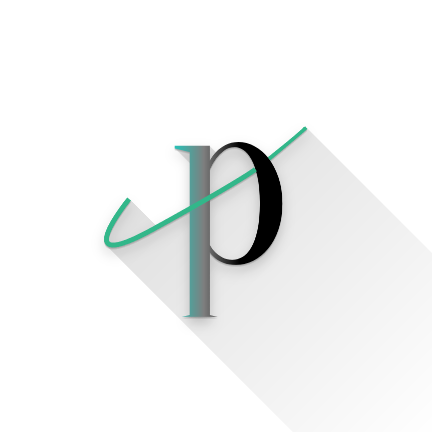


PsychomeClick

עבודת גמר תכנון ותכנות מערכות – טלפונים חכמים באנדרואיד



שם: ארתור יושר

תעודת זהות: 215716150

שם המנחה: מירי ברקנאו

תאריך הגשה: 15.5.2024

תוכן עניינים

תוכן

לא נמצאו ערכי תוכן עניינים.

# מבוא

**הרקע לפרויקט:**

כדי להתחיל את הלימודים האקדמיים, אנשים יצטרכו לגשת פעם או פעמיים למבחן ניפוי בשם "הפסיכומטרי".

הפרויקט, פסיכו-מקליק, הינו אפליקציית לומדה לבחינה הפסיכומטרית. בחינה זו מורכבת משאלות אמריקאיות בעלות ארבע תשובות. שמונה פרקים בעלי 20 שאלות כאלו, ופרק כתיבת חיבור.

האפליקציה מאפשרת לתלמידים ללמוד נושאים מסוימים לבחינה הפסיכומטרית ממבחר של נושאים ושאלות. ההתקדמות של כל משתמש נשמרת ולכן אין צורך לפתור שאלות שכבר נפתרו על ידי המשתמש. ההתקדמות של המשתמש מוצגת למעשה בעזרת אחוזים. כל שאלה בפסיכומטרי מסווגת בעזרת נושאים ותתי נושאים. לדוגמא, בעיות אנלוגיות, שאלות הבנה והסקה, ושאלות קטעי קריאה , הינם נושאים שנכללים תחת הפרק המילולי. במידה והמשתמש עונה נכון על 100 אחוז מהאנלוגיות ואינו עונה על הבנה והסקה וקטעי קריאה, יהיה אחוז מסוים של התקדמות בפרק המילולי אותו יוכל לראות (כ33 אחוז). משתמשים יכולים לצפות בנוסחים של הבחינה הפסיכומטרית, ליצור כרטיסיות ללומדת מילים (שכן פסיכומטרי דורש אוצר מילים עשיר) ולצפות בכרטיסיות של משתמשים אחרים, הדמיה של שעון עצר בשביל תרגול בחינה (בבחינה משתמשים בסטופר) ובנוסף אופציה לתקשר בצ'אט עם בוט למענה על תשובות. למנהלים, יש את הגישה להוסיף, לשנות ולמחוק שאלות שהועלו למאגר השאלות הפסיכומטרי.

קהל היעד של הפרויקט הוא כל מי שרוצה למצוא דרך מהירה, נוחה ויעילה ללמוד למבחן הפסיכומטרי-מכל הגילאים.

הסיבה שבחרתי את הנושא – לומדה לבחינה הפסיכומטרית, היא שאני למדתי לבחינה הפסיכומטרית שבספטמבר במהלך החופש שבין יא ליב. לכן התחברתי מאוד לנושא. לאחר ניסיונות רבים ללמוד לבחינה הפסיכומטרי של מועד ספטמבר, הבנתי שאין לי הרבה אופציות חינמיות ללמוד למבחן הזה. נושא זה היה נושא שממש התחברתי עליו והוא ממש מתאים לדרישות הפרויקט מבחינה של מה שאפליקציה מהסוג הזה (אפליקציית לומדה) דורשת. לאחר התייעצות עם חברים, משפחה ועם המורה בבית ספר הבנתי שזה נושא יחסית קל ומעניין ולכן בחרתי בו כנושא לפרויקט.

**תהליך המחקר**

מחקר על תחום ידע-

עשיתי מחקר בקשר לדרכים הטובות ביותר ללמוד לפסיכומטרי ועם איזה אמצעים אנשים הכי מעדיפים. הסתכלתי באופן כללי מה הדברים שדורשת אפליקציה זאת לפסיכומטרי.

בדיקת אפליקציות קיימות בשוק:

בדקתי את האפליקציות שקיימות בשוק (בapp store או בplay store) , בין אם זה האפליקציות בתשלום ואפליקציות חינמיות. הם כוללות את תוכנות הלימוד של הקורסים "יואל גבע", "קידום", ו"אקס קמפוס פסיכומטרי".

סקירת המצב הקיים בשוק-

לאחר שבדקתי את כל האפליקציות בשוק הבנתי כי האפליקציות האלה לרוב דורשות מחיר יקר כדי לקבל חלק ואפילו את כל האופציות בהן. זאת ועוד, האפליקציות לרוב מיועדות ללמידה ביחד עם קורס מסוים ולכן אין סיבה להשתמש בהם בלי תשלום לקורס. התכנים החינמיים אינם מספקים, ולרובם יש רק שימוש אחד (לדוגמא, ללמוד רק מילים). אין אפליקציה אחת שיש בה גם לומדה של מילים וגם שאלות. לכן הבנתי שצריך אפליקציה שתכלול את שתי אלו ותהיה פשוטה ויעילה ללמידה בניגוד לאפליקציות האחרות.

**טכנולוגיות שהלומד משתמש בה שאינם חלק מתוכנית הלימודים:**

בינה מלאכותית – באפליקציה יש אפשרות לדבר עם צאט בוט שמטרתו לענות על שאלות הקשורות .

Gemini apiלפסיכומטרי. דבר זה משתמש ב

גירוד אינטרנט - כדי להוציא מהאתר של הבחינה הארצית את הקישורים למבחני הפסיכומטרי .Jsoupהאפליקציה משתמש בספריית שנקראת

ניהול גייסון -כדי להפוך מחרוזות גייסון לאובייקטים ומערכים שניתן לנהל אותם ולהוציא מהן מידע, .gsonמשתמשת האפליקציה בספרייה שנקראת

**אתגרים מרכזיים:**

בעיה אחת שעלתה לי בפרויקט היא בפעולות על הפיירבייס. פעולות אלו, כמו קריאת נתונים ממסמך באוספים בפיירסטור, הם פעולות אסינכרוניות. כלומר התוכנה ממשיכה לפעול גם לפני שהפעולות האלו הסתיימו. דבר זה היה לא רק חדש בשבילי, אלא גם גרם להגבלה למה שאני יכול לעשות בפרויקט. לדוגמא, כדי לגשת לתמונות ב"אחסון" בפיירבייס, אני יכול לגשת רק למסמך אחד בכל פעם ולכן אני צריך שהתוכנה תחכה לכל התמונות. רק שהיא לא חכתה – התמונות גם לא "מורדות" לפי הסדר שאני רוצה, וגם התכונית ממשיכה לרוץ ולבצע דברים שקשורים לתמונות – שרק מצקתן הורדו. לא יכולתי למנות את כמות התמונות שכבר הורדו בעזרת משתנה רגיל ונאלצתי להשתמש במערך של מספרים שלמים בעל גודל אחד, או ב"משתנים אטומיים", כדי לבדוק מתי להמשיך עם התוכנית (לספור כמות תמונות שהורדו). רק בשלב מתקדם של הפרויקט הבנתי שניתן להשתמש באובייקט של רשימת משימות, שמכילה את כל הפעולות האסינכרוניות שאני רוצה לבצע-ובסיום של כל הפעולות האלה אני יכול להגיד מה לעשות בתכנית.

בעיה נוספת שעלתה לי היא למעשה לחשוב על כיצד לממש את השאלות במאגר הנתונים ככה שאני יוכל ליצור סוג של "עץ" נושאים, ככה שכל נושא מתפצל לכמה נושאים, כשבתחתית העץ נמצאים השאלות בהתאם.

הייתי צריך לחשוב על זה יחסית הרבה זמן ואיך אני יעשה את זה ביעילות הטובה ביותר. שרטטתי על נייר את קובץ ה"גייסון" שמכיל את הנושאים ואת השאלות לפסיכומטרי וככה הגעתי למסקנה שאני צריך ליצור אותו כפי שייצרתי. המחרוזת הזו-שנשמרת בפיירסטור, משמשת למיקום של השאלות במאגר הנתונים במקומות המתאימים להם. (תיאור נרחב ל"עץ" של השאלות אסביר בהמשך תיק הפרויקט).

את עץ השאלות הזה רציתי להציג אותו ממש כעץ, וזה בשביל שהאדמינים של האפליקציה, יוכלו לבחור היכן למקם את שאלות הפסיכומטרי על העץ. כלומר באיזה נושא. אני ניסיתי להשתמש בספריות של עצים באנדרואיד (ולא מצאתי הרבה כאלו כי רוב הספריות של "עצים" באנדרואיד מתכוונות להציג רשימת קבצים – כמו כשאתה פותח את "סייר הקבצים" במחשב וכדומה). אך אף אחד מהספריות האלו לא באמת עבד או לא היה ניתן באמת לשנות אותם דינמית. לכן בחרתי שאני כן יעשה עץ, אך האדמינים יוכלו לשים את השאלות רק ב"קצוות"\ בענפים העליונים של העץ\ברמה הגובה ביותר של העץ...

בנוסף היה לי אתגר של כיצד אני למעשה ישמור את ההתקדמות של המשתמש בשאלות. בפשטות, יש רשימה של שאלות מנושא מסוים (שלוקחים אותו מעץ השאלות) ורשימה של השאלות שהמשתמש ענה. רשימה זו הייתה אמורה להיות במקור דינמית, כלומר, המשתמש יכול היה ללכת ל"שאלה הבאה" כל פעם, באיזשהו מצב רשימת השאלות המשתמש ענה עליהן תגמר. התוכנית תקח שאלה אחת אוטומטית מהרשימה של השאלות (הרשימה מהעץ) ותוסיף אותה לרשימה של השאלות שהמשתמש פתר.

הבעיה היא שכאשר אני משנה את המיקום של השאלות או מוחק אותם, הם נמחקים מהרשימה מהעץ, אך לא מהרשימה של המשתמש. וכאשר אני מוחק שאלות, משנה מיקום שלהן, ואפילו מכניס שאלות לרשימה כאשר בעבר הם היו במיקום אחר באותה הרשימה- אז התחילו לצוץ הרבה בעיות ובאגים בתכנית שאת חלקם אני אפילו לא הצלחתי לפצח (ואפילו באגים שניתן לקבל רק במקרים מאוד מסוימים). פתרתי את זה על ידי כך שברגע שמשתמש לוחץ על נושא מסוים כל הרשימה של השאלות (מהעץ) נשמר ברשימה של השאלות שהוא פתר (מתואר בהמשך התיק).

אך האתגר הקשה ביותר בפרויקט הוא התיק פרויקט. זה מכיוון שנתנו לנו דוגמאות לתיקי פרויקט שבכלל לא היו לפי התבנית של התיק פרויקט. לכן הייתי צריך לכתוב את התיק פרויקט מחדש ולפי התבנית ואת זה הבנתי רק כשסיימתי את התיק פרויקט בתבנית הלא נכונה, כפחות מארבעה ימים לפני ההגשה של התיק.

הפרויקט עונה על הצורך להוות אפליקציה לומדה לפסיכומטרית שהיא חינמית. הוא מכיל בדיוק את הכלים הנדרשים לתרגול לבחינה הזאת ואינו עמוס באפשרויות מיותרות שמסיחות את הדעת של משתמשי האפליקציה.

הפרויקט נותן את הפתרון לבעיות שצצו בחקר המקדים, הוא מאפשר למשתמש גם ללמוד מילים וגם לתרגל לבחינה הפסיכומטרית (שני האופציות האלו לא קיימות באפליקציות חינמיות לפסיכומטרי). הוא מכיל גם הדמיית סטופר שהיא חיונית לבחינה ואפילו דיבור עם צאט בוט. לכן הוא כולל בתוכו מספיק תכנים שאינם קיימים באפליקציות אחרות והוא יעיל ופשוט לשימוש.

חידושים, התאמות ועדכונים של אלמנטים טכנולוגיים: שימוש ב"תצוגת מחזור" במקום "תצוגת רשימה".

# תיאור תחום ידע

**אבייקטים נחוצים**

UserData –אובייקט שמייצג את נתוני המשתמש.

CardSet- אובייקט שמכיל את סט הכרטיסיות.

Card - אובייקט שמייצג כרטיסיה בודדת.

Message – אובייקט שמייצג הודעה בודדת בצאט.

Node – אובייקט שמייצג חוליה בעץ.

|  |  |
| --- | --- |
| **האובייקט** | **סוג הנתונים** |
| **UserData**  **מכיל את נתוני המשתמש** | **name - String**  **email – String**  **phone-String**  **userProgress-String** |
| **CardSet**  **מייצג סט בודד של כרטיסיות** | **Id- String**  **Title- String**  **Date – String**  **isPublic – Boolean**  **Description- string**  **Cards- string**  **userId - string** |
| **Card**  **מייצג כרטיסייה בודדת** | **Term – String**  **Meaning - String** |
| **Message**  **הודעה בצאט.** | **Text – String**  **isLeft-String**  **graduallyWrite – Boolean**  **icon - int** |
| **Node**  **חולייה בעץ "הנושאים" (כמו נושא "אנלוגיות"). יכול להכיל או רשימה של נושאים נוספים, או רשימה של שאלות.** | **Name – String**  **Nodes – Nodes[]**  **questionList – String[]**  **percent - int** |

**ייצוג המידע**

כדי שיהיה לי גישה לנתונים מסוימים בנוחות (כמו האובייקט של המשתמש), אני שומר אותם כמשתנים פומביים במחלקה "FirebaseManager".

-ברגע שהמשתמש נכנס למערכת, המידע שלו נשמר באובייקט "נתוני המשתמש". אני לא שומר את הסיסמה כי אין צורך בשמירתה, וגם לא שומר את המזהה של המשתמש, משום שיש אובייקט מיוחד שניתן לגשת עליו ב"FirebaseManager", שנקרא "firebaseAuth" והוא מנהל את מה שקשור לכניסה של המשתמש עצמו.

מכיוון שיש באפליקציה אופציה לניהול כרטיסיות, כל המשתמשים יוכלו לראות את הסטים של הכרטיסיות שלהם ואת הסטים שהוגדרו כPublic של שאר המשתמשים (ולבצע חיפוש) בעזרת recycler views. בתוך כל סט יהיה recycler view נוסף שמייצג את הכרטיסיות. נתוני הסטים נשמרים כאובייקט של CardSet, הכרטיסיות עצמן, אשר נמצאות בכל סט, נשמרות כאובייקט של Card.

-משתמשים יוכלו גם לפתור שאלות מהמאגר. הם יראו חמש תמונות. אחת מייצגת את השאלה, וארבע האחרות את התשובות. השאלות מיוצגות על ידי רשימה של JsonArray, המכילה את המזהה של השאלה ואת התשובה שמשתמש ענה עליה.

-מנהלים יוכלו לצפות במאגר השאלות ולשנות את ההגדרות של כל שאלה. להם יוצג גם התשובה הנכונה לשאלה וגם לאיזה נושא שייכת כל שאלה. כדי להציג למעשה לאדמין את כל מאגר השאלות יש להשתמש גם בrecycleview, ואוסף השאלות נשמר ברשימה של מזההי השאלות. הנתונים על כל שאלה, כמו התמונות של השאלה והתשובה הנכונה, ישלפו מהאחסון וממסד הנתונים בפיירבייס בהתאם למזהה של השאלה.

כל משתמש יוכל לראות את ההתקדמות שלהם ב"עץ השאלות" שמתפצל לנושאים בעזרת אחוזי התקדמות. כל נושא מיוצג על המסך בעזרת האובייקט של Node, שמכיל את אחוזי ההתקדמות באותו נושא ומתפצל לנושאים אחרים. "עץ השאלות\נושאים" שמור בפיירסטור, ולמעשה ניתן להגיד שבעזרת הNode הוא מיוצג למשתמש. בתוכנית מציגה למשתמש בהתחלה את הרמה העליונה של העץ, ואחר כך (בלחיצה על על חולייה) היא מציגה את כל החוליות של חוליה מסויימת (כלומר אם יש חולייה מסוימת שנלחצה, החוליה הזאת נחשבת שורש של עץ ברמה 0, והתוכנה תציג את כל החוליות שלה ברמה 1, שהן בעצמן ניתנות ללחיצה).

יש לציין, שיש גם מפה "questionMap", המפה מכילה את הid (המזהים) של השאלות במאגר כמפתח, ואת התשובה הנכונה לשאלות האלו, כvalue. מפה זו משומשת אך ורק בשביל מטרה אחת – לתת את התשובה הנכונה לכל שאלה.

**תיאור פעולות על המידע**

-המשתמשים יוכלו לשנות את כל הגדרות הסטים שלהם, להוסיף ולמחוק סטים. בנוסף הם יוכלו כמובן להוסיף כרטיסיות לכל סט ולהסיר אותם, הוספת ועריכת הסטים מתבצעת באופן ישיר ולכן הסיבוכיות היא O(1). המחיקה דורשת הרכבה של הרשימת סטים מחדש ולכן היא O(n). דבר זה תקף גם לכרטיסיות שבתוך הסטים – עריכה והוספה היא O(1), מחיקה היא .O(n)

חיפוש של הסטים, עובר על כל הרשימה של הסטים ובודק ליניארית האם מתקיים תנאי מסוים. לכן הסיבוכיות שלו היא O(n).

-מנהלים יכולים להוסיף, למחוק שאלות ולשנות את ההגדרות של כל שאלה. כלומר, לשנות את כל חמשת התמונות של השאלה (תמונה אחת מייצגת את השאלה עצמה, וארבע התמונות האחרות מייצגות את התשובות) ולשנות את התשובה הנכונה. דבר זה קורה באופן ישיר ברגע שהמשתמש משנה נתון זה ישר משנה אותו בלי לולאות שעוברות על נתונים אחרים לכן, גם כאן הסיבוכיות לעריכת השאלות (חוץ משינוי הנושא) היא O(1).

כאשר האדמין משנה את התשובה הנכונה לשאלה מסוימת, או מוחק את השאלה, יש צורך לעדכן את questionMap, שהרי מכיל שאלות ואת התשובות הנכונות להן. מכיוון שזה אובייקט של מפה, עריכת ערך של מפתח היא O(1).

אך האדמין יכול גם לשנות מיקום של שאלה (בעץ ה"נושאים"\"שאלות) וזה הרבה יותר מורכב. ראשית, התוכנה צריכה בכלל לדעת מה הנושא ששאלה משתייכת עליו ולהציג אותו לאדמין. היא פונה לפעולה רקורסיבית, שמוצאת בעץ (עץ json) את המיקום של השאלה. זה מוצא את המיקום בעזרת אלגוריתם DFS - - depth first search. הסיבוכיות תהייה O(n), מכיוון שזה עדיין עובר דרך כל הנתיבים האפשריים פעם אחת בלבד. במקרה הגרוע, כמות הcalls לפעולה בתוך הרקורסיה, תהיה שווה לכמות החוליות O(n), ולכן זה הסיבוכיות שלה.

כדי לשנות את מיקום השאלה, התוכנה קודם מוחקת אותה מהנושא שהיא נוכחית בו. היא מוצא את המיקום שלה באותו האופן , רק שהפעם היא גם מוחקת את החוליה מהמיקום. לכן, הסיבוכיות עדיין O(n). דבר זה זהה להוספה של שאלה למיקום מסוים. היא מחפשת מיקום (במקום לחפש שאלה) וברגע שהיא מוצאת אותו היא מוסיפה את השאלה לרשימת השאלות (questionList). שינוי של מיקום שאלה זה זה O(n)\*2 (נדרש למחוק שאלה ולהוסיף אותה). לכן עדיין סיבוכיות לינארית.

כדי שהמשתמש יוכל לבדוק את ההתקדמות שלו ב"עץ השאלות", התוכנה צריכה רשימה שתציג (בRecycleView) את החוליות המתאימות. ברגע שנלחץ חוליה מאחת החוליות האלו, מוצגות על המסך חוליות חדשות-שהן החוליות של החוליה שנלחצה. אך יש צורך לחזור בחזרה לחוליות הקודמות, שאינם נשמרות ברשימה. לכן נשמור כל רשימה של חוליות במחסנית. כל פעם שאנחנו לוחצים על חולייה, הרשימה הנוכחית נכנסת למחסנית, ומוחלפת ברשימה החדשה. כשנחזור אחורה המחסנית תוציא את הרשימה העליונה והיא זו שתוצג למשתמש. בנוסף, כל חוליה תציג בעצמה אחוזים, שמחושבים בעזרת פעולה רקורסיבית שמשתמשת שוב פעם בשיטת הDFS. בכל פעם שאנחנו לוחצים על חוליה, זה קודם מכניס למחסנית שזה פעולה של O(1) ועובר ומציג את הרשימה החדשה O(n). לכן זה חיבור סיבוכיות קבוע עם סיבוכיות לינאריות ויצא סיבוכיות לינארית. יש לציין שאילו אנחנו רוצים להגיע מהמיקום העליון למקום התחתון בעץ או הפוך, הסיבוכיות תהיה שונה במקרה הגרוע ביותר -כל פעם שנחזור אחורה או קדימה מתוך כמות מסויימת של פעמים, התוכנה תעבור על הרשימה הקודמת או החדשה באופן לינארי. לכן, אם נחזור n פעמים אחורה וכל פעם נעבור על הרשימה n פעמים הסיבוכיות תהיה O(n^2).

כאשר המשתמשים פותרים שאלות, ההתקדמות שלהם כמובן נשמרת ומשתנה בהתאם. כאשר המשתמש לוחץ על נושא מסוים, הוא מועבר למצב של פתירת שאלות. נוצרת לו רשימה חדשה, בהתאם לרשימה של השאלות שהוא כבר פתר בעבר. הרשימה למעשה לוקחת את כל השאלות מנושא מסוים שנבחר ב"עץ השאלות", ומחפשת את התשובה שהמשתמש בחר בשאלות שהוא כבר פתר. התשובות לשאלות שהמשתמש כבר פתר ואינן מופיעות ברשימה מ"עץ השאלות", לא נשמרות (כי הם נמחקו או הועברו לנושא אחר ולכן מה שהמשתמש פתר לא רלוונטי). פעולה זו בעלת סיבוכיות זמן של O(n^2), כי היא עוברת בכל הרשימה של שאלות קיימות מנושא מסוים, ובודקת לעורך כל הרשימה של השאלות שפתר המשתמש האם השאלה נמצאת אצלו ומה התשובה שהוא ענה. שינוי של תשובה של משתמש לשאלה מסוימת, מתבצע בכך שהוא מקבל את האינדקס (של השאלה הנוכחית) וישר ניגש למיקום של השאלה ברשימה. לכן, סיבוכיות O(1).

**להלן מבני נתונים שהשתמשתי בהם בפרויקט:**

ArrayList<> - קיים לרוב בשביל לייצג רשימת מידע (רשימה של סטים, רשימה של כרטיסיות, רשימה של החוליות ורשימה של הודעות) וגם קיים בשביל לשמור את התקדמות השאלות של המשתמש בנושא מסוים (שומר רשימה של סטרינגים).

Node – מחלקה שאני יצרתי בפרויקט, למעשה מייצגת חולייה בעץ ויכולה לשמש כעץ בעצמה. בפרויקט משומשת לייצוג של "עץ הנושאים".

Stack<List<Node>> - מחסנית של רשימות המכילות חוליות. זה עוזר בהצגה של "עץ הנושאים".

HashMap<Integer,Uri> - אובייקט שמכיל מפתחות וערכים שניגשים עליהם בעזרת מפתחות אלו. אני משתמש בו בפרויקט בשני מקרים :

1) כאשר אני מוסיף שאלות, אני צריך לבחור להם 5 תמונות. בכל בחירה של תמונה אני צריך לשמור אותה. לכן אני שומר את המספר הסידורי של התמונה (0 – התמונה שמייצגת את השאלה, ו1-4 הם תמונות שמייצגות תשובות לשאלה בהתאמה), לכתובת URI , מזהה ייחודי של התמונות, שבפשטות מייצג את המיקום שלהם בטלפון של המשתמש (המשתמש בוחר תמונה וזה שומר במפה בהתאם).

2) שמירת תשובות לשאלות בquestionMap.

Bundle- אובייקט באנדרואיד הממפה בין ערכי סטרינג לערכים מסויימים. משומש בפרויקט בשביל להעביר מחרוזת ממסך אחד לאחר.

# מבנה הפרויקט

****

דף הרשמה- דף המאפשר ליצירת משתמש חדש באפליקציה בהזנת נתונים מתאימים כמו אימייל טלפון שם וסיסמה.

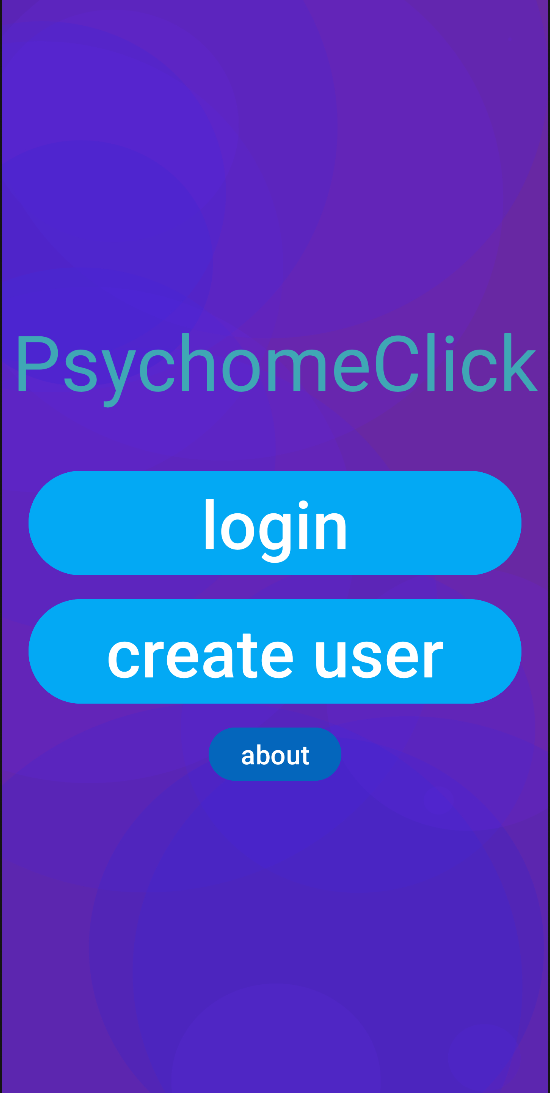
**שלב תכנון ותיעוד מסכי הפרויקט:**

מסך התחלה-

תחילה המשתמש נחשף למסך הקדמה

שמשמש לטעינה ולאחר כמה שניות עובר

למסך הראשי



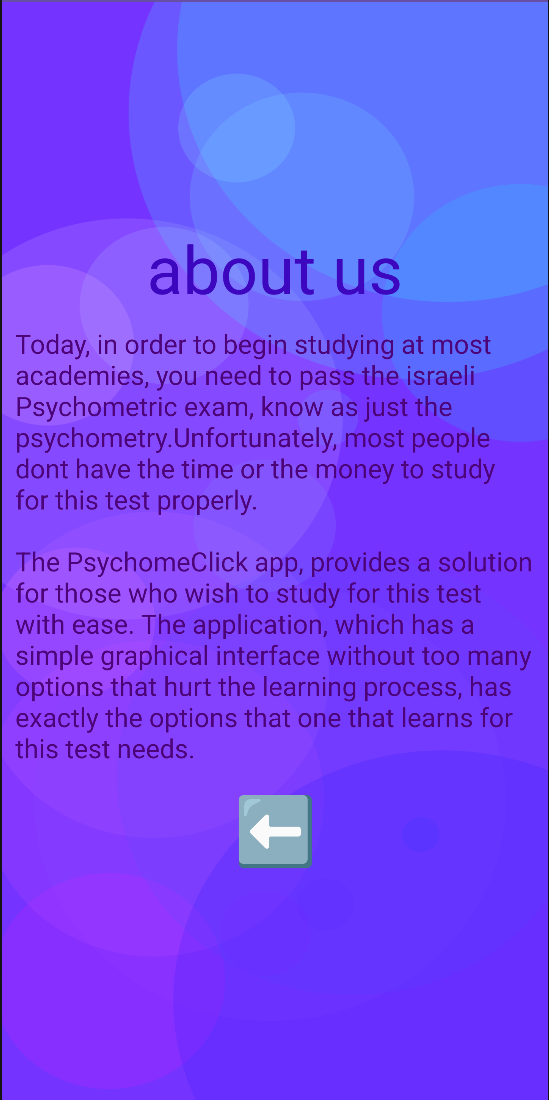
מסך ראשי- הדף שמופיע למי שאינו מחובר

לאפליקציה וניתן בעזרתו לבדוק את אודות

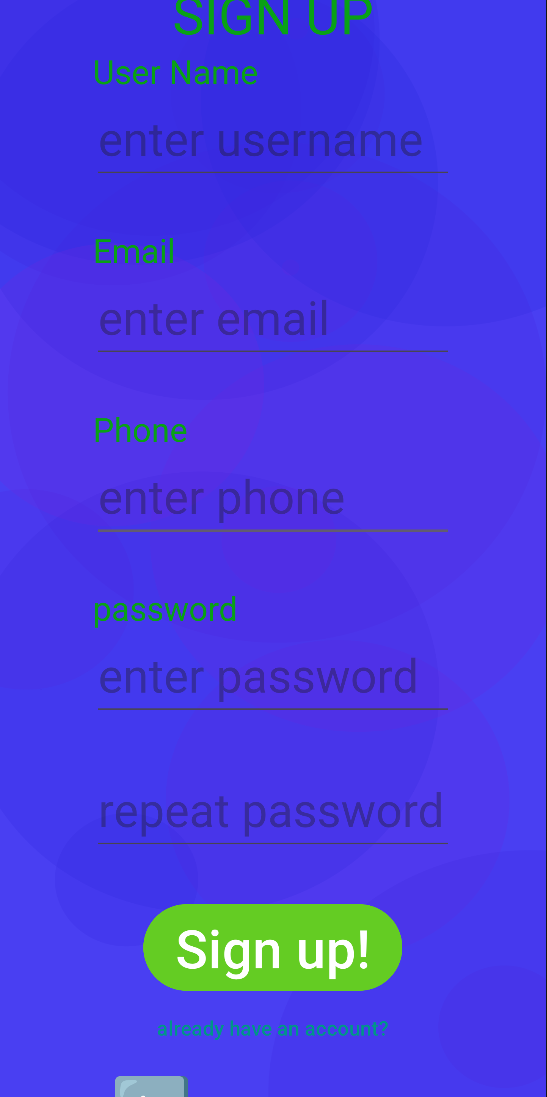
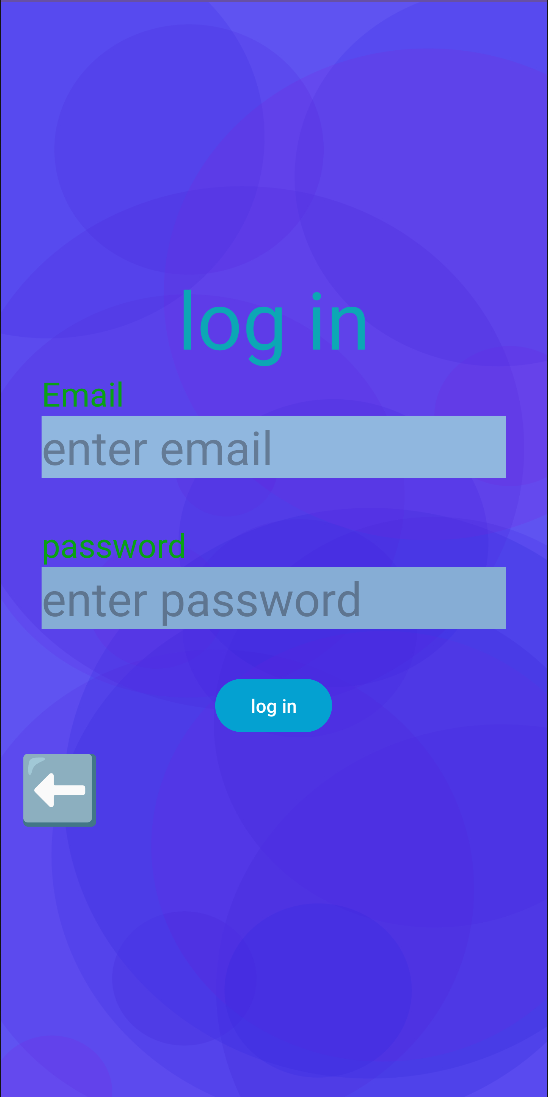
האפליקציה, להירשם ולהתחבר עם

משתמש קיים.

יש במסך תיבת טקסט שמציגה

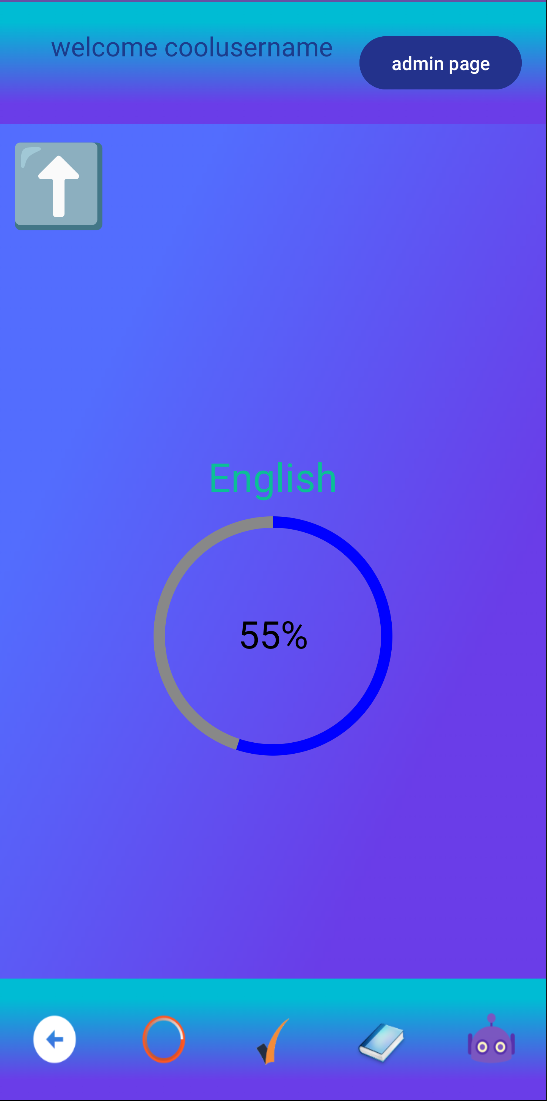
****

מסך אודות- מסך המספר על האפליקציה

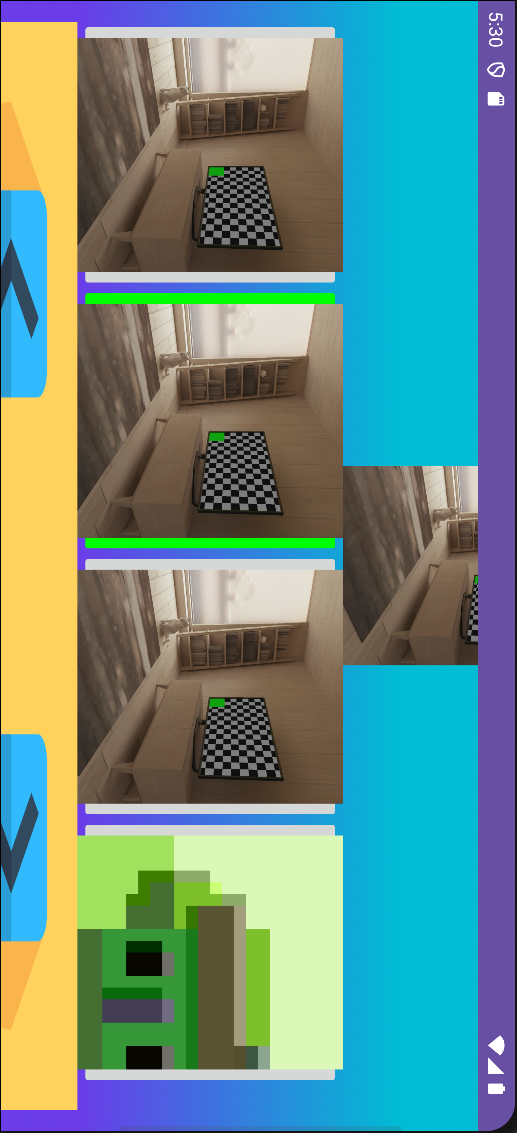
****

דף הרשמה- דף המאפשר ליצירת משתמש חדש באפליקציה בהזנת נתונים מתאימים כמו אימייל טלפון שם וסיסמה.

דף התחברות – דף המאפשר למשתמש קיים להתחבר למערכת בעזרת אימייל וסיסמה

****

דף משתמש-כללי- דף המאפשר לצפות באחוזי ההתקדמות בכל נושא ולבחור באיזה נושא לפתור שאלות

****

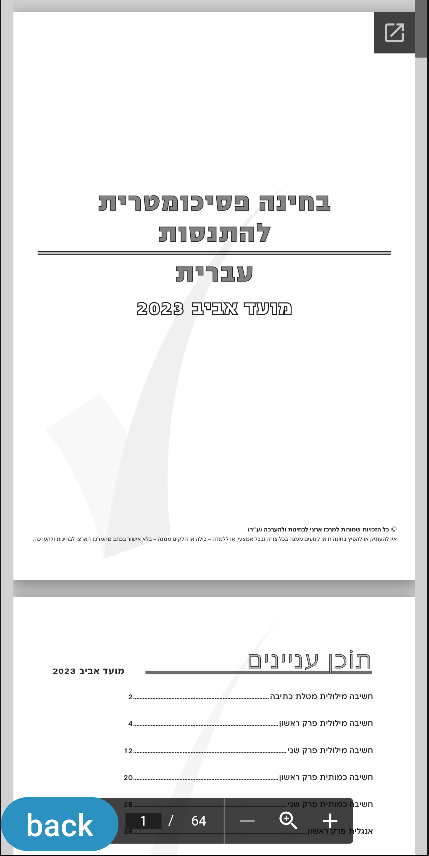
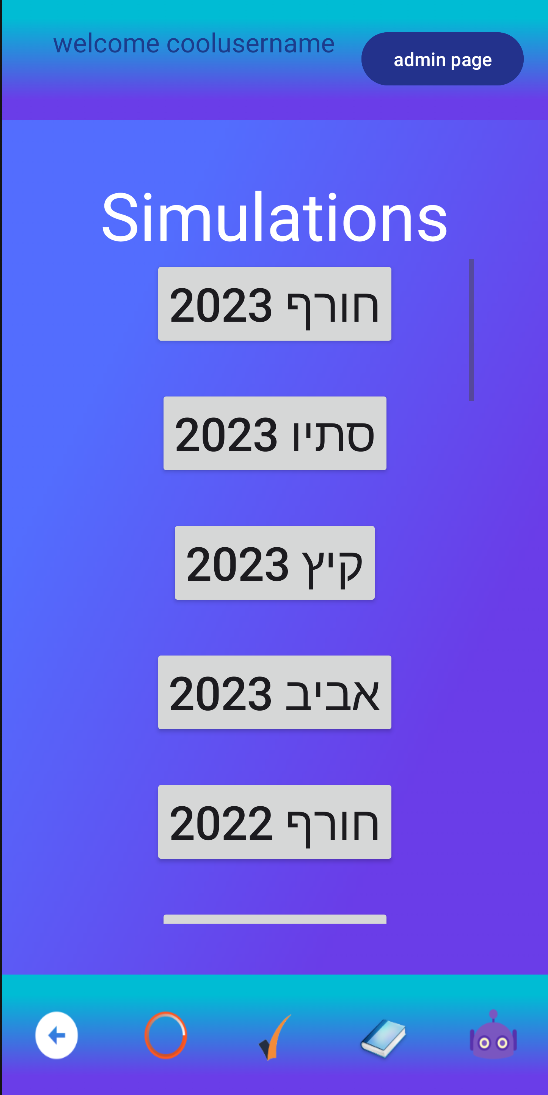
דף היבחנות – דף המאפשר למשתמש לפתור שאלות בעזרת לחיצה על אחת התשובות (אחת מן התמונות)

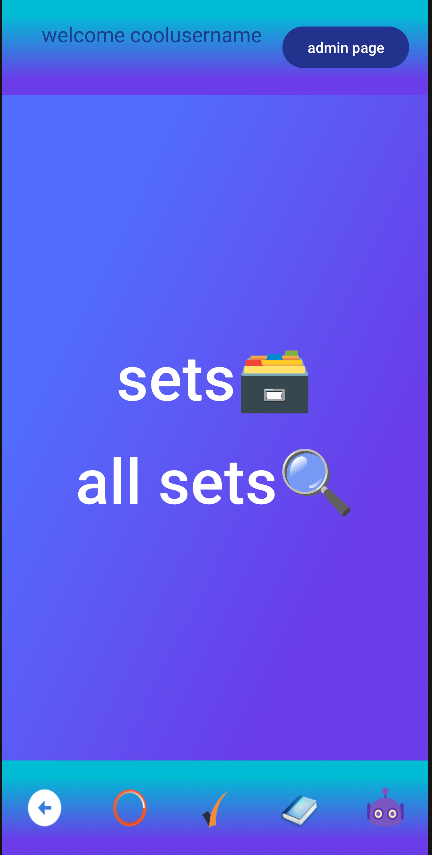
דף היבחנות – דף המאפשר למשתמש לצפות בסימולציה שנבחרה

דף משתמש – סימולציות –

דף המאפשר למשתמש לבחור

סימולציה לצפייה

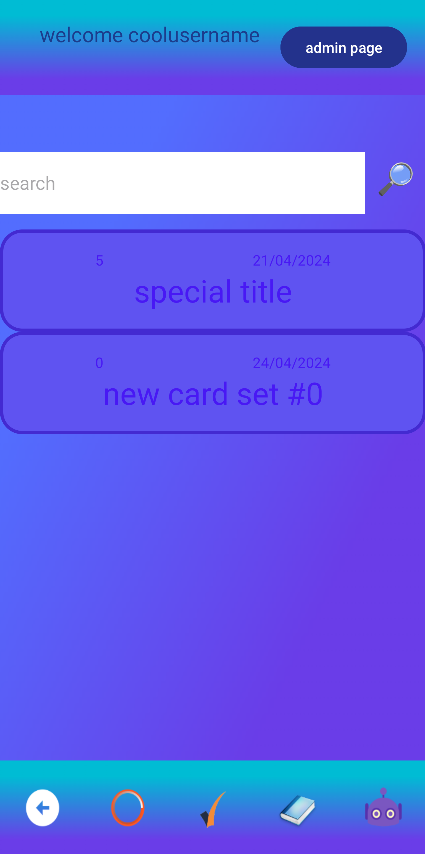
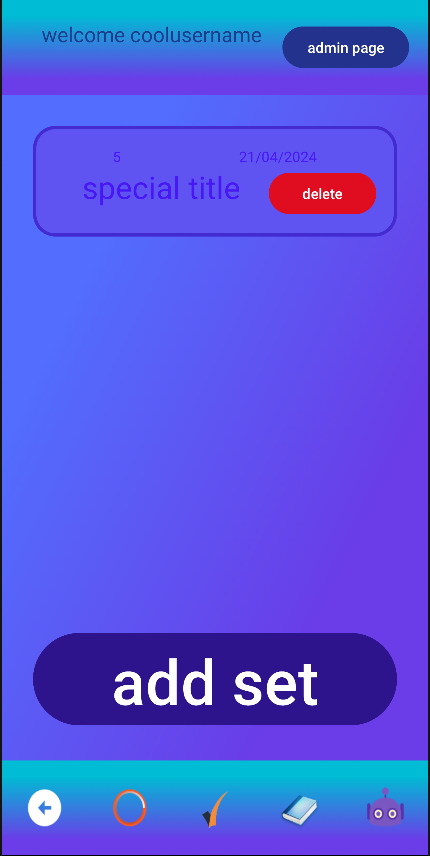
****

****

דף משתמש - דף סטים – מאפשר למשתמש לבחור האם לחפש מכל הסטים של הקלפים או לראות רק את הסטים שלו

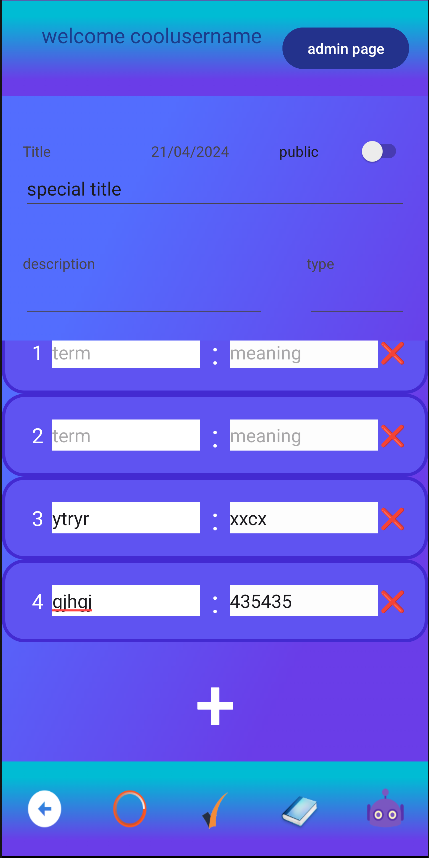
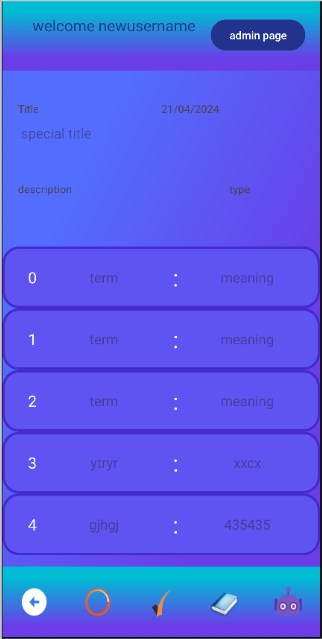
דף של סטים של המשתמש- מאפשר למשתמש לראות את הסטים שלו ולהסיר\להוסיף בהתאם

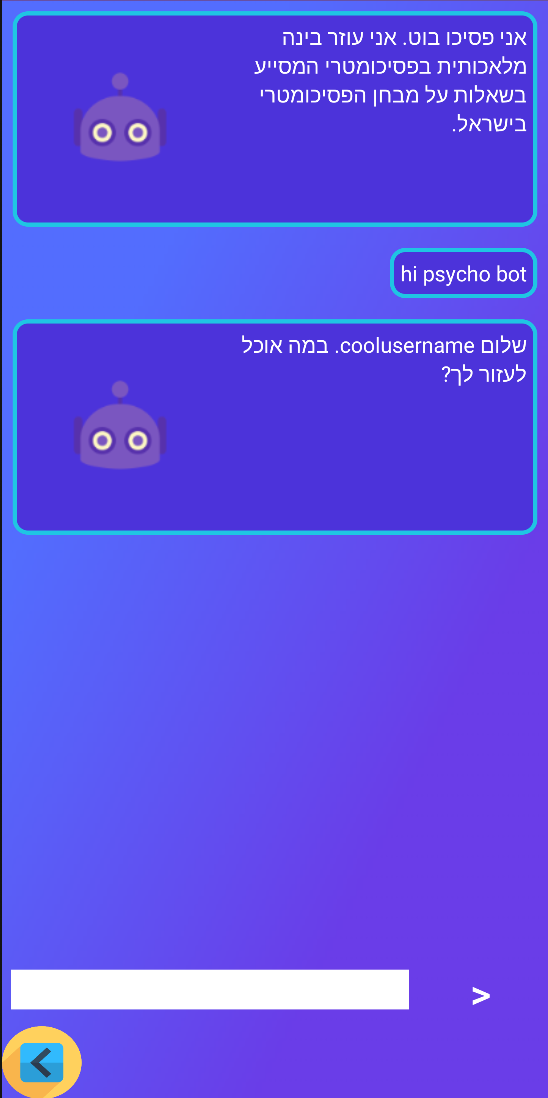
דף של כל הסטים- מאפשר למשתמש לראות את כל הסטים שמוגדרים כפומביים ולחפש את הסטים בעלי כותרת מסויימת

****

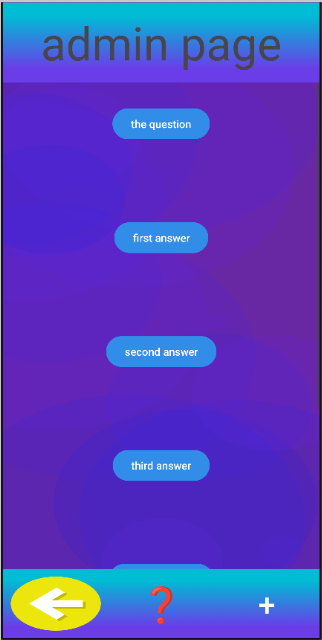
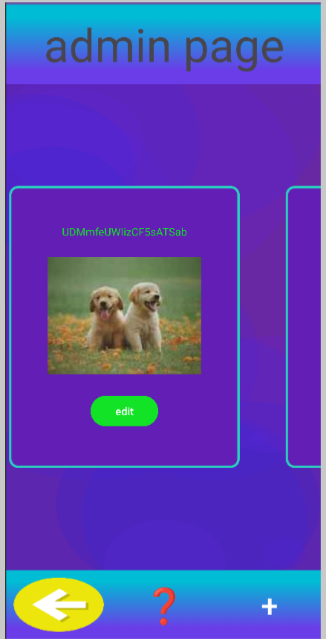
דף סט של המשתמש –בלחיצה על סט עובר המשתמש למסך עריכת סט והוא יכול לערוך אותו בהתאם.

דף סט שלא של המשתמש – בלחיצה על סט עובר המשתמש לתצוגה של סט ממשתמש אחר אותו לא ניתן לערוך

****

****

דף צאט בוט - מסך המאפשר למשתמש האפליקציה לתקשר עם צאט בוט בקשר לפסיכומטרי

****

דף אדמין – דף הוספת שאלות – דף המאפשר לאדמין להוסיף שאלה חדשה למאגר

דף אדמין – דף רשימת שאלות- דף המאפשר לאדמין לצפות בכל השאלות מהמאגר

****

דף אדמין- דף עריכת שאלות – דפ המאפשר לאדמין לערוך את השאלות במאגר

**מחלקות הפרויקט והקשרים ביניהם:**

**תרשים מסכים:**

Activity\_test

מסך מענה שאלות

Activity\_chat\_bot

מסך צאט בוט

Fragment\_simulations

מסך סימולציות מבחן

Fragment\_all\_sets

מסך סטים על פי חיפוש

Fragment\_edit\_set

מסך עריכת\הצגת סט

Activity\_log\_in

מסך כניסה

Activity\_sign\_up

מסך הרשמה

Fragment\_question\_list

מסך הצגת שאלות

Fragment\_edit\_question

מסך עריכת שאלה

Activity\_user

מסך משתמש

Activity\_main

מסך ראשי

Activity\_start

מסך פתיחה

Activity\_about\_us

מסך אודות האפליקציה

Fragment\_book

מסך סטים

Activity\_admin

מסך אדמין

Fragment\_general

מסך המציג את התקדמות המשתמש

Fragment\_Stopper

מסך הדמיית סטופר

Fragment\_add\_question

מסך הוספת שאלה

Activity\_pdftestpage

מסך מציג סימולציות

Fragment\_sets

מסך סטים של משתמש

# 

# מימוש הפרויקט

# מדריך למשתמש

# רפלקציה

# ביבליוגרפיה

# נספחים