|  |
| --- |
|  |
| Family Hub  The Place Where Your Family Members Comes Together |
| |  |  |  | | --- | --- | --- | | Ben Sembira nahum | 5/25/19 | Project in Android Studio | |

# תוכן עניינים

[תוכן עניינים 2](#_Toc9703225)

[קצת בכלליות 3](#_Toc9703226)

[לוגו 3](#_Toc9703227)

[מורה – עוז דרורי 3](#_Toc9703228)

[בית ספר – מקיף רגר 3](#_Toc9703229)

[הסבר כללי 3](#_Toc9703230)

[מבנה האפליקציה 4](#_Toc9703231)

[תיאור מסכים 5](#_Toc9703232)

[Login 5](#_Toc9703233)

[משתנים ושיטות 6](#_Toc9703234)

[Register – Motivation 7](#_Toc9703235)

[משתנים ושיטות 7](#_Toc9703236)

[Register 8](#_Toc9703237)

[משתנים ושיטות 9](#_Toc9703238)

[Home Screen 11](#_Toc9703239)

[משתנים ושיטות 12](#_Toc9703240)

[Profile 14](#_Toc9703241)

[משתנים ושיטות 15](#_Toc9703242)

[Family Tree 22](#_Toc9703243)

קצת בכלליות

מורה: עוז דרורי, מוסד לימודים: מקיף רגר

## לוגו

------ להוסיף לוגו-------

## הסבר כללי

ה יישומון Family Hub הוא רשת חברתית המיועדת למשפחות גדולות. ה יישומון מאפשר צפייה וניווט בעץ המשפחה תוך גילוי שורשים חדשים וקשרים חדשים, הוא מאפשר ניהול הזמנות לאירועים גדולים בהם על כלל המשפחה להיות מוזמנים. בנוסף הוא מאפשר העלאת סטטוסים ותמונות לקיר המשפחה (ה-Feed).

# מבנה האפליקציה

Login

Home Screen

Register - Motivation

Register

Profile

Family Tree

Upcoming Events

New Event

Family Members

Settings

Side Drawer

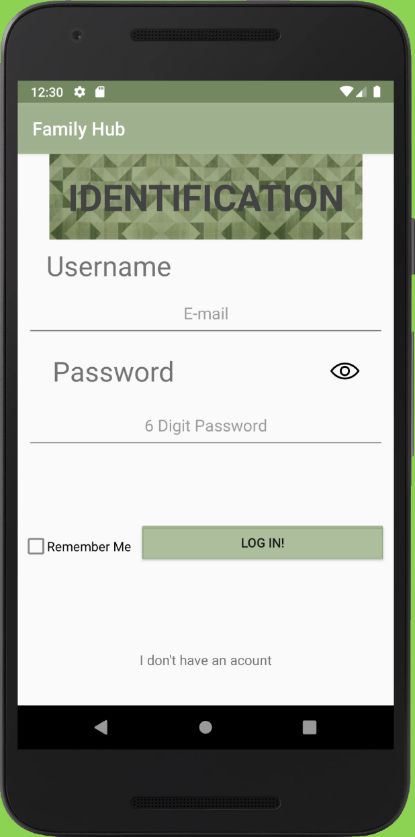
Two Way Navigation

# תיאור מסכים

## Login

LoginActivity.java, activity\_first\_login.xml

המסך הראשון שנפתח למשתמש הוא מסך ההתחברות בו המשתמש מתבקש להירשם עם אימייל וסיסמא. על המסך ארבעה כפתורים: התחבר, צור חשבון חדש, הצג\הסתר סיסמא וכפתור זכור אותי להתחברות מהירה יותר. לחיצה על הכפתורים תעביר את המשתמש לפעולה הרצויה.



### משתנים ושיטות

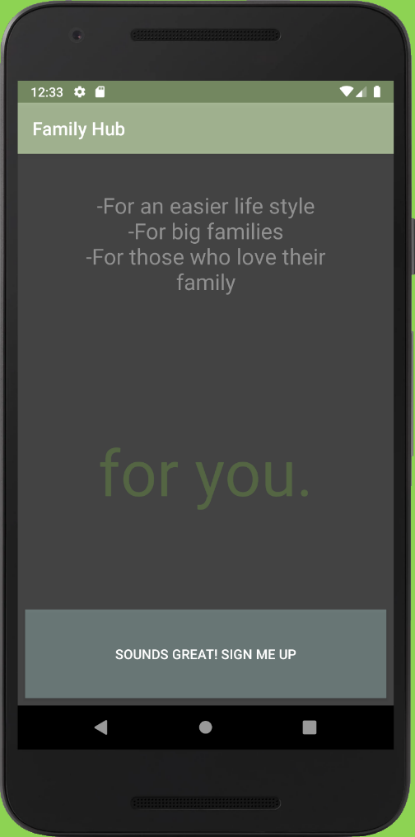
|  |  |  |
| --- | --- | --- |
| מטרת השיטה\המשתנה | חתימת השיטה\המשתנה | מספר שורה |
| מצביע לתמונת העין שבמסך האחראית על שקיפות הסיסמא | ImageView eye |  |
| מצביע לתיבת הטקסט שמחזיקה את הסיסמא | EditText pass |  |
| מצביע לתיבת הטקסט שמחזיקה את שם המשתמש | EditText user |  |
| משתנה בוליאני (דגל) של שקיפות הסיסמא | Boolean show |  |
| מצביע לאובייקט משתמש של FIREBASE, דרכו נקבל מתבצעות כל פעולות ההחברות, הרשמה וגישה לנתוני המשתמש. | private FirebaseAuth mAuth |  |
| מצביע לכפתור ה-radio "זכור אותי" | RadioButton b |  |
| מצביע לאובייקט המקשר בין כל מבנה הנתונים בשרתי גוגל לבין האפליקציה בצד המשתמש. | FirebaseDatabase database |  |
| מצביע לאובייקט המייצג רשומה במבנה הנתונים | DatabaseReference myRef |  |
| הפעולה מופעלת בעת יצירת המסך, והיא פועלת כבנאי. קוראת לפעולה המקבילה במחלקת האב ולאחר מכן משימה ערכים בכל המשתנים שהוצהרו מעל (1-8). | protected void onCreate(Bundle savedInstanceState) |  |
| מזומנת בעת לחיצה על תמונת העין, תפקיד השיטה הוא לשנות את שקיפות הסיסמא כל לחיצה. | public void showPass(View view) |  |
| שיטה אשר נקראת בעת לחיצה על כפתור "אין לי משתמש", השיטה תעביר את המשתמש למסך "Register - Motivation" | public void newAcount(View view) |  |
| שיטה אשר נקראת בעת לחיצה על "Login". השיטה תשלח בקשה למבנה הנתונים בשרתי גוגל עם שם המשתמש והסיסמא (מוצפנים), אם המשתמש קיים במערכת תקרא השיטה ל-login() אם לא תודיע למשתמש באמצעות Toast. | public void loginAttempt(View view) |  |
| השיטה תעביר את המשתמש למסך  "Home Screen" | private void login() |  |
| שיטה אשר מממשת את פעולת "זכור אותי" | public void rememberMeClicked(View view) |  |

## Register – Motivation

newAcount.java, activity\_new\_acount.xml

מסך זה הינו מסך קצר המכיל מעט משפטים קצרים על האפליקציה, וכפתור המשך להרשמה.

מטרת המסך הינו להחדיר מוטיבציה וסיפוק לאדם הנרשם וכך לגרום לו בתקווה לאהוב יותר את האפליקציה – מאין רושם ראשוני, למנוע את מחיקתה ובתקווה גם לגרום לאלו הנרשמים להמליץ עליה לחבריהם.



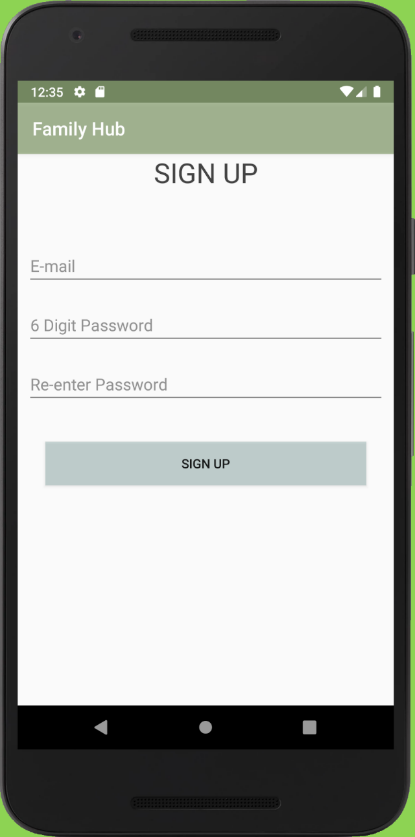
### משתנים ושיטות

|  |  |  |
| --- | --- | --- |
| מטרת השיטה\המשתנה | חתימת השיטה\המשתנה | מספר שורה |
| הפעולה מופעלת בעת יצירת המסך, והיא פועלת כבנאי. היא קוראת לפעולה המקבילה במחלקת האב. אין היא עושה דבר מעבר לזה. | protected void onCreate(Bundle savedInstanceState) |  |
| שיטה אשר נקראת בעת לחיצה על הכפתור במסך זה, השיטה תעביר למסך הבא שהוא "Register" | public void signUp(View view) |  |

## Register

signUp.java, activity\_sign\_up.xml

מסך זה הינו מסך הרשמה המכיל 3 שדות להכנסה של מייל וסיסמא (פעמיים). מתבצעת בדיקה שהשדות תקינים ואם אכן כך מתבצע ניסיון הרשמה. אם הניסיון צולח המשתמש מוכנס למבנה הנתונים ומתבצע מעבר למסך "Profile" במצב עריכה.



### משתנים ושיטות

|  |  |  |
| --- | --- | --- |
| מטרת השיטה\המשתנה | חתימת השיטה\המשתנה | מספר שורה |
| משתנים המחזיקים מצביעים לשדות המילוי של שם המשתמש, הסיסמא ואישור הסיסמא. | EditText usernameEditT, passwordEditT, checkPasswordEditT |  |
| מצביע לכפתור ההרשמה במסך. | Button signUpBtn |  |
| מצביע לאובייקט טקסט בצבע אדום הנמצא מתחת לשדות המילוי והאחראי להודיע אם יש בעיה באחד השדות. | TextView comment |  |
| מצביע לאובייקט משתמש של FIREBASE, דרכו נקבל מתבצעות כל פעולות ההחברות, הרשמה וגישה לנתוני המשתמש. | FirebaseAuth mAuth |  |
| מספרים שלמים המחזיקים את הסיסמאות שהוזנו | int passA, passB |  |
| הפעולה מופעלת בעת יצירת המסך, והיא פועלת כבנאי. היא קוראת לפעולה המקבילה במחלקת האב. אין היא עושה דבר מעבר לזה. | protected void onCreate(Bundle savedInstanceState) |  |
| שיטה אשר לוקחת את הקלט מהשדות של הסיסמאות, בודקת האם אורכן ושהן מכילות רק מספרים כנדרש. אם לא מסמנת את הסיסמאות כלא תקינות באמצעות הערך . | private void setPasswordsInVars() |  |
| שיטה אשר נקראת מהשיטה setPasswordsInVars עם כל אחד מ-הקלטים של הסיסמאות והאורך 6. תפקיד השיטה לוודא כי אורך המחרוזת pass הינו בדיוק length וכי המחרוזת הנ"ל מכילה רק מספרים (תווי אסקי בין 48 ל- 57). | private boolean isLegalPassword(String pass, int length) |  |
| שיטה אשר משנה את צבע הטקסט בכל השדות כך שכותרת כל שדב תיצבע בc וה-"רמז" שבכל שדה יצבע בhc. | public void setETColor(int c, int hc) |  |
| שיטה אשר קוראת לשיטות 7,9 ו11 וכך למעשה מכניסה את הסיסמאות מהשדות למשתנים, משנה את צבע השדות לאפור כדי לסמן שניסיון ההרשמה מתבצע וקוראת לשיטה המרכזית שאחראית על ההרשמה. | public void signUpAtempt(View view) |  |

|  |  |  |
| --- | --- | --- |
| זו השיטה המרכזית האחראית על ההרשמה, היא תחילה בודקת האם שם המשתמש והסיסמאות תקינים ולאחר מכן מבצעת ניסיון הכנסה של המשתמש למבנה הנתונים בשרתי גוגל. אם הניסיון מצליח מתבצעת הכנסה של שם המשתמש לרשימת כלל המשתמשים, נפתחת רשומה חדשה עבור המשתמש תחת האב Users, והמשתמש מועבר למסך Profile במצב עריכה. אם הניסון לא מצליח נשלחת הודעה למשתמש על ידי קריאה לשיטה מספר 13 עם הפרמטר 1 (משתמש קיים). | private void signUp() |  |
| שיטה אשר מחזירה מספר כתלות בתקינות הסיסמא:  0 – תקין.  4 – לא תקין. | private int usernameLegal(String usrName) |  |
| בהתאם למספר שקיבלה, תשלח הודעה למשתמש:  0-user created  1-username exists  2-password not proper  3-passwords not match  4-username not proper  0 מסמל הכנסה מוצלחת של המשתמש למערכת ולכן אם תיקרא שיטה זו עם המספר 0, תשלח הודעה של EROR לא ידוע. | public void data\_not\_proper\_message\_generator(int message) |  |
| שיטה המשנה את הצבע של השדה שקיבלה לאדום. שיטה זו תיקרא משיטה מס. 13 בהתאם למספר ההודעה. | private void showErorInSignUp(EditText t) |  |
| שיטה הנקראת משיטה מס. 14 והמשנה את צבע הטקסט של השדה ואת צבע ה"רמז" של השדה. | private void changeColor(EditText t, int color) |  |
| שיטה המחזירה מספר בהתאם לתקינות הסיסמה:  2 – הסיסמא לא עומדת בחוקים שהוגדרו (6 תווים מספריים)  3 – הסיסמא שהוכנסה בשדה הסיסמא לא זהה לזו שבשדה הבדיקה.  0 – הסיסמאות תקינות. | private int passMatch(int passA, int passB) |  |

## Home Screen

[homeScreen.java, activity\_home\_screen.xml]

app\_bar\_home\_screen.xml

[navHeader.java, nav\_header\_home\_screen.xml]

content\_home\_xcreen.xml

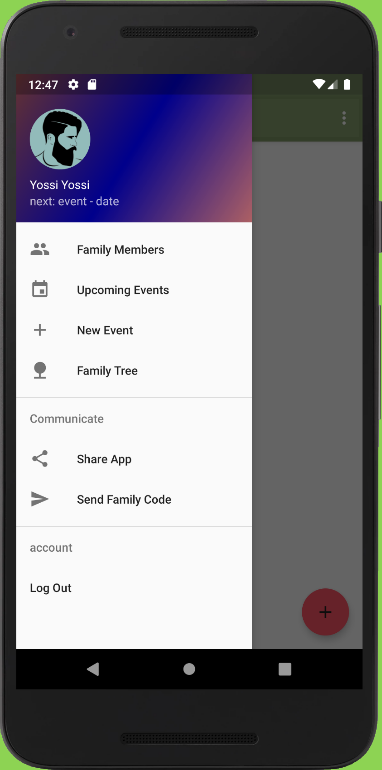
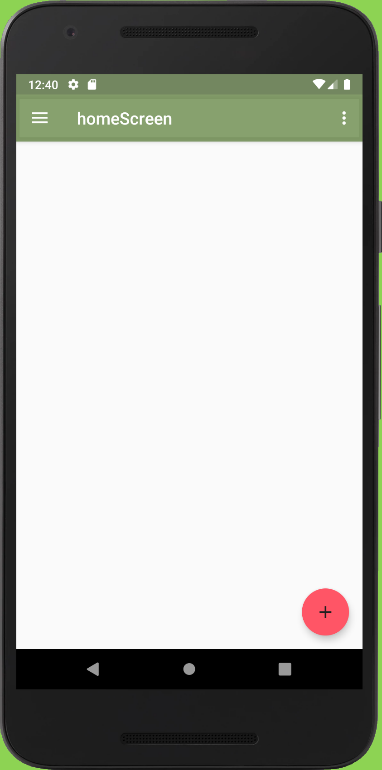
menu/activity\_home\_screen\_drawer.xml, menu/home\_screan.xml

מסך זה הינו המסך הראשי של האפליקציה והוא מכיל חלון ניווט הנשלף משמאל והמאפשר ניווט ל:

* פרופיל
* רשימת חברי המשפחה
* רשימת האירועים
* יצירת אירוע חדש
* תצוגת המשפחה (עץ המשפחה)
* שיתוף האפליקציה
* **שליחת קוד המשפחה (אופציונלי) – התגלה כלא יעיל, ניתן לשלוח סתם משפט או לבטל פונקציה זו.**
* התנתקות

בנוסף בחלק העליון של המסך ישנן 3 נקודות המובילות להגדרות האפליקציה – **עוד לא מומש.**

בחלק התחתון של המסך ישנו כפתור צף הפותח חלון למען העלאת פוסט חדש לקיר המשפחה – **עוד לא מומש.**



### משתנים ושיטות

|  |  |  |
| --- | --- | --- |
| מטרת השיטה\המשתנה | חתימת השיטה\המשתנה | מספר שורה |
| navHeader.java | | |
| הפעולה מופעלת בעת יצירת המסך, והיא פועלת כבנאי. היא קוראת לפעולה המקבילה במחלקת האב. מעבר לכך אינה עושה דבר. | protected void onCreate(Bundle savedInstanceState) |  |
| homeScreen.java | | |
| מצביעים למופעי טקסט במסך: שמו המלא של המשתמש והתאריך לאירוע הבא. | TextView navFullName, navNextEvent |  |
| מצביע למופע תמונת הפרופיל של המשתמש. | ImageView navProfilePic |  |
| הפעולה מופעלת בעת יצירת המסך, והיא פועלת כבנאי. היא קוראת לפעולה המקבילה במחלקת האב. היא מאתחלת את המשתנים וקוראת לשיטה setNav. בנוסף היא מקשרת בין כל הקבצים השונים שהוגדרו בחלק זה. היא גם אחראית להגדיר את הפעולה שתתבצע בעת לחיצה על הלחצן הצף (פוסט חדש). | protected void onCreate(Bundle savedInstanceState) |  |
| השיטה פותחת חלון חדש לעריכת הפוסט החדש שברצון המשתמש להעלות לקיר המשפחה. | private void newPost() |  |
| שיטה המופעלת מיד לאחר יצירת המסך, תפקידה לוודא שלא ניתן יהיה לחזור למסך הבית עם פרטים של משתמש כלשהו במידה ואין המשתמש מחובר (לדוגמה ביצע התנתקות). | protected void onStart( |  |
| שיטה אשר קוראת ל3 השיטות האחרות (5,6,7). היא אחראית לקשר בין רשומות בבסיס הנתונים לחלק העליון של התפריט הנשלף. בסיום ריצתה החלק העליון של התפריט הנשלף משמאל יכיל פרטים מעודכנים מבסיס הנונים בשרתי גוגל. | private void setNav() |  |
| שיטה אשר מגדירה את navFullName להיות השם הפרטי של המשתמש. | private void setName() |  |
| שיטה אשר מגדירה את navProfilePic להיות תמונת הפרופיל של המשתמש. | private void setpic() |  |
| שיטה אשר מגדירה את navNextEvent להיות התאריך לאירוע הבא של המשתמש. | private void setNextEvent() |  |

|  |  |  |
| --- | --- | --- |
| שיטה אשר מופעלת בכל לחיצה על 'חזור', בשיטה זו הוגדר אופי הפעילות של האפליקציה כאשר התפריט הנשלף פתוח, כך שאם הוא פתוח, ואז המשתמש לוחץ על חזור, ייסגר התפריט ולא תתבצע פעולת ה-'חזור' האופיינית. | public void onBackPressed() |  |
| בעת יצירת התפריט, תקשר את התפריט הנפתח לפרטים שהוגדרו מראש. | public boolean onCreateOptionsMenu(Menu menu) |  |
| בעת לחיצה על כפתורים שאינם בתפריט תבצע את הפעולות הרצויות. | public boolean onOptionsItemSelected(MenuItem item) |  |
| בעת בחירת אופציה מהתפריט הנשלף תבצע פעולה בהתאם. | public boolean onNavigationItemSelected(MenuItem item) |  |
| בעת לחיצה על תמונת הפרופיל תופעל שיטה זו ותעביר את המשתמש למסך "Profile" | public void goToProfile(View view) |  |
| בעת לחיצה על "האירוע הבא" שיטה זו תופעל ותעביר את המשתמש למסך "Upcoming Events" | public void goToEvents(View view) |  |
| שיטה אשר בהינתן עצם מטיפוס מודל של פוסט תכניס את הפוסט החדש למבנה הנתונים. | public void uploadNewPost(PostModel post) |  |

## Profile

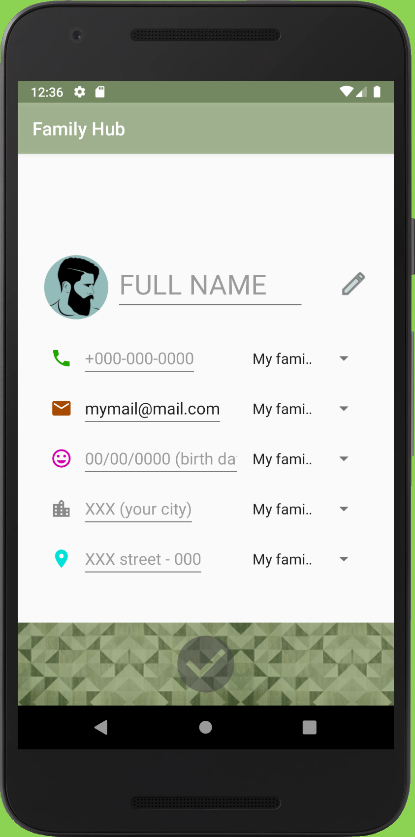
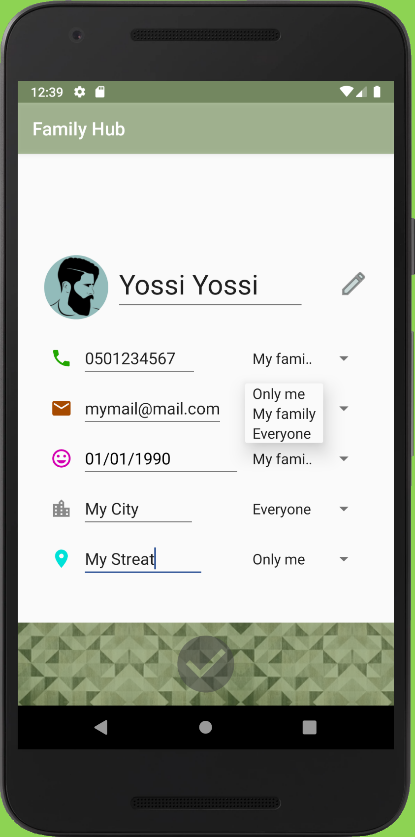
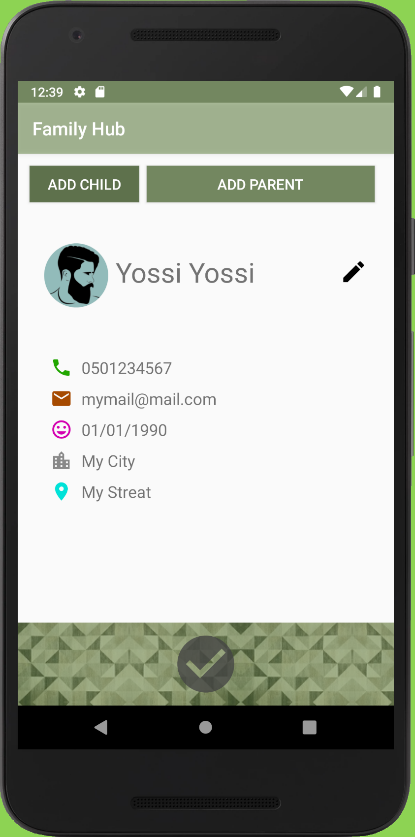
Profile.java, activiry\_profile.xml, ProfileModel.java

מסך זה מכיל את כל פרטי המשתמש הרצוי השמורים במערכת והגלויים למשתמש הנוכחי, מאפשר עריכה (עם לחיצת העיפרון) של פרטי המשתמש הנוכחי והגדרת הפרטיות של כל שדה. כל שדה בפרופיל מאפשר מעבר חכם:

* לחיצה על מספר הטלפון – מעבר ל-יישום הטלפון והתקשרות למספר זה.
* לחיצה על המייל – מעבר ל-יישום הדואר בטלפון וכניסה לחלון כתיבת הודעה חדשה למייל זה.
* לחיצה על הכתובת – מעבר ל-יישום המפות בטלפון עם הכתובת הרצויה.

ישנה אפשרות להוסיף הורה, ילד ובן\בת זוג(יופיע ליד 2 הכפתורים למעלה)

בכניסה למסך זה תופיע רשימה ל בקשות להוספת בן משפחה (הורה, ילד או בן\בת זוג) וניתן לצפות בפרופיל ולאחר מכן לאשר או לדחות.



### משתנים ושיטות

|  |  |  |
| --- | --- | --- |
| מטרת השיטה\המשתנה | חתימת השיטה\המשתנה | מספר שורה |
| ProfileModel.java | | |
| השדות הרצויים בפרופיל | private String name private String phone private String email private String date private String city private String address |  |
| בנאי המקבל שם, טלפון, מייל, תאריך לידה, עיר מגורים וכתובת. | public ProfileModel(String name, String phone, String email, String date, String city, String address) |  |
| בנאי ריק | public ProfileModel() |  |
| פעולות get ו- set לכל תכונות המחלקה. | Getters and setters | 4-15. |
| Profile.java | | |
| נייצג את השדות של מודל הפרופיל במחלקה זו עם מערכים למיניהם ועם ערך זה נגדיר את גודלם. ההערה מתחת מגדירה את משמעות כל תא בהם. | final static int *NUMBER\_OF\_PARAMETERS* = 6  //0-Name,1-Phone,2-Email,3-Bday,4-City,5-Adress |  |
| מערך של כותרות השדות. | TextView[] TextDataArray |  |
| מערך של תאי מילוי השדות. | EditText[] EditTextDataArray |  |
| משתנה בוליאני המגדיר את מצב המסך, עריכה או לא. | //Flag - enable/disable editing static Boolean *EDIT\_MODE* = false |  |
| מחרוזת המגדירה את המשתמש הנוכחי. | private static String *currentUser* |  |
| מערך מחרוזות של שדות המשתמש שבו אנו צופים, לפי הגדרות הפרטיות שהגדיר. | String[] publicData |  |
| מערך הכפתורים העליונים של הוספת בן משפחה. | Button[] RequestButtonsArray |  |
| תמונה המשמשת ככפתור לחזרה ל-"Home Screen" | ImageView goToHomeBtn |  |
| תמונת העיפרון בחלק העליון של המסך, משמשת ככפתור לשינוי מצב המסך – עריכה צפייה. | ImageView editIcon |  |
| מצביע לאובייקט משתמש של FIREBASE, דרכו נקבל מתבצעות כל פעולות ההחברות, הרשמה וגישה לנתוני המשתמש. | FirebaseAuth mAuth |  |
| רשימה מקושרת של כל המשתמשים ברשת Family-Hub. מתעדכנת כל הזמן. | LinkedList<String> allUsers |  |
| רשימה מקושרת של כל בקשות החברות שנשלחו למשתמש הנוכחי: בקשה לאב, לילד או לבן\בת זוג. | LinkedList<DataSnapshot> allRequests |  |
| רשימה מקושרת של כל בקשות החברות שנשלחו על ידי המשתמש ואושרו: בקשה לאב, לילד או לבן\בת זוג. | LinkedList<DataSnapshot> allPermissions |  |
| מצביעים לכל הספינרים של הגדרת הפרטיות המוצגים במצב עריכה. | Spinner phoneSpinner, emailSpinner, bdaySpinner, citySpinner, addressSpinner |  |
| עצם מטיפוס מודל של פרופיל המכיל את כל הפרטים של המשתמש בו אנו צופים כרגע או עורכים כרגע. עצם זה יכיל את כל הפרטים ממבנה הנתונים וישלח חזרה למבנה עם שינויים. | private ProfileModel profileData |  |
| עצם המקשר בין האחסון בשרתי גוגל (לתמונות) לאפליקציה. | private StorageReference mStorageRef |  |
| משתנה המכיל את המקום הבא שאליו המשתמש ירצה להגיע במידה ויסמן שסיימנו עם מסך זה. | private Class nextClass |  |
| פעולת GET רגילה | public static String getCurrentUser() |  |
| פעולת SET רגילה | public static void setCurrentUser(  String currentUserNameUid) |  |
| מגדירה את המשתמש הנוכחי להיות זו של המשתמש המחובר. | public static void setCurrentUserToMyUser() |  |
| הפעולה מופעלת בעת יצירת המסך, והיא פועלת כבנאי. היא קוראת לפעולה המקבילה במחלקת האב.  בנוסף היא מגדירה את כל המשתנים, המערכים, העצמים, הספינרים והרשימות. לאחר שסיימה להגדיר את כל התכונות, מגדירה את כל המאזינים של בסיס הנתונים (שיטות ההאזנה של גוגל נקראות בכל שינוי ואם הגדרתן בכניסה למסך), מגדירה את כל השדות לפי בסיס הנתונים ומקבלת ומציגה את כל הבקשות להתחברות (הורה, ילד, בן\בת זוג) | protected void onCreate(  Bundle savedInstanceState) |  |
| שיטה הנקראת משיטה מס. 21, יוצרת האזנה לשינוי במבנה הנתונים בחלק של בקשות ההתחברות וקוראת לשיטה מס. 23. | private void setRequestsListener() |  |
| שיטה הנקראת מתוך שיטת ההאזנה 22, היא מעדכנת את המערך allRequests וקוראת ל 24. | private void handleRequests)  DataSnapshot dataSnapshot) |  |

|  |  |  |
| --- | --- | --- |
| שיטה הנקראת משיטה מס. 23, מציגה את כל הבקשות מהמערך allRequests. | private void showRequests() |  |
| שיטה הנקראת משיטה מס. 21, יוצרת האזנה לשינוי במבנה הנתונים בחלק של אישורי ההתחברות מאנשים אחרים וקוראת לשיטה מס. 26. | private void setPermissionsListener() |  |
| שיטה הנקראת מתוך שיטת ההאזנה 25, היא מעדכנת את המערך allPermissions וקוראת ל 27. | private void handlePermissions(  DataSnapshot dataSnapshot) |  |
| שיטה הנקראת משיטה מס. 26, מציגה את כל הבקשות מהמערך allPermissions. | private void showPermissions() |  |
| שיטה הנקראת משיטה מס. 21, יוצרת האזנה לשינוי במבנה הנתונים בחלק של אישורי ההתחברות מאנשים אחרים וקוראת לשיטה מס. 29. | private void setAllUsersListener() |  |
| שיטה הנקראת מתוך שיטת ההאזנה 28, היא מעדכנת את הרשימה allUsers. | private void setAllUsersLL(  DataSnapshot dataSnapshot) |  |
| שיטה המגדירה את כל הספינרים באמצעות 31. | private void setSpinners() |  |
| שיטה המגדירה לכל ספינר את האפשרויות בו. | private void setSpinner(  Spinner spinner) |  |
| שיטה המגדירה את פרטיות השדה ה-i לפי ההגדרה  0-Name,1-Phone,2-Email,3-Bday,  4-City,5-Adress  לפי המחרוזת שהתקבלה אם המשתמש בחר only me השיטה תכניס לשדה הציבורי מחרוזת ריקה, אם המשתמש בחר my family השיטה תכניס לשדה הציבורי את הערך עם %f בהתחלה, ואם ביקש המשתמש שהשדה יהיה ציבורי לכולם, תכניס השיטה את הערך ללא שינויים. | private void setPrivacy(  int i, String privacy) |  |
| בסיס הנתונים של גוגל FIREBASE מאפשר הכנסה של ערכים אשר אינם מכילים '.' ולכן אנחנו מחליפים את על הנקודות '\*'. שיטה זו מבצעת את ההחלפה ובהינתן מחרוזת מחזירה מחרוזת חוקית בעבור Firebase. | public static String prepareStringToDataBase(String s) |  |

|  |  |  |
| --- | --- | --- |
| שיטה המוסיפה סלאשים בעת הצורך ומשנה את הצבע לאדום אם התאריך לא תקין, שחור עם התאריך תקין | private void setDateListener() |  |
| שיטה אשר במידה ולא הגדרנו את המשתמש ברצוננו לראות כרגע, מגדירה את המשתמש המחובר להיות זה שיוצג כעת. | public void setMyUserName() |  |
| שיטה הנקראת משיטה מס. 21, יוצרת האזנה לשינוי במבנה הנתונים בחלק של שדות הפרופיל הפרטיים וקוראת לשיטה מס. 37. | private void setDataUpdaterFromDataBase() |  |
| שיטה המגדירה את profileData לפי השדות בבסיס הנתונים. קוראת בסוך פעולתה ל 40. | private void setDataUpdaterFromDataBase(  DataSnapshot dataSnapshot) |  |
| שיטה הנקראת משיטה מס. 21, יוצרת האזנה לשינוי במבנה הנתונים בחלק של שדות הפרופיל הציבוריים וקוראת לשיטה מס. 39. | private void setPublicDataUpdaterFromDataBase() |  |
| שיטה המגדירה את המערך publicData לפי השדות בבסיס הנתונים. קוראת בסוף פעולתה ל 44. | private void setPublicDataUpdaterFromDataBase(  DataSnapshot dataSnapshot) |  |
| שיטה הקוראת ל46,48,49,52,50, תפקידה להתמודד עם נתונים חדשים המגיעים מבסיס הנתונים. | private void refreshFromDataUpdater() |  |
| מקבלת משתנה בוליאני המעביר האם שדה כלשהו שונה על ידי המשתמש ובהתאם לכך קוראת ל46 ול43, בסוך שיטה זו הנתונים ובסיס הנתונים יעברו סנכרון וישולבו ביחד לפי שינויי המשתמש. | private void refresh(  boolean hasChangedData) |  |
| השיטה תפעיל את 44 על כל אחד מהספינרים. בסוף תהליך זה, יסונכרנו נתוני הפרטיות שהמשתמש שינה יחד עם בסיס הנתונים. | private void savePrivacyInDatabase() |  |
| מקבלת משתנה בוליאני בהתאם ל-האם המשתמש שינה שדה כלשהו. השיטה תקרא ל48 ו49 ולאחר מכן אם ערך כלשהו שונה, תסנכרן עם בסיס הנתונים באמצעות 42 ו51, ואם לא תסנכרן באמצעות 52 ו44. בסוף תהליך זה יוגדרו הערכים הנכונים לפי פעולות המשתמש בעצמי הטקסט המופיעים כאשר מצב עריכה כבוי. | private void refreshEditMode(  boolean hasChangedData) |  |
| מגדירה את הספינרים לפי הפרטיות שמוגדרת בבסיס הנתונים. | private void setPrivacySpinners() |  |
| שיטה המגדירה באמצעות השדות (לפי ההסבר בשיטה מס. 32) את הערך בספינר הנתון. | private void setSelectionPrivacySpinnersPrivate(Spinner spinner,  String spinnerSelection) |  |
| השיטה תבדוק האם המשתמש הכניס את שמו למערכת, אם לא תעביר אותו ישירות למצב עריכה. | private void refreshFirstTime() |  |
| השיטה תגדיר ערכים דיפולטיביים לפרטיות השדות – My Family. | private void setPrivacyFirstTime() |  |
| השיטה תציג את הרכיבים המתאימים בהתאם למצב (עריכה או לא). | private void changeVisibality() |  |
| השיטה תשנה את תמונת העיפרון לפי מצב המסך (עריכה או לא). | private void changeIcon() |  |
| השיטה תעדכן את שדות הטקסט במסך בהתאם לבסיס הנתונים. | private void saveChangesInTextViews() |  |
| השיטה תעדכן את ביס הנתונים בהתאם לשינויים של המשתמש. | private void saveChangesInDataBase() |  |
| השיטה תעדכן את שדות העריכה במסך בהתאם לבסיס הנתונים. | private void setEditTextFromDataBase() |  |
| השיטה תחזיר את הערך בשדה העריכה בהתאם למספר השלם i  0-Name,1-Phone,2-Email,3-Bday,  4-City,5-Adress | public String getPersonalDataFromEditTexts(  int i) |  |
| השיטה תחזיר בהתאם למבנה הנתונים האם זו היא הפעם הראשונה שהמשתמש נכנס לפרופיל שלו. (ייתכן ותוסר) | private boolean isFirstTime() |  |
| השיטה נקראת מ21, היא מגדירה את כל שדות העריכה במסך. | private void setEditTextDataArrray() |  |
| השיטה נקראת מ21, היא מגדירה את כל שדות הטקסט במסך. | private void setTextDataArrray() |  |
| השיטה נקראת מ21, היא מגדירה את כל כפתורי הבקשה להתחברות במסך. | private void setRequestButtonsArray() |  |
| השיטה קוראת לשיטה הפרטית 59 כאשר נלחץ על העיפרון. | public void onOffEditMode(  View view) |  |
| השיטה תשנה את מצב העריכה () | private void onOffEditModePrivate() |  |
| השיטה תשנה מצב ל"לא עריכה" תסתנכרן עם בסיס הנתונים ולבסוף תבדוק האם כל הנתונים תקינים (לפי 64) אם כן תעבור למסך המקושר למחלקה s באמצעות 67. | private void viBtnMoveOn(Class s) |  |
| במידה והמשתמש לא הכניס שם, ובכל זאת המשיך הלאה לאפליקציה, יוכנס לו שם אוטומטי :  “VERY LAZY” | private void setDefaultName() |  |

|  |  |  |
| --- | --- | --- |
| במידה והמשתמש יבקש לצאת מעריכת הפרופיל לפני שהוא סיים להגדיר את כל השדות, יוצג בפניו תיבת דיאלוג עם הודעה כי הוא לא סיים למלא הכול ותשאל אותו האם בכל זאת רצונו הוא להמשיך הלאה. במידה והמשתמש בחר להמשיך היא תעביר אותו למסך הבא (nextClass) בהתאם למה שהוגדר קודם לכן. | private void createEmptyFieldsDialog() |  |
| השיטה תבדוק האם כל השדות מלאים ותחזיר בהתאם אמת או שקר. | private boolean allFieldsCompleted() |  |
| תחזיר אמת אם שיטה 65 תחזיר 0. | private boolean dataIsLegal() |  |
| תחזיר מספר בין 0 ל5,  0 – good, 1 – problem in 1, 2 – problem in 2, 3 – … | private int checkData() |  |
| השיטה תגדיר את nextClass להיות מסך הבית, ותבדוק באמצעות 63 אם כל הפרטים מולאו. אם לא תשלח למשתמש תיבת דיאלוג באמצעות 62 כך שלמשתמש תהיה אופציה להמשיך בכיוונו הרצוי, להתקדם הלאה, או להישאר ולמלא את מה שנשאר. | public void goToHomeAttempt((  View view) |  |
| השיטה תעבור למסך המקושר למחלקה s. | private void goTo(Class S) |  |
| השיטה תופעל בעת לחיצה על המייל במסך, היא תעבור ל-יישום הדואר האלקטרוני ותיכנס לחלון כתיבת הודעה חדשה למייל שעליו לחצנו. | public void sendEmail(View view) |  |
| השיטה תופעל בעת לחיצה על הטלפון במסך, היא תעבור ל-יישום הטלפון ותחייג למספר שעליו לחצנו. | public void call(View view) |  |
| השיטה תופעל בעת לחיצה על הכתובת במסך, היא תעבור ל-יישום המפות ותיכנס לכתובת שעליה לחצנו. | public void navigate(View view) |  |
| Uri אינו מקבל מחרוזות עם רווחים ועל רווחים להיות מוחלפים ב"+", לכן נכתבה שיטה זו. בהינתן מחרוזת שיטה זו מחליפה את כל הרווחים ב"+". | private static String setStringsCompatibleWithUri(  String s) |  |
| השיטה תשלח בקשה למשתמש ברשת להיות אבא של המשתמש הנוכחי. היא תפתח תיבת דיאלוג עם תיבת טקסט הניתנת לעריכה, בתיבה זו המשתמש יתבקש להכניס מייל של משתמש. השיטה תשלח למייל זה בקשה באמצעות 73. | public void addParent(View view) |  |
| השיטה תבדוק האם המשתמש mail קיים במערכת ולאחר מכן, אם כן תשלח למשתמש זה בקשה. אם לא תציג הודעה למשתמש.  רק לאחר שהמשתמש mail יאשר כי הוא אכן מקושר למשתמש הנוכחי בדרך שבה המשתמש ציין, הקשר יתווסף. | public void sendFather\_newKidRequest(  String mail) |  |
| השיטה תשלח בקשה למשתמש ברשת להיות בן של המשתמש הנוכחי. היא תפתח תיבת דיאלוג עם תיבת טקסט הניתנת לעריכה, בתיבה זו המשתמש יתבקש להכניס מייל של משתמש. השיטה תבדוק האם המשתמש קיים במערכת ולאח מכן, אם כן תשלח למשתמש זה בקשה באמצעות 73. | public void addKid(View view) |  |
| השיטה תבדוק האם המשתמש mail קיים במערכת ולאחר מכן, אם כן תשלח למשתמש זה בקשה. אם לא תציג הודעה למשתמש.  רק לאחר שהמשתמש mail יאשר כי הוא אכן מקושר למשתמש הנוכחי בדרך שבה המשתמש ציין, הקשר יתווסף. | public void sendKid\_newFatherRequest(  String mail) |  |
| השיטה תשלח בקשה למשתמש במערכת להיות בן\בת זוג של המשתמש הנוכחי. היא תפתח תיבת דיאלוג עם תיבת טקסט הניתנת לעריכה, בתיבה זו המשתמש יתבקש להכניס מייל של משתמש. השיטה תבדוק האם המשתמש קיים במערכת ולאח מכן, אם כן תשלח למשתמש זה בקשה. | public void addPartner(View view) |  |
| השיטה תבדוק האם המשתמש mail קיים במערכת ולאחר מכן, אם כן תשלח למשתמש זה בקשה. אם לא תציג הודעה למשתמש.  רק לאחר שהמשתמש mail יאשר כי הוא אכן מקושר למשתמש הנוכחי בדרך שבה המשתמש ציין, הקשר יתווסף. | public void sendPartner\_newPartnerRequest(  String mail) |  |
| השיטה תיקרא כאשר המשתמש ילחץ על תמונת הפרופיל שלו במצב עריכה. לאחר הלחיצה היא תבקש מהמשתמש תמונה מהגלריה ולאחר מכן תוסיף אותה לאחסון בשרתי גוגל ותקשר אותה למשתמש הזה. | public void changeImage(View view) |  |

## Family Tree