).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                Object existingClassesArray =
documentSnapshot.get("Students");
                if (existingClassesArray instanceof List) {
                    Students = (ArrayList<String>)
documentSnapshot.get("Students");
                }
            }
        })
    }

```

```
    );  
  
    }  
}
```

מחלקה AddLessonFragment

```

package com.example.educare;

import android.app.TimePickerDialog;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

public class AddLessonFragment extends Fragment {

    //General data variables
    *****

    //Activity variables

    int day = 0;
    int startHour;
    int startMinute;
    int endHour;
    int endMinute;

    //General data variables
    *****

    //Views
    TextView Start;
    TextView End;
    Button Add;
    Button[] days = new Button[7];

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_add_lesson, container,
        false);

        //Find views
        days[0] = view.findViewById(R.id.FRBTNDay1);

```



```

days[1] = view.findViewById(R.id.FRBTNDay2);
days[2] = view.findViewById(R.id.FRBTNDay3);
days[3] = view.findViewById(R.id.FRBTNDay4);
days[4] = view.findViewById(R.id.FRBTNDay5);
days[5] = view.findViewById(R.id.FRBTNDay6);
days[6] = view.findViewById(R.id.FRBTNDay7);
Start = view.findViewById(R.id.FRTVStart);
End = view.findViewById(R.id.FRTVEnd);
Add = view.findViewById(R.id.FRBTNAddLesson);

//Sets views
for (int i = 0; i<7; i++) days[i].setOnClickListener(dayOnClick);

Start.setOnClickListener(timeOnClick);
End.setOnClickListener(timeOnClick);

Add.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        if(day == 0){
            Toast.makeText(getActivity(), "need to chose a day",
Toast.LENGTH_SHORT).show();
            return;
        }

        //return to AddClassFragment

AddClassFragment.AddLesson(day, startHour, startMinute, endHour, endMinute);
        AppCompatActivity activity = (AppCompatActivity)
view.getContext();
        AddClassFragment fragment = new AddClassFragment();
        activity.getSupportFragmentManager().beginTransaction()
            .replace(R.id.FContainerClassesT, fragment)
            .addToBackStack(null).commit();
    }
});

return view;
}

//Sets start and ent times
OnClickListener timeOnClick = new OnClickListener() {
    int hour, minute;
    @Override
    public void onClick(View view) {
        TimePickerDialog.OnTimeSetListener onTimeSetListener = new
TimePickerDialog.OnTimeSetListener() {
            @Override

```

```

        public void onTimeSet(TimePicker timePicker, int selectedHour,
int selectedMinute) {
            hour = selectedHour;
            minute = selectedMinute;
            Time time = new Time(selectedHour,selectedMinute);
            if (view == Start){
                startHour = hour;
                startMinute = minute;
                Start.setText("Start:  "+time.toString());
            }
            if (view == End){
                endHour = hour;
                endMinute = minute;
                End.setText("End:      "+time.toString());
            }
        }
    };

        TimePickerDialog  timePickerDialog  =  new
TimePickerDialog(getActivity(),onTimeSetListener,hour,minute,true);

        if (view == Start)
            timePickerDialog.setTitle("Start");
        if (view == End)
            timePickerDialog.setTitle("End");
        timePickerDialog.show();
    }
};
//change the lesson day to the day
OnClickListener dayOnClick = new OnClickListener() {
    @Override
    public void onClick(View view) {
        for (int i = 0; i < 7; i++) {
            if (view == days[i]) {
                day = i + 1;

                days[i].setBackgroundColor(ContextCompat.getColor(getActivity(),
R.color.green));
            } else {

                days[i].setBackgroundColor(ContextCompat.getColor(getActivity(),
R.color.purple_500));
            }
        }
    }

};
}

```

AddMenuActivity מחלקה

```
package com.example.educare;

import android.content.Intent;
import android.content.SharedPreferences;
import android.view.Menu;
import android.view.MenuItem;

import androidx.appcompat.app.AppCompatActivity;

import org.checkerframework.checker.nullness.qual.NonNull;

//Its not a real Activity its just for adding the menu
//each Activity that have Menu extending this Activity instead of
AppCompatActivity
public class AddMenuActivity extends AppCompatActivity {

    Intent i;
    SharedPreferences UserData;

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        UserData = getSharedPreferences("UserData", MODE_PRIVATE);

        if (item.getItemId() == R.id.MIHome) {
            i = new Intent(getApplicationContext(), HomePageActivity.class);
            startActivity(i);
            return true;
        } else if (item.getItemId() == R.id.MILogOut) {
            SharedPreferences.Editor EUserData = UserData.edit();
            EUserData.remove("org");
            EUserData.remove("UserName");
            EUserData.remove("tORs");
            EUserData.apply();
            i = new Intent(getApplicationContext(), SignInActivity.class);
            startActivity(i);
            finish(); // Optional: Finish the current activity
            return true;
        } else if (item.getItemId() == R.id.MIClasses) {
            if (UserData.getString("tORs", "not found").equals("Teacher")) {
```

```

                                i = new Intent(getApplicationContext(),
ClassesTActivity.class);
                                startActivity(i);
                                } else {
                                i = new Intent(getApplicationContext(),
StudentClassesActivity.class);
                                startActivity(i);
                                }
                                return true;
                                } else {
                                return super.onOptionsItemSelected(item);
                                }
                                }
                                }
```

מחלקה AddStudentToClassFragment

```

package com.example.educare;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

public class AddStudentToClassFragment extends Fragment {

                                //General                data                variables
    *****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //Activity variables
    StudentsToClassAdapter studentsToClassAdapter;
    ArrayList<String> StudentsList = new ArrayList<>();

    //User variables
    String org;

                                //General                data                variables
    *****

    //Views
    RecyclerView students;
    Button add;

    @Override

```

```

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_add_student_to_class,
container, false);

//getting general vars
*****

        org = AddClassFragment.org;

//getting general vars
*****

//Find views
students = view.findViewById(R.id.FRRVStudentsToClass);
add = view.findViewById(R.id.FRBTNEndAddingStudents);

//Sets views
findStudentsInFirestore();

add.setOnClickListener(new View.OnClickListener() {
    //come back to AddClassFragment
    @Override
    public void onClick(View v) {
        AppCompatActivity activity = (AppCompatActivity)
view.getContext();
        AddClassFragment fragment = new AddClassFragment();
        activity.getSupportFragmentManager().beginTransaction()
            .replace(R.id.FContainerClassesT, fragment)
            .addToBackStack(null).commit();
    }
});

return view;
}

public void findStudentsInFirestore(){

db.collection("organizations").document(org).collection("Student").get()
    .addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {
        @Override
            public void onSuccess(QuerySnapshot
queryDocumentSnapshots) {

                for (QueryDocumentSnapshot documentSnapshot :
queryDocumentSnapshots) {
                    String documentName = documentSnapshot.getId();
                    StudentsList.add(documentName);

```

```

        }
        updateStudentsList();
    }
});
}

//Update the students views in the students RecyclerView
private void updateStudentsList() {
    RecyclerView.LayoutManager studentsToClassLayout = new
LinearLayoutManager(getContext());
    studentsToClassAdapter = new StudentsToClassAdapter();
    students.setLayoutManager(studentsToClassLayout);
    students.setAdapter(studentsToClassAdapter);
}

    private class StudentsToClassAdapter extends
RecyclerView.Adapter<StudentsToClassAdapter.StudentsToClassViewHolder>{
        @NonNull
        @Override
        public StudentsToClassViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
            View studentView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.recycleritem_studentsname,parent,false);
            return new StudentsToClassViewHolder(studentView);
        }

        @Override
        public void onBindViewHolder(@NonNull StudentsToClassViewHolder
holder, int position) {
            String currentStudent = StudentsList.get(position);

            holder.name.setText(currentStudent);
            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    AddClassFragment.AddStudent(currentStudent);
                    StudentsList.remove(currentStudent);
                    StudentsList.remove(currentStudent);
                    studentsToClassAdapter.notifyDataSetChanged();
                }
            });
        }

        @Override
        public int getItemCount() {
            return StudentsList.size();
        }
    }
}

```

```
        public static class StudentsToClassViewHolder extends
RecyclerView.ViewHolder{
    TextView name;
    public StudentsToClassViewHolder(@NonNull View itemView) {
        super(itemView);

        name = itemView.findViewById(R.id.RITVStudentNameForClass);
    }
}
}
```


AddTestActivity מחלקה

```

package com.example.educare;

import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.Timestamp;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import org.checkerframework.checker.nullness.qual.NonNull;

import java.util.ArrayList;

public class AddTestActivity extends AddMenuActivity {

                                //General                data                variables
*****
    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //SharedPreferences variables
    SharedPreferences UserData;

    //Activity variables

    ArrayList<String> students = new ArrayList<>();
    StudentTestListAdapter studentsTestListAdapter;

    //User variables
    String org;

```

```

String ClassroomID;
String teachers_name;

//General data variables
*****

//Views
TextView Subject;
RecyclerView studentList;
Button add;
EditText testName;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_test);

//getting general vars
*****

//getting data from SharedPreferences
UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
org = UserData.getString("org", "not found");

//getting data from Intent
Intent i = getIntent();
ClassroomID = i.getStringExtra("ClassroomId");

//getting general vars
*****

//Find views
Subject = findViewById(R.id.TVAddTestSubject);
studentList = findViewById(R.id.RVTestStudentsList);
testName = findViewById(R.id.ETTestName);
add = findViewById(R.id.BTNAddTest);

//Sets views
//find the Subject and the teachers_name in the firestore
db.collection("organizations").document(org).collection("Classes")
    .document(ClassroomID).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {

Subject.setText(documentSnapshot.get("Subject").toString());
teachers_name =
documentSnapshot.get("teacher").toString();
    }
}

```

```

    });

    createStudentsList();

    //adding the test grades to the firestore
    add.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (testName.getText().toString().equals("")){
                Toast.makeText(AddTestActivity.this, "not all the fields
are full", Toast.LENGTH_SHORT).show();
                return;
            }

            CollectionReference colRef =
db.collection("organizations").document(org).collection("Student");
            for (int i = 0; i < students.size();i++){
                DocumentReference docRef =
colRef.document(students.get(i)).collection("Tests").document(testName.getTex
t().toString());

                String Name = testName.getText().toString();
                String subject = Subject.getText().toString();
                int grade = studentsTestListAdapter.grades[i];
                long currentTimeMillis = System.currentTimeMillis();
                long currentTimeWithoutTime = (currentTimeMillis / (24 *
60 * 60 * 1000)) * (24 * 60 * 60 * 1000);
                Timestamp date = new Timestamp(currentTimeWithoutTime /
1000, 0);

                TestInput test = new TestInput(Name, grade, subject,
teachers_name, ClassroomID, date);
                docRef.set(test);
            }
            finish();
        }
    });

    //creating the views in the studentList RecyclerView
    private void createStudentsList() {
        //finding data from firebase

        DocumentReference docRef =
db.collection("organizations").document(org)
        .collection("Classes")
        .document(ClassroomID);

        docRef.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {

```

```

@Override
public void onSuccess(DocumentSnapshot documentSnapshot) {
    students = (ArrayList<String>)
documentSnapshot.get("Students");

    RecyclerView.LayoutManager studentsTestListLayout = new
LinearLayoutManager(AddTestActivity.this);
    studentsTestListAdapter = new StudentTestListAdapter(students,
org);
    studentList.setLayoutManager(studentsTestListLayout);
    studentList.setAdapter(studentsTestListAdapter);
}
});
}

//class Used in this activity to add the Test to firestore
private class TestInput extends Test{
    String ClassID;
    Timestamp date;

    public String getClassID() {
        return ClassID;
    }

    public void setClassID(String classID) {
        ClassID = classID;
    }

    public Timestamp getDate() {
        return date;
    }

    public void setDate(Timestamp date) {
        this.date = date;
    }

    public TestInput(String testName, int grade, String subject, String
teachers_name, String classID, Timestamp date) {
        super(testName, grade, subject, teachers_name);
        ClassID = classID;
        this.date = date;
    }
}

private class StudentTestListAdapter extends
RecyclerView.Adapter<StudentTestListAdapter.StudentTestListViewHolder>{

    ArrayList<String> students;

```

```

String org;

public static int[] grades;

public StudentTestListAdapter(ArrayList<String> students, String org){
    this.students = students;
    this.org = org;
    this.grades = new int[students.size()];
}

@Override
    public StudentTestListAdapter.StudentTestListViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View studentTestView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.recycleritem_studentstest,parent,false);
                                return new
StudentTestListAdapter.StudentTestListViewHolder(studentTestView);
}

@Override
    public void onBindViewHolder(@androidx.annotation.NonNull
StudentTestListAdapter.StudentTestListViewHolder holder, int position) {
        String currentStudent = students.get(position);

        int index = position;
        holder.name.setText(currentStudent);

        holder.grade.addTextChangedListener(new TextWatcher() {
            @Override
                public void beforeTextChanged(CharSequence s, int start, int
count, int after) {}
            @Override
                public void afterTextChanged(Editable s) {}
            @Override
                public void onTextChanged(CharSequence s, int start, int
before, int count) {
                    grades[index] = Integer.parseInt(s.toString());
                }
        });
    }

@Override
    public int getItemCount() {
        return students.size();
    }

```

```
        public static class StudentTestListViewHolder extends
RecyclerView.ViewHolder{
    TextView name;
    EditText grade;

    public StudentTestListViewHolder(@NonNull View itemView){
        super(itemView);

        name = itemView.findViewById(R.id.RITVStudentNameForTest);
        grade = itemView.findViewById(R.id.RIETGrade);
    }
}
}
```

מחלקה ClassDetailsActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.firebase.firestore.WriteBatch;

import java.util.ArrayList;

public class ClassDetailsActivity extends AddMenuActivity {

        //General          data          variables
        *****
        //Firestore variables
        FirebaseFirestore db = FirebaseFirestore.getInstance();

        //SharedPreferences variables
        SharedPreferences UserData;

        //User variables
        String org;
        String classID;

        //General          data          variables
        *****

```

```

//Views
RecyclerView lists;
TextView listKind;
TextView Subject;
Button student;
Button lessons;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_class_details);

    //getting general vars
    *****

    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
    org = UserData.getString("org", "not found");

    //getting data from Intent
    Intent intent = getIntent();
    classID = intent.getStringExtra("ClassroomId");

    //getting general vars
    *****

    //Find views
    listKind = findViewById(R.id.TVListKId);
    Subject = findViewById(R.id.TVClassSubject);
    lists = findViewById(R.id.RVStudentsInClass);
    student = findViewById(R.id.BTNStudentsList);
    lessons = findViewById(R.id.BTNLessonsList);

    //Sets views
    db.collection("organizations").document(org).collection("Classes")
        .document(classID).get().addOnSuccessListener(new
    OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            Subject.setText((String)
documentSnapshot.get("Subject"));
        }
    });

    RecyclerView.LayoutManager listLayout = new
LinearLayoutManager(ClassDetailsActivity.this);
    lists.setLayoutManager(listLayout);

```



```

        student.setOnClickListener(new View.OnClickListener() {
            //changing the lists RecyclerView to present StudentsList
            @Override
            public void onClick(View v) {
                listKind.setText("Students");
            }
        });

        db.collection("organizations").document(org).collection("Classes")
            .document(classID).get().addOnSuccessListener(new
        OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot
        documentSnapshot) {
                ArrayList<String> students =
        (ArrayList<String>) documentSnapshot.get("Students");
                if (students == null) students = new
        ArrayList<>();

                StudentsListInClassAdapter listAdapter = new
        StudentsListInClassAdapter(students);
                lists.setAdapter(listAdapter);
            }
        });

        lessons.setOnClickListener(new View.OnClickListener() {
            //changing the lists RecyclerView to present LessonsList
            @Override
            public void onClick(View v) {
                listKind.setText("Lessons");

                updateLessonsList();
            }
        });
    }

    public void updateLessonsList(){
        db.collection("organizations").document(org).collection("Classes")

        .document(classID).collection("Lessons").get().addOnCompleteListener(new
        OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task)
        {
                ArrayList<Lesson> Lessons= new ArrayList<>();
                if (task.isSuccessful()) {

```

```

        for (DocumentSnapshot document : task.getResult())
        {
            int day = Integer.parseInt(document.getId());
            ArrayList<String> lessons =
            (ArrayList<String>) document.get("lessonsID");
            if (lessons == null) lessons = new
            ArrayList<>();

            for (int i = 0; i < lessons.size(); i++) {
                ArrayList<String> finalLessons = lessons;
                int finalI = i;

                db.collection("organizations").document(org).collection("Lessons")

                .document(lessons.get(i)).get().addOnSuccessListener(new
                OnSuccessListener<DocumentSnapshot>() {

                    @Override

                    public void

                    onSuccess(DocumentSnapshot document) {

                        Time start = new

                        Time(Math.toIntExact((long)
                        document.get("startHour")),Math.toIntExact((long)
                        document.get("startMinute")));

                        Time end = new

                        Time(Math.toIntExact((long) document.get("endHour")),Math.toIntExact((long)
                        document.get("endMinute")));

                        String time =

                        start.toString() + "-" + end.toString();

                        Lessons.add(new

                        Lesson(day, time, finalLessons.get(finalI)));

                        LessonsListAdapter

                        listAdapter = new LessonsListAdapter(Lessons);

                        lists.setAdapter(listAdapter);

                    }

                });

            }

        }

    }

    private class StudentsListInClassAdapter extends
    RecyclerView.Adapter<StudentsListInClassAdapter.StudentsListInClassViewHolder
    >{

        ArrayList<String> students;

```

```

public StudentsListInClassAdapter(ArrayList<String> students) {
    this.students = students;
}

@NonNull
@Override
    public StudentsListInClassViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
    View StudentView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.recycleritem_studentinclass,parent,false);
    return new StudentsListInClassViewHolder(StudentView);
}

@Override
    public void onBindViewHolder(@NonNull StudentsListInClassViewHolder
holder, int position) {
    String currentStudent = students.get(position);

    holder.name.setText(currentStudent);
    holder.unShow.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i =new Intent(getApplicationContext(),
ShowDisturbanceActivity.class);
            i.putExtra("name", currentStudent);
            i.putExtra("disturbance","unShow");
            i.putExtra("ClassId", classID);
            startActivity(i);
        }
    });
    holder.noHomework.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i =new Intent(getApplicationContext(),
ShowDisturbanceActivity.class);
            i.putExtra("name", currentStudent);
            i.putExtra("disturbance","noHomeWork");
            i.putExtra("ClassId", classID);
            startActivity(i);
        }
    });

    holder.Tests.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

```

```

        Intent i =new Intent(ClassDetailsActivity.this ,
StudentTestsActivity.class);
        i.putExtra("ClassId", classID);
        i.putExtra("UserName", currentStudent);
        startActivity(i);
    }
});
}

@Override
public int getItemCount() {
    return students.size();
}

    public static class StudentsListInClassViewHolder extends
RecyclerView.ViewHolder{
        TextView name;
        Button unShow;
        Button noHomework;
        Button Tests;

        public StudentsListInClassViewHolder(@NonNull View itemView) {
            super(itemView);

            name = itemView.findViewById(R.id.RITVStudentInClassName);
            unShow = itemView.findViewById(R.id.RIBTNUnShow);
            noHomework = itemView.findViewById(R.id.RIBTNNoHomework);
            Tests = itemView.findViewById(R.id.RIBTNTests);
        }
    }
}

private class Lesson{
    static String[] daysOfWeek = {
        "Sunday",
        "Monday",
        "Tuesday",
        "Wednesday",
        "Thursday",
        "Friday",
        "Saturday"
    };
    int intDay;
    String day;
    String time;
    String lessonID;

    public Lesson(int day, String time, String lessonID) {
        this.day = daysOfWeek[day-1];

```

```

        this.intDay = day;
        this.time = time;
        this.lessonID = lessonID;
    }

    public String getDay() {
        return day;
    }

    public void setDay(String day) {
        this.day = day;
    }

    public String getTime() {
        return time;
    }

    public void setTime(String time) {
        this.time = time;
    }

    public String getLessonID() {
        return lessonID;
    }

    public void setLessonID(String lessonID) {
        this.lessonID = lessonID;
    }

    public int getIntDay() {
        return intDay;
    }

    public void setIntDay(int intDay) {
        this.intDay = intDay;
    }
}

        private class LessonsListAdapter extends
RecyclerView.Adapter<LessonsListAdapter.LessonsListViewHolder>{
    ArrayList<Lesson> lessons;

    public LessonsListAdapter(ArrayList<Lesson> lessons) {
        this.lessons = lessons;
    }

    @NonNull
    @Override

```

```

        public LessonsListAdapter.LessonsListViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View lessonView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.recycleritem_lessoninclass,parent,false);
    return new LessonsListAdapter.LessonsListViewHolder(lessonView);
}

@Override
        public void onBindViewHolder(@NonNull
LessonsListAdapter.LessonsListViewHolder holder, int position) {
    Lesson currentLesson = lessons.get(position);

    holder.time.setText(currentLesson.getTime());
    holder.day.setText(currentLesson.getDay());
    holder.itemView.setOnLongClickListener(new
View.OnLongClickListener() {
        @Override
        public boolean onLongClick(View v) {
            Dialog dialog = new Dialog(ClassDetailsActivity.this);
            dialog.setContentView(R.layout.dialog_deletelesson);

            // Find the buttons in the dialog layout
            Button yes = dialog.findViewById(R.id.DialogBTNYes);
            Button no = dialog.findViewById(R.id.DialogBTNNo);

            // Set click listeners for the buttons inside the dialog
            yes.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

db.collection("organizations").document(org).collection("Lessons").document(c
urrentLesson.getLessonID())

                    .delete();
                    DocumentReference docRef =
db.collection("organizations").document(org).collection("Classes").document(c
lassID)

.collection("Lessons").document(currentLesson.getIntDay()+"");
                    docRef.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                        @Override
                        public void onSuccess(DocumentSnapshot
documentSnapshot) {

                            ArrayList<String> lessonsIDs =
                            (ArrayList<String>) documentSnapshot.get("lessonsID");
                            if(lessonsIDs == null) lessonsIDs
= new ArrayList<>();

```

```

        if (lessonsIDs.size() <= 1)
            docRef.delete();
        else {

lessonsIDs.remove(currentLesson.getLessonID());
docRef.update("lessonsID",
lessonsIDs);

        }

        updateLessonsList();
    }
    });
    dialog.dismiss();
}
});

no.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        dialog.dismiss();
    }
});

dialog.show();
return false;
}
});
}

@Override
public int getItemCount() {
    return lessons.size();
}

    public static class LessonsListViewHolder extends
RecyclerView.ViewHolder{
        TextView day;
        TextView time;
        public LessonsListViewHolder(@NonNull View itemView) {
            super(itemView);

            day = itemView.findViewById(R.id.RITVDay);
            time = itemView.findViewById(R.id.RITVLessonTime);
        }
    }
}

```

}

מחלקה ClassesTActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;
import java.util.List;

public class ClassesTActivity extends AddMenuActivity {

    //General data variables
    *****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //SharedPreferences variables
    SharedPreferences UserData;

    //User variables
    String org;
    String UserName;

    //General data variables
    *****

    //Views
    RecyclerView classesList;

```

```

Button addClass;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_classes_tactivity);

                                //getting      general      vars
    *****

    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
    org = UserData.getString("org", "not found");
    UserName = UserData.getString("UserName", "not found");
                                //getting      general      vars
    *****

    //Find views
    addClass = findViewById(R.id.BTNAddClass);
    classesList = findViewById(R.id.RVClassesList);

    //Sets views
    addClass.setOnClickListener(new View.OnClickListener() {
        //Open the AddClassFragment
        @Override
        public void onClick(View v) {
            Bundle bundle = new Bundle();
            bundle.putString("org", org);
            bundle.putString("user name", UserName);
            AddClassFragment myFragment = new AddClassFragment();
            myFragment.setArguments(bundle);
            FragmentManager fragmentManager = getSupportFragmentManager();
                                FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
                                fragmentTransaction.replace(R.id.FContainerClassesT,
myFragment);
                                fragmentTransaction.commit();
        }
    });

    updateClassesList();
}

private void updateClassesList() {

    db.collection("organizations").document(org).collection("Teacher")
        .document(UserName).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {

```

```

        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            Object existingClassesArray =
documentSnapshot.get("Classes");
            if (existingClassesArray instanceof List) {
                ArrayList<String> classes = (ArrayList<String>)
documentSnapshot.get("Classes");

                RecyclerView.LayoutManager classesListLayout = new
LinearLayoutManager(ClassesTActivity.this);
                ClassAdapter classesListAdapter = new
ClassAdapter(classes, org, ClassesTActivity.this);
                classesList.setLayoutManager(classesListLayout);
                classesList.setAdapter(classesListAdapter);
            }
        }
    });
}

private class ClassAdapter extends
RecyclerView.Adapter<ClassAdapter.ClassViewHolder> {
    ArrayList<String> classes;
    String org;

    private Context context;

    public ClassAdapter(ArrayList<String> classes, String org, Context
context) {
        this.classes = classes;
        this.org = org;
        this.context = context;
    }

    @NonNull
    @Override
    public ClassAdapter.ClassViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
        View classView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.recycleritem_class_teacher, parent, false);
        return new ClassAdapter.ClassViewHolder(classView);
    }

    @Override
    public void onBindViewHolder(@NonNull ClassAdapter.ClassViewHolder
holder, int position) {
        String currentClass = classes.get(position);

```

```

//General views
db.collection("organizations").document(org).collection("Classes")
    .document(currentClass).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
    @Override
        public void onSuccess(DocumentSnapshot
documentSnapshot) {

holder.Subject.setText((String)documentSnapshot.get("Subject"));
        }
    });

holder.Homework.setOnClickListener(new View.OnClickListener() {
    @Override
        public void onClick(View view) {

db.collection("organizations").document(org).collection("Classes")

.document(currentClass).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
    @Override
        public void onSuccess(DocumentSnapshot
documentSnapshot) {

        Bundle bundle = new Bundle();
        bundle.putString("Subject",
(String)documentSnapshot.get("Subject"));
        bundle.putString("ClassID", currentClass);
        bundle.putString("org", org);

        AppCompatActivity activity =
(AppCompatActivity) view.getContext();
        AddHomeworkFragment fragment = new
AddHomeworkFragment();
        fragment.setArguments(bundle);

activity.getSupportFragmentManager().beginTransaction()

.replace(R.id.FContainerClassesT, fragment)
        .addToBackStack(null).commit();
        }
    });
}

holder.Test.setOnClickListener(new View.OnClickListener() {
    @Override
        public void onClick(View view) {

```

```

        Intent intent = new Intent(ClassesTActivity.this,
AddTestActivity.class);
        intent.putExtra("ClassroomId", currentClass);
        context.startActivity(intent);
    }
});

holder.itemView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(context,
ClassDetailsActivity.class);
        intent.putExtra("ClassroomId", currentClass);
        context.startActivity(intent);
    }
});
}

@Override
public int getItemCount() {
    return classes.size();
}

public static class ClassViewHolder extends RecyclerView.ViewHolder{

    Button Homework;
    Button Test;
    TextView Subject;

    public ClassViewHolder(@NonNull View itemView) {
        super(itemView);

        Homework = itemView.findViewById(R.id.RIBTNAddHomework);
        Test = itemView.findViewById(R.id.RIBTNAddTest);
        Subject = itemView.findViewById(R.id.RITVSubject);
    }
}
}
}

```

מחלקה HomePageActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;

import android.Manifest;
import android.content.pm.PackageManager;
import android.util.Log;
import android.widget.Toast;
import androidx.core.content.ContextCompat;

import android.os.Build;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;
import java.util.Calendar;

public class HomePageActivity extends AddMenuActivity {

                                //General                                data                                variables
    *****

    //Firestore variables

```

```

FirebaseFirestore db = FirebaseFirestore.getInstance();

//SharedPreferences variables
SharedPreferences UserData;

//Activity variables
int day;
Calendar calendar = Calendar.getInstance();
AlarmManager alarmManager;

ArrayList<String> Classes = new ArrayList<>();
ArrayList<Lesson> lessons = new ArrayList<>();

//User variables
String org;
String tORs;
String UserName;

//General data variables
*****

//Views
TextView Name;
RecyclerView timetable;
Button SetNotification;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home_page);

//getting general vars
*****

    UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
    org = UserData.getString("org", "not found");
    tORs = UserData.getString("tORs", "not found");
    UserName = UserData.getString("UserName", "not found");
    day = calendar.get(Calendar.DAY_OF_WEEK);

    alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);

//getting general vars
*****

//Find views
Name = findViewById(R.id.TVName);

```

```

timetable = findViewById(R.id.RVTimetable);
SetNotification = findViewById(R.id.BTNSetNotification);

//Sets views

Name.setText(Username);

SetNotification.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        setAlarm();
    }
});

// Building lessons from Firestore

db.collection("organizations").document(org).collection(tORs).document(UserName)
                                                                    .get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {
        if (documentSnapshot.exists()) {
                                                                    Classes = (ArrayList<String>)
documentSnapshot.get("Classes");
            updateTimeTable();
        }
    }
});
}

private void setAlarm() {
    for (Lesson lesson : lessons){
        Calendar calendar = lesson.getStart().subtractMinutes(5);

        long alarmTimeInMillis = calendar.getTimeInMillis();
        int notificationId = (int) System.currentTimeMillis();

        AlarmManager alarmMgr = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent(HomePageActivity.this,
NotificationReceiver.class);
        intent.putExtra("title", lesson.subject);
        intent.putExtra("notes", "you have a lesson in 5 minutes");
        intent.putExtra("id", notificationId); // Use notificationId as
the ID

        PendingIntent pendingIntent = PendingIntent.getBroadcast(

```



```

        HomePageActivity.this,
        notificationId,
        intent,
        PendingIntent.FLAG_UPDATE_CURRENT // Use
FLAG_UPDATE_CURRENT instead of FLAG_IMMUTABLE
    );
    alarmMgr.set(AlarmManager.RTC_WAKEUP, alarmTimeInMillis,
pendingIntent);
    }
}

public void updateTimeTable() {
    if(Classes == null) {return;}
    CollectionReference colRef = db.collection("organizations")
        .document(org).collection("Classes");

    for (int i = 0; i <= Classes.size(); i++) {
        if (i < Classes.size()) {
            DocumentReference docRef = colRef.document(Classes.get(i));
            int finalI = i;

            docRef.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                @Override

                public void onSuccess(DocumentSnapshot
documentSnapshotOrig) {
                    ArrayList<Boolean> lessonDays = (ArrayList<Boolean>)
documentSnapshotOrig.get("Days");
                    if (lessonDays == null) return;
                    if (lessonDays.get(day-1)) {
                        docRef.collection("Lessons").document(day + "")
                            .get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                                @Override

                                public void onSuccess(DocumentSnapshot
documentSnapshot) {

                                    ArrayList<String> lessonsID =
(ArrayList<String>) documentSnapshot.get("lessonsID");
                                    if (lessonsID == null) return;
                                    for (int j = 0; j <
lessonsID.size(); j++) {

                                        db.collection("organizations").document(org)

                                        .collection("Lessons").document(lessonsID.get(j))

                                        .get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
                                            @Override

```

```

public void
onSuccess(DocumentSnapshot doc) {
    Integer
    startHour = doc.getLong("startHour").intValue();
    Integer
    startMinute = doc.getLong("startMinute").intValue();
    Integer
    endHour = doc.getLong("endHour").intValue();
    Integer
    endMinute = doc.getLong("endMinute").intValue();
    String subject
    = documentSnapshotOrig.getString("Subject");

    lessons.add(new Lesson(subject, new Time(startHour, startMinute), new
    Time(endHour, endMinute), Classes.get(finalI)));

    RecyclerView.LayoutManager timetableLayout = new
    LinearLayoutManager(HomePageActivity.this);
    LessonAdapter
    timetableAdapter = new LessonAdapter(lessons, tORs);

    timetable.setLayoutManager(timetableLayout);

    timetable.setAdapter(timetableAdapter);
}
}
}
}
}
}
}
}
}
}
}

private class Lesson {
    String subject;
    Time Start;
    Time End;
    String ClassroomID;

    public Lesson(String className, Time start, Time end, String
classroomID) {
        this.subject = className;
        Start = start;

```

```

        End = end;
        ClassroomID = classroomID;
    }

    public String getClassroomID() {
        return ClassroomID;
    }

    public void setClassroomID(String classroomID) {
        ClassroomID = classroomID;
    }

    public String getSubject() {
        return subject;
    }

    public void setSubject(String subject) {
        this.subject = subject;
    }

    public Time getStart() {
        return Start;
    }

    public void setStart(Time start) {
        Start = start;
    }

    public Time getEnd() {
        return End;
    }

    public void setEnd(Time end) {
        End = end;
    }
}

public class LessonAdapter extends
RecyclerView.Adapter<LessonAdapter.LessonViewHolder>{
    ArrayList<Lesson> lessons;
    String tORs;

    public LessonAdapter(ArrayList<Lesson> lessons, String tORs) {
        this.lessons = lessons;
        this.tORs = tORs;
    }

    @NonNull

```

```

@Override
public LessonViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
    View lessonView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.recyclerviewitem_lesson, parent, false);
    return new LessonViewHolder(lessonView);
}

@Override
public int getItemCount() {
    return lessons.size();
}

@Override
public void onBindViewHolder(@NonNull LessonViewHolder holder, int
position) {
    Lesson currentLesson = lessons.get(position);

    holder.className.setText(currentLesson.getSubject());
    holder.Time.setText(currentLesson.getStart().toString() + "-" +
currentLesson.getEnd().toString());

    // Set OnClickListener on the view
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Start new activity with the ClassroomID as extra
            if (tORs.equals("Teacher")){
                Intent i = new Intent(view.getContext(),
LessonTActivity.class);
                i.putExtra("ClassroomId",
lessons.get(holder.getAdapterPosition()).getClassroomID());
                i.putExtra("Subject",
lessons.get(holder.getAdapterPosition()).getSubject());
                i.putExtra("StartHour",
lessons.get(holder.getAdapterPosition()).getStart().Hour);
                i.putExtra("StartMinutes",
lessons.get(holder.getAdapterPosition()).getStart().Minutes);
                i.putExtra("EndHour",
lessons.get(holder.getAdapterPosition()).getEnd().Hour);
                i.putExtra("EndMinutes",
lessons.get(holder.getAdapterPosition()).getEnd().Minutes);
                view.getContext().startActivity(i);
            }
            else {
                Bundle bundle = new Bundle();
                bundle.putString("ClassroomId",
lessons.get(holder.getAdapterPosition()).getClassroomID());

```

```

        AppCompatActivity activity = (AppCompatActivity)
view.getContext();
        LessonSFragment fragment = new LessonSFragment();
        fragment.setArguments(bundle);

activity.getSupportFragmentManager().beginTransaction()
        .replace(R.id.FContainerLessonS, fragment)
        .addToBackStack(null).commit();
    }

    }
});
}

public static class LessonViewHolder extends RecyclerView.ViewHolder{

    public TextView className;
    public TextView Time;

    public LessonViewHolder(@NonNull View itemView) {
        super(itemView);

        className = itemView.findViewById(R.id.RITVClassName);
        Time = itemView.findViewById(R.id.RITVTime);
    }
}
}
}

```

מחלקה HomePageActivity

```

package com.example.educare;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;

import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

public class LessonSFragment extends Fragment {

                                //General                data                variables
    *****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //Firestore variables
    SharedPreferences UserData;

    //User variables
    String org;
    String ClassID;

                                //General                data                variables
    *****

    //Views
    TextView TVSubject;
    Button tests;
    Button Homeworks;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

```

```

View view = inflater.inflate(R.layout.fragment_lesson_s, container,
false);

//getting general vars
*****

//getting data from bundle
Bundle args = getArguments();
ClassID = args.getString("ClassroomId");

//getting data from SharedPreferences
UserData = getActivity().getSharedPreferences("UserData",
Context.MODE_PRIVATE);
org = UserData.getString("org", "not found");

//getting general vars
*****

//Find views
TVSubject = view.findViewById(R.id.TVFragmentSubject);
tests = view.findViewById(R.id.BTNFragmentTests);
Homeworks = view.findViewById(R.id.BTNFragmentHomework);

//Sets views
db.collection("organizations").document(org).collection("Classes")
    .document(ClassID).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {
        TVSubject.setText((String)
documentSnapshot.get("Subject"));
    }
});

tests.setOnClickListener(new View.OnClickListener() {
//Move to StudentTestsActivity
@Override
public void onClick(View view) {
    Intent i =new Intent(getActivity().getApplicationContext(),
StudentTestsActivity.class);
    i.putExtra("ClassId",ClassID);
    i.putExtra("UserName", UserData.getString("UserName", "not
found"));

    startActivity(i);
}
});

Homeworks.setOnClickListener(new View.OnClickListener() {
//Move to StudentHomeworksActivity
@Override

```

```
        public void onClick(View v) {  
            Intent i =new Intent(getActivity().getApplicationContext(),  
StudentHomeworksActivity.class);  
            i.putExtra("ClassId",ClassID);  
            startActivity(i);  
        }  
    });  
  
    return view;  
}
```


LessonTActivity מחלקה

```
package com.example.educare;

import static java.lang.Thread.sleep;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import android.Manifest;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.Timestamp;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class LessonTActivity extends AddMenuActivity {
```

```

//General data variables
*****

//Firestore variables
FirebaseFirestore db = FirebaseFirestore.getInstance();

//RealtimeDatabase variables
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef;

//RealtimeDatabase variables
SharedPreferences UserData;

//Activity variables
Timestamp lessonTime;
ArrayList<Student> students = new ArrayList<>();
StudentsListAdapter studentsListAdapter;

//User variables
String org;
String SubjectName;
String ClassroomID;

//General data variables
*****

//Views
RecyclerView studentsList;
TextView Subject;
Button update;
Button findStudents;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_lesson_tactivity);
    //getting general vars
*****
    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
    org = UserData.getString("org", "not found");
    //getting data from Intent
    Intent i = getIntent();
    ClassroomID = i.getStringExtra("ClassroomId");
    SubjectName = i.getStringExtra("Subject");

    myRef = database.getReference(org).child(ClassroomId);

```

```

//getting      general      vars
*****

//Find views
Subject = findViewById(R.id.TVSubject);
update = findViewById(R.id.BTNUpdateStudentsData);
studentsList = findViewById(R.id.RVStudentsList);
findStudents = findViewById(R.id.BTNFindData);

//Sets views
db.collection("organizations").document(org).collection("Classes")
    .document(ClassroomID).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {
        Subject.setText((String)
documentSnapshot.get("Subject"));
    }
});

update.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY, 15);
        calendar.set(Calendar.MINUTE, 30);
        calendar.set(Calendar.SECOND, 0);
        calendar.set(Calendar.MILLISECOND, 0);
        lessonTime = new Timestamp(calendar.getTime());
        for (int i = 0; i < students.size(); i++) {
            //Adding Attendance to Student
            if (!students.get(i).Attendance) {
                CollectionReference colRef =
db.collection("organizations").document(org)
                .collection("Student").document(students.get(i).getName())
                .collection("disturbance");
                int finalI = i;

                DocumentReference docRefUnShow =
colRef.document("unShow");
                docRefUnShow.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                    @Override
                    public void onSuccess(DocumentSnapshot
documentSnapshot) {
                        if (documentSnapshot.exists()) {

```

```

Object existingDatesArray =
documentSnapshot.get("date");
existingDatesArray;
documentSnapshot.get("Class");
HashMap<>();

if (existingDatesArray instanceof List) {
    List<Object> dates = (List<Object>)
        existingDatesArray;
    List<Object> Classes = (List<Object>)
        documentSnapshot.get("Class");
    dates.add(lessonTime);
    Classes.add(ClassroomID);
    Map<String, Object> data = new
        HashMap<>();
    data.put("date", dates);
    data.put("Class", Classes);
    docRefUnShow.set(data);
} else {
    // The document does not exist
    ArrayList<Object> dates = new
        ArrayList<>();
    ArrayList<Object> Classes = new
        ArrayList<>();
    dates.add(lessonTime);
    Classes.add(ClassroomID);
    Map<String, Object> data = new
        HashMap<>();
    data.put("date", dates);
    data.put("Class", Classes);

    db.collection("organizations").document(org)
        .collection("Student").document(students.get(finalI).getName())
        .collection("disturbance").document("unShow")
            .set(data);
}
});
//Adding Homework to Student
if (!students.get(i).DidHomeWork) {
    CollectionReference colRef =
db.collection("organizations").document(org)
    .collection("Student").document(students.get(i).getName())
        .collection("disturbance");
    int finalI = i;

```

```

        DatabaseReference docRefNoHomeWork =
colRef.document("noHomeWork");
        docRefNoHomeWork.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override
                public void onSuccess(DocumentSnapshot
documentSnapshot) {
                    if (documentSnapshot.exists()) {
                        Object existingDatesArray =
documentSnapshot.get("date");
                        if (existingDatesArray instanceof List) {
                            List<Object> dates = (List<Object>)
existingDatesArray;
                            List<Object> Classes = (List<Object>)
documentSnapshot.get("Class");
                            dates.add(lessonTime);
                            Classes.add(ClassroomID);
                            Map<String, Object> data = new
HashMap<>();
                            data.put("date", dates);
                            data.put("Class", Classes);
                            docRefNoHomeWork.set(data);
                        }
                    } else {
                        // The document does not exist
                        ArrayList<Object> dates = new
ArrayList<>();
                        ArrayList<Object> Classes = new
ArrayList<>();
                        dates.add(lessonTime);
                        Classes.add(ClassroomID);
                        Map<String, Object> data = new
HashMap<>();
                        data.put("date", dates);
                        data.put("Class", Classes);

db.collection("organizations").document(org)

.collection("Student").document(students.get(finalI).getName())

.collection("disturbance").document("noHomeWork")
                        .set(data);
                    }
                }
            });
        }
    }
}

```

```

        finish();
    }
});

updateStudents();

findStudents.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        getLocation();
    }
});
}

/* This method find the teachers location and send it to all of the
students by
that the user find the distance between him and the teacher and returns if
the
student is in reasonable distance from the teacher and if it is the
teacher set
the student as Attendance in lesson */
FusedLocationProviderClient fusedLocationClient;
public void getLocation() {
    fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION}, 111);
        return;
    }
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION}, 112);
        return;
    }
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

```

```

        ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.ACCESS_FINE_LOCATION}, 113);
        return;
    }

    fusedLocationClient.getLastLocation()
        .addOnSuccessListener(this, new OnSuccessListener<Location>()
{
    @Override
    public void onSuccess(Location location) {
        if (location != null) {
            Toast.makeText(LessonTActivity.this, "Latitude:"
+ location.getLatitude(), Toast.LENGTH_SHORT).show();
            Toast.makeText(LessonTActivity.this, "Longitude:
" + location.getLongitude(), Toast.LENGTH_SHORT).show();

            myRef.addValueEventListener(new
ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot
snapshot) {
                    for (DataSnapshot childSnapshot :
snapshot.getChildren()) {
                        if
(childSnapshot.getKey().equals("Students")) {
                            for (DataSnapshot studentSnapshot
: childSnapshot.getChildren()) {
                                String student =
studentSnapshot.getValue(String.class);
                                addAttendanceToStudent(student);
                            }
                        }
                    }
                }
            });
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError
error) {
    }
});

    UserLocation userLocation = new
UserLocation(location.getLatitude(), location.getLongitude());
    myRef.child("Teacher").setValue(userLocation);

    new Thread(new Runnable() {
        @Override

```

```

        public void run() {
            try {
                sleep(5000);
                myRef.child("Teacher").setValue(new
UserLocation(0, 0));

            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
        }
    }).start();
} else {
    Toast.makeText(LessonTActivity.this, "having
problem find your location", Toast.LENGTH_SHORT).show();
}
}
});
}

public void updateStudents() {
    //finding data from firebase
    DatabaseReference docRef =
db.collection("organizations").document(org)
    .collection("Classes").document(ClassroomID);
    docRef.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            ArrayList<String> studentsNames = (ArrayList<String>)
documentSnapshot.get("Students");
            for (int i = 0; i < studentsNames.size(); i++)
                students.add(new Student(studentsNames.get(i)));

            RecyclerView.LayoutManager studentsListLayout = new
LinearLayoutManager(LessonTActivity.this);
            studentsListAdapter = new StudentsListAdapter(students, org);
            studentsList.setLayoutManager(studentsListLayout);
            studentsList.setAdapter(studentsListAdapter);
        }
    });
}

public void addAttendanceToStudent(String student) {
    for (int i = 0; i < StudentsListAdapter.students.size(); i++) {
        if (StudentsListAdapter.students.get(i).Name.equals(student)) {
            StudentsListAdapter.students.get(i).Attendance = true;
            RecyclerView.LayoutManager layoutManager =
studentsList.getLayoutManager();

```



```

StudentsListAdapter.holders.get(i).Attendance.setText("UnShow");
    }
}

public class Student {
    String Name;
    Boolean Attendance;
    Boolean DidHomeWork;

    public Student(String name) {
        Name = name;
        Attendance = false;
        DidHomeWork = false;
    }

    public void changeAttendance() {Attendance = !Attendance;}

    public void changeHomeWork() {DidHomeWork = !DidHomeWork;}

    public String getName() {
        return Name;
    }

    public void setName(String name) {
        Name = name;
    }
}

private class StudentsListAdapter extends
RecyclerView.Adapter<StudentsListAdapter.StudentsListViewHolder> {
    static ArrayList<Student> students;
    String org;
    static ArrayList<StudentsListAdapter.StudentsListViewHolder> holders
= new ArrayList<StudentsListAdapter.StudentsListViewHolder>();

    public StudentsListAdapter(ArrayList<Student> students, String org) {
        holders = new ArrayList<>();
        this.students = students;
        this.org = org;
    }

    @NonNull
    @Override
    public StudentsListAdapter.StudentsListViewHolder
onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

        View studentView = LayoutInflater.from(parent.getContext())

```

```

        .inflate(R.layout.recycleritem_student, parent, false);
        return new
StudentsListAdapter.StudentsListViewHolder(studentView);
    }

    @Override
        public void onBindViewHolder(@NonNull
StudentsListAdapter.StudentsListViewHolder holder, int position) {
        holders.add(holder);
        Student currentStudent = students.get(position);
        FirebaseFirestore db = FirebaseFirestore.getInstance();

        DocumentReference docRef = db.collection("organizations")
            .document(org).collection("Student")
            .document(currentStudent.getName());

        holder.Name.setText(currentStudent.getName());

        holder.Attendance.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

students.get(holder.getAdapterPosition()).changeAttendance();

                                                                    if
(holder.Attendance.getText().toString().equals("Attendance"))
                holder.Attendance.setText("UnShow");
                else holder.Attendance.setText("Attendance");
            }
        });

        holder.HomeWork.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

students.get(holder.getAdapterPosition()).changeHomeWork();

                                                                    if
(holder.HomeWork.getText().toString().equals("HomeWork"))
                holder.HomeWork.setText("no HomeWork");
                else holder.HomeWork.setText("HomeWork");
            }
        });

    }

    @Override
    public int getItemCount() {
        return students.size();
    }
}

```

```
public static class StudentsListViewHolder extends
RecyclerView.ViewHolder {
    public TextView Name;
    public Button HomeWork;
    public Button Attendance;

    public StudentsListViewHolder(@NonNull View itemView) {
        super(itemView);

        Name = itemView.findViewById(R.id.RITVStudentName);

        Attendance = itemView.findViewById(R.id.BTNAttendance);
        HomeWork = itemView.findViewById(R.id.BTNHomeWork);
    }
}
```

MainActivity מחלקה

```
package com.example.educare;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onResume() {
        super.onResume();
        finish();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Intent i =new Intent(getApplicationContext(), SignInActivity.class);
        startActivity(i);
    }
}
```

מתלקה NotificationReceiver

```

package com.example.educare;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.util.Log;

import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

public class NotificationReceiver extends BroadcastReceiver {

    private static final String CHANNEL_ID = "my_channel";

    @Override
    public void onReceive(Context context, Intent intent) {
        String title = intent.getStringExtra("title");
        String notes = intent.getStringExtra("notes");
        int id = intent.getIntExtra("id", 0);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, CHANNEL_ID)
            .setSmallIcon(R.drawable.calender_iv)
            .setContentTitle(title)
            .setContentText(notes)
            .setPriority(NotificationCompat.PRIORITY_DEFAULT);
        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(context);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "My Channel";
            String description = "My Channel Description";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
            channel.setDescription(description);
            notificationManager.createNotificationChannel(channel);
        }
        notificationManager.notify(id, builder.build());
    }
}

```

ShowDisturbanceActivity מחלקה

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.Timestamp;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class ShowDisturbanceActivity extends AddMenuActivity {

                                //General                data                variables
    *****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //SharedPreferences variables
    SharedPreferences UserData;

    //User variables
    String org;
    String disturbance;
    String name;
    String classID;

                                //General                data                variables
    *****

    //Views
    TextView TVName;
    TextView TVDisturbance;

```

```

RecyclerView disturbanceList;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_show_disturbance);

    //getting general vars
    *****

    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
    org = UserData.getString("org", "not found");

    //getting data from Intent
    Intent intent = getIntent();
    name = intent.getStringExtra("name");
    disturbance = intent.getStringExtra("disturbance");
    classID = intent.getStringExtra("ClassId");

    //getting general vars
    *****

    //Find views
    TVName = findViewById(R.id.TVStudentDisturbanceName);
    TVDisturbance = findViewById(R.id.TVDisturbance);
    disturbanceList = findViewById(R.id.RVDisturbanceList);

    //Sets views
    TVName.setText(name);
    TVDisturbance.setText(disturbance);

    updateDisturbanceList();
}

private void updateDisturbanceList() {
    db.collection("organizations").document(org).collection("Student")

    .document(name).collection("disturbance").document(disturbance)
    .get().addOnSuccessListener(new
    OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            ArrayList<String> subjects = (ArrayList<String>)
documentSnapshot.get("Class");
            if (subjects == null) subjects = new ArrayList<>();
            ArrayList<Timestamp> dates = (ArrayList<Timestamp>)
documentSnapshot.get("date");
            if (dates == null) dates = new ArrayList<>();

```

```

        RecyclerView.LayoutManager disturbanceListLayout = new
LinearLayoutManager (ShowDisturbanceActivity.this);
        DisturbanceListAdapter disturbanceListAdapter = new
DisturbanceListAdapter (subjects,dates,classID);

disturbanceList.setLayoutManager(disturbanceListLayout);
        disturbanceList.setAdapter (disturbanceListAdapter);
    }
});

}

private class DisturbanceListAdapter extends
RecyclerView.Adapter<DisturbanceListAdapter.DisturbanceViewHolder>{
    ArrayList<String> subjects = new ArrayList<>();
    ArrayList<String> dates = new ArrayList<>();

    public DisturbanceListAdapter(ArrayList<String> subjectsIDs,
ArrayList<Timestamp> Dates, String requiredClass) {
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        for (int i = 0; i < Dates.size(); i++) {
            if((requiredClass == null) ||
(requiredClass.equals(subjectsIDs.get(i)))) {
                subjects.add(subjectsIDs.get(i));
                Date date = Dates.get(i).toDate();
                dates.add(sdf.format(date));
            }
        }
    }

    @NonNull
    @Override
    public DisturbanceViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View disturbanceView = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.recycleritem_disturbance,parent,false);
        return new DisturbanceViewHolder(disturbanceView);
    }

    @Override
    public void onBindViewHolder(@NonNull DisturbanceViewHolder
holder, int position) {
        db.collection("organizations").document(org).collection("Classes")

        .document(subjects.get(position)).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override

```



```
public void onSuccess(DocumentSnapshot
documentSnapshot) {
    holder.subject.setText((String)
documentSnapshot.get("Subject"));
    }
    });
    holder.date.setText(dates.get(position));
}

@Override
public int getItemCount() {
    return subjects.size();
}

    public static class DisturbanceListViewHolder extends
RecyclerView.ViewHolder{
        TextView subject;
        TextView date;

        public DisturbanceListViewHolder(@NonNull View itemView) {
            super(itemView);

            subject = itemView.findViewById(R.id.RITVDisturbanceSubject);
            date = itemView.findViewById(R.id.RITVDisturbanceDate);
        }
    }
}
```

מחלקה SignInActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

public class SignInActivity extends AppCompatActivity {

    //General data variables
    *****
    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //Authentication variables
    FirebaseAuth mAuth = FirebaseAuth.getInstance();

```

```

//SharedPreferences variables
SharedPreferences savedUserData;
SharedPreferences.Editor UserData;

//User variables
String UID;
String org;
String tORs;
String Name;

//General data variables
*****

//Views
TextView Email;
TextView Password;
Button enter;
Button signUp;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sign_in);

//getting general vars
*****

//getting data from SharedPreferences
savedUserData = getSharedPreferences("UserData", MODE_PRIVATE);
UserData = savedUserData.edit();

//getting general vars
*****

//find if the user is already register from this device
if (
    savedUserData.getString("org", "nope") != "nope" ||
    savedUserData.getString("tORs", "nope") != "nope" ||
    savedUserData.getString("UserName", "nope") != "nope"
) {
    org =savedUserData.getString("org", "nope");
    tORs =savedUserData.getString("tORs", "nope");
    Name =savedUserData.getString("UserName", "nope");
    if (tORs.equals("Student")) {
        /* adding listeners for the student classes in
RealtimeDatabase the
        listeners get the teacher location and find the distance
between the
        teacher to the student and return the teacher if it a
reasonable

```

```

        distance to be in the lesson */
        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED &&
            ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
            String[]{android.Manifest.permission.ACCESS_COARSE_LOCATION,
            android.Manifest.permission.ACCESS_FINE_LOCATION}, 111);
        } else if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
            String[]{android.Manifest.permission.ACCESS_COARSE_LOCATION}, 112);
        } else if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 113);
        }

        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)
            == PackageManager.PERMISSION_GRANTED &&
            ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {
            db.collection("organizations").document(org)
                .collection("Student").document(Name)
                    .get().addOnSuccessListener(new
            OnSuccessListener<DocumentSnapshot>() {
                @Override
                public void onSuccess(DocumentSnapshot
            documentSnapshot) {
                    ArrayList<String> classesIDs =
            (ArrayList<String>) documentSnapshot.get("Classes");
                    if (classesIDs == null) classesIDs = new
            ArrayList<>();

                    FirebaseDatabase database =
            FirebaseDatabase.getInstance();

                    for (String id : classesIDs) {

            database.getReference(org).child(id).child("Teacher")
                .addValueEventListener(new
            ValueEventListener() {
                @Override

```

```

public void
onDataChange(@NonNull DataSnapshot snapshot) {
    Iterable<DataSnapshot>
    children = snapshot.getChildren();
    for (DataSnapshot
    child : children) {
        try {
            Long
            latitudeLong = child.child("Latitude").getValue(Long.class);
            Long
            longitudeLong = child.child("Longitude").getValue(Long.class);
            double
            latitude = latitudeLong != null ? latitudeLong.doubleValue() : 0.0;
            double
            longitude = longitudeLong != null ? longitudeLong.doubleValue() : 0.0;
            UserLocation
            TeacherLocation = new UserLocation(latitude, longitude);

            final
            UserLocation[] studentLocation = new UserLocation[1];

            FusedLocationProviderClient fusedLocationClient =
            LocationServices.getFusedLocationProviderClient(SignInActivity.this);
            if
            (ActivityCompat.checkSelfPermission(SignInActivity.this,
            Manifest.permission.ACCESS_FINE_LOCATION)
            !=
            PackageManager.PERMISSION_GRANTED
            &&
            ActivityCompat.checkSelfPermission(SignInActivity.this,
            Manifest.permission.ACCESS_COARSE_LOCATION)
            !=
            PackageManager.PERMISSION_GRANTED) {return;}

            fusedLocationClient.getLastLocation()

            .addOnSuccessListener(SignInActivity.this, new OnSuccessListener<Location>()
            {

            @Override

            public void onSuccess(Location location) {

            if (location != null) {

            studentLocation[0] = new UserLocation(location.getLatitude(),
            location.getLongitude());

```

```

if (TeacherLocation.isWithinRange(studentLocation[0]) || true){

database.getReference(org).child(id).child("Students").child(Name).setValue(N
ame);

}

}

}

});
} catch (Exception
e) {
Log.d("error",
e.toString());
}
}

@Override
public void
onCancelled(@NonNull DatabaseError error) {}

});
}
});
}

}

Intent i =new Intent(getApplicationContext(),
HomePageActivity.class);
startActivity(i);
}

//Find views
Email = findViewById(R.id.ETSignInEmail);
Password = findViewById(R.id.ETPassword);
signUp = findViewById(R.id.BTNSignUp);
enter = findViewById(R.id.BTNEnter);

//Sets views

signUp.setOnClickListener(view -> {
Intent i = new
Intent(getApplicationContext(),SignUpActivity.class);
startActivity(i);

```

```

});

enter.setOnClickListener(view -> {
    if (Password.getText().toString().equals("") ||
    Email.getText().toString().equals(""))
        Toast.makeText(SignInActivity.this, "not all the fields are
", Toast.LENGTH_SHORT).show();
    else{
        //Connecting user using Authentication
        Log.d(Email.getText().toString(),
Password.getText().toString());
        mAuth.signInWithEmailAndPassword(Email.getText().toString(),
Password.getText().toString())
            .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult>
task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the
signed-in user's information
                    Log.d("log in",
"signInWithEmail:success");
                    FirebaseUser user =
mAuth.getCurrentUser();
                    UID = user.getUid().toString();

                    DatabaseReference docRef =
db.collection("Users").document(UID);
                    docRef.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                        @Override
                        public void onSuccess(DocumentSnapshot
documentSnapshot) {
                            if (documentSnapshot.exists()) {
                                //Taking data from firestore
                                org =
documentSnapshot.getString("org");
                                tORs =
documentSnapshot.getString("tORs");
                                Name =
documentSnapshot.getString("Name");

                                //adding data to
SharedPreferences
                                UserData.putString("org",org);
                                UserData.putString("UserName",
Name);

```


מחלקה SignUpActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.Switch;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;
import java.util.Map;

public class SignUpActivity extends AppCompatActivity {

    //General data variables
    *****

    //Firestore variables
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    //Authentication variables
    FirebaseAuth mAuth = FirebaseAuth.getInstance();

    //SharedPreferences variables
    SharedPreferences.Editor UserData;

    //User variables
    String userUID;

    //General data variables
    *****

    //Views
    Button SignUp;

```

```

Switch Teacher_Student;
EditText Name;
EditText ETEmail;
EditText Password1;
EditText Password2;
EditText Organization;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sign_up);

    //getting general vars
    *****
    UserData = getSharedPreferences("UserData", MODE_PRIVATE).edit();
    //getting general vars
    *****

    //Find views
    Name = findViewById(R.id.ETSighInUserName);
    ETEmail = findViewById(R.id.ETEmail);
    Password1 = findViewById(R.id.ETSighInPassword1);
    Password2 = findViewById(R.id.ETSighInPassword2);
    Organization = findViewById(R.id.ETOrganizationName);
    Teacher_Student = findViewById(R.id.SWTeacher_Student);
    SignUp = findViewById(R.id.BTNaddUser);

    //Sets views

    Teacher_Student.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
        public void onCheckedChanged(CompoundButton compoundButton,
boolean b) {
            switch (Teacher_Student.getText().toString()) {
                case "Teacher":
                    Teacher_Student.setText("Student");
                    break;
                case "Student":
                    Teacher_Student.setText("Teacher");
            }
        }
    });

    SignUp.setOnClickListener(new View.OnClickListener() {
    @Override
        public void onClick(View view) {

```

```

        //Check if input is correct
        if (Name.getText().toString().equals("") ||
        Password1.getText().toString().equals("") ||
        ETEmail.getText().toString().equals("") ||
        Organization.getText().toString().equals("") ||
        Password2.getText().toString().equals("")) {
            Toast.makeText(SignUpActivity.this, "not all the fields
are full", Toast.LENGTH_SHORT).show();
            return;
        }

        if
        (!Password1.getText().toString().equals(Password2.getText().toString())) {
            Toast.makeText(SignUpActivity.this, "the two passwords
don't match", Toast.LENGTH_SHORT).show();
            return;
        }
        if (Password1.getText().toString().length() < 6) {
            Toast.makeText(SignUpActivity.this, "the password must be
ut least 6 characters long", Toast.LENGTH_SHORT).show();
            return;
        }

        //Create User
        String Email = ETEmail.getText().toString();
        String Password = Password1.getText().toString();

        mAuth.createUserWithEmailAndPassword(Email, Password)
            .addOnCompleteListener(new
        OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult>
task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the
signed-in user's information

                    FirebaseUser user =
mAuth.getCurrentUser();

                    userID = user.getId();
                    Toast.makeText(SignUpActivity.this,
userID+"", Toast.LENGTH_SHORT).show();

                    //Add User To Users List
                    Map<String, Object> userToLUsersList = new
HashMap<>();

                    userToLUsersList.put("org",
Organization.getText().toString());

                    userToLUsersList.put("tORs", Teacher_Student.getText());

```

```

userToLUsersList.put("Name",Name.getText().toString());

db.collection("Users").document(userID).set(userToLUsersList)
                                .addOnCompleteListener(new
OnCompleteListener<Void>() {
                                @Override
                                public void
onComplete(@NonNull Task<Void> task) {

                                //Add User to Organization
                                User userToOrganization =
new      User(ETEmail.getText().toString(),      Name.getText().toString(),
Teacher_Student.getText().toString());

db.collection("organizations").document(Organization.getText().toString())

.collection(Teacher_Student.getText().toString()).document(Name.getText().toS
tring())

.set(userToOrganization);//line 127

                                //adding data to
SharedPreferences
                                UserData.putString("org",
Organization.getText().toString());

UserData.putString("UserName", Name.getText().toString());
                                UserData.putString("tORs"
, Teacher_Student.getText().toString());
                                UserData.apply();

                                //Move to HomePageActivity
                                Intent i =new
Intent(getApplicationContext(), SignInActivity.class);
                                startActivity(i);
                                }
                                });
        } else {
            // If sign in fails, display a message to
the user.
            Toast.makeText(getApplicationContext(),
"Authentication failed.",Toast.LENGTH_SHORT).show();
            task.getException().printStackTrace();
        }
    }
});

```

```
    }  
    });  
}  
  
private class User {  
    public String email;  
    public String name;  
    public String stu_teach;  
  
    public User(String email, String name, String stu_teach) {  
        this.email = email;  
        this.name = name;  
        this.stu_teach = stu_teach;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getStu_teach() {  
        return stu_teach;  
    }  
  
    public void setStu_teach(String stu_teach) {  
        this.stu_teach = stu_teach;  
    }  
}  
}
```

מחלקה StudentClassesActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

public class StudentClassesActivity extends AddMenuActivity {

        //General          data          variables
        *****

        //Firestore variables
        FirebaseFirestore db = FirebaseFirestore.getInstance();

        //SharedPreferences variables
        SharedPreferences UserData;

        //Activity variables
        ArrayList<String> classes;

        //User variables
        String org;
        String UserName;

        //General          data          variables
        *****

        //Views
        RecyclerView classesList;

```

```

Button unShow;
Button noHomework;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_student_classes);

    //getting general vars
    *****

    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", Context.MODE_PRIVATE);
    org = UserData.getString("org", "not found");
    UserName = UserData.getString("UserName", "not found");

    //getting general vars
    *****

    //Find views
    classesList = findViewById(R.id.RVStudentClassesList);
    unShow = findViewById(R.id.BtnDisturbanceUnShows);
    noHomework = findViewById(R.id.BtnDisturbanceNoHomeworks);

    //Sets views

    unShow.setOnClickListener(disturbanceOnClick);
    noHomework.setOnClickListener(disturbanceOnClick);

    db.collection("organizations").document(org).collection("Student")
        .document(UserName).get().addOnSuccessListener(new
    OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            ArrayList<String> dbClassesList = (ArrayList<String>)
documentSnapshot.get("Classes");
            if (dbClassesList == null) dbClassesList = new
ArrayList<>();

            classes = dbClassesList;
            updateClassesList();
        }
    });

    View.OnClickListener disturbanceOnClick = new View.OnClickListener(){
        @Override
        public void onClick(View v) {
            String disturbance = v==unShow ? "unShow" : "noHomeWork";

```

```

        Intent i =new Intent(getApplicationContext(),
ShowDisturbanceActivity.class);
        i.putExtra("name", UserName);
        i.putExtra("disturbance",disturbance);
        startActivity(i);
    }
};

private void updateClassesList() {
    RecyclerView.LayoutManager classesListLayout = new
LinearLayoutManager(StudentClassesActivity.this);
    classesAdapter classesListAdapter = new classesAdapter();
    classesList.setLayoutManager(classesListLayout);
    classesList.setAdapter(classesListAdapter);
}

private class classesAdapter extends
RecyclerView.Adapter<classesAdapter.classesViewHolder>{
    @NonNull
    @Override
    public classesViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View classView = LayoutInflater.from(parent.getContext())

.inflate(R.layout.recycleritem_studentclasses,parent,false);
        return new classesAdapter.classesViewHolder(classView);
    }

    @Override
    public void onBindViewHolder(@NonNull classesViewHolder holder, int
position) {
        String currentClassID = classes.get(position);

        db.collection("organizations").document(org).collection("Classes")

.document(currentClassID).get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot
documentSnapshot) {
                holder.className.setText((String)
documentSnapshot.get("Subject"));
            }
        });
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```



```

        Bundle bundle = new Bundle();
        bundle.putString("ClassroomId", currentClassID);
        LessonSFragment myFragment = new LessonSFragment();
        myFragment.setArguments(bundle);

        FragmentManager fragmentManager =
getSupportFragmentManager();

        FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();

        fragmentTransaction.replace(R.id.FContainerStudentClasses, myFragment);
        fragmentTransaction.commit();
    }
});
}

@Override
public int getItemCount() {
    return classes.size();
}

public static class classesViewHolder extends RecyclerView.ViewHolder{
    TextView className;

    public classesViewHolder(@NonNull View itemView) {
        super(itemView);

        className = itemView.findViewById(R.id.RITVStudentClassName);
    }
}
}
}

```

StudentHomeworksActivity מחלקה

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.Timestamp;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class StudentHomeworksActivity extends AddMenuActivity {

        //General          data          variables
        *****

        //Firestore variables
        FirebaseFirestore db = FirebaseFirestore.getInstance();

        //SharedPreferences variables
        SharedPreferences UserData;

        //User variables
        String org;
        String UserName;
        String ClassID;

        //General          data          variables
        *****

        //Views
        TextView Name;
        RecyclerView homeworksList;

        @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_student_homeworks);

    //getting general vars
    *****

    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", MODE_PRIVATE);
    org = UserData.getString("org", "not found");
    UserName = UserData.getString("UserName", "not found");

    //getting data from Intent
    Intent intent = getIntent();
    ClassID = intent.getStringExtra("ClassId");

    //getting general vars
    *****

    //Find views
    Name = findViewById(R.id.TVStudentHomeworksName);
    homeworksList = findViewById(R.id.RVHomeworkList);

    //Sets views
    Name.setText(UserName);

    updateHomeworksList();
}

private void updateHomeworksList() {
    ArrayList<Homework> homeworks = new ArrayList<>();
    db.collection("organizations").document(org).collection("Student")

    .document(UserName).collection("Homeworks").document(ClassID).get()
        .addOnSuccessListener(new
    OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            ArrayList<Timestamp> date = (ArrayList<Timestamp>)
documentSnapshot.get("date");
            if (date == null) date = new ArrayList<>();
            ArrayList<String> text = (ArrayList<String>)
documentSnapshot.get("text");
            if (text == null) text = new ArrayList<>();

            for(int i = 0; i<date.size(); i++){
                homeworks.add(new
Homework(date.get(i),text.get(i)));
            }
        }
    });
}

```

```
RecyclerView.LayoutManager homeworksListLayout = new
LinearLayoutManager(StudentHomeworksActivity.this);
HomeworksAdapter homeworksListAdapter = new
HomeworksAdapter(homeworks);
homeworksList.setLayoutManager(homeworksListLayout);
homeworksList.setAdapter(homeworksListAdapter);
    }
});
}

private class Homework {
    Timestamp TimeDate;
    String date;
    String text;

    public Homework(Timestamp timeDate, String text) {
        TimeDate = timeDate;
        Date date = timeDate.toDate();
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        this.date = sdf.format(date);
        this.text = text;
    }

    public Timestamp getTimeDate() {
        return TimeDate;
    }

    public void setTimeDate(Timestamp timeDate) {
        TimeDate = timeDate;
        Date date = timeDate.toDate();
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        this.date = sdf.format(date);
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public String getText() {
        return text;
    }

    public void setText(String text) {
        this.text = text;
    }
}
```

```

    }
}

public class HomeworksAdapter extends
RecyclerView.Adapter<HomeworksAdapter.HomeworkViewHolder>{
    ArrayList<Homework> homeworks;

    public HomeworksAdapter(ArrayList<Homework> homeworks) {
        this.homeworks = homeworks;
    }

    @NonNull
    @Override
    public HomeworkViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View HomeworkView = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.recycleritem_homework,parent,false);
        return new HomeworkViewHolder(HomeworkView);
    }

    @Override
    public void onBindViewHolder(@NonNull HomeworkViewHolder holder, int
position) {
        Homework currentHomework = homeworks.get(position);

        holder.text.setText(currentHomework.text);
        holder.date.setText(currentHomework.date);
    }

    @Override
    public int getItemCount() {
        return homeworks.size();
    }

    public static class HomeworkViewHolder extends
RecyclerView.ViewHolder{
        TextView text;
        TextView date;

        public HomeworkViewHolder(@NonNull View itemView) {
            super(itemView);
            date = itemView.findViewById(R.id.RITVHomeworkDate);
            text = itemView.findViewById(R.id.RITVHomeworkText);
        }
    }
}

```

מחלקה StudentTestsActivity

```

package com.example.educare;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.Timestamp;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class StudentTestsActivity extends AddMenuActivity {

        //General          data          variables
        *****

        //Firestore variables
        FirebaseFirestore db = FirebaseFirestore.getInstance();

        //SharedPreferences variables
        SharedPreferences UserData;

        //User variables
        String org;
        String UserName;
        String ClassID;

        //General          data          variables
        *****

        //Views
        TextView Name;
        RecyclerView testsList;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_student_tests);

    //getting general vars
    *****

    //getting data from SharedPreferences
    UserData = getSharedPreferences("UserData", MODE_PRIVATE);
    org = UserData.getString("org", "not found");

    //getting data from Intent
    Intent intent = getIntent();
    UserName = intent.getStringExtra("UserName");
    ClassID = intent.getStringExtra("ClassId");

    //getting general vars
    *****

    //Find views
    Name = findViewById(R.id.TVStudentTestName);
    testsList = findViewById(R.id.RVTestsList);

    //Sets views

    Name.setText(UserName);

    updateTestsList();
}

private void updateTestsList() {
    ArrayList<TestOutput> tests = new ArrayList<>();
    db.collection("organizations").document(org).collection("Student")
        .document(UserName).collection("Tests").get()
        .addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {
            @Override
                public void onSuccess(QuerySnapshot
queryDocumentSnapshots) {
                    for (QueryDocumentSnapshot documentSnapshot :
queryDocumentSnapshots) {
                        if
(documentSnapshot.get("classID").toString().equals(ClassID)) {
                            Timestamp timestamp =
documentSnapshot.getTimestamp("date");
                            Date date = timestamp.toDate();

                            SimpleDateFormat sdf = new
SimpleDateFormat("dd/MM/yyyy");
                            String formattedDate = sdf.format(date);

```

```

        String TestName =
documentSnapshot.getString("testName");

        int grade =
documentSnapshot.getLong("grade").intValue();

        String subject =
documentSnapshot.getString("subject");

        String teachersName =
documentSnapshot.getString("teachers_name");

        tests.add(new TestOutput(TestName, grade,
subject, teachersName, formattedDate));
    }
}

RecyclerView.LayoutManager testsListLayout = new
LinearLayoutManager(StudentTestsActivity.this);
TestAdapter testsListAdapter = new TestAdapter(tests);

testsList.setLayoutManager(testsListLayout);
testsList.setAdapter(testsListAdapter);
    }
});
}

public class TestOutput extends Test {
    String date;

    public TestOutput(String testName, int grade, String subject, String
teachers_name, String date) {
        super(testName, grade, subject, teachers_name);
        this.date = date;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}

    public class TestAdapter extends
RecyclerView.Adapter<TestAdapter.TestViewHolder> {

        ArrayList<TestOutput> tests;

        public TestAdapter(ArrayList<TestOutput> tests) {
            this.tests = tests;

```



```

    }
    @NonNull
    @Override
    public TestViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View testView = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.recycleritem_test, parent, false);
        return new TestViewHolder(testView);
    }

    @Override
    public void onBindViewHolder(@NonNull TestViewHolder holder, int
position) {
        TestOutput currentTest = tests.get(position);

        holder.testName.setText(currentTest.getTestName());
        holder.Grade.setText(currentTest.getGrade()+"");
        holder.subject.setText(currentTest.getSubject());
        holder.date.setText(currentTest.getDate());
        holder.teachersName.setText(currentTest.getTestName());
    }

    @Override
    public int getItemCount() {
        return tests.size();
    }

    public static class TestViewHolder extends RecyclerView.ViewHolder{

        public TextView testName;
        public TextView Grade;
        public TextView subject;
        public TextView date;
        public TextView teachersName;

        public TestViewHolder(@NonNull View itemView) {
            super(itemView);
            testName = itemView.findViewById(R.id.RITVTestName);
            Grade = itemView.findViewById(R.id.RITVTestGrade);
            subject = itemView.findViewById(R.id.RITVTestSubject);
            date = itemView.findViewById(R.id.RITVTestDate);
            teachersName = itemView.findViewById(R.id.RITVTestTeacher);
        }
    }
}

```

מחלקה Test

```
package com.example.educare;

public class Test {
    String TestName;
    int grade;
    String Subject;
    String teachers_name;
    public Test(String testName, int grade, String subject, String
teachers_name) {
        TestName = testName;
        this.grade = grade;
        Subject = subject;
        this.teachers_name = teachers_name;
    }
    public String getTestName() {
        return TestName;
    }

    public void setTestName(String testName) {
        TestName = testName;
    }

    public int getGrade() {
        return grade;
    }

    public void setGrade(int grade) {
        this.grade = grade;
    }

    public String getSubject() {
        return Subject;
    }

    public void setSubject(String subject) {
        Subject = subject;
    }

    public String getTeachers_name() {
        return teachers_name;
    }

    public void setTeachers_name(String teachers_name) {
        this.teachers_name = teachers_name;
    }
}
```

מחלקה Time

```
package com.example.educare;

import java.util.Calendar;

public class Time {
    int Hour;
    int Minutes;

    public Time(int hour, int minutes) {
        Hour = hour;
        Minutes = minutes;
        addTime(0,0);
    }

    public void addTime(int hour, int minutes){
        Hour += hour;
        Minutes += minutes;
        while(Minutes >= 60) {
            Hour++;
            Minutes-=60;
        }
        while(Hour >= 24) Hour -= 24;
    }

    public int getHour() {
        return Hour;
    }

    public void setHour(int hour) {
        Hour = hour;
    }

    public int getMinutes() {
        return Minutes;
    }

    public void setMinutes(int minutes) {
        Minutes = minutes;
    }

    public Calendar subtractMinutes(int minutes) {
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY, this.Hour);
        calendar.set(Calendar.MINUTE, this.Minutes);
        calendar.add(Calendar.MINUTE, -minutes);
        return calendar;
    }
}
```

```
}

@Override
public String toString() {
    String hour;
    if (Hour < 10) hour = "0"+Hour;
    else hour = ""+Hour;
    String minute;
    if (Minutes < 10) minute = "0"+Minutes;
    else minute = ""+Minutes;
    return hour+":"+minute;
}
}
```

UserLocation מחלקה

```
package com.example.educare;

public class UserLocation {
    double Latitude;
    double Longitude;

    public UserLocation(double latitude, double longitude) {
        Latitude = latitude;
        Longitude = longitude;
    }

    public double getLatitude() {
        return Latitude;
    }

    public void setLatitude(double latitude) {
        Latitude = latitude;
    }

    public double getLongitude() {
        return Longitude;
    }

    public void setLongitude(double longitude) {
        Longitude = longitude;
    }

    public boolean isWithinRange(UserLocation location) {
        int earthRadius = 6371000; // Radius of the Earth in meters
        double lat1 = Math.toRadians(Latitude);
        double lon1 = Math.toRadians(Longitude);
        double lat2 = Math.toRadians(location.getLatitude());
        double lon2 = Math.toRadians(location.getLongitude());

        double deltaLat = lat2 - lat1;
        double deltaLon = lon2 - lon1;

        double a = Math.sin(deltaLat / 2) * Math.sin(deltaLat / 2)
            + Math.cos(lat1) * Math.cos(lat2)
            * Math.sin(deltaLon / 2) * Math.sin(deltaLon / 2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

        double distance = earthRadius * c;
        return distance < 15; // Check if distance is less than 15 meters
    }
}
```