



# Software Design Document

## Study Buddy

אסתר בינס 211328109  
טליה מיזליש 207883430  
נעה נוסבאום 206664278

### 1. מבוא

#### 1.1 מטרה

מטרת מסמך עיצוב תוכנה זה היא לתאר את הארכיטקטורה ואת עיצוב המערכת של הפרויקט Study Buddy, בין בהקשר של חווית המשתמש ובין בצורת אכסון הנתונים של התוכנה.

#### 1.2 תחום

הפרויקט שלנו מיועד לסייע לסטודנטים באוניברסיטה למצוא מורים פרטיים וקבוצות למידה מצד אחד ולספק להם הכנסה כלכלית מצד שני. האפליקציה יוצרת קשר בין סטודנטים שמחפשים ללמד סטודנטים אחרים לבין סטודנטים המחפשים מורה פרטי, וכן בין סטודנטים המחפשים להיות חלק מקבוצת למידה, כל זאת בהתאמה לצרכים של הסטודנט. האפליקציה בנויה בעזרת התוכנה Android Studio, ונתוני האפליקציה מאוחסנים ב-Firebase.

#### 1.3 סקירה כללית

מסמך זה מסודר באופן הבא: מבוא, סקירת מערכת, עיצוב נתונים, עיצוב רכיבים, עיצוב ממשק משתמש, וטבלת דרישות. בכל פרק נתאר את הרכיב הרלוונטי באפליקציה ונפרט עליו.

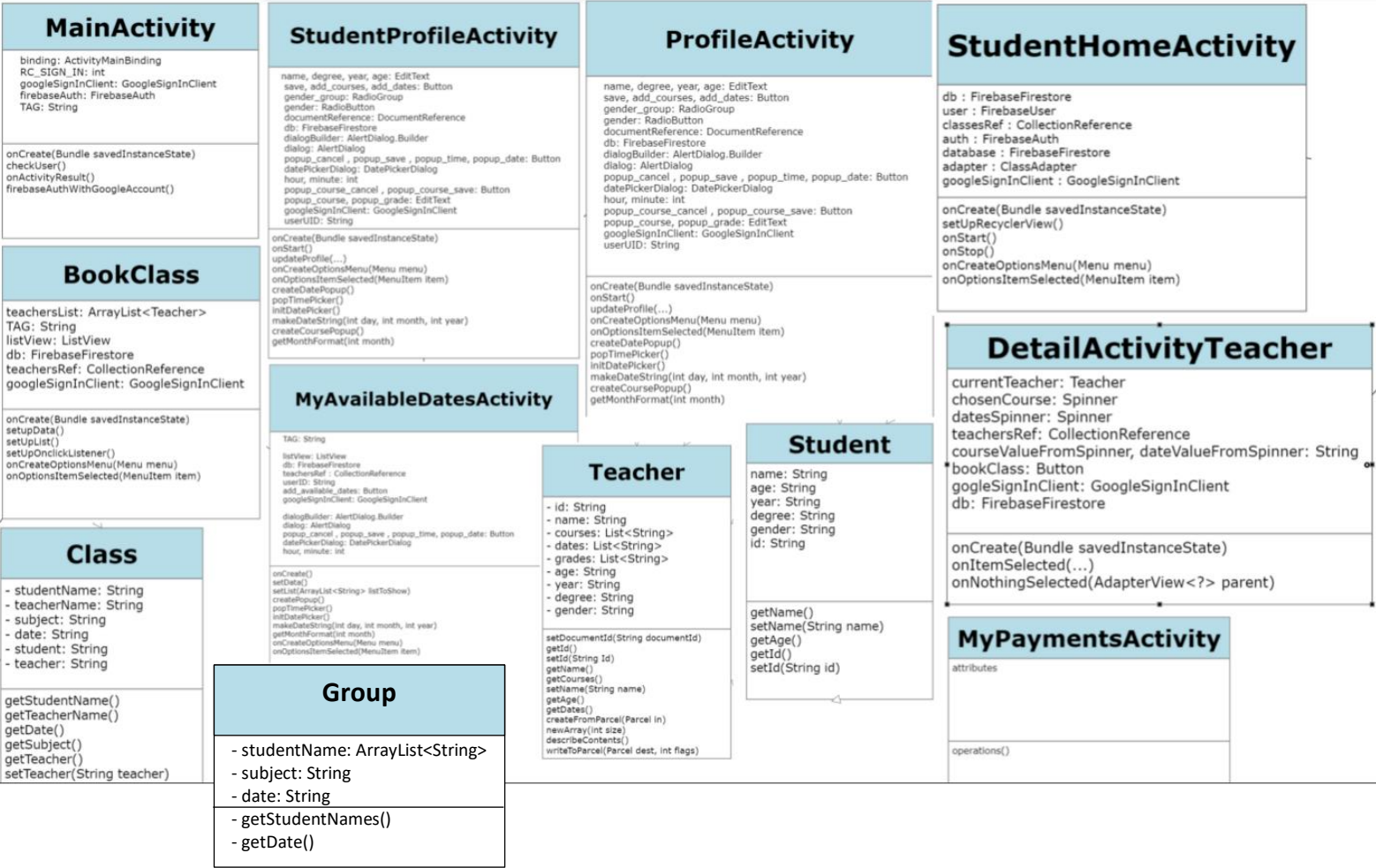
### 2. סקירה כללית של המערכת

האפליקציה שלנו בנויה עבור שני סוגי משתמשים: מורה ותלמיד. סטודנט יכול להרשם בתור שני הסוגים. ההרשמה מתבצעת דרך Google, ולאחר הרשמה ראשונית על הסטודנט למלא שאלות על פרטיו האישיים. בתור מורה, עליו גם לספק רשימת קורסים אותם הוא ילמד וכן שעות בהן הוא פנוי להעביר שיעורים באופן שבועי. באופן דומה עבור שתי המשתמשות, בעמוד הבית מוצגת רשימת התאריכים והשעות של השיעורים העתידיים שנקבעו, ומשם ניתן לבטל אותן וכן לשלוח למורה הודעה. מעמוד הבית גם ניתן לפתוח רשימת בחירה, המכילה אפשרויות לקביעת שיעורים נוספים, מציאת קבוצת לימוד, עדכון מידע אישי, תשלום, יציאה מהמשתמש ושינוי משתמש. כל המידע, כולל מידע אישי, פרטי השיעורים והקבוצות וכן הלאה נשמר ב-Firebase המקושר לפרויקט. בעת קביעת שיעור, ניתן למצוא מורה פרטי על ידי סינון לפי קורס, תאריכים ועוד. עיצוב האפליקציה נעשה בהשראת צבעי הלוגו שלה.

### 3. ארכיטקטורת המערכת

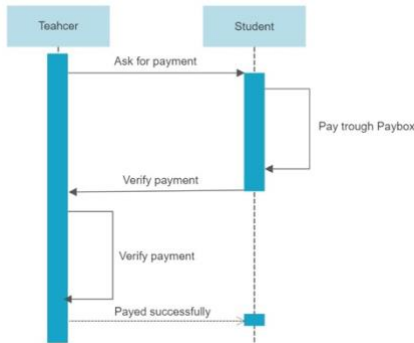
#### 3.1 עיצוב ארכיטקטורי

מצורפת מטה דיאגרמה המתארת את המבנה הכולל של התוכנה שלנו, כולל אובייקטים, הפונקציות השייכות להם, המשתנים שלהם וסוגיהם.

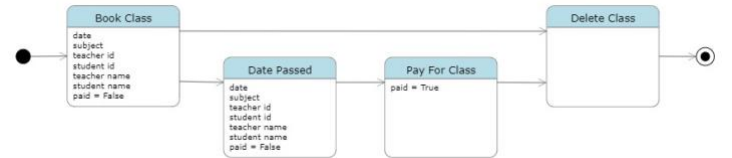


אובייקט שיעור מקושר לעמוד הבית של המורה ותלמיד וכן לעמוד הסיווג של שיעורים לפי קורסים ב DetailActivityTeacher. בנוסף, עמודי עדכון הפרטים האישיים מקושרים לאובייקטים של מורה ותלמיד בהתאמה, וכן עמוד קביעת השיעור מקושר למורה שילמד אותו והשעות האפשריות עבורו, ומשם לעמוד התשלום, התלמיד ודף הבית המראה את השיעורים והקבוצות שנקבעו. כל העמודים כאן מקושרים בצורה המאפשרת ניהול שיעורים וקביעת קבוצות למידה והשלמת פרטי מידע בצורה נוחה ופשוטה.

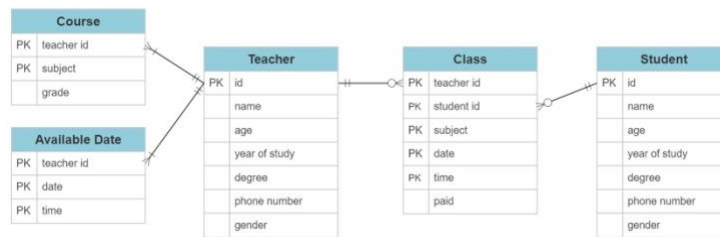
3.2 תיאור פירוק



בדיאגרמה זו מתואר הקשר בין התשלום לבין איש המשתמש, בין אם הוא מורה ובין אם הוא תלמיד. כפי שניתן לראות, התשלום נעשה דרך אפליקציית Paybox.



בדיאגרמה זו מתואר הקשר בין קביעת שיעור לבין הפונקציות המקושרות לclass זה. לאחר קביעת השיעור, נשמר התאריך בו השיעור נקבע, ניתן לשלם על השיעור ולמחוק אותו.



בדיאגרמה זו מוצג הקשר בין class של שיעור, המורה והתלמיד. גם מוסבר הקשר בין התאריכים הרלוונטיים עבור השיעור וכן הסינון לפי קורסים. התהליך אותו עובר התלמיד – בחירת קורס, בחירת מורה, תאריך רלוונטי, יצירת השיעור ע"י קישור התלמיד והמורה לשיעור זה, ושמירת התאריך הקורס הנבחרים.

### 3.3 היגיון עיצובי

נסביר את ההיגיון בבחירת המבנה שהצגנו ב3.1. בעת פיתוח האפליקציה, דנו בנושא מבנה המשתמש. התלבטנו בין הרשמה דרך Google ולאחר מכן בחירת סוג משתמש, לבין בחירת סוג משתמש ואז הרשמה. לבסוף בחרנו בדרך השנייה, שכן הבנו שהמידע שיוחזק בכל סוג יהיה שונה, ומכאן שזו תהיה הדרך היעילה ביותר להרשמה ולכניסה לאפליקציה. את class של DetailActivityTeacher בנינו בצורה הזו על מנת שנוכל לסנן שיעורים לפי מורים המלמדים את השיעור עבורו הסטודנט רוצה לקבוע שיעור.

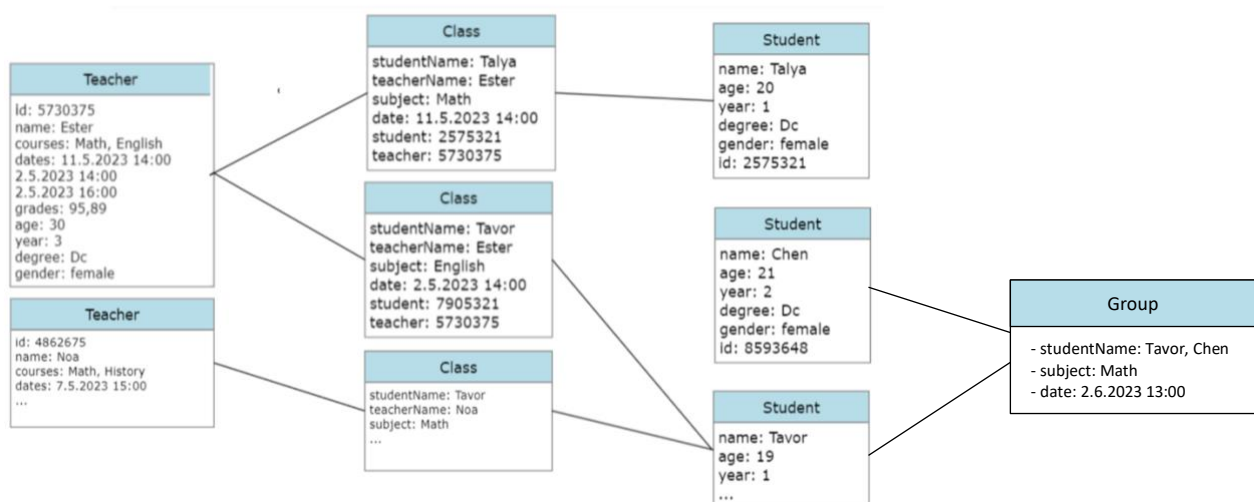
## 4. עיצוב נתונים

### 4.1 תיאור נתונים

כל אובייקט מסוג מורה, תלמיד, קורס, קבוצת למידה ושיעור מוחזק בתור אובייקט מסוג Document בבסיס הנתונים Firebase שבו אנו משתמשים. בכל אחד מהם הנתונים שלו מוחזקים בתור ArrayList, כל משתנה באובייקט מכיל כותרות ותוכן המקושר לכל כותרת (שם המסביר מה המידע המוחזק שם). לכל משתמש קיים מספר משתמש, איתו ניתן לזהות את המשתמש, ובעזרתו ניתן למצוא את השיעור ואת קבוצות הלמידה המקושרים אליו.

### 4.2 מילון נתונים

את החלק הבא נסביר באמצעות הדיאגרמה הבאה.



בדיאגרמה זו ניתן לראות את האובייקטים classes עליהם דובר קודם, וכן את המשתנים של כל אחד. לכל אובייקט קיימות פונקציות constructor וכן פונקציות לשינוי המשתנים ושליפתן (Getters and Setter). פירוט על הפונקציות הקיימות עבור הclasses האחרים ניתן למצוא ב3.2.

## 5. עיצוב רכיבים

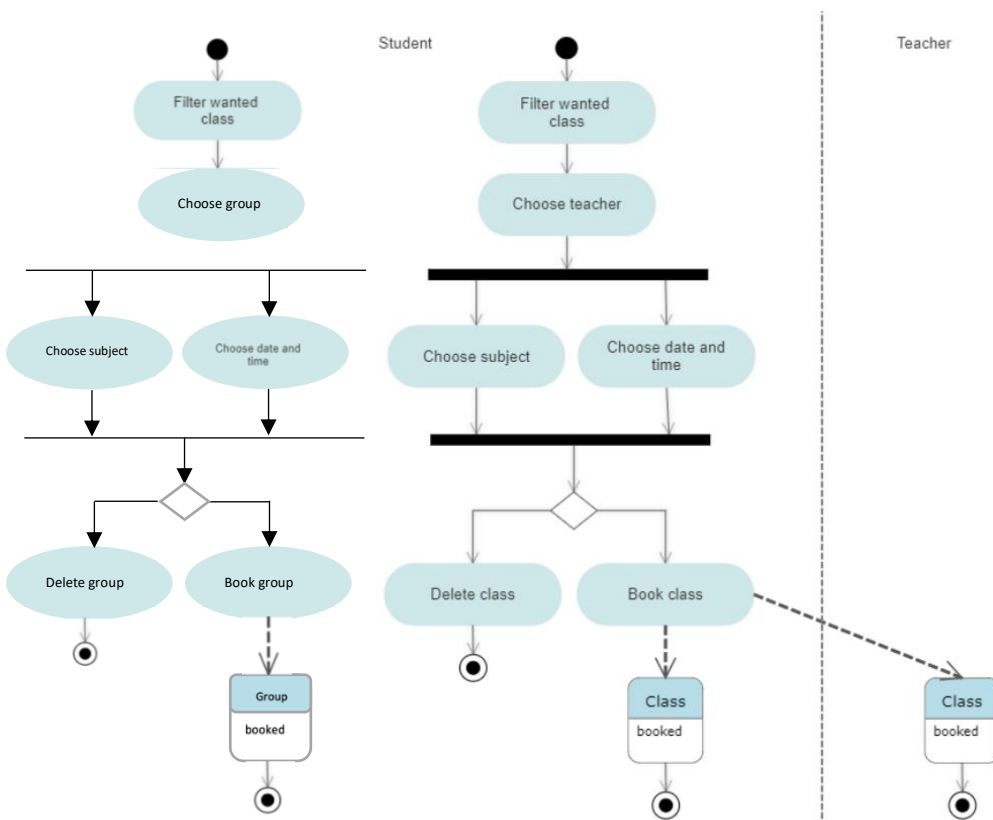
על מנת לקבוע שיעור, על התלמיד לכתוב עבור איזה קורס הוא מחפש שיעור. לאחר שכתב את שם הקורס, באובייקט DetailActivityTeacher מתרחש סינון מורים לפי שמות הקורסים אותם הם מלמדים. שמות המורים הרלוונטיים נשלפים ומוצגים למסך. מכאן מועברים לעמוד BookClass, שם נשלפת רשימת הקורסים אותם מלמד המורה ולאחר בחירת הקורס יישלפו התאריכים הרלוונטיים. מכאן נותר ללחוץ על Book Class, שייצור אובייקט חדש של שיעור, בו יישמרו המספרים המזהים של המורה והתלמיד, התאריך והקורס של השיעור. על מנת לשלם, המורה שולח לתלמיד בקשה לתשלום. התלמיד מאשר ומשלם דרך Paybox ולאחר מכן נשלח למורה אישור על מעבר תשלום. המורה מאשר את קבלת הכסף והכסף מועבר לחשבון הבנק שלו.

על מנת לקבוע קבוצת למידה עם קבות תלמידים, על תלמיד לבחור את הקורס בו הוא מעוניין למצוא קבוצת למידה ולאחר מכן לבחור מבין התלמידים האחרים הרלוונטיים. אחר כך התלמידים קובעים תאריך המתאים לכולם ומידע זה מעודכן במאגר הנתונים.

## 6. עיצוב ממשק משתמש

### 6.1 סקירה כללית של ממשק משתמש

בדיאגרמה זו מתואר כיצד המשתמש מתפעל את המערכת.



תחילה, המשתמש בוחר מאיזה סוג משתמש הוא רוצה להכנס למערכת.

המשתמש יכול למלא פרטים אישיים, להחליף סוג משתמש, ולראות בעמוד הבית אילו שיעורים וקבוצות הוא קבע שעתידים להתקיים.

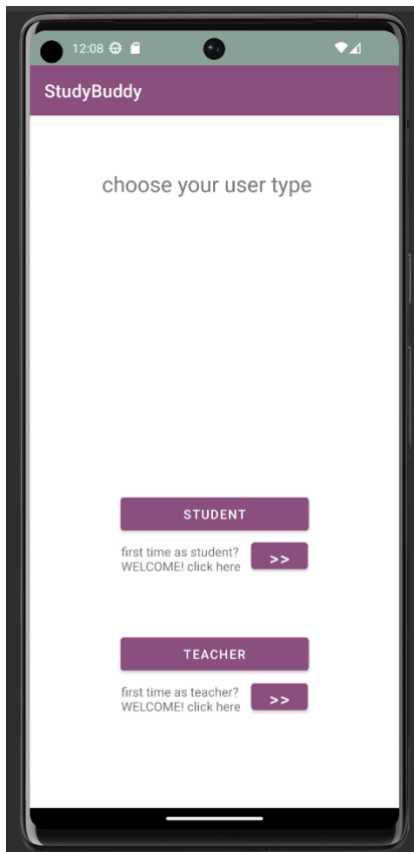
בתור סטודנט:

הוא יכול להכנס ל Book Class ולקבוע שיעור כפי שהראנו קודם או להכנס ל Book Group ולקבוע קבוצת למידה. בנוסף, הוא יכול לבטל שיעורים וקבוצות, ליצור קשר עם מורה דרך אפליקציית WhatsApp ולשלם על השיעורים.

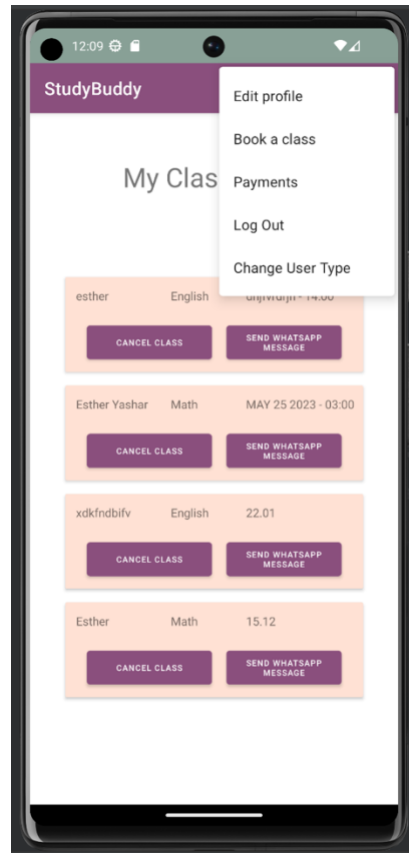
בתור מורה:

הוא יכול לבחור שעות בהן הוא פנוי ללמד שיעורים, קורסים אותם הוא יכול ללמד, לבקש תשלום מתלמידים וליצור קשר עם התלמיד דרך אפליקציית WhatsApp.

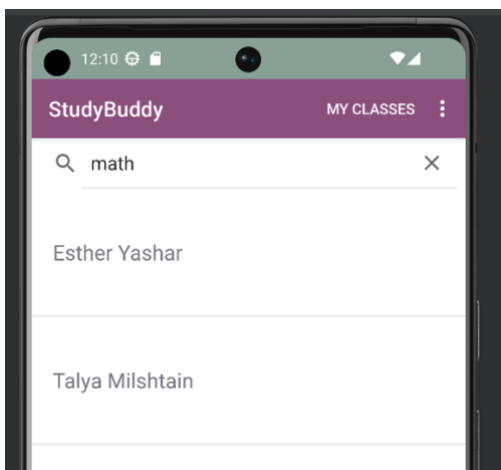
## 6.2 תמונות מסך



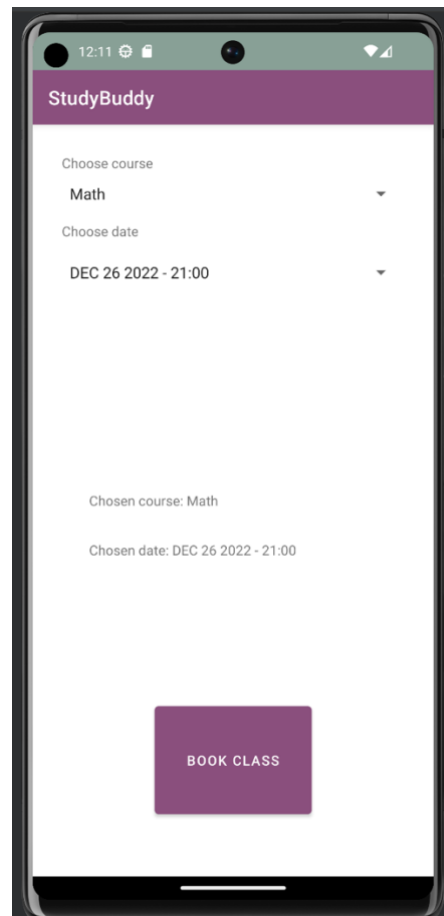
בחירת סוג  
משתמש והרשמה



עמוד הבית המראה  
אילו שיעורים נקבעו  
ועתידים לקרות, וכן  
בחירת פעולות נוספות



חיפוש מורים לפי  
שם קורס



קביעת שיעור לפי  
תאריך ושעה  
רלוונטיים

### 6.3 אובייקטים ופעולות מסך

לאחר לחיצה על כפתור Book Class, Book Groupי כלומר לאחר בחירת תאריך ושעה, מורה וקורס על ידי התלמיד, נוצר אובייקט חדש מסוג שיעור השומר את כל המידע הזה, או אובייקט חדש של קבוצה בהתאמה.

### 7. טבלת דרישות

נתייחס למסמך הדרישות שיצרנו עבור הפרויקט ונראה שמימשנו את הדרישות ואת הצרכים שקבענו:

מס' מזהה	תיאור	סוג הרשאה/משתמש	סוג (פונקציונאלית/לא פונקציונאלית)	הערות
1	קביעת שיעור	תלמיד	פונקציונאלית	עידכון אצל מורה ותלמיד על הזמן שנקבע עידכון שעות המורה לשעות הפנויות שנתרו
2	פרסום ימים ושעות	מורה	פונקציונאלית	עידכון פרטי המורה לימים והשעות הרלוונטים לקביעת שיעורים ופרסומם אצל התלמיד
3	הרשמה	תלמיד	פונקציונאלית	פתיחת חשבון תלמיד והכנסת פרטים רלוונטים למאגר נתונים
4	הרשמה	מורה	פונקציונאלית	פתיחת חשבון מורה והכנסת פרטים רלוונטים למאגר נתונים
5	עדכון פרטים	מורה	פונקציונאלית	עדכון שנתון לימודים, ממוצע ציונים, קורסים ללמידה
6	מעקב תשלום	תלמיד	פונקציונאלית	נרצה לאפשר למורה לעקוב אחר תשלומים מהתלמידים, למשל ע"י העלאת אישור תשלום מהתלמיד
10	עדכון שעות בזמן אמת	תלמיד	לא פונקציונאלית	נרצה לוודא שברגע שמורה מעדכן את שעות הפעילות שלו, המידע מתעדכן מיידית לנוחות התלמידים
11	עריכת שינויים ותיקונים במערכת	מנהלי התוכנה	לא פונקציונאלית	היכולת לעדכן ולשנות את הקוד בכדי לייעל להתאים לצרכי המשתמשים
12	ממשק משתמש נוח והבנה מהירה של איך להשתמש במוצר	מורה ותלמיד	לא פונקציונאלית	נכון לבנות ממשק פשוט להבנה
13	ביטול ועריכת שיעורים	תלמיד	פונקציונאלית	נרצה לוודא שמידע מתעדכן מיידית

14	סינון מורים לבחירה	תלמיד	פונקציונאלית	זאת על מנת לאפשר לתלמידים למצוא מורים המתאימים למילוי צרכיהם בצורה מדויקת כמה שיותר
15	פתיחת קבוצה	מנהלי התוכנה	פונקציונאלית	פתיחת קבוצה לפי תיאום בין תלמידים
16	קביעת קבוצה	תלמיד	פונקציונאלית	קבלת שיוך לקבוצה לפי התאמה וכן אישור הצטרפות או דחייה
17	סינון קבוצה לבחירה	תלמיד	פונקציונאלית	זאת על מנת לאפשר לתלמידים למצוא קבוצות המתאימות למילוי צרכיהם בצורה מדויקת כמה שיותר