

מסמך ניתוח

שם הפרויקט: Study Buddy

חברי הקבוצה:

ליאור טלמן, שגית מלכה, גייה סרחאן, ליאור שולמן.

מבוא

סקירה כללית

"Study Buddy" נועדה כדי לתת מענה לאתגרים המלווים סטודנטים בניהול למידה משותפת ומציאת שותפים ללמידה או למטלות. היישום מתמקד בהנגשת פתרון יעיל ומרוכז המאפשר חיבור מהיר ונוח בין סטודנטים לבין שותפים פוטנציאליים ללמידה, מורים פרטיים, ואפילו קבוצות קיימות, תוך הפחתת התלות בכלים מסורבלים כמו קבוצות וואטסאפ או חיפושים אינטרנטיים. הפרויקט מציע חוויית משתמש משופרת שמרכזת את הצרכים של הסטודנטים בפלטפורמה אחת.

מטרות הפרויקט

1. לסייע לסטודנטים לתאם קבוצות למידה בקלות ובמהירות.
2. לאפשר חיפוש ומציאה של מורים פרטיים המתאימים לצרכי הסטודנט מבחינת תחום, זמינות ואופי הלימוד (מקוון או פרונטלי).
3. לשפר את חוויית הלימוד באמצעות כלים מותאמים אישית ותמיכה בלמידה מרחוק.
4. לאפשר יצירת רשתות חברתיות ואקדמיות עבור סטודנטים ומורים פרטיים כאחד.

בעיות קיימות שהפרויקט פותר

סטודנטים רבים נתקלים בקשיים הבאים:

- קושי בארגון קבוצות למידה בצורה יעילה.
- חיפוש ממושך אחרי מורים פרטיים מתאימים, לעיתים ללא הצלחה.
- שימוש בכלים לא מותאמים, כגון קבוצות וואטסאפ, המגבילים את יכולת התיאום והארגון.

קהל היעד

1. **סטודנטים:** המשתמשים העיקריים שיחפשו שותפים ללמידה, מטלות, ומורים פרטיים.
2. **מורים פרטיים:** יציעו את שירותיהם לקבוצות למידה או ליחידים.
3. **הנהלות אקדמיות:** יוכלו לשפר הישגי סטודנטים באמצעות עידוד שיתופי פעולה ולמידה יעילה.

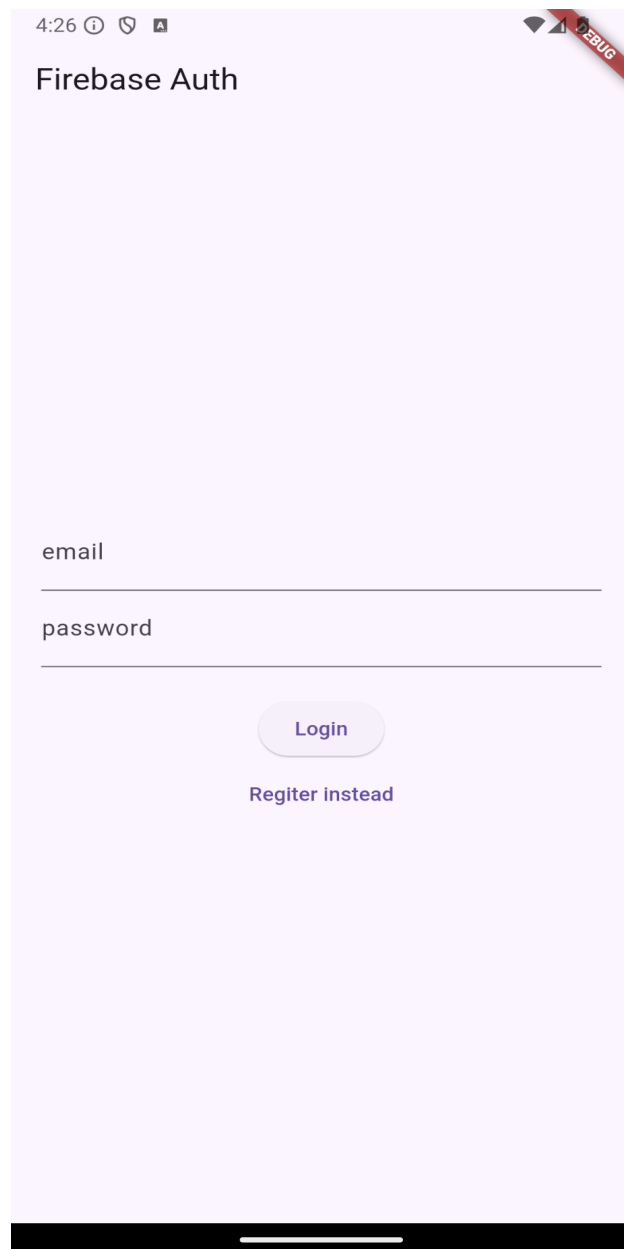
חזון המערכת

חזון המערכת הוא ליצור פלטפורמה שמחברת בין סטודנטים לבין שותפים ללמידה ולמורים פרטיים בצורה פשוטה, ידידותית ומהירה, תוך שיפור תהליך הלימוד האקדמי והוזלת העלויות הכרוכות בה. הפלטפורמה מתמקדת בשילוב פתרונות טכנולוגיים מודרניים המותאמים לצרכי המשתמשים ומציעה חוויית משתמש מותאמת אישית.

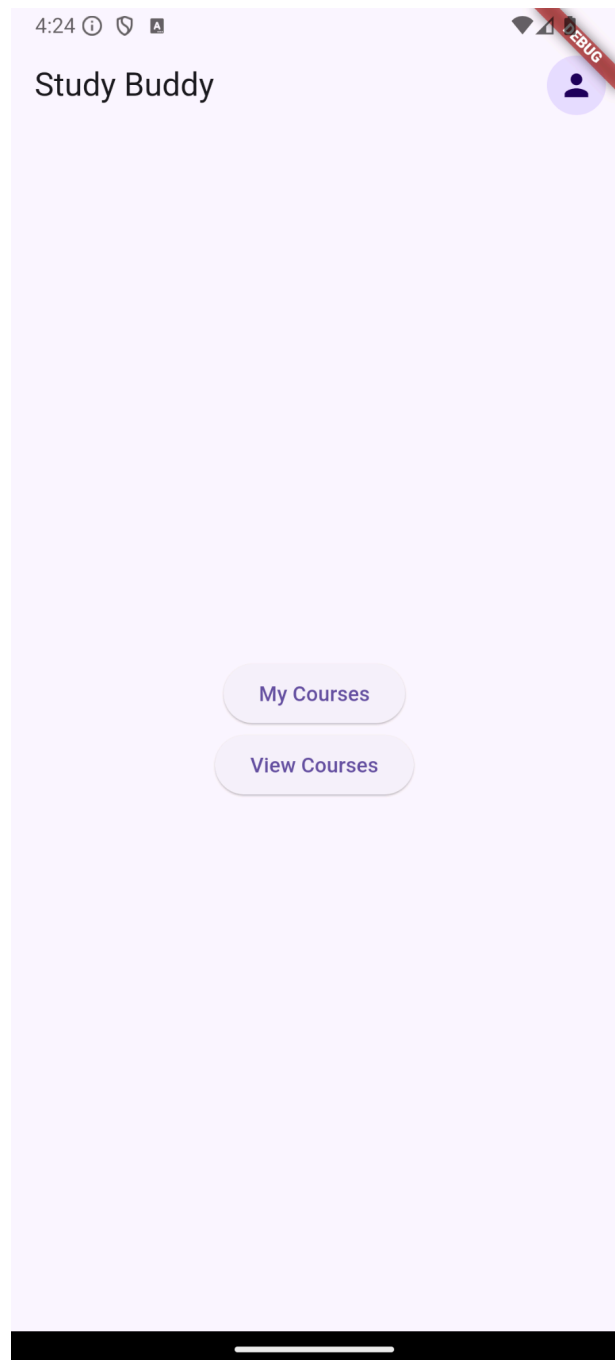
מסכי האפליקציה:

מסך כניסה:

- הכנסת שם משתמש + סיסמא על מנת להתחבר באמצעות כפתור "log in", הרשמה ע"י לחיצה על כפתור "register instead".
- במידה והמשתמש רשום - האימות מתבצע ע"י Firebase auth. אם המשתמש איננו רשום עדיין - הוא יצורף ל firebase user collection.
- כאשר המשתמש מחובר לחשבון, מסך הכניסה לא יטען בעת פתיחת האפליקציה והוא יועבר ישירות לדף הבית



דף הבית: בשלבי תכנון
מצב נוכחי:



התכנון:

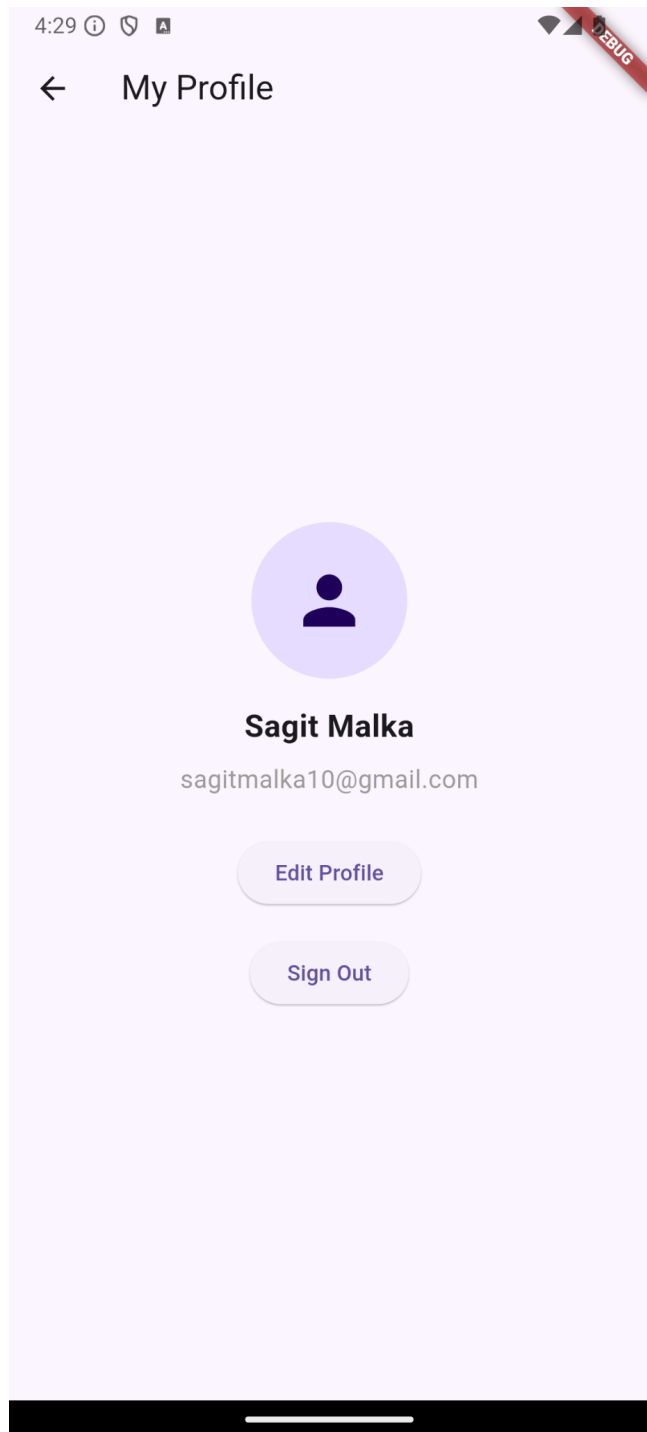
- הצגת הפעילויות הקרובות כרשימה
- פרישת כפתורים בצורה אחרת

מעברים לדפים אחרים:

- לחיצה על הסמל  מובילה לדף "my profile"

- לחיצה על כפתור "my courses" מובילה לדף הקורסים שאני רוצה להירשם אליהם (להירשם - ליצור/ להצטרף לקבוצות למידה/ קבוצות פרויקטים)
- לחיצה על כפתור "view courses" מובילה לדף כל הקורסים הקיימים במערכת, מסודרים לפי תחומי לימוד (major).

דף פרופיל משתמש:



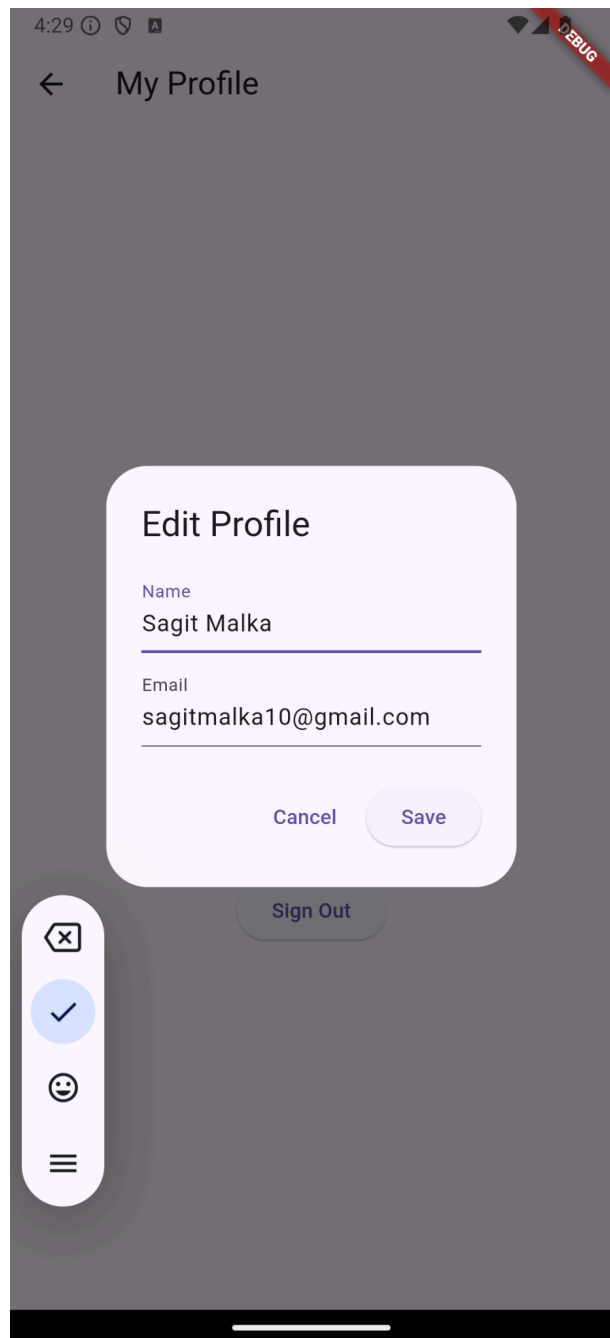
בדף זה מוצג:

- מוצגת תמונת פרופיל (פונקציונאליות בתכנון, לא קיימת כרגע)
- שם לתצוגה בתוך האפליקציה (מה משתמשים אחרים רואים)
- כתובת מייל

פונקציות/ כפתורים:

- עריכת פרטים אישיים (display name וכתובת מייל)
- התנתקות מהחשבון ומעבר לדף הכניסה
- כפתור חזרה לעמוד הבית

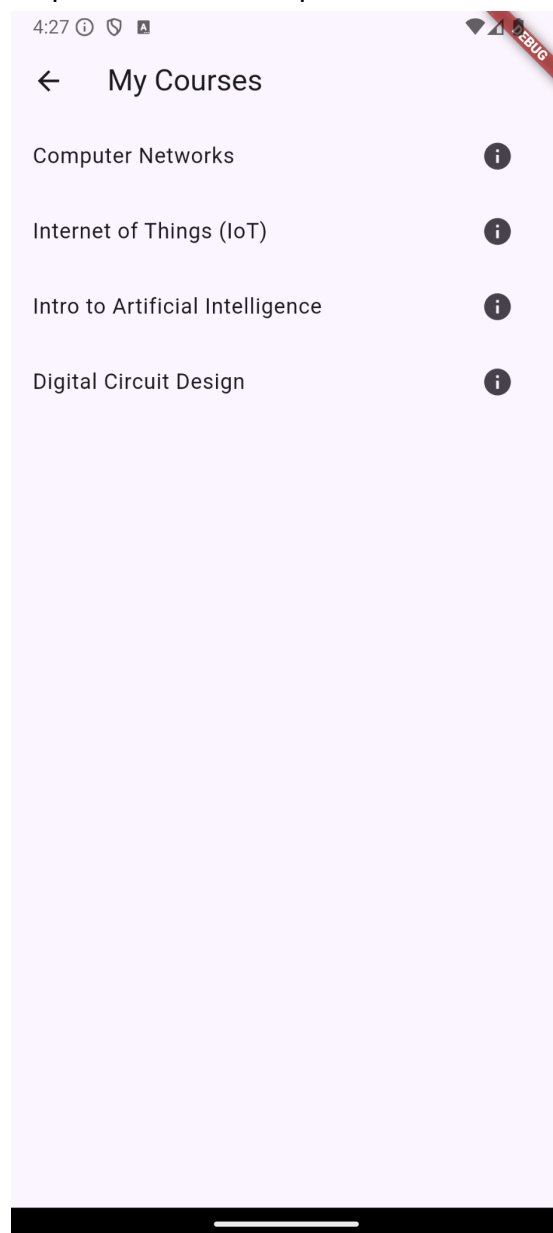
דף עריכת פרטים אישיים:



- כפתור ביטול
- כפתור שמירת השינויים שביצעתי

דף הקורסים שלי:

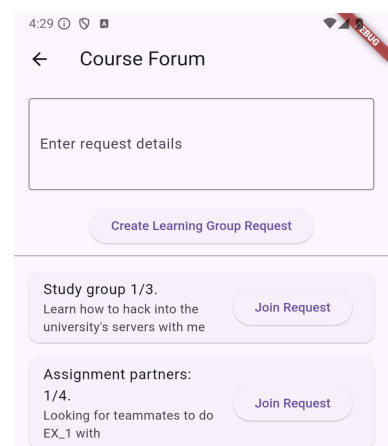
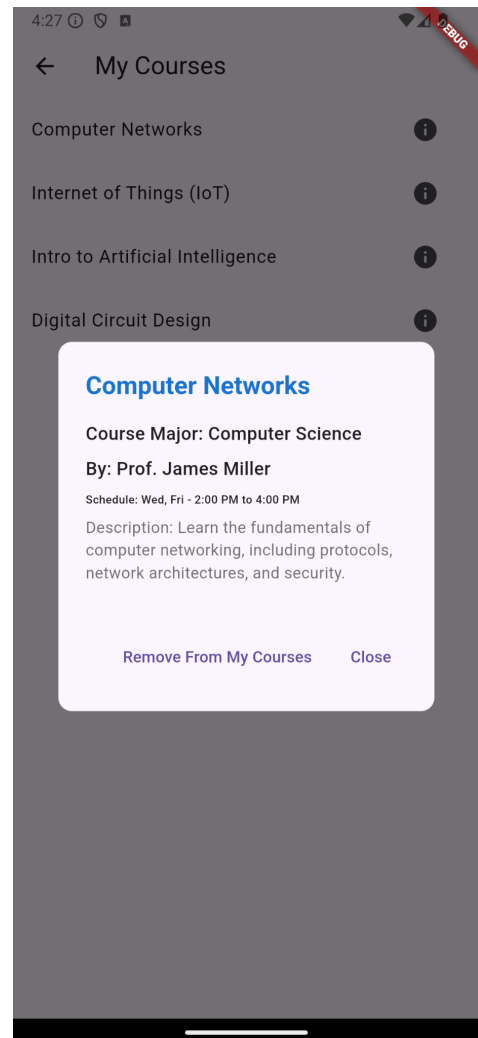
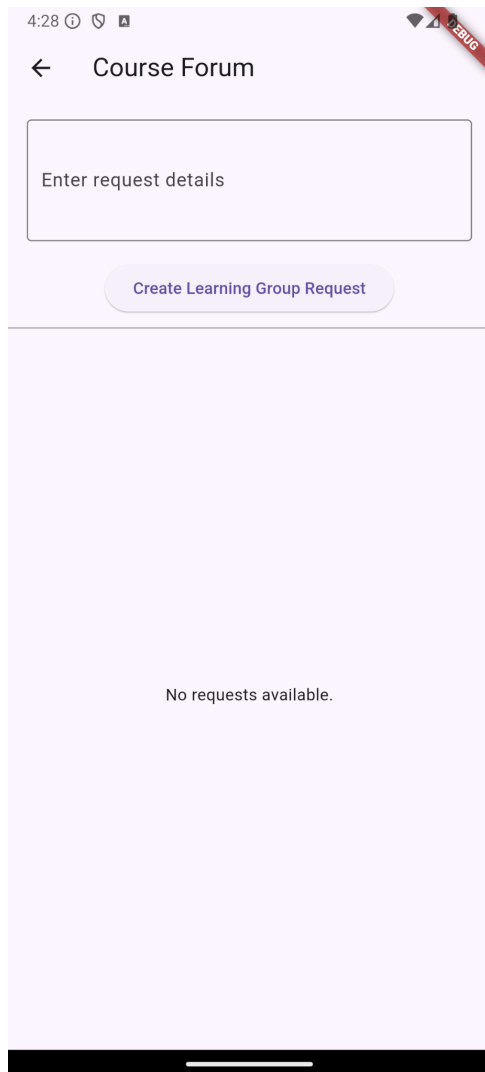
● מציג את כל הקורסים שהוספתי למעקב אחריהם



● כפתורים:

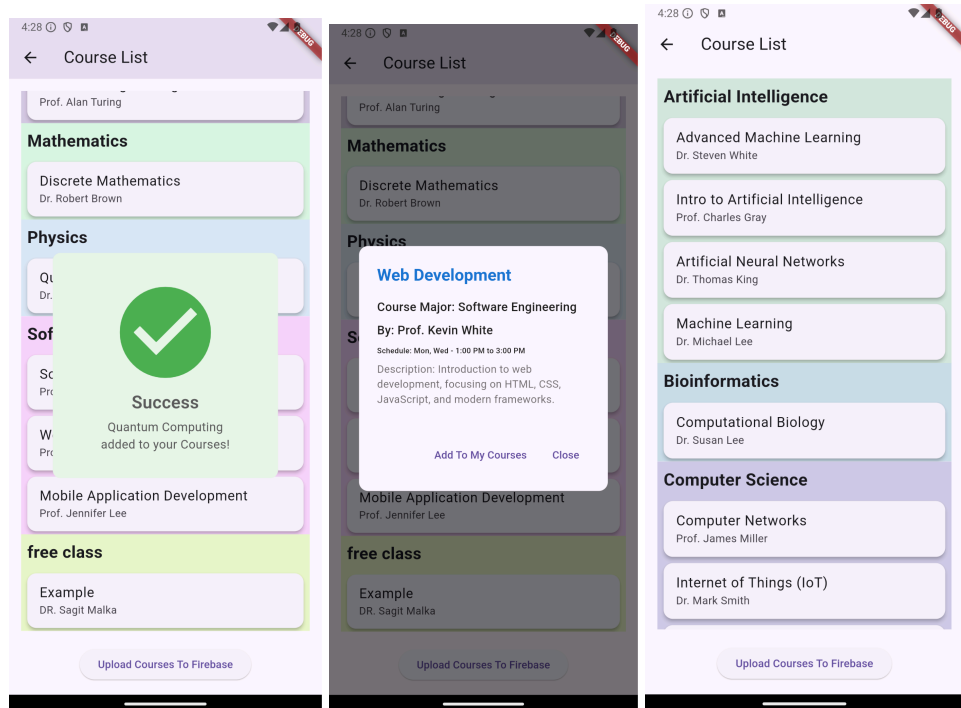
- כפתור חזרה לדף הבית
- לחיצה על סימן הקריאה בקצה השורה תציג את פרטי הקורס: מועדים, תיאור, שם המרצה
- לחיצה על שם הקורס מעביר לדף "course forum"

● ניתן לסגור את החלון וניתן למחוק את הקורס מהרשימה



- דף כל הקורסים "view courses":
- מציג את כל הקורסים הקיימים במערכת
 - חלוקה לפי נושאים
 - כפתורים:

- לחצן חזרה לדף הבית
- לחיצה על שם הקורס פותח חלון עם המידע עליו: ניתן לסגור וניתן להוסיף את הקורס ל"my courses".
- ל admin: כפתור הוספת קורסים מקובץ JSON לכלל המשתמשים



תבנית עיצוב:

תבנית העיצוב שנבחרה לפרוייקט היא **MVVM**. הפיתוח של האפליקציה מתחלק בטבעיות לשלוש החלקים שתבנית זו מגדירה: רכיבי בסיס מרכזיים באפליקציה נמצאים בתיקיות `services` ו `models` - בכוונתו לשמור את שתי התיקיות תחת תיקיית `Model` ראשית אחת. בנוסף, ב**MVVM**, הגישה של כל הUI היא תגובתית ופשוטה - מה שמשתלב בצורה טובה עם הבחירה של שפת התכנות - `Flutter`, כי היא `widget-based`. החלק העסקי מנותק לחלוטין מהUI, אחד המאפיינים הבולטים של **MVVM**. זה יאפשר לבצע שינויים בלי להשפיע על צד ג. חלק ה**ViewModel** גם כן מתאים לדרישות האפליקציה שלנו במדויק, שכן יש אינטראקציה תמידית בין פעילות המשתמש באפליקציה ובין `firebase`, מידע נמשך משם או נדחף לשם בכל תרחיש בסיסי של שימוש באפליקציה.

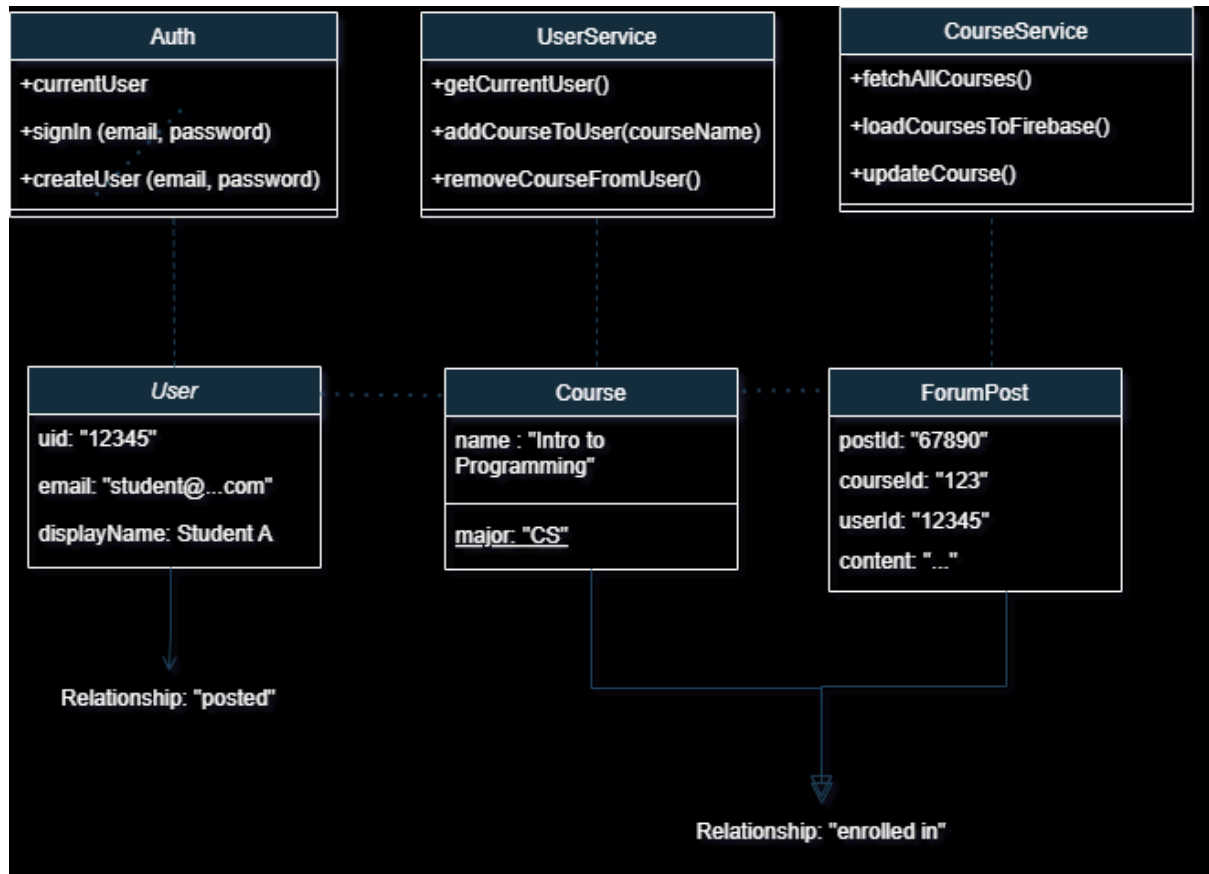
מטרת האפליקציה והבחירה ב**flutter** מתמזגים יחד - הקוד מודולרי וקריא מלכתחילה והחלוקה לפי **MVVM** מדגישה את זה אף יותר. ממשק משתמש דינמי הוא הכרחי לפרוייקט שלנו והשימוש ב**Data Binding** קורה בכל מקרה - לכן זה לא חסרון מהותי.

דיאגרמות:

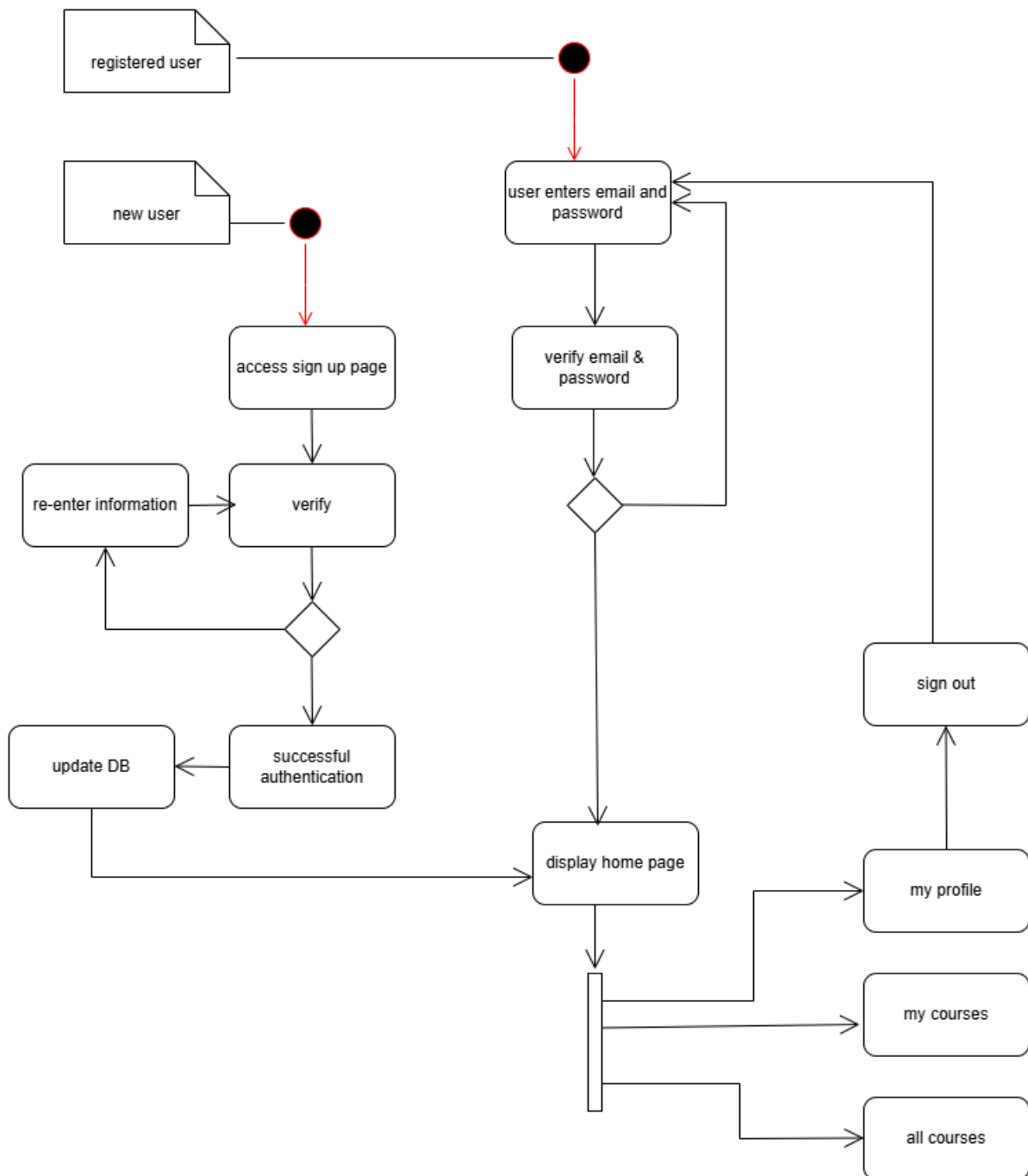
class diagram:



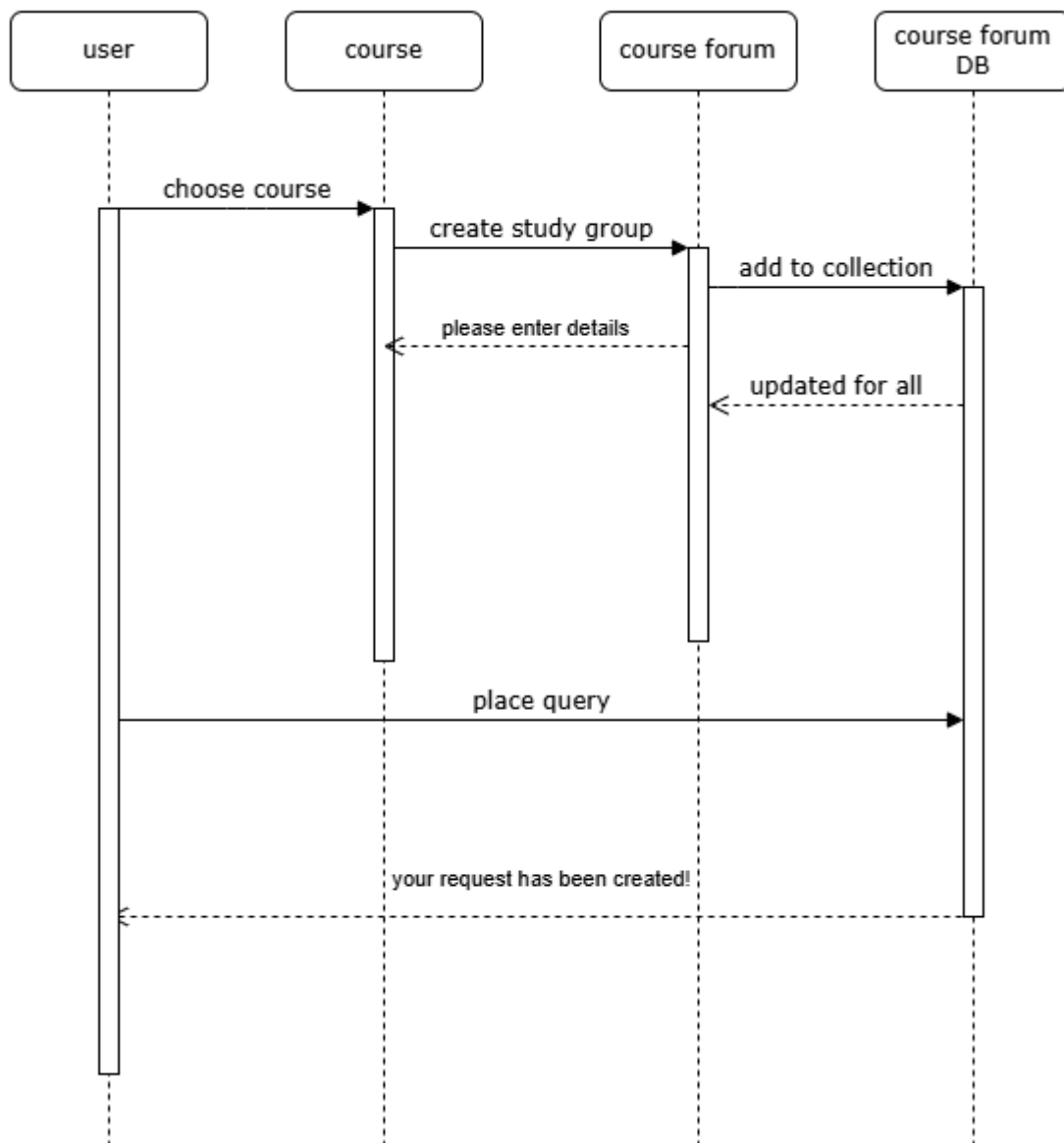
object diagram:



activity diagram:

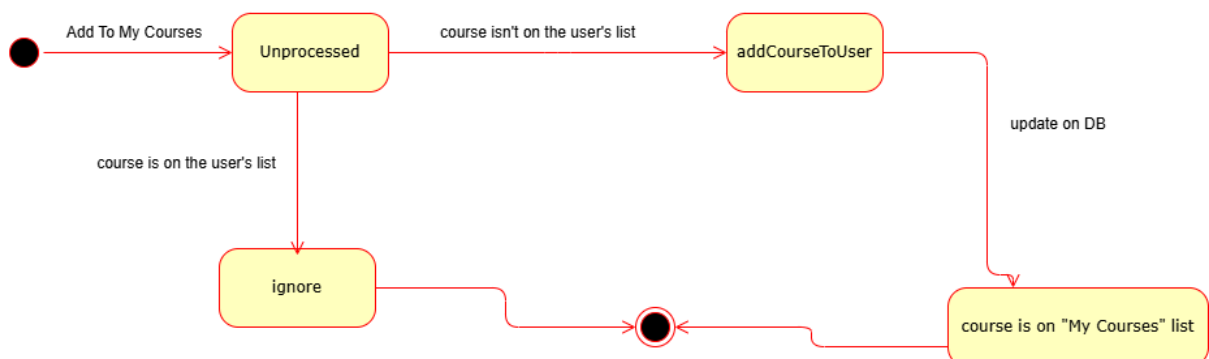


sequence diagram:



state machine diagram:

Page: View Courses



ERD diagram:

