

שער



בית הספר  
העירוני השש שנתי  
ע"ש מ.שרת  
נתניה

בית ספר: משה שרת נתניה

שם האפליקציה: Mapnector

שם: תימור פיצ'חדזה

ת.ז: 324881986

המורה: עופר גלעדי

שם החלופה: תכנון ותכנות מערכות טלפונים ניידים בjava

תאריך הגשה: יוני 2021/תשפ"א

תוכן עניינים:

עמוד שער - 1

תוכן עניינים - 2

מבוא- 3

ארכיטקטורה/מבנה הפרוייקט ומטרותיו- 5

אלגוריתמים עיקריים- 10

פירוט מחלקות - 13

מבני נתונים-16

שימוש באבני יסוד-16

שמירת נתונים- 17

מדריך למשתמש- 21

אילוצי תכנה ודרישות מערכת- 25

מדריך למפתח- 26

רפלקציה- 33

ביבליוגרפיה- 34

נספחים - 34

## מבוא:

בחשיבת הרעיון לפרוייקט לא הצלחתי להגיע למשהו חד משמעי - רציתי מצד אחד לנסות לפתח רשת חברתית, תיבות דואר, אולי אפשרות כמו שיש Snapchat שמאפשרת לראות לאנשים איפה כל אחד למצוא וכך להפגש בקלות? מכיוון שהיו הרבה מאוד רעיונות בראש ומוטיבציה לנסות ליישם כמה שיותר דברים בפרוייקט החלטתי להתמקד לא בתחום אחד אלה בלנסות ולהכניס באפליקציה את כל הרעיונות שהיו בראש, אז אולי היא תהיה שילוב של google maps, gmail, whatsapp. אך תמיד צריך להתחיל מקטן ולשאוף לגדול.

אז את ההשראה שאבתי מאפליקציות רבות אחרות כגון מסנג'רים, תיבות דואר ועוד.

התחלתי את הפרוייקט מיצירת מערכת שבה אנשים יכולים להכנס לקבוצות ולצאת מהן, וה firebase ישמור את כל המידע, אבל עשיתי בהתחלה את הכל בקלאס אחד שיצא יותר מ-800 שורות והקוד שלי אכל אותי לארוחת בוקר. הייתי צריך להתחיל את הפרוייקט מחדש.

הבנתי שצריך לתכנת בשיטת הפרד ומשול - כמה שאפשר להפריד בין קטעי קוד שעוסקים בעולמות שונים כדי להמנע מבאגים. לבדוד מערכות קלאסים שמתעסקות במשהו אחד לבין אלו התעסקות במשהו אחר.

אז נכון לעכשיו החלקים העיקריים בקוד לפי הסדר שרשמתי אותם:

Login System - החלק שאחראי על ה login וה signup, החלק הכי פשוט בפרוייקט

Group System - אחראי על יצירת קבוצות והצטרפות אנשים אליהם.

Location System - אחראי על שליחת מידע ואסיפתו והצגתו על מיקום של משתתפים אחרים במידה וההרשאות מאפשרות זאת.

Mail System - לכל משתמש תהיה תיבת דואר משלו - שבה יוכל לקבל הודעות מערכת כגון על כך שהודח מקבוצה, שבקשתו להצטרפות לקבוצה אושרה, או הודעות רגילות בין משתמשים אחד לאחד.

Chat System - לכל קבוצה יהיה chat, שבו המשתמשים יוכלו להתכתב. כמו כן יהיה אפשר לסמן listened chat, כלומר המשתמש יקבל notifications על הודעות מקבוצה זו גם מחוץ לאפליקציה.

אלו הם החלקים הלוגיים שחילקתי בראש שאותם מבצעת האפליקציה, פירוט רחב יותר יהיה בהמשך התיק, אבל נכון לעכשיו אלו הם החלקים העיקריים בפרוייקט.

בהתחלה, כמו כל פרוייקט נהנתי לכתוב אותו אבל ככל שהקוד התרחב כך גם כמות הבאגים, וכל הזמן נדרשתי לשנות את אופן סידור הקוד. הדבר הכי קשה היה לבדוק באגים שקשורים לbroadcast receiver, כי הוא הקשיב למצב שבו הטלפון נדלק והפעיל service בעת הדלקת הטלפון, כלומר לא הייתה לי אפשרות לדאבג בצורה יעילה חלק זה. אני מרגיש שיכולתי לעשות את הפרוייקט בצורה יותר מעניינת ויותר רחב אבל הרבה מאוד זמן ושורות קוד הלכו על ניסיון לעשות את האפליקציה user-friendly, לדוגמה היא דינמית, אפשר להשתמש בה גם במצב portrait וגם במצב landscape, כמו כן היא לא internet sensitive, לדוגמה אם אדם אחד ניסה להצטרף לקבוצה אבל בקשתו לא הגיעה כי אין אינטרנט ובו זמנית קבוצה זו התמלאה צריך לטפל במקרים כמו זה...

טוב זה על החלק הטכני - מה הפרוייקט מציע נכון לעכשיו? התכנה Mapnector שקראתי לה כך כי היא מקשרת(connect) בין אנשים עם אפשרות לשימוש במפה. נכון לעכשיו היא מהווה מקום שבו אנשים יכולים להצטרף לקבוצות ולהתכתב בצ'אט, גם כן הם יכולים לראות את מיקומם אחד של השני במידת הצורך וכך

לדוגמה להפגש במקום מסויים בפשטות. לכל אחד יש אפשרות לשלוח דואר פרטי בלי חובה להכנס לצאט של קבוצה ולהקשיב לקבוצה בnotifications כמו שווטסאפ עושה גם כשהאפליקציה סגורה.

בשתי מילים - התכנה מהווה רשת חברתית.

הדילמה העיקרית בטרם כתיבת הפרוייקט הייתה לאיזה minapi לעשות target. מצד אחד, היה לי צורך בrecyclerView, אבל בעת תחילת כתיבת הפרוייקט כמות המשתמשים שהשתמשו בapi גבוהה מ24 עמדה על כ70%. קראתי באינטרנט שמומלץ לעשות target לapi שכמות המשתמשים בו עומדת על יותר מ90%. אולי באמת היה עדיף לקחת 24 api כי יעילות הrecycleView גבוהה בהרבה מזו של ה listView ובטענה שכמות המשתמשים תגדל מ70% אבל יש לנו בבית שני טלפוני אנדרואיד ישנים (מתחת לapi24), ומכיוון שאני החלטתי לבנות מאין רשת חברתית יהיה קשה לבדוק באגים ללא כמות מספיקה של מכשירים. אז במבט לאחור, טוב שהשתמשתי בapi נמוך יותר כי זה אפשר לי לבדוק את האפליקציה עם כמות גדולה יותר של טלפונים ולמצוא באגים יותר מהר. (לדוגמה בעזרת זה שניסיתי להכנס דרך שלושה טלפונים במקביל לקבוצה גיליתי על הבעיה של race condition ושהייתי צריך להתמש בtransaction). אז שמתי minapi=19 שזה של הטלפון בעל api הקטן ביותר.

\*אני רוצה לציין כי תאריך הגשת המסמך הסופי הוא 13/6/2021, אמנם אני אמשיך לעבוד על באגים/לבצע שינויים קטנים לשיפור הפרוייקט עד למועד הגשת הפרוייקט עצמו - אז יכול להיות שיהיו תוספות או שינויים קטנים אבל החלק העיקרי (99%) ישאר כמו שהוא.

## ארכיטקטורה, מבנה הפרוייקט ומטרותיו

### הפתרון המוצע

אז כמו שכבר ציינתי במבוא, מטרתי העיקרית בכתיבת הפרוייקט הייתה לא להגיע לאפליקצייה מסויימת אלה יותר ליישם כמה שיותר תחומים מן החומר הנלמד וללמוד עוד דברים בעצמי, ולממש זאת בצורה יעילה ושימושית בפרוייקט, מה שהתגבש למטרה סופית שהיא ליצור רשת חברתית תוך התמקדות בחווית השימוש של המשתמש באפליקצייה (לדוגמה לא להעיף אותו מהאפליקצייה כל פעם שהולך האינטרנט...). נסביר עכשיו יותר בפירוט את יכולות האפליקצייה:

### יכולות האפליקצייה

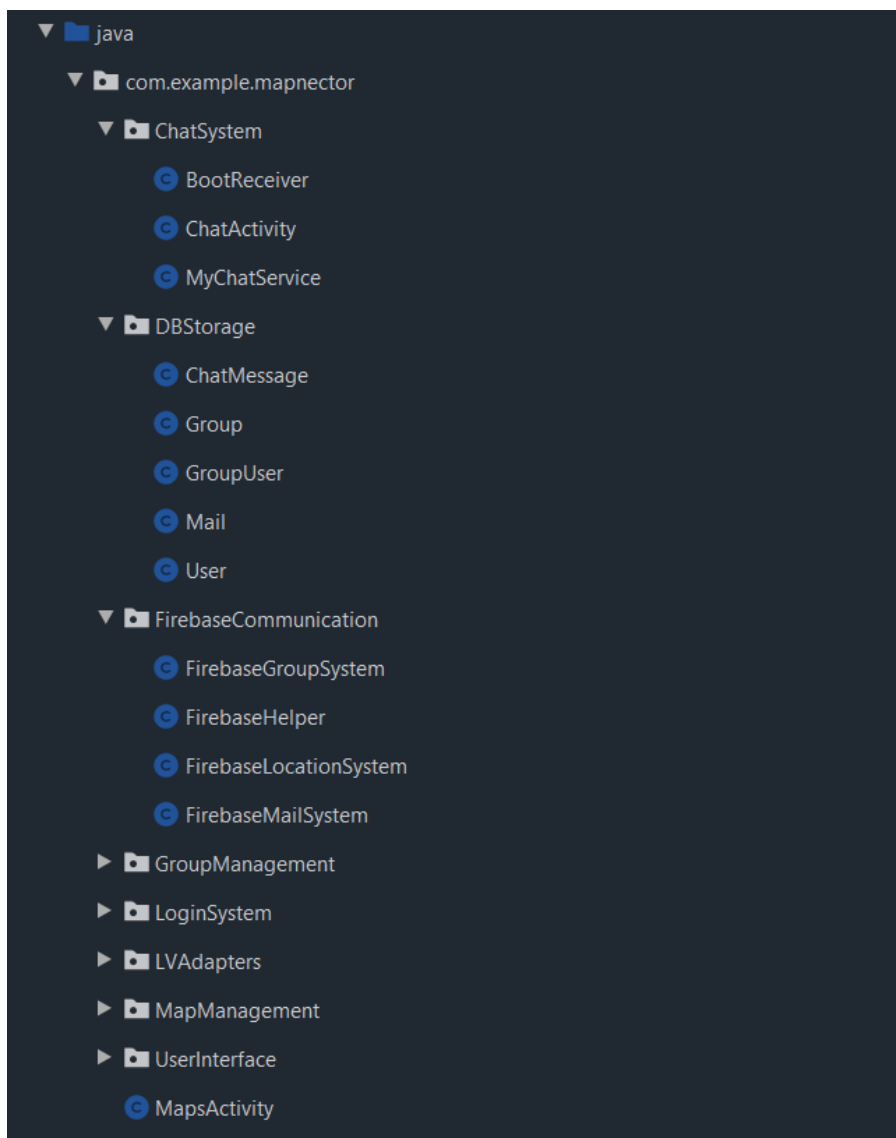
- \*מאפשרת לאנשים ליצור משתמש ולהכנס לקיים, תוך ביצוע email verification.
- \*מאפשרת ליצור קבוצות משתמשים ולנהל אותן, כגון את מספר המשתמשים בקבוצה, להעיף או להכניס אנשים, לקבוע כי הם יהיו סגורות(כלומר אין כניסה לקבוצה) או פתוחות(כל אחד יכול להכנס) ועוד.
- \*מאפשרת דיבור בין המשתמשים באמצעות צאט והאזנה לצאט מחוץ לגבולות האפליקצייה באמצעות service.
- \*קיום תיבת דואר נותן אפשרות לתקשר בין אנשים אחד על אחד ולא רק על ידי הצ'אט ולקבל הודעות מערכת(הודעות שאין צורך במשתמש כדי לשלוח, האפליקצייה עושה לבדה), כגון על כך שמישהו מבקש להכנס לקבוצה ואני בוחר אם לאפשר לו או לא.
- \*היכולת לראות מיקום של משתמשים אחרים ולהראות לחברי הקבוצה את שלהם וכך להקל על הפגשות עם חברים, לאפשר להורים לוודא כי ילדיהם במקום שהם מצפים לו ועוד.

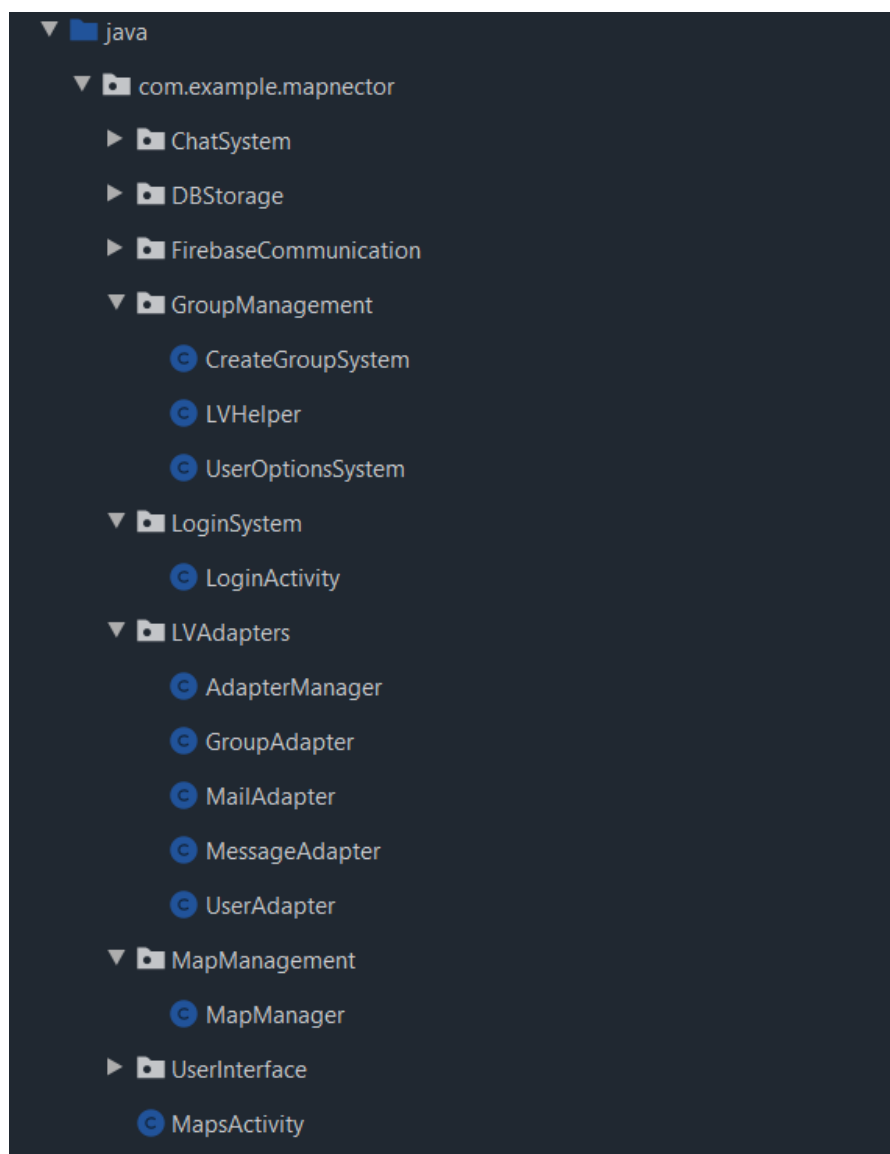
### מסכים

- באפליקצייה יש 3 מסכים בלבד - LoginActivity, MapsActivity, ChatActivity.
- LoginActivity - הקלאס והמסך היחיד האחראי על חלק הכניסה של המשתמש למערכת firebase. יכול לעבור לMapsActivity.
- MapsActivity - המסך העיקרי. מצדו השמאלי יש רשימה של קבוצות/משתמשים בקבוצה שעליה המשתמש מסתכל עתה, מצד שמאל יש חלונית שאפשר לפתוח ולראות את תיבת הדואר. באחת מגרסאות הפרוייקט היו לי 5 מסכים, אבל לבסוף מצאתי ליותר נוח לאפשר למשתמש לראות מקומים, תיבת דואר ואת הקבוצות במסך אחד - אז היה צריך למצוא פתרון שהמסך יהיה קומפקטי. יכול לעבור לLoginActivity או לChatActivity.
- ChatActivity - המסך שאליו אנו מובלים במידה ורוצים להכנס לצאט של קבוצה מסויימת, מאפשר להתכתב עם משתמשי קבוצה אחרים. על מנת לחזור ממנו ל MapsActivity משתמשים ב ArrowBack.

## כל הקלאסים

נציג עתה איך נראה הפרוייקט באנדרואיד סטודיו: על מנת להראות את כל הארכיטקטורה היה צריך לקחת שתי צילומים:





## הרשאות

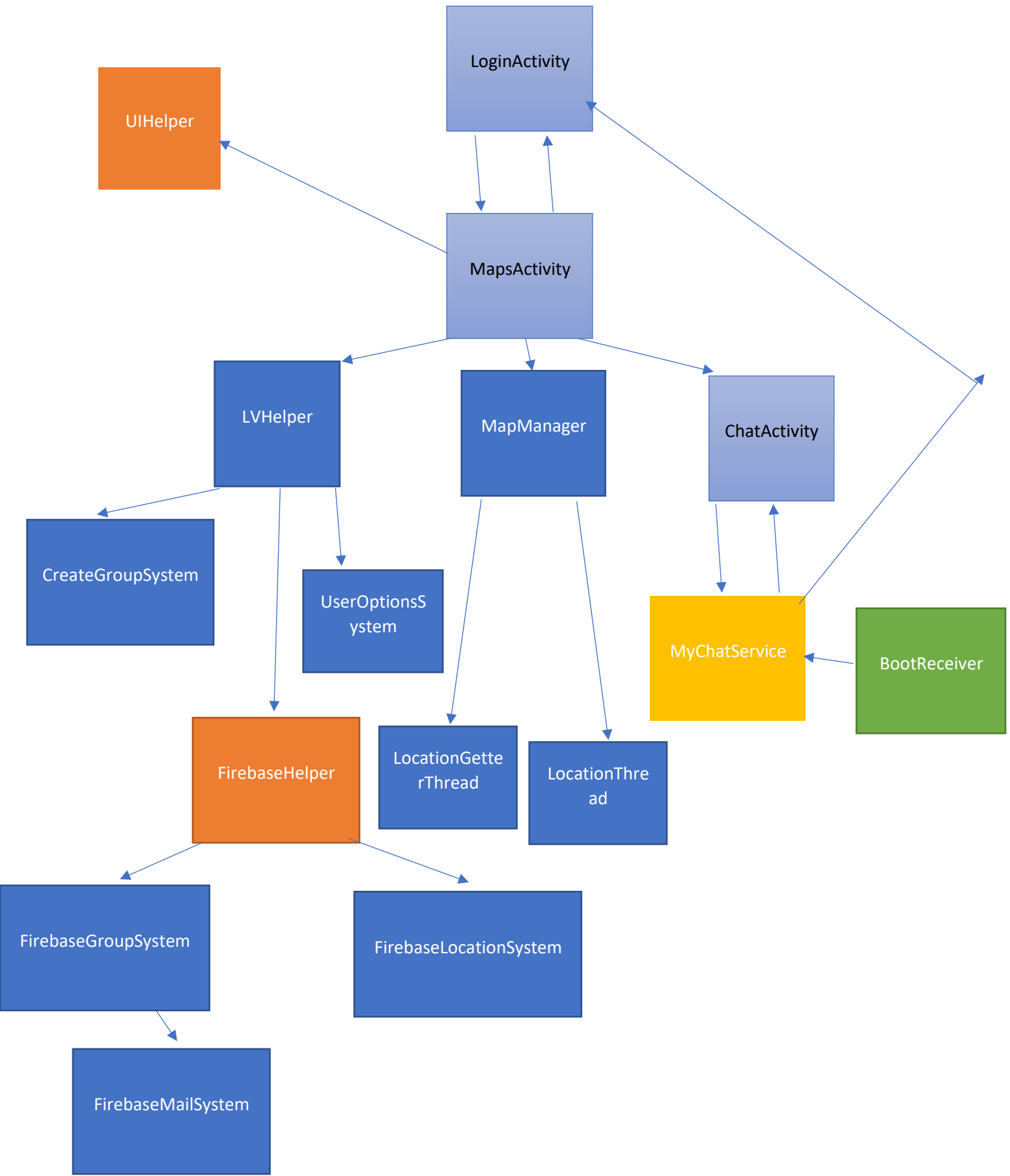
נדבר על הרשאות שנחשבות לdangerous. לפעולת האפליקציה הבסיסית אין חובה בשום הרשאה מסוג זה. במידה ומשתמש ירצה לשתף את מיקומו כאשר בפעם הראשונה ילחץ על checkbox תוצג בפניו בקשה להרשאה. כלומר הרשאות מחייבות אין, הרשאה אופציונאלית היא הרשאת שליחת מיקום - ACCESS\_FINE\_LOCATION.

הרשאות שאינן dangerous-RECEIVE\_BOOT\_COMPLETED,VIBRATE,FOREGROUND\_SERVICE

תרשים מסכים/מחלקות/UMI:

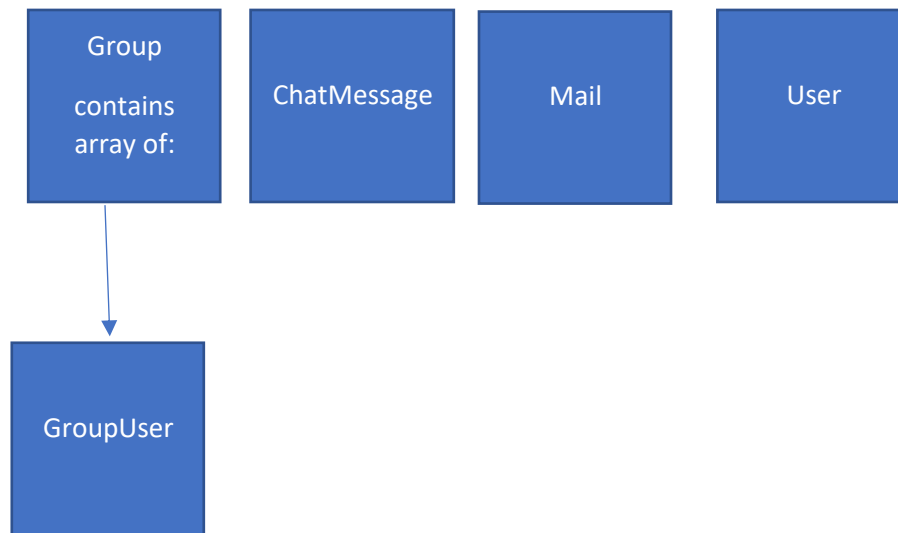
בכתום - קלאסים שהם singleton לוגית, כלומר נוצרים פעם אחת ומועברים לאובייקטים רבים אחרים כפרמטר.

תכלת - Activity. ירוק - BroadcastReceiver. צהוב-service. כחול-קלאס רגיל

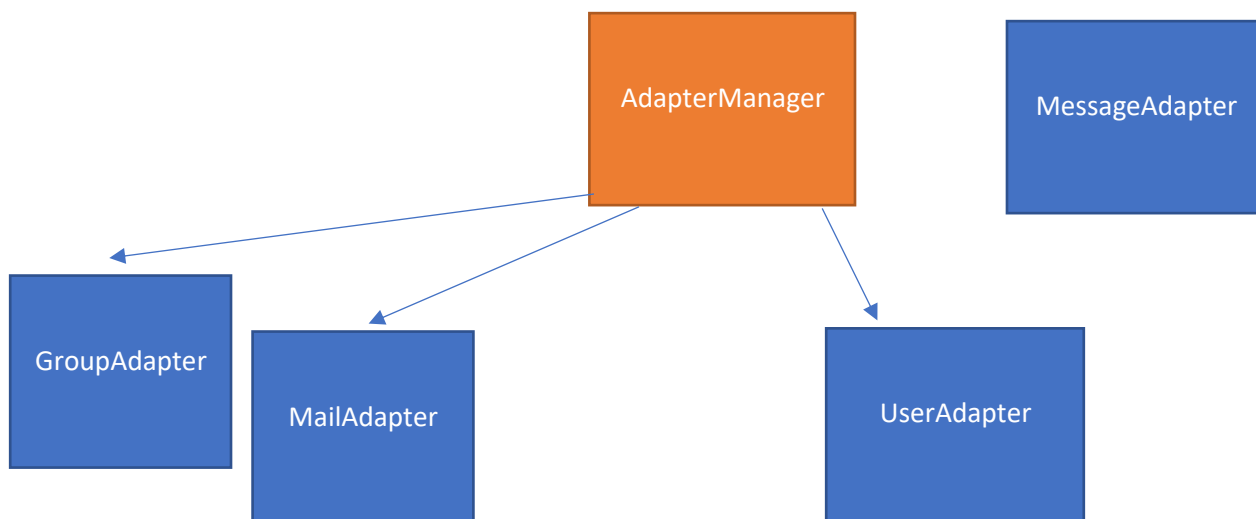




אובייקטים שנשמרים ב-Firebase:



Adapters Hierarchy:



ננסה לפשט את התרשים. הקלאס הראשון הוא גם אקטיביטי והוא לוגין סיסטם. הוא יכול לבקש להזין ססמה או פשוט להכניס את המשתמש תלוי בהאם הוא נכנס לאפליקציה ממכשיר זה מקודם. ניתן גם לחזור למסך זה מה-MapsActivity.

MapsActivitiy זהו המסך הראשי והוא שומר מצביעים לשלושה אובייקטים מרכזיים - LVHelper שבעיקר דרכו מתבצעת פנייה לשיטות שאחראיות על ה ListView של ה groups ושל המיילים (לפתוח חלונית ימנית). תפקידו העיקרי של קלאס זה ליצור ממשק המפריד בין פעולות המבצעות משהו הקשור למידע של הליסט ויאו לבין פעולות אחרות.

MapManager כשמו לוקח על עצמו את כל תפקיד הלקבל ולשתף את מיקומו הנוכחי של הסמארטפון.

ChatActivity נקרא כשרוצים להכנס לצאט של קבוצה, משתמש יכול להאזין לצאט וכך יופעל סרביס (foreground service) שבמידה וילחצו על ה notification יעביר אותנו חזרה למסך הצאט אקטיביטי. כמו כן ישנו broadcast שכאשר הטלפון נדלק מפעיל סרביס שמציג הודעה ובמידה והnotification יולחץ יפתח את האפליקציה.

LVHelper יש הפניה ל CreateGroupSystem שלוקח על עצמו את פעולת יצירת קבוצה חדשה (דיאלוג, אימות נתונים ועוד), בהמשך אולי אוסיף אפשרות לעדכן נתונים של קבוצה ואם כן זה כנראה יהיה בקלאס זה.

כמו כן יש הפנייה ל UserOptionsSystem שתפקידו להאזין ולבצע פעולות שקשורות לmailSystem, כמו להקשיב אם יש לחיצה ארוכה לשאול האם להדיח יוסר או לא (במידה ואתה הלידר של הקבוצה). במילים פשוטות הקלאס מהווה אוסף של דיאלוגים listeners אליהם שכל אחד יופעל במידה והevent מתבצע ולאחר מכן מעביר מידע על כך לקלאסים הרלוונטיים.

עכשיו LVHelper יוצר FirebaseHelper שדרכו מתבצעות כל הפניות ל Firebase של האפליקציה למעט פעולות שקשורות ללוגין ולצאט - ששם הקשר ל Firebase מתבצע בתוכו (כי אם האפליקציה נסגרה ורוצים שעל ידי לחיצה על notification יגיע הצאט לא נוכל לסמוך על קיומו של FirebaseHelper כי האפליקציה סגורה).

הקלאס עצמו מחזיק כמה קבועים ולא יותר מזה, העבודה עצמה מתבצעת בילדים שלו - FirebaseGroupSystem

בתרשים FirebaseHelper בצבע כתום כי אמנם הוא נוצר על ידי LVHelper אבל קלאסים רבים אחרים יכולים לשמור הפניה ל FirebaseHelper המקורי. זה מה שמסמל הצבע הכתום. מטעמי נוחיות וכדי לאפשר לגשת למספר Views מהרבה מקומות ישנו קלאס UIHelper שנוצר ב MapsActivity ומחזיק את רוב הפקדים של האפליקציה. נעשה לנוחיות וכדי שבשאר הקלאסים יהיה יותר לוגיקה ופחות דברים המתעסקים בגרפיקה. ישנו גם את AdapterManagern שמחזיק מצביעים לכל האדפטרס של האפליקציה למאט זה של הצאט (ושבו, כי נדרש שיעבוד בלי להסתמך על יצירת AdapterManger אם מפעילים לאחר סגירת אפליקציה).

טוב זה היה מה שנקרא הקוד על רגל אחת, הסברים יותר מפורטים יהיו בהמשך וממש מפורטים ב javadoc.

## אלגוריתמים עיקריים

הקוד עצמו יצא סביבות ה 3400 שורות. דבר זה קרה כי היו הרבה מאוד באגים ומקרי קצה שהיה צורך לבדוק. נעבור על האלגוריתמים העיקריים. לדעתי לא היו באפליקציה אלגוריתמים מסובכים - הרגשתי שבאפליקציה רשמתי לא אלגוריתם אחד מסובך אלה הרבה מאוד חלקי קוד קטנים והקושי האמיתי היה לגרום לכל החלקים האלה לעבוד יחד וכך ליצור את האפליקציה. ב FireBaseSystem כל האלגוריתמים יצאו גדולים בגלל הצורך בlisteners. לדעתי החלקים הכי חשובים נמצאים ב FireBaseSystem כי הם אלו המנהלים את לוגיקת הקשר בין האפליקציה לfirebase. אקח כמה חלקים שבזמנו הסתבכתי איתם ואסביר את הדילמות ואיך פתרתי אותן.

אוקיי בעת כתיבת קטע הקוד הזה רציתי שני דברים- שאני(המשתמש) אוכל לקבל מידע על מיקום של משתמשים אחרים שרוצים בזאת, ולשלוח את שלי רק אם אני רוצה בכך. מכיוון שמיקום של אדם יכול להשתנות כל מילישנייה לא נוכל לקרוא לValueEventListener כי כל פעם שזה קורא זה יעמיס את הרשת ואת המעבד של הטלפון - במקום זאת נגדיר Thread שכל זמן מסויים(5 שניות) יקח מידע על מיקומים של אחרים מהפירבייס. אוקיי אבל הבעיה היא שנרצה לעדכן את google map והיא נחשבת לממשק גרפי, ולא ניתן לעדכן אותה מthread שהוא לא הmain thread. לכן השתמשנו בhandler וכל פעם שהגיע הודעה עדכנו כך את googlemap. אוקיי זה היה פשוט יחסית עכשיו נעבור לקטע שבו אנחנו שולחים לפירבייס מידע על המיקום שלנו. קודם כל, בשביל זה צריך הרשאה. נניח וקיבלנו הרשאה ואנחנו שולחים כל 5 שניות בthread אחר מידע לפירבייס(אחר כדי להפריד בין הלוגיקה וכי אין צורך כאן בhandler), נניח ועודכן המיקום שלנו. לצורך בטיחות הגדרתי כי כדי לשלוח מידע על מיקום לחברי קבוצה מסויימת צריך שתי דברים - צריך שהמיקום של המשתמש יתעדכן בfirebase(יהיה שונה מ300) ושמשנתה המתאר האם המשתמש מאפשר לשותף מידע לקבוצה יהיה true. ננים והמשתמש שולח מידע על עצמו לקבוצה מסויימת, אם נסגור את האפליקציה המיקום שלנו ישאר בfirebase, וזה לא בטיחותי(יכולים לראות את המיקום האחרון שלנו שנשמר בfirebase). אנחנו רוצים שכאשר המשתמש יצא מהאפליקציה לעשות reset למיקום שלנו בfirebase, ומכיוון שקבעתי כי 300 בlongitude,latitude זה כמו null(כי הטווחים נעים בין 180-180 ו-90-90) אז נרצה לשים ערך 300 במיקום של המשתמש. מכיוון שלא נוכל לסמוך על onDestroy כי לא תמיד מתבצע(לפעמים הprocess נשמד לפני הקריאה) את החלק הזה ממשנו בonPause. אוקיי זה דבר אחד. נניח והמשתמש לחץ על הצק בוקס שאומרת שהוא רוצה לשלוח מידע על מיקומו לפירבייס, אנחנו רוצים שרק קבוצות שהוא הרשה להן לגשת למיקומו יעשו זאת. עכשיו נניח ואנחנו שלחנו מידע לטלפון שלא היה אמור לקבל על מיקומו של מישהו ופשוט לא הצגנו לו את זה ויזואלית(נניח עשינו משפט תנאי אם אמת תציג מידע אחרת לא) - זה עדיין לא בטיחותי. מה שעשיתי זה לפני קבלת המיקומים בדקנו אם היוסר מאפשר לנו לגשת למידע זה, ורק אם כן נקבל אותו - כלומר לא יהיה מצב של הסנפת פקטות עם מידע כי מידע רגיש שלא אמור להשלח לא ישלח מלכתחילה.

חלק בטיחותי שלישי זה firebase rules, שאתעסק בו מאוחר יותר אם יהיה זמן.

אז הסיטואציה היא שכל חמש שניות מתבצע הקוד בthread, אם נכנסנו לקבוצה מסויימת נרצה לקבל עליה מידע ויופעל thread שמקבל מידע, ואם נרצה לשלוח מידע thread שמקבל מידע יופעל ויפסק ויעשה reset למידע בonPause.

באג שלא הצלחתי לתקן זה אם עושים restart לטלפון לא נקרא onPause והמיקום נשמר...

כמו שציינתי בהתחלה אין פה הרבה אלגוריתמים מסובכים לחשוב עליהם, אלה יותר לתכנן את כל המערכת נכון כך שתעבוד נכון.

## (2) זיהוי מיילים בfirebase.

הבעיה שהייתה כאן היא כיצד להבדיל בין מיילים של אותו משתמש. נניח המשתמש עשה LongClick על מייל מה שאומר שהוא רוצה למחוק אותו - נניח יש לנו מייל עם אותה כותרת, אותו שולח ומקבל ואותו גוף. צריך פה עוד גורם שיאפשר לנו להבדיל בfirebase איזה מייל בדיוק נרצה למחוק, לא נוכל להסתמך על שדות המייל(שכן בחלק מהמיילים אפשרתי שדות ריקים ולפעמים הכמות משתנה לפי סוג המייל). אפשר היה להשתמש בhashCode אולי, אבל הדרך שהחלטתי להשתמש בה היא ב

Date.getTime. שיטה זו מחזירה את כמות המילישניות שעברו מאז 1900 אם אני לא טועה, וביצירת אובייקט מסוג Mail חדש שדה זה מתעדכן. אוקיי כך הצלחתי לזהות בין מיילים ברוב המקרים(לדוגמה מיילים בעלי אותו תוכן אבל זמן יצירה שונה). עכשיו נניח ששני משתמשים שונים יצרו מייל באותו זמן(כלומר הערך במילישניות

זהה) ושלחו אליי. מכיוון שהשדה נקבע בעת יצירת מייל אין לי דרך להבדיל בין המיילים(נניח קיבלתי מידע שצריך למחוק מייל בעל שדה x, אבל יש שני מיילים בעלי שדה x). אז כאן ראיתי כי יש צורך לכלול גם את uid של השולח. אז השדה mailIdentificator שווה לשרשור של uid עם מספר המלישניות בזה עתה. אז אם רוצים לבדוק אם mail1.equals(second mail from firebase) שווה בין שתי השדות האלה. תיאורטית, אם אותו משתמש יכנס משתי מכשירים שונים, ויצור איכשהוא מייל וישלח באותה מלישניה, עדיין יכול להיווצר באג אבל הסיכוי לכך נמוך מאוד, וגם במידה וזה יקרה אז פשוט המערכת תמחק בפעם אחת 2 מיילים במקום מייל אחד כמו שהיה רצוי, אז זה סביל.

(3 דיבאג לBootReceiver. מה שהקלאס עושה זה עם סיום הדלקת המכשיר מפעיל service שמציג הודעה

have a nice day

הבעיה היא שזמן רב עניין זה לא עבד ולא הבנתי למה, ולא רק זה אלה בגלל שצריך לעשות ריסטרט לטלפון/אמולטור הדיבאגר חושב כי האפליקציה נסגרה ומפסיק את עבודתו, כך שאין לי שום סיכוי לגלות איפה הטעות. מה שעשיתי זה החלפתי את ה bootReceiver במאזין לשינוי מצב טיסה, שזה עניין שכן יכולתי לעשות לו דיבאג וכך מצאתי את השגיאה וחזרתי לbootReceiver.

(4 אלגוריתם DB\_InitCurrentUser - אלגוריתם אולי לא הכי מסובך אבל חשוב מאוד. המשתמש נחסם מהאפליקציה על ידי דיאלוג עד לסיום ביצועו. מטרתו לקבל מידע על המשתמש שנכנס, את הקבוצות שהוא שייך אליהם ולאתחל אותן. הוא מאתחל גם את המערכת mailSystem.

```
private void DB_InitCurrentUser() {
    FirebaseHelper.refUsers.orderByChild("uid").equalTo(FirebaseHelper mAuth.getCurrentUser().getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for(DataSnapshot sp : snapshot.getChildren()) {
                FirebaseHelper.current = sp.getValue(User.class);
                activity.setTitle("Welcome " + FirebaseHelper.current.name + "!");
                if(FirebaseHelper.current.groupsPendingKick == null)
                    FirebaseHelper.current.groupsPendingKick = new HashMap<String, String>();
                if(FirebaseHelper.current.myGroups == null) { //if im not in groups yet and have no arraylist for them create one
                    FirebaseHelper.current.myGroups = new ArrayList<String>();
                }
                refMineGroups = FirebaseHelper.refUsers.child(FirebaseHelper.current.key).child("myGroups");
                refMineGroupsPendingKick = FirebaseHelper.refUsers.child(FirebaseHelper.current.key).child("groupsPendingKick");
                refMineGroupsPendingJoin = FirebaseHelper.refUsers.child(FirebaseHelper.current.key).child("groupsPendingJoin");

                for(String groupKey : FirebaseHelper.current.myGroups) {
                    FirebaseHelper.refGroups.child(groupKey).addValueEventListener(groupListener);
                }

                refMineGroupsPendingKick.addValueEventListener(groupsPendingKickListener);
                refMineGroupsPendingJoin.addValueEventListener(groupsPendingJoinListener);
            }
            if(FirebaseHelper.current.myMails == null)
                FirebaseHelper.current.myMails = new ArrayList<Mail>();
            ADmanager.initMailAdapter();
            fms = new FirebaseMailSystem(ADmanager, uiHelper, activity, edtTitleSendMailRef);
            uiHelper.lvMails.setAdapter(ADmanager.getMailAdapter());
        }
    });

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

כל שאר האלגוריתמים לא יכול להגיד כי הם אלגוריתמים "עיקריים" ומבוצעים בהתאם לצורך, אך אלגוריתם זה הוא must to have כי אחרת לא נוכל אפילו להכנס לאפליקציה. הסבר יותר מפורט עליו בהמשך או ב-javadoc.

## פירוט מחלקות:

### BootReceiver

יורש מBroadcastReceiver ונקרא עם סיום הדלקת הטלפון - תפקידו להפעיל את MyChatService.

### ChatActivity

מהווה אקטיביטי להעברת הודעות בין משתמשים(הצ'אט של הקבוצה הנוכחית)

### MyChatService(foreground service)

מופעל כאשר עושים לו listen או עם הדלקת הטלפון. לאחר הדלקת טלפון מציג הודעה ועם לחיצה על notification מעביר ל loginSystem, אם נוצר בגלל הקשבה לצאט יראה תמיד את ההודעה האחרונה שנשלחה בקבוצה.

### ChatMessage, Group, GroupUser, Mail, User

אלה הם קלאסים שנוצרו לשמירת מידע בהתאם בפיירבייס.

### FirebaseHelper

מחזיק קבועים והפניות לאובייקטים שמטפלים בבקשות הקשורות לfirebase. מחזיק כמו כן את current שהוא אובייקט המשתמש הנוכחי.

### FirebaseGroupSystem

מתקשר עם firebase ומבצע דברים הקשורים לקבוצות(הכנסה לקבוצה, יציאה ממנה, יצירת קבוצה חדשה, עדכון מספר האנשים בה ועוד). מפנה לFirebaseMailSystem שמתעסק בדבר יותר ספציפי בחלק הקבוצות.

### FirebaseLocationSystem

מטפל בבקשות לקבלת מידע על מיקומם של משתמשים בקבוצה מסויימת, ושולח את המיקום של היוסר הנוכחי לשמירה בfirebase.

DoubleDoubleName(inner class of FirebaseLocationSystem)

מאין struct ששומר מידע שנדרש לגבי משתמשים על מנת להציג marker שלהם על googlemap.

FirebaseMailSystem

אחראי על קבלת דואר נכנס והעברת דואר יוצא לנמענים. כמו כן בונה ומעביר הודעות שלא צריכות את עזרת המשתמש אוטומטית.

CreateGroupSystem

קלאס שכל תפקידו להוות ממשק ליצירת קבוצה (דיאלוג, views) ועוד.

UserOptionsSystem

תפקידו להאזין ולבצע פעולות הקשורות בעיקר לmailSystem, לדוגמה להחזיר מייל חוזר כתשובה לשולח, למחוק מייל, אך מכיל גם חלק קטן של דברים המטפלים ב groups שיש להם קשר כלשהוא למיילים (לדוגמה בלחיצה ארוכה על יוסר ב listview השמאלי יוצר דיאלוג שישאל אם רוצים להעיף אותו או לשלוח מייל).

LVHelper

קלאס שמכיל ברובו את הלוגיקה של הקשר בין המשתמשים לUI. לדוגמה ביצוע פעולות לאחר לחיצה על group, אנשים העוזבים קבוצה להפנות ל FirebaseGroupSystem ועוד. נועד להפריד בין שיטות שמתעסקות בgroups, mails וב ui מבין דברים לוגיים בלבד כגון locationSystem.

LoginActivity

מכיל את כל הלוגיקה והפקדים הנדרשים לביצוע באופן מלא של הלוגין (מכיל login, signup, verification שליחת verification למייל, בדיקה אם יוסר כבר קיים ועוד)

GroupAdapter, MailAdapter, MessageAdapter, UserAdapter

אין הרבה מה להגיד, אדפטרים שנוצרו על מנת להציג אובייקטים ב listView ויורשות מ ArrayAdapter. ב mailAdapter השינוי היחיד הוא שנקבע צבע ה item בהתאם לסוג המייל.

AdapterManager

קלאס נוצר ב listView ומועבר כפרמטר לכל אובייקט שצריך לתקשר עם אחד ה adapters למעט MailAdapter, מחזיק בכל ה adapters למעט המייל. (כי חשובה אי תלות של mailAdapter).

## mapManager

קלאס שתפקידו להתעסק בכל הלוגיקה שקורת בקשר למיקום המשתמשים, בקשת הרשאות ועוד. מכיל שני תת קלאסים המייצגים threads לקבלת מידע על מיקום ולשליחה כפי בתואר ב"אלגוריתמים עיקריים". פונה בהתאם לצורך ל-FirebaseLocationSystem.

LocationGetterThread(inner class of mapManager)

קלאס היורש מthread שמטרתו לקבל מיקומם של משתמשים אחרים המאפשרים זאת

LocationThread(inner class of mapManager)

קלאס היורש מThread ושולח מידע על מיקום המשתמש לשרת. במידה והמשתמש לא רוצה זאת או שאין הרשאה לא ייוצר אף פעם אובייקט מסוג זה

## MapsActivity

"כתר" האפליקציה. שורש העץ. זהו האקטיביטי עליו נלבשת המפה והListviews ומהווה אקטיביטי שעל פי התכנון היוסר יבלה בו רוב הזמן. ישנה אפשרות לבצע פעולות באמצעות buttons ולחזור לloginActivity או לעבור לצא'ט של קבוצה מסויימת. מחזיק listeners לכמה buttons אבל מה שברובו עושה זה מאחד את כל שלושת החלקים העיקריים בפרוייקט ועושה בהם שימוש בקלאס אחד.

## UIHelper

קלאס שכל מה שעושה זה מחזיק פקדים כדי להקטין את כמות ה"לא לוגיקה" בקלאסים אחרים. ישנם כמה פונקציות שנקראות מMapsActivity להקטנת העומס עליו שמבצעות דברים קבועים.

## מבני נתונים

hashMap - כדי לשמור דברים מסויימים בסגנון firebase key-value, כי אם משתמשים במערך יש צורך לקבל ולהציב את כל המערך בfirebase מה שמהווה בזבוז, אז במקומות שניתן השתמשתי בזה במקום מערך או list. דוגמה היא השתמשתי בhashMap בMapManager כדי לשמור התאמה בין key של יוסר לבין struct, מיני קלאס ששומרת את המיקום שלו. (קראתי לstruct בשם DoubleDoubleName).

עוד יתרון של מבנה נתונים זה שמאפשר ביצוע חיפוש בסיבוכיות  $O(1)$ , בניגוד למערך שבו  $O(n)$ , זהו גם דבר שהשתמשתי בו.

list/arrayList - כדי לשמור אובייקטים בתוך adapters, list מוחזר מfirebase.

## שימוש באבני יסוד

Activity - כל המסכים יורשים AppCompatActivity שיורש Activity, היה צורך להעביר לא Context אלה דווקא Activity כפרמטר במקרים מסויימים באפליקציה כדי לאפשר גישה לפקדים מקלאסים אחרים על ידי activity.findViewById.

Intent - העברה בין מסכים. מתבצעת מlogin לmapsActivity, מהchatActivity לservice, מהpendingIntent לLoginActivity ועוד.

BroadcastReceiver - השתמשתי בזה כי הייתי חייב מדרישות הפרויקט (לא היה אמור להיות באפליקציה שתכננתי). תכננתי לעשות שאם ישנו listener (כאילו myChatService) לקבוצה מסויימת והטלפון נכבה עם הדלקת הטלפון הסרביס יופעל שוב, הבעיה היא שהfirebase לא פועל בקלות כאשר האפליקציה לא דלוקה וצריך עוד הרשאות, לירוש לא service אלה FirebaseAuthServices וגם אז בתנאים מסויימים זה לא נותן גישה למידע ללא הפעלת האפליקציה. אז בחלק הזה נכשלתי והייתי צריך להסתפק בservice שמציג notification ידידותי שבעת לחיצה עליו מפעיל את האפליקציה. (שגם פעולה זו מאטה את הדלקת הטלפון, מה שדווקא יפגע בחווית הגלישה של המשתמש לדעתי, אבל עדיף על לעשות batteryReceiver).

Service - ניתן להאזין לצ'אט מסויים על ידי לחיצה על כפתור מתאים בmenu. בניגוד לווטסאפ ששולח notifications ללא הפסקה החלטתי לאפשר למשתמש להחליט אם הוא רוצה לקבל Notifications ואם כן לאיזו קבוצה כשהוא לא באפליקציה - לכן למרות שMyChatService הוא foregroundService שמחייב הצגת notification, המטרה של כל הקלאס היא התעסקות בNotification. (הצגת הודעה אחרונה שנשלחה בקבוצה שנבחרה).



## שמירת נתונים

### SharedPreferences

מתבצע שמירת מייל של אדם שעשה לוגין כך שאם ירצה לעשות logout יצטרך להזין רק את הססמה. ססמה כמובן שלא נשמרת מטעמי security. השאר בfirebase.

### Firebase

בfirebase שמרנו את כל ההיאררכיה של הנתונים שצריך. נדגים זאת. קודם כל למה firebase? כי האפליקציה שלנו בנויה על שיתוף מידע בין משתמשים בזמן את, בין אם עדכון קבוצות או שליחת מיילים/הודעות. לכן יש צורך בfirebase.

יש לנו שלושה טבלאות עיקריות - Users, Groups, Chats.



כך לדוגמה נראת טבלת Chats. בניגוד לkey רגיל Key של הצאט זהה לkey של group בטבלת Groups. לדוגמה MAUDW-... זהו גם המפתח של הקבוצה שאליה שייך Chat הנוכחי. בעקרון זהו מערך של אובייקטים מסוג ChatMessage שאת המבנה של אחד ניתן לראות באיור שלמעלה (וגם בקלאס ChatMessage).



כפי שניתן לראות טבלת groups קצת יותר מורכבת מזו של הצאט. יש בה את ה description, name שמוצגים בui למשתמשים, את id של המנהל, מפתח הקבוצה, כמות משתמשים, state שמתאר אם ניתן להכנס לקבוצה בחופשיות הוא Open אם Closed לא ניתן להכנס לקבוצה ואם Invite זה אומר שניתן לבקש מהמנהל לצרף לקבוצה(יופיע במקום כפתור Join כפתור Request שישלח מייל למנהל אוטומטי שיוכל לעשות accept או deny.

כמו כן יש מערך של groupUsers. groupUser מתאר את כל המידע שצריך לשמור על משתמש מסוים בתוך קבוצה. ניתן לראות את השם, האם הוא מנהל או סתם משתתף ואם מאפשר לקבוצה זו להאזין למיקומו.

נעבור לטבלה האחרונה:



זוהי הטבלה שנוצרת בעת signup. אנחנו שומרים את מיקומו של היוסר (300 אומר שמיקום לא ידוע, פשוט בגלל שזה מספר זה נוח יותר מלשים Null). זה שומר את המפתחות לקבוצות שהמשתמש משתמש בהן ובעת login דרכן מוצא את הקבוצות שאליו היוסר אמור להצטרף. כמו כן יש לו תיבת מיילים, ואת מבנה של Mail ניתן לראות באיור - לכל מייל יש event שמתאר את סוגו (יכול להיות custom, כלומר סתם מייל ממשתמש אחד לשני, יכול להיות reply ועוד). בדוגמה המייל אומר שבקשה להצטרפות לקבוצה אושרה והקבוצה הוספה לרשימת הקבוצות שלו. groupKeyToRequest הוא רק למיילים של בקשות להצטרפות לקבוצה, שהרי אם יש כמה קבוצות בעלי אותו שם נרצה לצרף לזו שהמשתמש ביקש ולא לאחרת. ה createTimeMillis מתאר את כמות הזמן במילישניות שעברה מ-1900 אם אני לא טועה, והיא נוצרה כדי להבדיל בין מיילים שונים של משתמש (כך שאם לשני מיילים בדיוק אותו תוכן נוכל להבדיל). \*הערה, createTimeMillis הוחלף ב mailIdentification מאז צילום הfirebase המקיים:

הסברתי בחלק "אלגוריתמים עיקריים".  
 $mailIdentification = uid + (current\ time\ passed\ from\ 1900\ in\ millis)$ , שזהו שרשור. למה עשיתי החלפה זו

Search by email address, phone number, or user UID					Add user	↺	⋮
Identifier	Providers	Created	Signed In	User UID ↑			
vipnet.channel@gmail.com	✉	May 22, 2021	Jun 7, 2021	pmG7sGHjN9WfVUyDqLZLXYJ2W...	📄		⋮
tim.pichkhadze@gmail.com	✉	May 21, 2021	Jun 6, 2021	xyrZK0pf0Edp8QJdai55vtNnKL12			
					Rows per page: 50	1 – 2 of 2	⏪ ⏩

ויש כמובן את כל עניין ההauthentication...

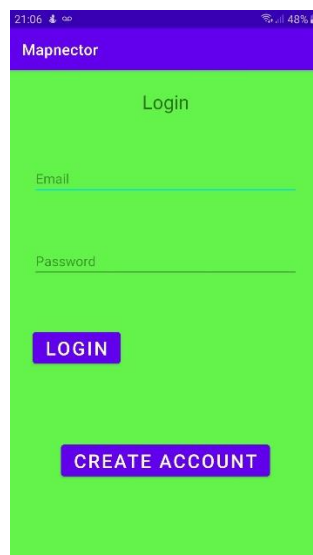
## מדריך למשתמש

### התקנה:

פשוט להתקין את קובץ הapk. אין צורך בהרשאות מיוחדות בעת התקנה. במידה והמשתמש יחליט שלאפשר גישה למיקומו הבקשה להרשאה תופיע אבל מהלך זה הינו אופציונאלי. ניתן להשתמש באפליקציה כportrait או כlandscapes לנוחיות המשתמש.

### מסכים:

#### LoginActivity



מטרתו לאפשר חיבור לאפליקצייה. לאחר ההתחברות הכניסה לאפליקציה תהיה אוטומטית אך ניתן לחזור למסך על ידי logout. אפשר או לעשות Login או ללחוץ על Create Account מה שיקפיץ דיאלוג למילוי, שלאחריו תצטרכו לעבור mail verification ולאמת שהמייל שהזנתם אכן שייך לכם. לאחר מכן תקבלו גישה לאפליקצייה.

21:25 45%

Mapnector

### Signup

Email

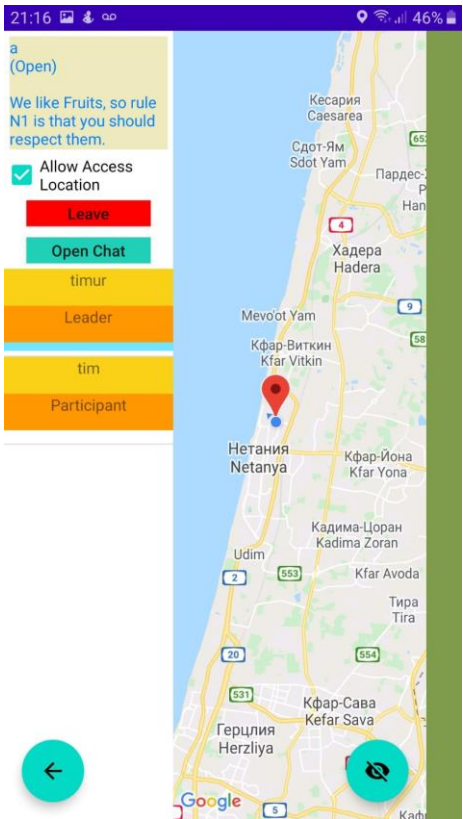
Password

Repeat Password

Username(Seen to others)

**SIGNUP**

MapsActivity



myTitle

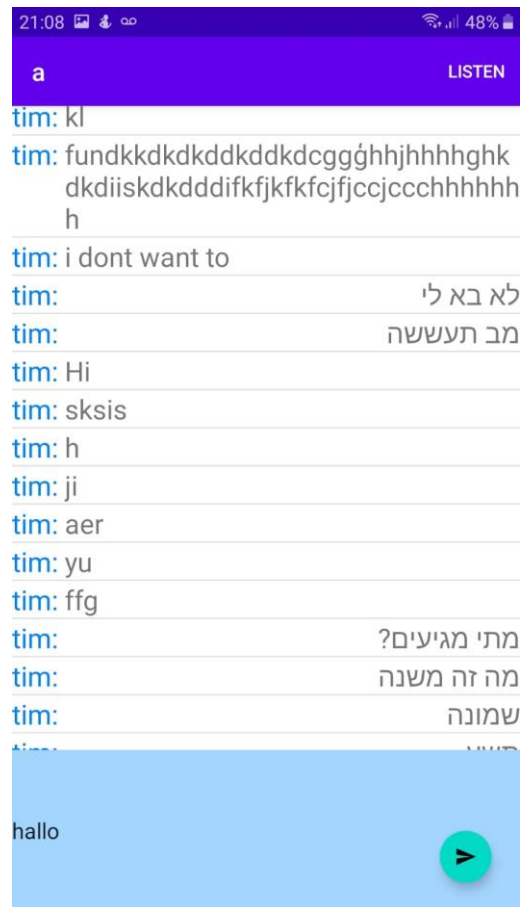
From:

\*הערה המפה אצלי ברוסית רק כי הטלפון שלי ברוסית. בטלפונים אחרים היא תהיה באנגלית/עברית/ בהתאם להגדרות טלפון פנימיות של משתמש.

זהו המסך הראשי. משמאל ניתן לראות את כל הקבוצות שאני שייך אליהן(מדבר על התמונה הראשונה, אם אתם המנהלים מתחת יהיה קו בצבע תכלת). ניתן לחפש קבוצות באמצעות searchBar שיביא את כל הקבוצות בעלי השם שחיפשתם, כדי לחזור שוב לראות את הקבוצות שאתם שייכים אליהם פשוט תנקו את search bar. אם אתם מעוניינים לשלוח מידע על מיקומכם אז תשימו V בצ'ק בוקס שמעל הקבוצות שלכם, וכדי לאשר לקבוצה לראות את מיקומכם צריך להכנס אליה ולסמן V במקום המתאים. אם תרצו להחביא את החלק השמאלי(זה שמציג את הקבוצות) תלחצו על כפתור העין וכך גם כדי להחזיר אותו. אם אתם לחצתם על קבוצה אתם תראו את כל המשתמשים שלה, ואם תלחצו על משתמש מסויים אם הוא מראה את מיקומו המסך יוביל אתכם למיקום המשתמש על המפה. תלחצו על כפתור ה+שלמטה כדי ליצור קבוצה חדשה, ותמלאו את הפרטים המתאימים ותלחצו על הכפתור ליצירתה. אם תיקחו את החלונית שמימין ותגררו אותה שמאלה תיפתח לכם תיבת הדואר שלכם, שבה תוכלו לראות את ההודעות שקיבלתם. לחיצה ארוכה על הודעה מוחקת אותה, ולחיצה קצרה על הודעות ספציפיות מאפשרת לבצע פעולה מסויימת, לדוגמה לחיצה קצרה על הודעה לבנה זו תשובה למייל שקיבלתם. כדי לחזור למפה פשוט תגררו את המסך מימין לשמאל. אם תיכנסו לקבוצה שאתם נמצאים בה תוכלו על ידי לחיצה ארוכה על משתמשים אחרים לשלוח להם מיילים או לבצע פעולות כגון להעניף מקבוצה/להפוך למנהל במידה ואתם מנהלי הקבוצה. תהיה לכם אפשרות ללחוץ על כפתור Open Chat, ואז תעברו למסך ההודעות שבו תוכלו לתקשר עם שאר חברי הקבוצה. כדי לעשות logout תלחצו על כפתור העין ועל הכפתור שהופיע לכם מצד שמאל.



## ChatActivity



למסך זה תגיעו במידה ותלחצו על כפתור Open Chat באחת הקבוצות שלכם. אתם יכולים לראות ולגלול את כל ההודעות שנשלחו בקבוצה. בשביל לשלוח הודעה יש להקליד אותה בחלק הכחול שלמטה וללחוץ על כפתור השלח שמצד ימין. במידה ותרצו לקבל התראות על הודעות מהאפליקציה גם כשהאפליקציה איננה פתוחה(או פשוט פתוחה ולא במסך זה) אז תלחצו על כפתור LISTEN שלמעלה, IGNORE בשביל להפסיק. תוכלו לראות את ההודעה האחרונה שנשלחה בקבוצה בהתראות שלכם ואם תלחצו עליה בהתראות תעברו למסך ההתכתבות.

## אילוצי תכנה ודרישות המערכת

צריכה להיות מותכנת גרסת אנדרואיד 19 API לפחות.(טכנית האפליקציה עובדת עבור 17 ומעלה צריך פשוט להכנס להגדרות ולשנות את minApi, אבל גם ככה מספר הטלפונים שיוכלו להריץ אותה עומד על כ-99.2%).

על מנת להשתמש באפשרות שליחת מיקום לאחרים יש צורך בGPS, אך התכנה תעבוד ללא GPS בכל שאר התחומים שאינם קשורים לשליחת מיקום. לכן אין דרישה בGPS, אבל יכול להגדיל את כמות האפשרויות של האפליקציה.

## מדריך למפתח

קודם כל חשוב לציין שיש כמה קלאסים שנוצרים פעם אחת בלבד, ומועברים כפרמטרים לאובייקטים רבים(המצביע אליהם). קלאסים אלה הם FirebaseAuth, FirebaseAuth, FirebaseAuth.

אעבור על הכל לפי קלאסים, אם יש קלאסים שלא משמעותיים לא אציין אותם.

כאן בגלל חוסר הנוחיות וכדי לא להגיע ל80 עמודים אעבור רק על הדברים המשמעותיים ביותר לדעתי בכל קלאס, הסבר מלא ומפורט ב javadoc שקישור להורדה אזורק גם פה וגם בנספחים:

<https://vipnet.itch.io/mapnector>

סומה: 135135642642

\*צריך להוריד את javadoc למחשב ולפתוח את index.html. נ.ב אפשר להוריד מהאתר גם את האפליקציה.

הקלאסים שאדלג עליהם - קלאסים מ DBStorage package -  
ChatMessage, Group, GroupUse, Mail, User, כי הסברתי את המשתנים שבהם בחלק שבו הצגתי את מבנה ה Firebase וכמעט אין בהם פונקציות, למעט override של equals.

כל מה שב Package LVAdapters, כי ה Adapters נוצרו כפי שהיה בחומר הנלמד כדי להיות מוכנסים בהמשך ל ListView, AdapterManager פשוט שומר את שאר ה Adapters (שוב, חוץ מ MessageAdapter).

UIHelper - כי כל מה שעושה זה מחזיק פקדים והפונקציות שבו הן פונקציות בעיקר של setVisibility.

createGroupSystem - כי כל מה שעושה בודק קלט ופונה ל Firebase ליצירת קבוצה.

MyChatService - כי זה פשוט service שמשנה notification כאשר יש עדכון מה firebase, הפונקציות שבו הם buildNotification ובנאי שנלמדו מ appSchool.

BootReceiver - כי זהו broadcastReceiver שמאזין להדלקת מסך ואז מפעיל service. מה שכן אציין שזה שהוא שם ב putExtra "-" יסמל בהמשך ל service כי הגיעו אליו מהדלקת מסך.

## MapsActivity

### משתנים:

LVHelper lvHelper	מצביע לקלאס מסוג LVHelper
UIHelper uiHelper	מצביע לקלאס מסוג UIHelper
MapManager mapManager	מצביע לקלאס מסוג MapManager


#### פונקציות:

void onClick	טיפול בארועי לחיצה של כפתורים במסך זה
void onItemClickListener	טיפול בלחיצה ארוכה על קבוצות משמאל
boolean onItemLongClick	טיפול בלחיצה על משתמשי קבוצה, לרוב לצורך ביצוע פעולה הקשורה למייל, ותועבר ל UserOptionsSystem

#### LVHelper

##### משתנים:

AdapterManager ADmanager	מצביע לקלאס שמחזיק את רוב האדפטרים - לגישה נוחה
UIHelper	מצביע לקלאס מסוג UIHelper
CreateGroupSystem cgs	קלאס האחראי על יצירת קבוצה חדשה
UserOptiondSystem system	מחזיק דיאלוגים וליסטנרים לביצוע פעולות של משתמש על משתמשים אחרים
FirebaseHelper FBhelper	קלאס שיתקשר עם הפירבייס
LV_STATE state	מתאר מה מראה נכון לעכשיו listview שמשמאל(קבוצות, משתמשים, קבוצות שחיפשו...)
LV_STATE prevstate	מתאר מה הראה פעם קודמת Listview שמשמאל

#### פונקציות:

void removeAllListeners	להוריד את listeners מfirebase
void onClick	האזנה לכפתור יצירת קבוצה חדשה שמצוי בcgs
void changeLVAdapter	פונקצייה נוחה לשינוי state,prevstate
void addGroupWindow	להראות דיאלוג ליצירת קבוצה חדשה
void search	לקרוא לFirebaseGroupSystem לחפש קבוצות עם שם מסויים

void leaveGroup	לקרוא ל-FirebaseGroupSystem לעזוב קבוצה נוכחית
void joinGroup	להצטרף לקבוצה נוכחית דרך firebase
void logout	לצאת מהמערכת(למחוק את ה token ששמור בdevice)

## UserOptionsSystem

משתנים:

firebaseHelper FBhelper	מצביע ל-firebaseHelper
AdapterManager ADmanager	מצביע ל-ADmanager
AlertDialog/Dialog all dialogs	dialogs לקשר עם משתמש. למה כל כך הרבה? כי רוצים להראות אפשרויות שונות עבור משתמשים שונים, לדוגמה ללידר תיהיה אפשרות להעיף מהקבוצה למשתמש רגיל לא מתאר את המשתמש שנבחר
GroupUser longClickSelected	מתאר אם ליצור מייל שהוא מוחזר למשתמש או לא(אם כן יש צורך לא לאפשר לשנות כותרת)
Boolean isReplyMail	

פונקציות:

void requestJoinGroup	לשלוח מייל בקשה הצטרפות לקבוצה
vois show....Dialog	להראות דיאלוגים מתאימים למינהם בהתאם למי לחץ על משתמש ואיזה משתמש לחץ(מנהל או משתמש רגיל של קבוצה)

## FirebaseHelper

משתנים:

ישנם משתנים שמתארים מסלול למקום מסויים בfirebase, הם קבועים. לא אציין אותם.

FirebaseGroupSystem fgs	מתקשר עם Firebase בכל הנוגע לבקשות הקשורות לקבוצות
FirebaseLocationSystem	מתקשר עם ה Firebase בכל הקשור לlocations

User current	המידע על המשתמש שנכנס בעת הlogin שמור במשתנה סטטי זה ב-FirebaseHelper

void logout	למחוק את הtoken של ה login לאפליקציה
-------------	--------------------------------------

## FirestoreGroupSystem

### משתנים:

ישנם משתנים שהם קבועים, ליסטת ודיאלוג להראות במידה ואין אינטרנט.

boolean canJoinLeaveGroup	נניח ולמשהו הלך האינטרנט. הזי הוא יכול ללחוץ מספר פעמים על כפתור להכנס או לצאת מקבוצה. משתנה זה גורם לכך שתשלח בקשה אחת בלבד עד לקבלת תשובה
boolean isLastInGroup	אם אני האחרון בקבוצה ויוצא ממנה, אני מצפה שהיא תימחק. אם אני לחצתי על יציאה מהקבוצה, הלך לי האינטרנט ומישהו אחר הצטרף אליה משתנה זה גורם לכך שאני לא אצא מהקבוצה(ואעביר את כל נתוני הצאט למשהו זר) אלא שאשר בה למרות שרציתי לצאת לפני מי שנכנס
FirestoreMailSystem fms	משתנה המתעסק בעניין המיילים ב-Firebase

### פונקציות

void DB_InitCurrentUser	להציב ערך במשתנה User. לקבל את הקבוצות שאני נמצא בהן. לבדוק את רשימות groupsPendingKick/Join למקרה ומשהו העיף
-------------------------	---

	אותי/הכניס לקבוצה בזמן שלא הייתי באפליקציה ולבצע פעולות בהתאם
void DB_kickUser	להעיף משתמש מקבוצה, אפשר להשתמש רק אם אני המנהל של הקבוצה הנוכחית
void DB_giveLeadership	להביא את הניהול למשהו אחר, אני הופך למשתמש רגיל
void DB_AddGroup	הוספת קבוצה חדשה שיצרתי כרגע לFirebase
DB_joinGroup	הצטרפות לקבוצה
DB_leaveGroup	יציאה מקבוצה

### FirestoreLocationSystem

משתנים:

ArrayList<Group> groupsAccessMyLocation	מחזיק את כל הקבוצות שיכולות לגשת למיקום שלי
---	---

פונקציות:

void DB_updateUsersLocations	לקבל מידע חדש על מיקום של משתמשים
void DB_groupMineLocationAccess	לשנות אם אפשר או אסור לגשת למיקום שלי לקבוצה מסויימת
void clearAllGroupsWithMineLocationAccess	להוריד את האפשרות לגשת למיקום שלי מכל הקבוצות
void DB_changeMineLocation	לעדכן את המיקום שלי בFirebase

### FirestoreMailSystem

משתנים:

ArrayList<Mail> mailsToRemove	מערך של מיילים למחיקה
Mail selectedRequestMail	המייל שנבחר על ידי בחירה(לחיצה ארוכה או קצרה)


#### פונקציות:

Mail createMail	יצירת מייל בעל תבנית אחידה(לדוגמה בקשה להצטרפות לקבוצה אין שם צורך בטקסט של משתמש, יעשה אוטומטית)
void DB_sendMail	לשלוח מייל דרך firebase

### MapManager

#### משתנים:

LocationThread lt	חוט לשליחת מידע על המיקום שלי
LocationGetterThread lgt	חוט לקבלת מיקום של אחרים
Boolean keepThreadCheckbox	אמת אם הצק בוקס בעל V false אחרת
Boolean keepThreadPause	גורם לכך שאם נקרא onPause ה thread יסתיים תוך לכל היותר חמש שניות(אלה אם כן המשתמש יחזור לאפליקציה)
Handler handler	משמש לעדכון ה UI כשאנחנו בתוך LocationGetterThread
HashMap<String,DoubleDouble> map	שומר התאמה בין משתמש למיקום שלו הידוע לנו

#### פונקציות:

void drawMapData	לצייר סימונים על המפה לפי הידוע ב hashMap
void activityResumed	נקרא מה mapsActivity אם activity חזר onPause

## ChatActivity

משתנים:

ArrayList<Message> messages	ההודעות שנשלחו בקבוצה זו
MessageAdapter messageAdapter	אדפטר להצגת ההודעות
String gKey,gName	מתארים את המפתח והשם של הקבוצה שלה שייך הצאט

void onClick	האזנה ללחיצה על כפתור שליחת הודעה חדשה
--------------	--

עוד פעם, הסבר מלא מפורט ובאנגלית מצוי ב javadoc, שלצערי לא מצאתי דרך לעשות לו host למרות שתכננתי וצריך להוריד אותו למחשב.



## רפלקציה

בהתחלה נהנתי לעבוד על הפרוייקט, כל רעיון שבא לראש כמעט מיידית הצלחתי לממש בקוד. לאחר מכן כאשר הוא גדל וגדל זה ניהיה הרבה יותר קשה למצוא את המיקום המתאים, ולשים כך שלא יהיו באגים. היה צריך לחשוב על כל הדברים שיכלו לשבש את האפליקציה ולפתור אותם אחד אחד. לקראת הסוף כבר פחות נהנתי לכתוב את הפרוייקט בגלל שבמקום לכתוב את הרעיונות שהיו לי בראש השקעתי המון זמן מחשבה על כל שורה ושאלתי את עצמי האם היא תגרום לבאג מסויים.

מה למדתי? כמו שכבר אמרתי לא רשמתי אלגוריתמים מסובכים במיוחד, אבל לדעתי לסדר את כל הקוד כך שהוא יעבוד זה היה אתגר גדול בשבילי, מכיוון שכמו שאמרתי במבוא לא לקחתי לעצמי מטרה צ'אט, מפות, קבוצות או משהו אחד - לקחתי על עצמי את כל הדברים האלה וניסיתי לממש אותם בפרוייקט אחד.

אז קיבלתי ניסיון מסויים בדיבוג - לדוגמה פעם ראשונה שבמקום סתם להשתמש בדיבאגר או לוג הייתי צריך להחליף קטע קוד בקטע פשוט יותר. קיבלתי ניסיון לכתוב את הקוד נכון(כי לפחות פעמיים הייתי צריך למחוק הכל ולהתחיל מפרוייקט חדש) ונהנתי בכתיבתו(לפחות בהתחלה). קיבלתי ידע מחוץ לתחום שלמדנו בבית הספר, למדתי להיעזר באינטרנט כדי להגיע לפתרון מה שמהווה כלי מאוד חשוב במיוחד בתכנות, ובכללי אם לפני הפרוייקט הייתה לי תחושה שאנדרואיד זה לא בשבילי ואני לא יודע שום דבר - עכשיו התחושה הזו עברה. אני מבין שהמטרה היא לא לזכור כמה שיותר שיטות, משתנים, פונקציות בעל פה, כי את כל הdocumentation ניתן למצוא בקלות באינטרנט. המטרה היא להבין עד הסוף מהו הדבר שאני משתמש בו וכיצד - ומתי להכניס אותו נכון בקוד - וזהו הדבר שהפרוייקט לימד אותי במשך שעות רבות.

האם אני אתכנת עוד משהו באנדרואיד? לא בזמן הקרוב יש לי קצת טראומה מהנוכחי, אבל אני מעוניין אפילו מחוץ לגבולות בית הספר לסיים את הפרוייקט הזה עד הסוף, שלא יהיו לו בכלל באגים ושיהיה איכותי ברמה שלא אתבייש לשים אותו בgoogle play.(לא בקטע של שיהיה מפורסם בקטע שעשיתי משהו "איכותי מספיק") אבל בינתיים, זה מה שקיבלתי מהפרוייקט.

ביבליוגרפיה

[/https://stackoverflow.com](https://stackoverflow.com)

[/https://developer.android.com](https://developer.android.com)

quora.com

<https://firebase.google.com/docs>

[/https://appschool.co.il](https://appschool.co.il)

נספחים:

הקישור:

<https://vipnet.itch.io/mapnector>

מכיל באתר גם Javadoc, גם את apk הפרוייקט וגם את כל הפרוייקט עצמו(כדי לראות את ה manifest והxml, כי יקח עוד הרבה מקום להכניס אותם בתיק פרוייקט) נכון לעכשיו(אולי אעדכן), וגם סרטון קצר שיכיר עם האפליקציה. האתר הוא אותו אתר עבור כל ההורדות אבל אפשר לבחור מה רוצים להוריד.

סומה: 135135642642(כדי שאנשים אחרים לא יוכלו לגשת למידע)