



## מסמך עיצוב תוכנה R.Cook

### 3. Software Design Document

מגישה: רותם סויסה ת"ז: 207788027  
קורס: נושאים נבחרים - פיתוח בסביבת אנדרואיד  
תאריך: יולי 2025

## 1. תיאור ארכיטקטורת התוכנה (Software Architecture Description)

### 1.1. דפוס ארכיטקטורה (MVVM - (Model-View-ViewModel

אפליקציית R.Cook פותחה לפי דפוס הארכיטקטורה MVVM – Model-View-ViewModel. הארכיטקטורה מספקת הפרדה ברורה בין רכיבי המערכת ומבטיחה קוד נקי, בר-תחזוקה ובר-בדיקה.

### 1.2. שכבות הארכיטקטורה

#### 1.2.1. שכבת ממשק המשתמש (UI Layer)

##### Activities עיקריות:

- WelcomeActivity - מסך הכניסה וההרשמה הראשונית
- MyLessonsActivity - מסך עיקרי המציג את רשימת השיעורים
- LessonDetailActivity - מסך צפייה בפרטי שיעור בודד
- EditLessonActivity - מסך הוספה ועריכה של שיעורים (למדריכים)
- SettingsActivity - מסך הגדרות המשתמש
- BaseActivity - מחלקת בסיס משותפת לכל ה-Activities

##### Fragments:

- LessonsFragment - מציג רשימת שיעורים לפי רמה (מתחילים, בינוניים, מתקדמים)

##### Adapters:

- LessonsAdapter - מתאם לתצוגת רשימת השיעורים
- LevelsPagerAdapter - מתאם לניהול כרטיסיות הרמות

#### 1.2.2. שכבת ViewModel

LessonViewModel מתווכת בין שכבת ה-UI לשכבת הנתונים:

- מנהלת LiveData לעדכון אוטומטי של ה-UI
- מבצעת פעולות אסינכרוניות על מסד הנתונים
- שומרת על מצב הנתונים במהלך שינויי קונפיגורציה

#### 1.2.3. שכבת ניהול נתונים (Data Layer)

##### DataManager:

- מחלקה מרכזית המנהלת את כל הגישה לנתונים
- מגשרת בין Room Database ל-SharedPreferences
- מספקת API מאוחד לכל פעולות הנתונים

##### Room Database:

- RCookRoomDatabase - מסד נתונים מקומי
- LessonDao - ממשק לגישה לנתוני השיעורים
- UserDao - ממשק לגישה לנתוני המשתמש

#### 1.2.4. שכבת Entities

- Lesson - ישות המייצגת שיעור בישול
- User - ישות המייצגת את המשתמש

#### 1.2.5. שכבת Utils

- BackgroundColorUtils - כלי עזר לניהול צבעי רקע

### 1.3. זרימת הנתונים

User Action → Activity/Fragment → Adapter → ViewModel → DataManager → DAO → Room Database → Entity



## R.Cook עיצוב תוכנה

### 3. Software Design Document

## 2. ישויות תוכנה – DAO (Software Entities)

האפליקציה כוללת שתי ישויות נתונים מרכזיות הנשמרות במסד הנתונים Room :

### 2.2. ישות Lesson

```
@Entity(tableName = "lesson_table")
public class Lesson {
    @ PrimaryKey
    @ NonNull
    private String id;

    private String name;
    private String guideName;
    private String shortDescription;
    private String fullDescription;
    private String imageUrl;
    private String videoUrl;
    private String level;
    private boolean completed;
    private boolean favorite;
}
```

### 2.1. ישות User

```
@Entity(tableName = "user_table")
public class User {
    @ PrimaryKey
    @ NonNull
    private String id = "current_user";

    private String firstName;
    private String lastName;
    private String role;
    private String level;
    private String phone;
    private int age;
    private String gender;
    private String backgroundColor;
}
```

### 2.3. ממשיקי DAO

#### 2.3.2 UserDao

```
@Dao
public interface UserDao {
    @ Insert(onConflict = OnConflictStrategy.REPLACE)
    void insertOrUpdate(User user);

    @ Query("SELECT * FROM user_table WHERE id = 'current_user'")
    LiveData<User> getCurrentUser();

    @ Query("SELECT * FROM user_table WHERE id = 'current_user'")
    User getCurrentUserSync();

    @ Query("DELETE FROM user_table WHERE id = 'current_user'")
    void deleteCurrentUser();
}
```

#### 2.3.1 LessonDao

```
@Dao
public interface LessonDao {
    @ Insert(onConflict = OnConflictStrategy.REPLACE)
    void insert(Lesson lesson);

    @ Update
    void update(Lesson lesson);

    @ Delete
    void delete(Lesson lesson);

    @ Query("SELECT * FROM lesson_table")
    LiveData<List<Lesson>> getAllLessons();

    @ Query("SELECT * FROM lesson_table WHERE level = :level")
    LiveData<List<Lesson>> getLessonsByLevel(String level);

    @ Query("UPDATE lesson_table SET favorite = :favorite WHERE id = :id")
    void updateFavoriteStatus(String id, boolean favorite);

    @ Query("UPDATE lesson_table SET completed = :completed WHERE id = :id")
    void updateCompletionStatus(String id, boolean completed);
}
```



## R.Cook עיצוב תוכנה

### 3. Software Design Document

### 3. תיאור עיצוב מסד נתונים (Database Design Description)

3.1. מבנה מסד הנתונים האפליקציה משתמשת ב, Room Database-המספק שכבת אבסטרקציה מעל SQLite.

#### 3.2 טבלאות מסד הנתונים

טבלת השיעורים - `lesson_table`:

שדה	סוג	תיאור
id	String (PK - Primary Key)	מזהה ייחודי לשיעור
name	String	שם השיעור
guide_name	String	שם המדריך שהוסיף את השיעור
short_description	String	תיאור קצר
full_description	String	תיאור מלא ומתכון מפורט
image_url	String	תמונה מקומית לשיעור
video_url	String	מזהה וידאו לשיעור
level	String	רמת קושי: Beginners, Medium, Advanced
completed	boolean	האם השיעור הושלם ע"י המשתמש
favorite	boolean	האם השיעור סומן באהוב

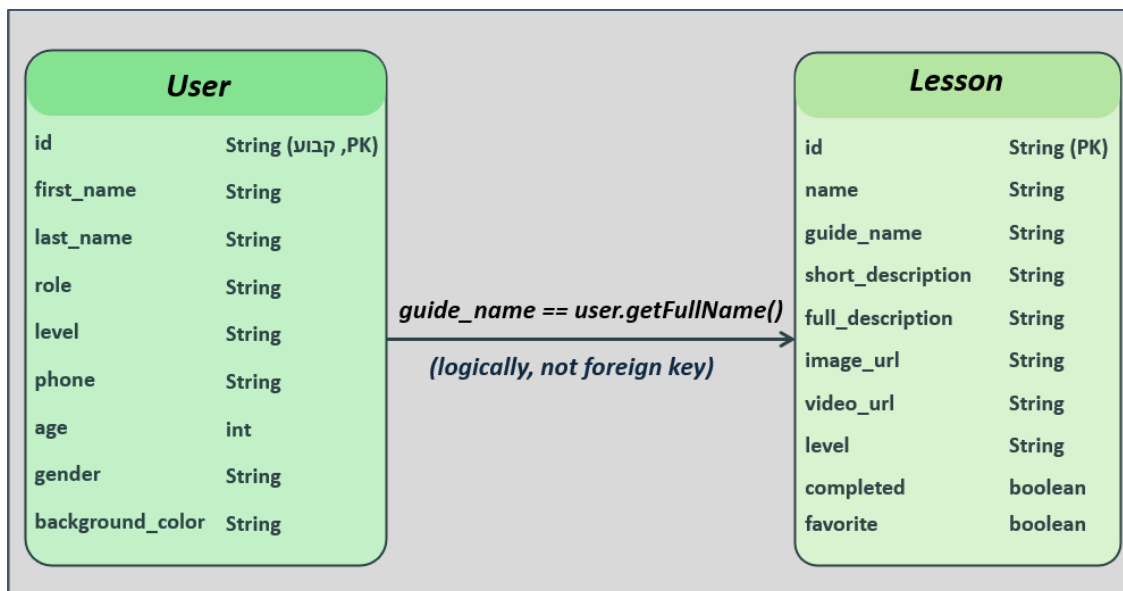
טבלת המשתמשים - `user_table`:

שדה	סוג	תיאור
id	String (PK, קבוע)	"current_user" - מזהה של המשתמש היחיד באפליקציה
first_name	String	שם פרטי
last_name	String	שם משפחה
role	String	תפקיד המשתמש - Student או Guide
level	String	רמת קושי מועדפת - משפיעה על ברירת המחדל של הטאב
phone	String	טלפון
age	int	גיל
gender	String	מין
background_color	String	צבע רקע מועדף: gray, blue, yellowish, default

#### 3.3. אינדקסים ואילוצים

- Primary Keys: מבטיחים ייחודיות הרשומות
- Foreign Keys: לא נדרשים במבנה הנוכחי
- Validation: מתבצעת ברמת האפליקציה (גיל 1-120, טלפון מעל 10 ספרות)

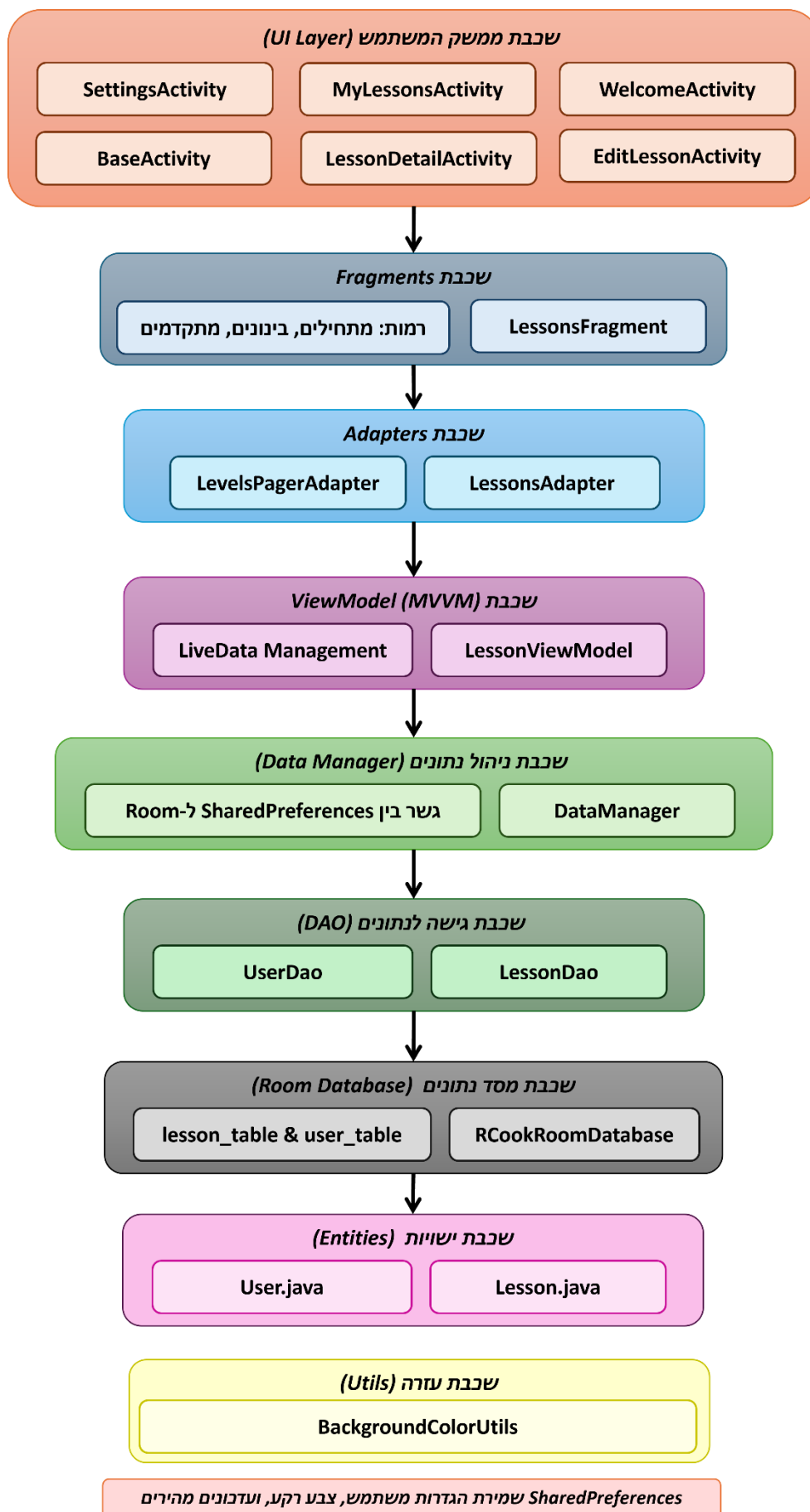
תרשים ER (Entity Relationship Diagram) להמחשה:





*R.Cook* מסמך עיצוב תוכנה  
3. Software Design Document

4. תרשים ארכיטקטורת המערכת (System Architecture Diagram)





## מסמך עיצוב תוכנה R.Cook

### 3. Software Design Document

#### זרימת הנתונים באפליקציה:

המשתמש מבצע פעולה ב-Entity → Room Database → DAO → DataManager → ViewModel → Adapter → Activity/Fragment → UI

#### תכונות מרכזיות של הארכיטקטורה:

- MVVM Pattern: הפרדה בין UI, לוגיקה עסקית ונתונים
- Room Database: מסד נתונים מקומי עם LiveData
- SharedPreferences: שמירת הגדרות וגיבוי
- LiveData: עדכון אוטומטי של UI בעת שינוי נתונים
- BaseActivity: מבנה משותף לכל מסכי האפליקציה

#### סיכום

- האפליקציה מבוססת על עקרונות MVVM, עם שימוש ב-Room לניהול נתונים, ו-LiveData לעדכון דינאמי של ה-UI.
- כל נתוני המשתמש והשיעורים נשמרים במסד נתונים.
- קיים ממשק שלם לניהול שיעורים (הוספה, עריכה, סימון כמועדף או הושלם).
- ה-UI מותאם אישית לפי המשתמש (רקע).
- תהליך הפיתוח מ-SharedPreferences ל-Room תוך שמירה על תאימות לאחור (דרך DataManager).

ארכיטקטורת R.Cook מספקת מבנה יציב, מודולרי ובר-תחזוקה. השימוש בדפוס MVVM יחד עם Room Database מבטיח ביצועים טובים, נוחות פיתוח ויכולת הרחבה עתידית. המבנה תומך בכל הדרישות הפונקציונליות של האפליקציה תוך שמירה על עקרונות עיצוב טובים.