

# קורס פיתוח אפליקציות לאנדרואיד

## סילבוס מלא

| נושאים   | מודול   |
|--|---|
| <ul style="list-style-type: none"> <li>• Android – הסבר כללי על מערכת ההפעלה מבית גוגל, על הגרסאות והפלטפורמות.</li> <li>• Android Studio – היכרות עם סביבת העבודה הנפוצה ביותר לפיתוח אפליקציות ל Android.</li> <li>• Project files &amp; structure – מבנה האפליקציה על המרכיבים השונים: קבצי Xml לעיצוב סטטי של המסך, קבצי ה Java לכתובת הקוד הדינמי לטיפול באירועים, תיקיית המשאבים ולוקליזציה לפי שפה ולפי גדלי מסכים שונים, קובץ ה Manifest המרכז את הגדרות ורכיבי האפליקציה וקבצי ה Gradle המחברים את חלקי הפרויקט יחדיו.</li> <li>• Synching device – הרצת האפליקציות על המכשיר דורשת הכנה קצרה למען סביבת הפיתוח וכולל להתחבר אליו, ולהריץ אפליקציות בצורה מקומית ולא דרך אמולטורים(מכשירים וירטואלים).</li> </ul>   | <p>Introduction</p> <p>הקדמה</p>  |
| <ul style="list-style-type: none"> <li>• הגדרת רכיבי המסך בקובץ ה XML ובאופן דינמי בקובץ ה Java.</li> <li>• קבלת האירועים מן הרכיבים היוזואלים ב Java על ידי מנגנון ה Listeners וישומו בשיטות שונות.</li> <li>• ארגון האלמנטים היוזואלים בתוך ההורה שלהם לפי המאפיינים הספציפים של כל Layout.</li> <li>• שינוי הקונפיגורציות של כל רכיב באופן דינמי - תוך כדי ריצת התכנית.</li> <li>• בין המחלקות הנלמדות במודול זה: View, ViewGroup, TextView, EditText, ImageView, Button, ImageButton, Switch, ToggleButton, RadioButton, LinearLayout, RelativeLayout, TableLayout, FrameLayout.</li> <li>• State List – הגדרה מראש של תמונות או צבעים למצבים שונים של רכיבי ה UI.</li> <li>• Shapes – קביעת תצורה מותאמת אישית הניתנת להחלה על כל אלמנט וויזואלי.</li> <li>• Toast notifications – שימוש במחלקת ה Toast למען הקפצת הודעות חיווי קצרות למשתמש.</li> </ul>                    | <p>User Interface</p> <p>עיצוב ממשק המשתמש</p>  |
| <ul style="list-style-type: none"> <li>• הוספת רכיב נוסף המהווה מסך מלא (Activity) ומעבר ביניהם על ידי Explicit Intent כולל הטמנת מידע ב Bundle המועבר בו.</li> <li>• פתיחת יישומים מובנים של מערכת ההפעלה על ידי הגדרת הפעולה הרצויה ויצירת Implicit Intent, בינהם החייגן, אנשי הקשר, המפות, דפדפן, יישום המייל ואף יישומים חיצונים כגון WhatsApp© ו Waze©.</li> <li>• מנגנון ההרשאות הישנות (לפני אנדרואיד Marshmallow) והחדשות הניתנות בזמן ריצה (Runtime Permissions) ולא בזמן ההתקנה.</li> <li>• הגדרת רכיב העונה לפעולה או לצורך באמצעות Intent Filter כגון היכולת שלנו להוות אפליקציה לשליחת דואר אלקטרוני.</li> <li>• פתיחת רכיב לצורך קבלת תשובה באמצעות startActivityForResult בין היתר נבצע שימוש במצלמה והצגת התמונה, פתיחת Google Voice recognition ושימוש ב Text to speech. יוצג גם פתיחת רכיב אישי כדוגמת מבחן אמריקאי החוזר אלינו עם הציון של התלמיד.</li> </ul> | <p>Activities &amp; Intents</p> <p>הוספת רכיבים נוספים לאפליקציה ופתיחת יישומי המערכת המובנים</p> |
| <ul style="list-style-type: none"> <li>• מחזור החיים של ה Activity והצורך לשמור מידע במערכת הקבצים.</li> <li>• שמירת נתונים בסיסיים ב Shared preferences ואיחזורם.</li> <li>• שמירה ב Internal Storage של האפליקציה וקריאה ממנו. שמירת מידע מסוגים שונים: טקסט, תמונות וכל אובייקט המיישם את הממשק Serializable. כולל ביצוע החרגה על ידי transient וטיפול אישי על ידי writeObject ו readObject.</li> <li>• שמירת נתונים שיתופיים וארוכי טווח ב External Storage.</li> </ul>  | <p>Persistent Storage</p> <p>שמירת נתונים במערכת הקבצים</p>                                       |

|  |   |
|--|---|
| <ul style="list-style-type: none"> <li>• Activity's Action menu – יצירת תפריט ייחודי לכל מסך וקבלת האירועים ממנו.</li> <li>• Widget Context Menu – הוספת תפריטים קופצים לרכיבים ויזואליים העולים בלחיצה ארוכה.</li> <li>• PopupMenu – תפריט קופץ הניתן להצמדה לכל רכיב ומציג אפשרויות נוספות לפעולה.</li> <li>• Contextual Action Mode – יצירת Context Menu אשר יופיע בסרגל המשימות של המסך.</li> <li>• Alert Dialog – חלון קופץ אשר דורש את תשומת ליבו המיידית של המשתמש (כגון ההתראה על סוללה נמוכה) ברמה בסיסית ומתקדמת על ידי הוספת Custom view.</li> <li>• Multi\Single choices dialogs – הוספת אפשרות לבחור אפשרויות מתוך רשימה.</li> <li>• Date\Time pickers Dialogs – פתיחת חלון סטנדרטי של המערכת לבחירת תאריך ושעה.</li> <li>• Progress Dialog – דימוי תהליך רקע ארוך טווח ועדכון ההתקדמות בחלון מתאים.</li> </ul> | <p>Menus &amp; Dialogs</p> <p>תפריטים ודיאלוגים</p>   |
| <ul style="list-style-type: none"> <li>• ListView, GridView, Gallery – בקרים וויזואליים מתקדמים המאפשרים גלילה מהירה על המידע באמצעות מנגנון מיחזור תאים.</li> <li>• ArrayAdapter, SimpleAdapter, CustomAdapter – הבקרים הוויזואליים המצוינים לעיל שואבים את המידע מה Adapter אשר מחזיק בתוכו את המודלים (מבני הנתונים) ומספק להם את המידע על פי הדרישה.</li> <li>• RecyclerView – בקר וויזואלי מתקדם אשר מיועד להוות תחליף ל AdapterViews בכלל ול ListView בפרט, המנהל את מנגנון המיחזור באופן אוטומטי והנעזר ב ViewHolder לצורך שיפור יעילות וביצועים ומאפשר גמישות מירבית.</li> <li>• CardView – רכיב UI המאפשר הצגת מידע בתוך קלפים עם פינות עגולות והרמה קלה.</li> </ul>  | <p>Adapter, &amp; AdapterViews<br/>RecyclerView</p> <p>רשימות נגלות וספקי המידע עבורן למען הצגת מידע הגדול מהחלון</p> |
| <ul style="list-style-type: none"> <li>• Contact App - יצירת אפליקציית אנשי קשר. שמירה של אובייקטים מורכבים, הצגתם ברשימה, וממשק משתמש דינמי.</li> </ul>   | <p>פרוייקט אמצע עבודה עצמית</p>   |
| <ul style="list-style-type: none"> <li>• Java multithreading – שימוש במנגנון הירושה מ Thread על מנת ליצור תהליך רקע נוסף לתהליך הראשי.</li> <li>• Handler – שליחת הודעות וקטעי קוד להרצה ב Main thread למען עדכוני UI מתוך background threads</li> <li>• AsyncTask – מחלקה יעודית של Android אשר מאפשרת יצירת משימות רקע המעדכנות את ה UI ב Main thread באופן רציף.</li> </ul>   | <p>Asynchronous programming</p> <p>ביצוע מקבילי של משימות</p>   |
| <ul style="list-style-type: none"> <li>• Java Timer class – מחלקה של Java אשר בשילוב עם Handler יכולה לתזמן משימות לביצוע חוזר או מאוחר יותר בתוך האפליקציה.</li> <li>• Android Alarm Manager – שליחת משימות למערכת ההפעלה לביצוע מאוחר יותר או חוזר גם כאשר המשתמש לא נמצא בתוך האפליקציה.</li> <li>• JobScheduler – אפשרות לקביעת משימות עתידיות עם אופטימיזציה של משאבי המערכת.</li> </ul>  | <p>Alarm Manager &amp; Timer</p> <p>תזמון משימות</p>  |
| <ul style="list-style-type: none"> <li>• Notifications &amp; Notifications manager – שליחת התראות למערכת ההפעלה להצגה על מסך הבית של הטלפון. כולל Custom actions &amp; views.</li> <li>• Pending Intent – מתן הרשאות ליישומים חיצוניים לביצוע Intent פנימי.</li> <li>• Progress notifications – הצגת Progress bar בנוטיפיקציה ועדכונה באופן רציף.</li> <li>• Auto Reply Notifications – יצירת נוטיפיקציות של הודעות עם אפשרות כתיבת תשובה.</li> </ul>  | <p>Notifications</p> <p>מערכת ההתראות</p>   |
| <ul style="list-style-type: none"> <li>• Service – רכיב נוסף המאפשר הרצת משימה ארוכת טווח ללא User interface.</li> <li>• IntentService – מחלקה היורשת מ Service ומריצה את המשימות אחת אחרי השניה ברקע.</li> <li>• Foreground Service – מעבר של Service ל Foreground באמצעות נוטיפיקציה המעידה על קיומו נותנת לו טווח חיים ארוך אף יותר ואינו מועמד לסיום על ידי המערכת במצב זכרון נמוך.</li> <li>• MediaPlayer – מחלקה לניגון מדיה, באמצעותה ובשילוב עם ה Service נייצר נגן מוזיקה.</li> </ul>   | <p>Services</p> <p>מבצעי שרות</p>   |

|   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• BroadcastReceiver – רכיב נוסף המאפשר קבלת שדרים העוברים במערכת, כגון שיחה נכנסת, הודעת טקסט, סוללה נמוכה ועוד.</li> <li>• System and Custom broadcast – הרשמה באמצעות IntentFilter לשדרי מערכת שונים או לשדר שהגדרנו בעצמנו. הרשמה בקובץ ה Manifest או באופן דינמי בקוד ה Java על יד רכיב אחר.</li> <li>• בין השדרים המודגמים: Received sms, boot completed, screen on/off</li> <li>• SmsManager – מחלקה לטיפול וקריאה של הודעות טקסט.</li> <li>• LocalBroadcastManager – שליחת שדרים פנימיים ומאובטחים בתוך האפליקציה שלנו.</li> </ul>  | <p>Broadcast Receiver</p> <p>מעבירי ההודעות</p> |
| <ul style="list-style-type: none"> <li>• Fragments &amp; Fragment manager – פיצול המסך הגדול לתתי חלקים ומתן עצמאות לכל חלק.</li> <li>• Activity &amp; Fragment communication – יצירת פרטוקול תקשורת מוסכם.</li> <li>• DialogFragment, ListFragment – תתי מחלקות של Fragment להצגת רשימה ודיאלוג.</li> <li>• PreferenceFragment – יצירת תפריט הגדרות לאפליקציה באמצעות רכיבים וויזואלים יעודיים.</li> <li>• ViewPager – רכיב וויזואלי מתקדם המאפשר החלפת Fragments על ידי החלקה אופקית.</li> </ul>  | <p>Fragments</p> <p>חלקות המסך לתתי יחידות</p>  |
| <ul style="list-style-type: none"> <li>• Location &amp; LocationManager – שימוש בשרותי המיקום של המכשיר באמצעות שימוש ב GPS, Network או מיקום פאסיבי. אובייקט Criteria אשר מאפשר להגדיר קריטריונים ולקבל את הספק המתאים ביותר.</li> <li>• Fused Location API – שימוש בשרותי המיקום החדשים של Google play כדי לקבל עדכוני מיקום אמינים בהם המערכת בוחרת את הספק המיטבי לפי הצרכים שלנו (קצב עדכון, רמת דיוק וצריכת סוללה). שימוש בספרייה זו יחליף בעתיד לחלוטין את ה LocationManager.</li> <li>• Geocoder – מחלקה המאפשרת לנו לשלוח את המיקום המתקבל מאובייקט ה Location אל שרתי גוגל וקבלת אובייקט Address המייצג את הכתובת המדוייקת של נקודת הציון שהתקבלה.</li> <li>• Sensor &amp; SensorManager – כל יתר החיישנים ב Android עובדים לפי מודל דומה. חיישן ה Accelerometer כדוגמה מייצגת לכל יתר החיישנים במכשיר.</li> <li>• Touch – אירוע ה Touch מועבר לאפליקציה על ידי מערכת ההפעלה ואף בה עובר היררכיה פנימית לטיפול בו. נלמד כיצד לקבל ולנתח את ה TouchEvent וכיצד ניתן למצות אותו או להעבירו הלאה.</li> </ul>                                     | <p>Sensors</p> <p>חיישנים</p>                   |
| <ul style="list-style-type: none"> <li>• Overriding existing widget - ניתן לרשת מכל אחד מהאלמנטים הוויזואלים הקיימים ולאפיין אותם לפי דרישה ספציפית.</li> <li>• Creating new Widget – על ידי ירושה מהמחלקה View נוכל לייצר רכיב וויזואלי חדש, תודגם יצירת בלוק ציור כרכיב וויזואלי עצמאי אותו ניתן לשלב בכל אפליקציה עתידית.</li> <li>• Handling orientation changes – טיפולים אפשריים בשינויי אוריינטציה. ניתן להגדיר Layout יעודי לכל אוריינטציה, ולבקש מהמערכת לטפל בזאת עצמאית כחלק משינויי הקונפיגורציה.</li> <li>• SavedInstanceState – אובייקט Bundle המכיל את מצב העצם טרם הריגתו על ידי המערכת.</li> <li>• Drawable Animation – אנימציה כדוגמת GIF אשר מאפשרת מעבר בין תמונות.</li> <li>• Tween Animation – אנימציות הניתנות להפעלה על כל רכיב וויזואלי: Rotate, Alpha, Scale, Translate (סיבוב, שקיפות, גודל והזזה).</li> <li>• LayoutTransition - אפשרות להגדרת אנימציות שיחולו על כל ה Views בתוך ה ViewGroup</li> <li>• ObjectAnimator - ניתן להגדיר את האנימציות שצויינו לעיל באופן דינמי (בג'אווה) מיידית על תכונות האובייקט.</li> </ul> | <p>Advance UI</p> <p>ממשק משתמש מתקדם</p>       |
| <ul style="list-style-type: none"> <li>• NavigationView &amp; DrawerLayout – שילוב של שני המרכיבים הללו יחדיו יאפשר הצגת תפריט המחליק מן הצד לבחירת אפשרויות.</li> <li>• Floating Action Button – כפתור הצף מעל האפליקציה ומייצג פעולה מרכזית בה.</li> <li>• Snackbar – תחליף להודעות ה Toast המציג הודעה באנימציה בתחתית המסך ופעולה יעודית.</li> </ul>  | <p>Material Design with support library</p>     |

|   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• CoordinatorLayout – תבנית מתקדמת אשר מטרתה לתאם בין מספר אלמנטים במסך.</li> <li>• Floating Labels for Edit Text – מעטפת לתיבות הטקסט הרגילות למען השארת הרמז למשתמש.</li> <li>• TabLayout – שילוב שלו עם ViewPager יציג ה Fragments המוחלפים בתפריט Tab.</li> <li>• AppBarLayout – תבנית אשר מטרתה לתאם בין רשימות נגללות לתפריטים בראש המסך ולהעלימם בעת גלילתו.</li> <li>• CollapsingToolbarLayout – אפשרות למוטט ולהרחיב את התפריט העליון בעת הגלילה.</li> </ul>  | <p>תבנית עיצוב חדשנית מבית גוגל</p>   |
| <ul style="list-style-type: none"> <li>• Http protocol – פרוטוקול התקשורת של רשת האינטרנט בין שרת ללקוח.</li> <li>• Get\Post requests – בקשות מהלקוח (האפליקציה) לשרת (אינטרנט).</li> <li>• Receiving XML &amp; JSON – תשובת השרת ב XML ו JSON. ניתוח התשובות באמצעות המימוש של אנדרואיד ל Xml DOM וה JSON Parser.</li> <li>• Picasso – שימוש בספרייה חיצונית להורדה וטעינה של תמונות.</li> <li>• Volley – שימוש בספרייה המנהלת תור משימות אסינכרוני לבקשות HTTP.</li> </ul>  | <p>Communication</p> <p>תקשורת לקוח-שרת</p>                                     |
| <ul style="list-style-type: none"> <li>• SQLite DB – מסד נתונים לוקלי לשמירה מרוכזת של כמות גדולה של נתונים חוזרים.</li> <li>• Content Provider – רכיב שמטרתו לספק מידע לאפליקציות אחרות.</li> <li>• Address Book API – ספר הטלפונים והכתובות כדוגמה לספק מידע כללי.</li> <li>• SimpleCursorAdapter – Adapter המייצר מידע מבסיסי נתונים ליחידות הוויזואליות.</li> <li>• CursorLoader – טעינה וניהול אסינכרוני של מסדי הנתונים למטרת שיפור ביצועים.</li> </ul>   | <p>Database &amp; Content Providers</p> <p>מסדי נתונים</p>                      |
| <ul style="list-style-type: none"> <li>• Firebase - שרות צד שרת דינמי מבית גוגל המאפשר ריכוז והעברת מידע בין הלקוחות.</li> <li>• Messaging service – שליחת הודעות (Push notifications) לכל המשתמשים או למשתמש ספציפי.</li> <li>• User Authentications – ניהול משתמשים ויצירת פרופיל אישי באפליקציה (הרשמה באמצעות חשבון גוגל, פייסבוק או מייל), הגדרת ה Session והשימוש בה.</li> <li>• Remote Database – שמירת מידע אישי לכל משתמש בצד השרת. גישה לנתונים מפלטפורמות שונות כגון אפליקציית iOS, android, Web וממכשירים שונים. ניתן לרכז ולהעביר מידע בין לקוחות שונים כדוגמת בלוג או רשת חברתית.</li> <li>• Topics messaging – שליחת הודעות למשתמשים לפי הרשמה לנושאים ספציפים השליחה מתאפשרת גם דרך ממשק הניהול וגם באמצעות בקשות POST מתוך הלקוח עצמו.</li> <li>• Firebase UI &amp; Firebase Adapter – רכיבים מוכנים מאת Firebase אשר מקלים באופן משמעותי את העבודה ה"סטנדרטית".</li> <li>• <b>תרגיל מסכם צד שרת</b> – ניהול מערכת תזכורות מבוססת ענן. כל משתמש יכול לייצר חשבון ולהעלות לענן תזכורת ומשימות לביצוע. ניתן יהיה לגשת ליומן המשימות גם מפלטפורמות נוספות. כלומר המשימות ירוכזו בשרת לפי חשבון המשתמש.</li> <li>• <b>תרגיל מסכם הודעות</b> - יצירת מספר קבוצות של נושאים ושליחת הודעות מתוך האפליקציה באמצעות מערכת ניתוב ההודעות של Firebase לכל הרשומים לאותו הנושא (עם "מעט" שכלול זהו הבסיס גם לרשתות חברתיות).</li> <li>• <b>תרגיל chat</b> – תקשורת בזמן אמת בין משתמשים שונים של האפליקציה.</li> </ul> | <p>צד שרת –</p> <p>Firebase בניית מאגר משתמשים, הודעות ומסדי נתונים מרוחקים</p> |
| <ul style="list-style-type: none"> <li>• Android JetPack – חבילת התוספות האחרונה לאפליקציית האנדרואיד.</li> <li>• AndroidX – ספריות התמיכה האחרונות להוספת רכיבים חדשים עם תמיכה אחורנית.</li> <li>• View Model – הפרדה בין ה UI ל Data באמצעות מחלקה ייעודית השומרת את המידע במנותק מהממשק ומאפשרת לו לשרוד את שינויי הקונפיגורציה.</li> <li>• LiveData – עדכון המידע באופן אוטומטי באמצעות תבנית ה Observer.</li> <li>• Room Database – בניית מסדי נתונים של SQLite באמצעות מעטפת חובקת SQL.</li> </ul>   | <p>ארכיטקטורה App Architecture</p>  |

|   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Retrofit 2 – שימוש בספריה נוספת לקריאת מידע מהרשת הממירה באופן אינטגרטיבי לאובייקטים.</li> <li>• תרגיל מסכם ארכיטקטורה – תרגיל המרכז את כל הנושאים הנלמדים במודל ליצירת אפליקציה השואבת מידע משרת אינטרנט ושומרת אותו גם במסד נתונים מקומי כאשר היא עומדת בתנאי הארכיטקטורה המודרניים והנפוצים בתעשייה.</li> </ul>                 |  |
| <ul style="list-style-type: none"> <li>• ניתן לייצר אייקונים המתרחבים לתפריטים שלמים (בעת לחיצה עליהם) אשר יכולים לצוף על פני מסך הבית ועל פני כל האפליקציות המוצגות, ללא תלות בהיכן המשתמש נמצא.</li> <li>• מתאים מאוד למגני מוזיקה הרוצים לאפשר למשתמש שליטה לא משנה באיזה יישום אחר הוא נמצא – ניתן לראות דוגמה לכך בראשי השיחה הצפים של ה Facebook Messenger</li> </ul> | <p>Floating Widgets<br/>איקונים מרחפים</p> |
| <ul style="list-style-type: none"> <li>• החל מאנדרואיד 7.1 ניתן להוסיף שורה של קיצורי דרך לאפליקציה המופעלים בעת לחיצה ארוכה על האייקון. נלמד להוסיפם בצורה סטטית (מוכנים מראש) ודינמית (תוך כדי ריצה).</li> </ul>  | <p>App Shortcuts<br/>קיצורי דרך</p>        |