



תיכון הדרים הוד השרון



Blackspace

שם התלמיד: עידן הירש

ת.ז: 214290249

שם המורה: ניר סליקטר

תאריך הגשה: 30/05/2021

תוכן עניינים

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

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 קודuproject.....
41	שימוש ב - accessories של הטלפון
41	תכנות מונחה עצמים
41	Remote procedure call.....
41	Asynchronous programming.....
42	קישור לקודuproject ב - GitHub.....
42	הסבר כללי על תיקיית הקבצים שלuproject
43	הסבר על התיקייה lib
44	הסבר על התיקייה assets
45	הסבירים על קבצים, פונקציות וחלקה קוד חשובים
45	הגדרת Packages לשימושuproject.....
46	קובץ שימוש במכשיר
48	התקשרות עם השרת ובסיס נתונים.....
50	רפלקסיה.....
50	תחושתי מהעבודה עלuproject
50	כלים שקיבנתי ואקח איתי להמשך
50	אתגרים שעמדו בפניו במהלךuproject

50	מסקנות מהפרויקט
50	מה הייתה עשו אחרת אילו הייתה מתחילה היום
51	מה היה יכול להפוך את העבודה ליעילה 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	אפליקציות קניות
63	אפליקציות מתכונין
66	אפליקציות מעקב על הוצאות כלכליות

מבוא

קישור לקוד הפרויקט ב – GitHub

<https://github.com/IdoHirsh0/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 מתבצע בו, מדריך למשתמש וצילומי מסך אשר יעזור להמחשה. בנוסף קיימים בו גם פירוט על מבנה הנתונים והטכנולוגיות השונות בהם השתמשתי לפיתוח הפרויקט, מדריך למפתח ולבסוף רפלקציה אישית וביבליוגרפיה על הנעשה. ומובן שגם נספחים... :)

הרקע לפרויקט

לפני כ5 חודשים עברו (דצמבר 2020), עלהה סדרה ישראלית חדשה בטלוויזיה שנקראת Blackspace ולאחר המלצות רבות מחברי החלהטי 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 וערכנו יחד את השני דרך אפליקציה שנבנתה על ידי תלמיד באאותה השכבה. לאחר שצפיתי בסדרה לא יכולתי שלא לחשב כמה كيف זה יכול להיות לבנות אפליקציה כזו שכל החברים שלך ישתמשו בה يوم יומם. ומשם קיבלתי את הרעיון לאפליקציה של Blackspace.

תהליכי מחקר

סקירה המצביעים בשוק

כאשר חשבתי על לבנות אפליקציית Chat חשבתי על חלופות אחרות אשר קיימות בשוק. דוגמאות לכך הם: Gadgets כלו, בעלות משאים עצומים וצורות פיתוח מוכשרים, אך הרעיון על לעשות משהו כזה ממשי סקון אותי מאד.

חידושים בפרויקט

על מנת בכל 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 פרטים 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 ללמוד להכין אפליקציה, התחלתי ללמידה על התחום. תחילת חיפשתי הרבה על פיתוח אפליקציות גם ל-iPad וגם ל-OS iOS ללא הצלחה. לאחר שניות רבות של מחקר על התחום גיליתי את Codebase Dart i Flutter. בקצרה היא פלטפורמת בנייה אפליקציות שפותחה על ידי גугл, שתחתה Flutter אחד, האפליקציות עובדות על סוגים רבים של מערכות הפעלה, כדוגמת Flutter, Web, Desktop, Android, iOS, וכו'. וכך, ידעתי שהוא יהיה הפתרון הכי טוב בשביב הפרויקט שלי, שכן 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 מהאתר Udemy שסייעתי בהצלחה את הקורס של Certificate



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

במהלך כל הזמן הזה הצטרכתי גם לפROYKT MAX, פרויקט לנוער המשתמש במוגמות סיבר ברחבי הארץ על מנת לנסוט 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 ה 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 לכת לאיזה כיוון ותחום שהוא רוצה, הפרויקט שברחטי לעשות היה חדש לגמרי, 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, לחברוי ואני נוכל להשתמש בה כדי לדבר אחד עם השני, הייתה מעלה 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

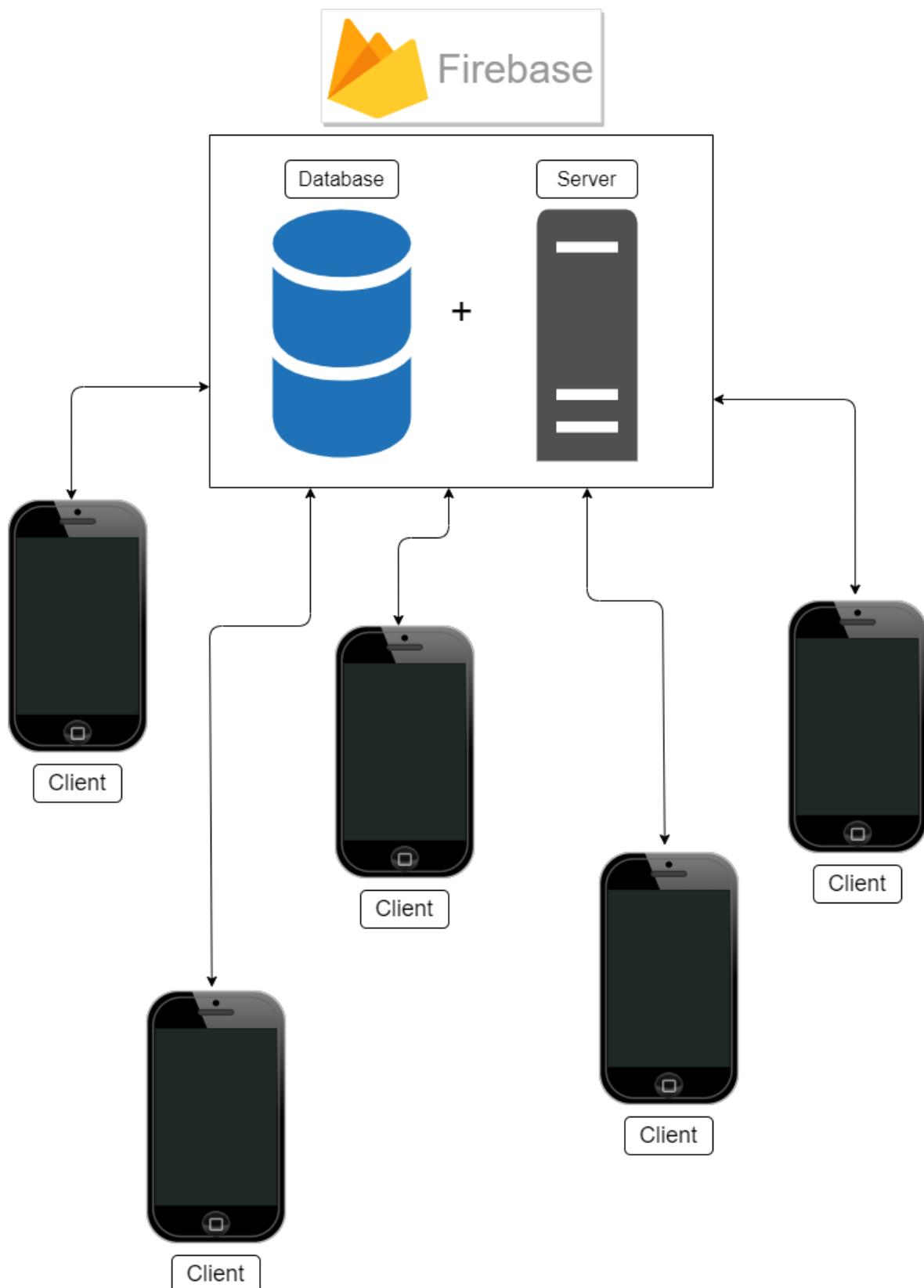
הפרויקט עונה על הצורך של תקשורת בין אנשים וחבריהם. בעזרת הפרויקט, אנשים שונים, שלאו דואק מכירים אחד את השני, יכולים לדבר יחד וליצור שיח משותף.

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

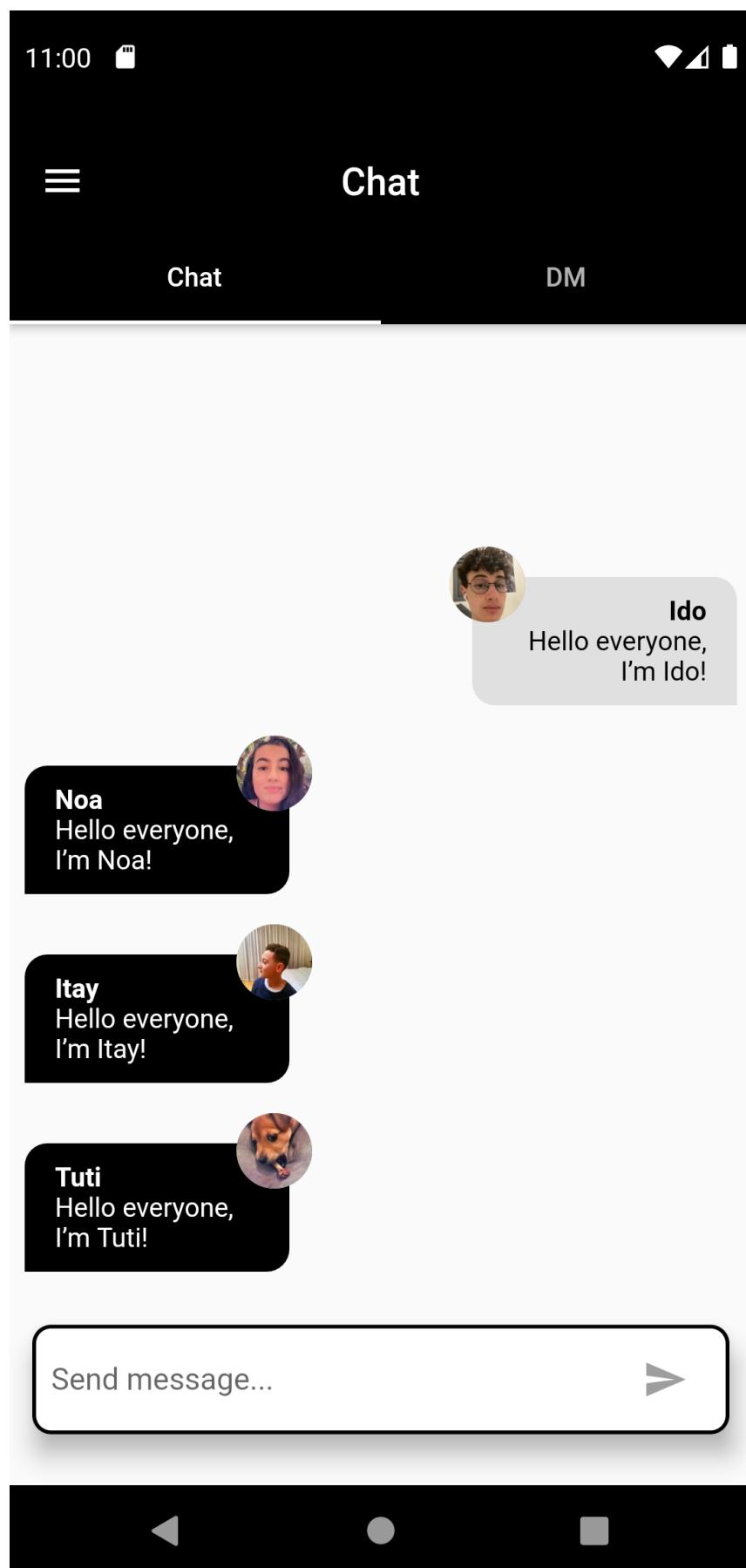
ארQUITקטורה של הפרויקט

הציג שרטוט על של הפרויקט

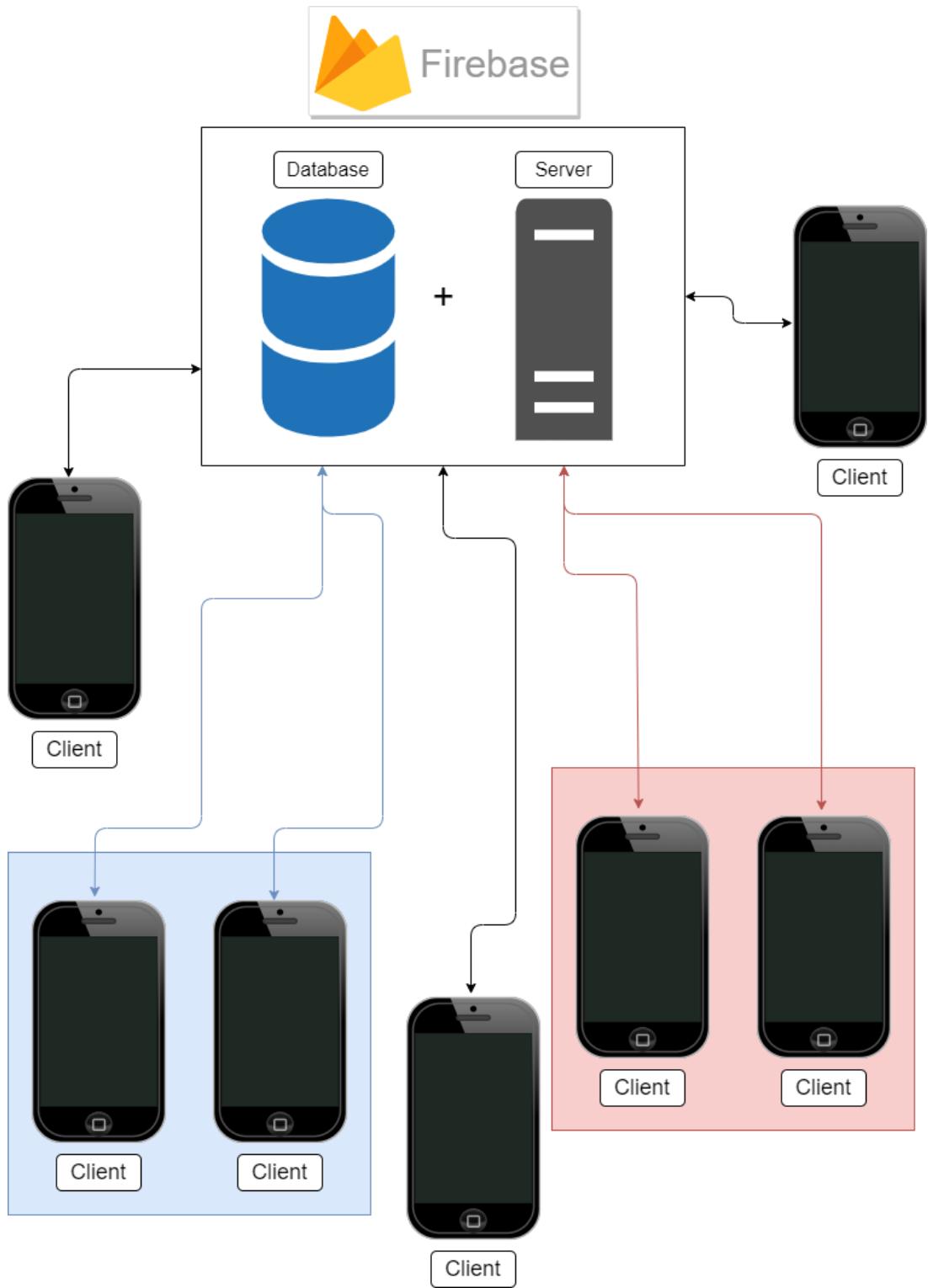
מייצג גם את הארכיטקטורה של הצ'אט הראשי של האפליקציה.



תמונה של מסך הצ'אט הכללי, ה – Broadcast, באפליקציה, בהתאם לארכיטקטורה למעלה.



ארQUITקטורת שיחות פרטיות בין שני משתמשים

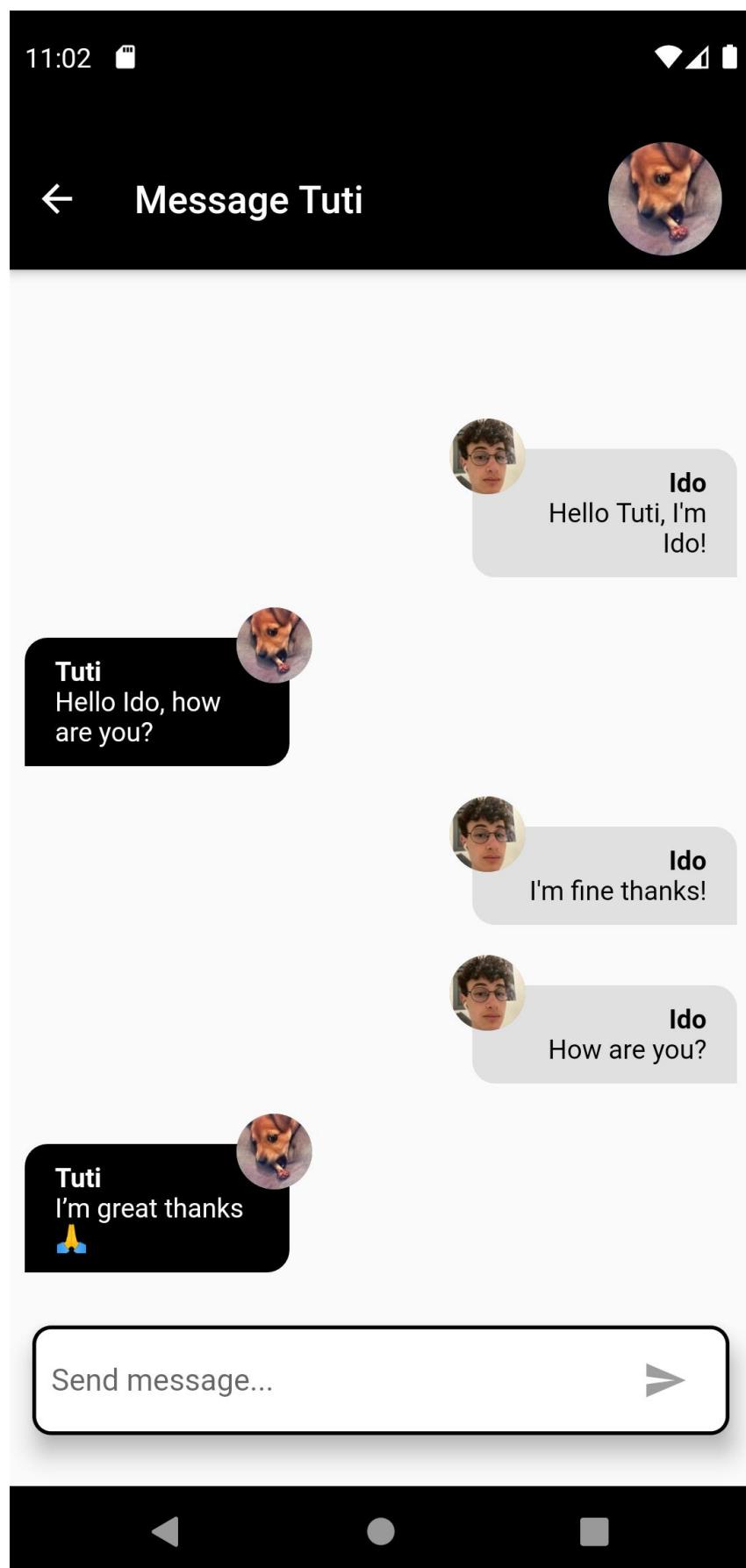


ניתן לראות כי יש שני משתמשים שמסומנים **באדום**, שניים **בכחול** וכמה **שחורים**.

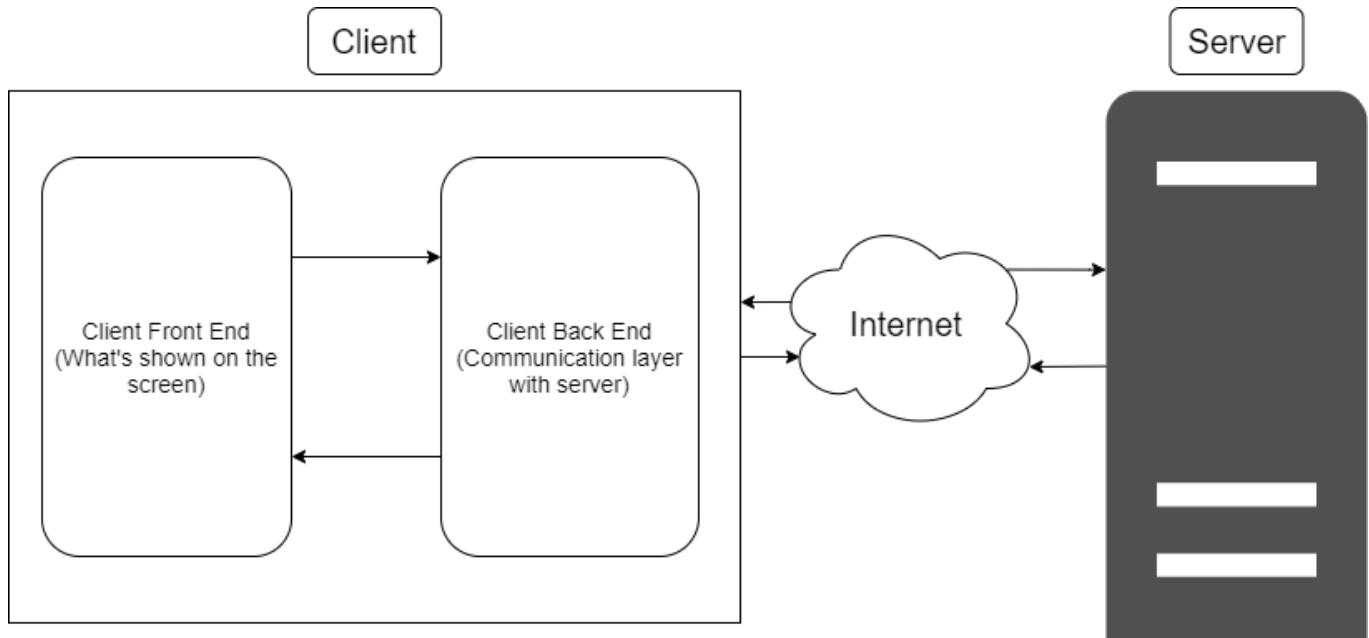
אלו **שחורים** מדברים בין עצמם בז'אט פרטי, אלו **שבכחול** גם כן, בעוד המשתמשים האחרים, **באדום**, עדין מדברים בז'אט הכללי.

ההודעות הפרטיות, כמו בז'אט הכללי, מאוחסנות בסיס הנתונים, ועובדות מלקוח ללקוח דרך השרת.

תמונה של מסך הצ'אט הפרט בownload נמצאה בין שני משתמשים, בהתאם לארכיטקטורה לעלה.



תפקידה של כל יחידה



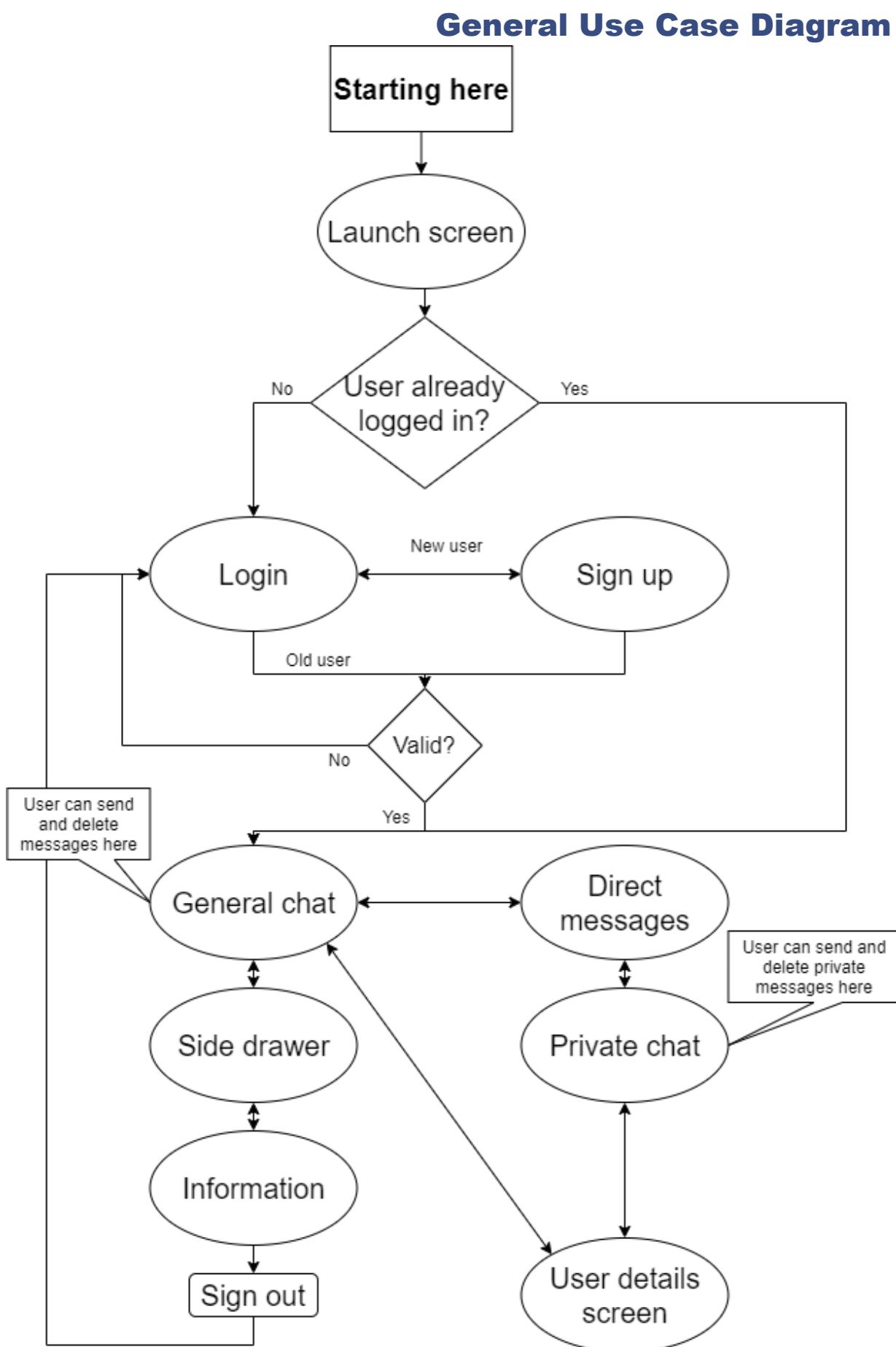
שרת - Server

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' אליו הן שייכות.

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

לקוח - Client

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



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 של 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 פרטים 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 של 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 כך ומוצגת על המסך הודעה על כך ששם המשתמש או הסיסמה אינם נכונים, ושצריך להכיסםשוב.
5. לאחר הכנסת פרטים נכונים ותקינים, הלוקו יתחבר למערכת, והאפליקציה יתבצעה תעבור 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 שבסיס הנתונים בזמן אמת, כל המשתמשים מקבלים את ההודעה מבסיס הנתונים ישיר אחרי שהיא נוספה לשם.

שליחת הודעה בצד'אט פרטי 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 פרטי עם האדם אותו 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 הכתפו, המשתמש יכול לשלוח הודעה פרטית למשתמש האחר, בדיק כי שהסברתי בצד'אט הכללי, רק שהפעם יוכל להבדוק שמדובר בנפרד לצד'אט זה, כך שרק שני משתמשים אלו יכולים לראותו.
5. כל צ'אט פרטי חדש נשמר כ – Document ב – Collection chatRoom, ובמסגר 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 במסגר משלה.

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

צפיה בכל השיחות הפרטיות של המשתמש, וכניסה לשיחה פרטית קודמת

תיאור הפעולה: המשתמש מנסה לשלוח הודעה למשתמש שכבר דיבר איתו בעבר.

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

תנאי סיום: המשתמש מדבר ביצ'אט פרטי עם האדם אותו רצה לדבר.

שלבי פעולה:

1. המשתמש נכנס למסך היצ'אט 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' שהסברתי ביצ'אט הכללי, רק שהפעם ההודעות יישמרו בנפרד לצ'אט זה, כך שרק שני משתמשים אלו יכולים לראותו.
5. כל ההודעות של היצ'אט הפרטי שכבר נוצר נשמרות באותו Document שנוצר עבורי היצ'אט הפרטי זהה כאשר דיברו בפעם הראשונה.

מחיקת הודעה

תיאור הפעולה: המשתמש מנסה למחוק הודעה שלחו בעבר, או ביצ'אט הכללי או ביצ'אט פרטי מסוים.

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

תנאי סיום: המשתמש מחק את ההודעה שרצה.

שלבי פעולה:

1. המשתמש לוחץ לחיצה ארוכה על ההודעה שלו שהוא רוצה למחוק.
2. קופץ חלון שואל אותו אם הוא בטוח שהוא רוצה למחוק את ההודעה מכלם.
3. אם לחץ כן, ההודעה תמחק מהמקום בו היא נמצאת בבסיס הנתונים, וכן גם מהיצ'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 הת לחבר למערכת והוא נכנס למסך היצ'אט הכללי.

תנאי סיום: המשתמש התנטק מאפליקציה ו חוזר למסך ההתחברות.

שלבי פעולה:

1. המשתמש לוחץ על המגירה הצדית לצד שמאל העליון במסך היצ'אט 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 לעשות אפליקציית צ'אט, שתהיה בעלת אפשרות לשלוח הודעות לכל משתמשי האפליקציה (broadcast) ובעלת אופציה לשלוח הודעות פרטיות בין שני משתמשים, لكن כרגע עשיתי. לפתרון שלי השתמשתי, כפי שכבר ציינתי, בפיתוח אפליקציה בטכנולוגיה של Flutter.

בחرتني להשתמש ב- Flutter משום שזה היה נראה הפתרון הטוב ביותר עבורו. לאחר שהחיפשתי הרבה על איך אפשר לבנות אפליקציית צ'אט, וגיליתי אופציות שונות כמו Kivy, Android studio ועוד, הבנתי שFlutter הוא התשובה. הכל נכון לשימוש, אפשרeli לפתח אפליקציה גם ל – Android וגם ל – iOS באותו הזמן, ובעל תמיכה ודוקומנטציה שלא ראייתי ממש.

בשביל השירות של הלקוח Firebase, יכולר אני כתבתי את קוד הלקוק, האפליקציה, ו – Firebase היא השירות של, כדי שלא יצאר גם לכתוב את קוד השירות, עקב חוסר בזמן, וגם כי זהה פלטפורמה שבנינו 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 הפותחה על ידי גугл. הוא משמש לפיתוח יישומים חזוי פלטפורמות עברו Web,iOS, Linux, Mac, Windows,iOS,mac,Android,Android studio ועוד 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, משום 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 עליה ללמידה, ולהשתפר בה מהר מאוד.

בנוסף, מכיוון ש – Flutter היא פלטפורמה שפותחה על ידי גугл, והיא נפוצה מאוד כיום, יש עליה דוקומנטציה אינסופית באינטרנט, מה שמקל על הליך הלמידה.

Firebase

Firebase היא פלטפורמת שירותי בסיס נתונים, שפותחה על ידי גугл ליצירת 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, עקיבך קר שהפלטפורמה מפותחת על ידי גугл, כמו Flutter, ישנים SDK's, Libraries ו – Packages מואוד 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, שקיימות שעוזרת מאוד לפתור בעיות ולהבין טעויות שנעשו.

מדריך למשתמש

הוראות התקינה

את האפליקציה שלינו ניתן להריץ על 3 פלטפורמות שונות: Android, iOS, Web

בעתיד אוליendra יעלה את האפליקציה לחניות האפליקציות, ואז הוראות ההתקנה יהיו רק להוריד את האפליקציה מחנות האפליקציות שתואמת למכשיר זה.

ארעוט אלו הם הוראות ההתקנה

עבור iOS ו - Android

1. להתקין Flutter <https://flutter.dev/docs/get-started/install/windows> - Flutter

2. עבור Android: להתקין Android studio וליצור סימולטור של מכשיר אנדרואיד שירוץ על המחשב
https://www.youtube.com/watch?v=WkEf1fa1sn0&ab_channel=NiyasC

עבור OS(עובד רק על macOS): להתקין Xcode מה – App store וליצור סימולטור של מכשיר OS שירוץ על המחשב – <https://bit.ly/3tUKqYi>

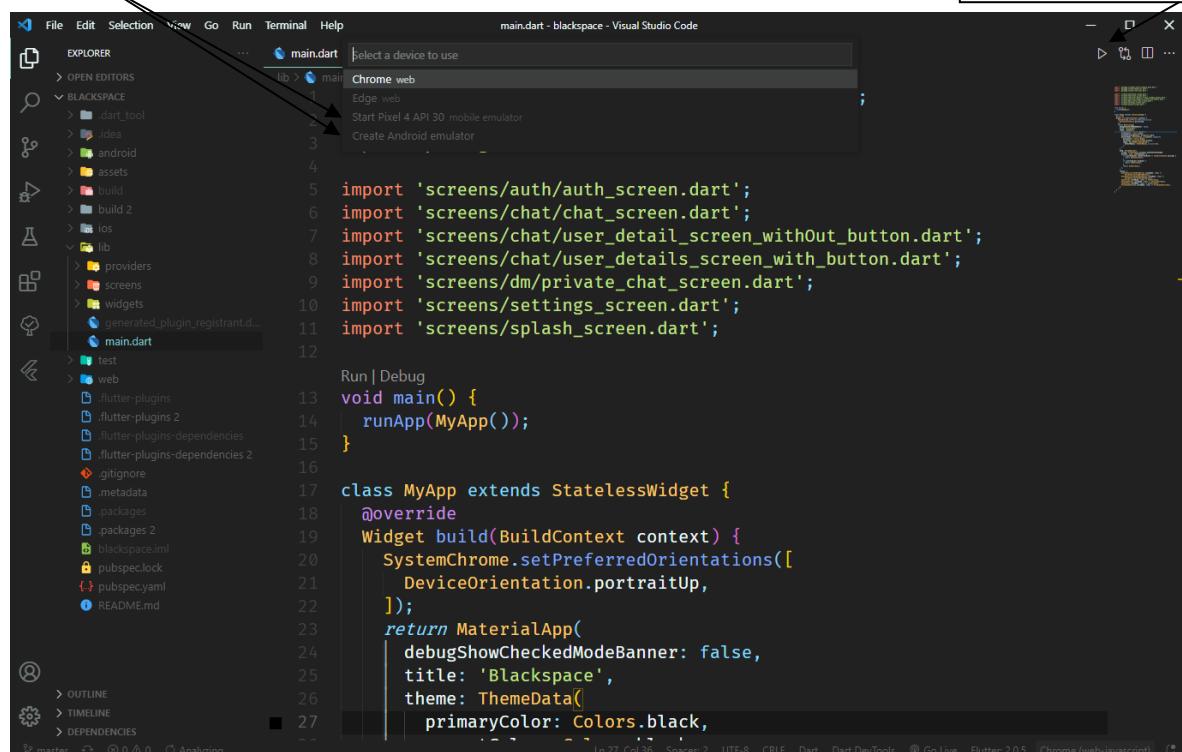
3. להתקין VSCode (לא חובה אבל כך וכי נוח להריץ את הפרויקט) -
<https://code.visualstudio.com>

4. להוריד את הקוד של הפרויקט שלי מ – GitHub ולפתחו אותו באמצעות VSCode
<https://github.com/IdoHirsh0/Blackspace>

5. בטור הTerminal לבחור את הקובץ main, לבחור את הסימולטור שייצרנו מתוך VSCode על מנת שהקוד יירוץ עליי, וללחוץ חנוך.

2. לבחור באחת מהאפשרויות הללו

3. ללחוץ כאן בשביב הרצה - run



```

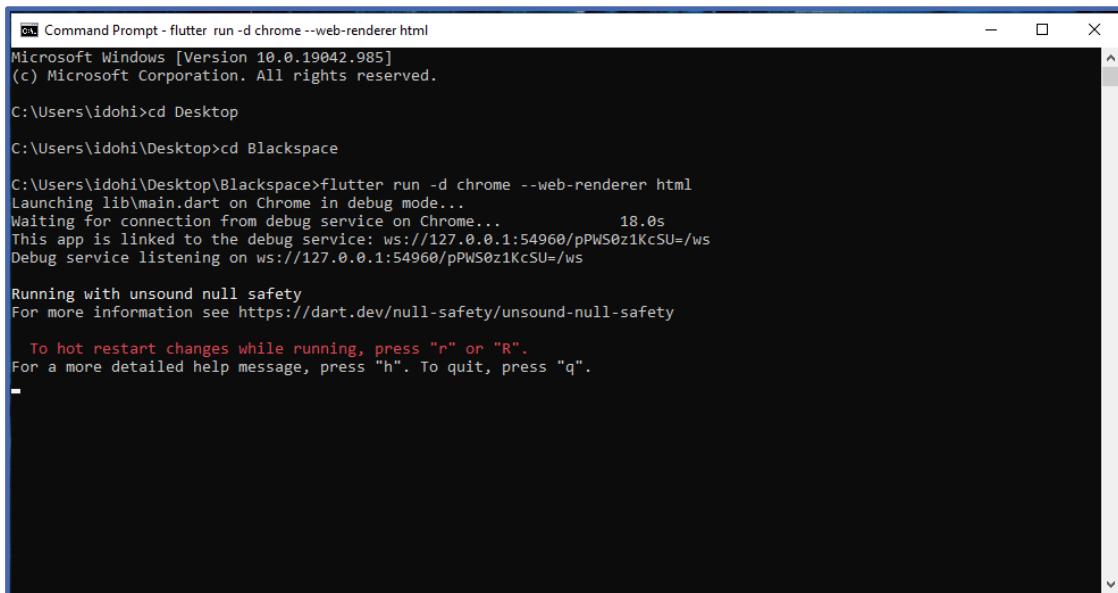
File Edit Selection View Go Run Terminal Help
main.dart - blackspace - Visual Studio Code
LIB > main.dart select a device to use
LIB > main
1 Chrome web
2 Edge web
3 Start Pixel 4 API 30_ mobile emulator
4 Create Android emulator
5 import 'screens/auth/auth_screen.dart';
6 import 'screens/chat/chat_screen.dart';
7 import 'screens/chat/user_detail_screen_withOut_button.dart';
8 import 'screens/chat/user_details_screen_with_button.dart';
9 import 'screens/dm/private_chat_screen.dart';
10 import 'screens/settings_screen.dart';
11 import 'screens/splash_screen.dart';
12
13 void main() {
14   runApp(MyApp());
15 }
16
17 class MyApp extends StatelessWidget {
18   @override
19   Widget build(BuildContext context) {
20     SystemChrome.setPreferredOrientations([
21       DeviceOrientation.portraitUp,
22     ]);
23     return MaterialApp(
24       debugShowCheckedModeBanner: false,
25       title: 'Blackspace',
26       theme: ThemeData(
27         primaryColor: Colors.black,

```

1. ללחוץ כאן בשביב לבחור מכשיר

מעבר Web

- .1 להתקין <https://flutter.dev/docs/get-started/install/windows> - Flutter
- .2 להוריד את הקוד שלuproject שלי מ <https://github.com/IdoHirsh0/Blackspace> - GitHub –
- .3 להיכנס לתיקיה של הקובץ דרך – **Command Prompt**
- .4 לכתוב את הפקודה – **flutter run -d chrome --web-renderer html** וללחוץ **Enter**
- .5 יפתח חלון ב – Google chrome עם האפליקציה



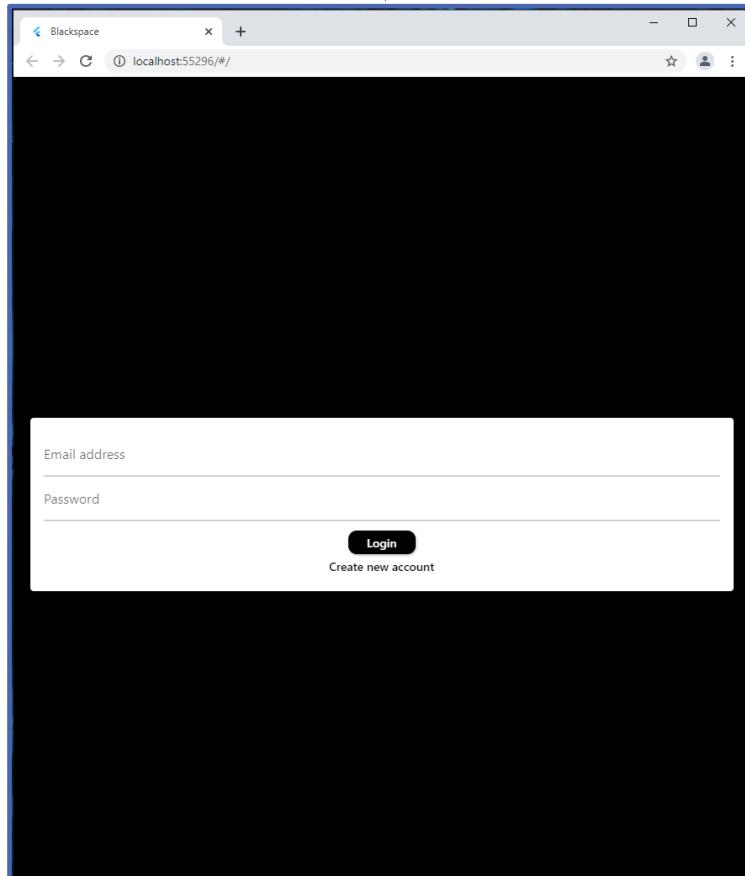
```
Command Prompt - flutter run -d chrome --web-renderer html
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\idohi>cd Desktop
C:\Users\idohi\Desktop>cd Blackspace

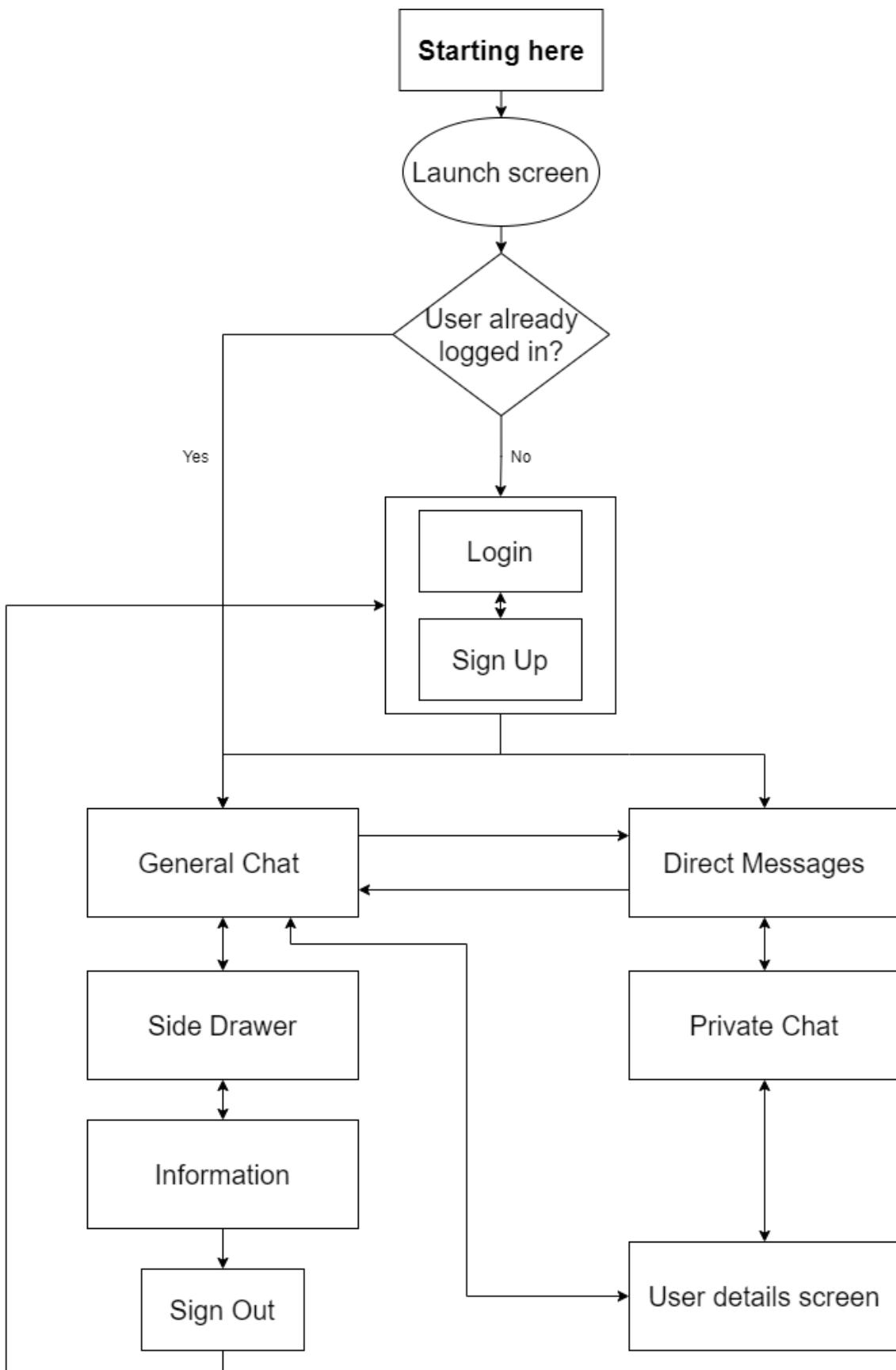
C:\Users\idohi\Desktop\Blackspace>flutter run -d chrome --web-renderer html
Launching lib/main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...           18.0s
This app is linked to the debug service: ws://127.0.0.1:54960/pPWS0z1KcSU=/ws
Debug service listening on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



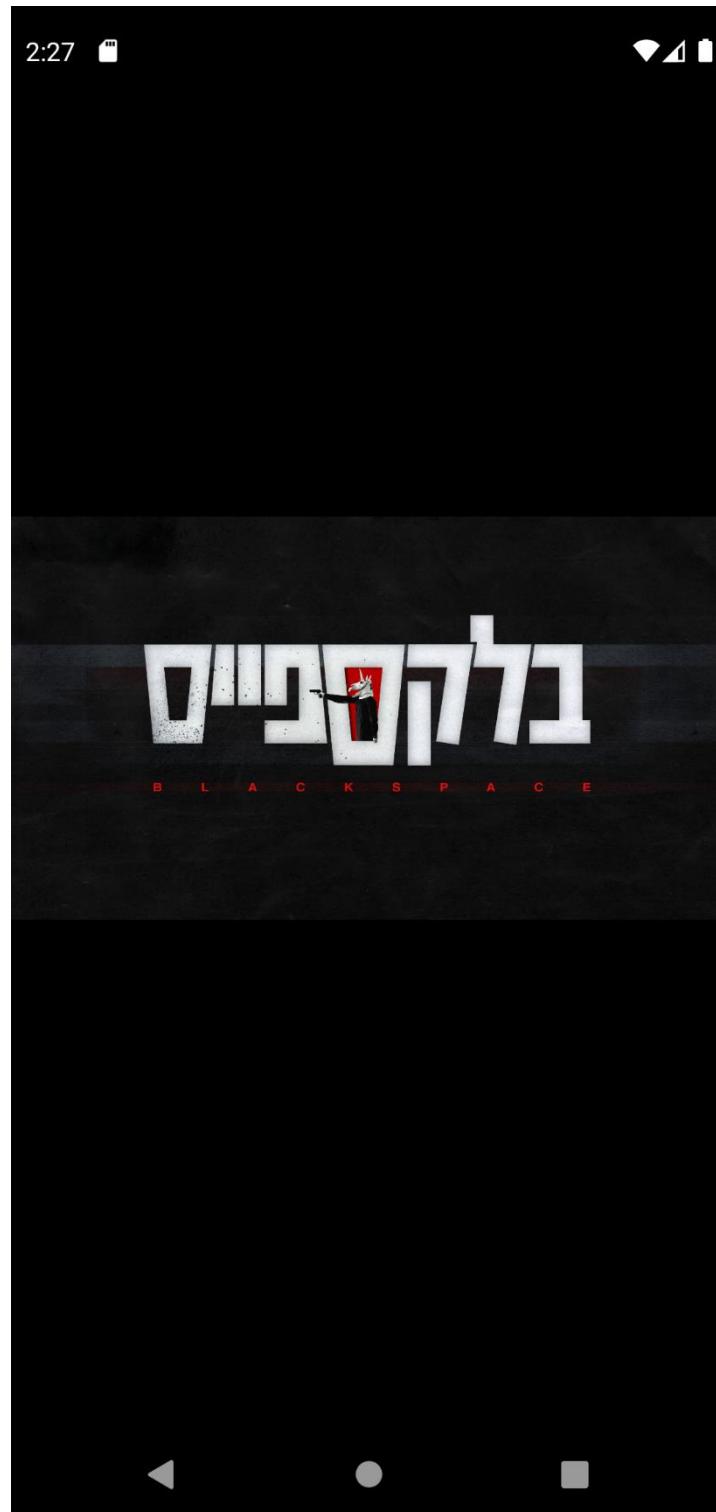
היררכיית מסכים



תרשימים מסכימים

מסמך טעינה

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



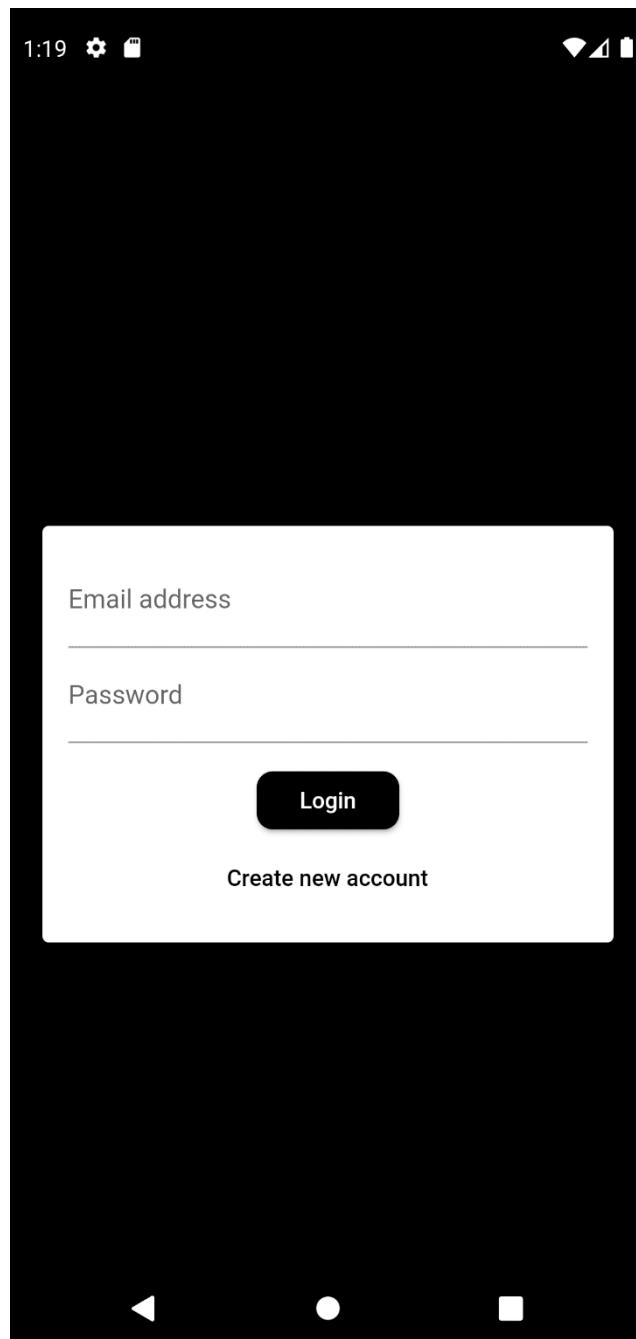
התחברות

המספר הראשוני שהמשתמש רואה כאשר הוא פותח את האפליקציה בפעם הראשונה. בפעמיים הבאות, אם לא התנתק מהאפליקציה, המשמש והסימן שלו ישמרו ויתחבר ישירות למערכת. (תיה אופציה להטנתקוות ממספר ההגדרות).

קלט: אימייל וסיסמה של המשתמש.

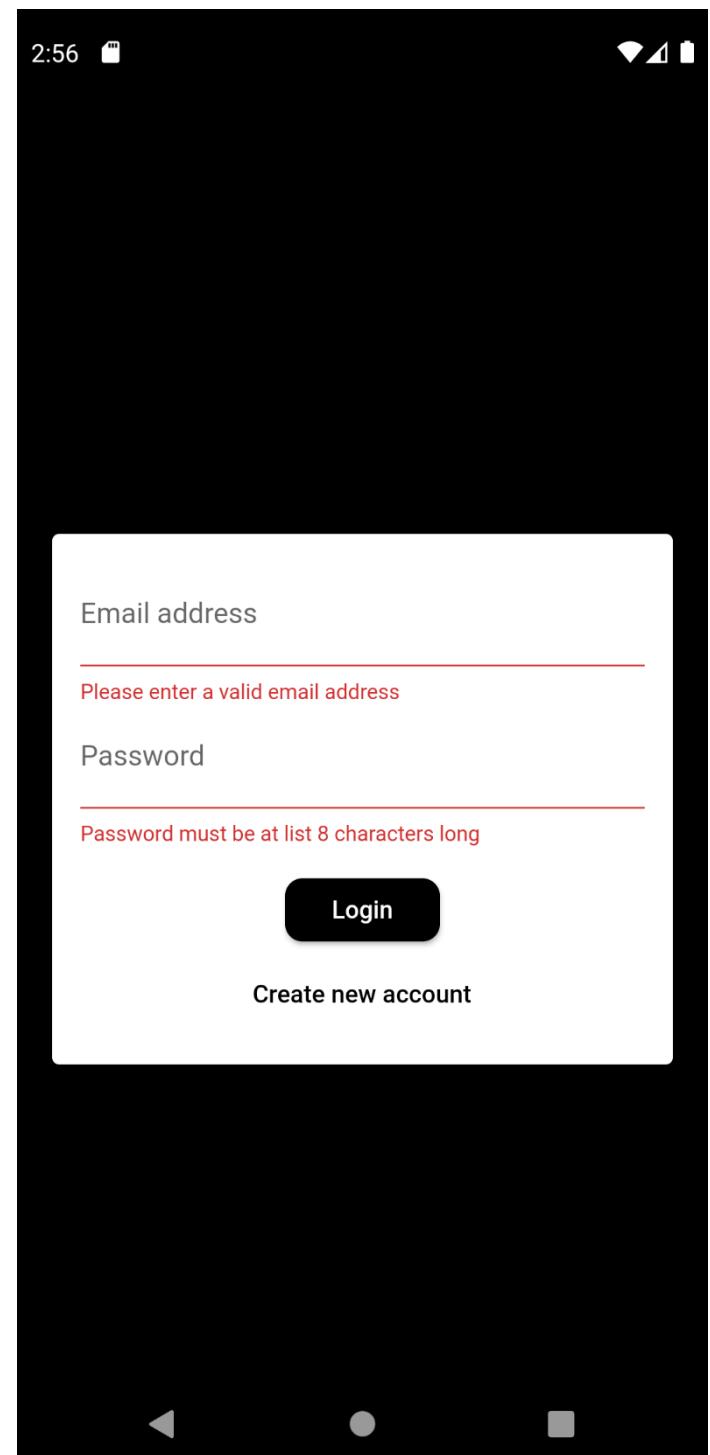
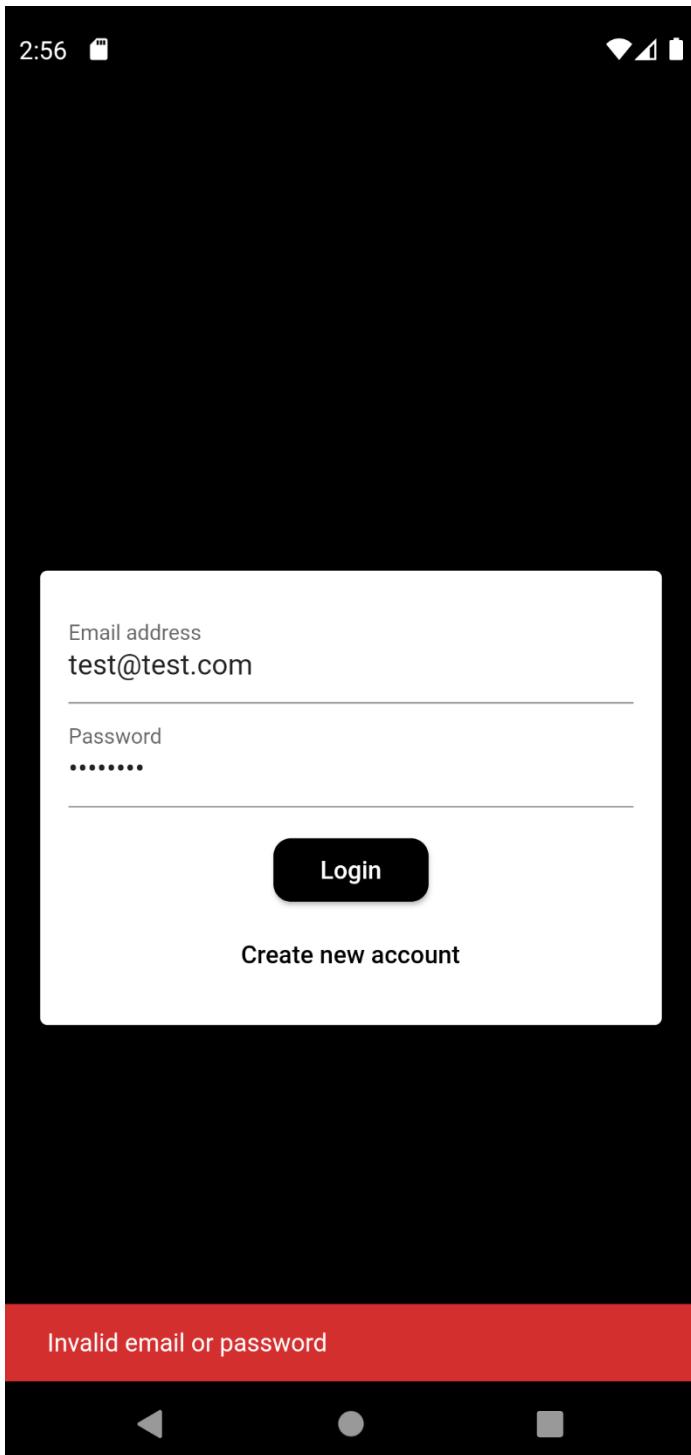
פלט: במקרה של שגיאה, בעקבות הולידציה, תציג הודעה מתאימה למשתמש. במקרה של הצלחה, יועבר המשתמש למסך הראשי (מסך הציג).

מספר זה ניתן לעבור שירותי למסך ההרשמה, באמצעות לחיצה על הכפתור המופיע לcker, וחזרה.



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

בדיקות תקינות האימייל והסיסמה
אצל הלוקוח



הרשמה

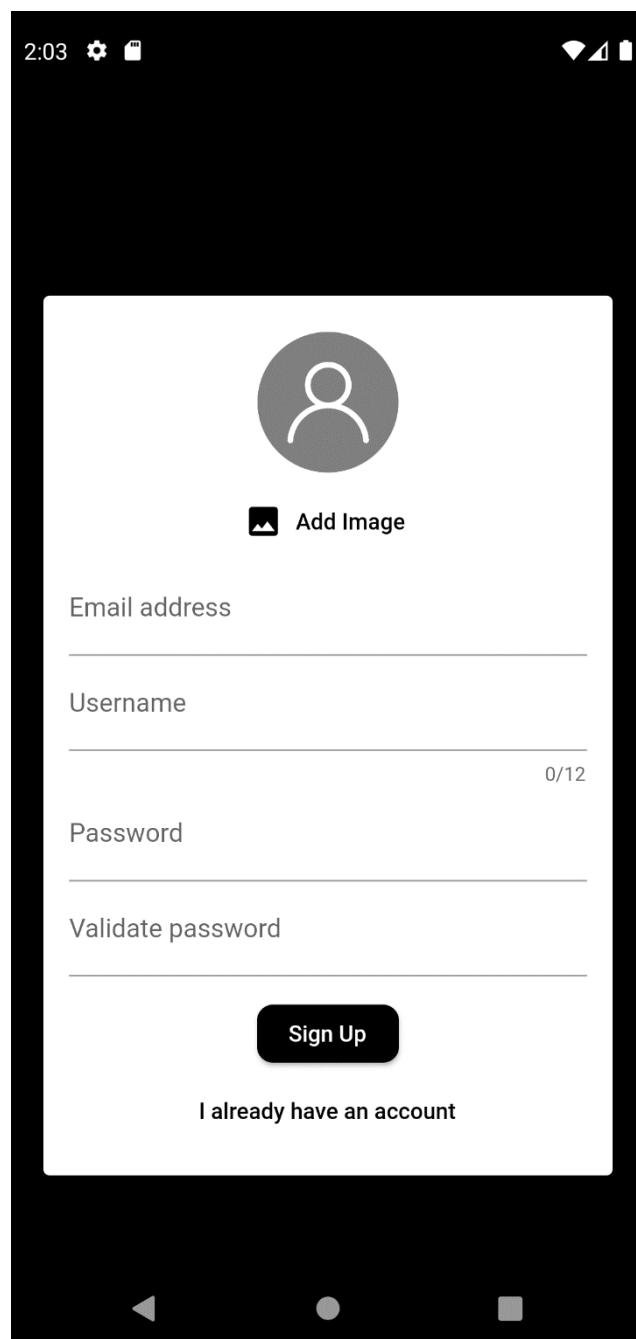
מסך ליצירת משתמש חדש באפליקציה.

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

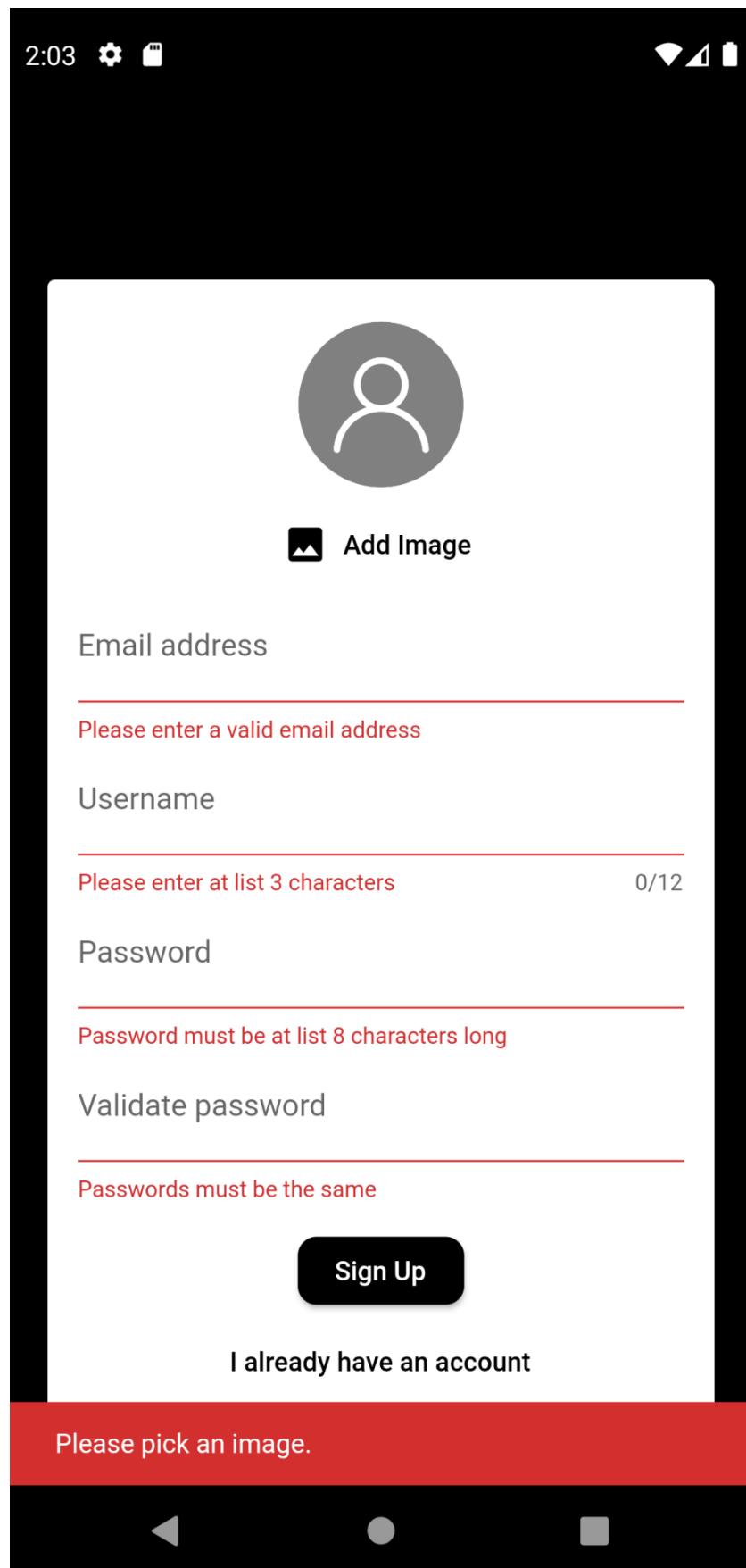
אני מבקש אימייל משומם שבשביל הרשמה לשרת, Firebase, יש צורך באימייל וסיסמה. בנוסף, אני מבקש אימייל מסוים הפורטים האישיים על כל משתמש. בשבייל יצירט קשר דרך אימייל בין שני משתמשים מחוץ לאפליקציה. כמו כן, אפשר להשתמש באימייל על מנת לאפס סיסמה. (אך אני לא הספקתי להוסיף את האופציה הצעאת עדיין).

פלט: אם כל הפורטים הגיוניים (עברו את הווילדייניה) ייווצר משתמש חדש, הוא יתווסף לבסיס הנתונים, והמשתמש יועבר למסך הציג הראשי.

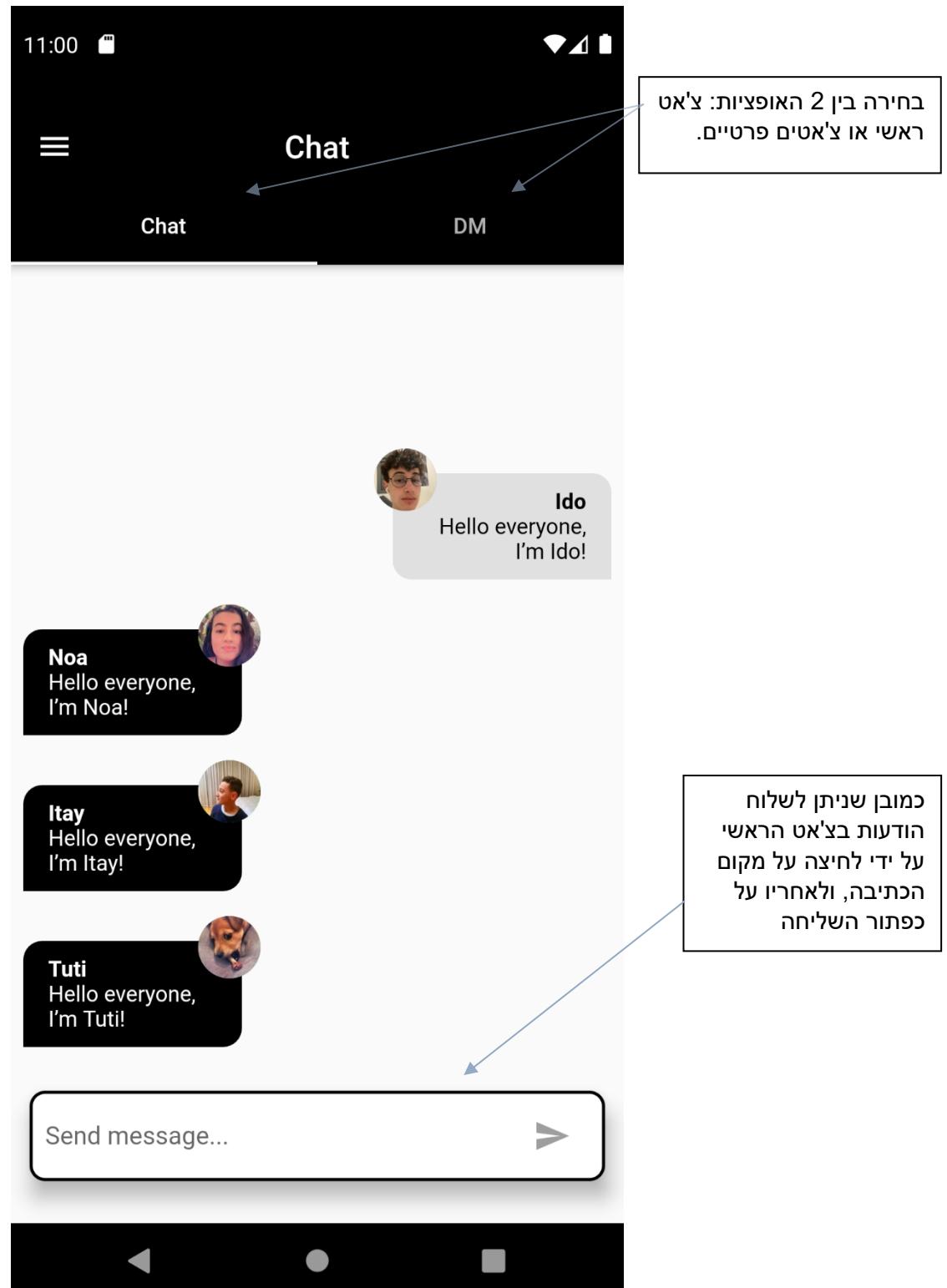
מסך זה ניתן לעבור שירות למסך ההתחברות, באמצעות לחיצה על הכפתור המועד לכך, וחזרה.



כל הودעות הוולידציה שיכלות להיות בעת ניסיון יצירת משתמש חדש



מסר צ'אט ראשי
מסך בו המשתמש רואה את כל ההודעות שנמצאות בצ'אט הגדול של האפליקציה, צ'אט הBroadcast. כל ההודעות שנשלחו בצ'אט. גם אלו לפני שהמשתמש נוצר, וגם אלו לאחריו.

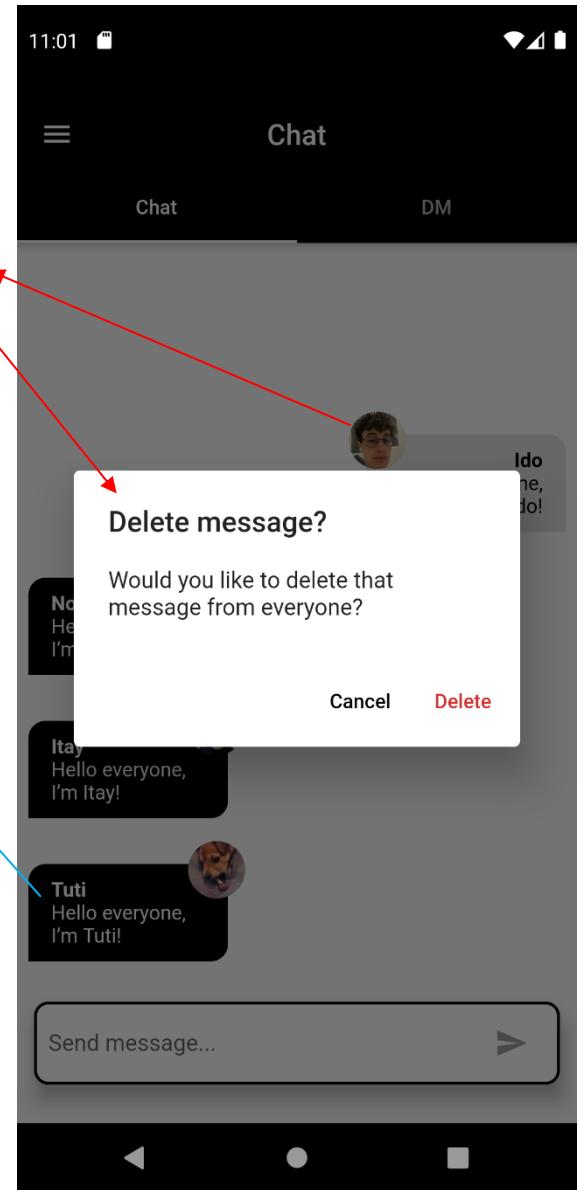
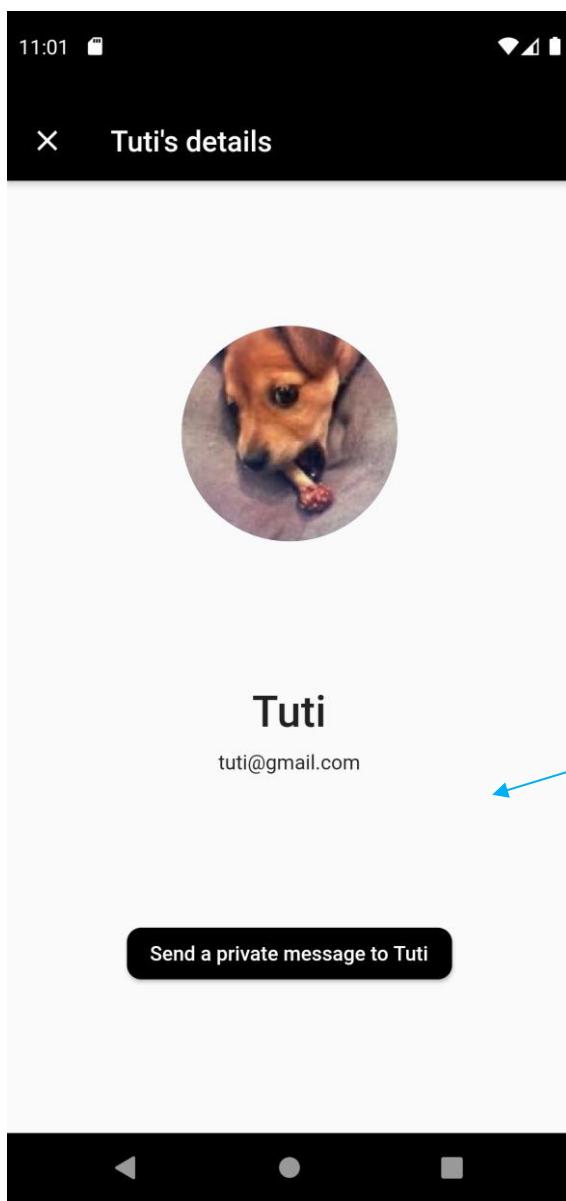


משתמש בחומר יכול למחוק את ההודעות שנשלחו על ידי המשתמש המחבר הינו בבע אפור, בעוד ההודעות שנשלחו על ידי אחרים הן במצב שחור), על ידי לחייב ארוכה על ההודעה אותה הוא רוצה למחוק.

כמובן שם משתמש לוחץ ארוך על הودעה שימושה אחר שלח, הוא לא יכול למחוק אותה. במקום זאת, הוא מעבר לדף פרטיים על אותו אדם.

לאחר לחיצה ארוכה על ההודעה שנשלחה על ידי משתמש אחר, במקרה זה Tut_i (המשתמש של הכלבה שלו) (๖)

לאחר לחיצה ארוכה על ההודעה שנשלחה על ידי המשתמש המחבר, במקרה זה soId (המשתמש שלו)



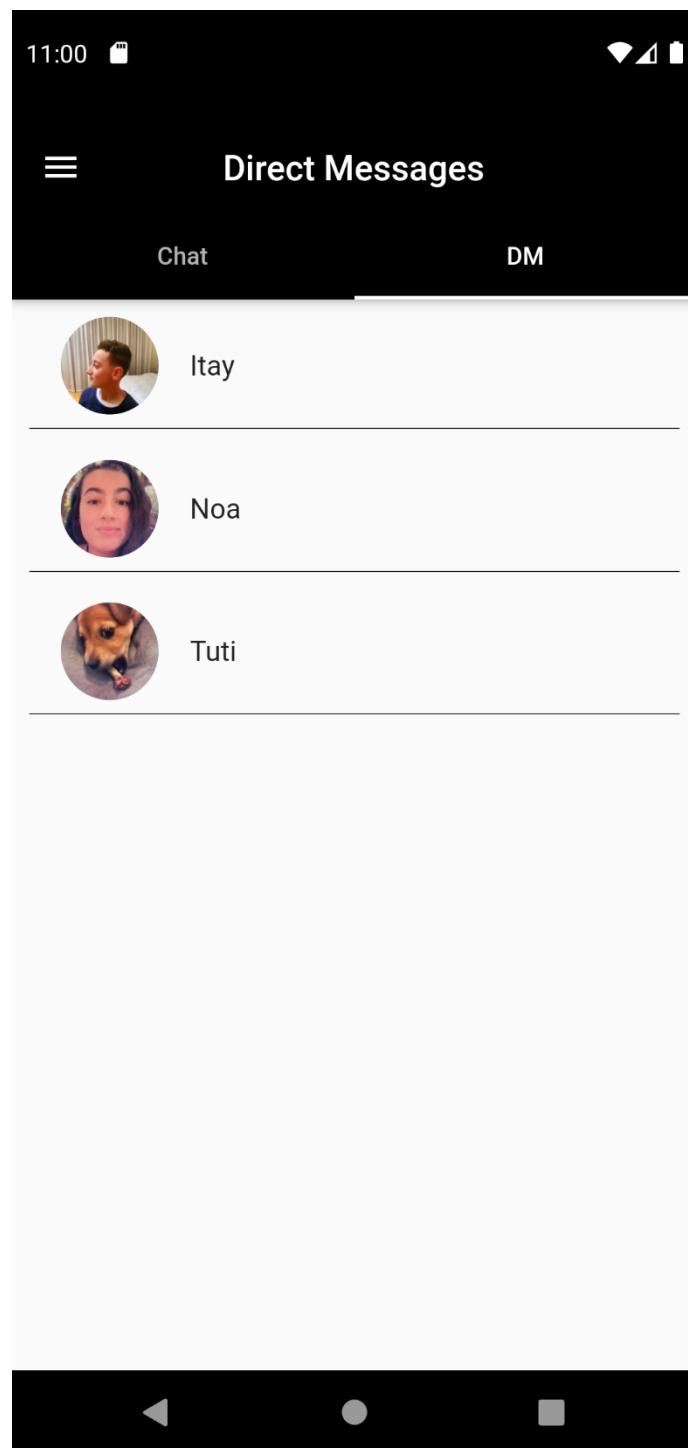
ממסך זה ניתן לעבור, על ידי לחיצה על כפתור
... "Send a private message to ..."
פרטית עם אותו אדם.

מסר כל השיחות הפרטיות

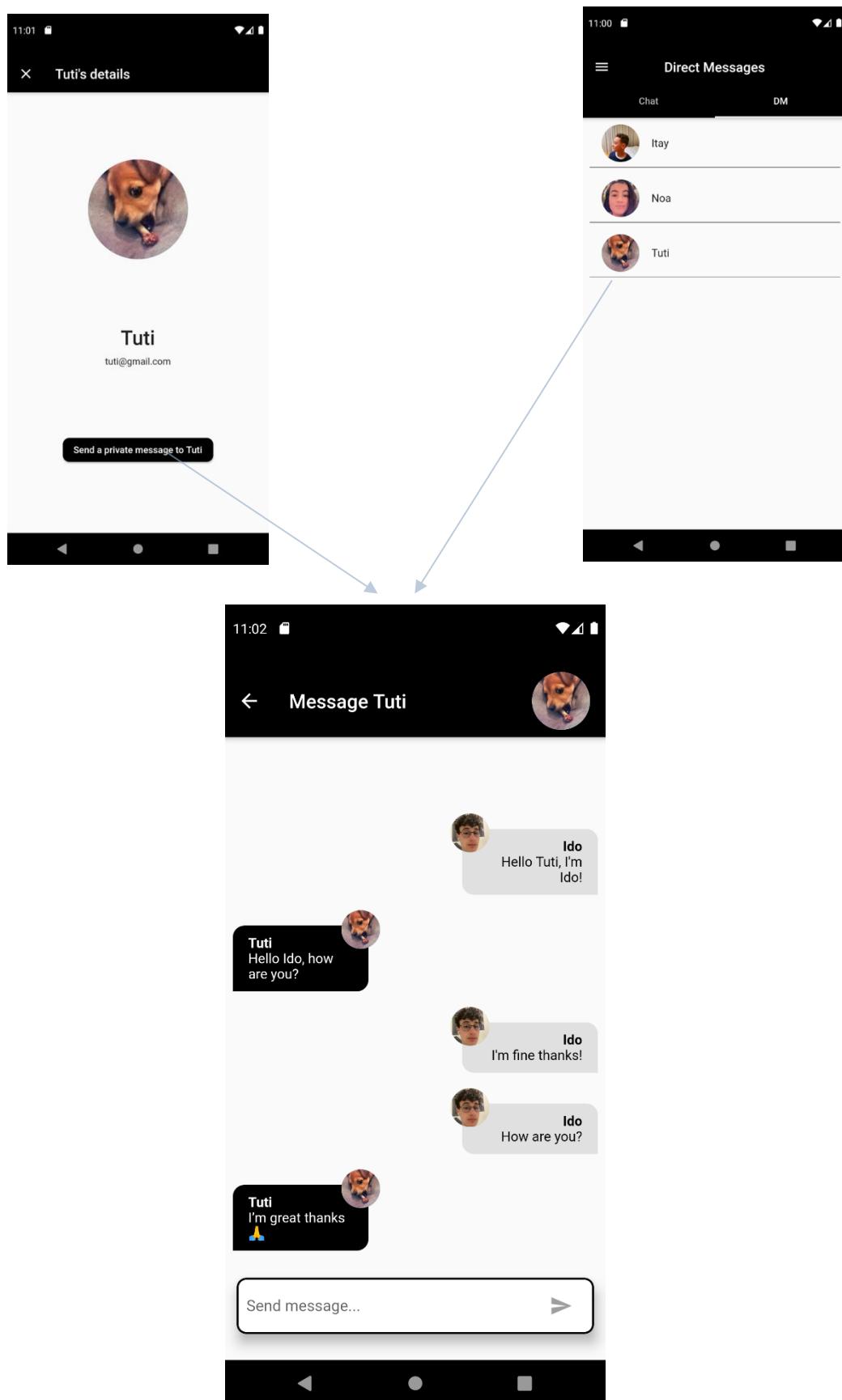
במסך זה המשתמש רואה רשימה של כל האנשים איתם הייתה לו כבר שיחה פרטית.

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

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



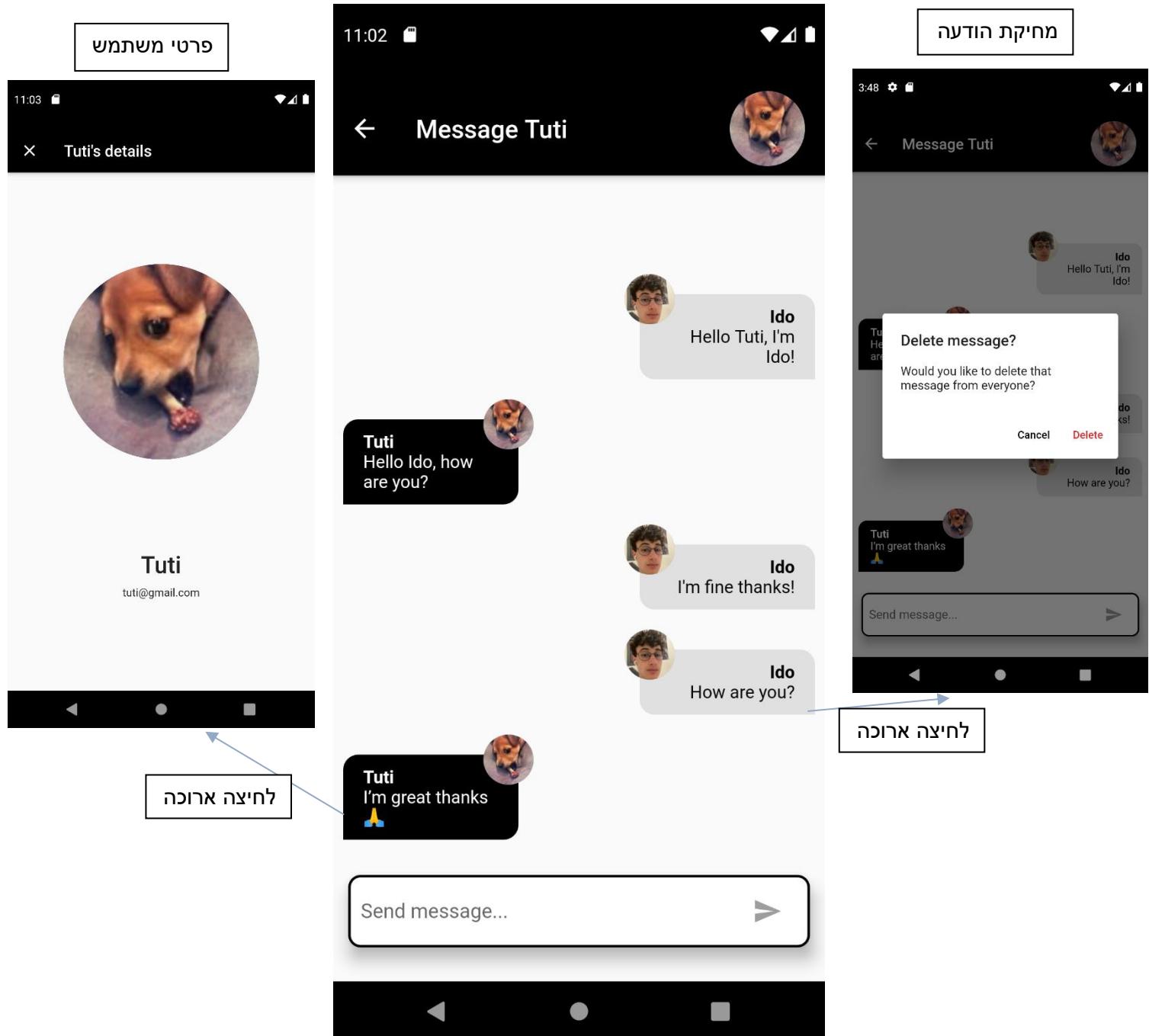
לחיצה על אחת מהאפשרויות האלו תוביל לשיחה הפרטית עם האדם אותו אנו רוצים לדבר. במקרה זהה, Tuti.



מסמך שיחה פרטית

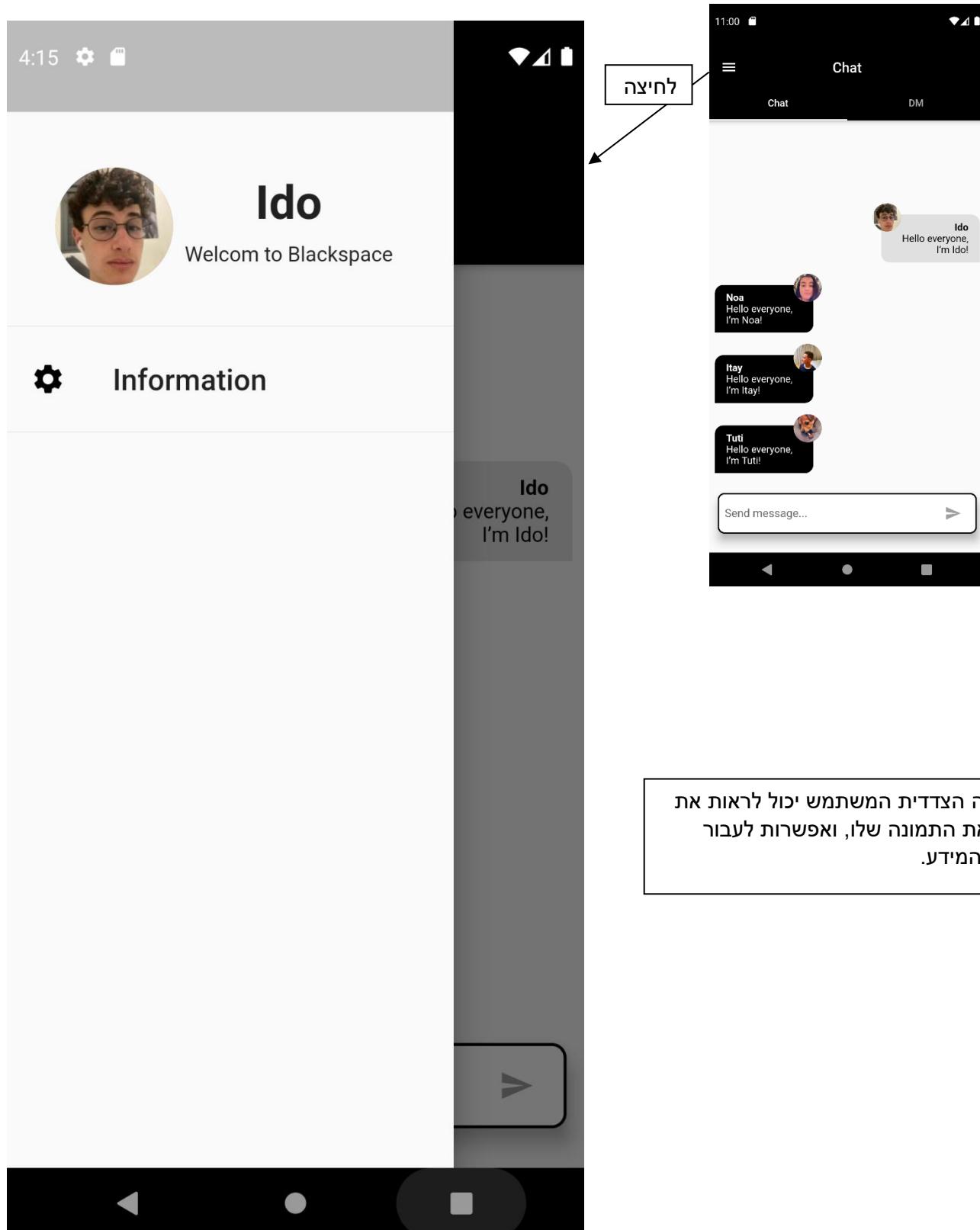
במסך זה מתנהלת שיחה בין שני משתמשים. לכל משתמש יש אפשרות לשלוח הודעות, בדיקן כמו בצד'אט הראשי.

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



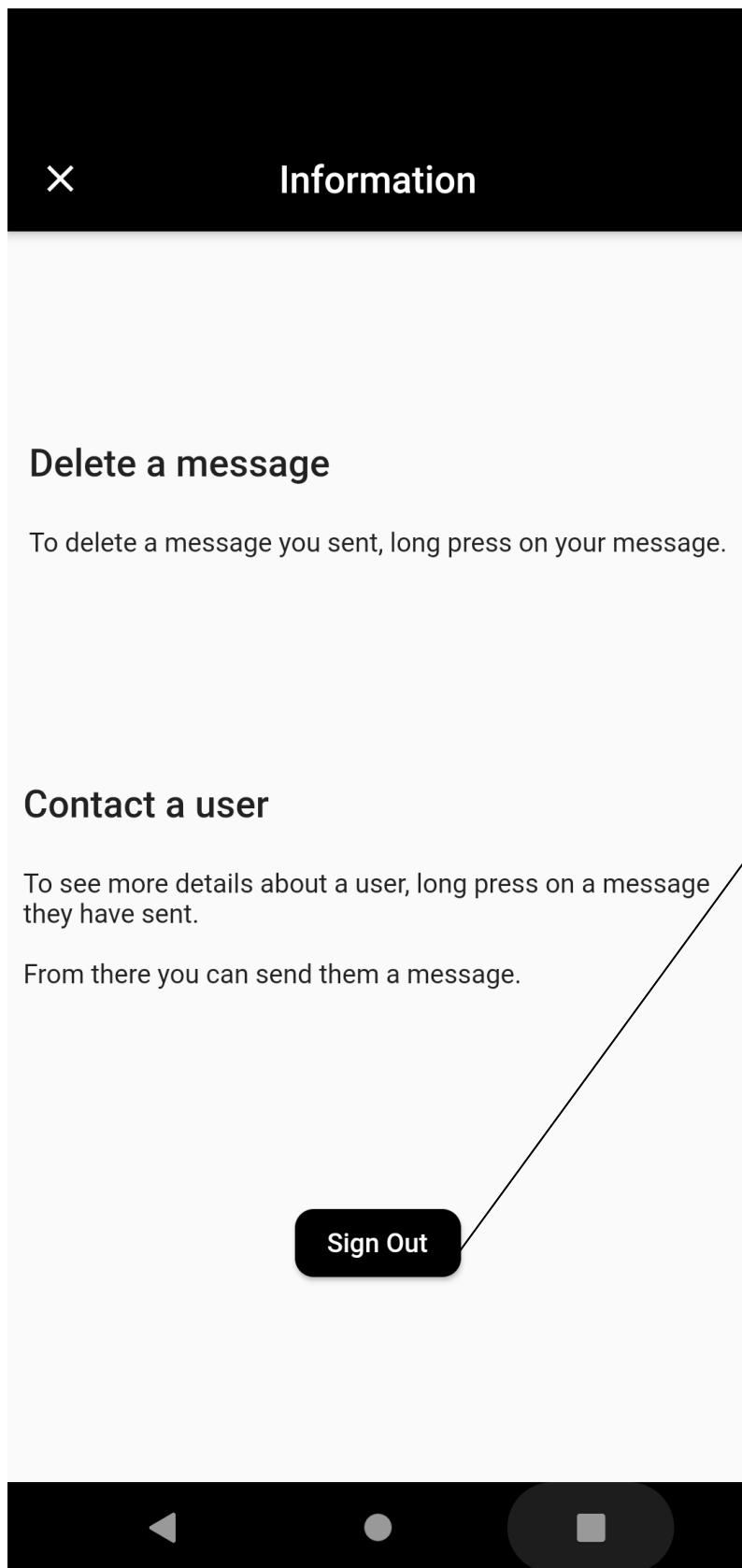
Side drawer

מה מסך הראשי ניתן לפתח מגירה צדדיות עם אפשרות שוננות. הcptor שנראה כמו "המברגר" ממוקם לצד השמאלי העליון במסך הראשי של האפליקציה.

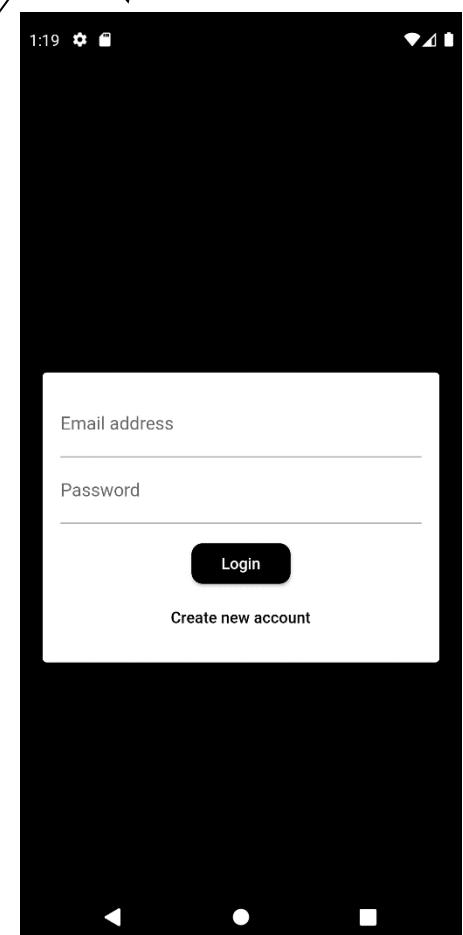


מסמך מידע

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



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



בסיס הנתונים

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

בדומה لما שכתבתי בטכנולוגיות שהן השתמשתי בפרויקט, Firebase היא פלטפורמת שירותים ובסיסי נתונים, שפותחה על ידי גוגל ליצירת 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 בעל שרת (Server), בסיס נתונים Firestore (Real time database) ובasis נתונים בזמן אמת שנקרא –

Flutter היא פלטפורמה מאוד נוחה לשימוש, עקב כך שהפלטפורמה מפותחת על ידי גוגל, כמו זו של Firebase, ישנים SDK's, Libraries ו – Packages מואוד נוחים לקישור בין הפלטפורמות הללו, מה שעזר לייעול העבודה, וליקיזור זמן הלמידה.

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

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

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

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

NoSQL

שלא בדומה לבסיס נתונים אחרים, SQL, אין מסודר במבנה טבלי, אלא מסודר במבנה של "עץ". הנתונים מסודרים בדומה יותר לאיך שהם מסודרים על המחשב, תקין בתוך תיקייה ובתוכה נמצאים הנתונים.

ב – Firestore התיקיות נקראות collections ובתוך כל אחד יכולים להיות כמה documents, בתוך כל אחד יכולים להיות שדות שונים בהם נמצא חלק מהמידע, וגם מתוך כל document אפשר להתחילה collection חדש. וכך הלאה והלאה.

בעבודה עם DB יש 4 פעולות מרכזיות CRUD

CREATE – הוספה מידע

READ – קריאת מידע

UPDATE – עדכן מידע

DELETE – מחיקת מידע

כיוון שהשתמשתי ב – Packages על מנת לקרוא, לכתוב, לעדכן ולמחוק הودעות מבסיס הנתונים, אני לא עבדתי שירותים עם NoSQL, لكن לא השתמשתי בפקודות שירות של NoSQL אלא בfonkcioot שקוראות לפונקציות אלו.

הfonקציות עברו הפעולות השונות

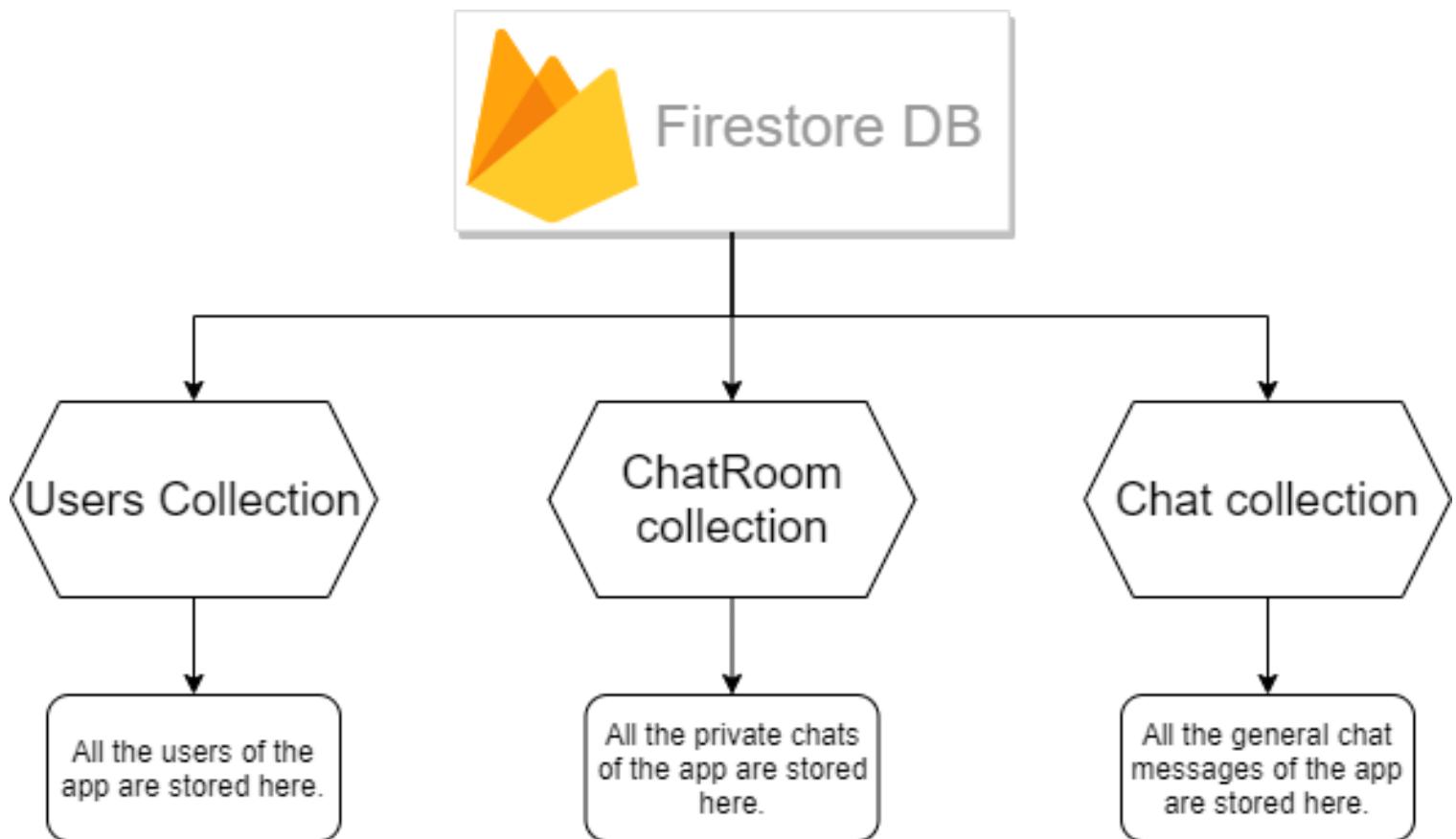
Firestore.instance.collection('collection name').add({map of data}) – Update / Create

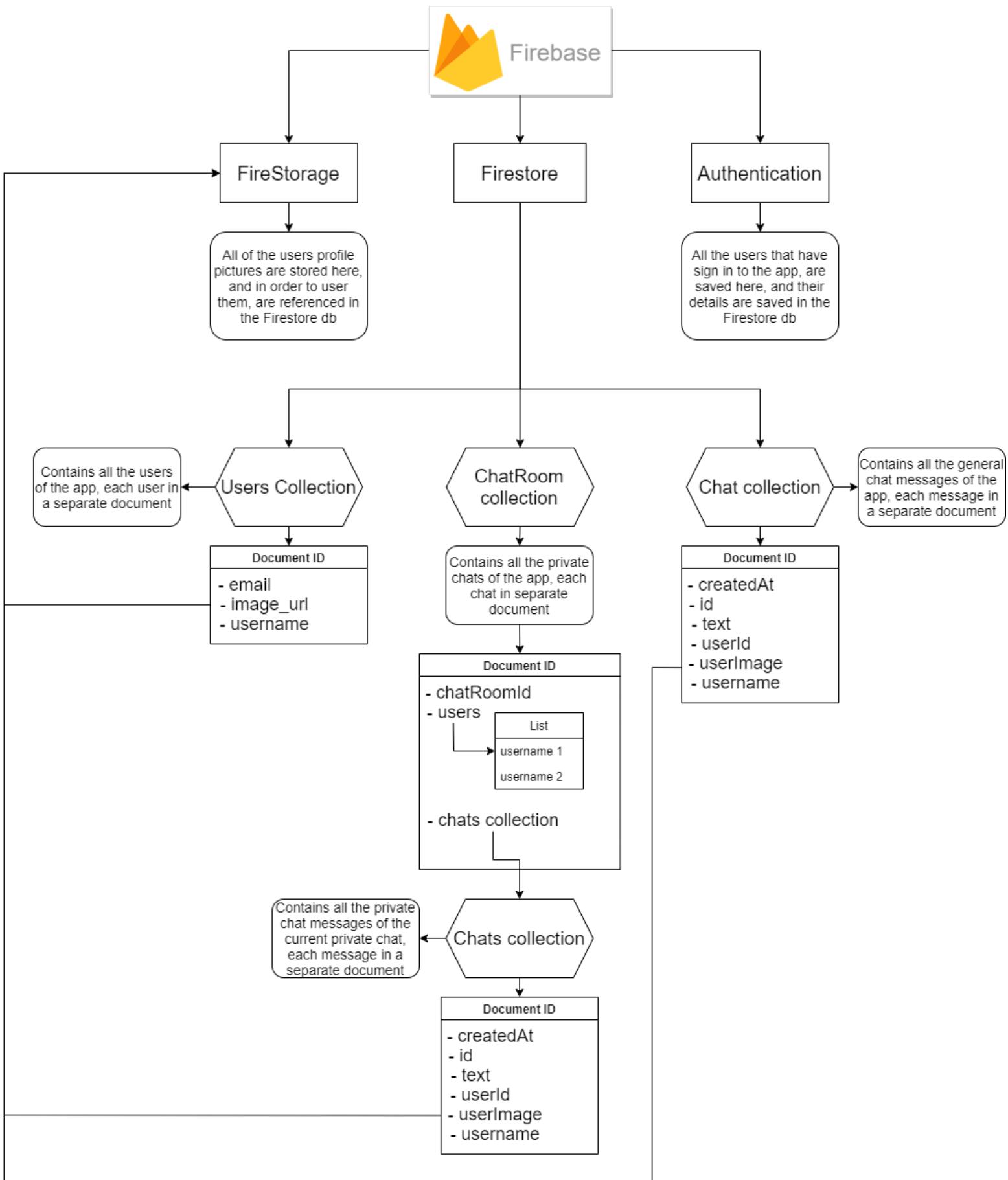
Firestore.instance.collection('collection name').document('document name').get() - Read

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 four columns: Identifier, Providers, Created, and Signed in. There are four entries in the table:

Identifier	Providers	Created	Signed in
itay@gmail.com	✉️	13 May 2021	13 May 2021
tuti@gmail.com	✉️	13 May 2021	13 May 2021
ido@gmail.com	✉️	13 May 2021	14 May 2021
noa@gmail.com	✉️	13 May 2021	13 May 2021

At the bottom right, there are buttons for 'Rows per page' (set to 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 צריך על מנת לחבר אותו אל האפליקציה, נשמר גם בחלק של Firestore ב – Collection Users. 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:

- Document ID: blackspace-7d9fd
 - Subcollection: chat
 - Subcollection: chatRoom
 - Subcollection: users
 - Document ID: 2zb6h5C6uLTceQc9vpIwfpJ0qG72
 - Fields:
 - 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-beda-ae06702af675"
 - username: "Itay"

FireStorage

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

Blackspace ▾

Storage

[Files](#) [Rules](#) [Usage](#)

gs://blackspace-7d9fd.appspot.com

Name	Size	Type	Last modified
<input type="checkbox"/> user_images/	—	Folder	—



Blackspace ▾

Storage

[Files](#) [Rules](#) [Usage](#)

gs://blackspace-7d9fd.appspot.com > user_images

Name	Size	Type	Last modified
<input type="checkbox"/> 2zb6h5C6uLTceQc9vplwfpJ0qG72.jpg	10.98 KB	image/jpeg	13 May 2021
<input type="checkbox"/> 3H1UE47TJCZ8Z0cqDGRyD3Pz023.jpg	10.36 KB	image/jpeg	13 May 2021
<input type="checkbox"/> 40tVaQy69TYUXx4ygBxU7xbykdd2.jpg	9.84 KB	image/jpeg	13 May 2021
<input type="checkbox"/> Zg8FDw1EKeZ74ELXv5pxrqASB502.jpg	12.46 KB	image/jpeg	13 May 2021

3H1UE47TJCZ8Z0cqDGRyD3Pz023.jpg



Name: 3H1UE47TJCZ8Z0cqDGRyD3Pz023.jpg
Size: 10,607 bytes
Type: image/jpeg
Created: 13 May 2021, 21:44:57
Updated: 13 May 2021, 21:44:57
File location:
Other metadata:

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%2F2zb6h5C6uLTeeQc9vplwfpJ0qG72?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"
  
```

Blackspace ▾

Cloud Firestore

Data Rules Indexes Usage

chatRoom > Itay_Ido > chats

+ Add document + Start collection + Add document

Itay_Ido > chats > lifcHg8zIE9IEAjby2yc

Noa_Ido
Tuti_Ido

+ Add field
chatRoomId: "Itay_Ido"

users
0 "Ido"
1 "Itay"

Cloud Firestore location: eur3 (europe-west)

Blackspace ▾

Cloud Firestore

Data Rules Indexes Usage

itay_Ido > chats > lifcHg8zIE9IEAjby2yc

+ Start collection + Add document + Start collection

chats > lifcHg8zIE9IEAjby2yc > lifcHg8zIE9IEAjby2yc

yUw70avGj4IU9DwEsaGW

+ Add field
chatRoomId: "Itay_Ido"

users
0 "Ido"
1 "Itay"

createdAt: 13 May 2021 at 21:58:06 UTC+3
id: 13 May 2021 at 21:58:06 UTC+3
text: "Hello Ido, nice to meet you 😊"
userId: "2zb6h5C6uLTceQc9vplwfpJ0qG72"
userImage: "https://firebasestorage.googleapis.com/v0/b/blackspace-7d9fd.appspot.com/o/user_images%2F2zb6h5C6uLTceQc9vplwfpJ0qG75?alt=media&token=b586c160-3937-45b2-bed4-ae06702af675"
username: "Itay"

Cloud Firestore location: eur3 (europe-west)

מדריך למפתח

הסבר כללי על קודuproject

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

את הפרויקט כתבת בשפה Dart שהיא השפה בה נכתב Packages. והשתמשת Flutter כאלו ואחרים על מנת לקצר את תהליך כתיבת הקוד ולהפוך אותו לכמה שיותר יעיל.

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

שימוש ב - accessories של הטלפון

באפליקציה השתמשתי במכשיר המצלמה של הטלפון, משומם שבעת הרשמה לאפליקציה רציתי שככל משתמש יוסיף

תמוננה של עצמו, מה שהפוך את האפליקציה ליותר חברותית.

הסבר על הקוד של השימוש במכשיר המצלמה נמצא בהמשך.

תכנות מונחה עצמים

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

Flutter במאחורי בנייה שנקרוות 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 כתבת קוד שרת הייתה צריך להשתמש בשרת אחר. עבור כך השתמשתי בשרת קיימ 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 כתובות מקומית 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 בשביל 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 שכל הקריאה 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 ייחידת עבודה פועלה בנפרד משרשור היישום הראשי ומודיעת לשרשור המתקשר על השלמתה, כישלונה או התקדמותה.

לדוגמא, האפליקציה לא נתקעת כל פעם כאשר מבצעים פעולה שלוקחת זמן באינטרנט, אלא היא רצתה 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/IdoHirsh0/Blackspace>

הסבר כללי על תיקיות הקבצים של הפרויקט

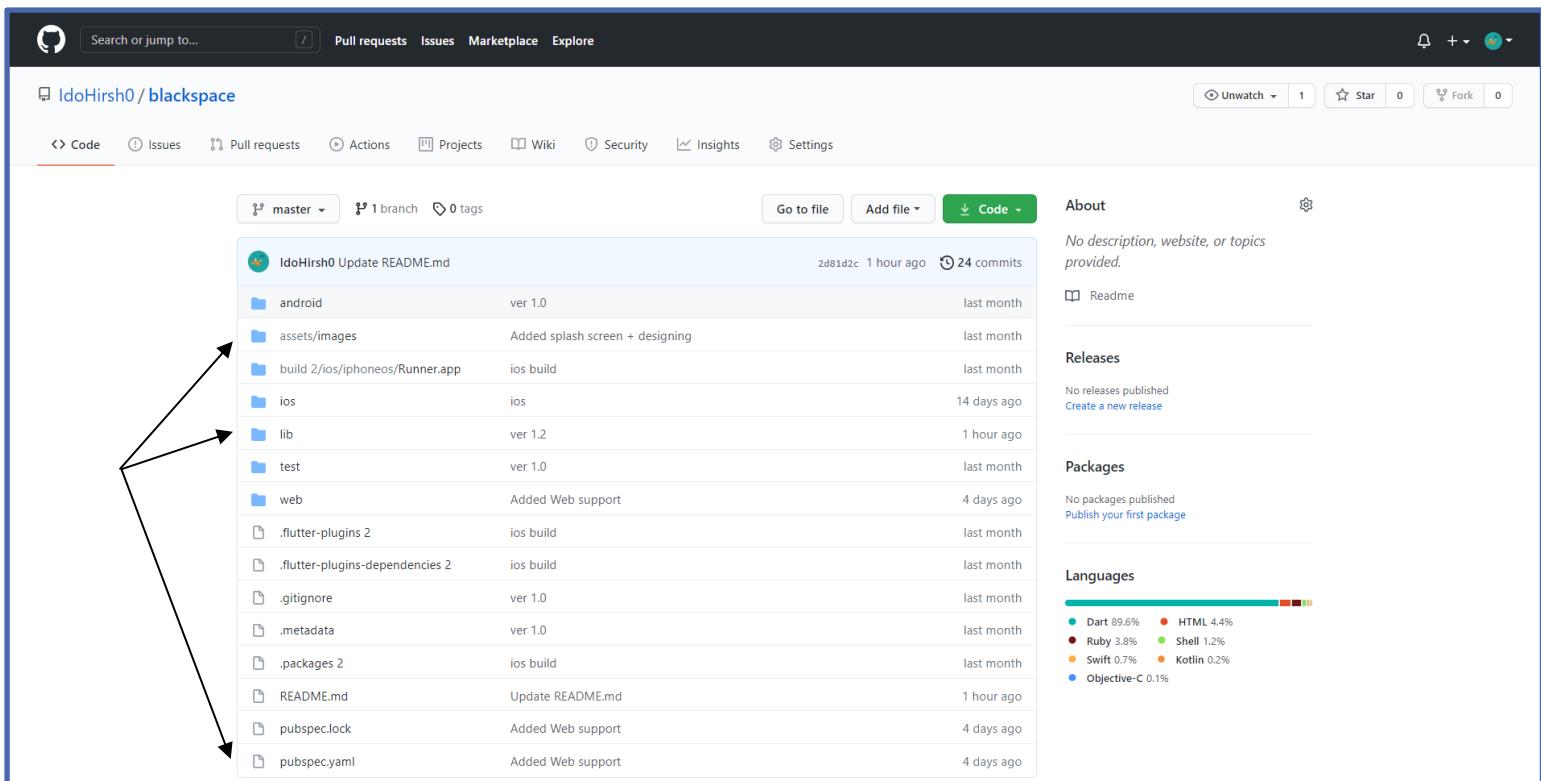
בתוך תיקיות הקבצים של הפרויקט, אותה ניתן להוריד מה – GitHub, ישנו מספר תיקיות וקבצים שונים.

בתיקייה `lib` נמצאים הקבצים בשפת Dart שניי כתבתה.

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 `IdoHirsh0/blackspace`. The 'Code' tab is active. The commit history is as follows:

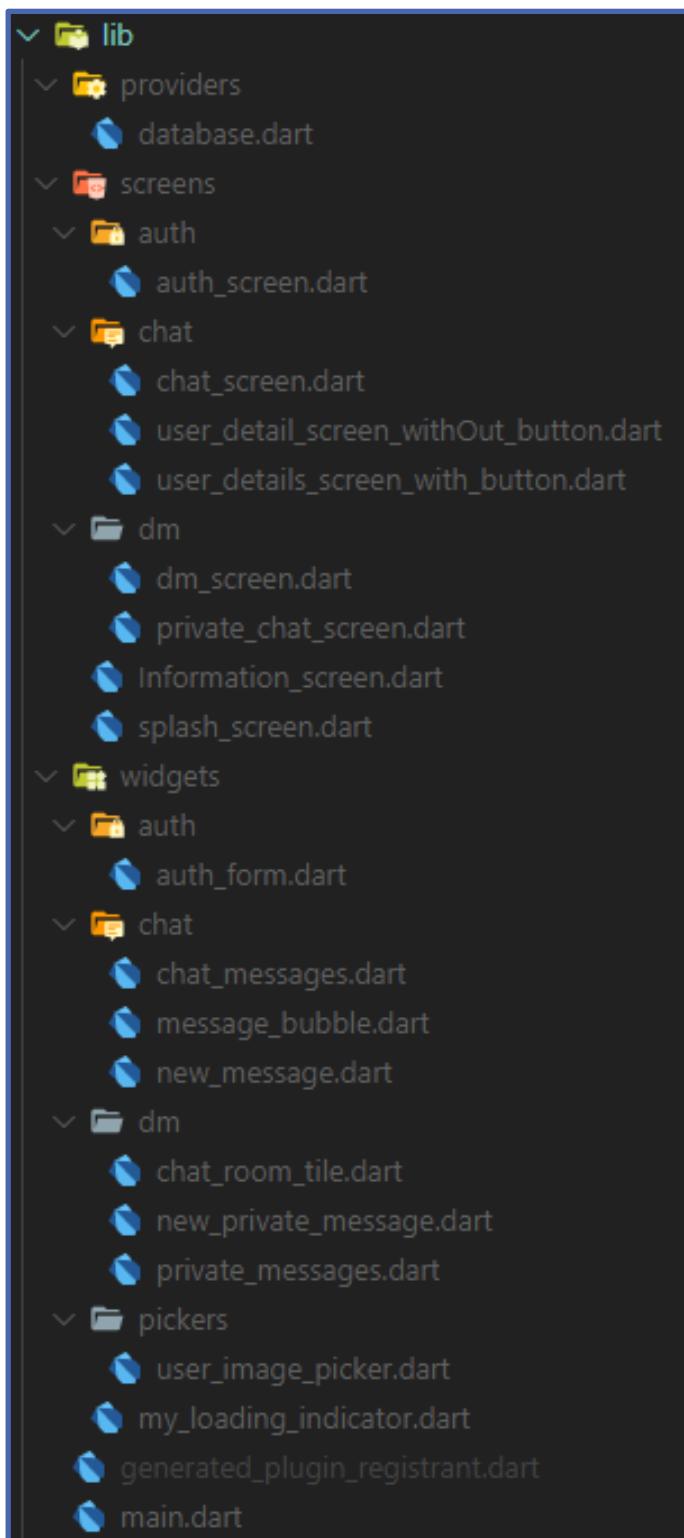
- IdoHirsh0 Update README.md (2d81d2c, 1 hour ago, 24 commits)
- android ver 1.0 (last month)
- assets/images Added splash screen + designing (last month)
- build 2/ios/iphoneos/Runner.app ios build (last month)
- ios ios (14 days ago)
- lib ver 1.2 (1 hour ago)
- test ver 1.0 (last month)
- web Added Web support (4 days ago)
- .flutter-plugins 2 ios build (last month)
- .flutter-plugins-dependencies 2 ios build (last month)
- .gitignore ver 1.0 (last month)
- .metadata ver 1.0 (last month)
- .packages 2 ios build (last month)
- README.md Update README.md (1 hour ago)
- pubspec.lock Added Web support (4 days ago)
- pubspec.yaml Added Web support (4 days ago)

On the right side, there are sections for About, Releases, Packages, and Languages. The Languages section shows a bar chart with Dart at 89.6%, Ruby at 3.8%, Swift at 0.7%, and others at lower percentages.

הסבר על התיקייה **lib**

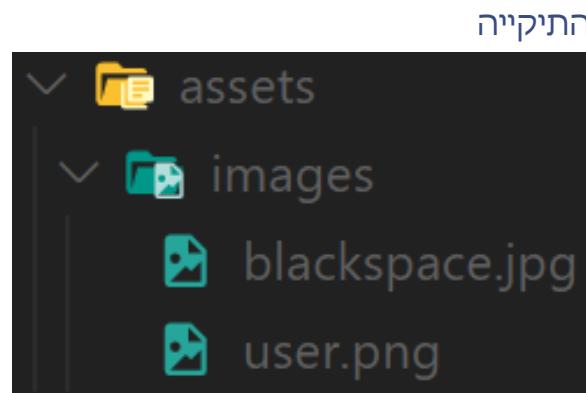
בתיקייה זו נמצאים כל קבצי הפרויקט אוטם אני כתבתי בשפה Dart. כל הקבצים בתיקייה זו הם קבצי הלוקוּן הקיימים הציגו במסך האפליקציה – "GUI", התקשורת עם בסיס הנתונים, וקשרים בתוך האפליקציה עצמה, כמו תכונות מונחה עצמים.

קובץ הרצה 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 מכובץ בו מגדירים את ה – packagespackages בהן משתמשים בפרויקט.

[להשתמש ניתן למצוא כאן](https://pub.dev/packages) - Packages

```
{
  pubspec.yaml
  1 name: blackspace
  2 description: A new Flutter project.
  3
  4 publish_to: 'none'
  5
  6 version: 1.0.0+1
  7
  8 environment:
  9   sdk: "≥2.7.0 <3.0.0"
 10
 11 dependencies:
 12   flutter:
 13     sdk: flutter
 14
 15   firebase_core: ^1.2.0
 16   firebase_auth: ^1.2.0
 17   cloud_firestore: ^2.2.0
 18   firebase_storage: ^8.1.0
 19
 20   image_picker: ^0.7.5+3
 21
 22   cupertino_icons: ^1.0.2
 23
 24 flutter:
 25   uses-material-design: true
 26
 27 assets:
 28   - assets/images/
}
```

כל ה –
 Packages –
 בהן השתמשתי

קישור לתקינה assets

קובץ שימוש במכשירמה של הטלפון

בunnyi 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

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

ב – Dart ו- **Flutter** על מנת ליצור widget חדש, צריך לרשף (תכונות מונחה עצמים) מאחד מшибיו `classes` – משתמשים בו כאשר רוצים ליצור widget שלא משנה דבר על המסר. כלליים: 1. `StatelessWidget` – משתמשים בו כאשר רוצים ליצור widget שמשנה משהו על המסר.

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

במקרה הזה השתמשתי ב – `StatefulWidget` מכיוון שארצה להציג את התמונה הנבחרת בסופו של דבר על המסך, ובכך אשנה את מה 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 StatelessWidget {
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

באפשרות לדעת כמה זמן ייקח למשתמש לבחור תמונה, אם בכלל) מתבצעת בחירת התמונה של המשתמש מהמכשירמה.

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

בשורה 19 – 23, אני מפעיל את הפעולה מה – `getImage()` 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` (catch) מדפס את השגיאה שנוצרה. ולאחר מכן קורא לפועלה `shutter()` את התמונה ל – `class` `Firebase`.
SSH מושר את התמונה ב `Cloud Storage`.

חולקו השני של הקובץ הוא – GUI

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

השתמשתי כאן בכמה `widgets` בסיסיים שmaguireים עם flutter:

1. `Column` – מאפשר לשים כמה `widgets` שונים אחד ליד השני (הרשימה `[children]`).
2. `CircleAvatar` – מאפשר לשים תמונה או רקע אחר בתוך מסגרת עגולה.
3. `FileImage` – מאפשר לשים תמונה מסווג `File` (כמו שקיבלו מfpfella()) מתוך מסוג `File`.
4. `AssetImage` – מאפשר לשים תמונה שנמצאת מקומית בקבצי הפרויקט.
5. `FlatButton` – כפתור על המסך, בעל הפעורי `onPressed` אותו אפשר להגדיר לפועלה אותה רוצים להציג כאשר נלחץ הכפתור.
6. `Icon` – מאפשר להציג חסוך כלשהו על המסך.
7. `Text` – מאפשר לכתוב טקסט על המסך.

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

כאשר נלחץ הכפתור, נקראת הפיעולה `_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 }
```

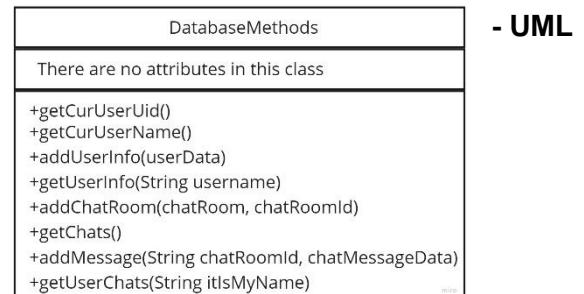
החלק ממוקם הרשמה אותו קוד קובץ זה, `,user_image_picker.dart` מייצג.



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 לתקשרות 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
main.dart database.dart - Visual Studio Code
EXPLORER
OPEN EDITORS
BLACKSPACE
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 defines a class "DatabaseMethods" with several methods for interacting with a Firestore database. Annotations with arrows pointing to specific lines of code provide explanations for each method's purpose:

- An arrow points to the first line of the "getCurUserId" method, explaining it returns the current connected user's ID.
- An arrow points to the first line of the "getCurUserName" method, explaining it returns the current connected user's name.
- An arrow points to the first line of the "addUserInfo" method, explaining it adds user information to the "users" collection.
- An arrow points to the first line of the "getUserInfo" method, explaining it returns the info of a user by its username.

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 |

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

```

database.dart - Visual Studio Code

```

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 | Stream<QuerySnapshot> getUserChats(String itIsMyName) {
70 |   return Firestore.instance
71 |     .collection("chatRoom")
72 |     .where('users', arrayContains: itIsMyName)
73 |     .snapshots();
74 |
75 |
76 |

```

רפלקציה

תחושתי מהעבודה על הפרויקט

העבודה על הפרויקט הייתה מהנה מאוד עבורי. עצם כך שהצלחתו למש את מה שרציתי, אפליקציה שראיתי בסדרה בטלוויזיה בגרסה משלי, גורם לי לאושר ולחושת סיפוק רבה.

麥意oun שהשנה הייתה שנת 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 התקופה הזאת, לא הייתה יכולה לפתוח את מה שפיתחתי כיום.

כלים שקיבلت ואני איתן להמשך

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

עוד kali שקיבلت מהפרויקט הוא שיתוף פעולה, כאשר נתקעתי בעיה שלא הצלחתי לפתור, פניתי לחבר לי כתיה שעסוקו בנושא דומה, ואלו עזרו לי להגיא לפתרון.

אתגרים שעמדו בפניי במהלך הפרויקט

בנוספ' לאתגר של למידת נושא חדש לגמרי בכוחות עצמי, כפי שתכתבתי במאמר, במהלך הפרויקט עלו עוד כמהאתגרים.

ראשית, אני לוקח חלק בתוכנית MAX, גם אליה דיברתי במאמר, והיה לי קשה להחליט בין אם להמשיך או תום ואולי לא להיות עם פרויקט גמור בסופו של דבר, או להיפרד מהם בינו לבין מנת לעבוד על הפרויקט שלי, Blacksphere. לאחר התחבויות רבות, ועזרה הרבה של ניר, החלטתי לעבוד על הפרויקט שלי.

בנוספ', רציתי לפתח את האפליקציה שלי ל – Android ו – iOS במקביל, لكن גם בחרתי לעבוד עם Flutter.ammen ההוספה של iOS לבסיס הנתונים, ו קישור בין מכשירי 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

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

מסקנה נוספת היא שאני אוהב לעבוד בשיתוף פעולה ושבועתיד נראה איהנה לעבוד כחלק מצוות. כאשר נתקלתי בעיה שלא הצלחתי לפתור, גם לא בעזרתו של האינטרנט, פניתי לחבר לי כתיה שעסוקו בנושא דומה ועזרתי בהם.

מה הייתה עשו אחרת אילו הייתה מתחילה היום

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

בנוספ' הייתה אומר לעצמי בשלב מוקדם יותר, שאני נראה לא אסיים את הפרויקט עם MAX בזמן, כך היה מתפנה לי יותר זמן לעבוד על הפרויקט הזה, ואולי הייתה יכולה לשפר אותו אפילו יותר, ולהוסיף את הפיצרים שלא הספקתי להוסיף.

מה היה יכול להפוך את עבודתי ליעילה יותר

אילו הייתי מצליח לניר את הפרויקט לעתים יותר תכופות (וכל פעם מתקדם קצת פחות),(Clomar כל שבוע בערך במקום כל שלושה שבועות, עצם זה שיש דד-ליין שוצר להגish 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 אولي הדבר היה מייעל את העבודה, והיהתי מנצל את זמני טוב יותר.

תכונות שהייתי רוצה להוסיף לפרויקט

ישנן מספר תכונות שהייתי רוצה להוסיף לפרויקט אם היה לי את הזמן לכך.

ראשית, ארצתה לאפשרות לצירט קבוצות של מספר משתמשים באפליקציה, מלבד השיחות הפרטיות והצ'אט הכללי.

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

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

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

נהני 😊

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 בספר פרויקט זה ב – **GitHub** –

<https://github.com/IdoHirsh0/Cyber-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.

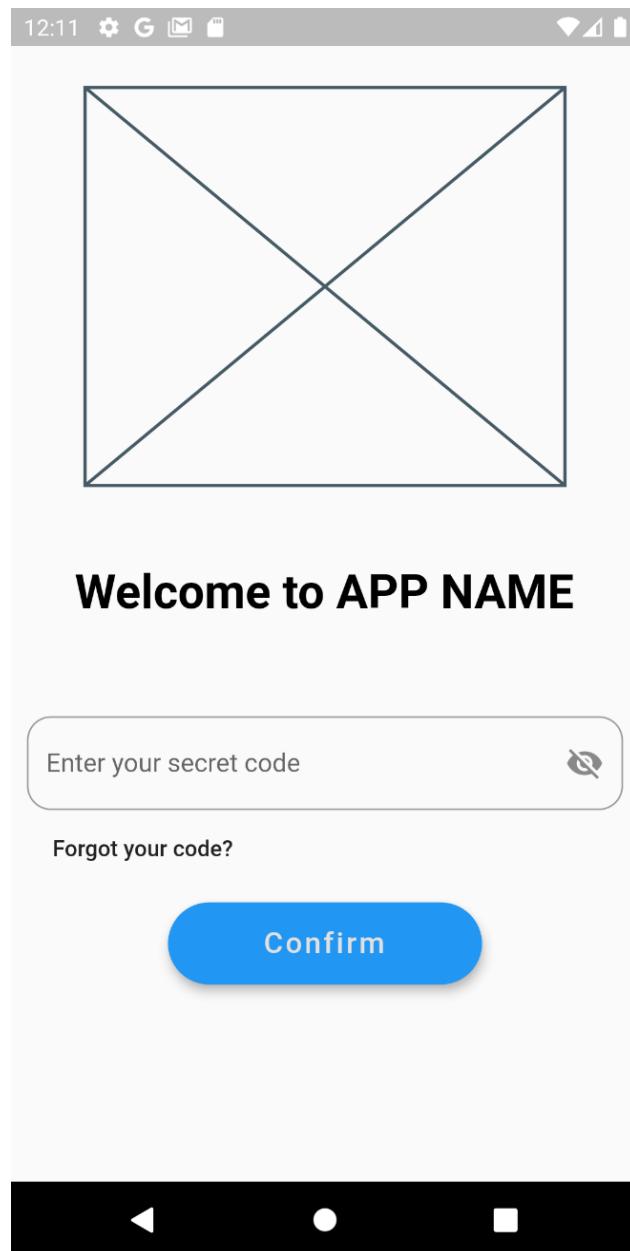
קישור: -
<https://drive.google.com/drive/folders/1-v18DgIvWqSaLbt27HRZ5W8qNwRiHOD4?usp=sharing>

אפליקציות שונות שבניתי במלר הלמידה

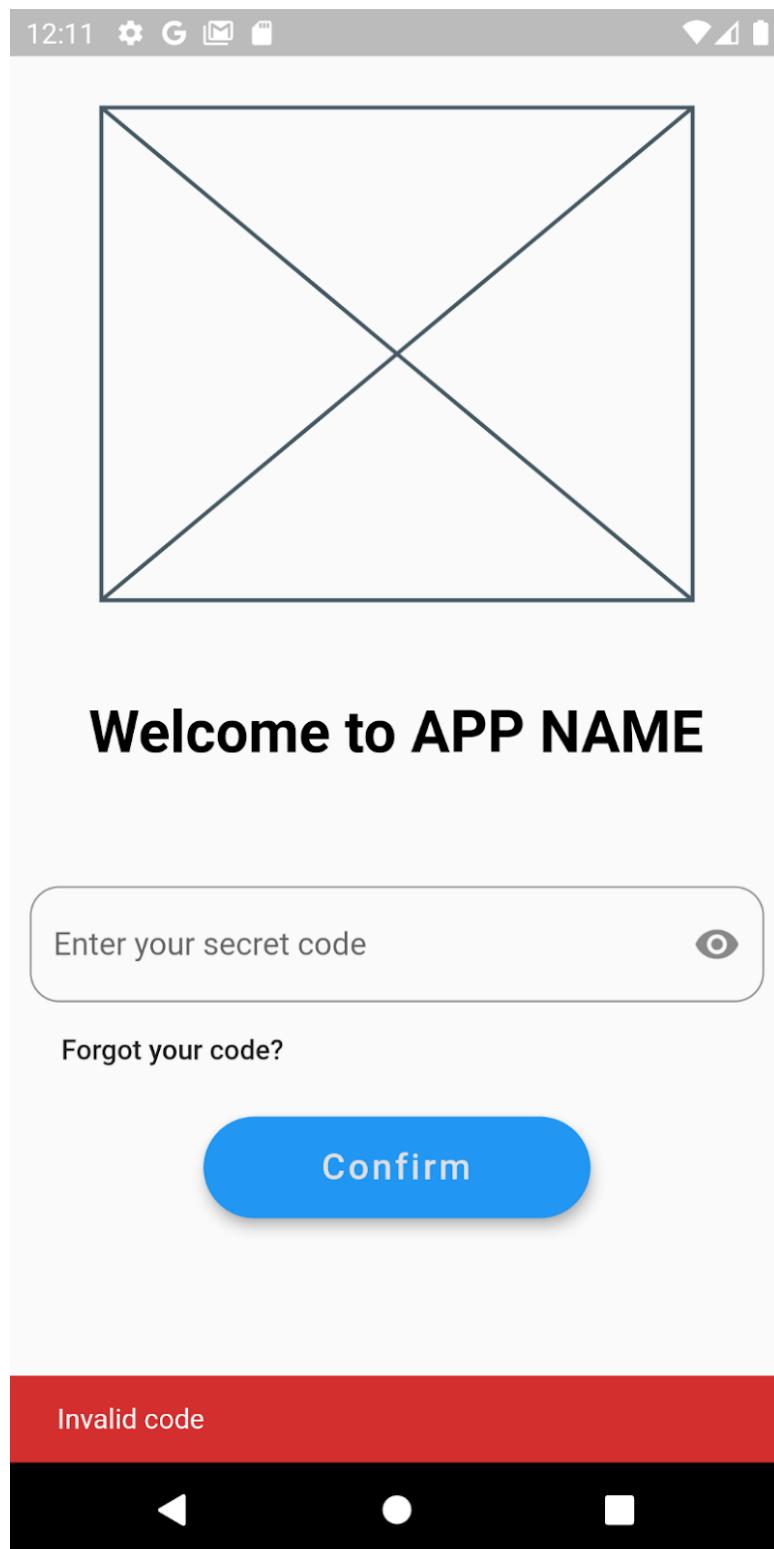
אפליקציה לשליחת מסרים – MAX

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

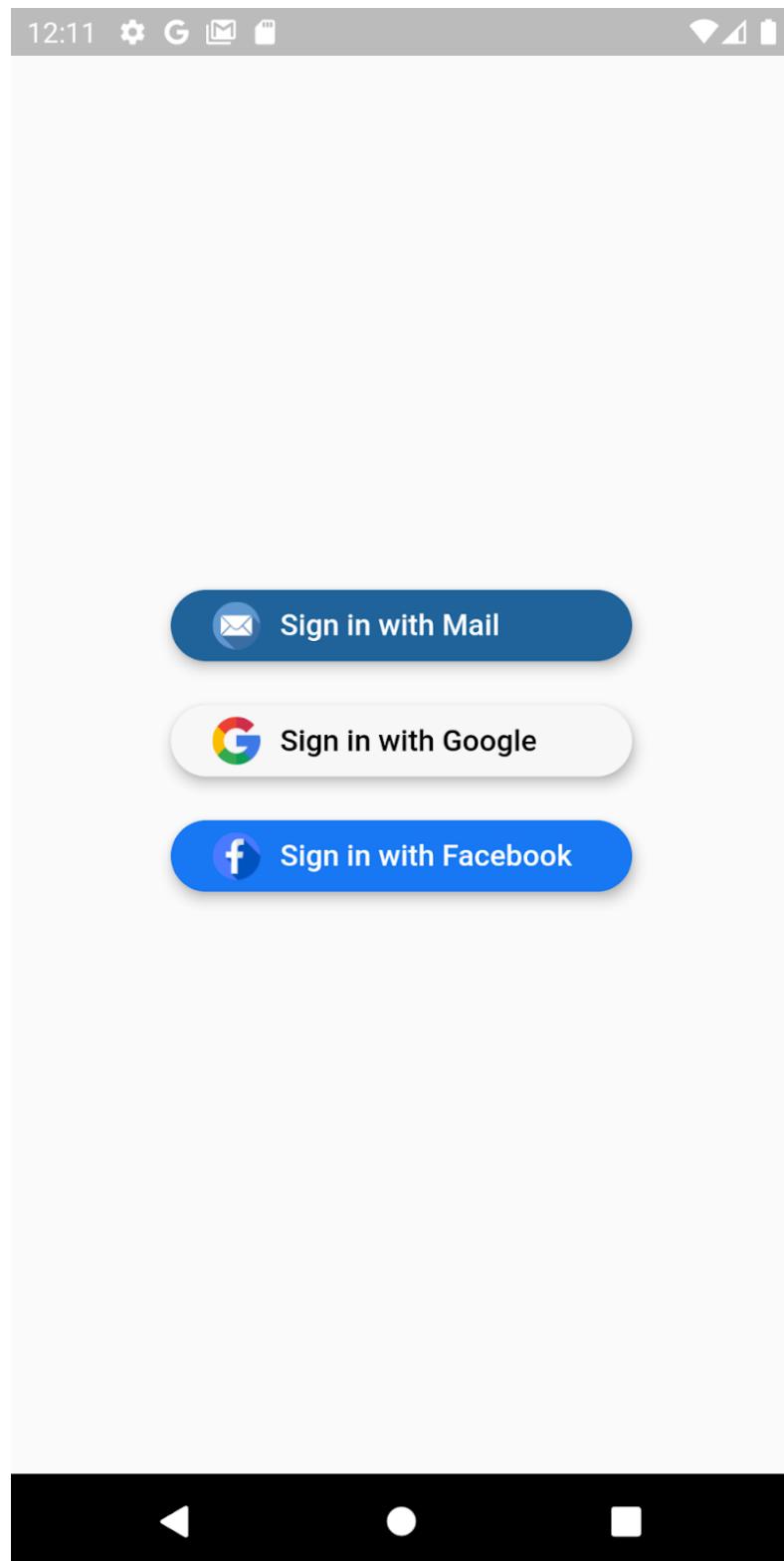
מסך ראשון בו מזינים את הקוד הסודי שאנו בתוכו אפליקציה ניתן לאותו מועד כדי שייתן לעובדים שלו. אך רק אנשים בעלי גישה לקוד זה יהיו יכולים להתחבר למערכת ותיזכר סוג של חמהה בה כולם מכירים את כולם.



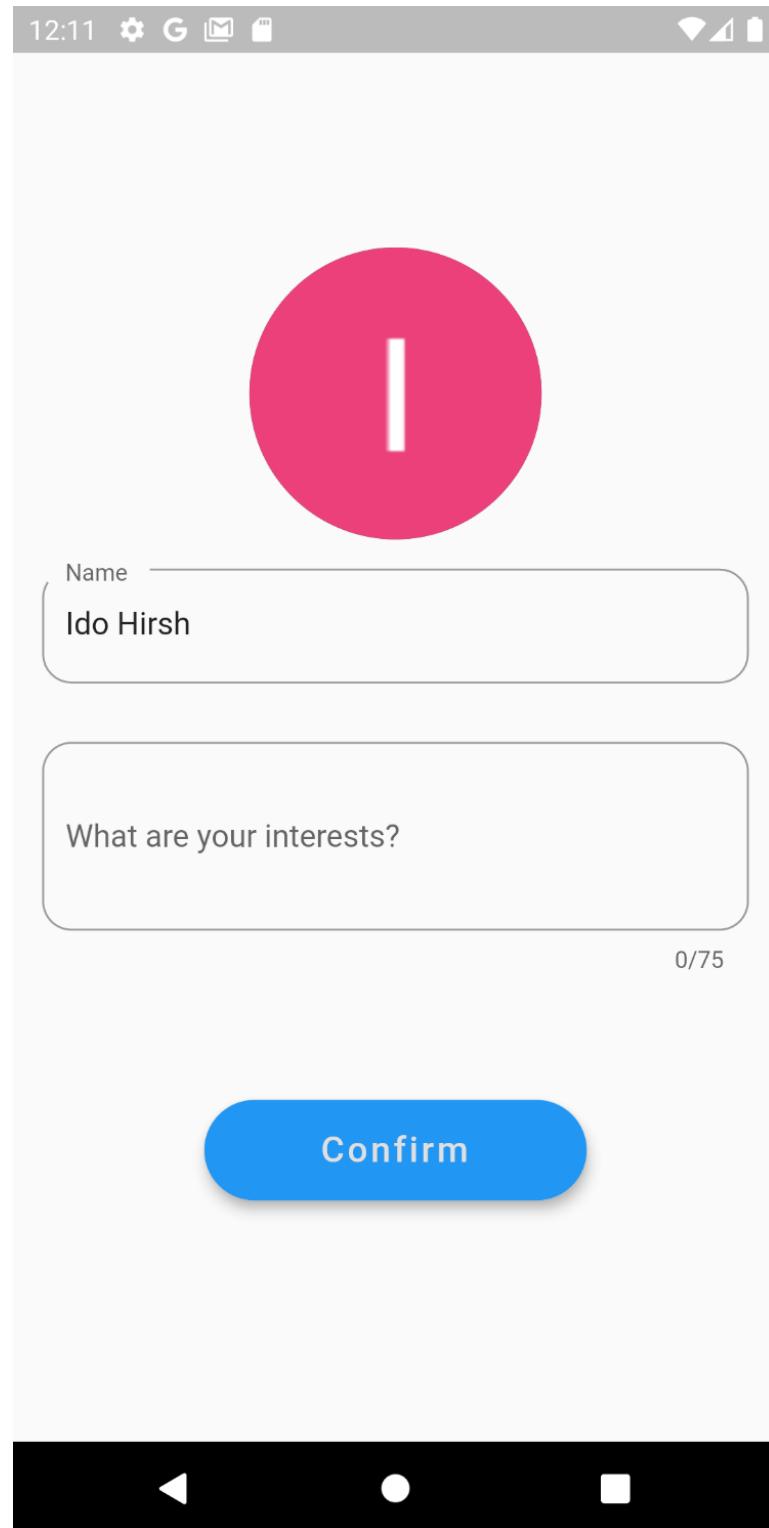
כמובן שיש בדיקה של הקלט אל מול databases בו מאוחסנים הקודים הסודיים. אם הוזן קוד שגוי תופיע הודעה:



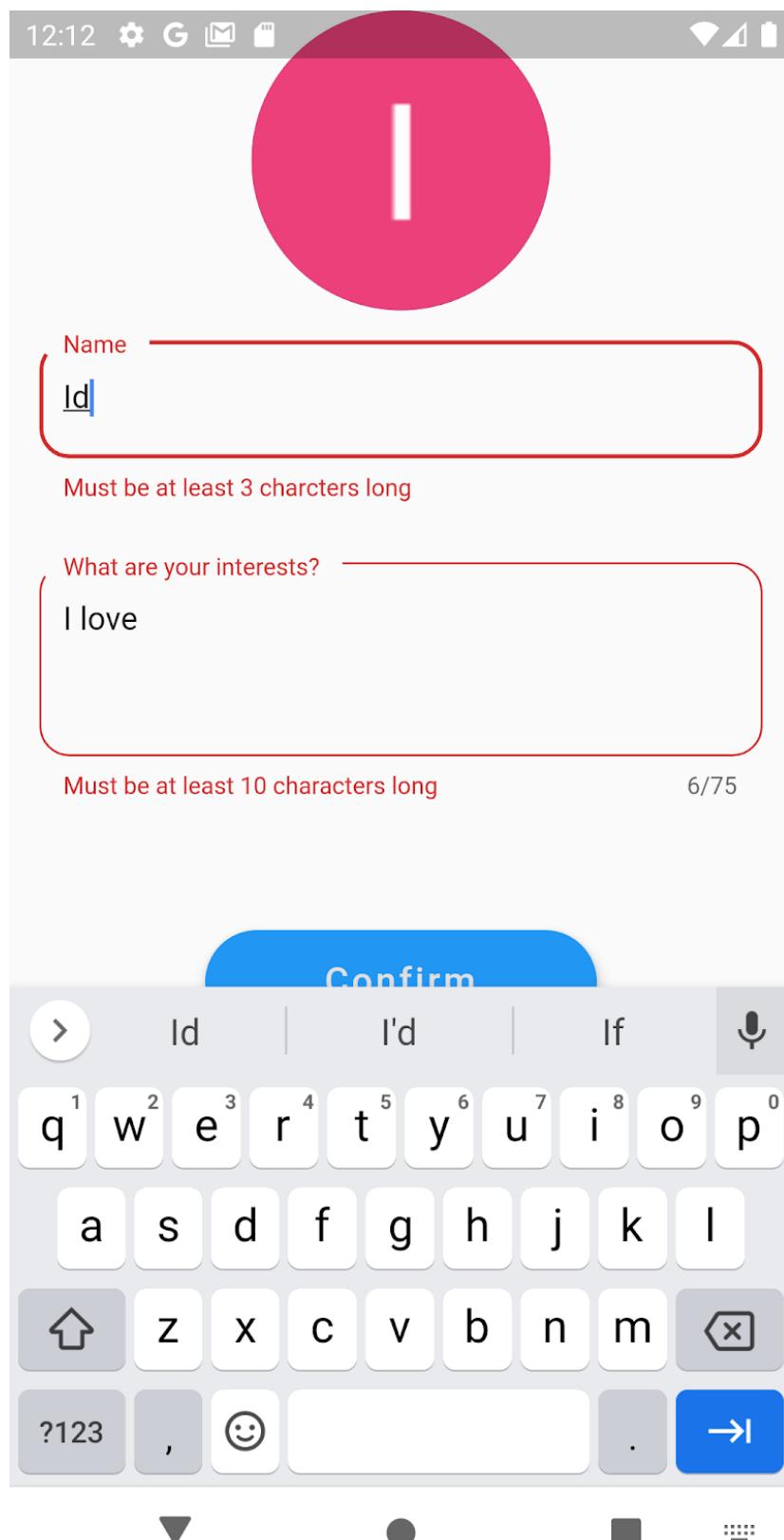
לאחר ההתחברות עם הקוד הסודי של האפליקציה שלנו, יש אופציה ל-3 סוגי התחברויות, גугл פייסבוק
ואימייל וויסמה.
בינתיים אני עובד רק עם ההתחברות של גול.



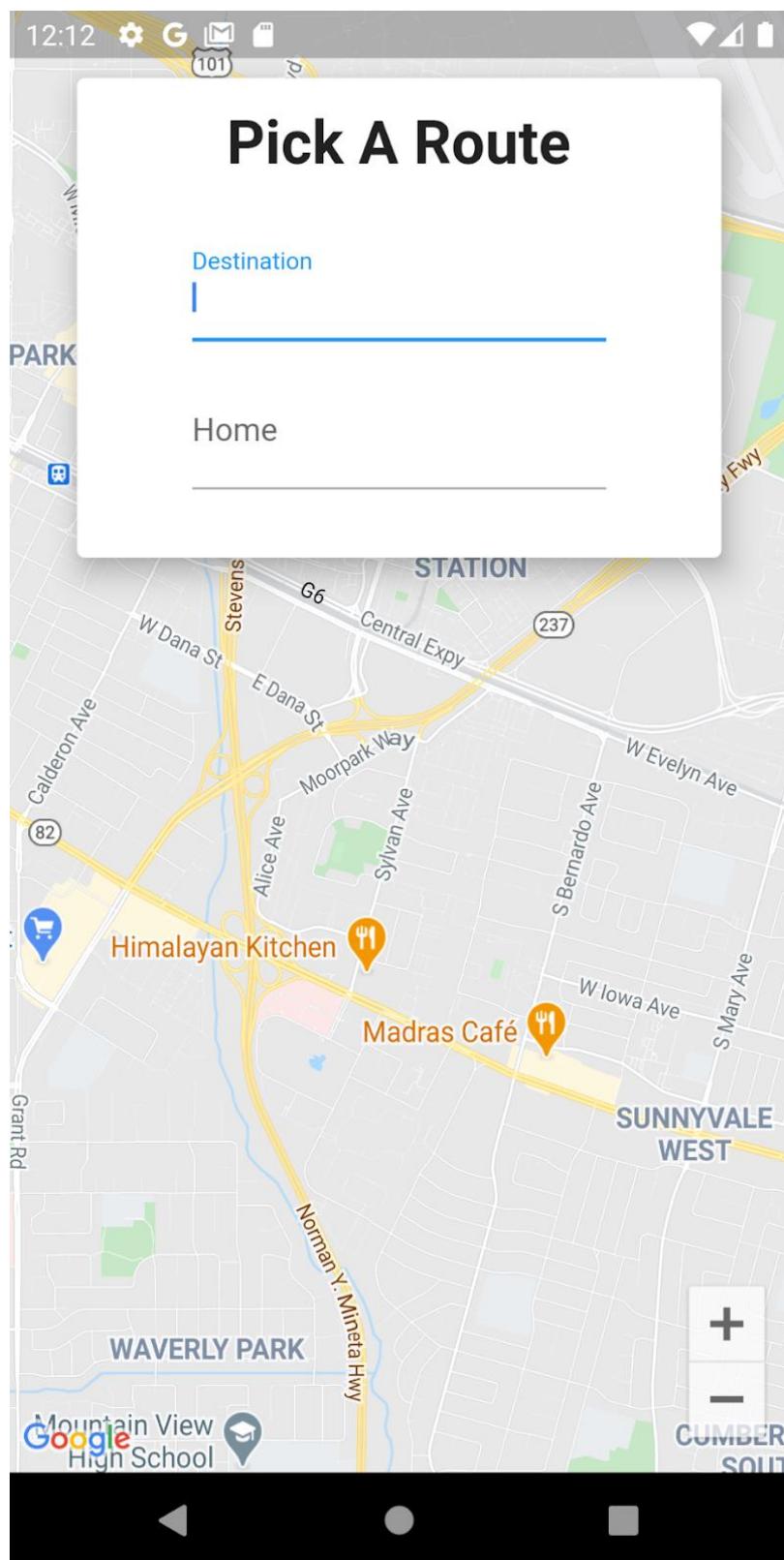
לאחר ההתחברות עם גוגל, אני לוקח את פרטי המשתמש מגוגל, כדוגמת שם משתמש, תמונה, אימייל ועוד. אבל ניתן למשתמש את האפשרות לשנות את הפרטים האלה באפליקציה. גם מקבל מהמשתמש תיאור קצר על עצמו, על מנת שഫופיל של כל אחד יהיה לפחות איזשהו משפט קצר. כך יהיה אפשר להכיר קצת יותר את הבן אדם.



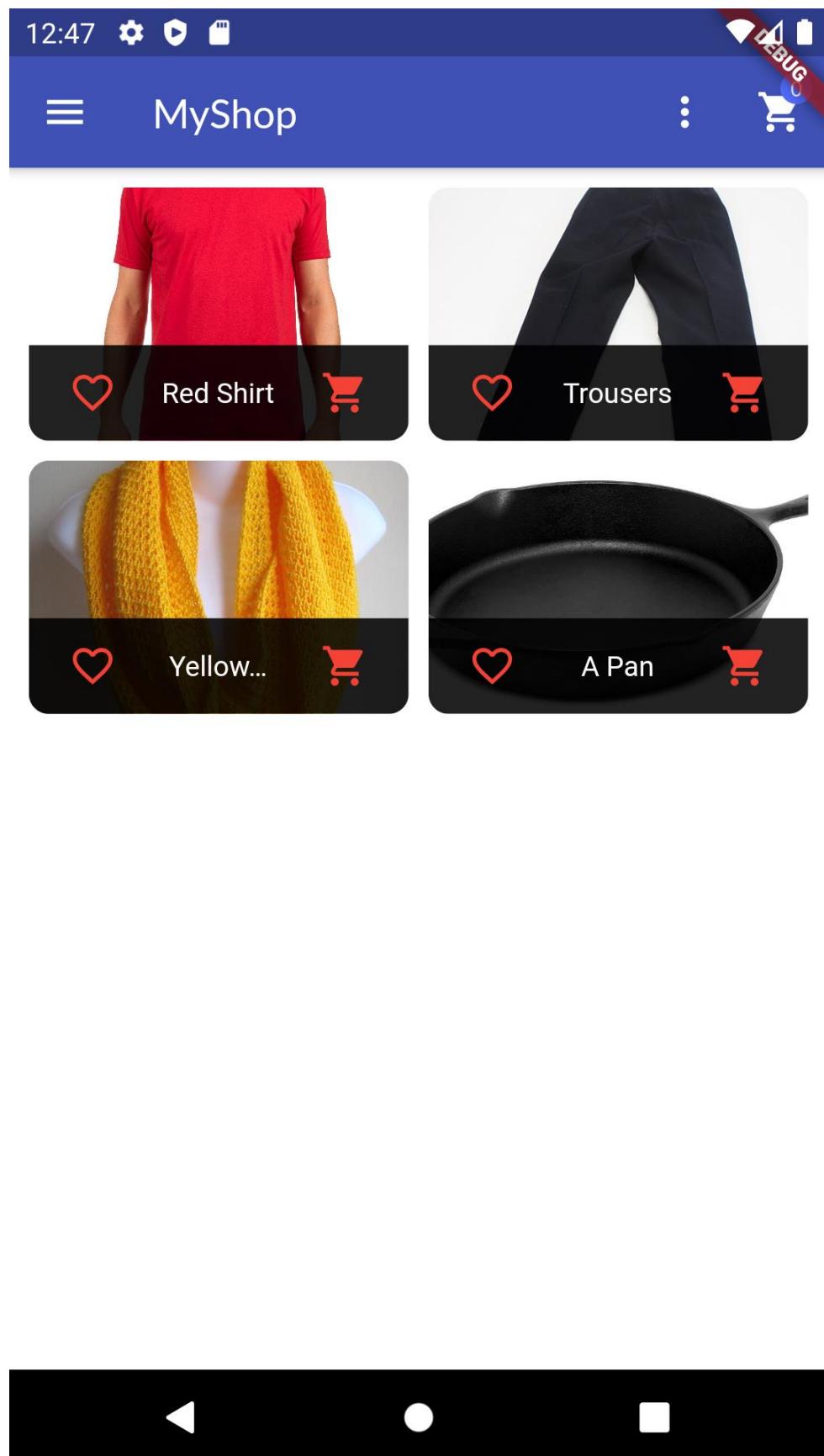
כמובן שיש בדיקה של הקלט...

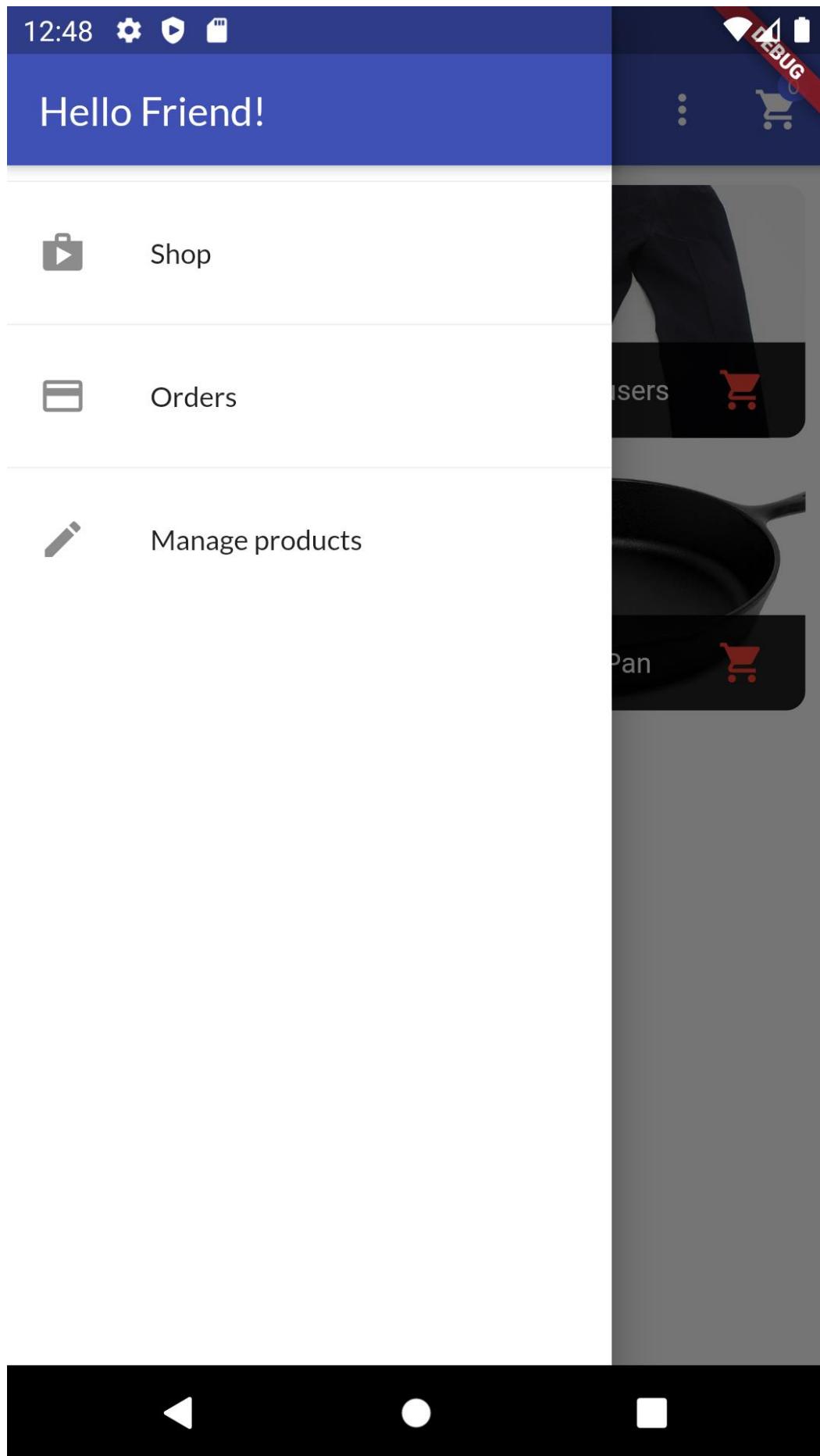


לאחר מכון מגיעים למסך בו המשתמש בוחר את נתיב ה-"default" שלו. נבחר על ידי גול מפו (ممש אפשר להציג את המפה ולראות מה סביר). כמו גם שם יהיה לי זמן אוסף בחירה על המפה עצמה).

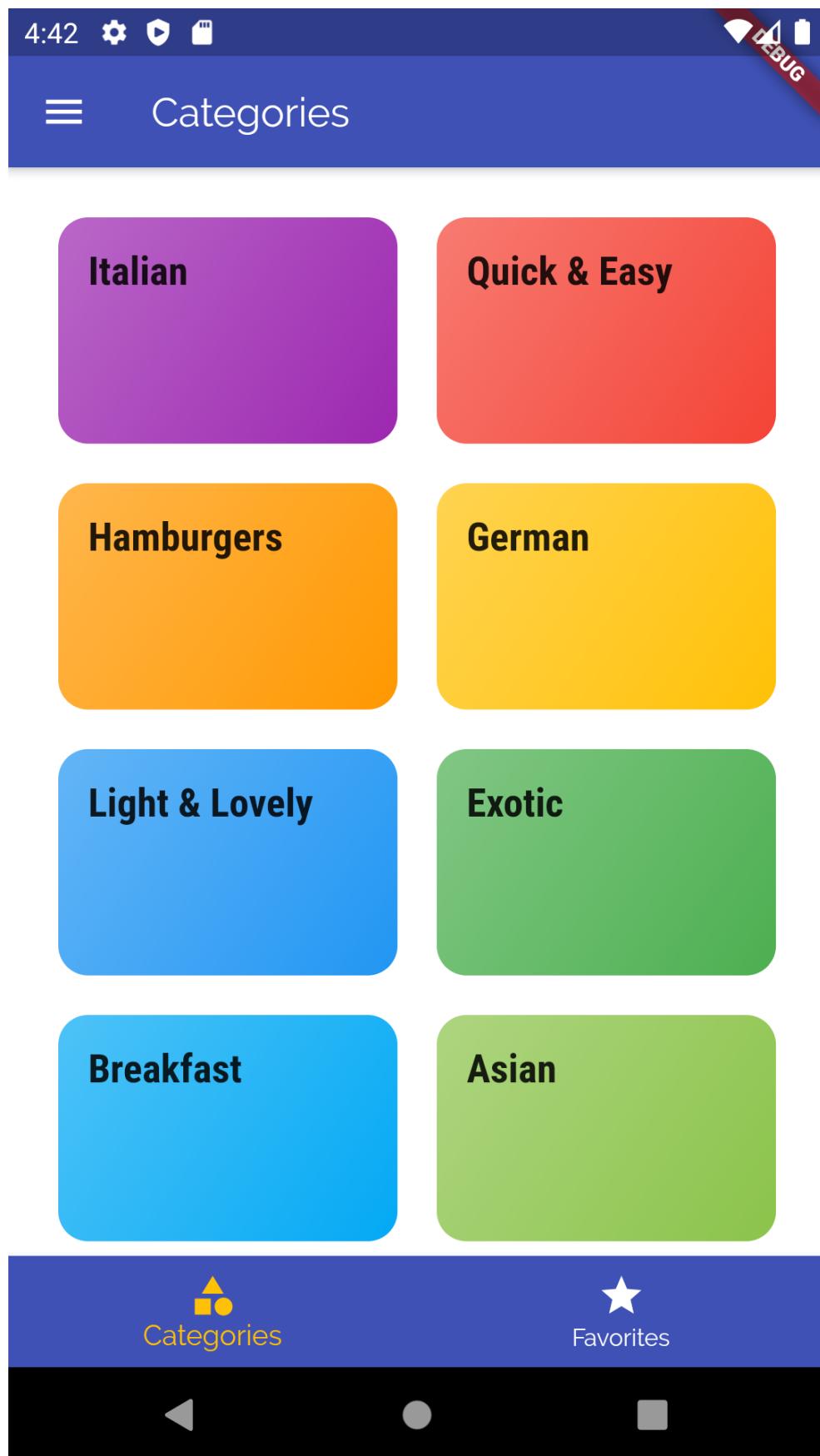


אפליקציית קניות





אפליקציית מתכונים



12:54 ⚡ DEBUG

← Quick & Easy

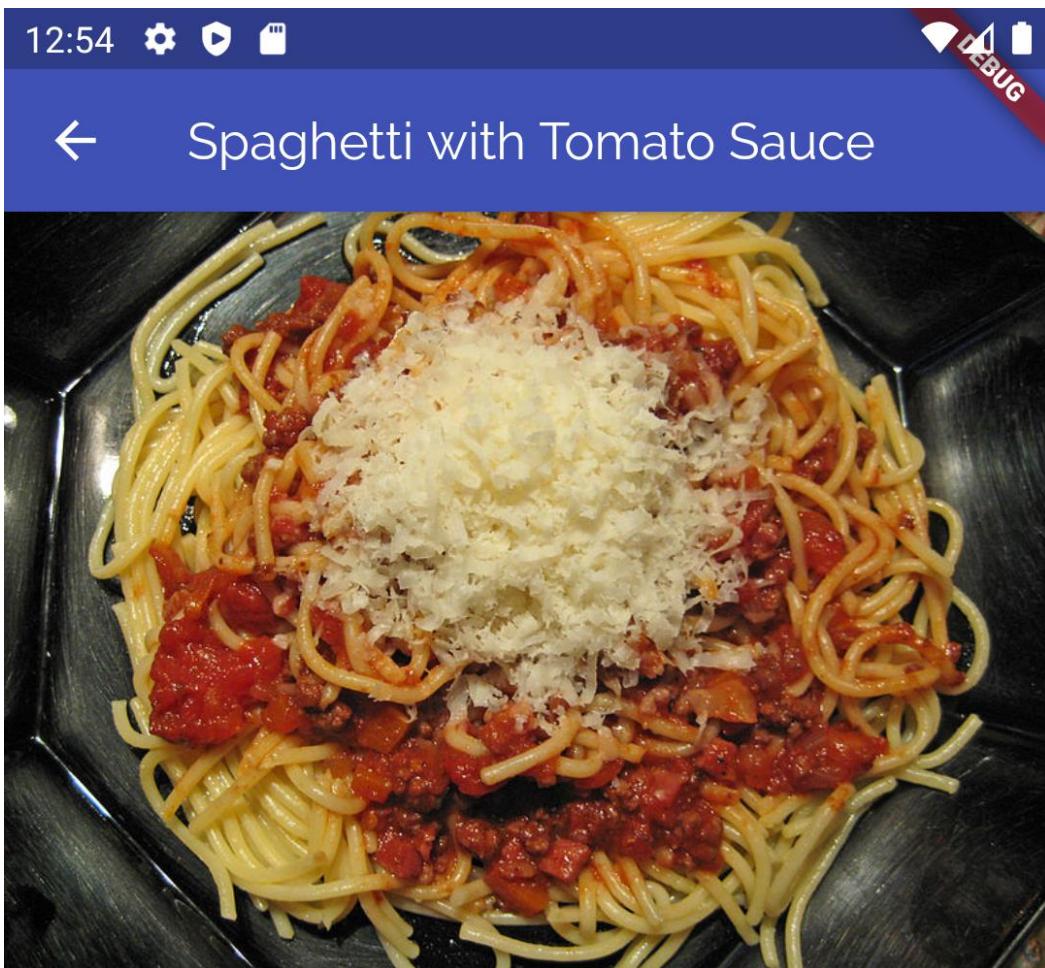


Spaghetti with Tomato Sauce

⌚ 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 מזקב על הוצאות כלכליות

