

פרוייקט בהנדסת תוכנה

אפליקציה לאנדרואיד



מגישה: גל רואש

ת.ז: 211680749

בית ספר: קרית החינוך למדעים, רחובות

כיתה: יב' 4

שנה: תשע"ח, 2018

מנחה: זהבה לביא



תוכן עניינים

3.....	הצעת פרוייקט
4.....	מרעיון לאפליקציה
5.....	מדריך למשתמש
13.....	מדריך למפתח
23.....	קוד מקור

הצעת פרוייקט

שם הפרוייקט:

DogTime

תיאור הנושא:

DogTime היא אפליקציה לחובבי כלבים, בעלי כלבים או כאלה השוקלים לאמץ כלב. האפליקציה ידיוותית למשתמש ומיועדת לספק מידע על כלבים ולעיתים גם מידע ייחודי בתחומים השונים. ניתן למצוא בה הסברים מפורטים ותמונות זנים שונים של כלבים, הצעות מומחים לטיפול בכלבים, ציטוטים על כלבים, משחקים ועוד.

למשתמשים יש מספר אופציות באפליקציה: מאגר מידע רחב על זנים שונים, הצעות ורעיונות בכל הקשור לגידול כלב, ציטוטים משעשעים על כלבים, משחק זיכרון עם תמונות כלבים, ואף פאזלים של כל התמונות שהמשתמש יבחר מהגלריה תוך יכולת לבחור את מספר חלקי הפאזל.

במסך הראשי מוצג תפריט

רשימה של שחקנים סביב המיקום הפיזי שלו. כאשר המשתמש לוחץ על שחקן הוא יועבר לפרופיל האישי של השחקן, שם יוכל לראות את פרטיו האישיים הציבוריים. ממסך זה יוכל המשתמש ליצור קשר עם המשתמש האחר כדי לקבוע עימו משחק. כדי לתת לכל משתמש אינדיקציה על המשתמש הנוסף, יידרשו בהרשמה לאפליקציה את הפרטים הרגילים (שם, מייל, מין, תאריך לידה, מדינה) וגם רמת משחק (מ1.5 עד 7) ותמונת פרופיל. את המיקום הפיזי של כל המשתמשים האפליקציה אוספת באופן אוטומטי. בכל כניסה של משתמש לאפליקציה מיקומו נרשם והוא מתעדכן במסד הנתונים של כלל המשתמשים. כך שהמשתמש יקבל בכל רגע רשימה של שחקנים סביב המיקום העדכני שלו, וגם המיקום של המשתמשים סביבו יהיה המיקום העדכני (האחרון שנרשם בכניסה לאפליקציה).

פירוט הדרישות מהמערכת:

- המערכת תבדוק את טופס ההרשמה ברמת תקינות הקלט וברמת מסד הנתונים ברשת.
- המערכת תבדוק את טופס ההתחברות ותציג למשתמש הודעת שגיאה במקרה שהפרטים לא מתאימים.
- בהרשמה המערכת תכניס את פרטיו של המשתמש למסד נתונים ברשת, מסד המשותף לכל משתמשי האפליקציה.



- בכל כניסה לאפליקציה (כאשר המשתמש מחובר) המערכת תאסוף את המיקום הפיזי של המשתמש ותעדכן אותו במסד הנתונים ברשת.
- בכניסה למסך הראשי, המערכת תיקח ממסד הנתונים ברשת את נתוני כל המשתמשים ותציג למשתמש את המשתמשים שמיקומם קרוב אליו לפי סדר (מהקרוב לרחוק).
- המערכת תאפשר לכל משתמש לעדכן את פרטיו האישיים.

מבני נתונים:

- מסד נתונים אונליין Parse (ז"ל....), המרכז את פרטי כל המשתמשים.
- ArrayList – חילוץ המשתמשים מתוך הטבלה ב- Parse לתוך ArrayList של משתמשים.

עמדת פיתוח:

- מעבד: Intel® Core™ i5
- RAM: 8 GB
- Hard Disk: 1 TB
- מערכת הפעלה: Windows 7
- סביבת פיתוח: Android Studio

מרעיון לאפליקציה

כחובבת כלבים, המגדלת כלבים כל חי, אני מוצאת את עצמי לעיתים קרובות מחפשת באינטרנט מידע, הן מתוך סקרנות והן מתוך צורך לשם טיפול בכלבים שלי. תמיד חשבתי שהיה נוח יותר אם כל המידע הרלוונטי היה מאורגן במקום אחד. הרעיון היה לפתח אפליקציה לחובבי כלבים, שתכלול מידע על זני כלבים השונים, על הטיפול בהם ואפילו משחקים.

השם **DogTime** נבחר לאחר התלבטות בין מספר הצעות. השם שנבחר הוא קליט, ומציע למשתמש מעין אתנחתא/הפסקה (Time out) לזמן איכות עם ועל כלבים.

לאחר בחירת השם ניגשתי לעיצוב הסמל בפוטושופ – היה חשוב לי שהאייקון יהיה ייחודי ובעיצוב נקי, תוך שילוב ציור של כלב, שם האפליקציה ומחוגי שעון, המתקשרים לשמה.





מדריך למשתמש ליישום

האפליקציה **DogTime** מיועדת למכשירים בעלי תוכנת ההפעלה Android. האפליקציה מתוכנתת בסביבה AndroidStudio.

יש להתקין את האפליקציה מחנות האפליקציות.

שפת המכשיר הנייד **חייבת להיות אנגלית**. במידה והשפה אחרת, ניתן לשנות את הגדרות השפה במכשיר הנייד.

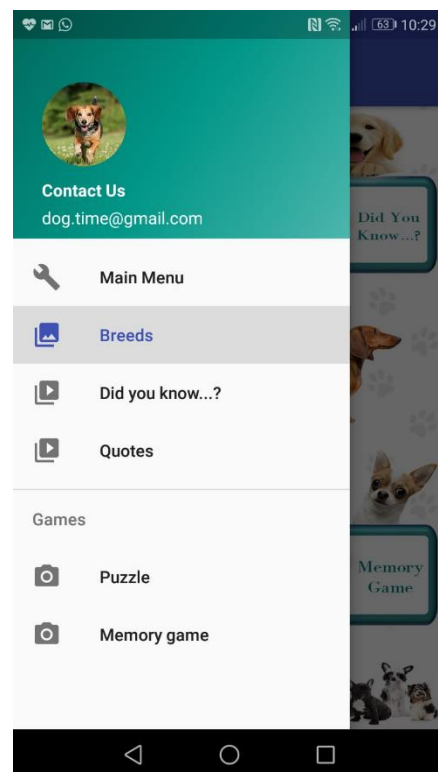
נדרשת הרשאה לקריאת זיכרון חיצוני (External storage), על מנת לשחק בפאזל.

הניווט בין דפי האפליקציה מתאפשר בשתי דרכים:

או באמצעות המסך הראשי בו ישנם סמלונים, שלחיצה עליהם מובילה לעמודים הרלוונטיים.

באמצעות לחיצה על התפריט הנשלף

המופיע בצד שמאל



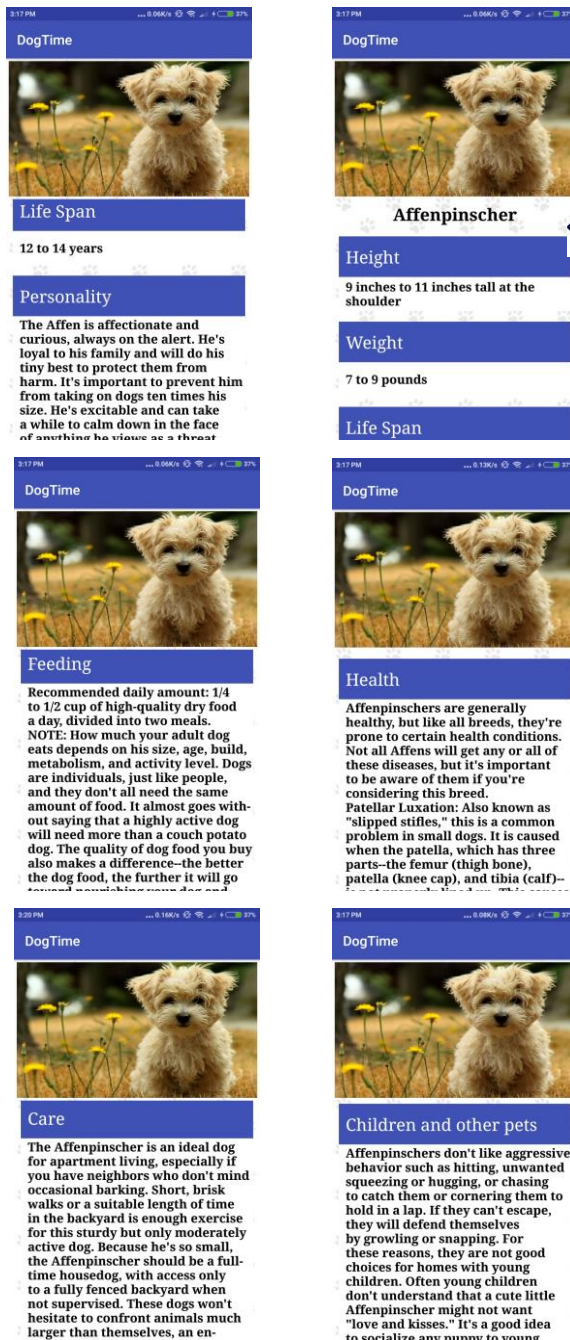
בתפריט הראשי ניתן להגיע לדפים הבאים:



1. סמלון Breeds

באפליקציה מוצע מידע אינפורמטיבי רחב על זני כלבים – גובה, משקל, תוחלת חיים, מאפיינים אישיותיים, מדע בריאותי, האכלה ועוד

ובכניסה לאחד הזנים, מתקבל מידע מפורט שניתן לגלגל את המסך ולהתקדם בו:



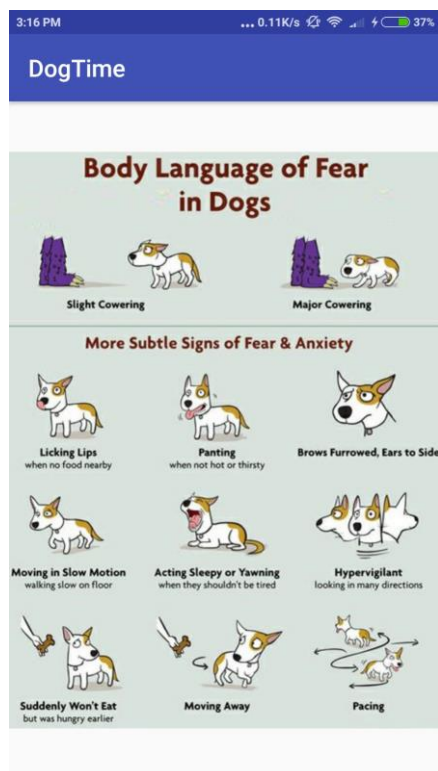
לחיצה על זכוכית המגדלת, נותנת אופציה לחיפוש זן על ידי הקלדת אותיות המוכלות בשמו.



2. סמלון? Did you know...

באפליקציה מוצע מגוון הצעות מומחים לטיפול בכלבים, עצות של "עשה ואל תעשה" מסוגים שונים – איך ילדים צריכים ולא צריכים להתנהג עם כלבים, קריאת שפת גוף של כלבים ועוד.

ובכניסה לאחת השורות, מתקבל מידע המתאים לכותרת בשורה, בו ניתן לגלגל את המסך ולהתקדם בו. כמו כן, ניתן גם להקטין ולהגדיל את התמונה הנבחרת בעזרת מגע במסך:





3. מכלון Dogs Quotes

תוספת מחוייכת המוצעת באפליקציה היא מגוון ציטוטים משעשעים ומרגשים על כלבים.
הציטוטים מופיעים על תמונות כלבים.

ניתן לעמוד על אחד הציטוטים ולראות אותו
בהגדלה. ניתן גם להקטין ולהגדיל את
התמונה הנבחרת בעזרת מגע במסך.

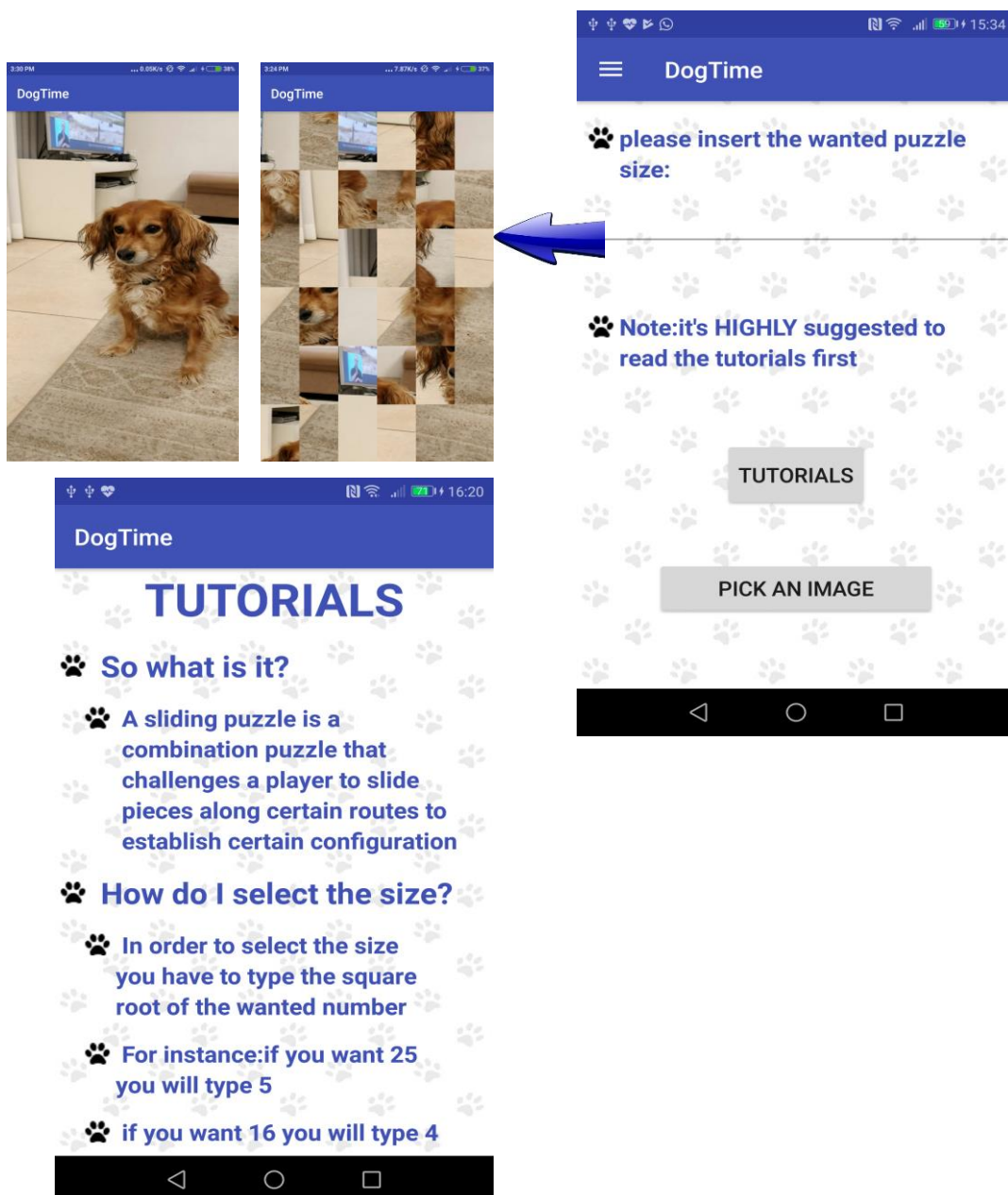




4. סמלון Puzzle

באפליקציה מוצע משחק פאזל.

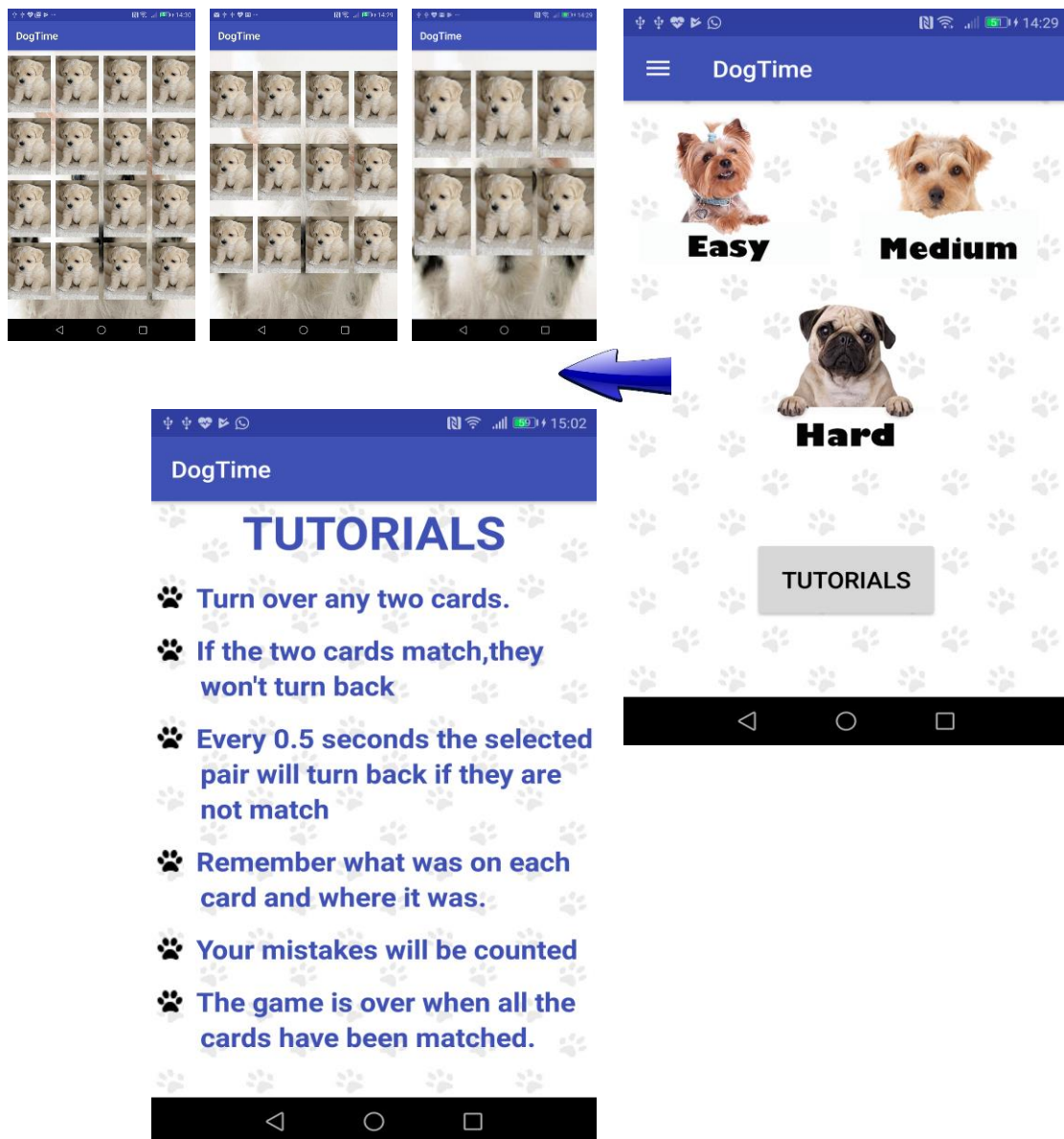
המשתמש בוחר תמונה מתוך הגלריה שלו, ובוחר מספר חלקים מבוקש בפאזל. האפליקציה מפצלת את התמונה למספר החלקים המבוקש, ומערבבת ביניהם באופן רנדומלי, כך שאפילו בבחירה שניה של אותה תמונה, סדר החלקים יהיה שונה. הפאזל הוא מסוג "sliding puzzle", כך שעל המשתמש לגרור כל חלק על ידי מגע בסך. הערה: על מנת שהפאזל יעבוד כראוי המשתמש חייב להשתמש במכשיר נייד ללא כל בעיית זכרון, כמו כן השפה חייבת להיות אנגלית.





מחלון Memory game

המשחק השני המוצע באפליקציה הוא משחק זכרון, עם תמונות כלבים. בתפריט הראשי של המשחק ניתן לבחור רמת קושי – קל (6 תמונות), בינוני (12 תמונות) או קשה (16 תמונות). לנוחות המשתמש, מצורפות הוראות למשחק. במהלך המשחק, האפליקציה מונה את מספר הטעויות שהמשתמש טועה, עד לסיום המשחק.





מדריך למפתח

האפליקציה **DogTime** מיועדת למכשירים בעלי תוכנת הפעלה Android. האפליקציה מתוכנתת בסביבת Android Studio.

סוגי הקבצים החשובים הם java class, בהם כתוב הקוד היותר אלגוריתמי והשינויים בממשק בזמן ריצה, וקבצי xml, שהם קבצים שמעצבים את הממשק למשתמש. האפליקציה מחולקת למחלקות, כשכל דף המוצג למשתמש הוא בעצם מחלקה, או בשם אחר Activity.

באפליקציה יש שימוש ב navigation drawer אשר מטרתו לאפשר למשתמש מעברים נוחים וידידותיים בין הדפים. כדי לאפשר למשתמש לעבור בין הדפים בצורה זו, כל אחת מן המחלקות אשר ניתן להגיע אליהן באופן ישיר מה navigation drawer הן fragment.

Fragments מהווים למעשה תתי activities. הרעיון הוא, שלהבדיל מ Activity אשר מספק Feature reuse בכל המערכת, Fragments מספקים Feature reuse בתוך האפליקציה עצמה. כל מופע של Fragment משויך עם מופע יחיד של Activity מארח. הוא יכול לגשת אליו, ולכן גם ל Intent object שהריץ אותו.

שימוש בספריות

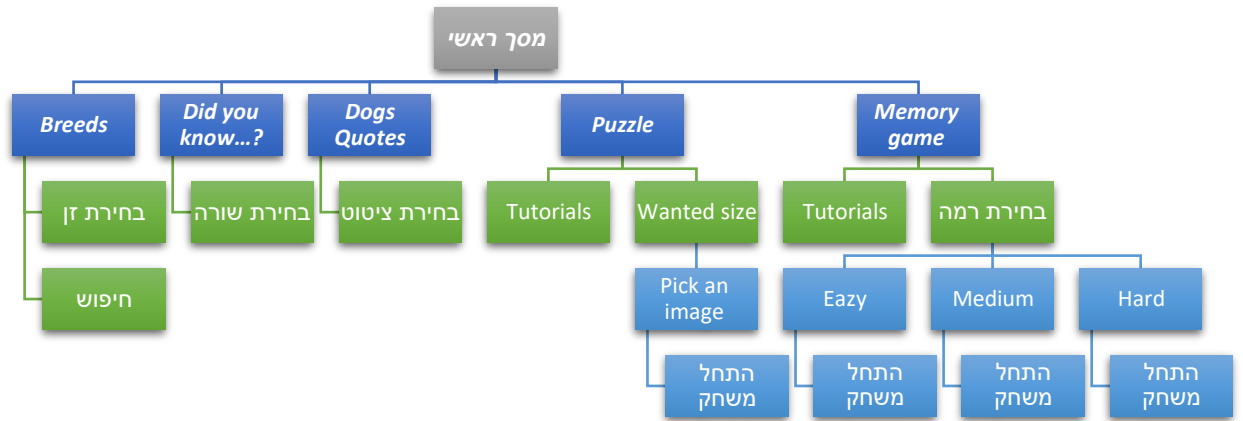
במסגרת הפרויקט עשיתי שימוש בספריית glide, אשר מטרתה היא העלאת תמונות במהירות רבה, כך שגם בטלפונים איטיים מהממוצע, התמונות תעלינה במהירות רבה והאפליקציה לא תקרוס או תיתקע.

הקישור לספרייה: <https://github.com/bumptech/glide>

בנוסף לכך השתמשתי בספרייה, אשר מטרתה לעשות zoom in and out לתמונות על מנת לאפשר למשתמש להביט בתמונות בנוחות רבה יותר.

הקישור לספרייה: <https://github.com/jsibbold/zoomage>

מסד נתונים – SQLite



מסך הבית למשתמש MainPageFragment

מסך זה הוא המסך הראשון אשר המשתמש רואה כאשר הוא נכנס לאפליקציה.

במסך זה מוצגים כל האופציות אשר האפליקציה מציגה בצורת סימלונים אשר לחיצה עליהם מובילה את המשתמש אל העמוד המבוקש.



מחלקה זו היא fragment והמחלקה MainActivity למעשה קוראת לה על ידי הקוד:

```
// goes to my wanted main fragment
MainPageFragment fragment=new MainPageFragment();
android.support.v4.app.FragmentTransaction fragmentTransaction=
    getSupportFragmentManager().beginTransaction();
fragmentTransaction.replace(R.id.fragment_container,fragment);
fragmentTransaction.addToBackStack(null);
fragmentTransaction.commit();//making this happen
```



למעשה, אני קוראת ישר לfragment שאני רוצה שהמשתמש יראה כאשר הוא יכנס לאפליקציה.

MainActivity-מחלקת האם\מארחת של כל ה"Fragments"

מחלקה זו היא בעצם הActivity "המארח" של כל הFragments. במחלקה זו נוצר הnavigation drawer-התפריט ובפונקציה onNavigationItemSelected אני בעצם דואגת שהמשתמש יועבר לאופציה אותה הוא בחר בתפריט ולסגירת את התפריט.

BreedsActivity.java

מחלקה זו היא Fragment. המשתמש מגיע למחלקה זו מהעמוד הראשי או מהתפריט. מסך זה כולל זנים רבים של כלבים ומאפשרת חיפוש זן מסוים על ידי הקלדת אותיות המוכלות בשמו.

בלחיצה על אחד מן הזנים המשתמש יעבר לדף בו יוצג המידע על הזן המסוים. כל הכלבים אשר מוצגים במסך זה נקראו ממסד הנתונים שלי וכל התכונות שלהם והמידע עליהם הוכנס לתוך רשימה ששמה myBreedsList אשר כל תא בה מסוג Dog.

כדי לעצב את המסך נעזרתי ב2 קבצי xml:

dog_list_layout.xml-כאן בעצם עיצבתי איך אני רוצה שכל תא ברשת(grid) שלי יראה.

כדי לשים טקסט על תמונה השתמשתי בframelayouth ובתוכו שמתי ImageView וTextView

fragment_breeds.xml-במחלקה זו יצרתי GridView שזה view שמציג פריטים ברשת דו מימדית ניתנת לגלילה. הפריטים מוכנסים לgridview שלי אוטומטית על ידי הadapter שכתבתי

בנוסף לכך, על מנת להכניס את כל הפריטים שלי ל gridview כתבתי מחלקת adapter אשר עושה זאת ושמה DogListAdapter מחלקה זו מרחיבה את ArrayAdapter ומממשת את Filterable. אפרט על מחלקה זו בהמשך



BreedsInfoActivity.java

למסך זה המשתמש מגיע לאחר שהוא לוחץ על אחד מזני הכלבים, מסך זה כולל מידע רב ומפורט על הזן הנבחר. גם כאן המידע נלקח מהמסד נתונים שלי אשר נתוניו נקראו והם שמורים ברשימה myBreedsList אשר העברתי כמשתנה מהמחלקה Breeds. את המידע במסך זה המשתמש יכול לגלגל, כדי שהמשתמש יוכל לגלגל עשיתי שימוש בScrollView.

FactsFragment.java

למסך זה המשתמש יכול להגיע ישירות דרך הnavigation drawer או דרך מסך הבית, מסך זה מציג מידע כללי על כלבים, עובדות, מה אסור להם לאכול, מה מותר להם לאכול, גילם בשנות בני אדם וכו', בלחיצה על אחת מן השורות במסך זה המשתמש יועבר לDisplayFcatsActivity.

גם במסך זה עשיתי שימוש בספריית glide כדי שגם בטלפונים איטיים האפליקציה תרוץ באופן חלק.

DisplayFcatsActivity.java

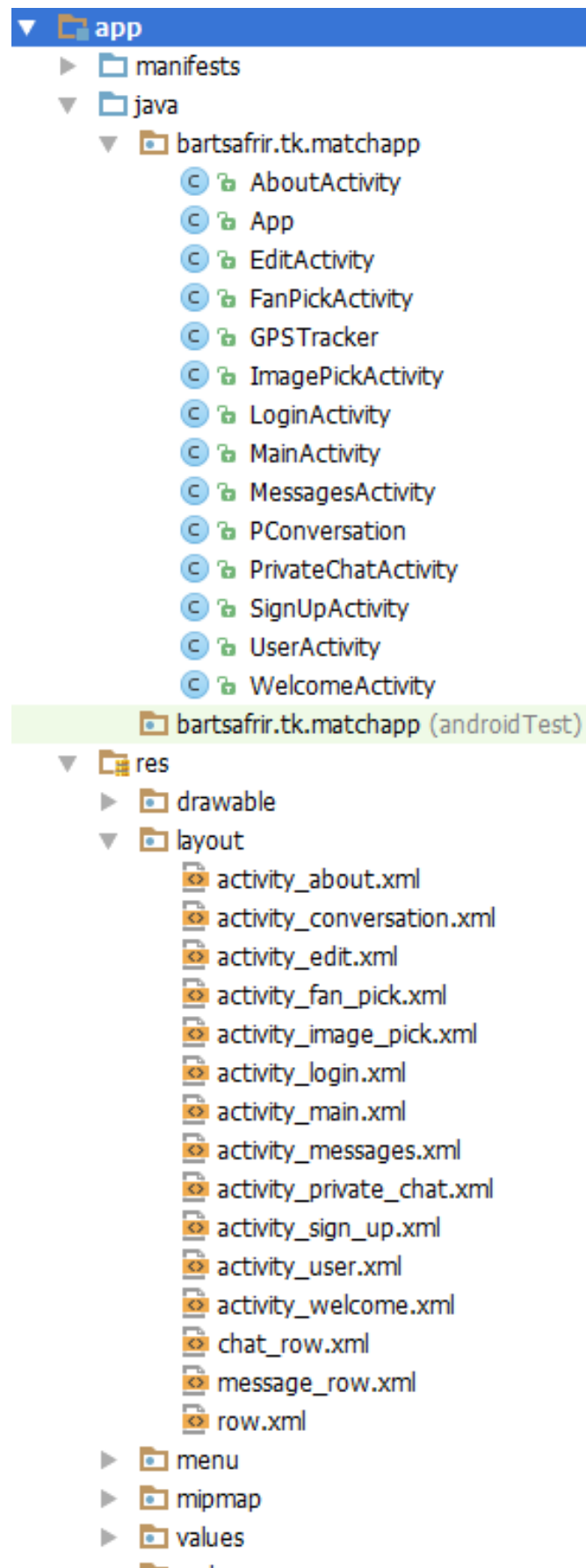
למסך זה המשתמש מגיע לאחר שהוא לחץ על אחד מן השורות שהיו במסך הקודם, במסך זה מוצג המידע אשר המשתמש רצה לראות. **כמו כן ניתן להתקרב ולהתרחק מן התמונה המוצגת על ידי מגע במסך, עשיתי זאת על ידי שימוש בספריית zoomage שלקוחה מ-github.**

QuotesFragment

DisplayQuotesActivity.java

למסך זה המשתמש מגיע לאחר שהוא לחץ על אחד מן הציטוים במסך הקודם, במסך זה מוצג הציטוט הנבחר וניתן להגדיל ולהקטין אותו







MainPageFragmentActivity.java – מסך הפתיחה

מסך זה הוא המסך הראשון שרואה המשתמש כאשר הוא נכנס לאפליקציה הוא מציג את כל האופציות המוצעות באפליקציה. (להוסיף תמונה)

מחלקה ושמה MainActivity קוראת למסך זה ישר כשהמשתמש נכנס לאפליקציה, אפרט על מחלקה זו בהמשך.

מסך זה מציג את לוגו האפליקציה, שמה וכפתור להתחלת שימוש בה. זהו המסך הראשון שנפתח בכל הפעלה של האפליקציה.

public void onClick(View v)

פונקציה המופעלת בלחיצה על כפתור "Get Started" או על לוגו האפליקציה, פונקציה זו מעבירה את המשתמש אל מסך ההתחברות (LoginActivity) שם גם תיעשה הבדיקה האם המשתמש כבר מחובר.

LoginActivity – מסך התחברות

מסך הפתיחה מעביר את מסך ההתחברות. אלא רק מי שאינו מחובר לאפליקציה מוצג בפניו מסך זה, מי שמועבר אל מסך זה ממסך הפתיחה אך הוא מחובר לחשבון MatchApp הוא מועבר ישירות אל המסך הראשי (MainActivity).

public void signUp(View v)

פונקציה המופעלת בלחיצה של המשתמש על מעבר למסך ההרשמה. היא מעבירה את המשתמש למסך ההרשמה (SignUpActivity).

protected void startLoading()

פונקציה המפעילה את הדיאלוג המציג כי האפליקציה בודקת את פרטי ההתחברות.

protected void stopLoading()

פונקציה המפסיקה את הצגת דיאלוג הטעינה.

SignUpActivity – מסך ההרשמה

דרך מסך ההרשמה יכול כל משתמש לפתוח חשבון MatchApp, מסך זה מכיל שדות למילוי פרטים, מנגנון בדיקה הבודק את כל השדות ומתריע למשתמש במקרה שאחד מהם אינו נכון בזמן אמת ולא רק בשליחת הטופס. מסך זה מציג גם אפשרות לבחור את מדינת המגורים של המשתמש, כדי לחסוך בזמן בניית מנגנון המזהה את מדינתו של המשתמש הנרשם זה עתה



(ע"י רשת טלפונית) והמערכת אוטומטית בוחרת את מדינתו ברשימה כברירת מחדל. אם ירצה המשתמש לבחור מדינה אחרת כמובן שיוכל ידנית.

public void submit(View v)

פונקציה המופעלת בעת שליחת פרטי ההרשמה ע"י המשתמש, בלחיצה על כפתור "Submit". הפונקציה בודקת תחילה שאין טעויות בטופס, ולאחר מכן יוצרת ParseUser חדש עם פרטי המשתמש ומכניסה אותו לרשימת המשתמשים. המשתמש החדש מתחבר אוטומטית לאחר שליחת הטופס. לאחר שליחת הטופס המשתמש מועבר אוטומטית למסך בחירת תמונת פרופיל.

public void signIn(View v)

פונקציה זו מופעלת כאשר לוחץ המשתמש על "Log in" ומעבירה אותו אל מסך ההתחברות.

protected void startLoading()

פונקציה המפעילה את דיאלוג הטעינה בזמן שליחת טופס ההרשמה.

protected void stopLoading()

פונקציה המפסיקה את הצגת דיאלוג הטעינה.

public static boolean validateFirstName(String firstName)

פונקציה הבודקת אם מחרוזת השם הפרטי תקנית ומחזירה T/F.

public static boolean validateLastName(String lastName)

פונקציה הבודקת אם מחרוזת שם המשפחה תקנית ומחזירה T/F.

public static boolean validatePassword(String password)

פונקציה הבודקת אם מחרוזת הסיסמה תקנית ומחזירה T/F.

public static boolean isThisDateValid(int day, int month, int year)

פונקציה הבודקת אם תאריך הלידה תקני ומחזירה T/F.

public boolean validateGender()

פונקציה הבודקת אם סומן מגדר המשתמש ומחזירה T/F.

public final static boolean isValidEmail(CharSequence target)

פונקציה הבודקת אם מחרוזת הדוא"ל תקנית ומחזירה T/F.

public void genderClick(View v)

פונקציה המופעלת בעת לחיצה על אחד מלחצני ה-radio של המגדר, בעת לחיצה כזו הפונקציה מעלימה את הכתובית (אם מוצגת) המורה על כך שלא סומן מגדר.



ImagePickActivity – מסך בחירת תמונת פרופיל

אל מסך זה מגיע כל משתמש אחרי ששלח את טופס ההרשמה וגם כל משתמש רשום שבמסך עריכת פרטיו לחץ על שינוי תמונת פרופיל. המשתמש לוחץ על כפתור בחירת התמונה ומועבר לגלריית הטלפון שלו. ממנה הוא בוחר תמונה אחת ומוחזר אוטומטית אל המסך, בו תוצג תצוגה מקדימה של התמונה שבחר. אם ברצונו לקבוע תמונה זו כתמונת הפרופיל שלו הוא לוחץ על כפתור ההמשך בתחתית המסך, אם ברצונו להחליף את התמונה שבחר זה עתה הוא יכול לעשות זאת בעזרת לחיצה שוב על לחצן הבחירה.

public void showDialog(View v)

בעלת לחיצה על בחירת תמונה, מופעלת פונקציה זו המעבירה את המשתמש אל דיאלוג הגלריה שלו, לאחר לחיצה על תמונה יוחזר המשתמש אל מסך זה.

protected void onActivityResult(int requestCode, int resultCode, Intent data)
{

פונקציה המופעלת בעת חזרת המשתמש ממסך הגלריה אל ה"ל". הפונקציה מאחסנת באופן מקומי את התמונה עד שיבחר המשתמש להעלות אותה לשרת. היא גם מציגה על המסך את התצוגה המקדימה של התמונה הנבחרת.

public void upload(View v)

בעת לחיצה על העלאת התמונה הנבחרת מועלית התמונה אל השרת ונשמרת בשורת המשתמש המחובר, כך נקבעת כתמונת הפרופיל שלו. אם המשתמש הגיע אל מסך זה ממסך ההרשמה הוא יועבר למסך בחירת השחקנים. אם הוא הגיע ממסך עריכת הפרטים הוא יוחזר אל מסך זה.

protected void startLoading()

פונקציה המפעילה את דיאלוג הטעינה בזמן העלאת תמונת הפרופיל לשרת.

protected void stopLoading()

פונקציה המפסיקה את הצגת דיאלוג הטעינה.

FanPickActivity – מסך בחירת שחקנים אהובים

אל מסך זה מועברים הן משתמשים חדשים ממסך בחירת תמונת פרופיל והן משתמשים קיימים ממסך עריכת הפרטים. במסך זה מוצגות 2 שורות של שחקנים – גברים ונשים. לכל



שחקן מוצגת תמונתו וכפתור radio המאפשר את בחירתו כשחקן האהוב. על כל משתמש לבחור 2 שחקנים – אחד מהגברים ואחת מהנשים.

public void clickCheck(View v)

לשם הנוחות, גם לחיצה על תמונת השחקן תפעיל את כפתור ה-radio שלו. פונקציה זו נקראת בעת לחיצה על אילו מהתמונות המוצגות. הפונקציה מקבלת כפרמטר את האובייקט הנלחץ ובודקת איזו תמונה נלחצה, לתמונה הנלחצת מתבצעת אוטומטית לחיצה על כפתור ה-radio של אותו השחקן/ית.

public void update(View v)

פונקציה המופעלת בלחיצה על כפתור שמירת הבחירות, היא בודקת אילו 2 שחקנים נבחרו ומעדכנת זאת בשורת המשתמש בטבלת המשתמשים. אם המשתמש הגיע אל מסך זה ממסך בחירת תמונת פרופיל הוא יועבר למסך הראשי. אם הוא הגיע ממסך עריכת הפרטים הוא יוחזר אל מסך זה.

protected void startLoading()

פונקציה המפעילה את דיאלוג הטעינה בזמן שמירת בחירות המשתמש.

protected void stopLoading()

פונקציה המפסיקה את הצגת דיאלוג הטעינה.



MainActivity – המסך הראשי

זהו המסך הראשי של האפליקציה ממנו מתבצעת הפעילות העיקרית באפליקציה. מסך זה הוא מסך הניתן לריענון ע"י החלקת האצבע כלפי מטה. ההיגיון שבניתני עבור מסך זה: קואורדינטות המיקום של המשתמש נקבעות, הן נשמרות בעמודת המיקום של המשתמש ב- Parse. לאחר מכן מתבצע סריקה של כל שאר המשתמשים במסד, עבור כל משתמש ברשימה מחושב המרחק בינו לבין המשתמש הנוכחי. לאחר מכן מתבצע מיון לפי המרחק, מהקרוב לרחוק. לאחר המיון נבחרים 10 המשתמשים הראשונים ברשימה – אלו 10 המשתמשים הקרובים ביותר למשתמש הנוכחי. בסריקה שנייה, הפעם רק של 10 המשתמשים הקרובים, מיוצאים פרטיהם הרלוונטיים להצגה ברשימה (שם, רמה, מיקום). פרטים אלו מוכנסים אל תבנית עיצוב שבניתני, תבנית של שורה ברשימה. השם מוכנס, הרמה (מספר) נבדקת ובמקומה מוצגת הרמה באופן גרפי, והמיקום נבדק ומחושב ביחס למשתמש הנוכחי ומוצג עיגול המרחק ביחידות המתאימות. לחיצה על אחד מ-10 המשתמשים ברשימה מעביר את המשתמש הנוכחי אל פרופיל המשתמש הנבחר.

public void onRefresh()

פונקציה זו מופעלת בעת ריענון הרשימה ע"י המשתמש.

public boolean onCreateOptionsMenu(Menu menu)

פונקציה זו יוצרת את התפריט בצד הימני העליון של המסך.

public boolean onOptionsItemSelected(MenuItem item)

פונקציה זו מטפלת בבחירה של פריט בתפריט העליון.

public void logOut()

פונקציה זו מתנתקת מהמשתמש הקיים ומעבירו אותו אל מסך ההתחברות.

private class LongOperation **extends** AsyncTask<String, Void, String>

AsyncTask המתבצע בעת יצירת הפעילות ובעת ריענון.

public void showLoc()

פונקציה הנקראת מתוך ה-AsyncTask והיא אוספת את מיקום המשתמש ומכניסה אותו את המסד. לאחר השמירה נקראת הפונקציה `updateList`.

public void updateList()

פונקציה הנקראת לאחר שמירת המיקום הנוכחי של המשתמש במסד. פונקציה זו מייבאת את רשימת כל המשתמשים מטבלת המשתמשים ומעבירה אתה כפרמטר לפונקציה `updateList2`.



public void `updateList2`(**final** ParseUser[] userArray)

פונקציה הנקראת לאחר ייבוא רשימת כל המשתמשים. בפונקציה זו מתבצע מיון של כל המשתמשים לפי המרחק מהם למשתמש הנוכחי, מהקרוב לרחוק. לאחר המיון מוכנסים רק 10 הראשונים במיון לתוך מערך המכיל את 10 המשתמשים הקרובים אל המשתמש הנוכחי. אל מערך הפרטים מוכנסות המחרוזות שיוצגו ברשימה, המיקום מומר למרחק מהמשתמש הנוכחי בקירוב וביחידות מתאימות. אל מערך התמונות מוכנסות 10 התמונות של המשתמשים. לאחר סיום הכנסת הפרטים אל המערכים המתאימים נקראת הפונקציה `updateList3`.

public void `updateList3`()

בפונקציה זו מתבצעת ההכנסה של פרטי 10 המשתמשים לתבנית התצוגה שבנית, ובנוסף גם המרת הרמה המספרית של המשתמש לתמונה מתאימה.

UserActivity – מסך פרופיל משתמש

מסך זה הוא מסך פרופיל משתמש, ה-Intent המעביר אל מסך זה מכיל בתוכו extra של objectId של המשתמש שזהו הפרופיל שלו. המערכת לוקחת ע"פ ה-objectId את כל פרטי המשתמש ומציגה אותם. (אם מסך זה מוצג ע"י לחיצה על My Profile בתפריט הראשי כפתור שליחת ההודעה יוחלף לכפתור עריכת הפרטים האישיים).

public void `list`()

פונקציה זו מוציאה מהמסד את אובייקט המשתמש המבוקש מהטבלה.

private String `getAge`(String bday)

פונקציה זו מקבלת את תאריך הלידה של המשתמש ומחזירה את גילו.

public void `show`(ParseUser theUser)

פונקציה זו מציגה את פרטי המשתמש על המסך (תמונתו, שמו, גילו, רמתו, שחקניו האהובים).

private class LongOperation **extends** AsyncTask<String, Void, String>

AsyncTask הנקרא מתוך פונקציית `list` ומבצע את פונקציית `show`.

public boolean `onOptionsItemSelected`(MenuItem item)

בלחיצה על כפתור החזרה בראש המסך.

PrivateChatActivity – מסך הצ'אט



מסך זה הוא מסך שיחה פרטי בין המשתמש הנוכחי למשתמש אחר. בתחתית המסך יש תיבה לכתיבת הודעה חדשה. הודעות קודמות בצ'אט יוצגו לפי סדר כרונולוגי (מהחדש מטה לישן מעלה). הודעות שנשלחו ע"י המשתמש הנוכחי יוצגו בירוק בצד השמאלי של המסך והודעות שנשלחו ע"י המשתמש השני יוצגו בלבן הצד הימני של המסך. בעבור כל הודעה יוצג תוכנה ותאריך שליחתה. המערכת בעצם ניגשת את טבלת ההודעות של כל המשתמשים ולוקחת ממנה רק את רשימת ההודעות שנשלחו בין 2 המשתמשים הרלוונטיים, מסדרת אותן לפי סדר כרונולוגי ומציגה אותן בהתאם. המערכת בודקת אוטומטית בכל שנייה האם התקבלו הודעות חדשות כך שהמשתמש אינו צריך לעשות דבר כדי לקבל את ההודעות החדשות, הן יופיעו אוטומטית.

```
public boolean onCreateOptionsMenu(Menu menu)
```

פונקציה זו יוצרת את התפריט בצד הימני העליון של המסך (שמכיל כפתור אחד המוביל לפרופיל המשתמש עמו השיחה מתנהלת)

```
public boolean onOptionsItemSelected(MenuItem item)
```

פונקציה זו מטפלת בבחירה של הפריט בתפריט העליון.

```
private class LongOperation extends AsyncTask<String, Void, String>
```

AsyncTask המבצע את פונקציית createConversation.

```
public void createConversation()
```

פונקציה הניגשת אל טבלת ההודעות ומוציאה ממנה רק את ההודעות שנשלחו ממשתמש 1 למשתמש 2 או להפך. מתוך פונקציה זאת נקראת פונקציית updateList.

```
public void updateList()
```

פונקציה זו לוקחת את רשימת ההודעות הרלוונטיות ומציגה אותן ברשימה.

```
private class countOperation extends AsyncTask<String, Void, String>
```

AsyncTask המתבצע בכל פרק זמן של שנייה ומתוכו נקראת פונקציית countMessages.

```
public void countMessages()
```

פונקציה זו בודקת את מספר ההודעות בין 2 המשתמשים הרלוונטיים. אם מספר זה גדול יותר מהקודם משמע התקבלו הודעות חדשות ולכן ה-countOperation קורא ל-longOperation המרענן מחדש את התצוגה של ההודעות ובכך מציג את ההודעות החדשות.

```
public void send(View v)
```



פונקציה זו נקראת בלחיצה על כפתור שליחת ההודעה. היא יוצרת אובייקט הודעה עם הפרטים מתאימים ומכניסה אותה לטבלת ההודעות, בנוסף היא מוסיפה את ההודעה לרשימת ההודעות המוצגות.

MessagesActivity – מסך השיחות

במסך זה מוצגת רשימת המשתמשים עימם המשתמש הנוכחי מנהל שיחה. המשתמשים יוצגו לפי סדר שליחת ההודעה האחרונה, כך שהשיחה הפעילה האחרונה תוצג ראשונה. שיחות בהן הודעות חדשות יסומנו עם נקודה ירוקה לצידן.

public void onRefresh()

פונקציה זו מופעלת בעת ריענון הרשימה ע"י המשתמש.

private class LongOperation1 extends AsyncTask<String, Void, String>

.getAllUsers המפעיל את פונקציית

public void getAllUsers()

פונקציה המעדכנת את שדה chatWith במסך, שהוא רשימת הכתובות שאיתן המשתמש הנוכחי מצוי בשיחה. הפונקציה קוראת לפונקציה

public void getChatWithUsers(ParseUser[] userArray)

הפונקציה מקבלת את רשימת כל המשתמשים ומתכם יוצרת מערך רק של אלו שהמשתמש מצוי איתם בשיחה. היא מכניסה למערך מתאים את פרטיהם הרלוונטיים ומוצאת עבור כל משתמש את ההודעה האחרונה שלו עם המשתמש הנוכחי (דבר ההכרחי למיון ע"פ זמן ההודעה האחרונה).

public void sort()

פונקציה הממיינת את המשתמשים הרלוונטיים לפי זמן ההודעה האחרונה, מהחדש לישן.

public void show()

פונקציה המציגה את המשתמשים הרלוונטיים לפי הסדר ברשימה הניתנת לריענון.

EditActivity – מסך עריכת פרטים אישיים

במסך זה יכול המשתמש לערוך את רמתו, את סיסמתו, ולגשת למסכי שינוי תמונת הפרופיל ושינוי שחקנים אהובים.

public void changeImage(View v)



פונקציה הנקראת בלחיצה על שינוי תמונת פרופיל ומעבירה את המשתמש למסך המתאים.

public void changeFav(View v)

פונקציה הנקראת בלחיצה על שינוי שחקנים אהובים ומעבירה את המשתמש למסך המתאים.

public void changePass(View v)

פונקציה הנקראת אחרי שהמשתמש מילא את השדות המתאימים לשינוי הסיסמה. המערכת בודקת את תקינותם ומשנה את סיסמתו של המשתמש הנוכחי.

AboutActivity – מסך אודות

במסך זה מוצגות אודות האפליקציה ומפתחה.



קוד מקור