

תרגיל הגשה #3

חומר הלימוד בתרגיל: SQLite, Custom ListView, Alarm Manager, Calendar, Date&Time Picker

מועד אחרון להגשה: 13.8.2020

משקל התרגיל: 50% (מציון התרגילים)

הקדמה

בתרגיל זה תכתבו אפליקציה לניהול משימות (ToDoApp). המשתמש יוכל להזין משימות שהוא רוצה לבצע ("ללכת לקניות", "לסיים תרגיל בתכנות" וכו') עם תזכורת לתאריך ושעת יעד. האפליקציה תוכל לשרת מספר משתמשים מאותו מכשיר ולכן המשתמש יצטרך להזין שם משתמש וסיסמה, כדי לנהל את המשימות האישיות שלו. כל המשתמשים וכל המשימות ישמרו במסד נתונים SQLite בשם `todo.db` בשתי טבלאות נפרדות.

תיאור האפליקציה

האפליקציה תורכב משלושה מסכים (Activities) כמתואר כאן:

צילום מסך	קובץ ה-UI (ממשק משתמש)	קובץ הקוד	תיאור
1	<code>activity_login.xml</code>	<code>LoginActivity.java</code>	מסך התחברות
2	<code>activity_todolist.xml</code>	<code>ToDoListActivity.java</code>	מסך המשימות
3,4	<code>activity_editor.xml</code>	<code>EditorActivity.java</code>	מסך עורך משימה

מסך התחברות

האקטיביטי `LoginActivity` תאפשר למשתמש להיכנס לאפליקציה. האקטיביטי יכיל שני שדות טקסט – אחד לשם משתמש ואחד לסיסמה, וכן כפתור `Login` לכניסה (צילום מסך 1).

אופן פעולת האקטיביטי בלחיצה על הכפתור `Login`:

- אם שם המשתמש **כבר קיים** במערכת, נבדוק האם הסיסמה תואמת את סיסמתו. אם לא, תצא הודעת שגיאה `Toast` על סיסמה שגויה. אחרת יפתח האקטיביטי `ToDoListActivity` (הסבר בהמשך).
- אם שם המשתמש **אינו קיים** במערכת, ייווצר שם משתמש חדש ותירשם הסיסמה, ויפתח האקטיביטי `ToDoListActivity` (הסבר בהמשך).
- אם חסרים אחד או יותר מהפרטים (שם משתמש או סיסמה) תצא הודעת שגיאה `Toast` מתאימה.

כדי לממש התחברות/רישום של משתמש במערכת, האפליקציה צריכה לתחזק במסד הנתונים `todo.db` **טבלה בשם users** שתכיל את פרטי המשתמשים. בטבלה זו יהיו שתי עמודות:

- username שם משתמש** (מחרוזת כמפתח ראשי). יש לוודא **שלא** קיימים שני משתמשים באותו שם
- password סיסמה** (מחרוזת).

בנוסף, שימו לב שהאפליקציה צריכה לדעת באקטיביטי הבא (`ToDoListActivity`) איזה משתמש נכנס – שמו יופיע בסוגריים בכותרת האקטיביטי – ולכן יש להשתמש ב-`Shared Preferences` כדי לשמור פרטים אילו. **שימו לב:** בזמן פתיחת האקטיביטי `LoginActivity`, אם ב-`Shared Preferences` כבר קיימים פרטי משתמש – כלומר, משתמש ביצעה כבר התחברות, יפתח באופן אוטומטי האקטיביטי הבא (`ToDoListActivity`).

מסך המשימות

האקטיביטי `ToDoListActivity` תאפשר למשתמש רשום **לצפות** במשימות הקיימות שלו, **להוסיף** משימות חדשות, **לחפש** משימות ולצאת מהמערכת (`Logout`) (ראו צילום 2). האקטיביטי יכיל את הפרטים הבאים:

1. שם המשתמש שנכנס למסך יוצג בסוגריים ב- `ActionBar` לאחר שם האקטיביטי.
2. רכיב מסוג `SearchView` לחיפוש משימות ע"פ כותרת או תיאור המשימה.
3. `ListView` לתצוגת כל המשימות של המשתמש.
4. כפתור `FloatingActionButton` (כפתור עול עם הסימן +) הוספת משימה חדשה.
5. אייקון תפריט `LogOut` ליציאה מהמערכת.

אופן פעולת האקטיביטי –

- לחיצה על אייקון תפריט `LogOut` תמחק את פרטי המשתמש מה- `Shared Preferences` וחזרו לאקטיביטי `LoginActivity`.

- לחיצה על כפתור `FloatingActionButton` (+) להוספת משימה חדשה – יפתח את האקטיביטי `EditorActivity` במצב **הוספה** (צילום מסך 3) שבתוכו יוזנו נתוני המשימה החדשה שתישמר במסד הנתונים `todos.db` **בטבלה בשם `todos`** שתכיל את פרטי המשימות. בטבלה זו יהיו חמש עמודות:
 - **`_id` מזהה משימה** (מספר שלם אוטומטי כמפתח ראשי) כדי למנוע כפילויות ולאפשר זיהוי משימה.
 - **`username` שם משתמש** (מחרוזת) אליו שייכת משימה זו.
 - **`title` כותרת המשימה** (מחרוזת).
 - **`description` תיאור המשימה** (מחרוזת).
 - **`datetime` תאריך-שעה** (מספר שלם ארוך) לצורך ביצוע תזכורת.

- הנתונים שיש להזין באקטיביטי `EditorActivity` הם: **`title`** כותרת המשימה, **`description`** תיאור המשימה, **`date`** תאריך המשימה בפורמט `dd/mm/yyyy` אותו ניתן להקליד ידני או לבחור מה- `Date Picker Dialog` (צילום מסך 5), **`time`** זמן המשימה בפורמט `hh:mm` אותו ניתן להקליד ידני או לבחור מה- `Time Picker Dialog` (צילום מסך 6).
- השתמשו באובייקט מסוג `SimpleDateFormat` המכיל שיטה בשם `parse` שהופכת מחרוזת לתאריך, ושיטה בשם `format` שהופכת תאריך למחרוזת לפי פורמט מסוים.
- חשוב לציין שאת התאריך והזמן יש להמיר **למספר אחד שלם ארוך**, אותו יש לשמור במסד הנתונים בטבלה `todos` בעמודה `datetime`. השתמשו במחלקה `Calendar` כדי לבצע המרה זו.
- אם אחד מהפרטים חסרים (כותרת/תיאור/תאריך/שעה) או שהתאריך/שעה אינם חוקיים, תצא הודעת שגיאה `Toast` מתאימה. אחרת, שמרו המשימה בטבלה `todos` שבמסד הנתונים והציגו הודעה `"Todo was added to DB"`.

- **לחיצה ארוכה** על משימה ברשימה תאפשר **למחוק** את המשימה ממסד הנתונים. לפני המחיקה תוקפץ הודעה מסוג `Alert Dialog` שתוודא האם לבצע את המחיקה – כן או לא. אם כן, תמחק המשימה, תוצג הודעת `Toast` מתאימה ורשימת המשימות תעודכן מיד. אם לא, לא יתבצע דבר.

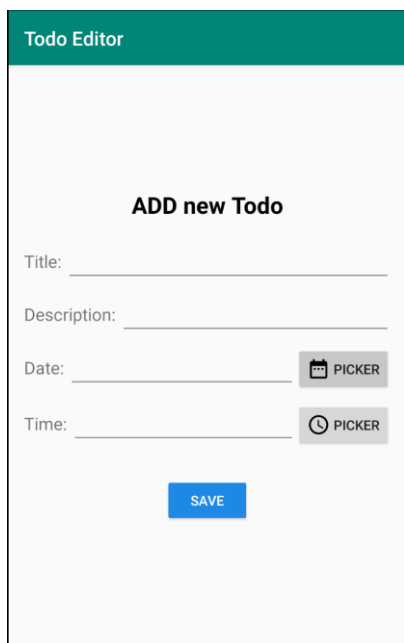
- **נגיעה** במשימה מסוימת ברשימה תאפשר **לעדכן** את פרטי המשימה ע"י פתיחת האקטיביטי EditorActivity במצב **עדכון** (צילום מסך 4) שלתוכו יטענו נתוני המשימה שנבחרה לעדכון (יש להשתמש במזהה `_id` של המשימה). כעת המשתמש יוכל לעדכן את פרטי המשימה ולשמור השינויים במסד הנתונים `todo.db` **בטבלה todos**. גם במצב זה יבדקו כל הפרטים המוזנים תקינים. בסיום ביצוע העדכון בהצלחה, תוצג הודעת Toast מתאימה.
- הקלדת טקסט ברכיב **SearchView** יאפשר **חיפוש** משימות ע"פ כותרת או תיאור המשימה. עבור כל אות שתוקלד יתבצע חיפוש אוטומטי ויוצגו ברשימה רק המשימות שבהן מופיע הטקסט שהוקלד. לחיצה על ה-X תאפס את החיפוש (ימחק הטקסט שהוקלט) ורשימת המשימות תציג את כל המשימות.
- לאחר שמשימה הוספה/עודכנה המערכת תיצור **תזכורת באמצעות Alarm שתקפיץ Notification** עם שם המשתמש וכותרת המשימה בתאריך והשעה שהוזנו במשימה. אם התאריך והשעה הם בעבר לא

דגשים והערות נוספות

- יש להקפיד על שם האפליקציה ושמות המחלקות כפי שנכתבו בטבלה שבתחילת התרגיל.
- הקפידו שכל רכיבי ה-UI יראו בשלמותם במסכים השונים. אין צורך לדאוג להתאמה בעת סיבוב המכשיר.
- הקפידו על כללי תכנות נכונים כפי שהוסבר בהרצאה (כתבו קוד **יעיל**, הפרידו בין הלוגיקה לתצוגה, הגדירו מחלקות ייעודיות, השתמשו נכון ב OOP בירושה, פולימורפיזם וממשקים).
- במסך LoginActivity בלחיצה על תפריט 3 הנקודות, יפתח תפריט ובו תת-תפריט **About** שלחיצה עליו תקפיץ AlertDialog (תיבת דו-שיח) ובו יוצגו פרטים אודות האפליקציה ושמות המתכנתים (שמות שני הסטודנטים שכתבו את האפליקציה).
- **דרישות טכניות לפיתוח האפליקציה – חובה!**
 1. שם ה-`package` של האפליקציה יהיה בנוי כך: `com.<student1name>_<student2name>.ex3` (לדוגמה: `com.ilanp_oritm.ex3`)
 2. קוד האפליקציה **יכתב** בסביבת הפיתוח **Android Studio 4.0** ומעלה.
 3. האפליקציה **תרוץ ללא בעיות על אמולטור Nexus 5X** עם מערכת הפעלה **Android 10 (API 29)**.

הנחיות הגשה

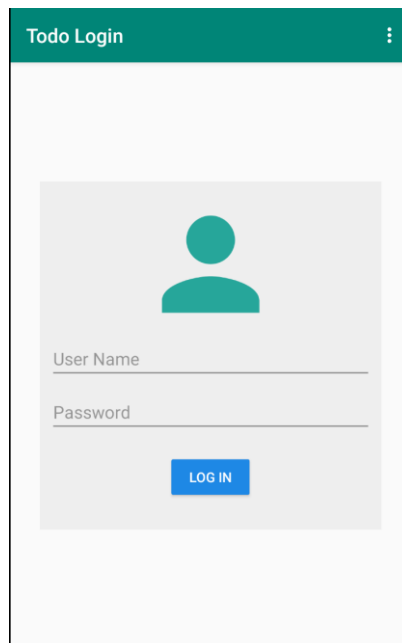
- את **תיקיית קוד הפרויקט** (המכווץ) וקובץ ה-`apk` יש **לארוז בקובץ יחיד** בשם `Ex3.zip`.
(יצירת קובץ `apk` דרך: `Build ⇒ Build APK(s)`. כיווץ תיקיית הפרויקט דרך: `File ⇒ Export to Zip file`)
- יש **להעלות** את הקובץ `Ex3.zip` **לאתר הקורס במערכת הגשת מטלות**.
- ההגשה **בזוגות בלבד**.
- מועד אחרון להגשת התרגיל בתאריך **13.8.2020** עד השעה **23:59**.

צילומי מסך

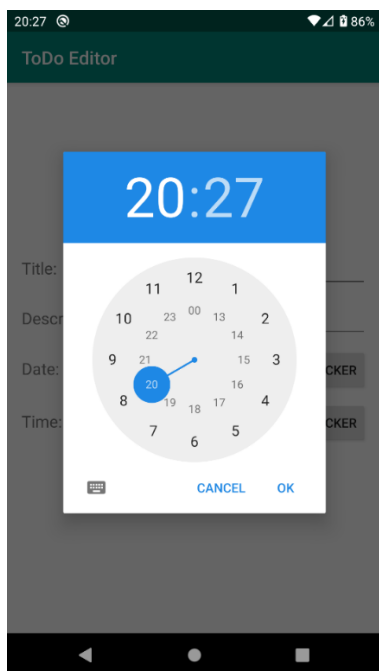
צילום 3 – מסך Editor (ADD)



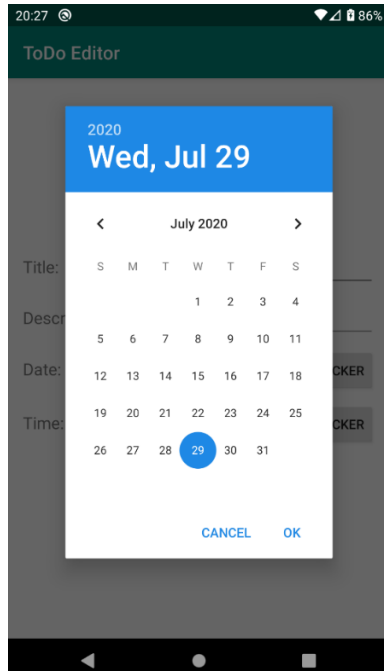
צילום 2 – מסך Todo List



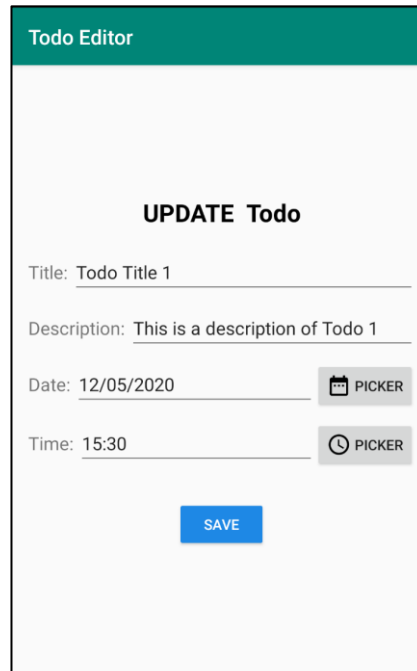
צילום 1 – מסך Todo Login



צילום 6 – Time Picker Dialog



צילום 5 – Date Picker Dialog



צילום 4 – מסך Editor (UPDATE)

בהצלחה!