



תיכון הדרים הוד השרון



Blackspace

שם התלמיד: עידן הירש

ת.ז: 214290249

שם המורה: ניר סליקטר

תאריך הגשה: 30/05/2021

תוכן עניינים

5	מבוא
5	קיושר לקוד הפרויקט ב – GitHub
5	תיאור תוכנות הספר
5	הרקע לפרויקט
5	תהליכי המחבר
5	סקירת המצב הנוכחיים בשוק
5	חדשושים בפרויקט
6	למידה לפרויקט
7	ארגוני מרכזיים
7	הבעיה אליה התמודדת
7	מוטיבציה לעובודה
7	הצורך עליו הפרויקט עונה
8	ארQUITקטורה של הפרויקט
8	הציג שרטוט על של הפרויקט
10	ארQUITקטורת שיחות פרטיות בין שני משתמשים
12	תפקידה של כל יחידה
12	שרת - Server
12	לקוח - Client
13	General Use Case Diagram
14	הציג המקרים (Use cases) עבור הפונקציות העיקריות בפרויקט
14	הרשמה
14	התחברות
15	שליחת הודעה ביצ'אט הכללי
15	שליחת הודעה ביצ'אט פרטי למשתמש חדש
16	צפיה בכל השיחות הפרטיות של המשתמש, וכניסה לשיחה פרטית קודמת
16	מחיקת הודעה
16	התנתקות מהאפליקציה
17	הציג הפתרון המוצע והסיבות לבחירתו
17	טכנולוגיות בהן נעשה שימוש בפרויקט
17	Flutter
17	Firebase
18	מדריך למשתמש
18	הוראות התקינה
20	היררכיה מסכים
21	תרשימים מסכימים

21	מסמך טעינה.....
22	התחברות.....
24	הרשמה.....
26	מסמך צ'אט ראשי.....
28	מסמך כל השיחות הפרטיות.....
30	מסמך שיחה פרטית.....
31	Side drawer.....
32	מסמך מידע.....
33	בסיס הנתונים.....
33	הסבר כללי על בסיס הנתונים בו השתמשתי.....
33	Firebase.....
33	NoSQL.....
35	סכמה כללית של הישיות והקשרים בסיס הנתונים.....
37	תמונהות וסבירים על בסיס הנתונים.....
37	Authentication.....
37	Users collection.....
38	FireStorage.....
39	Chat collection.....
39	ChatRoom collection.....
41	מדריך למפתח.....
41	הסבר כללי על קוד הפרויקט.....
41	שימוש ב - accessories של הטלפון.....
41	תכנות מונחה עצמים.....
41	Remote procedure call.....
41	Asynchronous programming.....
42	קישור לקוד הפרויקט ב - GitHub.....
42	הסבר כללי על תיקית הקבצים של הפרויקט.....
43	הסבר על התיקייה lib.....
44	הסבר על התיקייה assets.....
45	הסבירים על קבצים, פונקציות וחלקה קוד חשובים.....
45	הגדרת Packages לשימוש בפרויקט.....
46	קובץ שימוש במצלמה של הטלפון.....
48	התקשרות עם השרת ובסיס הנתונים.....
50	רפלקציה.....
50	תחושתי מהעבודה על הפרויקט.....
50	כלים שקיבנתי ולקח איתי להמשך.....
50	אתגרים שעמדו בפניו במהלך הפרויקט.....

50	מסקנות מהפרויקט
50	מה הייתה עשו אחרת אילו הייתה מתחילה היום
51	מה היה יכול להפוך את העבודה ל'עליה' יותר
51	תכונות שהייתי רוצה להוסיף לפרויקט
52	ביבליוגרפיה
53	נספחים
53	קישור לספר פרויקט זה ב – GitHub
53	תעודה מ – Udemy על סיום קורס ב – Flutter
54	סרטונים של תהליך למידה
55	אפליקציות שונות שבניתי במלך הלמידה
55	אפליקציה – MAX – חלק פרויקט MAX
61	אפליקציה – קניות
63	אפליקציה – מתכונים
66	אפליקציה – מעקב על הוצאות כלכליות

מבוא

קישור ל קוד הפרויקט ב – GitHub

<https://github.com/itsidohirsh/blackspace>

תיאור תוכנת הספר

ספר הפרויקט שלפניכם יציג את פרויקט הסיום שלי – אפליקציית Chat בהשראת הסדרה הישראלית Blackspace.

ספר זה מציג בפירוט רב את הפרויקט שהכנתי, הרקע לפרויקט, מטרתו וקהל היעד שלו. כמו כן, הספר כולל תרשימים רבים הממחישים את אופן הפעולה של הפרויקט ומה מתרכש בו, מדריך למשתמש וצילומי מסך אשר יעזרו להמחשה. בנוסף קיימים בו גם פירוט על מבנה הנתונים והטכנולוגיות השונות בהם השתמשתי לפיתוח הפרויקט, מדריך למפתח ולבסוף רפלקציה אישית וביבליוגרפיה על הנעשה. וכמוון שגם נספחים... :)

הרקע לפרויקט

לפני כ5 חודשים עברו (דצמבר 2020), עלהה סדרה ישראלית חדשה בטלוויזיה שנקראה Blackspace ולאחר המלצות רבות מחברי החלטתי לצפות בסדרה. בסדרה מתואר אירוע רצח שקרה בבית ספר תיכון בארץ, בו התלמידים והמשטרה מנסים לפענה מי הרוצחים. במקום לדבר על הנעשה באפליקציות כמו Whatsapp או Facebook תלמידי השכבה דיברו ועדכנו אחד את השני דרך אפליקציה שנבנתה על ידי תלמיד באאותה השכבה. לאחר שצפיתי בסדרה לא יכולתי שלא לחשב כמה كيف זה יכול להיות לבנות אפליקציה כזאת שכל החברים שלך ישתמשו בה يوم יומם. ומשם קיבלתי את הרעיון לאפליקציה של Blackspace.

תהליכי המחבר

סקירת המצב הנוכחי בשוק

כאשר חשבתי על לבנות אפליקציית Chat חשבתי על חלופות אחרות אשר קיימות בשוק. דוגמאות לכך הם: גודלות כללו, בעלות משאים עצומים וצוותי פיתוח מוכשרים, אך הרעיון על לעשות משהו כזה ממשי סקרן אותי מאד.

חידושים בפרויקט

על מנת בכלל זאת להבדיל את הפרויקט שלי מחלופות המוצעות בשוק, ביסsty את האפליקציה שלי על רעיון קצת אחר.

הבסיס של האפליקציה שלי הוא שהמשתמש נכנס לחדר Chat גדול עם כל המשתמשים המוחברים, ומשם, אם הוא רוצה לדבר עם מישוהו Chat רק שלהם, הוא יכול ללחוץ על אותו משתמש, ולאחר הצגת מספר פרטיים עלייו, יכול להחיליט אם לשולח לו הודעה או לא.

לדעתי זהו חידוש מעניין מאוד, כי במקרה להתחיל לחפש אנשים על פי שם, או דברים אחרים, אפשר לשולח הודעה ישירות למי שאתה חושב שהוא נחמד לדבר איתו, לפי איך שהוא כותב Chat.

למידה לפרויקט

ברגע שהבנתי שאני רוצה למדוד להכין אפליקציה, התחלתי ללמידה על התחום. תחילת חיפשתי הרבה על פיתוח אפליקציות גם ל-iOS וגם ל-Android ולא הצלחה. לאחר שניות רבות של מחקר על התחום גיליתי את Codebase ו-Flutter. בקצרה היא פלטפורמת בנייה אפליקציות שפותחה על ידי גугл, שתחתה Flutter אחד, האפליקציות עובדות על סוגים רבים של מערכות הפעלה, כדוגמת Desktop, Web, iOS, Android, וכו'. וכך,_flutter היה הפתרון הכי טוב בשביב הפרויקט שלי, لكن התחלתי לעבוד.

לקחת קורס באתר Udemy בעלות של \$70, הכולל 41.5 שעות על Flutter, בו למדתי שפה חדשה לגמרי, Dart, מהבסיס של הבסיס עד תכונות מתקדם. במהלך הקורס בניתי מספר רב של אפליקציות בשביב התרגול, כך שהחומר התאורטי הפך למעשה ממשי מאוד מהר...

Flutter Cartificate מאתר Udemy שסייעתי בהצלחה את הקורס של



בסוף הקורס החלתי לעבוד על אפליקציית Chat פשוטה יחסית, בעלת Chat אחד ראשי לכל המשתמשים, וושיטות, בתוך פרויקט הסיום שלי של הקורס.

במהלך כל הזמן הזה הצטרכתי גם לפרויקט MAX, פרויקט לנOUR המשתתף באתגר סיבר ברחבי הארץ על מנת לנסות להוציא לפועל מוצרים של ממש. יחד עם צוות של סטודנט יזם בן 23, ומעצבת UX UI בת 26 מטעם MAX, עבדנו על אפליקציית נסיעות משותפות – Carpool שתעבדה בתחום מוסדות, כדוגמת מקומות העבודה, אוניברסיטאות וכו'.

אמנם, לקרהת סוף העבודה על הפרויקט עם MAX (בערך חדש לפני ההגשה הסופית) החלטתי, עקב בעיות של זמן והשראה מהסדרה שראיתי "Blackspace", לחזור לעבוד על הפרויקט של אפליקציית Chat כך שהוא יהיה הפרויקט הסופי שלו.

אתגרים מרכזים

הבעיה איתה התמודדתי

עד היום, כאשר עבדתי על פרויקטים שונים, בין אם בכיתה ובין אם בבית, היה לי איזשהו ידע מוקדם על הנושא. אולם, עקב כך שבפרויקט כל תלמיד יכול לבחור לכתת לאיזה כיוון ותחום שהוא רוצה, הפרויקט שבחרתי לעשות היה חדש למני, שף פעם לא התעסקתי בו, תחום האפליקציות. הייתה צריך ללמידה תורה שלמה מחדש, שפה חדשה, פלטפורמה חדשה, חוקים חדשים. ובגלל שהמורה שלי (ניר סליקטר) לא בקיא בתחום זהה, הוא גם לא היה יכול לעזור לי. את כל המידע שלי רכשתי ולמדתי בלבד, מה שלקח ממני הרבה שעות ומאז.

מוטיבציה לעבודה

את המוטיבציה לעשיית הפרויקט קיבלתי בעיקר מההשראה מהסדרה. המחשבה על כך שאני בסופו של דבר אצור אפליקציה עובדת, לחבריו ואני נוכל להשתמש בה כדי לדבר אחד עם השני, הייתה מעל הכל בעיני. בנוסף, שיעורי הסיבר עם החברים (למרות שרובם היו בזום השנה, קורונה...) נתנו לי מוטיבציה. לראות איך כל אחד מחברי יכולתי מתקדם בתחוםים שונים לחולטין היה מרתך בעיני, ונתן לי רצון לתת השראה לאחרים. כמו כן, חלוקת ההגשה להגשות קטנות כל פעם נתנה לי את רוגע המחשבה שאני יכול בכל הגשה לעשות את מה שהצבתי לעצמי. עד שבסוף יהיה לי את הפרויקט הסופי שלי. וגם ההצעה של מטרות עצמאיות עזרה למוטיבציה, לא רציתי לאכזב את עצמי.

הצורך עליו הפרויקט עונה

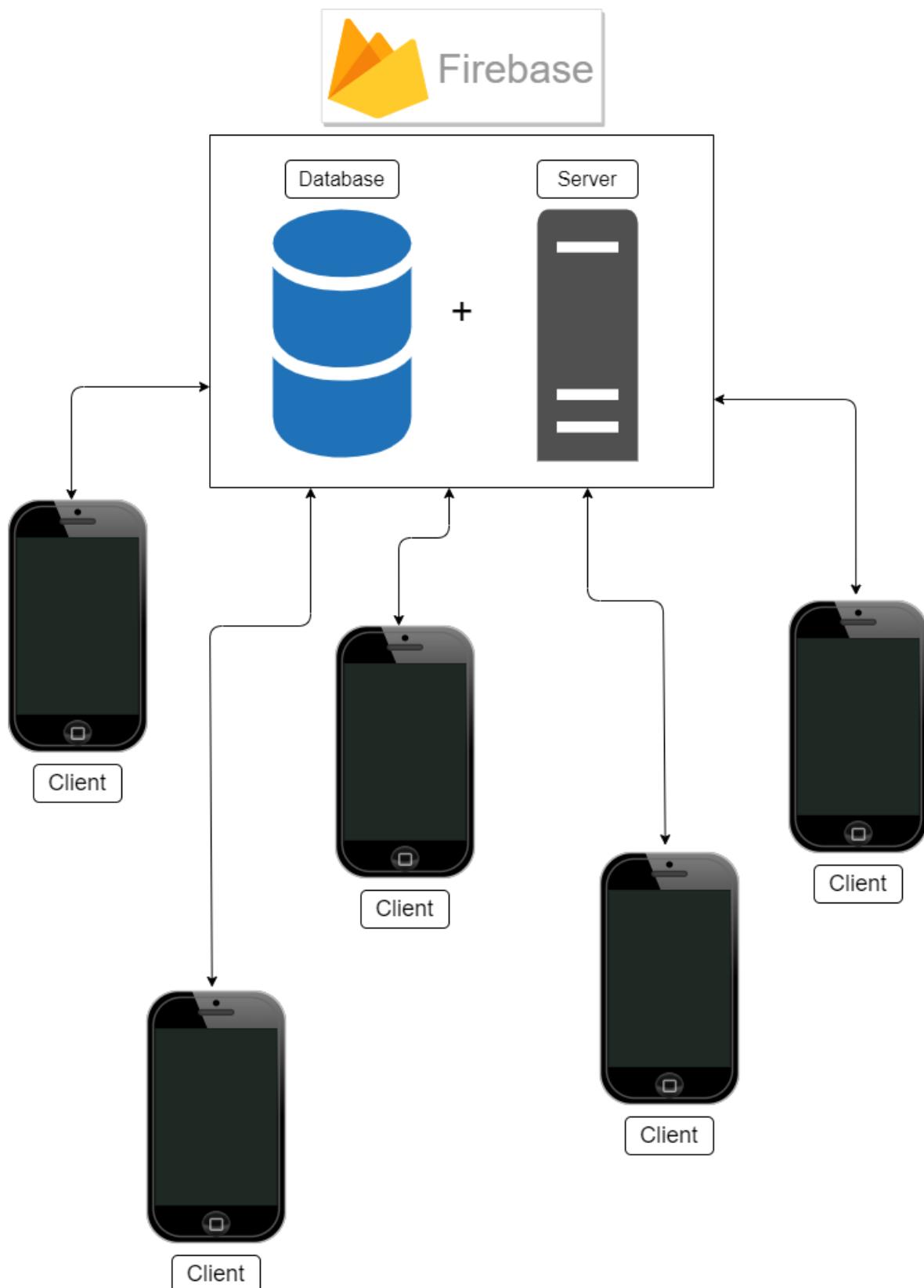
הפרויקט עונה על הצורך של תקשורת בין אנשים וחבריהם. בעזרת הפרויקט, אנשים שונים, שלאו דואק מכירים אחד את השני, יכולים לדבר יחד וליצור شيء מסוותף.

קהל היעד של הפרויקט היו רוב היגיאים, ילדים קטנים שרק קנו להם טלפון, עד אנשים מבוגרים שכבר יודעים דבר או שניים על טלפונים וטכנולוגיה...

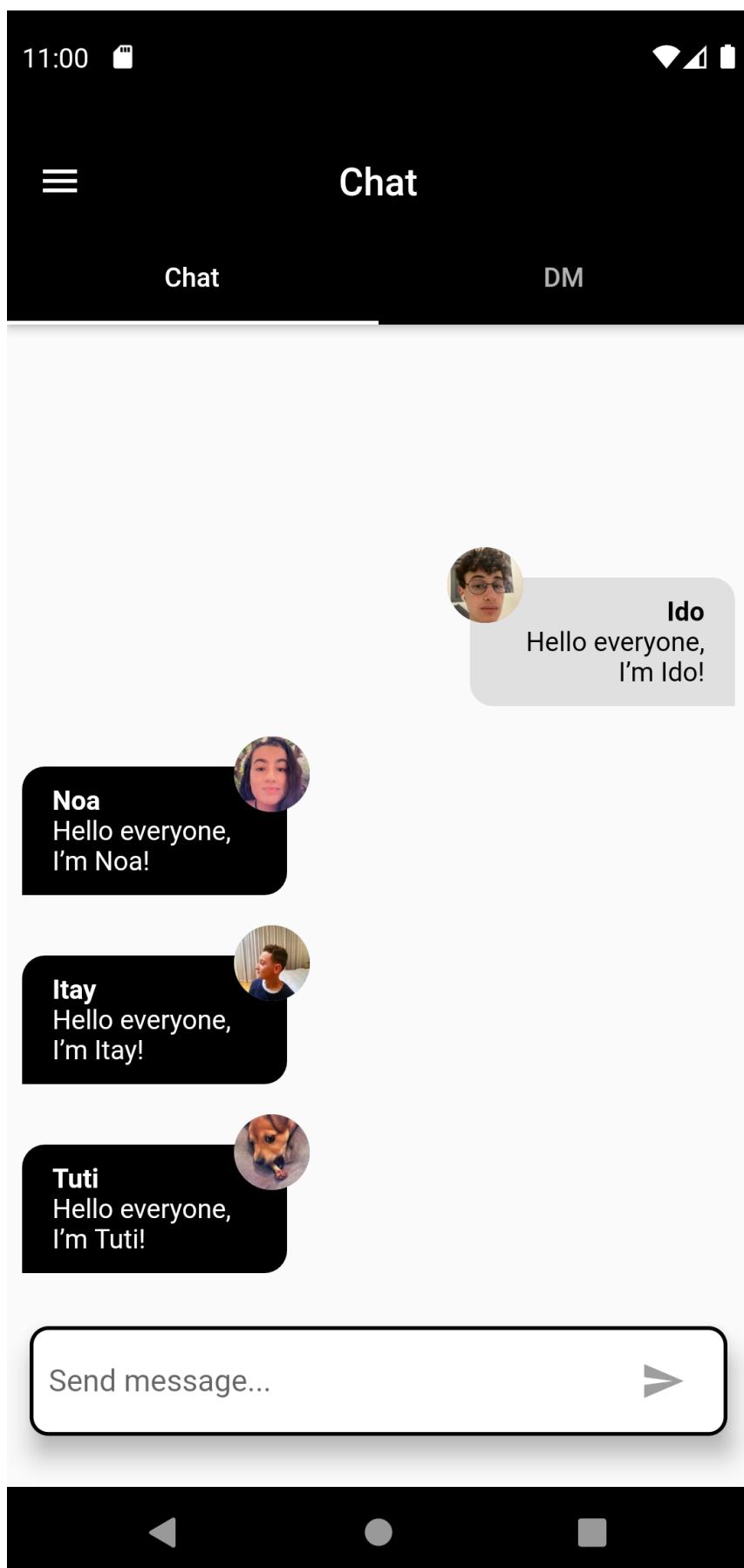
ארQUITקטורה של הפרויקט

הציג שרטוט על של הפרויקט

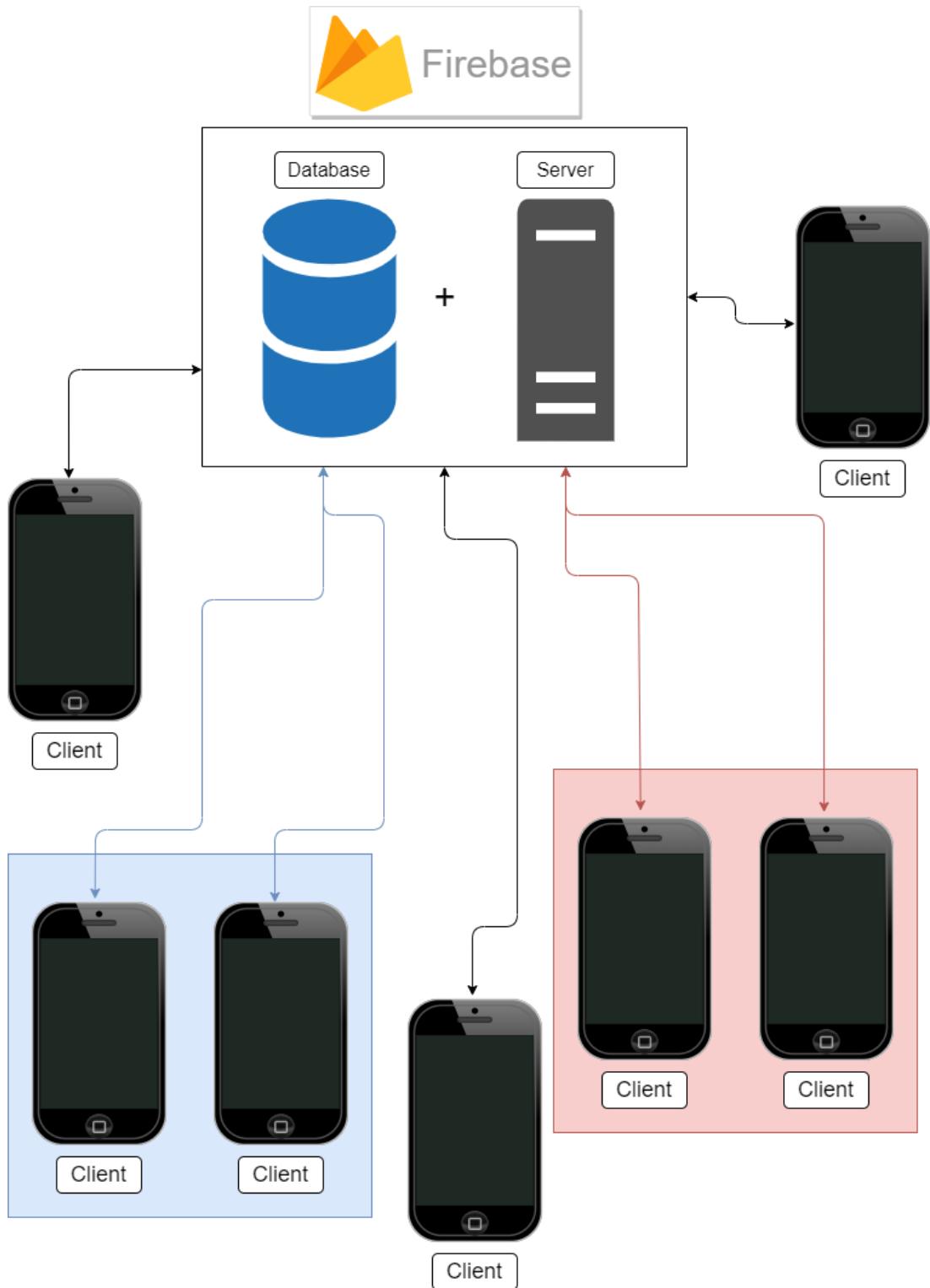
מייצג גם את הארכיטקטורה של הצ'אט הראשי של האפליקציה.



תמונה של מסך הציג את הכללי, ה – Broadcast, באפליקציה, בהתאם לארכיטקטורה למעלה.



ארQUITקטורת שיחות פרטיות בין שני משתמשים

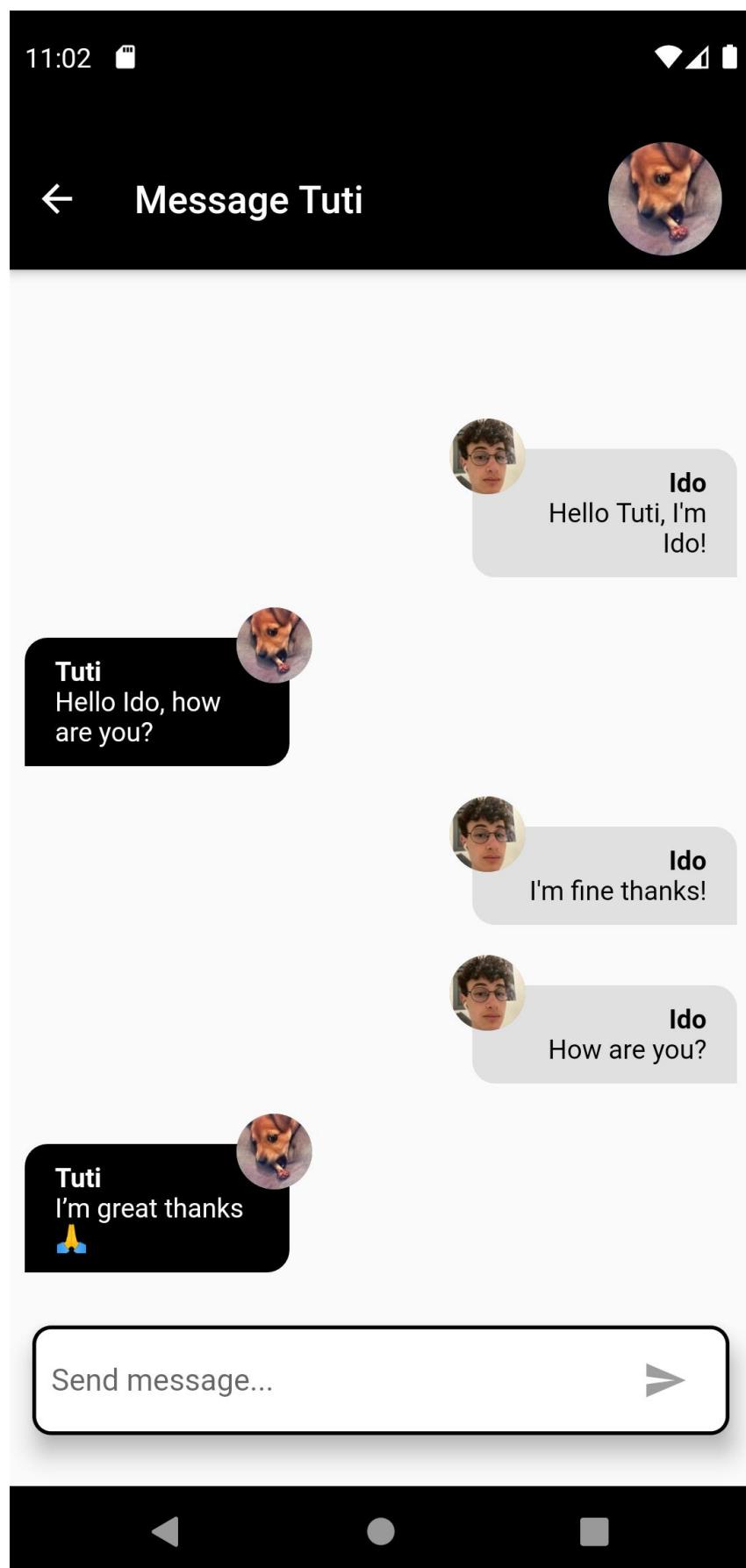


ניתן לראות כי יש שני משתמשים שמסומנים **באדום**, שניים **בכחול** וכמה **שחורים**.

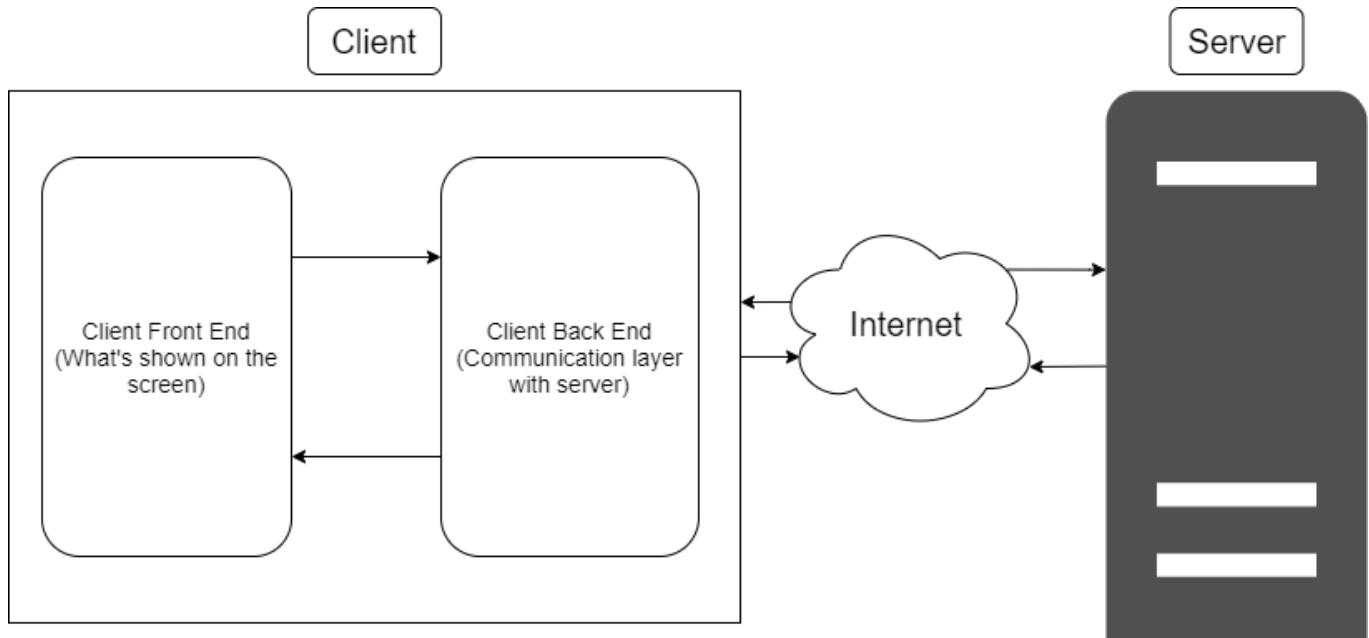
אלו **שחורים** מדברים בין עצמם ביצ'אט פרטי, אלו **שבכחול** גם כן, בעוד המשתמשים האחרים, **באדום**, עדין מדברים ביצ'אט הכללי.

ההודעות הפרטיות, כמו ביצ'אט הכללי, מאוחסנות בסיסי הנטונום, ועובדות מלקוח ללקוח דרך הרשת.

תמונה של מסך הציג הפעלת אפליקציה בין שני משתמשים, בהתאם לארכיטקטורה למעלה.



תפקידה של כל יחידה



Server -

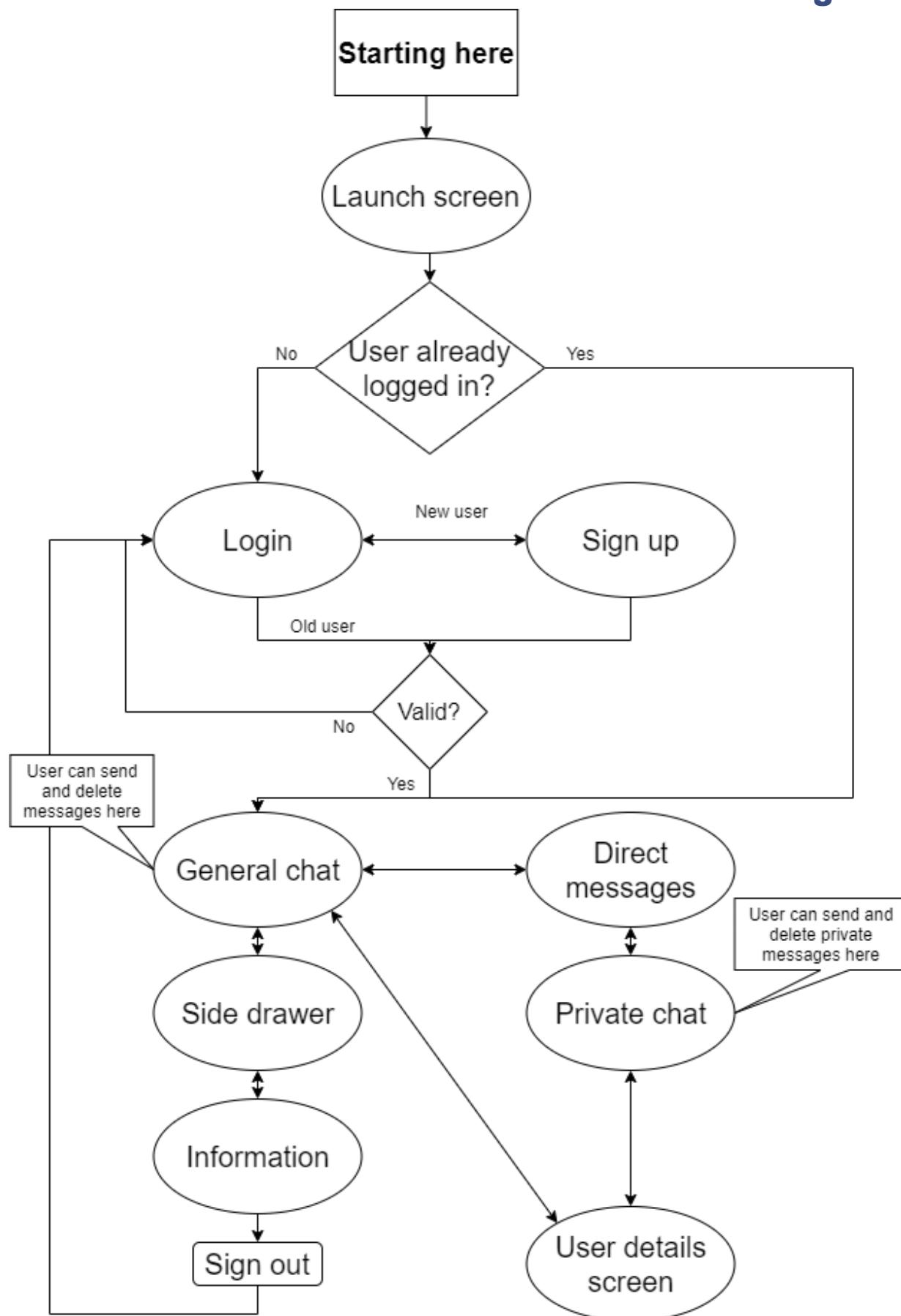
השרת מקבל אליו את כל ההודעות מכל הלוקוחות ומסדר אותן באופן הרצוי בסיס הנתונים. את ההודעות הכלליות לצ'אט הכללי, והודעות פרטיות לצ'אט הפרט'י אליו הן שייכות.

כמו כן, כאשר לוקוח מבצע רשותה לאפליקציה, נשמרם בסיס הנתונים כל הפרטים שלו – שם משתמש, אימייל, תמונה – חוץ מהטסמה, היא נשמרת מוצפנת בסיס הנתונים ואין给她 ישירה אליה. ניתן לשנותה באמצעות אימייל אך אי אפשר לראות את הנוכחית.

לקוח -

הלקוח מייצג כל משתמש של האפליקציה. הוא שולח בקשות, ומקבל תשובות מהשרת. כל לוקוח יכול לראות את כל ההודעות בצ'אט, אלו שהיו לפני שהגיע, אלו שהיו אחריו וallow שנשלחו כאשר לא הייתה לו קלייטה. הלקוח יכול לראות את כל ההודעות האלה גם בצ'אט הכללי וגם בצ'אטים הפרט'יים בהם הוא לוקח חלק.

General Use Case Diagram



הציג המקרים (Use cases) עבור הפונקציות הראשיות בפרויקט

הרשמה

תיאור הפעולה: המשתמש מנסה להירשם למערכת.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט והוא נכנס למסך ה – `Sign up`.

תנאי סיום: המשתמש רשום במערכת.

שלבי פעולה:

1. המשתמש נכנס למסך ההרשמה.
2. המשתמש מלא את השדות הריקים (אימייל, שם משתמש, סיסמה, בחירת תמונה) בהתאם לכתוב ליד השדות.
3. לאחר לחיצה על כפתור ה – `Sign up` מתבצעת בדיקה מצד הלוקו, בעל האפליקציה, (client) לגבי הקלטים, אם אינם תקינים (כלומר שם משתמש או סיסמה קצריים מדי, אימייל לא תקין, או שחרורה תמונה) המשתמש יתבקש למלא את הפרטים החסריםשוב.
4. לאחר הכנסת פרטיים נכונים ותקינים, פרטיו המשתמש ישלחו לשרת,¹ Firebase, המשתמש ירשם בסיס הנתונים, יתחבר לאפליקציה, והיא תעבור למסך היצוא הראשי.

התחברות

תיאור הפעולה: המשתמש מנסה להתחבר למערכת.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט והוא נכנס למסך ה – `Login`.

תנאי סיום: המשתמש מתחבר למערכת.

שלבי פעולה:

1. המשתמש נכנס למסך ההתחברות.
2. המשתמש מלא את השדות הריקים (אימייל, סיסמה) בהתאם לכתוב ליד השדות.
3. לאחר לחיצה על כפתור ה – `Login` מתבצעת בדיקה מצד הלוקו (client) לגבי הקלטים (כלומר אם אימייל אינו תקין, או סיסמה קצרה מדי) יתבקש למלא את הפרטים החסריםשוב.
4. לאחר בדיקה אצל הלוקו (client), המידע נשלח לשרת, Firebase, שם נבדקים אם האימייל והסיסמה קיימים במערכת. אם לא, נשלחת הודעה בחזרה ללוקו על קר ומוצגת על המסך הודעה על קר ששם המשתמש או הסיסמה אינם נכונים, וצרכי להכיסםשוב.
5. לאחר הכנסת פרטיים נכונים ותקינים, הלוקו יתחבר למערכת, והאפליקציה תעביר למסך היצוא הראשי.

¹ Firebase, השירות בו אני משתמש בפרויקט שלי, וגם לבסיס הנתונים, יסביר בהמשך בפרק של בסיס הנתונים.

שליחת הודעה בצד'אט הכללי

תיאור הפעולה: המשתמש מנסה לשלוח הודעה בצד'אט הכללי.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט, הוא מחובר לאפליקציה, ונכנס למסך הצד'אט הכללי.

תנאי סיום: המשתמש שולח הודעה בצד'אט הכללי, כך שכל המשתמשים של האפליקציה יוכל לראותה.

שלבי פעולה:

1. המשתמש נכנס למסך הצד'אט הכללי.
2. המשתמש מלא את השדה בתחלת העמוד של שליחת ההודעה בהודעה שהוא רוצה לשולח.
3. המשתמש לחץ על כפתור השליחה (דומה לצורת חץ).
4. ההודעה נשלחת את השירות, Firebase, הוא מוסיף אותה לבסיס הנתונים כ –² Document חדש, תחתה – chat Collection של .chat
5. בעקבות כך שבבסיס הנתונים בזמן אמת, כל המשתמשים מקבלים את ההודעה מבסיס הנתונים ישיר אחרי שהיא נסופה לשם.

שליחת הודעה בצד'אט פרט依 למשתמש חדש

תיאור הפעולה: המשתמש מנסה לשלוח הודעה למשתמש שהוא עוד לא דבר אליו.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט, הוא הת לחבר למערכת והוא נכנס למסך הצד'אט הכללי.

תנאי סיום: המשתמש מדבר בצד'אט פרט依 עם האדם אליו רצה לדבר.

שלבי פעולה:

1. המשתמש נכנס למסך הצד'אט הראשי.
2. המשתמש לחץ לחיצה ארוכה על הודעה שהמשתמש אליו הוא רוצה לשולח הודעה שלח.
3. המשתמש מגיעה למסך עם הפרטים של אותו משתמש - שם משתמש, אימייל, תמונה פרופיל. ובנוסף בתחלת יש כפתור ליצירת קשר עם אותו משתמש.
4. לאחר לחיצה עם הכפתור, המשתמש יכול לשלוח הודעה פרטיה למשתמש האחר, בדיק כפ' שהסבرت בצד'אט הכללי, רק שהפעם הודעה פרטיה בנפרד לצד'אט זה, כך שרק שני משתמשים אלו יכולים לראותו.
5. כל צ'אט פרט依 חדש נשמר כ – Document ב – Collection chatRoom, ובמסמך זה נשמרים השם של המסר, ורשימה של המשתמשים שנמצאים בו. בנוסף המסמך מכיל Collection חדש הנקרא chats, שמכיל את כל ההודעות שנשלחו בין שני המשתמשים, כל הודעה במסמך משלה.

² המושגים Document ו Collection יסבירו גם כן בפרק של בסיס הנתונים, יחד עם ההסבר על Firebase.

czpiaha בכל השיחות הפרטיות של המשתמש, וכניסה לשיחה פרטית קודמת

תיאור הפעולה: המשתמש מנסה לשלוח הודעה למשתמש שכבר דבר אליו בעבר.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט, הוא התחבר למערכת והוא נכנס למסך היצ'ט הכללי.

תנאי סיום: המשתמש מדבר בז'אט פרט' עם האדם אותו רצה לדבר.

שלבי פעולה:

1. המשתמש נכנס למסך היצ'ט הראשי.
2. המשתמש לוחץ על כפתור ה – DM על מנת לצפות בכל השיחות הקודמות שלו.
3. המשתמש לוחץ על השורה עם שמו של האדם אליו רוצה לדבר.
4. לאחר לחיצה עם השורה, המשתמש יכול לשלוח הודעה פרטית למשתמש האחר, בדיק כפ' שהסבرت בז'אט הכללי, רק שהפעם ההודעות יישמרו בנפרד לצ'אט זה, כך שרק שני משתמשים אלו יכולים לראותו.
5. כל ההודעות של היצ'ט הפרט' שכבר נוצר נשמרות באותו Document שנוצר עבורי היצ'ט הפרט' זהה כאשר דיברו בפעם הראשונה.

מחיקת הודעה

תיאור הפעולה: המשתמש מנסה למחוק הודעה שליח ב עבר, או בז'אט הכללי או בז'אט פרט' מסוים.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט, הוא התחבר למערכת והוא נכנס למסך היצ'ט הכללי.

תנאי סיום: המשתמש מחק את ההודעה שרצה.

שלבי פעולה:

1. המשתמש לוחץ לחיצה ארוכה על ההודעה שלו שהוא רוצה למחוק.
2. קופץ חלון השואל אותו אם הוא בטוח שהוא רוצה למחוק את ההודעה מכלם.
3. אם לחץ כן, ההודעה תמחק מהמקום בו היא נמצאת בבסיס הנתונים, וכן גם מהצ'אט.
4. אם לחץ לא, האפליקציה תחזיר אותו למקום מצב כמו שהיא היתה קודם.

התנטקות מהאפליקציה

תיאור הפעולה: המשתמש מנסה להtenantק מאפליקציה.

תנאים מקדים: המשתמש הפעיל את האפליקציה, יש לו חיבור פעיל לאינטרנט, הוא התחבר למערכת והוא נכנס למסך היצ'ט הכללי.

תנאי סיום: המשתמש התנטק מאפליקציה ו חוזר למסך ההתחברות.

שלבי פעולה:

1. המשתמש לוחץ על המגירה הצדית בצד שמאל העליון במסך היצ'ט הכללי.
2. לוחץ על כפתור ה – Information. שם יש לו מידע לגבי איך למחוק הודעה וגם איך לשלוח הודעה למשתמש אחר. בנוסף ישנו כפתור של out Sign.
3. לאחר לחיצה על הכפתור, הודעה על רצון להתנטקות נשלחת אל השרת, והוא מנתק את המשתמש מהאפליקציה. לאחר קבלת אישור על התנטקות מהשרת, האפליקציה (client), מעבירה את המשתמש למסך ההתחברות.

הציג הפתרון המוצע והסיבות לבחירתו

רציתי לעשوت אפליקציית צ'אט, שתיהיה בעלת אפשרות לשלוח הודעות לכל משתמשי האפליקציה (broadcast) ובעלת אופציה לשלוח הודעות פרטיות בין שני משתמשים, لكن כך עשייתי. לפתרון שלי השתמשתי, כפי שכבר ציינתי, בפיתוח אפליקציה בטכנולוגיה של Flutter.

בחратני לשימוש ב-Flutter מושם שזה היה נראה הפתרון הטוב ביותר עבורו. לאחר שהחיפשתי הרבה על איך אפשר לבנות אפליקציית צ'אט, וגיליתי אופציות שונות כמו Kivy, Android studio וכמו כן, הבנתי Flutter הוא התשובה. הכל נכון לשימוש, אפשר לי לפתח אפליקציה גם ל-Android וגם ל-OS באותו הזמן, ובועל תמיכה ודוקומנטציה שלא ראייתי כמותה.

בשביל השירות שלי השתמשתי ב- Firebase, קלומר אני כתבתי את קוד הלוקה, האפליקציה, ו- Firebase היה השירות שלי, כדי שלא אצטרך גם לכתוב את קוד השירות, עקב חוסר בזמן, וגם כי זויה פלטפורמה שבניה באופן מותאם ל-Flutter. לפניה ותקשות יותר נועה עם השירות, השימוש ב- Packages שונים.

טכנולוגיות בהן נעשה שימוש בפרויקט

Flutter

Flutter היא פלטפורמת פיתוח תוכנה לאפליקציות, עם ממשק משתמש פתוח הפתחה על ידי גугл. הוא משמש לפיתוח יישומים חזוי פלטפורמות עבר OS, Linux, Mac, Windows,iOS, Android, מbas ו- Web מבסיס קוד ייחיד.

לעומת פתרונות אחרים הנמצאים בשוק, כפי שהסבירתי לעיל, כגון Kivy ו- Android studio ו- Flutter היא הפלטפורמה הנוחה ביותר.

ראשית, בסיס הקוד שלו נכתב בשפת Dart, שפה שהייתה עבורי מאוד נוחה לשימוש, ממשום שהיא דומה בצורת כתיבתה ל- Java, שפה בה השתמשתי הרבה בעברית כאשר למדתי מדעי המחשב, וגם ל- Python ו- Script Dart היא שילוב נוח של שפות שהייתי בקיא בהן, מה שהקל עלי' ללמידה, ולהשתפר בה מהר מאוד.

בנוסף, מכיוון ש- Flutter היא פלטפורמה שפותחה על ידי גугл, והיא נפוצה מאוד כiom, יש אליה דוקומנטציה אינסופית באינטרנט, מה שמקל על הליך הלמידה.

Firebase

Firebase היא פלטפורמת שירותי בסיס נתונים, שפותחה על ידי גугл ליצור יישומים ניידים ורשת. במקור זו הייתה חברת עצמאית שנסודה בשנת 2011. בשנת 2014, גугл רכשה את הפלטפורמה והיא מהווה כעט הצעת דגל שלהם לפיתוח אפליקציות.

Firebase היא בעצם ה- Backend שוגול מציעה לאפליקציות. היא בעלת שירות (Server), בסיס נתונים (Real time database) ובסיס נתונים בזמן אמת שנקרא – Firestore.

Firebase היא פלטפורמה מאוד נוחה לשימוש, עקב כך שהפלטפורמה מפותחת על ידי גугл, כמו Flutter, ישנם SDK's, Libraries ו- Packages, מאוד נוחים לקישור בין הפלטפורמות הללו, מה שעזר לייעול העבודה, ולקיים זמן הלמידה.

כמו כן, האתר והגישה למידע בסיס הנתונים קלה ונוחה מאוד לשימוש, ניתן לראות בכל רגע נתון את כל הנתונים בסיס הנתונים, שקייפות שעוזרת מאוד לפתור בעיות ולהבין טעויות שנעשו.

מדריך למשתמש

הוראות התקינה

.Android, iOS, Web או שונות פלטפורמות של 3 להריץ ניתן שילוב האפליקציה את.

בעתיד אולי עליה את האפליקציה לחניותות האפליקציות, ואז הוראות ההתקינה יהיו רק להוריד את האפליקציה מחנות האפליקציות שתואמת למכשיר זהה.

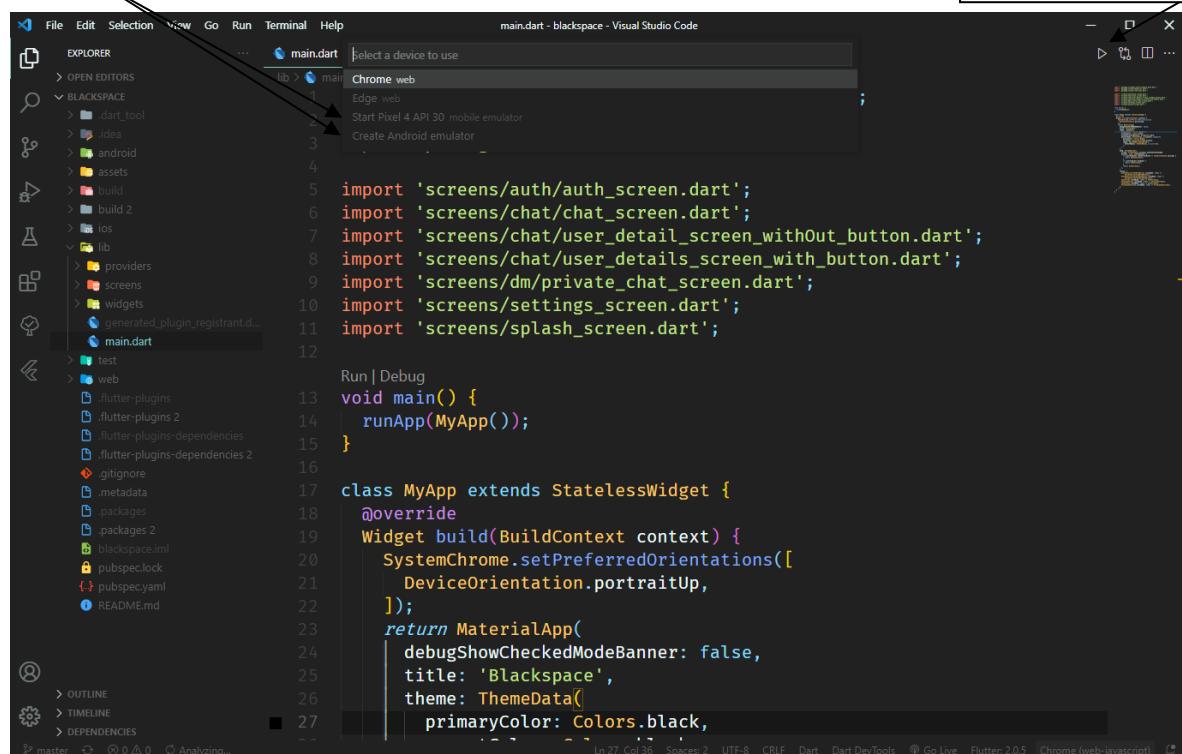
אר כעט אלּו הַנְּחֹרְאָות הַתְּקִנָּה

עבורי iOS ו-Android

- .1. להתקין Flutter - <https://flutter.dev/docs/get-started/install/windows> - Flutter .1
 - .2. עברו Android: להתקין Android studio וליצור סימולטור של מכשיר אנדרואיד שירוץ על המחשב -
https://www.youtube.com/watch?v=WkEf1fa1sn0&ab_channel=NiyasC
 - .3. עברו OS(macOS): להתקין Xcode מה – App store וליצור סימולטור של מכשיר OS שירוץ על המחשב – <https://bit.ly/3tUKqYi>
 - .4. להתקין VSCode (לא חובה אבל כך יוכל להריץ את הפרויקט) -
<https://code.visualstudio.com>
 - .5. להוריד את הקוד של הפרויקט שלי מ – GitHub ולפתחו אותו באמצעות VSCode
<https://github.com/itsidohirsh/blackspace>
 - .6. בטור התיקיה lib לבחרו את הקובץ main, לבחור את הסימולטור שיצרנו מתוך VSCode על מנת שהקובץ יירוץ עליון וללכוד צילום.

2. לבחר באחת מהopcיות האלו

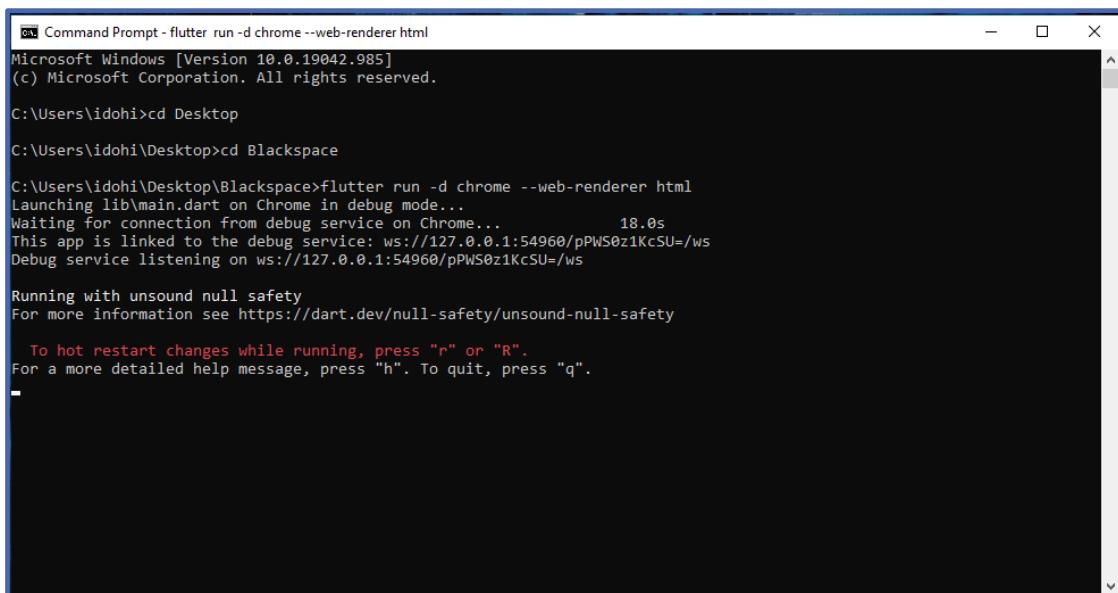
3. ללקוח צאו רשות פראס - חוץ



1. ללחוץ כאו בשבייל לבחור מקשריר

עבור Web

- .1 להתקין <https://flutter.dev/docs/get-started/install/windows> - Flutter
- .2 להוריד את הקוד של הפרויקט שלי מ [GitHub](https://github.com/itsidohirsh/blackspace) – Command Prompt
- .3 להיכנס לתיקייה של הקובץ דרכו – Enter `flutter run -d chrome --web-renderer html` וללחוץ .4
- .5 יפתח חלון ב – Google chrome עם האפליקציה



```
Command Prompt - flutter run -d chrome --web-renderer html
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

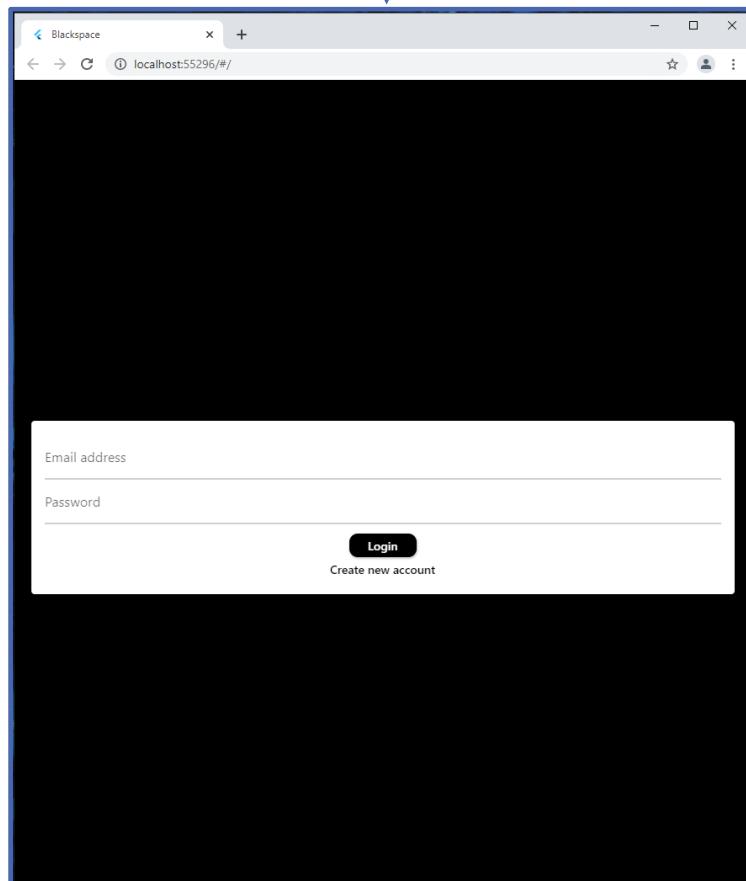
C:\Users\idohi>cd Desktop

C:\Users\idohi\Desktop>cd Blackspace

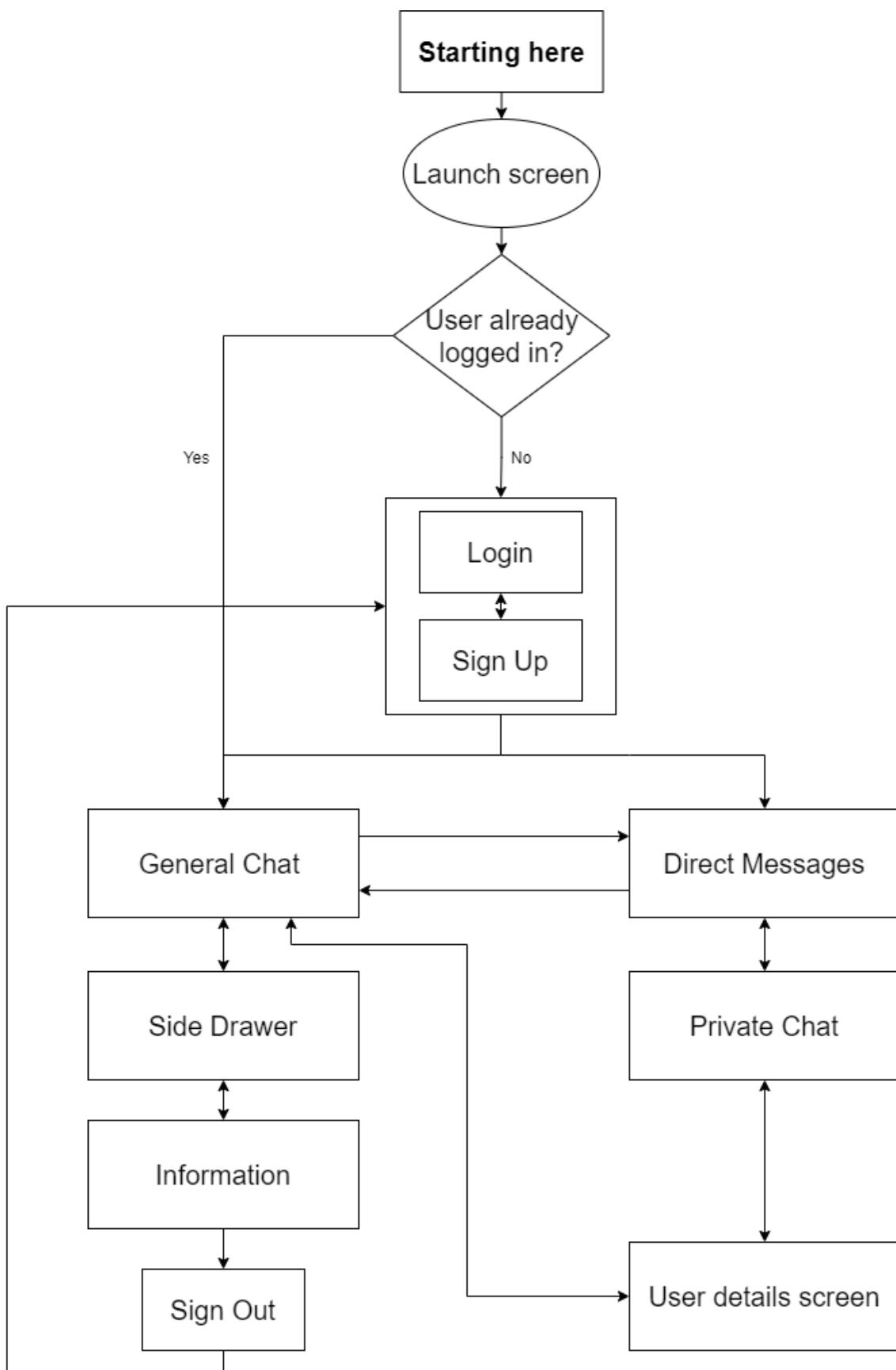
C:\Users\idohi\Desktop\Blackspace>flutter run -d chrome --web-renderer html
Launching lib/main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...           18.0s
This app is linked to the debug service: ws://127.0.0.1:54960/pPWS0z1KcSU=/ws
Debug service listening on ws://127.0.0.1:54960/pPWS0z1KcSU=/ws

Running with unsound null safety
For more information see https://dart.dev/null-safety/unsound-null-safety

To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".
```



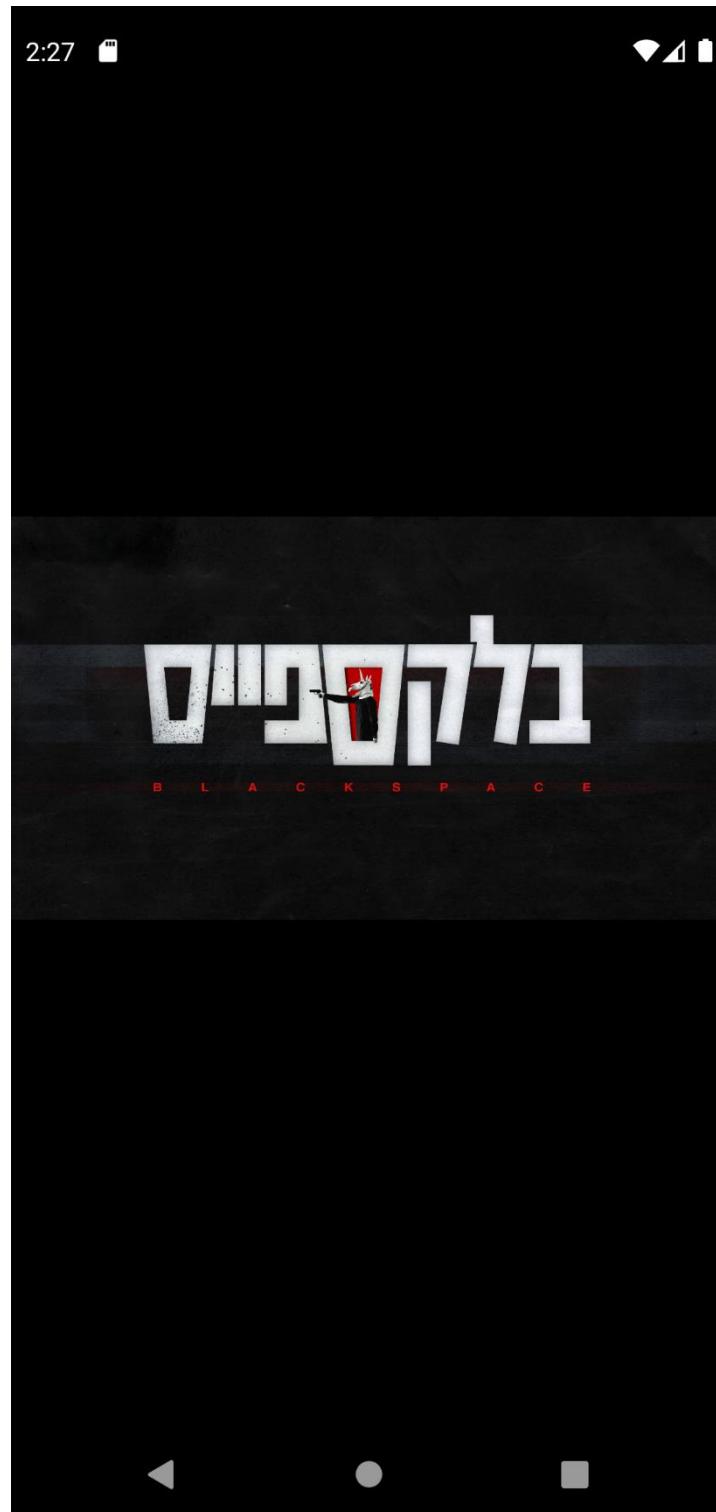
היררכיה מסכימים



תרשים מסכימים

מסמך טעינה

מסך הטעינה של האפליקציה, מה שהמשתמש רואה כאשר האפליקציה טוענת.



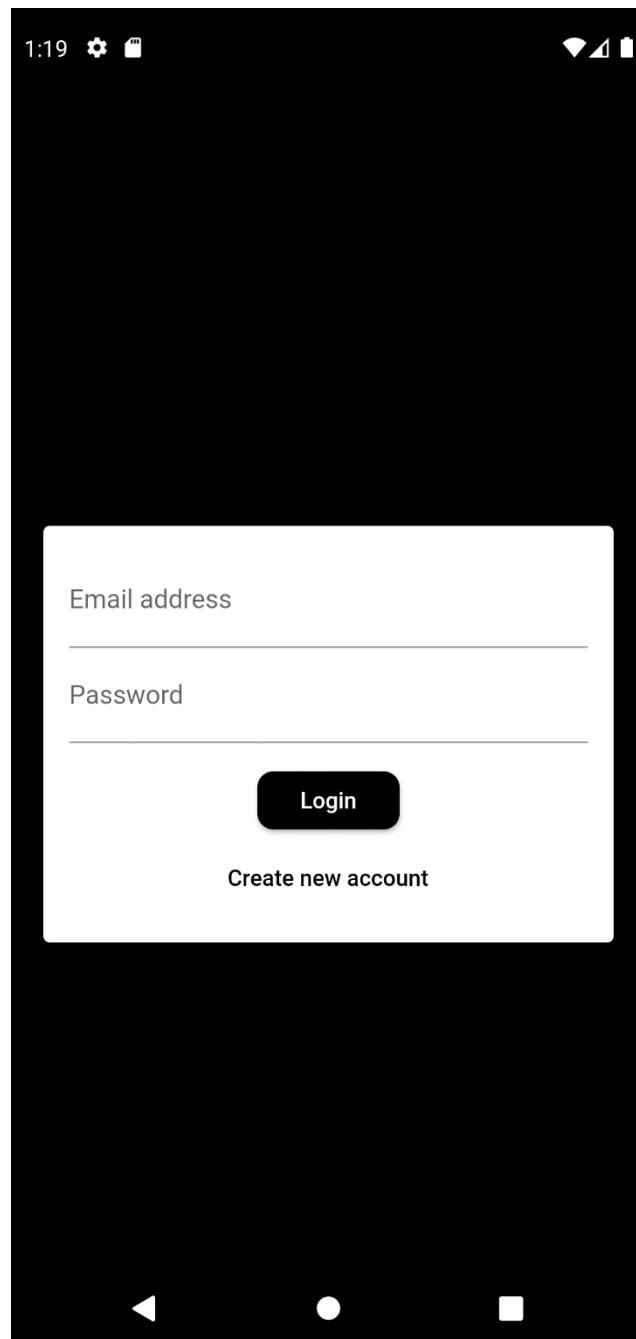
התחברות

המסך הראשון שהמשתמש רואה כאשר הוא פותח את האפליקציה בפעם הראשונה. בפעמיים הבאות, אם לא התנתק מהאפליקציה, המשמש והסימא שלו ישמרו ויתחבר ישירות למערכת. (תיה אופציה להתקנתוות מסך ההגדרות).

קלט: אימייל וסיסמה של המשתמש.

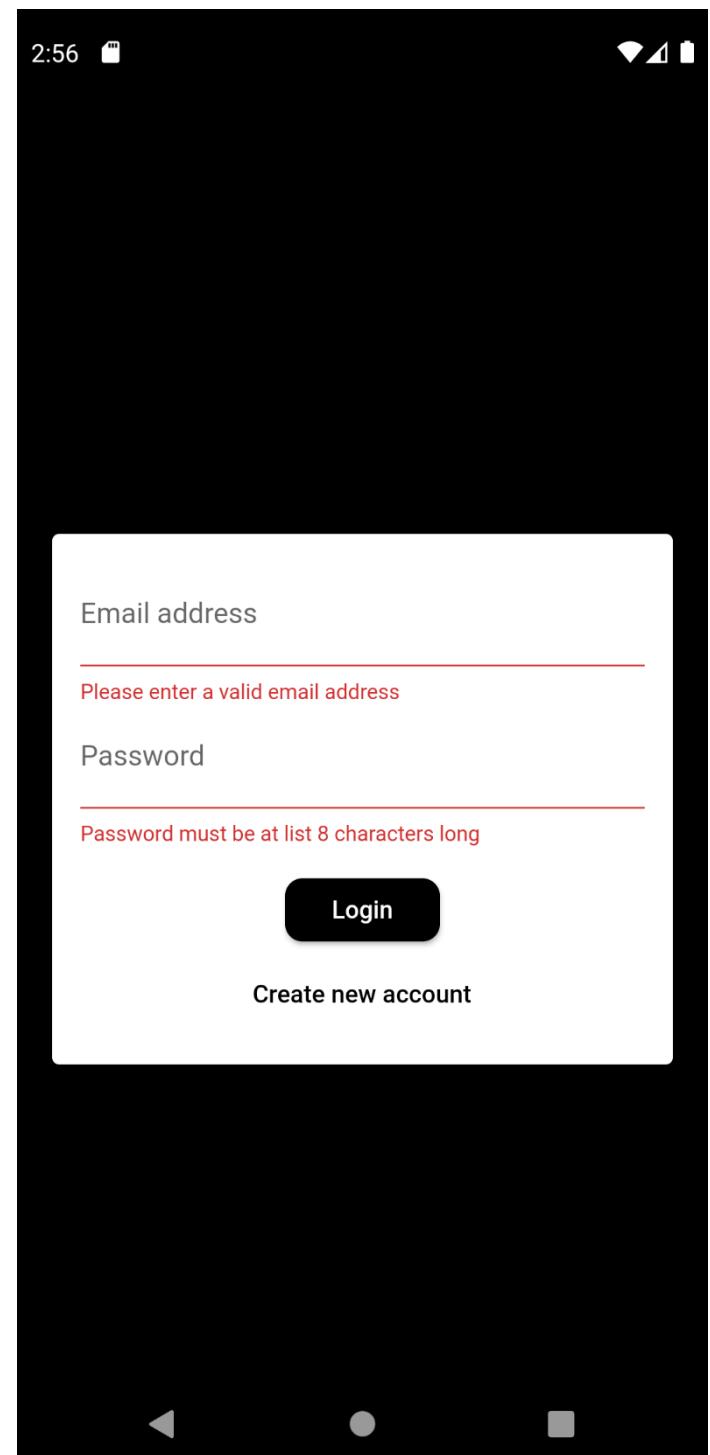
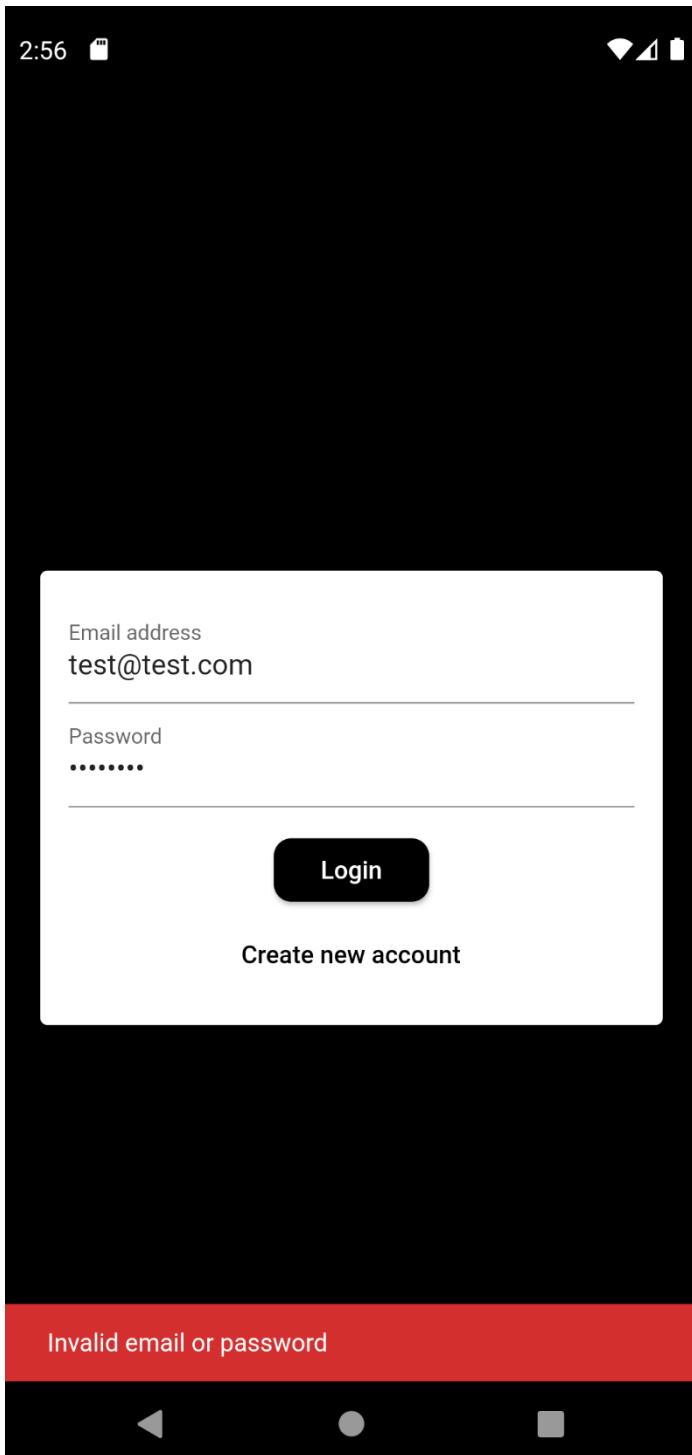
פלט: במקרה של שגיאה, בעקבות הולידציה, תציג הודעה מתאימה למשתמש. במקרה של הצלחה, יעבור המשתמש למסך הראשי (מסך הציג).

מסך זה ניתן לעבר ישירות למסך ההרשמה, באמצעות לחיצה על הכפתור המופיע בכרוך וזרה.



בדיקות האימייל והסיסמה מול הشرط
ובواسע הנתונים.
במקרה זה הנתונים אינם נכונים.

בדיקות תקינות האימייל והסיסמה
אצל הלוקוח



ההרשמה

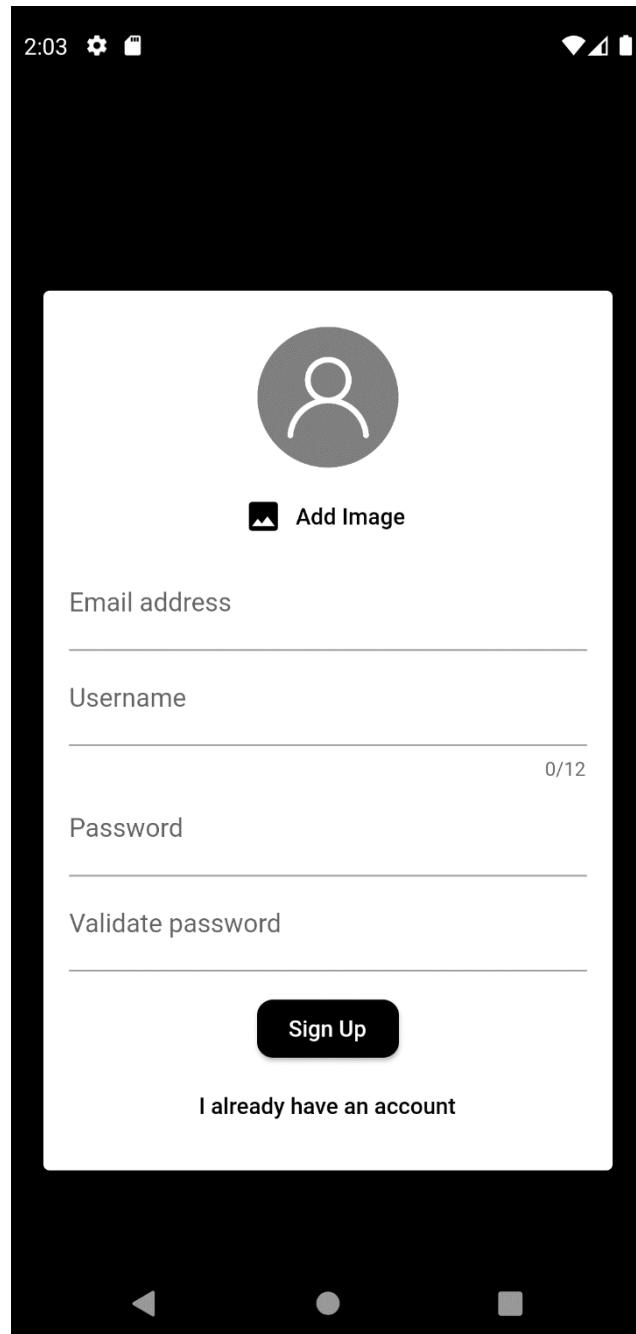
מסך ליצירת משתמש חדש באפליקציה.

קלט: אימייל, שם משתמש, סיסמה ותמונה שמצוולמת בזמן אמת מהמכשיר.

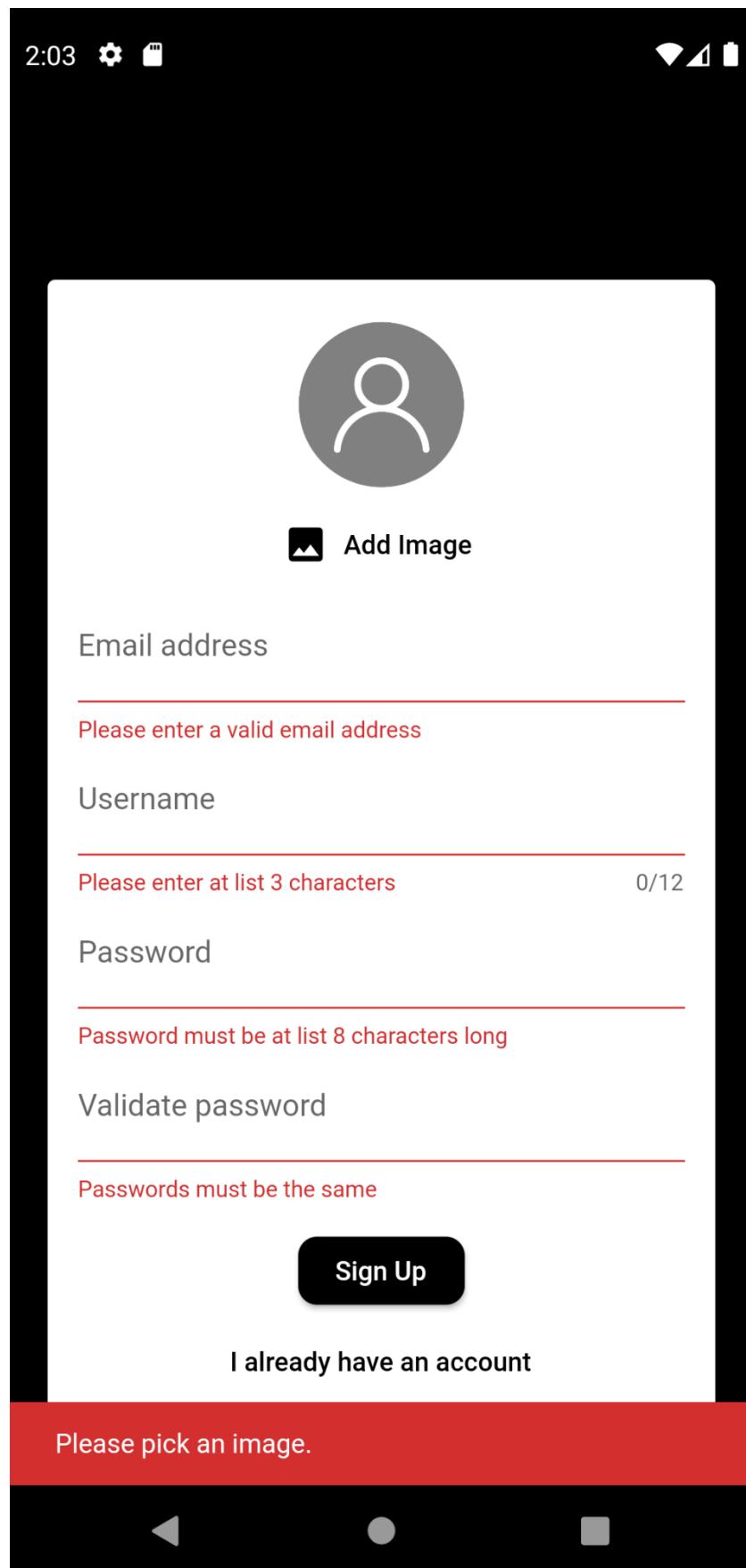
אני מבקש אימייל מסוים שבשביל הרשמה לשרת, Firebase, יש צורך באימייל וסיסמה. בנוסף, אני מבקש אימייל מסוים שבשביל הפרטים האישיים על כל משתמש. בשבייל יצירה קשר דרך אימייל בין שני משתמשים מחוץ לאפליקציה. כמו כן, אפשר להשתמש באימייל על מנת לאפס סיסמה. (אך אני לא הספקתי להוסיף את האופציה הצעאת עדיין).

פלט: אם כל הפרטים האישיים (עברו את הוולידציה) ייווצר משתמש חדש, הוא יתווסף לבסיס הנתונים, והמשתמש יועבר למסך הציג הראשי.

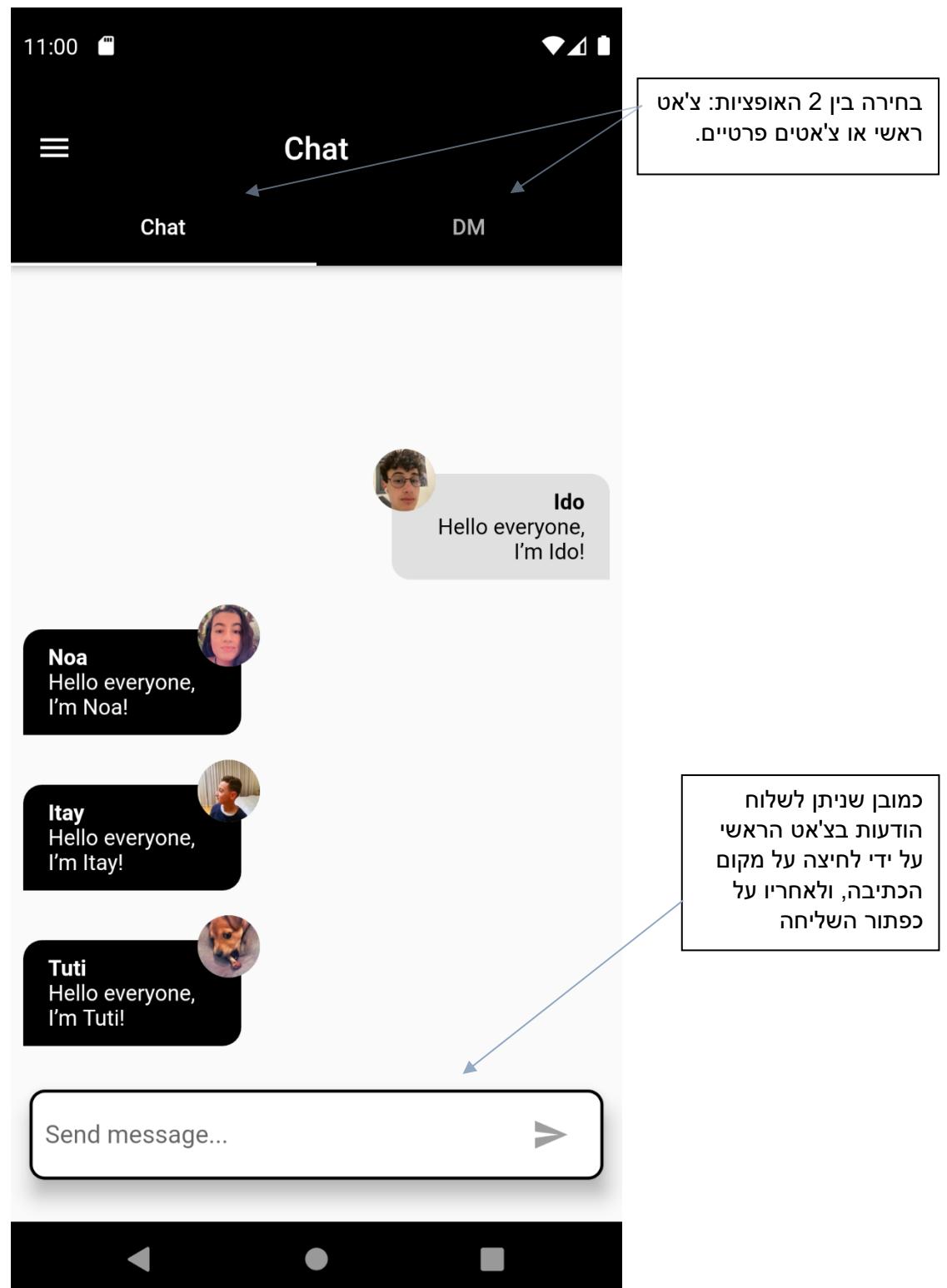
מססז זה ניתן לעבור שירותי לקוחות להתחברות, באמצעות לחיצה על הכפתור המופיע לך, וחזרה.



כל הودעות הולידציה שיכולה להיות בעת ניסיון יצירת משתמש חדש



מסמך צ'אט ראשי
מסמך בו המשתמש רואה את כל ההודעות שנמצאות בצ'אט הגדל של האפליקציה, צ'אט הפלט: כל ההודעות שנשלחו בצ'אט. גם אלו לפני שהמשתמש נוצר, וגם אלו שאחריו.

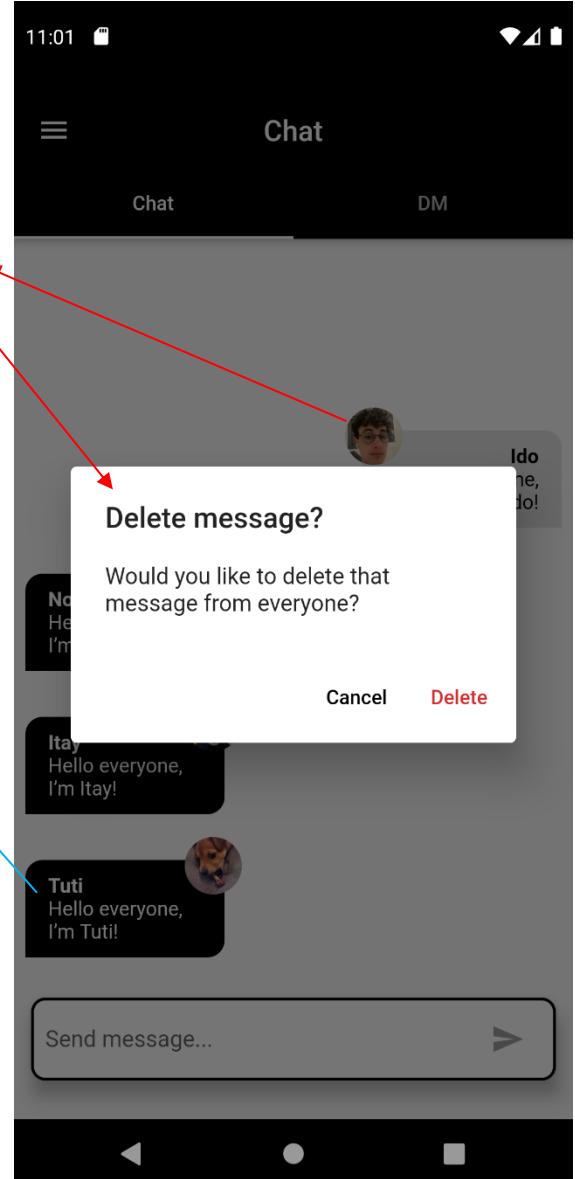
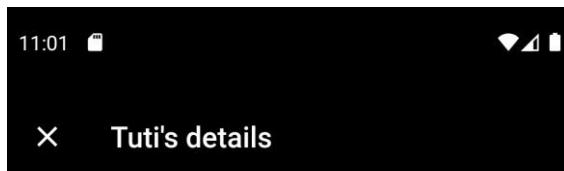


משתמש מחובר יכול למחוק את ההודעות שנשלחו על ידי המשתמש המחבר ה-**צבע אפור**, בעוד ההודעות שנשלחו על ידי אחרים הן בצבע שחור, על ידי לחיצה ארוכה על ההודעה אותה הוא רוצה למחוק.

כמובן שם משתמש לוחץ ארוך על הודעה שימושה אחר שלח, הוא לא יכול למחוק אותה. במקרה זה, הוא מועבר לדף פרטיים על אותו אדם.

לאחר לחיצה ארוכה על הודעה שנשלחה על ידי משתמש אחר, במקרה זהה **Tuti** (המשתמש של הכלבה שלו 😊)

לאחר לחיצה ארוכה על הודעה שנשלחה על ידי המשתמש המחבר, במקרה זהה **Ido** (המשתמש שלו)



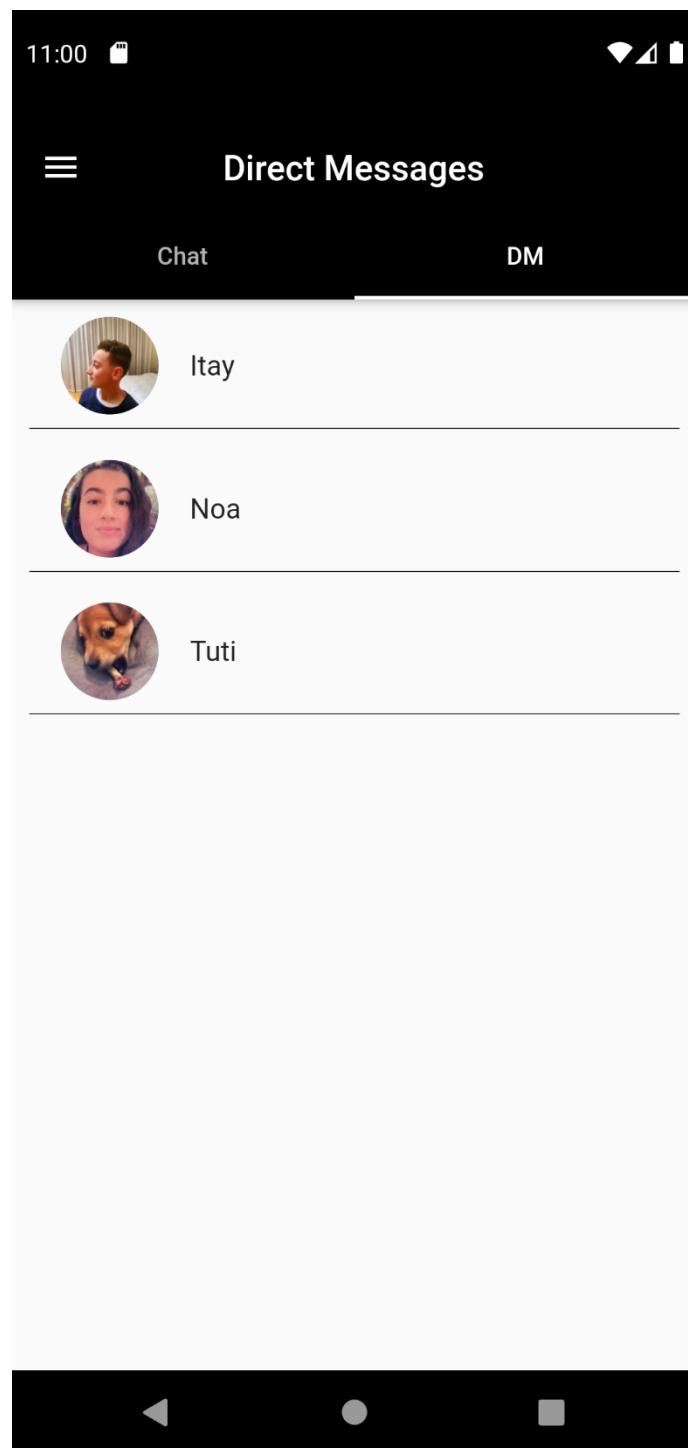
מסך זה ניתן לעבור, על ידי לחיצה על כפתור "Send a private message to ..." לשיחה פרטית עם אותו אדם.

מסמך כל השיחות הפרטיות

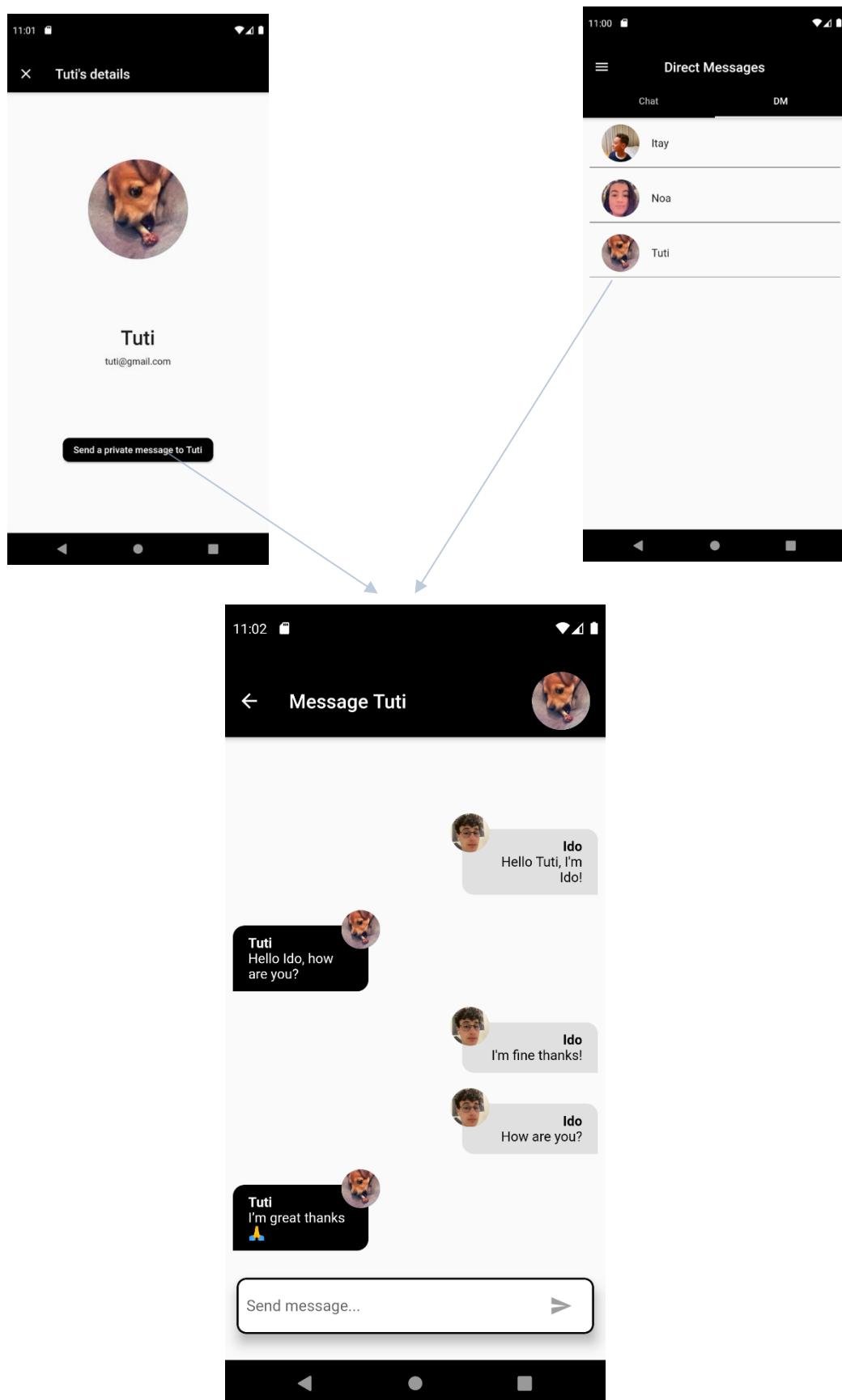
במסמך זה המשתמש רואה רשימה של כל האנשים איתם הייתה לו כבר שיחה פרטית.

במקרה זה, אנו רואים את כל השיחות הפרטיות שהיו ל Ido , המשתמש הנוכחי, עם אנשים אחרים.

כאשר נלחצת שורה מסוימת, המשתמש מועבר למסמך בו הוא יכול לנוהל שיחה פרטית עם אותו האדם, אותו המסמך אליו ניתן להגיע על ידי לחיצה על כפתור "Send a private message to ..." מתוך מסך הפרטים על האדם, אליו הגיע המשתמש על ידי לחיצה ארוכה על הודעה בצלע הראשי.



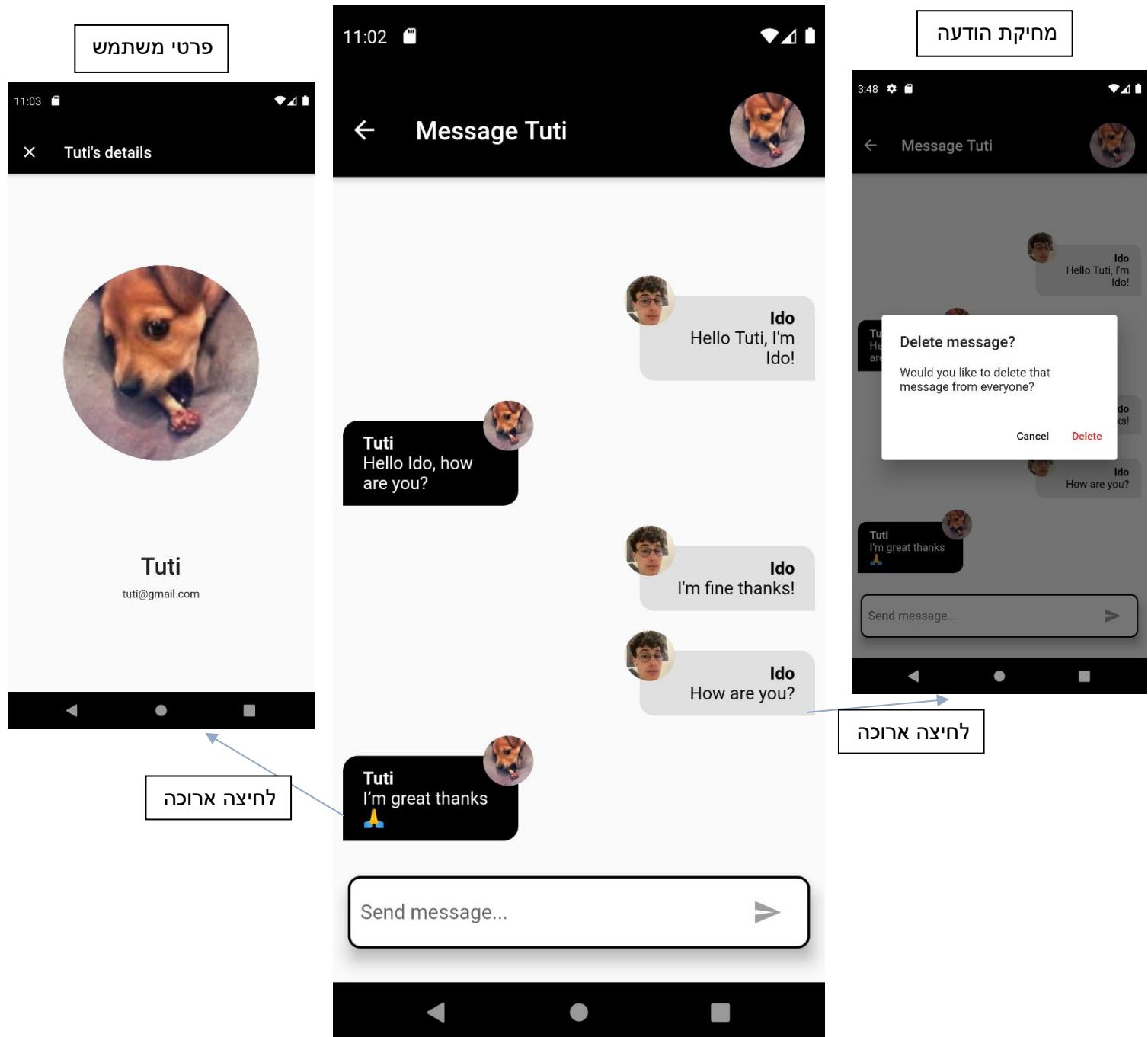
לחיצה על אחת מהאפשרויות האלו תוביל לשיחה הפרטית עם האדם אותו אנו רוצים לדבר. במקרה זהה, .Tuti



מסמך שיחה פרטית

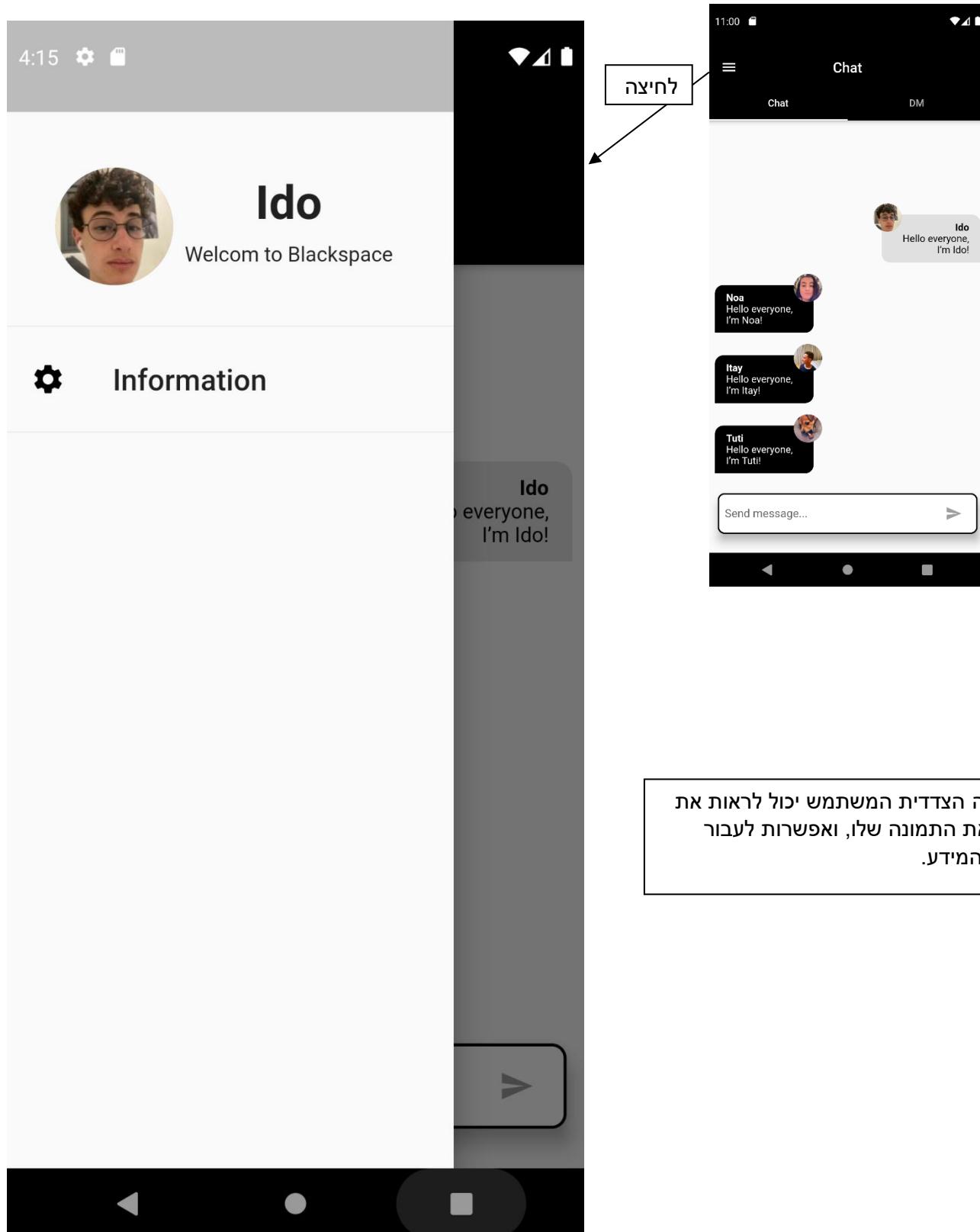
במסך זה מתנהלת שיחה בין שני משתמשים. לכל משתמש יש אפשרות לשלוח הודעה, בדיקן כמו בצל'אט הראשי.

לחיצה ארוכה על הודעה שנשלחה מהחובר, גם כזו בצל'אט הראשי, מאפשרת למחוק אותה. וכן לחיצה ארוכה על הודעה של המשתמש השני או לחיצה על תמונה הפרופיל של המשתמש השני, תעביר שבLPARAM עם הפרטים על אותו אדם, כמו בצל'אט הראשי, רק הפעם בלבד האפשרות לשלוח לו שם הודעה פרטית (על מנת למנוע מעגלים אינסופיים של לחיצה ארוכה על הודעה, כניסה לצ'אט הפרטי, לחיצה ארוכה על הודעה, כניסה לצ'אט הפרטי ...)



Side drawer

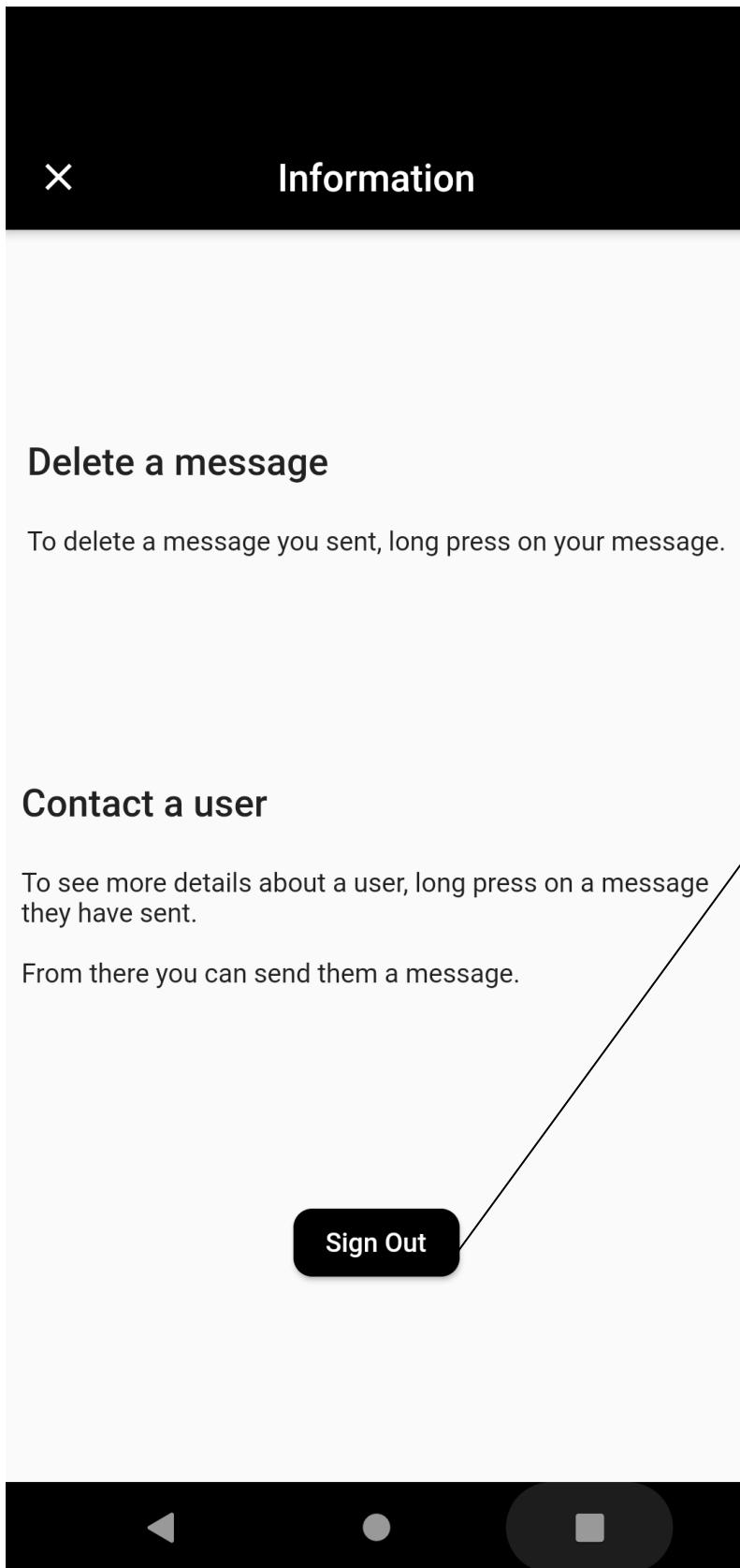
המסמך הראשי ניתן לפתח מגירה צדדיות עם אפשרויות שונות. הcptor שנראה כמו "המברגר" ממוקם לצד השמאלי העליון במסך הראשי של האפליקציה.



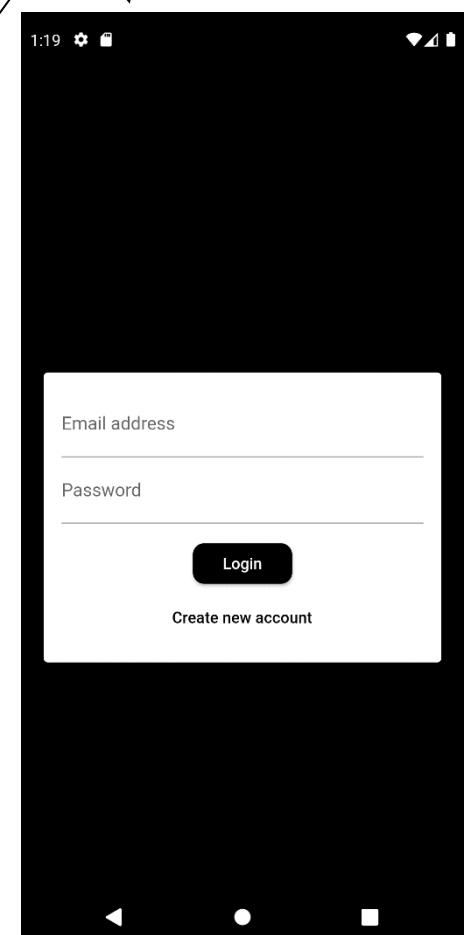
במגירה הצדית המשמש יכול לראות את
שמו, את התמונה שלו, ואפשרות לעבור
למסך המידע.

מסמך מידע

במסמך המידע המשמש יכול לקרוא כיצד הוא יכול למחוק הודעה, לשЛОח הודעה פרטית לאדם אחר, וכמונן להתנתק מהמערכת.



לאחר לחיצה על כפתור
ההתנתקות, המשתמש חוזר חזרה
למסך ההרשמה / כניסה
לאפליקציה.



בסיס הנתונים

הסבר כללי על בסיס הנתונים בו השתמשתי **Firebase**

בדומה لما שכתבתי בטכנולוגיות שבהן השתמשתי בפרויקט, Firebase היא פלטפורמת שירותי ובסיסי נתונים, שפותחה על ידי גוגל לצורכי יישומים ניידים ורשת. במקור זו הייתה חברת עצמאית שנסודה בשנת 2011. בשנת 2014, גוגל רכשה את הפלאטפורמה והיא מהווה כעת הצעת דגל שלהם לפיתוח אפליקציות.

Firebase היא בעצם ה – Backend שוגול מציעה לאפליקציות. היא בעלת שירות (Server), בסיס נתונים Firestore (Real time database) ובבסיס נתונים בזמן אמת שנקרא –

Firebase היא פלטפורמה מאוד נוחה לשימוש, עקב כך שהפלטפורמה מפותחת על ידי גוגל, כמו Flutter, ISDN's Libraries – Packages SDK, מה שעזר ליעול העבודה, וליקזור זמן הלמידה.

כמו כן, האתר והגישה למכשירים ניידים קללה ונוחה מאוד לשימוש, ניתן לראות בכל רגע נתון את כל הנתונים בבסיס הנתונים, שקיפות שעוזרת מאוד לפתור בעיות ולהבין טעויות שנעשו.

Firestore בלבד לשרת שהשתמשתי מ – Firebase, בסיס הנתונים בו אני משתמש של Firebase, הוא Firestore. Firestore הוא מסוג NoSQL. בנוסף, השימושתי ב – FireStorage, שם אני מאחסן את תמונות הפורפיל של המשתמשים, ובבסיס הנתונים יש רק קישור לתמונות אלו. כמו כן, השימושתי ב – Authentication של Firebase כך שאוכל לראות את המשתמשים שהתחברו כבר לאפליקציה, וגם כך אוכל להוציא את פרטייהם לבסיס הנתונים שלי.

אני בעצם משתמש ב – Firebase בתוך האפליקציה שלי בזמן אמת, כל הזמן מתבצעים עדכונים – קבלת הודעות, מחיקת הודעות, התנטקות וכדומה.

בנוסף, יש לי יכולת לנוהל את בסיס הנתונים לא דרך האפליקציה, אל דרך כניסה מהמחשב האישי שלי עם שם המשתמש והסיסמה שלי, כך שהמסכים של בסיס הנתונים שאראה בעמודים הבאים הם בעצם מסכי ניהול של בסיס הנתונים, והם זמינים רק עברו.

NoSQL

שלא בדומה לבסיס נתונים אחרים, SQL, אין מסודר מבנה טבלאי, אלא מסודר מבנה של "עץ". הנתונים מסודרים בדומה יותר לאר' שהם מסודרים על המחשב, תקין בתוך תיקייה ובתוכה נמצאים הנתונים.

ב – Firestore התיקיות נקראות Collections ובתוכן כל אחד יכול להיות כמה Documents, בתוך כל אחד יכולים להיות שדות מיידע שונים בהם נמצא חלק מהמידע, וגם מתוך כל Document אפשר להתחילה Collection חדש. וכך הלאה והלאה.

בעובדה עם DB יש 4 פעולות מרכזיות CRUD

CREATE – הוספה מידע

READ – קריאת מידע

UPDATE – עדכון מידע

DELETE – מחיקת מידע

כיוון שהשתמשתי ב – Packages על מנת לקרוא, לכתוב, לעדכן ולמחוק הודעות מבוסיס הנתונים, אני לא עבדתי שירותים עם NoSQL, لكن לא השתמשתי בפקודות שירות של NoSQL אלא בפונקציות הקשורות לפונקציות אלו.

הfonקציות עברו הפעולות השונות

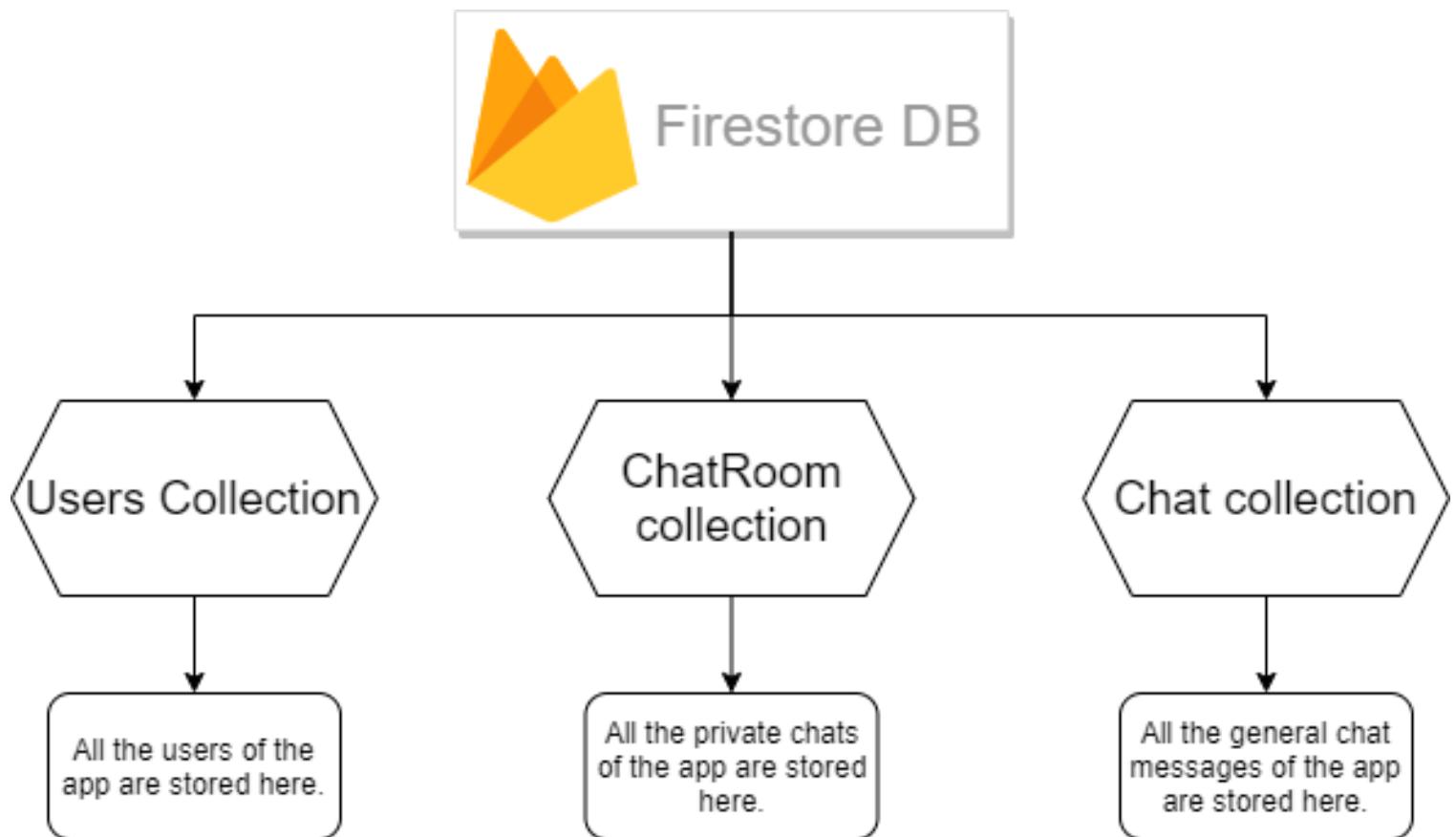
Firestore.instance.collection('collection name').add({map of data}) – Update / Create

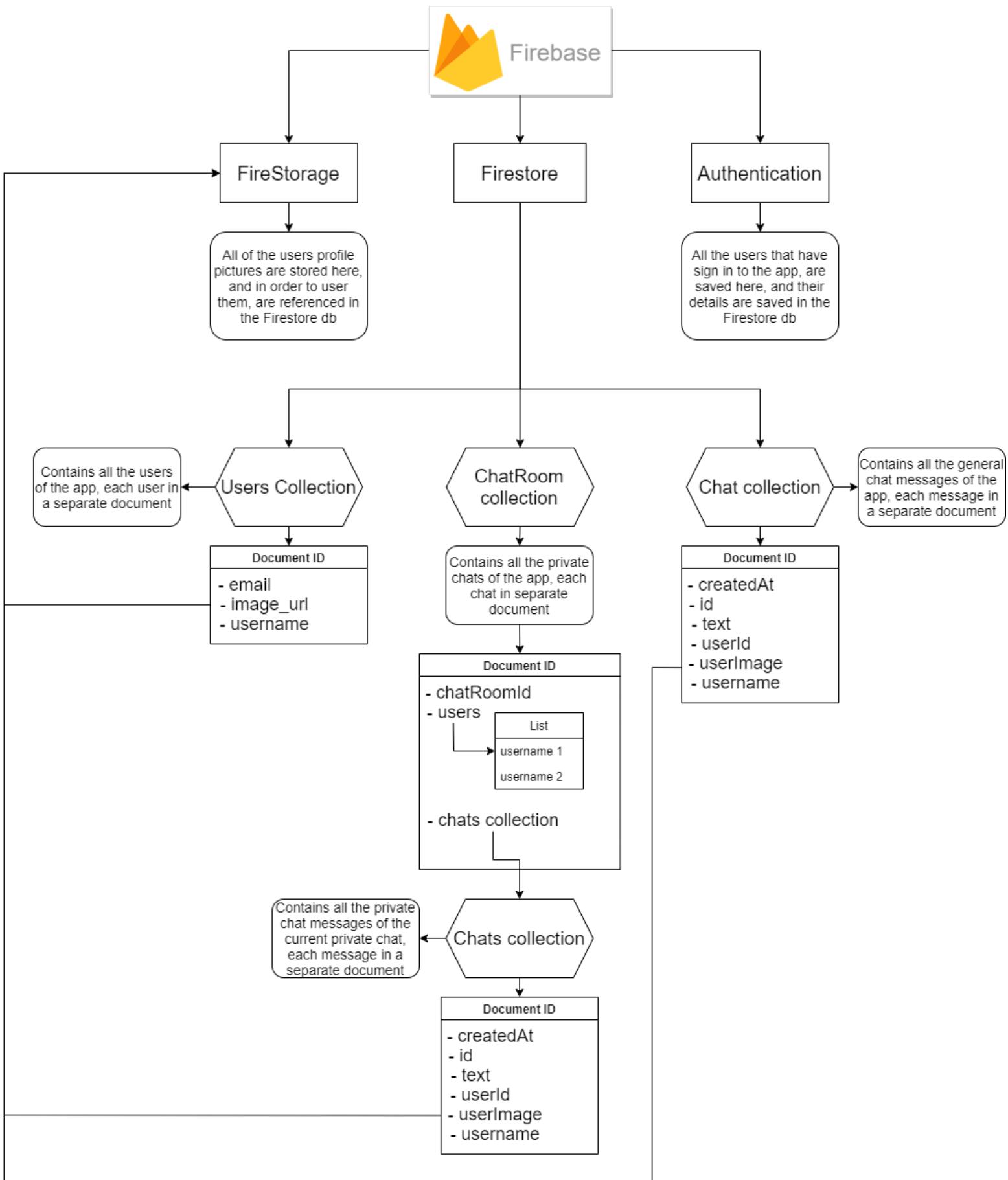
Firestore.instance.collection('collection name').document('document name').get() - Read

Firestore.instance.collection('collection name').document('document name').delete() - Delete

סכמה כללית של הышיות והקשרים בסיסי הנתונים

זהו סכמה בסיסית של איר הנתונים בסיסי הנתונים מאורגנים, בעמוד הבא תציג הסכמה המפורטת יותר.





תמונה והסבירים על בסיס הנתונים

המסכים של בסיס הנתונים שאני מראה כאן הם בעצם מסכי ניהול של בסיס הנתונים, והם זמינים רק עבורי.

Authentication

בחלק זה של ה – Firebase ניתן לראות רשימה של כל המשתמשים שהתחברו כבר לאפליקציה שלי וקצת פרטים עליהם.

The screenshot shows the Firebase Authentication console under the 'Users' tab. It displays a table with columns: Identifier, Providers, Created, Signed in, and User UID. There are four entries in the table:

Identifier	Providers	Created	Signed in	User UID
itay@gmail.com	✉️	13 May 2021	13 May 2021	2zb6h5C6uLTceQc9vpIwfpJ0qG72
tuti@gmail.com	✉️	13 May 2021	13 May 2021	3H1UE47TJCZ8Z0cqDGREyD3Pz023
ido@gmail.com	✉️	13 May 2021	14 May 2021	40tVaQy69TYUXX4ygBxU7xbkdd2
noa@gmail.com	✉️	13 May 2021	13 May 2021	Zg8FDw1EKeZ74ELXv5pxrqASB502

Below the table are buttons for 'Rows per page' (50), '1 - 4 of 4', and navigation arrows.

Users collection

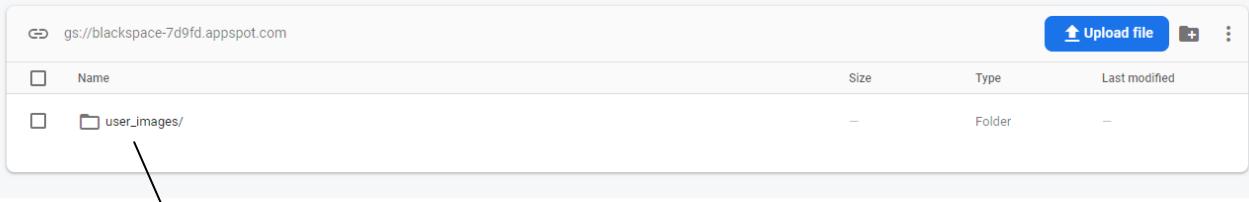
כאשר משתמש מתחבר למערכת, ראשית המידע הבסיסי שלו נשמר בחלק זה – Authentication ובנוסף כל המידע שלו, שאנו צריכים על מנת לחבר אותו אל האפליקציה, נשמר גם בחלק של Firestore – ב – Users Collection. כפי שמוסבר גם בסכמת הכללית של בסיס הנתונים.

The screenshot shows the Cloud Firestore console under the 'Data' tab. It displays a hierarchical view of documents in the 'users' collection:

- blackspace-7d9fd
 - + Start collection
 - chat
 - chatRoom
 - users
 - + Add document
 - 2zb6h5C6uLTceQc9vpIwfpJ0qG72
 - + Start collection
 - + Add field
 - email: "itay@gmail.com"
 - image_url: "https://firebasestorage.googleapis.com/v0/b/blackspace-7d9fd.appspot.com/o/user_Images%2F2zb6h5C6uLTceQc9vpIwfpJ0qG72?alt=media&token=b586c160-3937-45b2-bed4-ae06702af675"
 - username: "Itay"

FireStorage

כפי שהסבירתי לעיל, כאן נשמרות תמונות הפרופיל של המשתמשים כאשר הם נרשמים לאפליקציה בפעם הראשונה.

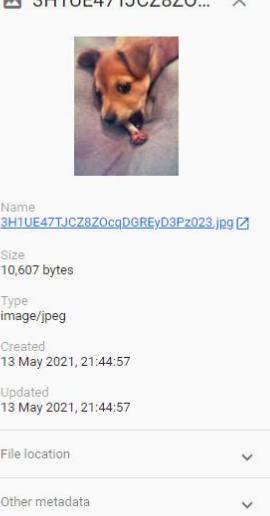


The screenshot shows the Google Cloud Storage interface for a project named 'blackspace-7d9fd.appspot.com'. A folder named 'user_images/' is visible. An arrow points from the 'user_images/' folder in the first screenshot to the same folder in the second screenshot, indicating the path where the uploaded image was stored.

Name	Size	Type	Last modified
2zb6h5C6uLTceQc9vplwfpJ0qG72.jpg	10.98 KB	image/jpeg	13 May 2021
3H1UE47TJCZ8Z0cqDGREyD3Pz023.jpg	10.36 KB	image/jpeg	13 May 2021
40tVaQy69TYUXx4ygBxU7xbykdd2.jpg	9.84 KB	image/jpeg	13 May 2021
Zg8FDw1EKeZ74ELXv5pxrqASB502.jpg	12.46 KB	image/jpeg	13 May 2021



3H1UE47TJCZ8Z0cqDGREyD3Pz023.jpg



Name: 3H1UE47TJCZ8Z0cqDGREyD3Pz023.jpg

Size: 10,607 bytes

Type: image/jpeg

Created: 13 May 2021, 21:44:57

Updated: 13 May 2021, 21:44:57

File location: [dropdown]

Other metadata: [dropdown]

Chat collection

כאן נשמרות כל ההודעות של הצ'אט הכללי של האפליקציה. כפי שמצוביר בסכמה הכללית של בסיס הנתונים.

The screenshot shows the Cloud Firestore interface with the following hierarchy:

- Project: Blackspace - 7d9fd
- Collection: chat
- Document ID: DBRAM68QreOB4TlzNbXD

The document details are as follows:

```

chatRoomId: "DBRAM68QreOB4TlzNbXD"
id: "DBRAM68QreOB4TlzNbXD"
text: "Hello everyone, I'm Itay!"
userId: "2zb6h5C6uLTeeQc9vplwfpJ0qG72"
userImage: "https://firebasestorage.googleapis.com/v0/b/blackspace-7d9fd.appspot.com/o/user_images%2F2zb6h5C6uLTceQc9vplwfpJ0qG72?alt=media&token=b586c160-3937-45b2-bed4-ae06702af675"
username: "Itay"
  
```

Cloud Firestore location: eur3 (europe-west)

ChatRoom collection

כאן נשמרות כל השיחות הפרטיות בין 2 משתמשים, כל שיחה ב – document נפרד. כפי שמצוביר בסכמה הכללית של בסיס הנתונים.

The screenshot shows the Cloud Firestore interface with the following hierarchy:

- Project: Blackspace - 7d9fd
- Collection: chatRoom
- Document ID: Itay_Ido

The document details are as follows:

```

chatRoomId: "Itay_Ido"
users:
  0: "Ido"
  1: "Itay"
  
```

Cloud Firestore location: eur3 (europe-west)

Cloud Firestore location: eur3 (europe-west)

מדריך למפתח

הסבר כללי על קודuproject

הקוד שכתבתי בפרויקט הוא הקוד של הלוקוח: הצגה במסך האפליקציה – "UI", התקשרות עם בסיס הנתונים, וקשרים בתוך האפליקציה עצמה, כמו תכונות מונחה עצמים. לא כתבתי קוד שרת בפרויקט.

את הפרויקט כתבתי בשפה Dart שהיא השפה בה נכתב Packages. והשתמשתי Flutter כאלו ואחרים עלי מנת לקצר את תהליך כתיבת הקוד ולהפוך אותו לכמה שיותר יעיל.

באפליקציה יש את החלק של הצגתה על המסך, ואת החלק של התקשרות עם השרת ובסיס הנתונים. אך שלא כמו ב בניית אתרים לדוגמה, בה יש חלוקה קוד מוגדרת עבור כל אחד מהתחומים – Html, CSS עבור עיצוב, JavaScript עבור backend – בפרויקט שלי כתבתי את כל הקוד כתוב בשפה אחת, Dart, ללא חלוקה מוגדרת לתחומים אלו.

שימוש ב - accessories של הטלפון

באפליקציה השתמשתי בצלמה של הטלפון, משום שבעת הרשמה לאפליקציה רציתי שככל משתמש יוסיף תמונה של עצמו, מה שהפוך את האפליקציה ליותר חברותית.

הסבר על הקוד של השימוש בצלמה נמצא בהמשך.

תכנות מונחה עצמים

תכנות מונחה עצמים התבטא בעיקר בכך שהשתמשתי ב – classes מובנים שmagics עם Dart ו – Widgets, מהם ירשתי ויצרתי myself שעצרו לי לכתוב את קוד הפרויקט שלי.

Flutter במאחורי בנייה שנקראות Widget. כל מחלוקת ב – Flutter היא בעצם Widget זהה או אחר כך שניתן להרכיב את ה – Widgets השונים אחד עם השני על מנת לבנות כל דבר שניתן לחשוב עליו.

Remote procedure call

כיוון שלא כתבתי קוד שרת הייתי צריך להשתמש בשרת אחר. עבור כך השתמשתי בשרת קים המותאם מראש ל – Firebase, Flutter. השימוש בשרת זה על ידי Remote procedure call - RPC .

RPC היא טכנולוגיה לתקשורת בין יישומים, כאשר יישום מקומי מבצע פעולה על יישום אחר מרוחק ממנו, כאילו הפעולות היו כ坨בות מקומיות אצל אותו יישום.

באמצעות RPC ניתן למשתמש תקשורת שרת ללקוח, כך שהלקוח מפעיל על השרת את הפונקציה הרצiosa תוך העברת הפרמטרים הנדרשים וממתין ממנו לקבלת התשובה.

אני משתמש ב – RPC בשבילו לקרוא לפעולות מוגדרות מראש ב – Firebase. כלומר, אני הלקוח ו – Firebase השירות. השימוש בזאת בעיקר על מנת לכתוב, לקרוא, לעדכן ולמחוק דברים מבוסיס הנתונים.

הסבר מפורט יותר על RPC - https://en.wikipedia.org/wiki/Remote_procedure_call

Asynchronous programming

מכיוון שככל הלקוחות לפעולות על השרת מתבצעות דרך האינטרנט, אין אפשרות לדעת متى כל פעולה תסתיים, ואם היא בכלל הצלילה. לכן, השימוש בתכנות אסינכרוני.

תכנות אסינכרוני הוא אמצעי לתוכנות מקביל בו יחידת עבודה פועלת בנפרד משרשרת היישום הראשי ומודיעה לשדרור המתקשר על השלמתה, כישלונה או התקדמותה.

לדוגמא, האפליקציה לא נתקעת כל פעם כאשר מבצעים פעולה שלוקחת זמן באינטרנט, אלא היא רצה במקביל. כאשר מקבלת תשובה על הצלחה או כישלון של הפעולה הא-סינכרונית ניתן לשנות את מצב האפליקציה בהתאם.

הסבר מפורט יותר על Asynchronous programming - [https://en.wikipedia.org/wiki/Asynchrony_\(computer_programming\)](https://en.wikipedia.org/wiki/Asynchrony_(computer_programming))

קישור לקוד הפרויקט ב – GitHub

<https://github.com/itsidohirsh/blackspace>

הסבר כללי על תיקיות הקבצים של הפרויקט

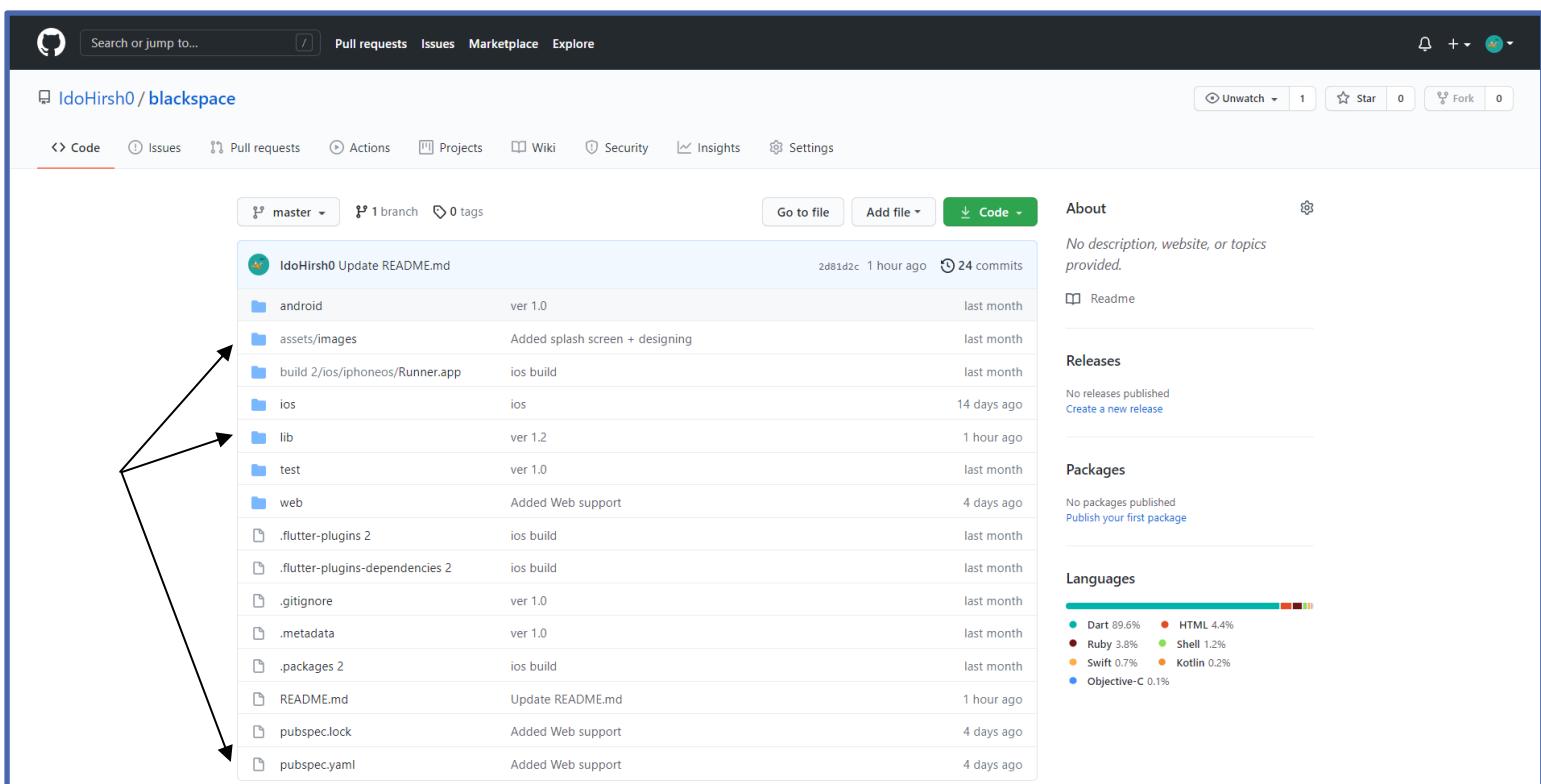
בתוך תיקיות הקבצים של הפרויקט, אותה ניתן להוריד מה – GitHub, ישנו מספר תיקיות וקבצים שונים.

בתיקייה `lib` נמצאים הקבצים בשפת Dart שניי כתבתו.

בתיקייה `assets` נמצאות תמונות מקומיות בהן השתמשתי בפרויקט.

בקובץ `pubspec.yaml` הגדרתי את ה – `packages` בהן השתמשתי בפרויקט.

בשאר התיקיות, נמצאים קבצי הקונפיגורציה ש – Flutter יצר כאשר יצרתי את הפרויקט, לצורך הריצה על מערכות הפעלה השונות וההגדרות של הפרויקט.



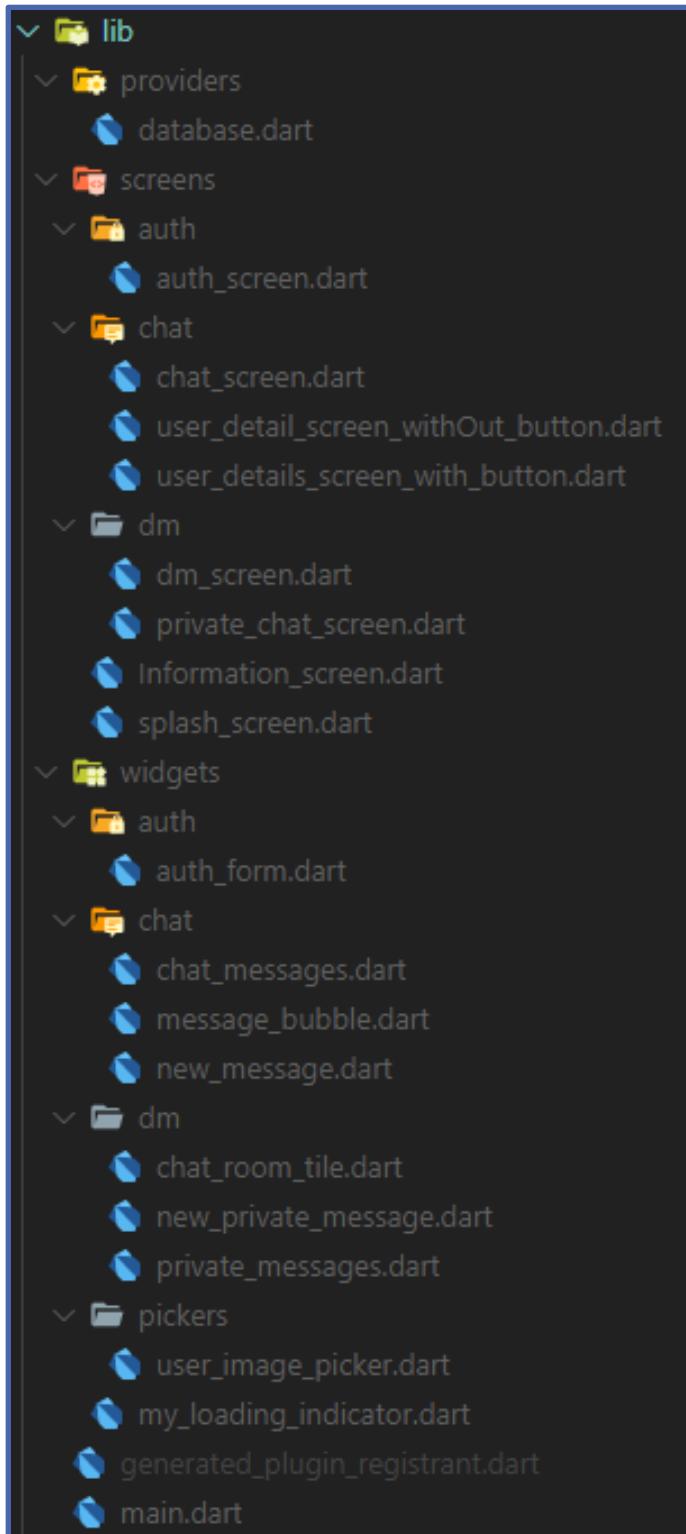
The screenshot shows the GitHub repository page for 'blackspace'. The 'Code' tab is selected. The main area displays a list of files and their commit history. Key files shown include `README.md`, `pubspec.yaml`, `pubspec.lock`, `build 2/ios/iphoneos/Runner.app`, `assets/images`, `lib`, and `test`. The right sidebar provides details about the repository, such as 'About' (no description), 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (Dart 89.6%, HTML 4.4%, Ruby 3.8%, Shell 1.2%, Swift 0.7%, Kotlin 0.2%, Objective-C 0.1%).

File	Description	Last Commit
<code>README.md</code>	Update README.md	2d81d2c 1 hour ago
<code>pubspec.yaml</code>	Added Web support	4 days ago
<code>pubspec.lock</code>	Added Web support	4 days ago
<code>build 2/ios/iphoneos/Runner.app</code>	ios build	last month
<code>assets/images</code>	Added splash screen + designing	last month
<code>lib</code>	ver 1.2	1 hour ago
<code>test</code>	ver 1.0	last month
<code>web</code>	Added Web support	4 days ago
<code>.flutter-plugins 2</code>	ios build	last month
<code>.flutter-plugins-dependencies 2</code>	ios build	last month
<code>.gitignore</code>	ver 1.0	last month
<code>.metadata</code>	ver 1.0	last month
<code>.packages 2</code>	ios build	last month
<code>android</code>	ver 1.0	last month
<code>READMD.md</code>	Update README.md	1 hour ago

הסבר על התקינה **lib**

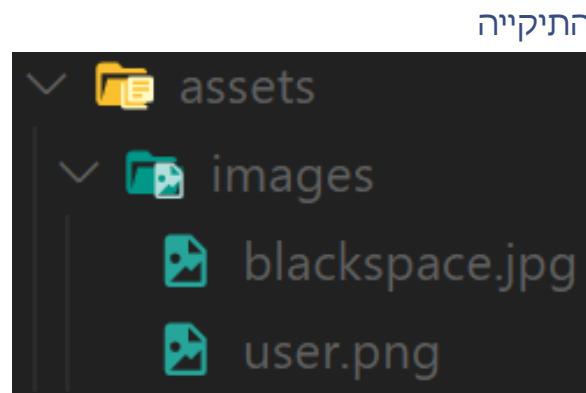
בתיק'יה זו נמצאים כל קבצי הפרויקט אוטם אני כתבתי בשפה Dart. כל הקבצים בתיק'יה זו הם קבצי הלוקוּן הכוללים הצגה במסך האפליקציה – "UI", התקשורת עם בסיס הנתונים, וקשרים כתוצאה האפליקציה עצמה, כמו תכונות מונחה עצמים.

קובץ ההרצה של האפליקציה נקרא – main.dart



הסבר על התיקייה assets

בתיקייה זו נמצאות התמונות המקומיות בהן השתמשתי בפרויקט. התמונה למסך הטעינה, ותמונה ברירת המחדל לייצור משתמש חדש.



התמונות

Blakcspace.jpg



user.png



הסבירים על קבצים, פונקציות וחלקה קוד חשובים הגדרת **Packages** לשימוש בפרויקט

שם הקובץ – pubspec.yaml

מיקומו - lib → pubspec.yaml

תוכנו – בקובץ זה מפורטוות כל ה – Packages בהן השתמשתי בפרויקט. בנוסף בקובץ זה מתבצע הקישור בין התקינה assets לקוד הפרויקט, כך שאוכל להשתמש בתמונות אלו באפליקציה שלי. קובץ זה מוגדר על ידי flutter כקובץ בו מגדירים את ה – packages בהן משתמשים בפרויקט.

<https://pub.dev/packages> לחשטמש ניתן למצוא כאן - Packages

```
{
  name: blackspace
  description: A new Flutter project.

  publish_to: 'none'

  version: 1.0.0+1

  environment:
    sdk: "≥2.7.0 <3.0.0"

  dependencies:
    flutter:
      sdk: flutter

    firebase_core: ^1.2.0
    firebase_auth: ^1.2.0
    cloud_firestore: ^2.2.0
    firebase_storage: ^8.1.0

    image_picker: ^0.7.5+3

    cupertino_icons: ^1.0.2

  flutter:
    uses-material-design: true

  assets:
    - assets/images/
}
```

כל ה Packages –
בها השתמשתי

קישור לתקינה assets

קובץ שימוש במכשירמה של הטלפון

בעיני זהו קובץ מעולה להדגמה איך Flutter עובד, משומם שהוא כולל גם logic וגם GUI (Graphic User Interface).

שם הקובץ – user_image_picker.dart

מיקומו – lib → widgets → pickers → user_image_picker.dart

תוכן – קובץ זה מבצע את בחירת התמונה של המשתמש בעט יצירה משתמש חדש לאפליקציה. הוא כולל בתוכו 2 classes – `UserImagePicker` ו- `_UserImagePickerState`.

ב – Dart ו – Flutter על מנת ליצור חישוב חדש, צריך לרשף (תכנים מונחים עצמים) מאחד מшибיו classes במשתמשים בו כאשר רוצים ליצור widget שלא משנה דבר על המסך. כלליים: 1. StatelessWidget – משתמשים בו כאשר רוצים ליצור widget שמשנה דבר על המסך.

2. StatefulWidget – משתמשים בו כאשר רוצים ליצור widget שמשנה משהו על המסך.

במקרה הזה השתמשתי ב – StatefulWidget מכיוון שרוצה להציג את התמונה הנבחרת בסופו של דבר על המסך, ובכך אשנה את מה שהוא שמוצג בו.

חלקו הראשון של הקובץ הוא ה – Logic

```

1 import 'dart:io';
2 import 'package:flutter/material.dart';
3 import 'package:image_picker/image_picker.dart';
4
5 class UserImagePicker extends StatefulWidget {
6   UserImagePicker(this.imagePickFn);
7
8   final void Function(File pickedImage) imagePickFn;
9
10  @override
11  _UserImagePickerState createState() => _UserImagePickerState();
12 }
13
14 class _UserImagePickerState extends State<UserImagePicker> {
15   File _pickedImage;
16
17   void _pickImage() async {
18     final picker = ImagePicker();
19     final pickedImage = await picker.getImage(
20       source: ImageSource.camera,
21       imageQuality: 50,
22       maxWidth: 150,
23     );
24     var pickedImageFile;
25
26     try {
27       pickedImageFile = File(pickedImage.path);
28     } catch (error) {
29       print(error);
30     }
31
32     setState(() {
33       _pickedImage = pickedImageFile;
34     });
35     widget.imagePickFn(pickedImageFile);
36   }
}

```

החלק העיקרי ב – logic בקובץ זה הוא הפעולה `pickImage()` (מכיוון שאין

בפערונותיות זו (מכיוון שאין אפשרות ללחוץ כמה זמן "ייקח למשתמש לבחור תמונה, אם בכלל) מתבצעת בחירת התמונה של המשתמש מהמכשירמה.

בשביל לבצע פעולה זו השתמשתי ב – package `image_picker` שנקלע את רמת הקוד.

בשורה 19 – 23, אני מפעיל את הפעולה מה – package `image_picker` שנקראת `getImage()`.

פעולה אסינכרונית זו פותחת את מצלמת המכשיר (אחרי בקשת גישה למצלמה מהמשתמש) ומחייבת שיצלם תמונה.

אם בחר תמונה, אני עובר לשורה 27 ושומר את התמונה שלו כ – `File` widget `_pickedImage`, שאני לאחר מכן אעביר לבסיס הנתונים.

אם לא בחר תמונה או שתהיליך זה לא צלח (חלק מה – `try and catch`) מדפס את השגיאה שנוצרה. ולאחר מכן קורא לפעולה שמעבירה את התמונה ל – `class Firebase`. ששמור את התמונה ב `Firestore`.

חלוקת השני של הקובץ הוא – GUI

חלק זה בקוד הוא החלק שקובע מה יוצג על מסך המשתמש. והוא נמצא ב פעולה (build), פעולה היא פעולה בסיסית של widget ב – flutter וכל מה שרוצים להציג על המסך צריך להמצא בתוכה.

השתמשתי כאן בכמה widgets בסיסיים שמאגים עם flutter:

1. Column – מאפשר לשים כמה widgets שונים אחד ליד השני (הראשינה [children]).
2. CircleAvatar – מאפשר לשים תמונה או רקע אחר בתוך מסגרת עגולה.
3. FileImage – מאפשר לשים תמונה מסווג File (כמו שקיבלו מהפעולה () _pickImage תמונה מסווג File).
4. AssetImage – מאפשר לשים תמונה שנמצאת מקומית בקבצי הפרויקט.
5. FlatButton – כפתור על המסך, בעל הפרופרטי onPressed אותו אפשר להגיד לפעולה אותה רצחים להריץ כאשר נלחץ הכפתור.
6. Icon – מאפשר להציג חסוך כלשהו על המסך.
7. Text – מאפשר לכתוב טקסט על המסך.

בעצם מוצג עיגול ומתחתיו כפתור 'Add Image', כאשר אם המשתמש כבר בחר תמונה היא תוצג בעיגול, ואם לא תוצג תמונה ברירת המחדל.

כאשר נלחץ הכפתור, נקריאת הפעולה _pickImage אותה הסברתי קודם.

```

38 @override
39 Widget build(BuildContext context) {
40   return Column(
41     children: [
42       CircleAvatar(
43         radius: 50,
44         backgroundImage: _pickedImage != null
45             ? FileImage(_pickedImage)
46             : AssetImage('assets/images/user.png'),
47         backgroundColor: Colors.white,
48       ), // CircleAvatar
49       FlatButton.icon(
50         textColor: Theme.of(context).primaryColor,
51         icon: Icon(Icons.image),
52         label: Text('Add Image'),
53         onPressed: _pickImage,
54       ) // FlatButton.icon
55     ],
56   ); // Column
57 }
58 }
59 }
```

החלק מספק ההרשמה אותו קוד קובץ זה, user_image_picker.dart מייצג.

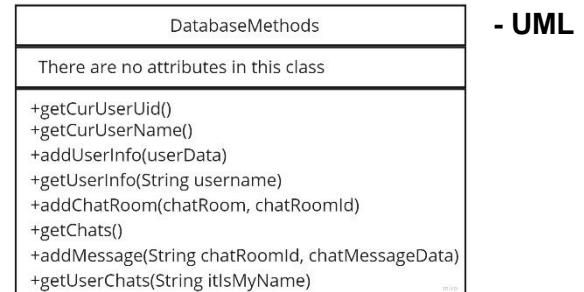


התקשרות עם השירות ובסיס הנתונים

שם הקובץ - database.dart

מיקומו – lib → providers → database.dart

תוכנו – מכיל את ה – DatabaseMethods class, שמכילה שאלת כל הפונקציות בהם השתמשתי לתקשורת עם השירות.



הסבר על פונקציות חשובות

ההסבר על כל פונקציה רשום בהערה בתוך הקוד, מסומן כאן ב - ←

```

File Edit Selection View Go Run Terminal Help
EXPLORER main.dart database.dart M X
lib > providers > database.dart > DatabaseMethods
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:firebase_auth/firebase_auth.dart';
3
4 class DatabaseMethods {
5     // Returns the current connected user, user id ←
6     Future<String> getCurUserId() async {
7         final user = await FirebaseAuth.instance.currentUser();
8         return user.uid;
9     }
10
11    // Returns the current connected user, user name ←
12    Future<String> getCurUserName() async {
13        final uid = await getCurUserId();
14        final doc =
15            await Firestore.instance.collection('users').document(uid).get();
16        return doc.data['username'];
17    }
18
19    // Adds user information to the user in the users collection in the db ←
20    Future<void> addUserInfo(userData) async {
21        Firestore.instance.collection("users").add(userData).catchError((e) {
22            print(e.toString());
23        });
24    }
25
26    // Returns the info of a user by its username ←
27    Future<Map<String, dynamic>> getUserInfo(String username) async {
28        final userDoc = await Firestore.instance
29            .collection('users')
30            .where('username', isEqualTo: username)
31            .getDocuments();
32        return userDoc.documents.single.data;
33    }
  
```

The screenshot shows the Visual Studio Code interface with the "database.dart" file open. The code implements a `DatabaseMethods` class with several asynchronous methods for interacting with a Firestore database. Annotations with arrows pointing to specific lines of code provide explanations for each function's purpose:

- `// Returns the current connected user, user id` points to the `getCurUserId()` method.
- `// Returns the current connected user, user name` points to the `getCurUserName()` method.
- `// Adds user information to the user in the users collection in the db` points to the `addUserInfo(userData)` method.
- `// Returns the info of a user by its username` points to the `getUserInfo(String username)` method.

database.dart - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER ... database.dart M X
OPEN EDITORS lib > providers > database.dart > DatabaseMethods
BLACKSPACE
> .dart_tool
> .idea
> android
> assets
> build
> build 2
> ios
lib
> providers
|   database.dart M
> screens
> widgets
generated_plugin_registrant.d...
main.dart
test
web
flutter-plugins
flutter-plugins 2
flutter-plugins-dependencies
flutter-plugins-dependencies 2
.gitignore
.metadata
.packages
.packages 2
.blackspace.iml
pubspec.lock
{ pubspec.yaml
README.md

34 | // Creates a new chat room for a private chat two users have started
35 | Future<void> addChatRoom(chatRoom, chatRoomId) async {
36 |   await Firestore.instance
37 |     .collection("chatRoom")
38 |     .document(chatRoomId)
39 |     .setData(chatRoom)
40 |     .catchError((e) {
41 |       print(e);
42 |     });
43 |
44 |
45 |
46 | // Returns the chat messages of a current chat room by its room id
47 | getChats(String chatRoomId) async {
48 |   return Firestore.instance
49 |     .collection("chatRoom")
50 |     .document(chatRoomId)
51 |     .collection("chats")
52 |     .orderBy('time')
53 |     .snapshots();
54 |
55 |
56 | // Adds a message to a certain chat room
57 | Future<void> addMessage(String chatRoomId, chatMessageData) async {
58 |   await Firestore.instance
59 |     .collection("chatRoom")
60 |     .document(chatRoomId)
61 |     .collection("chats")
62 |     .add(chatMessageData)
63 |     .catchError((e) {
64 |       print(e.toString());
65 |     });
66 |
67 |

File Edit Selection View Go Run Terminal Help
EXPLORER ... database.dart M X
OPEN EDITORS lib > providers > database.dart > ...
BLACKSPACE
> .dart_tool
> .idea
> android
> assets
> build
> build 2
> ios
lib
> providers
|   database.dart M
> screens
> widgets
generated_plugin_registrant.d...
main.dart
test
web
flutter-plugins
flutter-plugins 2
flutter-plugins-dependencies
flutter-plugins-dependencies 2
.gitignore
.metadata
.packages
.packages 2
.blackspace.iml
pubspec.lock
{ pubspec.yaml
README.md

68 | // Returns all the current user private chats
69 |
70 | Stream<QuerySnapshot> getUserChats(String itIsMyName) {
71 |   return Firestore.instance
72 |     .collection("chatRoom")
73 |     .where('users', arrayContains: itIsMyName)
74 |     .snapshots();
75 |
76 |

```

רפלקציה

תחושים מהעבודה על הפרויקט

העבודה על הפרויקט הייתה מהנה מאוד עבורו. עצם כך שהצלחתו למש את מה שרצתי, אפליקציה שראיתי בסדרה בטלוויזיה בגרסה משלו, גורם לי לאושר ולתחשות סיפוק רבה.

麥意oun שהשנה הייתה שנות קורונה, בה למדנו בעיקר מרוחק ב – zoomים, היה לי הרבה יותר זמן לעבוד על הפרויקט ממה שחשבתי שהיא יהיה. במקום להיות כל יום בבית הספר 8 שעות, בהפסכות של הלמידה מרוחק יכולתי לעבוד על הפרויקט, וגם ימי הלמידה היו יותר קצרים אז היו לי יותר שעות ביום מלכתחילה. אני חושב שלא תקופה הזאת, לא הייתה יכול לפתח את מה שפיתחתי כיום.

כלים שקיבلت ואני איתם המשך

בעבודה על הפרויקט קיבלתי המונח. למדתי שאני מסוגל ללמידה דברים רק בכוחות עצמי, לחזור על נושא שככל לא הכרתי לפני כן ולהצליח לפתח דברים מדהימים בו. למדתי על עצמי שאם אני רוצה, יש לי מוטיבציה אידירה לעובדה, לפני הפרויקט, אני לא חשב שהשאעתני כמו שעות צדו גדולת על מה שהוא למדוי, ימים כליאות עבדתי על הפרויקט וזהו דבר שאני מאוד שמח עליו, כי הוא גורם לי להבין שאני מסוגל לכל מה שאני רק ארצה, אני רק צריך לרוץ זאת.

עוד kali שקיבلت מהפרויקט הוא شيtopic פעללה, כאשר נתקעתי בעיה שלא הצלחתי לפתור, פניתי לחבר לי לכתבה שעסקו בנושא או בנושא דומה, ואלו עזרו לי להגיא לפתרון.

אתגרים שעמדו בפניי במהלך הפרויקט

בנוסף לאתגר של למידת נושא חדש לגמרי בכוחות עצמי, כפי שתבתי בתחילת, במהלך הפרויקט עלו עוד כמה אתגרים.

ראשית, אני לוקח חלק בתוכנית MAX, גם אליה דיברתי בתחילת, והוא היה לי קשה להחליט בין אם להמשיך איתם או אולי לא להיות עם פרויקט גמור בסופו של דבר, או להיפרד מהם בגיןטיים על מנת לעבוד על הפרויקט שלי, Blacksphere. לאחר התחבויות רבות, ועזרה הרבה של ניר, החלטתי לעבוד על הפרויקט שלי.

בנוסף, רציתי לפתח את האפליקציה שלי ל – Android ו – iOS במקביל, لكن גם בחרתי לעבוד עם Flutter.ammenה ההוספה של iOS לבסיס הנתונים, ו קישור בין מכשירי ה iPhone הוביל להרבה בעיות שלקחו ממנה שעות של ניסיונות על מנת לפתור, אך בסופו של דבר הצלחתי לעשות זאת והאפליקציה רצתה על שתי הפלטפורמות.

מסקנות מהפרויקט

מסקנתה העיקרית מהפרויקט היא שאהבתי לעבוד עליו! נהנית ללמידה עצמי דברים חדשים, וללמידה על עצמי דברים חדשים.

מסקנה נוספת היא שאהוב לעבוד בשיתוף פעולה ושבועתיד נראה איךנה לעבוד כחלק מצוות. כאשר נתקלתי בעיה שלא הצלחתי לפתור, גם לא בעזרתו של האינטרנט, פניתי לחבר לי לכתבה שעסקו בנושא או בנושא דומה ועזרתי בהם.

מה הייתה עשו אחרת אילו הייתה מתחילה היום

אילו הייתה מתחילה לעבוד על הפרויקט היום קודם כל הימי מפזר את העבודה על ספר הפרויקט לאורך כל העבודה על הפרויקט. את ספר הפרויקט הכנתי ורק לאחר סיום העבודה על הפרויקט וידעתי אם הייתי עשו אותם במקביל, גם הייתי חוסך לעצמי זמן בהליצר בדברים שעשית בעבר, וגם הכתיבה של הדברים לדעתן הייתה עוזרת לי לפתח את הפרויקט עצמו.

בנוסף הייתי אומר לעצמי בשלב מוקדם יותר, שאני נראה לא אסימ את הפרויקט עם MAX בזמן, כך היה מתפנה לי יותר זמן לעבוד על הפרויקט הזה, ואולי הייתי יכול לשפר אותו אפילו יותר, ולהוסיף את הפיצרים שלא הספקתי להוציא.

מה היה יכול להפוך את עבודתי ליעילה יותר

אילו הייתי מגיש לניר את הפרויקט לעיתים יותר תכופות (וכל פעם מתקדם קצת פחות), הכולמר כל שבוע בערך במקום כל שלושה שבועות, עצם זה שיש דד-ליין שzier להגשים בו משהו, היה נותן לי עוד יותר מוטיבציה לעבודה, ואילו הייתי מסיים את הפרויקט עוד קודם.

כמו כן, כפי שציינתי לעיל, אילו הייתי משלב את כתיבת ספר הפרויקט יחד עם עשיית הפרויקט אולי הדבר היה מייעל את העבודה, והייתי מנצל את זמני טוב יותר.

תכונות שהייתי רוצה להוסיף לפרויקט

ישן מספר תכונות שהייתי רוצה להוסיף לפרויקט אם היה לי את הזמן לכך. ראשית, ארצתה לאפשרות לצירמת קבוצות של מספר משתמשים באפליקציה, מלבד השיחות הפרטיות והצ'אט הכללי.

כמו כן, הייתה רוצה להוסיף את האפשרות לשלח תמונות בכל אחד מהמצאים.

בנוסף, הייתה רוצה להוסיף את האפשרות לשחזר סיסמה ולשנותה, לשנות את שם המשתמש, האימייל ואת תמונה הפרופיל של המשתמש.

לסיכום, למראות כל האתגרים והקשיים בדרכ, הנהני מואוד לעבוד על הפרויקט! למדתי ממנו המונע, ולקח ממנו הרבה דברים לח' האישים. ואני מקווה שכל מי שيشתמש באפליקציה שלי ייהנה ממנה לפחות כמו שאני הנהני 😊

ביבליוגרפיה

- ניר סליקטר - המורה שלי (;
- <https://firebase.google.com> - Firebase
- <https://flutter.dev> - Flutter.dev
- <https://github.com> - GitHub
- <https://stackoverflow.com> - Stack Overflow
- <https://www.udemy.com> - Udemy
- <https://www.youtube.com> - YouTube

נספחים

קישור לספר פרויקט זה ב – **GitHub** –

<https://github.com/itsidohirsh/blackspace-project>

תעודה מ – **Flutter** על סיום קורס ב – **Udemy** –

Certificate of Completion

*This is to certify that **Ido Hirsh** successfully completed 41.5 total hours of **Flutter & Dart - The Complete Guide [2021 Edition]** online course on Feb. 9, 2021*

Academind by Maximilian Schwarzmüller

Academind by Maximilian Schwarzmüller, Instructor

Maximilian Schwarzmüller

Maximilian Schwarzmüller, Instructor



Certificate no: UC-7712618f-80c8-4d87-b571-7ef0875a058d
Certificate url: ude.my/UC-7712618f-80c8-4d87-b571-7ef0875a058d
Version 3

#BeAble

סרטונים של תהליך למידה

בקישור הנ"ל ניתן למצוא סרטונים של תהליך הלמידה שלי לפROYיקט, סרטוני הסבר על אפליקציות, וסרטונים שלי עובד על הפROYיקט.

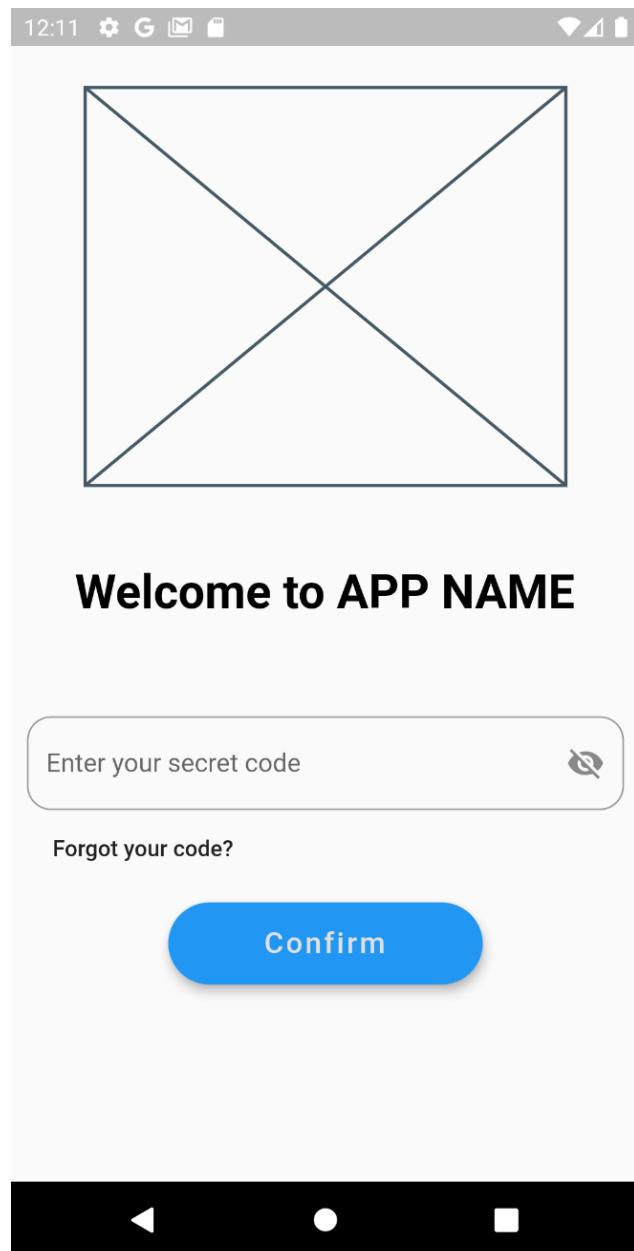
קישור: -
<https://drive.google.com/drive/folders/1-v18DgIvWqSaLbt27HRZ5W8qNwRiHOD4?usp=sharing>

אפליקציות שונות שבניתי במלר הלמידה

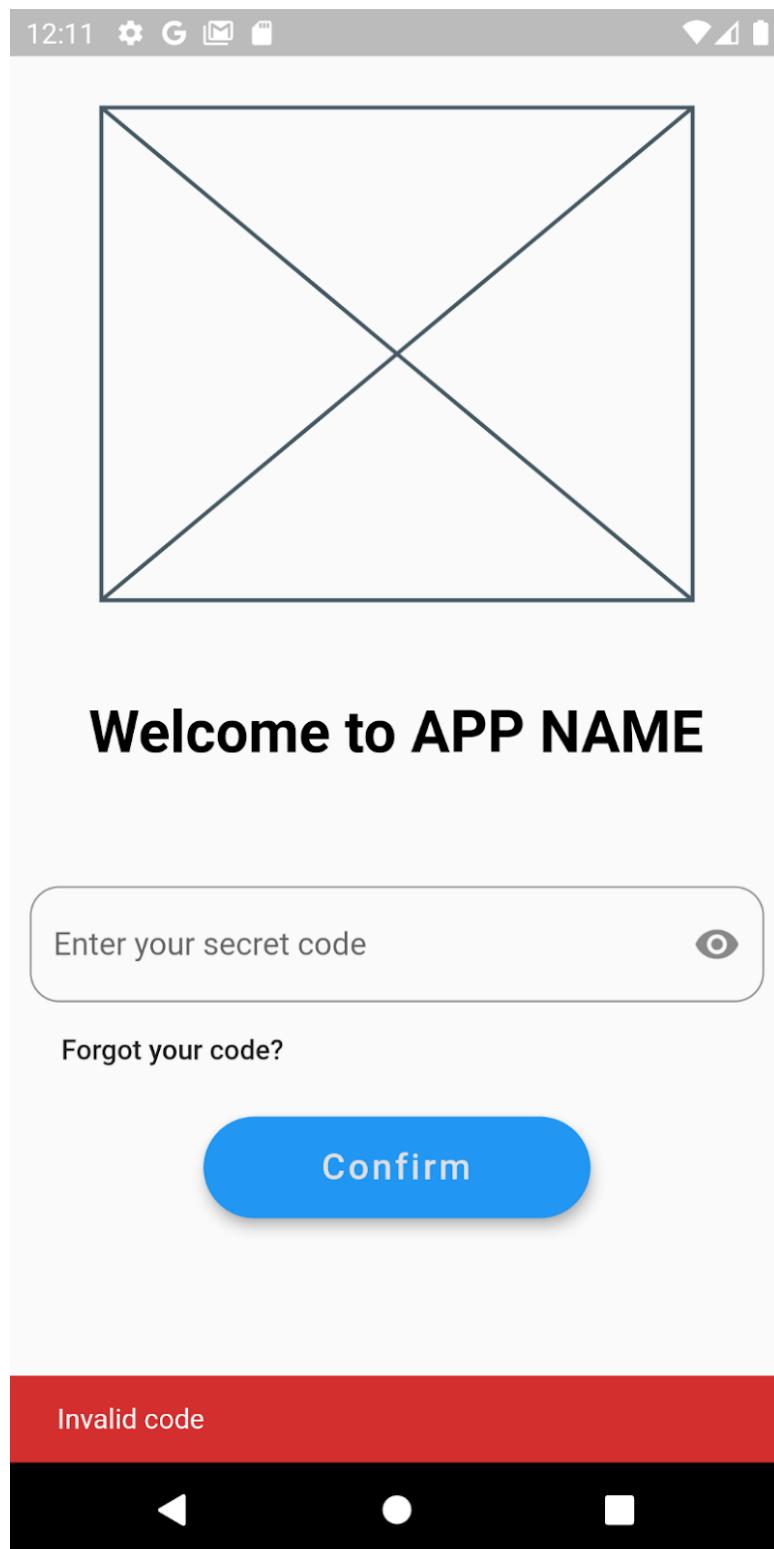
אפליקציה MAX – מפרויקט MAX

התמונה והסבירים על האפליקציה זו כוללים כאן משומש שכבר עשית הסברים אלו בעבר כאשר עבדתי על הפרויקט עם MAX, אפליקציית נסיעות משותפות, Carpool. עוד לפני שידעת שהפרויקט הסופי שלי יהיה .Blackspace

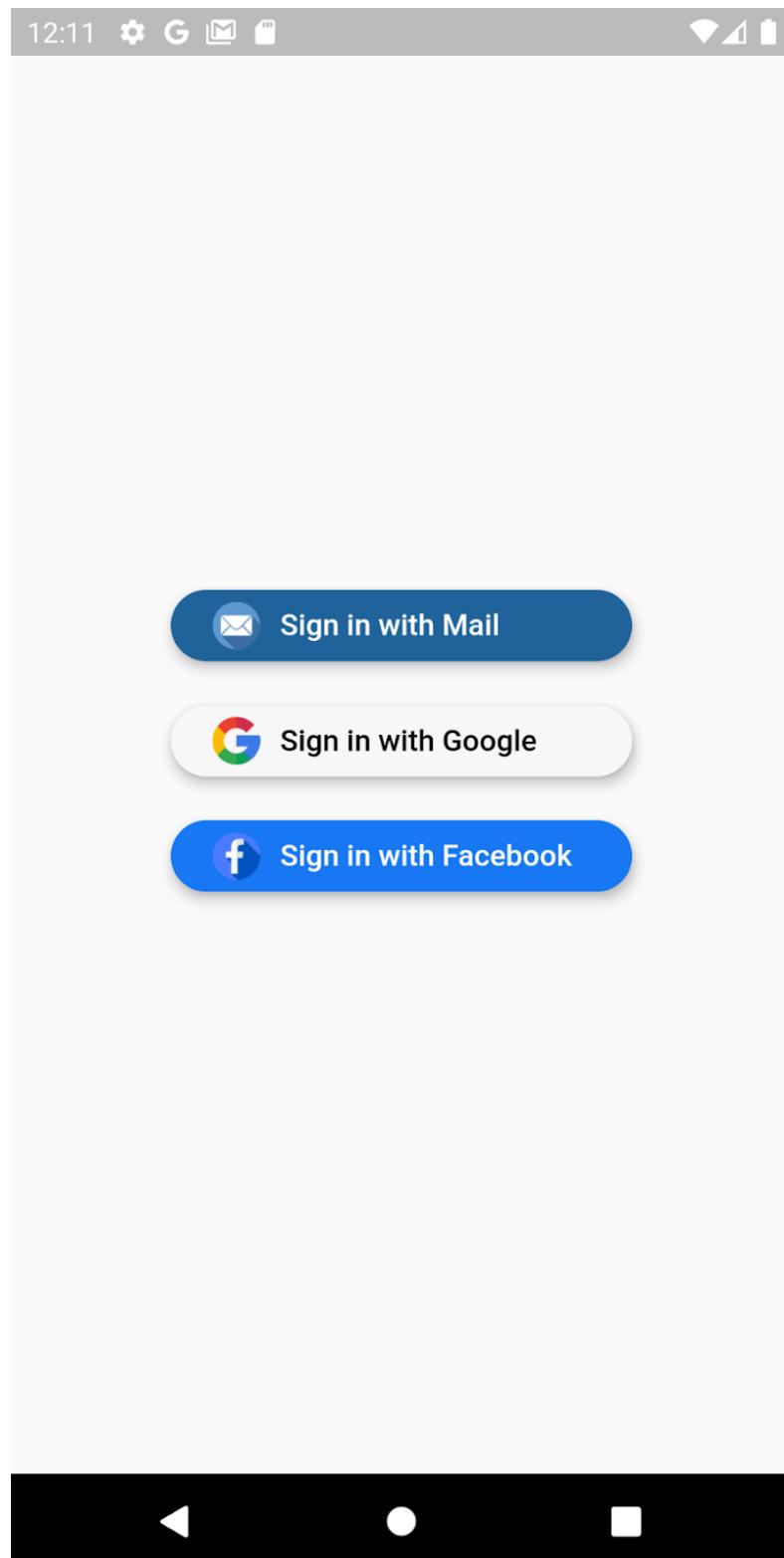
מסך ראשון בו מזינים את הקוד הסודי שאנו בתוכו אפליקציה ניתן לאותו מועד כדי שייתן לעובדים שלו. אך רק אנשים בעלי גישה לקוד זה יהיו יכולים להתחבר למערכת ותיזכר סוג של חמהה בה כולם מכירים את כולם.



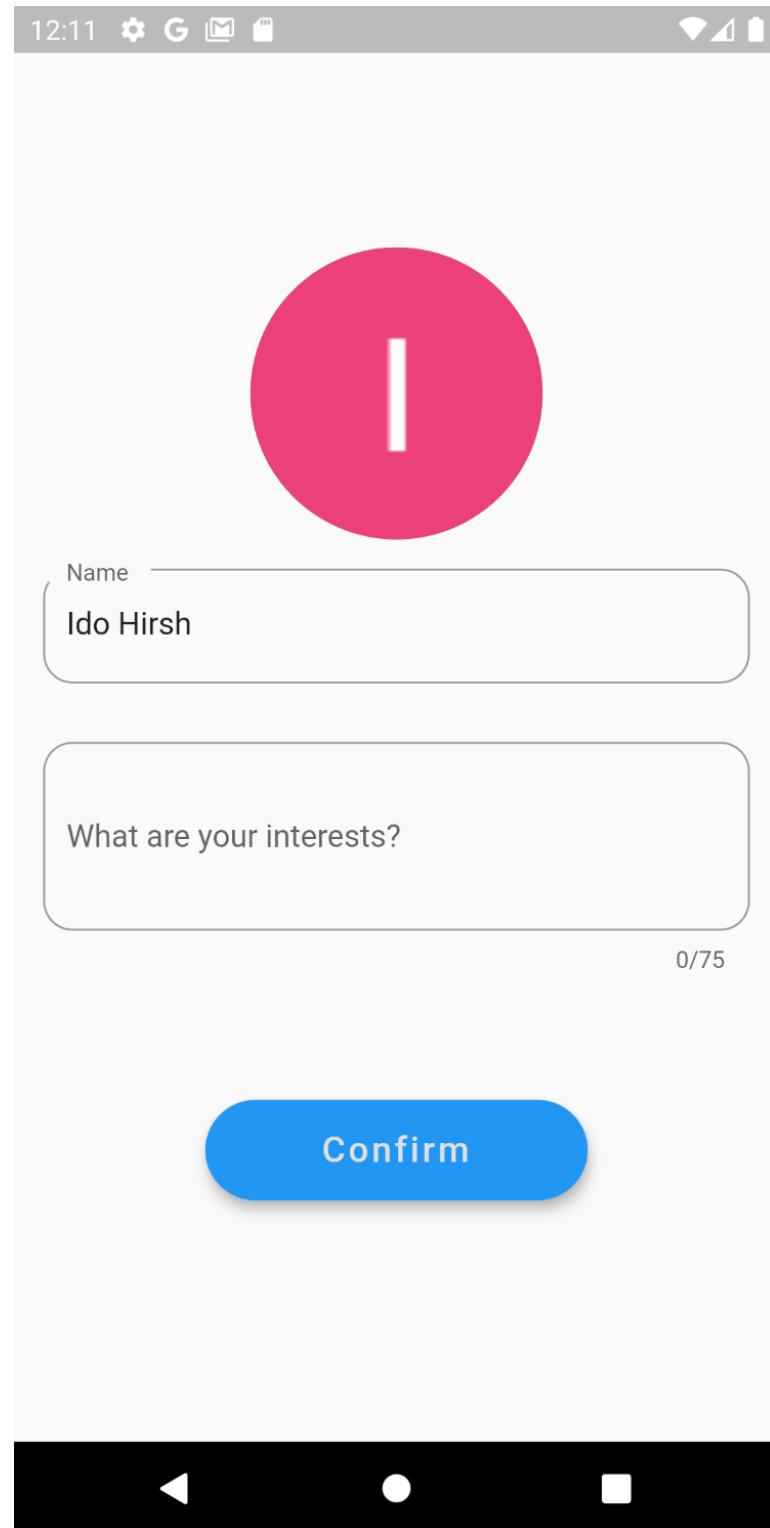
כמובן שיש בדיקה של הקלט אל מול databases בו מאוחסנים הקודים הסודיים. אם הוזן קוד שגוי תופיע הודעה:



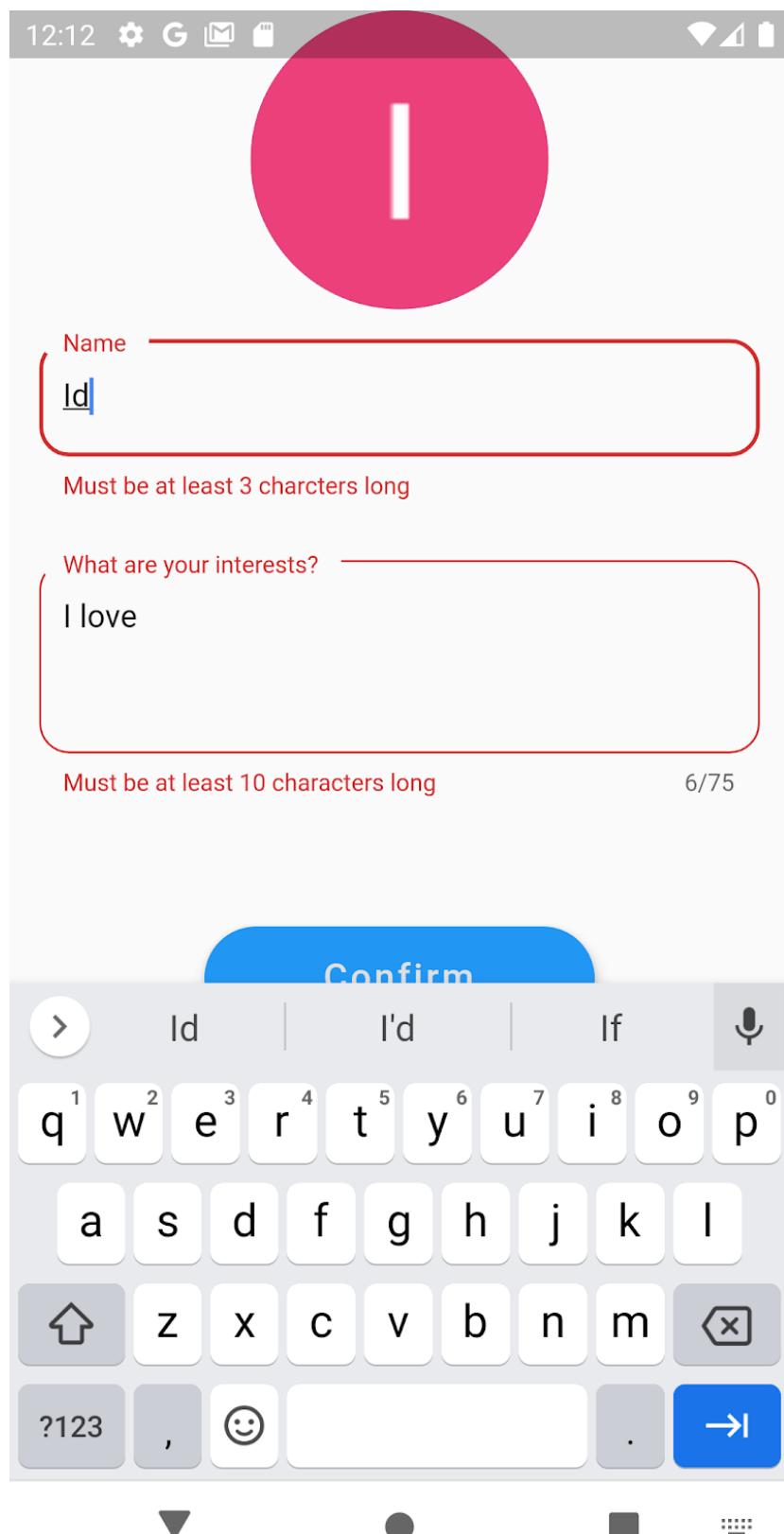
לאחר ההתחברות עם הקוד הסודי של האפליקציה שלנו, יש אופציה ל-3 סוגי התחברויות, גугл פייסבוק
ואימייל וויסמה.
בינתיים אני עובד רק עם ההתחברות של גול.



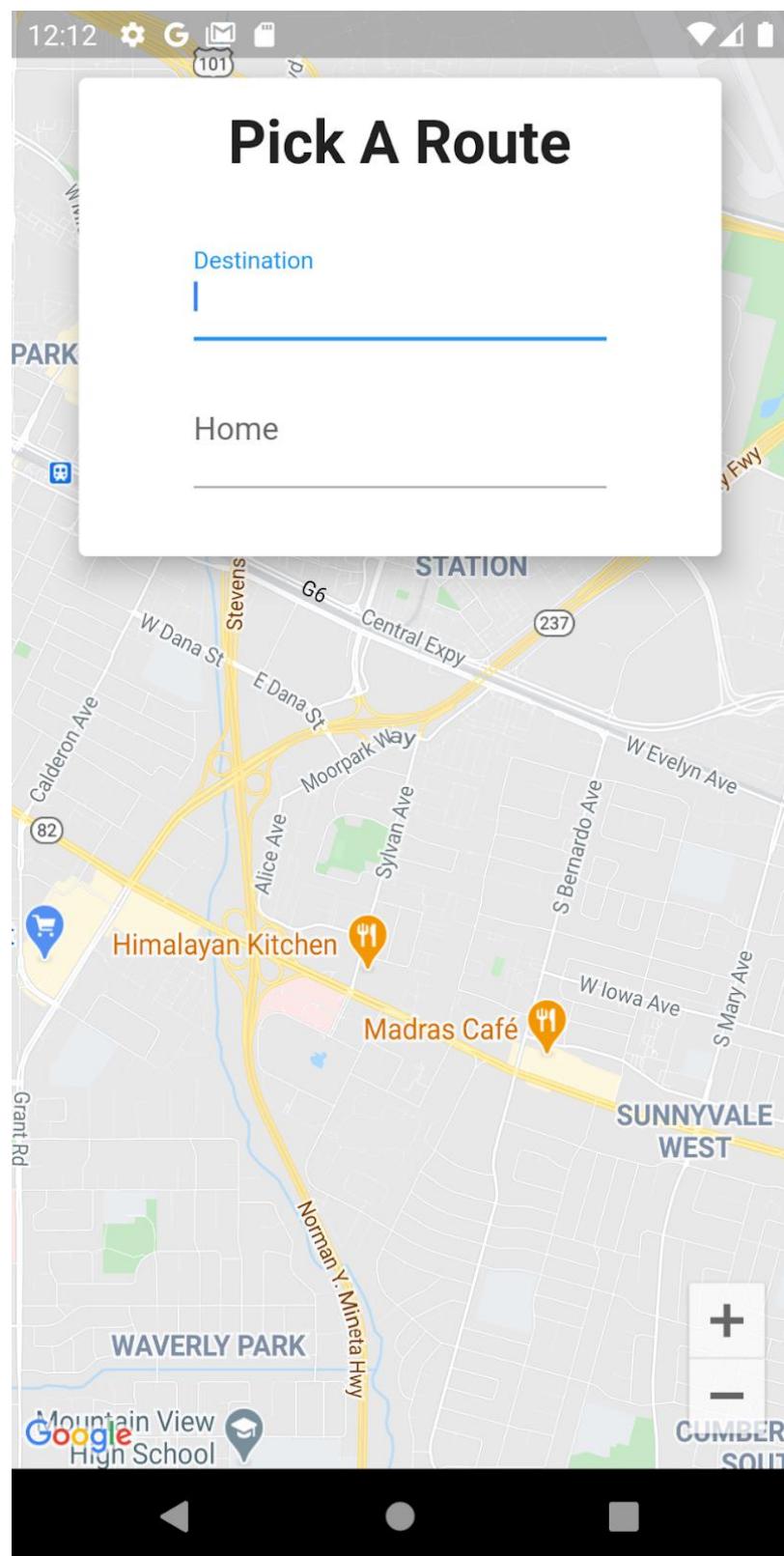
לאחר ההתחברות עם גוגל, אני לוקח את פרטי המשתמש מגוגל, כדוגמת שם משתמש, תמונה, אימייל ועוד. אבל ניתן למשתמש את האפשרות לשנות את הפרטים האלה באפליקציה. גם מקבל מהמשתמש תיאור קצר על עצמו, על מנת שഫופיל של כל אחד יהיה לפחות איזשהו משפט קצר. כך יהיה אפשר להכיר קצת יותר את הבן אדם.



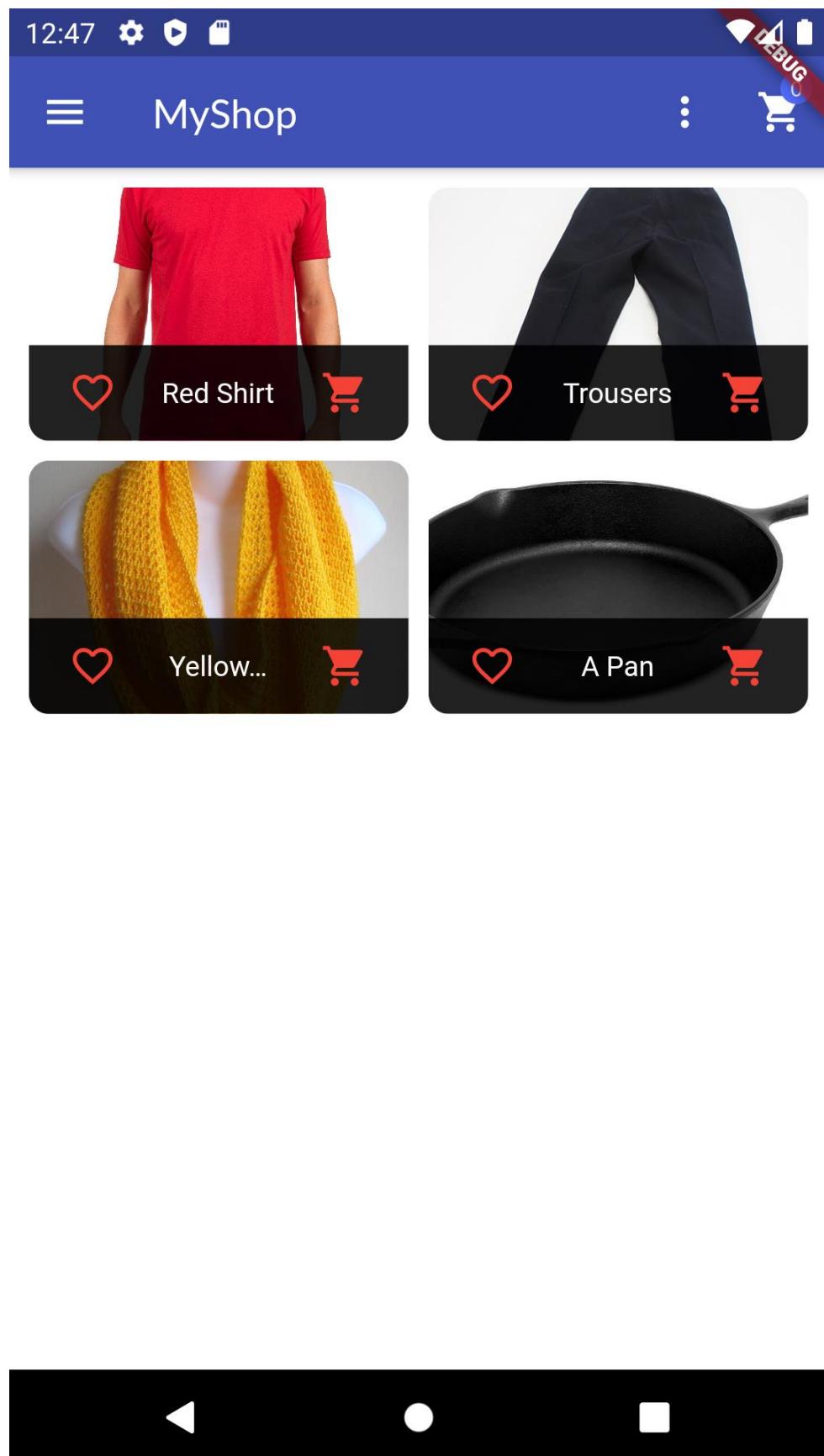
כמובן שיש בדיקה של הקלט...

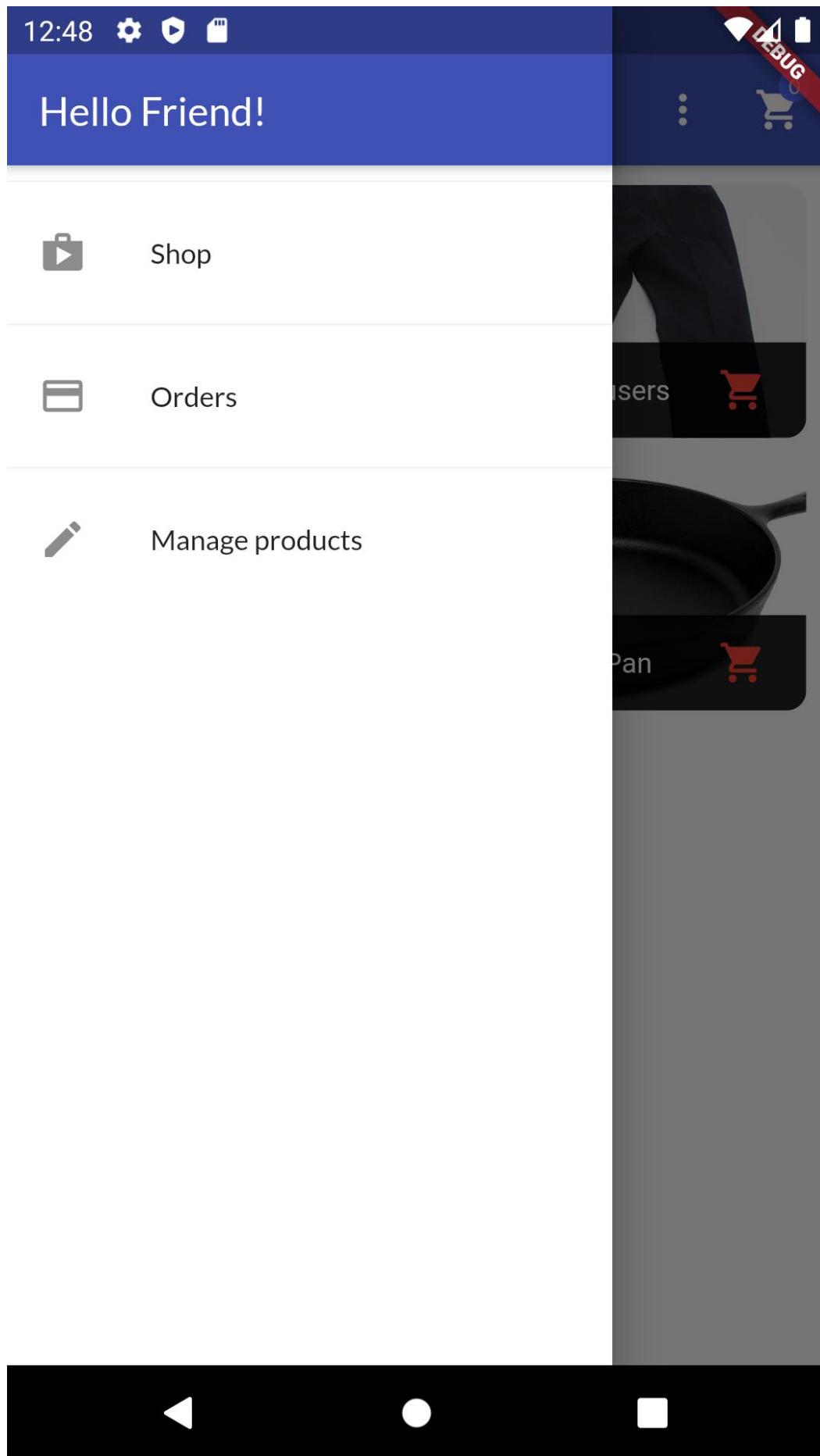


לאחר מכון מגיעים למסך בו המשתמש בוחר את נתיב ה-"default" שלו. נבחר על ידי גוגל מפה (ممמש אפשר להציג את המפה ולראות מה סביבה. כמו גם האם יהיה זמן נוסף בחירה על המפה עצמה).

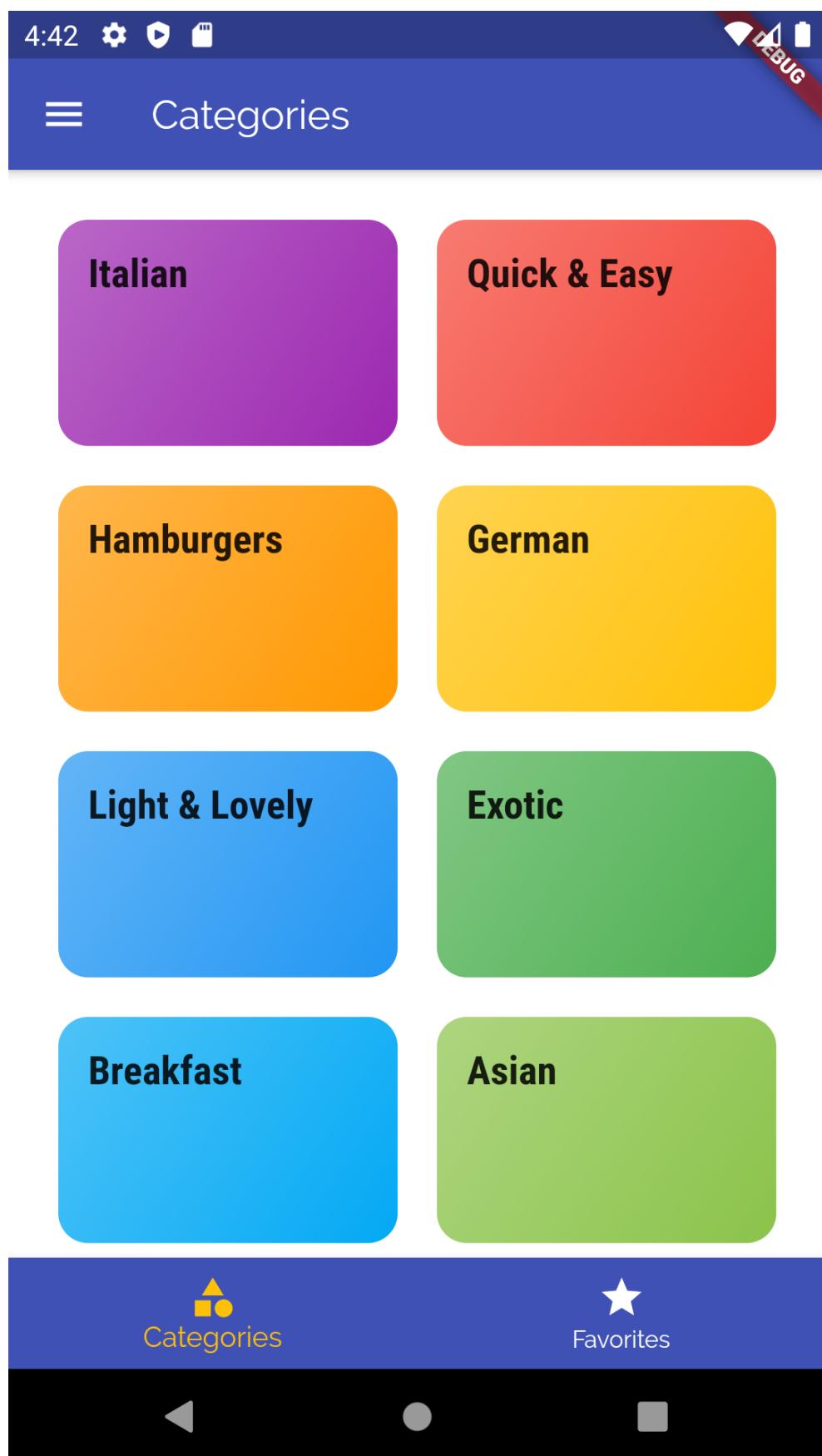


אפליקציית קניות





אפליקציית מתכונים



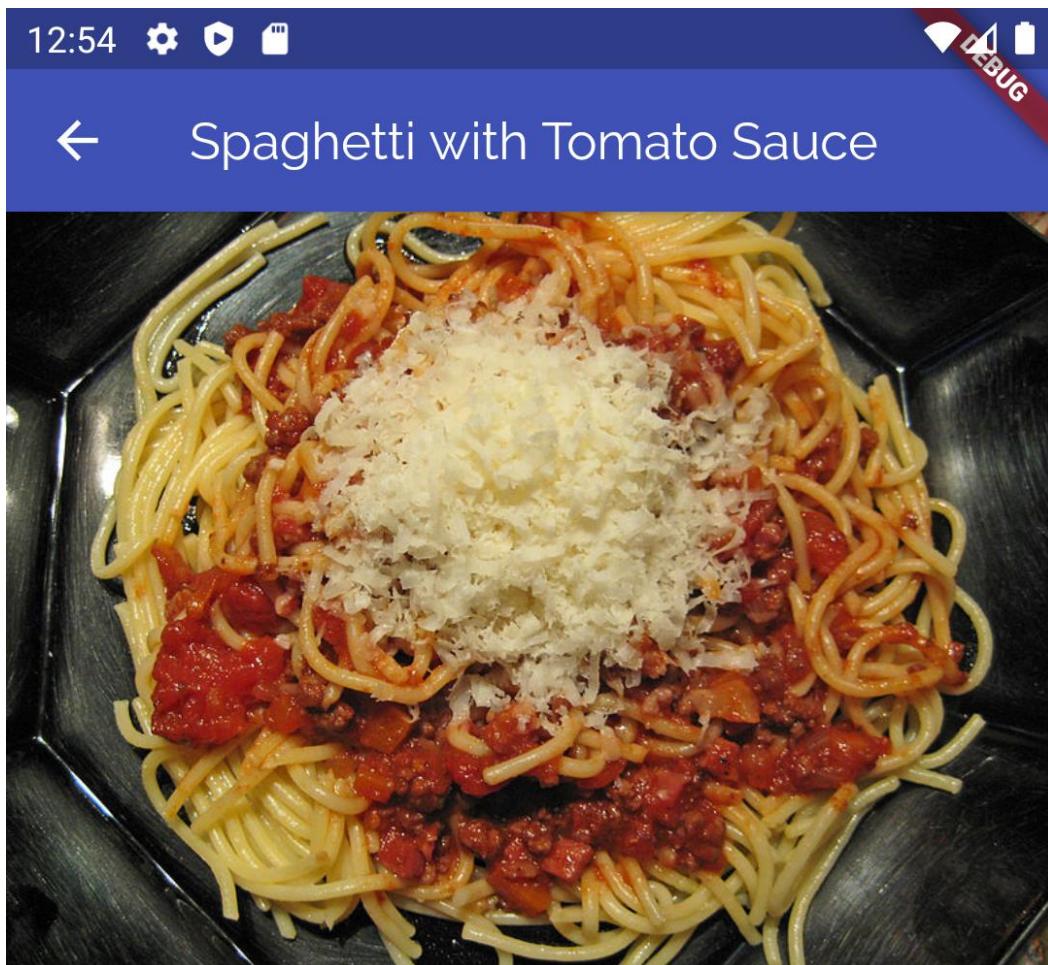
12:54 DEBUG

Quick & Easy

Spaghetti with Tomato Sauce

⌚ 20 min 🗂 Simple 💲 Affordable

Toast Hawaii



Ingredients

4 Tomatoes

1 Tablespoon of Olive Oil

1 Onion

250g Spaghetti

Steps



Cut the tomatoes and the

אפליקציית מעקב על הוצאות כלכליות

