

תוכן עניינים

2	הנחיות כלליות להגשה
3	נושא הפרויקט
3	תאור המערכת בצורה כללית
6	ישויות
7	שלבי ההגשה וניקוד עבור כל שלב:
7	שלב 1
7	חלק ראשון
8	חלק שני
8	חלק שלישי
9	שלב 2
9	חלק ראשון
9	חלק שני
9	חלק שלישי
10	שלב 3
10	חלק ראשון
11	חלק שני
12	חלק שלישי
14	שלב 4
15	דוגמאות לצילומי מסך מתוך האפליקציה

הנחיות כלליות להגשה

✓ הפרויקט יעבוד בתאימות מלאה תחת הגרסה:

```
android {  
    compileSdkVersion 25  
    buildToolsVersion "25.0.3"  
  
    defaultConfig {  
        applicationId "....."  
        minSdkVersion 15  
        targetSdkVersion 25  
        versionCode 1  
        versionName "1.0"  
    }  
}
```

הדוח הסופי יכלול:

- ✓ שרטוט דיאגרמה בסגנון UML שמתארת את הישויות והקשרים ביניהן.
- ✓ צילום של כל המסכים בצורה היררכית ותיאור האינטגרציה בין המסכים.
- ✓ כל אחד חייב להוסיף מדריך למשתמש כלומר דוקומנטציה שמכיל את הגדרות האמולטור + את סיסמאות הכניסה והוראות הפעלה בסיסיות.
- ✓ תיעוד של בעיות שנתקלתם בהן במהלך פתוח הפרויקט, תיעוד של הדרך שבה פתרתם את הבעיות האלה וכמו"כ תיעוד של קוד יצירתי שחורג ממה שנלמד במהלך השיעורים בכתה.
- ✓ כמו"כ יש לצרף לתיעוד הסופי הערות ענייניות בלבד ומשוב קצר על הקורס.

דגשים:

- ✓ את הקוד חייבים לתעד באמצעות JavaDoc.
- ✓ המשקל של התיעוד הפנימי בתוך הקוד + התיעוד הסופי בקובץ doc נפרד כנ"ל שווה ל- 5% בציון הסופי.
- ✓ ציון ההגנה יינתן בסוף הפרויקט ע"ס בדיקה מדגמית מתוך מכלול הנושאים שהוטמעו בתרגול ובפרויקט. הבדיקה תתבצע בצורה קשוחה. אי לכך תידרש שליטה מליאה של כל אחד מבין השותפים בנפרד בכל אחד הנושאים של התרגול והפרויקט.
- ✓ הציון הסופי הינו הערכה בלבד והכל נתון לשיקול דעתו הפרטית של המרצה.

כל שלב משלבי הפרויקט יש לשמור בקובץ zip בשם Android5778_XXXX_YYYY_00 כך ש XXXX זה 4 ספרות אחרונות בת"ז של בן הזוג הראשון ו YYYY זה 4 ספרות אחרונות בת"ז של בן הזוג השני ובמקום 00 יש לרשום את מספר השלב בפרויקט.

יש להעלות כל שלב משלבי הפרויקט למודל, לפי הנחיות המרצה !!!

נושא הפרויקט

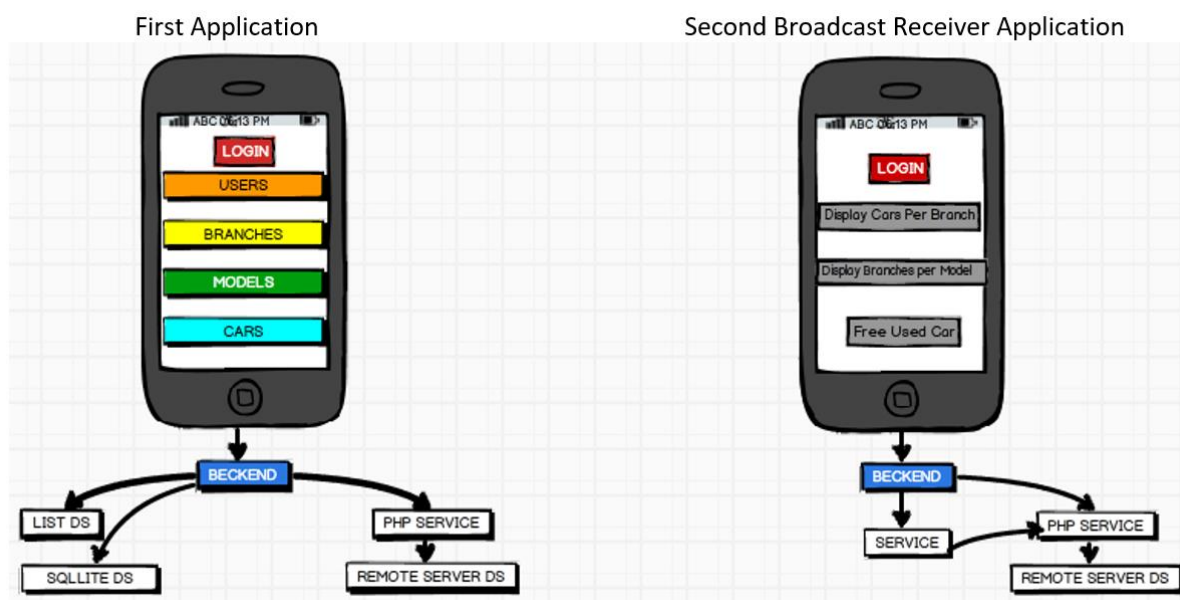
בפרויקט תתבקשו לכתוב שתי אפליקציות שמשרתות את חברת השכרת המכוניות "Take&Go" שעושה שימוש בשיטת השכרה מהפכנית.

החברה מעמידה לרשות לקוחותיה מכוניות להשכרה לשימוש פרטי בלבד. המכוניות חונות במקומות חניה מוסדרים בערים ובפריסה ארצית.

כל לקוח יכול לרשום את עצמו באופן חד-פעמי במאגר הלקוחות של החברה. הלקוח מקבל מפתח אוניברסלי למכוניות החברה וכן קוד פתיחה ולאחר מכן הוא מוזמן לעשות שימוש במערכת, לחפש מכונית הקרובה למקום בו הוא נמצא, לשכור את המכונית, להשתמש בה ולהחזירה למקום החנייה המוסדר שממנו המכונית נלקחה. הלקוח פטור מחובת אחזקה של המכונית ואף פטור ממילוי דלק וכו'. סכום החיוב של הלקוח מחושב ע"פ מספר הקילומטרים שהוא נהג ומספר השעות שהוא השתמש במכונית. החיוב של הלקוח מתבצע באמצעות כרטיס אשראי.

הרציונל המסחרי: ליצור מענה חילופי נגיש, זול ומהיר לתחבורה ציבורית המקרטעת.
הרציונל המקצועי-טכני: הטמעת כל המרכיבים הבסיסיים של אפליקציה תחת מערכת ההפעלה אנדרואיד.

תאור המערכת בצורה כללית



הסבר:

האפליקציה הראשונה משמשת את חברת השכרת הרכב, על מנת לנהל את הרכבים, הלקוחות והסניפים. האפליקציה השנייה משמשת את הלקוחות על מנת לבצע הזמנה והחזרה של רכב.

כמובן ש 'REMOTE SERVER DS' יש רק אחד ושתי האפליקציות מתקשרות מול אותו 'REMOTE SERVER DS'.

כל הזכויות שמורות למחלקת המחשבים של המרכז האקדמי לב אין להעתיק ולהעביר בדרך כלשהי ללא רשות

המערכת בנויה משלושה מרכיבים כאשר הפיתוח של כל מרכיב יתבצע בשלב נפרד כדלהלן:

1. האפליקציה הראשונה תבצע ארבעה תפקידים כדלהלן:

- א. מבצעת Registering או Login לאפליקציה.
- ב. מכניסה נתונים (פרטי סניף חדש, פרטי דגם חדש, פרטי מכונית חדשה) ל- DS שלה. יש לכתוב ממשק מתאים להכנסת הנתונים.
- ג. יצירת Data Source חדש שיהיה בסיס נתונים מסוג MySQL שימוקם בענן ויצירת PHP Service פשוט שאף הוא ימוקם בענן ושיאפשר גישה דו-כיוונית לבסיס הנתונים. לאחר מכן נממש מחדש את ה- Backend ע"מ שיוכל לעשות שימוש ב- PHP Service.
- ד. בהמשך הכנסות לקריאות נתונים של דגם חדש תתבצע באמצעות - Content Provider

2. האפליקציה השנייה ייעודית רק עבור קהל הלקוחות ומיועדת לשרת רק אנשים שמעוניינים להשתמש בשירותי החברה.

האפליקציה מתפקדת גם כ- Broadcast Receiver. הפעולה הראשונה שיש לבצע באפליקציה היא כמובן Login ובמקרה שלמערכת אין עדיין את פרטי הלקוח אז הוא יתבקש לבצע Registering.

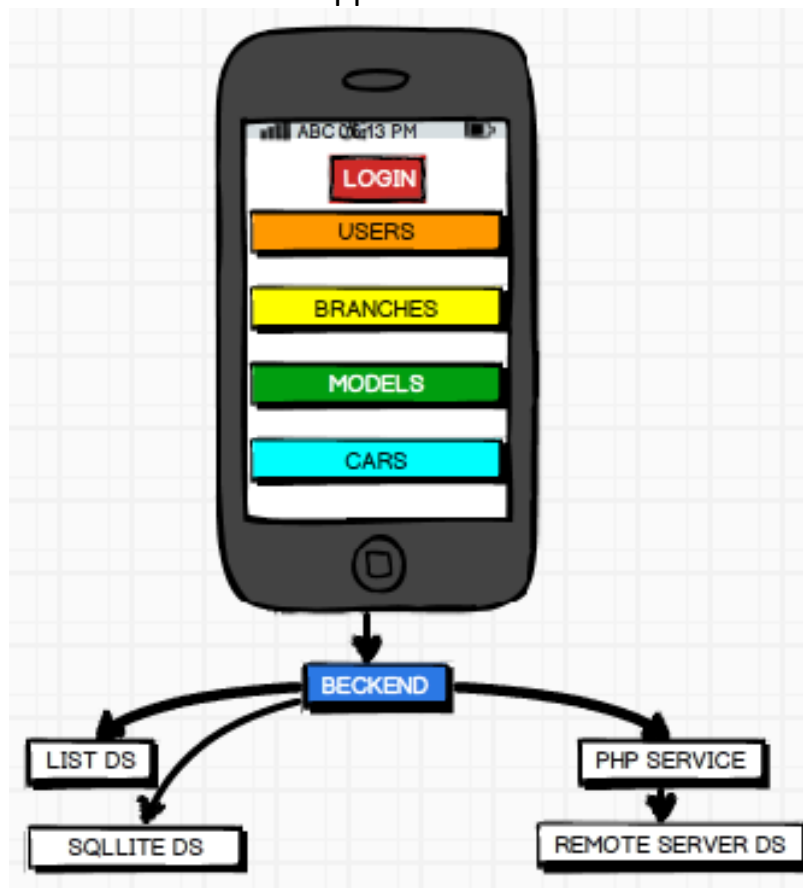
ה- Activity הראשי מכיל: Toolbar, Filtering וגם Navigation Drawer המכיל 5 אפשרויות ואלו הן:

- באמצעות האפשרות הראשונה ניתן יהיה לקבל את פרטי החברה "Take&Go" שהם: מספר טלפון, כתובת דוא"ל, קישור לאתר החברה. ניתן יהיה להתקשר לחברה או לשלוח דוא"ל.
- באמצעות האפשרות השנייה ניתן יהיה להציג את רשימת הסניפים (סניף=מקום חניה מוסדר עבור חברת "Take&Go"). כל סניף יופיע עם פרטיו הבסיסיים בלבד כולל לינק למפת הגעה לסניף. ניתן לסנן (Filtering) סניפים ע"פ העיר. עם בחירת אחד הסניפים יופיעו פרטי הסניף בפירוט ומתחת כל פרטי המכוניות הפנויות בתחתית ה- fragment כלומר ה- fragment יחולק לשלושה חלקים כדלהלן:
בחלק העליון תופיע רשימת הסניפים
בחלק האמצעי יופיעו הפרטים המלאים של הסניף שנבחר
בחלק התחתון תופיע רשימת המכוניות הפנויות ששייכות לאותו סניף.
בחירה של אחת המכוניות תיצור באופן אוטומטי הזמנה עבור מכונית זאת. בהזמנה יעודכנו באופן אוטומטי מספר הזיהוי של המשתמש, זמן נוכחי, קילומטרז' נוכחי.
- באמצעות האפשרות השלישית ניתן יהיה להציג רשימה של מכוניות פנויות. ניתן יהיה לסנן מכוניות ע"פ סוג הדגם. כל מכונית תוצג עם פרטים בסיסיים בלבד. כשבחרים מכונית מסוימת תופיע מתחתיה פרטיה המלאים של המכונית וכן פרטיו המלאים של הסניף שבו חונה המכונית.
- באמצעות האפשרות הרביעית ניתן יהיה להציג את פרטי המכונית שכרגע נמצאת בשימוש ע"י המשתמש הנוכחי. ניתן יהיה לשחרר את המכונית, חובה בעת שחרור המכונית לציין את מספר הקילומטרז' ובאופן אוטומטי ההזמנה תתעדכן במערכת כולל: מספר הקילומטרים שהמכונית נסעה מאז תחילת ההשכרה, זמן מדויק של ההחזרה, מספר השעות הכולל שהמכונית הייתה בשימוש הסכום שנזקף לחובתו של הלקוח. חישוב סכום החיוב יתחשב במספר הקילומטרים, כולל עלות דלק לכל ק"מ, וכן מספר השעות. במידה והמשתמש מילא דלק בעצמו יקוזז המחיר שהוא שילם.
- האפשרות החמישית היא יציאה מהתוכנית.

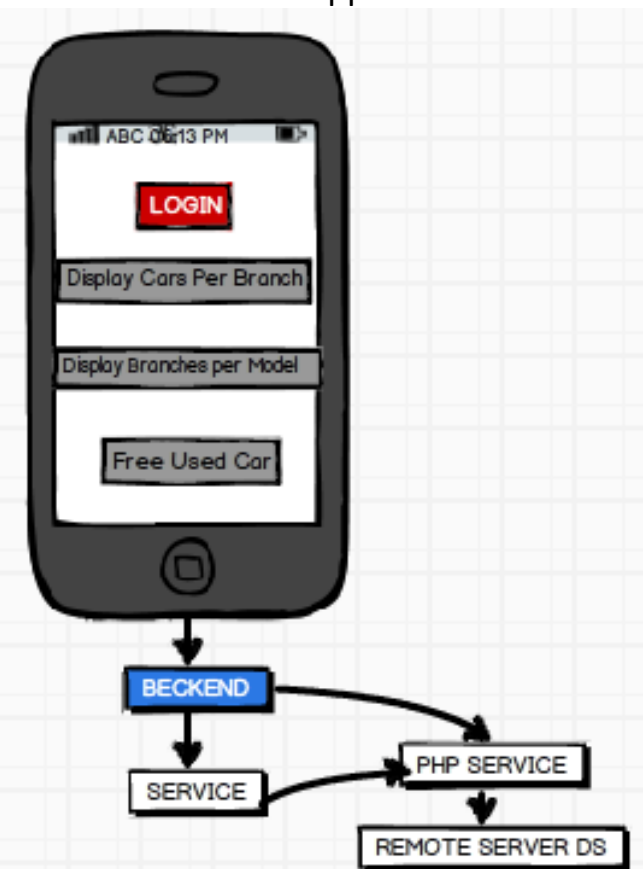
הערה: באופן רציף תתבצע בדיקה האם התפנתה מכונית ואם כן אזי היא תתווסף לרשימת המכוניות הפנויות.

כל הזכויות שמורות למחלקת המחשבים של המרכז האקדמי לב אין להעתיק ולהעביר בדרך כלשהי ללא רשות

First Application



Second Broadcast Receiver Application



כל הזכויות שמורות למחלקת המחשבים של המרכז האקדמי לב אין להעתיק ולהעביר בדרך כלשהי ללא רשות

ישויות

לקוח

- שם משפחה
- שם פרטי
- מספר ת.ז.
- מספר טלפון
- כתובת דוא"ל
- מספר כרטיס אשראי לחיוב.

סניף

- כתובת {עיר, רחוב, מספר}
- מספר מקומות חנייה
- מספר סניף

דגם מכונית

- קוד דגם
- שם חברה
- שם דגם
- נפח מנוע
- תיבת הילוכים {אוטומטי, ידני}
- מקומות ישיבה
- וכו'...

מכונית

- מספר הסניף בו המכונית חונה בדרך קבע.
- סוג דגם
- קילומטרז'
- מספר רכב

הזמנה

(הערה: למרות שלא נעשה שימוש בישות הזאת באפליקציה הראשונה בכל אופן כדאי לממש אותה כבר בשלב זה ע"מ ליצור בהמשך את טבלת ההזמנות. מאוחר יותר נעביר את הישות הזאת לאפליקציה השנייה ונסיר אותה מהאפליקציה הראשונה)

- מספר לקוח
- הזמנה פתוחה\סגורה
- מספר מכונית
- מועד תחילת ההשכרה
- מועד סיום ההשכרה
- ערך תחילת קילומטרז'
- ערך סוף קילומטרז'
- האם התבצע מילוי דלק ?
- אם מילאו דלק אזי יש לציין כמה ליטרים מולאו.
- סכום סופי לחיוב
- מספר הזמנה

כל הזכויות שמורות למחלקת המחשבים של המרכז האקדמי לב אין להעתיק ולהעביר בדרך כלשהי ללא רשות

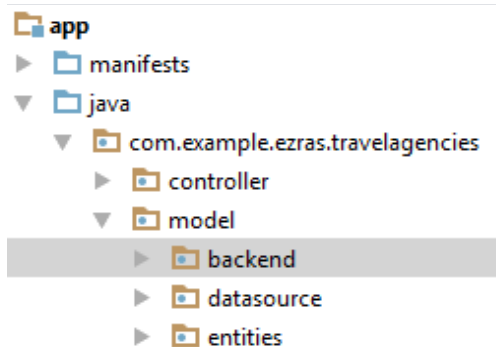
שלבי ההגשה וניקוד עבור כל שלב:

שלב 1

השלב הראשון על כל חלקיו שווה ל 22% מהציון הסופי של הפרויקט בשלב זה נתחיל לבנות את האפליקציה הראשונה שמשמשת את חברת ההשכרה. בחלק א של שלב זה נבנה את מחלקות הישויות השונות, ואת התשתית לשמירת כל אחת מהישויות במקור נתונים שבשלב זה יהיה רשימות ב JAVA. בחלק ב של שלב זה נממש את ה backend כך שישתמש בתשתית שבנינו בחלק א וכמו כן נבדוק את הפונקציונליות המתאימה. בחלק ג של שלב זה נעצב ונבנה את מסכי האפליקציה

חלק ראשון

כתוב מחלקות שמתארות את כל הישויות. עליך להשתמש במגוון משתנים ע"מ שיהיה ניתן להציגם בהמשך במקסימום סגנונות של פקדים. אופן הצגת הנתונים תוך שימוש במגוון של פקדי GUI מהווה פרמטר לבחינת ממד היצירתיות שלכם. כתוב את ה- List Data Source. כתוב את ה- Backend. ממש את ה- Backend לעבודה מול List Data Source. יש לכתוב Factory Method בתוך מחלקה נפרדת בתוך הפקג' Backend. חלוקת מרכיבי האפליקציה ע"פ פקג'ים תיראה כך:



פונקציונליות של ה- Backend שיש לממש:

- בדוק האם משתמש כבר קיים במקור הנתונים.
- הוסף משתמש חדש.
- הוסף דגם
- הוסף מכונית.
- החזר רשימת כל הדגמים הקיימים.
- החזר את כל רשימת המשתמשים.
- החזר את רשימת כל הסניפים.
- החזר את רשימת כל המכוניות.

מדיניות הניקוד עבור החלק הראשון של השלב הראשון: (בשקלול הסופי 5% מהציון הסופי של הפרויקט)

- תיאור הישויות והקשרים ביניהן באמצעות דיאגרמת UML תנוקד ב- 10%.
- כתיבה נכונה של הקוד תנוקד ב- 90%.

כל הזכויות שמורות למחלקת המחשבים של המרכז האקדמי לב אין להעתיק ולהעביר בדרך כלשהי ללא רשות

חלק שני

בחלק זה, יש לכתוב Factory Method שתחזיר מופע של מימוש ה- Backend כ- List DS. כל הפונקציות תשתמש במימוש הנוכחי של Backend. ע"מ לנסות את הפונקציות של Backend יש ליצור Activity אחד שבתוך ה- onCreate שלו יש להפעיל את כל הפונקציות ולוודא שזה עובד. את התשובות אם פונקציה הצליחה או לא יש לשלוח בשלב זה ל- Log.

מדיניות הניקוד עבור החלק השני של השלב הראשון: (בשקלול הסופי 5%)

- כתיבה נכונה של הקוד תנוקד ב- 50%.
- הרצה חלקה 50%.

חלק שלישי

יש לתכנן מראש (רצוי באמצעות כלי ייעודי כלשהו) את כל המסכים. יש לכתוב את כל ה- Activities הרלוונטיים. יש לעשות שימוש במירב סגנונות של פקדים ובשילוב מיטבי ונכון של Layouts. יש לחבר את ממשק המשתמש שיצרנו בחלק הנוכחי עם החלק הפונקציונלי שכתבנו בשלב הנ"ל. יש להקפיד שכל הקריאות לפונקציות ה- Backend מתוך ה- Activities יתבצעו רק מתוך - AsyncTask.

מדיניות הניקוד עבור החלק השלישי של השלב הראשון: (בשקלול הסופי 12%)

- ממשק נוח לשימוש 90%
- יצירתיות: (עד 10%)

שלב 2

השלב השני על כל חלקיו שווה ל 10% מהציון הסופי של הפרויקט בשלב זה נבנה את בסיס הנתונים הטבלאי בשרת שתקבלו מהמכון בחלק א של שלב זה, ניצור את כל הטבלאות בשרת שתקבלו בחלק ב של שלב זה, נבנה דף PHP שיאפשר לנו לתקשר מול בסיס הנתונים שבשרת בחלק ג של שלב זה, נממש את ה backend כך שהנתונים ישמרו ויקראו מהטבלאות בשרת שבנינו בחלקים הקודמים

חלק ראשון

ראשית יש ליצור את בסיס הנתונים בענן. בסיס הנתונים יכיל את הטבלאות הבאות:

- טבלת משתמשים
- טבלת סניפים
- טבלת דגמים
- טבלת מכונות
- טבלת הזמנות

מדיניות הניקוד עבור חלק זה: (בשקלול הסופי 2%)

- כתיבה נכונה של הקוד תנוקד ב- 100%.

חלק שני

לאחר מכן יש ליצור את ה- PHP Service ב- LocalHost ולנסות אותו בצורה עצמאית בתוסף Postman לדפדפן Chrome. באם זה עובד יש לבצע Deploy של ה- PHP Service לענן.

הפונקציונליות הנדרשת:

- בדוק האם משתמש כבר קיים במקור הנתונים.
- הוסף משתמש חדש.
- הוסף דגם
- הוסף מכונת.
- החזר רשימת כל הדגמים הקיימים.
- החזר את כל רשימת המשתמשים.
- החזר את רשימת כל הסניפים.
- החזר את רשימת כל המכונות.
- כל בדיקות האימות יתבצעו בצד השרת (לדוגמא: יש לוודא שתאריך התחלת השכרה מוקדם מתאריך ההחזרה)

מדיניות הניקוד עבור חלק זה: (בשקלול הסופי 3%)

- כתיבה נכונה של הקוד תנוקד ב- 50%.
- הרצה חלקה תנוקד ב- 50%.

חלק שלישי

לאחר מכן יש לשנות את מימוש ה- BackEnd של האפליקציה הראשונה לעבודה מול ה- PHP Service שבענן

מדיניות הניקוד עבור חלק זה: (בשקלול הסופי 5%)

- כתיבה נכונה של הקוד תנוקד ב- 50%.
- הרצה חלקה תנוקד ב- 50%.

שלב 3

השלב השלישי על כל חלקיו שווה ל 63% מהציון הסופי של הפרויקט

חלק ראשון

בחלק זה נבנה את האפליקציה השנייה
יש להעתיק לאפליקציה השנייה את כל הישגיות מהאפליקציה הראשונה למעט ישות: דגם.
יש להסיר מהאפליקציה הראשונה את הישגיות "הזמנה" ולהעביר אותה לאפליקציה השנייה.

פונקציונליות של ה- Backend שיש לממש:

- בדוק האם משתמש כבר קיים במקור הנתונים.
 - הוסף משתמש חדש.
 - עדכן מכונית (קילומטרז').
 - החזר את כל רשימת המשתמשים.
 - החזר את רשימת כל הסניפים.
 - החזר את רשימת כל המכוניות הפנויות.
 - החזר רשימת מכוניות פנויות עבור סניף מסוים.
 - החזר רשימת מכוניות פנויות בטווח מסוים של קילומטרים מהמקום הנוכחי.
 - החזר את רשימת כל הדגמים.
 - עבור כל דגם של מכונית החזר את רשימת הסניפים שמכונית פנויה מדגם זה נמצאת שם.
 - החזר רשימת כל ההזמנות שעדיין לא נסגרו
 - הוסף הזמנה חדשה
 - סגור הזמנה קיימת (על המשתמש יהיה לרשום את הקילומטרים שהוא ביצע. מספר השעות יעודכנו בצורה אוטומטית).
 - בדוק האם נסגרה הזמנה בתוך עשר שניות האחרונות.
- צור DS מסוג List ומלא את כל הרשימות כבר ב- onCreate של ה- main activity הראשון (בתחילה הוא יהיה ללא GUI משמעותי)
- מדיניות הניקוד עבור החלק הראשון של השלב השלישי:** (בשקלול הסופי 10% מהציון הסופי של הפרויקט)
- כתיבה נכונה של הקוד תנוקד ב- 100%.

חלק שני

יש לכתוב את ה- Android Service. תפקידו של ה- Service לבדוק בכל פרק זמן נתון (לדוגמא: כל 10 שניות) האם מכונית שנמצאת בטבלת המכוניות שינתה את הסטטוס שלה מ- "תפוס" ל- "פנוי" (הבדיקה תתבצע על כל ההזמנות שנסגרו בטווח העשר שניות האחרונות) לאחר מכן, ה- Service שולח intent מסוים שייקלט ע"י כל אפליקציה אחרת שתוגדר כ- Broadcast Receiver

אנחנו נגדיר את האפליקציה¹ השנייה כ-Broadcast Receiver. בהתקבל intent תופיע הודעה ב- Log והאפליקציה השנייה תרענן מחדש את רשימת המכוניות הפנויות שיש לה (יש להדגיש, שבגישה שהכתבנו כאן באופן שרירותי, יש הכרח לעבוד עם *Android Service* מפני שאיננו מעוניינים שאורך החיים של הפעולה יהיה תלוי ב- Activity כלשהו ולכן לא נשתמש כאן ב- *AsyncTask*. ה- Service יופעל מתוך הפונקציה הבונה של המחלקה שתממש את ה- *BackEnd* נידרש לפנות ל- *PHP Remote Service*.

ה- Intent יישלח כאשר ברשימת המכוניות תופיע מכונית ששינוי הזמן שמופיע בה (בין "תפוס" ל- "פנוי") הינו לכל היותר טווח הזמן שבו מתבצעת הבדיקה בלולאה. לדוגמא אם אנחנו מאלצים שלולאת הבדיקה תתבצע כל 10 שניות אזי אם תימצא מכונית שהזמן בה הוא לכל היותר 10 שניות מהזמן הנוכחי אז ה- Service יישלח Intent. חשוב לנסות את האפליקציה משני מכשירים שונים. במכשיר אחד נשנה באמצעות האפליקציה הראשונה סטטוס של מכונית ובמכשיר השני, שיפעיל את האפליקציה השנייה, נראה את התגובות שמתקבלות באפליקציה השנייה בעצמה ע"י ה- Broadcast Receiver.

הערה חשובה:

כמובן שבדרך כלל אנו שולחים Broadcast Receiver עבור אפליקציה אחרת ולא עבור אותה אפליקציה אבל, רק לצורך התרגול של service ו Broadcast Receiver אנו נבצע זאת באותה האפליקציה

מדיניות הניקוד עבור החלק השני של השלב השלישי: (בשקלול הסופי 8%)

- כתיבה נכונה של הקוד תנוקד 50%.
- הרצה מוצלחת תנוקד 50%.

¹ כל מקום שכתוב "האפליקציה השנייה" הכוונה לאפליקציה הזו (כלומר השנייה שאנו בונים ולא מלשון האפליקציה האחרת...)

חלק שלישי

יש לכתוב את כל תשתית ה- SharedPreferences ע"מ לשמור כל שם משתמש שניכנס לאפליקציה.

יש לכתוב את ה- GUI של ה- Login\Registering
ה- Login הוא מסך הפתיחה כפי שמוגדר ב- Manifest
כאשר מכניסים שם משתמש וסיסמא ולוחצים על הכפתור LOGIN מתבצעת בדיקה אם שם המשתמש והסיסמא רשומים בתוך ה- SaveSharedPreference אם כן – נכנסים, ואם לא - אז בודקים האם שם המשתמש והסיסמא מופיעים בתוך ה- DS מאחר וייתכן מצב שפשוט מנסים להיכנס מסמארטפון חדש שבו עדיין הפרטים לא נרשמו ב- SaveSharedPreference. אם שם המשתמש לא קיים אז ניתן להיכנס ל- Register מתוך המסך הזה. מכניסים שם משתמש וסיסמא וזה נרשם גם ב- SaveSharedPreference וגם ב- DS ואז ניתן לחזור ולבצע Login.

מדיניות הניקוד עבור החלק השלישי של השלב השלישי: (בשקלול הסופי 5%)
• הרצה מושלמת של תנוקד 100%.

חלק רביעי

- בחלק זה עליכם לכתוב את ה-ToolBar, את ה- Navigation Drawer על כל חמשת האופציות שלו כדלהלן:
 - באמצעות האפשרות הראשונה ניתן יהיה לקבל את פרטי החברה "Take&Go" שהם: מספר טלפון, כתובת דוא"ל, קישור לאתר החברה. ניתן יהיה להתקשר לחברה או לשלוח דוא"ל.
 - באמצעות האפשרות השנייה ניתן יהיה להציג את רשימת הסניפים (סניף=מקום חניה מוסדר עבור חברת "Take&Go"). כל סניף יופיע עם פרטיו הבסיסיים בלבד כולל לינק למפת הגעה לסניף. ניתן לסנן (Filtering) סניפים ע"פ העיר. עם בחירת אחד הסניפים יופיעו פרטי הסניף בפירוט ומתחת כל פרטי המכוניות הפנויות בתחתית ה- fragment (כלומר ה- fragment יחולק לשלושה חלקים כדלהלן: בחלק העליון תופיע רשימת הסניפים, בחלק האמצעי יופיעו הפרטים המלאים של הסניף שנבחר ובחלק התחתון תופיע רשימת המכוניות הפנויות ששייכות לאותו סניף. בחירה של אחת המכוניות תיצור באופן אוטומטי הזמנה עבור מכונית זאת. בהזמנה יעודכנו באופן אוטומטי מספר הזיהוי של המשתמש, זמן נוכחי, קילומטרז' נוכחי.
 - באמצעות האפשרות השלישית ניתן יהיה להציג רשימה של מכוניות פנויות. ניתן יהיה לסנן מכוניות ע"פ סוג הדגם. כל מכונית תוצג עם פרטים בסיסיים בלבד. כשבחרים מכונית מסוימת תופיע מתחתיה פרטיה המלאים של המכונית וכן פרטיו המלאים של הסניף שמארח את המכונית.
 - באמצעות האפשרות הרביעית ניתן יהיה להציג את פרטי המכונית שכרגע נמצאת בשימוש ע"י המשתמש הנוכחי. ניתן יהיה לשחרר את המכונית, חובה בעת שחרור המכונית לציין את מספר הקילומטרז' ובאופן אוטומטי ההזמנה תתעדכן במערכת כולל: מספר הקילומטרים שהמכונית נסעה מאז תחילת ההשכרה, זמן מדויק של ההחזרה, מספר השעות הכולל שהמכונית היתה בשימוש הסכום שנזקף לחובתו של הלקוח. חישוב סכום החיוב יתחשב במספר הקילומטרים, כולל עלות דלק לכל ק"מ, וכן מספר השעות. במידה והמשתמש מילא דלק בעצמו יקוזז המחיר שהוא שילם.
 - האפשרות החמישית היא יציאה מהתוכנית.

הערות:

- בחלק זה עליכם לכתוב את ה- Fragment שנקרא ע"י האופציה הראשונה של ה- Navigation Drawer וכן את ה- Fragment שנקרא ע"י האופציה השנייה של ה- Navigation Drawer, את האופציה השלישית, את האופציה הרביעית. לצורך זה יש לתכנן את ה- resources הרלוונטיים וכמובן לכתוב את ה- Adapters בצורה מיטבית. תכנון המבנה התצוגתי של כל אחד מה- Fragments נתון לשיקול דעתכם ובתנאי שזה יענה על הדרישות המופיעות הנז"ל. מומלץ לעשות שימוש גם ב- ExpandableListView.
- עליך לעשות שימוש במחלקת ה- DataFilter שמכילה לפחות את פונקציונאליות הסינון הבאות:
 - ✓ החזר את רשימת כל המכוניות הפנויות ששייכות לסניף מסוים.
 - ✓ החזר את כל רשימת המכוניות הפנויות מדגם מסוים.

מדיניות הניקוד עבור החלק הרביעי של השלב השלישי: (בשקלול הסופי 40%)

- ממשק GUI מוצלח ינוקד 40%
- כתיבה אופטימלית ונכונה תנוקד 10%
- הרצה מוצלחת של הקוד תנוקד 50%.

שלב 4

השלב הרביעי שווה ל 5% מהציון הסופי של הפרויקט.

בשלב זה חברת ההשכרה רוצה לאפשר ללקוחותיה לשמור באפליקציה דגמי רכב מעודפים וכך יוכלו הלקוחות לקבל מראש הצעות להשכרת רכבים מדגמים מסוימים שבחרו מראש. מתוך מחשבה לטווח הארוך, חברת ההשכרה רוצה לנהל את דגמי הרכב המועדפים ב content provider וכך תוכל לבנות בעתיד עוד אפליקציות שיעשו שימוש באוסף דגמי המכוניות המועדפים של הלקוח (לדוגמה, אפליקציה מכירת רכבים)

בשלב זה עליך להוסיף באפליקציה השנייה רשימה סטטית של דגמים רשימה זו תשמש לשמירת הדגמים המועדפים של הלקוח כל ההכנסה והוצאת הנתונים תתבצע באמצעות content provider

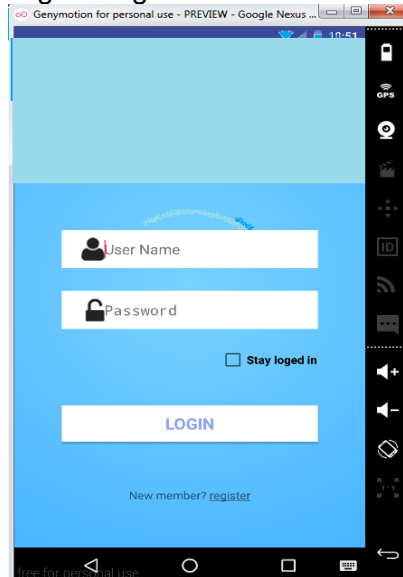
בנוסף, צור אפליקציה נוספת, מאוד פשוטה שתוכל להציג את כל הדגמים המועדפים של הלקוח.

הערה: באופן כללי עדיף לשמור את הדגמים המועדפים ב SQLite

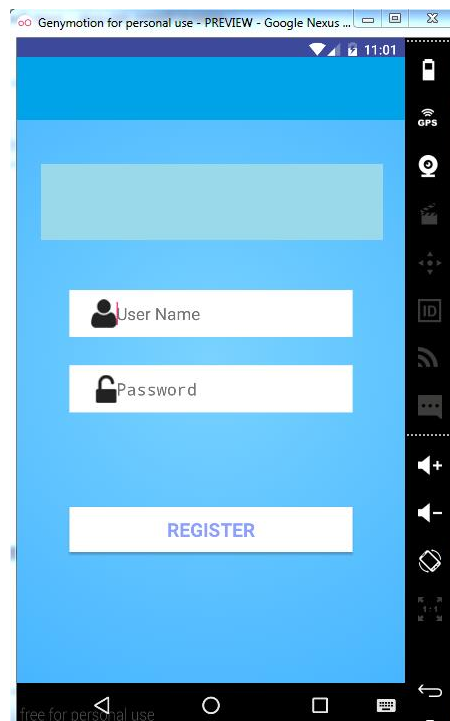
מדיניות הניקוד עבור השלב הרביעי: (בשקלול הסופי 5%)

דוגמאות לצילומי מסך מתוך האפליקציה

מצ"ב דוגמא אפשרית למסכי ה- Login\Registering



הרישום החדש ישמר גם ב- `SaveSharedPreference` וגם ב- `DS`
זו היא רק דוגמא כמובן שבצילום המסך הזה חסרה האפשרות, שחייבים להוסיף אותה, והיא `TextBox` למילוי מספר האשראי של הלקוח.



בהצלחה !!!

כל הזכויות שמורות למחלקת המחשבים של המרכז האקדמי לב אין להעתיק ולהעביר בדרך כלשהי ללא רשות