١٢

بسم الله الرحمن الرحيم

دولة فلسطين
وزارة التربية والتعليم

تطبيقات الهَواتِفُ الذَّكِيَّة

(عملي ونظري)

المسار المهني - الفرع الصّناعي

رزمة تعليمية - غزة

فريق التّأليف:

أ. مؤمن كيلاني "منسقاً"
د. بشار طحاينة
أ. سامية طه
أ. علاء أبو زهرة
أ. محمد أبو الحوف
أ. نسرين دويكات

قررت وزارة التربية والتعليم في دولة فلسطين
تدريس هذا الكتاب في مدارسها بدءًا من العام الدراسي ٢٠١٩/ ٢٠٢٠ م

الإشراف العام

رئيس لجنة المناهج: أ. د. مروان عورتاني
نائب رئيس لجنة المناهج: د. بصري صالح
رئيس مركز المناهج: أ. ثروت زيد

الدائرة الفنية

إشراف فني: أ. كمال فحماوي
تصميم فني: أسحار حروب

تحرير لغوي: أ. وفاء جيوسي
متابعة المحافظات الجنوبية: د. سميّة النخّالة

الطبعة التجريبية
٢٠١٩ م/ ١٤٤٠ هـ

حقوق الطبع محفوظة ©
دولة فلسطين
وزارة التربية والتعليم
مركز المناهج

حي المصيون، شارع المعاهد
ص. ب 719 – رام الله – فلسطين
pcdc.edu.ps | pcdc.mohe@gmail.com

moehe.gov.ps | mohe.pna.ps | mohe.ps
.com/MinistryOfEducationWzartAltrbytWaltlym
هاتف +970-2-2983280 | فاكس +970-2-2983250

تـقـديـم

يتصف الإصلاح التربوي بأنه المدخل العقلاني العلمي النابع من ضرورات الحالة، المستند إلى واقعية النشأة، الأمر الذي انعكس على الرؤية الوطنية المطورة للنظام التعليمي الفلسطيني في محاكاة الخصوصية الفلسطينية والاحتياجات الاجتماعية، والعمل على إرساء قيم تعزز مفهوم المواطنة والمشاركة في بناء دولة القانون، من خلال عقد اجتماعي قائم على الحقوق والواجبات، يتفاعل المواطن معها، ويعي تراكيبها وأدواتها، ويسهم في صياغة برنامج إصلاح يحقق الآمال، ويلامس الأماني، ويرنو لتحقيق الغايات والأهداف.

ولما كانت المناهج أداة التربية في تطوير المشهد التربوي، بوصفها علماً له قواعده ومفاهيمه، فقد جاءت ضمن خطة متكاملة عالجت أركان العملية التعليمية التعلُّميّة بجميع جوانبها، بما يسهم في تجاوز تحديات النوعية بكل اقتدار، والإعداد لجيل قادر على مواجهة متطلبات عصر المعرفة، دون التورط بإشكالية التشتت بين العولمة والبحث عن الأصالة والانتماء، والانتقال إلى المشاركة الفاعلة في عالم يكون العيش فيه أكثر إنسانية وعدالة، وينعم بالرفاهية في وطن نحمله ونعظمه.

ومن منطلق الحرص على تجاوز نمطية تلقّي المعرفة، وصولاً لما يجب أن يكون من إنتاجها، وباستحضار واعٍ لعديد المنطلقات التي تحكم رؤيتنا للطالب الذي نريد، وللبنية المعرفية والفكريّة المتوخّاة، جاء تطوير المناهج الفلسطينية وفق رؤية محكومة بإطار قوامه الوصول إلى مجتمع فلسطيني ممتلك للقيم، والعلم، والثقافة، والتكنولوجيا، وتلبية المتطلبات الكفيلة بجعل تحقيق هذه الرؤية حقيقة واقعة، وهو ما كان له ليكون لولا التناغم بين الأهداف والغايات والمنطلقات والمرجعيات، فقد تآلفت وتكاملت؛ ليكون النتاج تعبيراً عن توليفة تحقّق المطلوب معرفياً وتربوياً وفكرياً.

ثمّة مرجعيات تؤطّر لهذا التطوير، بما يعزّز أخذ جزئية الكتب المقررّة من المنهاج دورها المأمول في التأسيس؛ لتوازن إبداعي خلّاق بين المطلوب معرفياً، وفكرياً، ووطنياً، وفي هذا الإطار جاءت المرجعيات التي تم الاستناد إليها، وفي طليعتها وثيقة الاستقلال والقانون الأساسي الفلسطيني، بالإضافة إلى وثيقة المنهاج الوطني الأول؛ لتوجّه الجهد، وتعكس ذاتها على مجمل المخرجات.

ومع إنجاز هذه المرحلة من الجهد، يغدو إزجاء الشكر للطواقم العاملة جميعها؛ من فرق التأليف والمراجعة، والتدقيق، والإشراف، والتصميم، وللجنة العليا أقل ما يمكن تقديمه، فقد تجاوزنا مرحلة الحديث عن التطوير، ونحن واثقون من تواصل هذه الحالة من العمل.

وزارة التربية والتعليم

مركز المناهج الفلسطينية

أيار / ٢٠١٩

مقدمة

مواكبةً للتطور العلمي والتكنولوجي في عصر المعرفة، أصبحت الحاجة لمهن جديدة ضرورة حتمية، وانطلاقاً من توجهات وزارة التربية والتعليم بتحقيق تعلم وتعليم نوعي وريادي يلبي احتياجات المجتمع الفلسطيني وَفق أولوياته بالتحرير والتنمية، كان لا بد من تطوير مناهج التعليم المهني بِحُلتها الجديدة لتحاكي التوجهات العالمية بنشأة فلسطينية واقعية تعتمد العقلانية العلمية بالدمج بين الجانبين النظري والتطبيقي، مراعياً سوق العمل، وصولاً إلى جيل يمتلك الكفايات والقدرة على اكتشاف المعرفة العالمية، والإسهام في الإنتاج الكوني.

اعتمدت المناهج المهنية الجديدة منهجية الوحدات النمطية القائمة على الكفايات المهنية بمجالاتها الحرفية، والمنهجية، والاجتماعية والشخصية، وارتباط ذلك بسياقات واقعية حياتية تصف مواقف تعليمية تعلمية، توظف حل المشكلات بطريقة علمية وعملية، دون إغفال للجانب النظري المتضمن لأنشطة تعتمد على أداء الطلبة، والتأكيد على استراتيجيات التقويم التربوي الحديثة بما في ذلك التقويم الأصيل، والتحول من التقويم القائم على تحقيق الكفايات إلى تحقيق الجودة والامتياز، من خلال التركيز على شمولية أداء الطلبة، وتعزيز أنماط التفكير النقدي والتأملي.

جاء كتاب تطبيقات الهواتف الذكية للصف الثاني عشر المهني في خمس وحداتٍ نمطية، عالجت موضوعات:

برمجة واجهة التطبيق وعناصرها، ومهارات إضافية في بَرْمَجةِ تطبيقاتِ الأندرويد، إضافة إلى إنشاء تطبيقات تحتوي على قواعد بيانات باستخدام SQLite، وإعداد تنسيقات لتطبيق الأندرويد، وإخراج ملفِ التطبيقِ وَنَشره.

وإذ نقدّم هذا الكتاب بين أيدي ذوي العلاقة من معلمات ومعلمين وطلبة وتربويين ومهتمين، نرجو تحقيق الغايات التربوية المنشودة، وبعقلية منفتحة نحو التطوير والتعديل، فإنه يسرّنا استقبال أيّ ملاحظة تهدف إلى تطوير الكتاب وتجويده.

فريق التّأليف

المحتويات

الوَحْدةُ الأُولى واجهات التطبيق وعناصرها (Views)

الكفايات المِهْنيّة 4
1-1 قوائم البيانات العاديّة والمخصّصة (list view/grid view) 5
2-1 مستعرض إنترنت (web view) 23
3-1 الوسائط المتعدّدة (multimedia) 29
4-1 مُمَرِّر الصفحات (scroll view) 38

الوَحْدةُ الثّانيةُ (مواضيع خاصّة بالأندرويد)

الكفايات المِهْنيّة 48
1-2 انشاء fragments 49
2-2 مربّع الحوار (dialog) 64
3-2 الإشعارات (notifications) 72
4-2 التعامل مع الأخطاء 81

الوَحْدةُ الثالثةُ قواعد بيانات SQLite في تطبيقات الهاتف المحمول

الكفايات المِهْنيّة 92
3-1 مفهوم قواعد البيانات العلائقيّة وعناصرها المختلفة 93
3-2 بناء قواعد البيانات ومعالجتها باستخدام نظام SQLite 109

الوَحْدةُ الرابعة تنسيق التطبيق

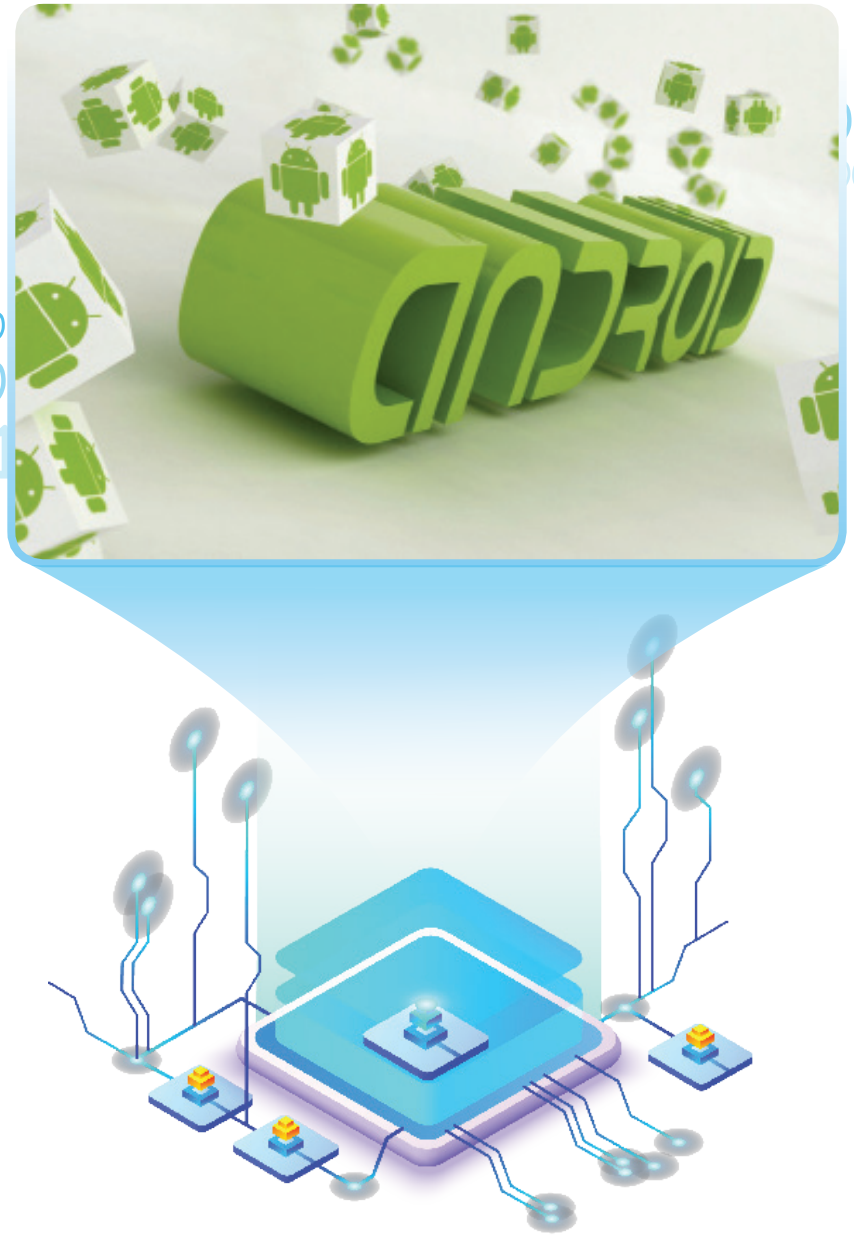
الكفايات المِهْنيّة 128
1-4 تعديل أيقونة التطبيق 129
2-4 إعداد التطبيق بأكثر من لغة 140
3-4 تحسين نمط التطبيق (style) 153
4-4 إعداد التطبيق للعمل على أجهزة الأندرويد المختلفة 161

الوَحْدةُ الخامسة إخراج التطبيق إلى النّور

الكفايات المِهْنيّة 172
1-5 تجهيز التطبيق للنشر على المتاجر الإلكترونيّة 173
2-5 نشر التطبيق 183
3-5 إضافة دعايات AdMob للتطبيق 202

الوَحْدةُ الأولى

واجهات التطبيق وعناصرها (Views)



أبحثُ مع زملائي عن عناصر الواجهات (views) واستخداماتها.

يُتوقَّعُ من الطلبة بعد دراسة هذه الوَحدة، والتفاعل مع أنشطتها أن يكونوا قادرين على برمجة واجهات التطبيق وعناصرها، من خلال تحقيق الآتي:

1. إنشاء قوائم البيانات العاديّة والمخصّصة (list view/grid view).
2. إنشاء مستعرض إنترنت (web view).
3. إنشاء الوسائط المتعدّدة (multimedia).
4. إنشاء مُمَرِّر الصفحات (scroll view).

الكفايات المِهْنيّة

الكفايات المهنيّة المُتوقّع من الطلبة امتلاكها بعد الانتهاء من دراسة هذه الوحدة، والتفاعل مع أنشطتها:

أوّلاً: الكفايات الحرفيّة.

- عمل قائمة list view بأكثر من طريقة.
- عمل قائمة grid view والتعامل معها بأكثر من طريقة.
- عمل مستعرض إنترنت web view.
- عمل عارض فيديو video view.
- عمل مشغّل الصوت.
- عمل مُمَرِّر الصفحات (scroll view).
- تقويم النتائج.

ثانياً: الكفايات الاجتماعيّة والشخصيّة.

- العمل ضمن فريق.
- تقبُّل التغذية الراجعة.
- احترام رأي الآخرين.
- القدرة على إدارة الحوار وتنظيم النقاش.
- مصداقيّة التعامل مع الزبون.
- حفظ خصوصيّة الزبون.
- الاستعداد للاستجابة لطلبات الزبون.
- التحلّي بأخلاقيّات المهنة أثناء العمل.
- التمتُّع بالفكر الريادي أثناء العمل.
- استخلاص النتائج ودقّة الملاحظة.
- الاتصال والتواصل الفعّال.
- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- الإفادة من مقترحات الآخرين.
- امتلاك مهارة التأمُّل الذاتيّ.
- الاستعداد التام لتقبُّل ملحوظات الزبون وانتقاداته.
- القدرة على الإقناع.

ثالثاً: الكفايات المنهجيّة.

- توظيف التعلُّم التعاونيّ.
- امتلاك مهارة البحث العلميّ، والقدرة على توظيف أساليبه.
- القدرة على التخطيط.
- القدرة على استمطار الأفكار.
- استخدام المصادر والمراجع المختلفة.
- توثيق نتائج العمل وعرضها.
- تقويم النتائج وعرضها.
- توظيف التكنولوجيا والبرامج الإحصائيّة وتمثيل البيانات.
- جمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:

- الالتزام بالجلسة الصحيّة أمام الحاسوب.
- إراحة العينين كلّ ربع ساعة.
- الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة.
- الابتعاد عن الحاسوب كلّ ساعة، والقيام ببعض التمارين الرياضيّة.

1-1 الموقف التعليميّ التعلُّميّ

قوائم البيانات العاديّة والمخصّصة (list view/grid view).

وصف الموقف التعليميّ: زار معلّم جغرافيا مشغل تطبيقات الهاتف المحمول في مدرسة القدس المِهْنيّة؛ لمساعدته في إنشاء تطبيق يعرض أسماء المحافظات على شكل قائمة، وعرض معلومات عنها.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من المعلّم عن: - إصدار نظام الأندرويد الخاصّ به. - أسماء المحافظات ومعلومات عنها. • أجمع البيانات عن: - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - نوع القائمة المستخدمة وطريقة برمجتها. - الجمل البرمجيّة المستخدمة.	• حوار ومناقشة. • التعلّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب المعلّم، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد، خاص بواجهات التطبيق وعناصرها وإنشاء قوائم البيانات). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّر	• أُصنّف البيانات (نظام الأندرويد المستخدم). • أُحدّد: - أسماء المحافظات ومعلومات عن كلّ محافظة. - العناصر(views) المستخدمة. - عدد الواجهات. - نوع القائمة وطريقة برمجتها. - الجمل البرمجيّة. • إعداد خطّة للتنفيذ.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة .	• وثائق (نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).

أُنفّذ	• أختار القائمة المناسبة (list view/grid view). • اختيار عدد من الواجهات المقترحة القابلة للاستخدام في تطبيق الأندرويد. • أتّفق مع (الزبون) معلّم الجغرافيا على اختيار واجهات التطبيق المناسبة. • أُصمّم وأُبرمج واجهات التطبيق. • أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة. • عصف ذهنيّ / استمطار الأفكار.	• حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.
أتحقّق	• أتحقّق من ملاءمة اختيار الواجهات المقترحة للتطبيق. • أتاكّد من ملاءمة التطبيق. • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل مناسب.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• الخطّة المعدّة سابقاً. الطلب الخاصّ بالمعلّم. • حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام الأندرويد.
أوثّق وأقدّم	• أوثّق البيانات التي تتعلّق بـ: - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - نوع القائمة المستخدمة وطريقة برمجتها. - الجمل البرمجيّة المستخدمة. • إعداد عرض تقديميّ. • إعداد ملف بالحالة (تطبيق يعرض أسماء المحافظات على شكل قائمة وعرض معلومات عنها).	• حوار ومناقشة. • تعلُّم تعاونيّ/مجموعات عمل.	• حاسوب. • شاشة عرض. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.
أقوّم	• رضا المعلّم عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات	• حوار ومناقشة.	• طلب المعلّم. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد.

سؤال ما الفرق بين list view وgrid view؟

أتعلّم: قوائم البيانات العاديّة والمخصّصة (list-grid view)

نشاط (1) للاطلاع: أبحث عن تطبيقات في متجر جوجل تستخدم list view.

قوائم البيانات:

هي قوائم تُستخدَم لعرض بيانات منوّعة، إمّا أن تكون نصّاً أو صورة، أو نصّاً مع صورة وغيرها، ولها شكلان رئيسيّان، هما:

1. list view.
2. Grid view.

List view:

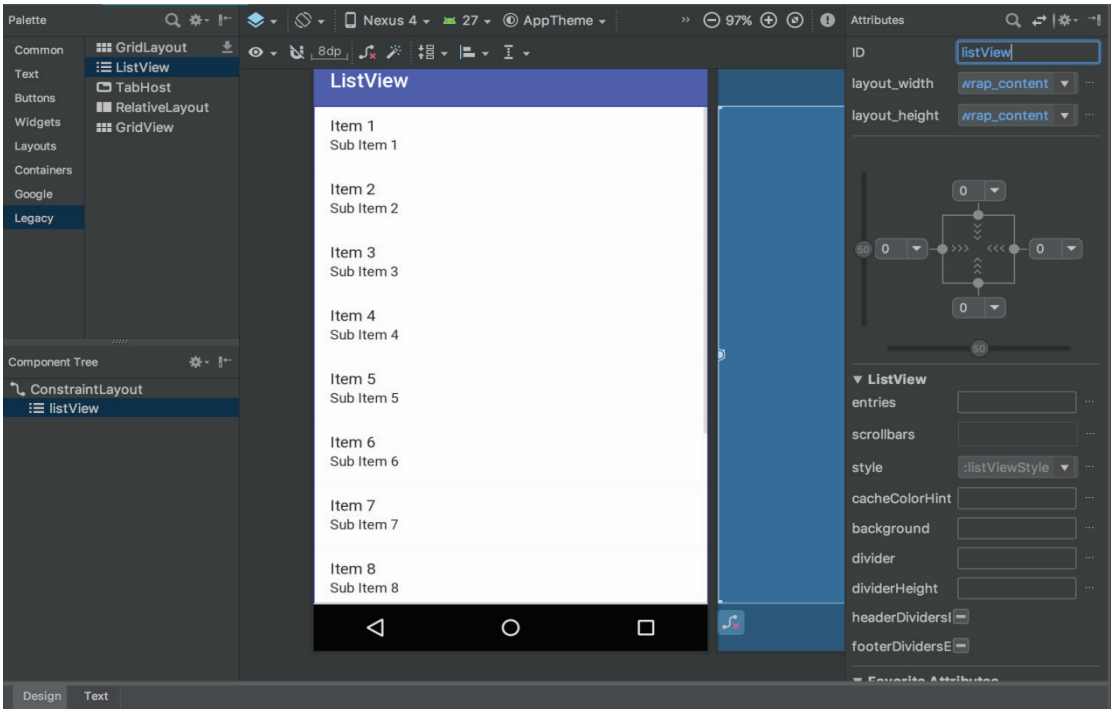
عرض عناصر البيانات في عمود واحد مرتّبة أُفقيّاً، تستخدم العناصر لعرض عدد غير محدود من البيانات بشكل منظّم، وقد تكون البيانات نصوصاً، أو صوراً، أو فيديو، ...

عدد البيانات المعروضه يكون وَفْقَ حجم الشاشة المستخدمة، ولا تستهلك من الذاكرة (Ram) إلّا وفق عدد العناصر المعروضة على الشاشة، مثلاً: لو كان لدي ١٠٠٠ جهة اتّصال مخزّنة على الهاتف، وقمت بفتح تطبيق جهات الاتّصال، ما يتمّ عرضه هو ٦ أسماءن أو ٧ وفق حجم الشاشة؛ أي تستهلك من الذاكره (Ram) بقدر ما ظهر على الشاشة فقط، وباقي الأسماء لا تستهلك من الذاكره (Ram) ما دامت لم تظهر للمستخدم، وفي حال تغيّرت قائمة الأسماء تتمّ إزالة الأسماء القديمة من الذاكرة (Ram) ووضع الأسماء الجديدة.

- **أنواع list view:**
  1. عاديّة: عرض البيانات بشكل العناصر الافتراضيّ.
  2. مخصّصة: عرض البيانات بشكل مخصّص للعناصر.

- **إنشاء list view عاديّة:**

أُنشئ مشروع أندرويد جديداً أُسمّيه listView، بعد فتح البرنامج بشكل كامل أفتح ملف activity_main.xml، وأحذف العناصر التي بداخله، ثمّ أُضيف listView وأعطيها id، كما يأتي:



قبل أن أنتقل إلى الكود البرمجيّ، لا بدّ من توضيح كيفيّة عمل القائمة، كما في الشكل (3 - 1):

الشكل (3 - 1)

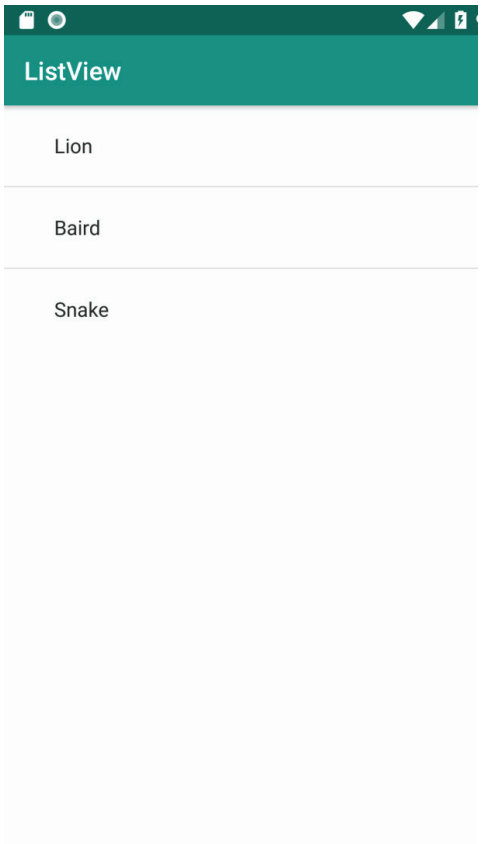
أنتقل إلى الكود البرمجيّ، وأكتب الكود الآتي:

```
val name = arrayOf("Lion","Baird","Snake")

val a:ArrayAdapter<String> = ArrayAdapter( context: this,android.R.layout.simple_expandable_list_item_1,name)


listView.adapter=a
```

أريد إضافة ثلاثة أسماء حيوانات إلى القائمة، كلّ معلومة بداخل عنصر في القائمة، ولكي أتمكّن من فعل ذلك أقوم بما يأتي:

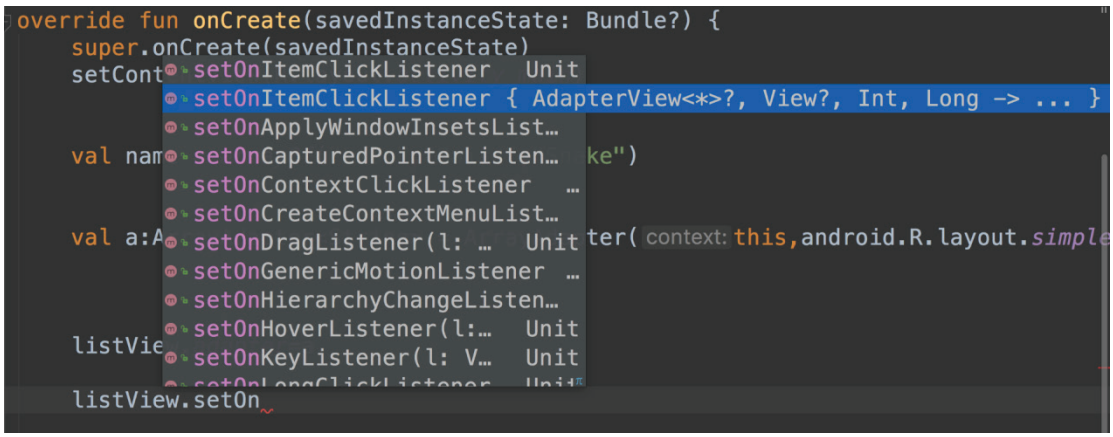
1. أُنشئ مصفوفة وأدخل لها أسماء الحيوانات، كما في السطر الأول في الصورة السابقة.
2. أُنشئ كائناً من الصنف ArrayAdapter<.....>، وبما أنّ البيانات نصوص أضع <String>، ثمّ أُعيد تعريفها بكتابة ArrayAdapter(....,....,...) التي تقبل ثلاث قيم: الأولى هي this التي تشير إلى الصنف الحالي، أمّا الثانية فتشير إلى واجهة تصميم جاهزة تحتوي على textView؛ ولأنّها جاهزة لا أستطيع الوصول لها إلّا إذا كتبت كلمة android التي تشير إلى مجلّد جاهز ويحتوي على ملفّات معدّة وجاهزة، للوصول إلى محتويات المجلّد أكتب نقطة فتظهر جميع الملفات، أكتب: R.layout.simple_expandle_list_item_1 هذه هي الواجهة المصمّمة التي أريد التعامل معها، أمّا العنصر الثالث فهو مصدر البيانات الذي عرّفته سابقاً.
3. لإظهار البيانات على القائمة listView، أكتب id الخاصّ بالقائمة، ثمّ نقطة (.)، وأكتب adapter الذي يستقبل الـ arrayAdapter الذي عرّفته وهو a، كما في السطر الثالث من الصورة أعلاه.

بعد إضافة البيانات، أريد أن أعرض النتائج، أشغّل المشروع وأرى النتائج، كما يأتي:



بعد أن تأكّدت من ظهور القائمة، أريد إعطاء أوامر للعناصر التي تحتوي على البيانات، عند اختيار عنصر يظهر النص في toast، كيف؟

أكتب listView (id الخاصّ بالقائمة)، ثم نقطة (.) وأكتب setOnItemClickListener فيظهر خياران أختار الثاني كما يأتي:



الصورة التالية تظهر بعد اختياري للخيار الثاني، تستقبل أربع قيم: adapterView وهي من نوع arrayAdapter تعادل a التي عرّفناها سابقاً، view من نوع view تعادل العنصر الذي تمّ اختياره ، i من نوع Int تعادل موقع العنصر الذي تمّ اختياره، L من نوع Long تعادل id العنصر. بعد الإشارة <- أكتب الأوامر التي أريد تنفيذها عند الضغط على العنصر:

```
listView.setOnItemClickListener { adapterView, view, i, l ->

}
```

أُضيف السطرين الآتيين داخل الدالّة:

```
val n = adapterView.getItemAtPosition(i).toString()
Toast.makeText( context: this,n,Toast.LENGTH_LONG).show()
```

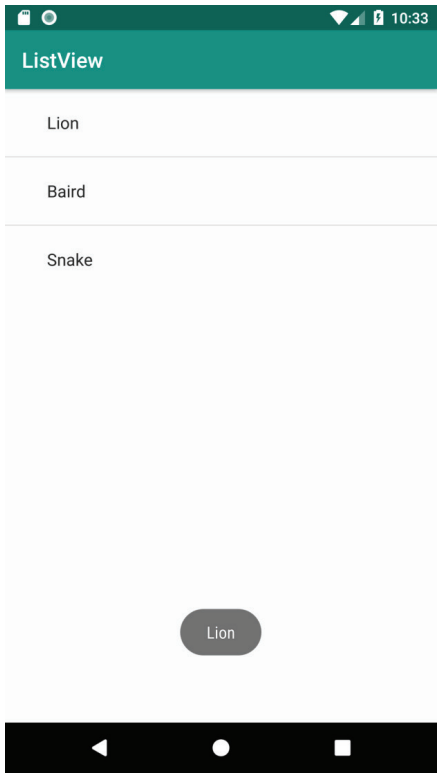
السطر الأوّل يخزّن محتوى العنصر الذي تمّ الضغط عليه في المتغيّر n، استخدمت من القيم المعطاة adapterView و i. هناك دالة اسمها getItemAtPosition(…) موجودة داخل صنف ArrayAdapter، وبما أنّ adapterView معرّفة من الصنف نفسه فيمكن لها الوصول إلى الدالّة المطلوبة، هذه الدالّة تستقبل قيمة رقم صحيح هو موقع العنصر، وتعيد النصّ الموجود في العنصر. والهدف من تعريف متغيّر من نوع String هو استخدامها في toast.
يمكن إضافة أي أوامر أخرى أريدها.

أتذكّر: فائدة (toString)، وطريقة عمل (toast).

يصبح الكود النهائيّ كالآتي:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)


    val name = arrayOf("Lion","Baird","Snake")


    val a:ArrayAdapter<String> = ArrayAdapter( context: this,android.R.layout.simple_expandable_list_item_1,name)



    listView.adapter=a

    listView.setOnItemClickListener { adapterView, view, i, l ->

        val n = adapterView.getItemAtPosition(i).toString()
        Toast.makeText( context: this,n,Toast.LENGTH_LONG).show()

    }




}
}
```

أعمل تشغيلاً للمشروع، وأفحص النتائج:

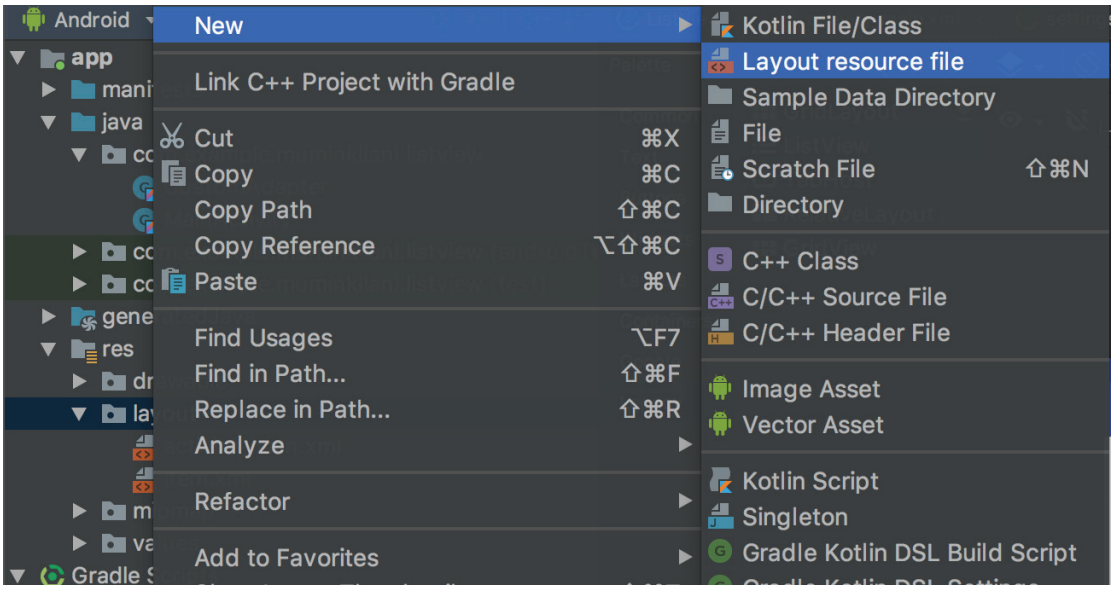


نشاط (2) عملي: أُغيّر مصدر العناصر من مصفوفة عادية (array) إلى مصفوفة القوائم (listArray).

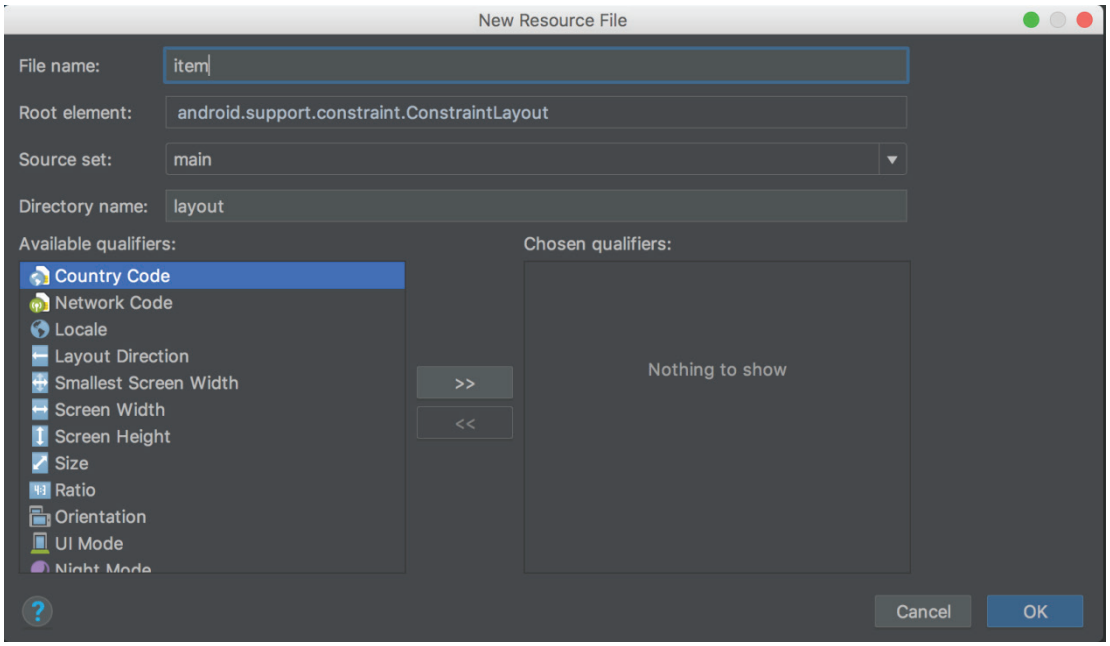
إنشاء list view مخصّصة باستخدام ArrayAdapter:

أُنشئ مشروعاً جديداً وأسمّيه ListViewCustom او أي اسم آخر، أُضيف listView كما تعلّمت سابقاً.

أُنشئ ملف واجهة تصميم جديداً وأسمّيه item، ويتمّ ذلك بالضغط بزر الفأرة اليمين على مجلّد layout، أختار new-> layout resourc file؛ لأنّ القائمة مخصّصة، ويعني ذلك تخصيص عناصر (items) القائمة.



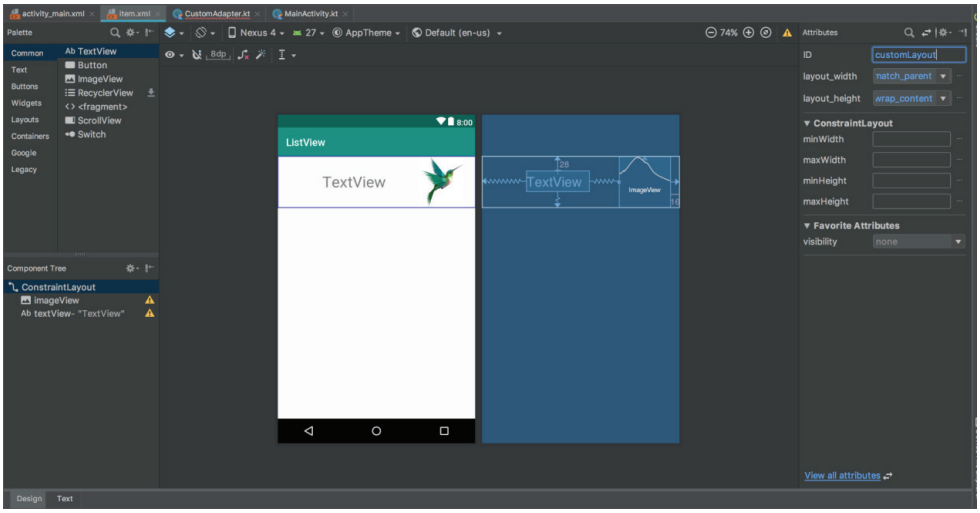
تظهر النافذة الآتية أكتب اسم الملف، ثمّ أضغط ok:



أريد عرض في كلّ عنصر من القائمة صورة ونصّاً؛ لذلك أضيف في ملف item textView، وimageView، ثمّ أُعطيها الخصائص المناسبه حتى تظهر في القائمه بشكل مناسب.

قبل أن أُضيف العناصر أجلب الصور المراد وضعها في عناصر القائمة، ويمكن أن تكون صورة واحدة وأضعها في مجلد كمجلد drawable.

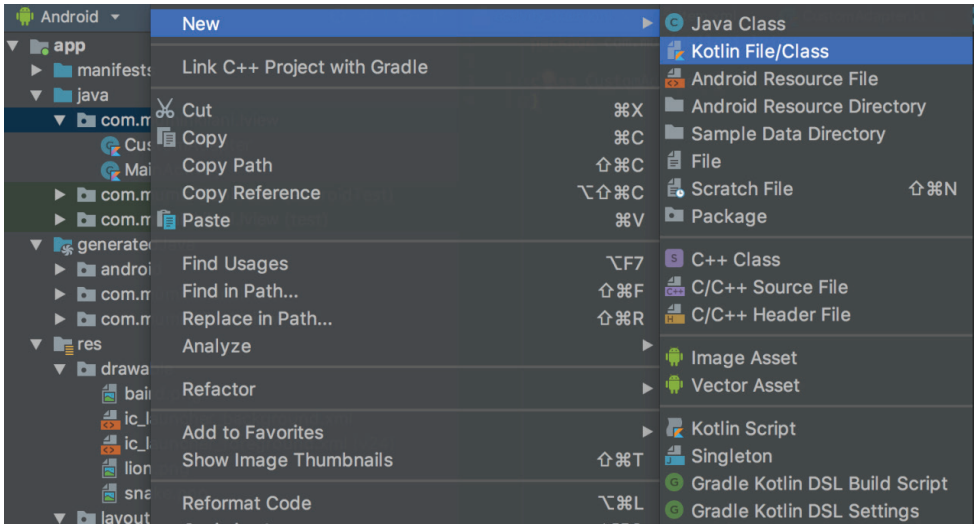
ما المقصود؟؟؟

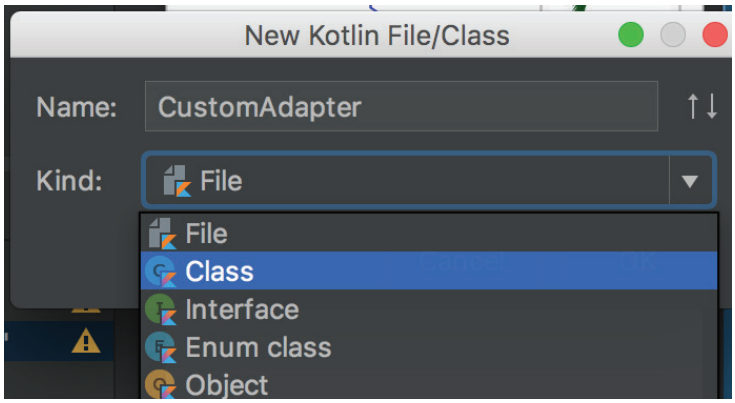


أعطي خصائص محدّدة لـ imageView من حيث الطول والعرض، هنا أعطيتها قيم 100*100، أمّا فيما يتعلّق بـ constraintLayout فمن الضروريّ تحديد ارتفاعها، هنا أعطيتها وفق ما بداخلها (wrap_content)، وهذا الملف يُعدّ الشكل الافتراضيّ لعنصر القائمة (item).

انتهيت من ملف التصميم فأنتقل إلى مجلد الكود، ذكرت سابقاً أنّه يجب أن يكون وسيط بين مصدر العناصر وقائمة العناصروهو arrayAdapter، واستخدمت layout افتراضيّة، لكنّني هنا أنشأت layout مخصّصة لعناصر القائمة؛ لذلك يجب أن أُنشئ arrayAdapter مخصّصة في ملف منفصل.

أضغط بزر الفأرة اليمين على مجلد الكود بعدها new-> kotlin file/class، أعطيه اسم CustomAdapter

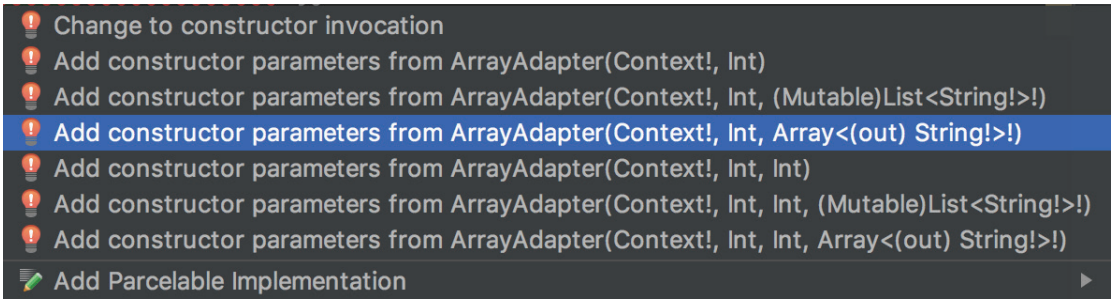




أجعل الصنف CustomAdapter يرث الصنف ArrayAdapter كما يأتي:

```
class CustomAdapter : ArrayAdapter<String> {
}
```

هنا يطلب البرنامج إضافة باني (construtor) تابع لـ ArrayAdapter، أضغط من لوحة المفاتيح على alt+enter، أوأضغط على فتظهر قائمة كما يأتي:



أختار الخيار الثالث أو الرابع، وفق مصدر البيانات الذي أريد إنشاءه، فإذا كان مصفوفة أختار الرابع، أمّا إذا كان مصفوفة قائمة فأختار الثالث.

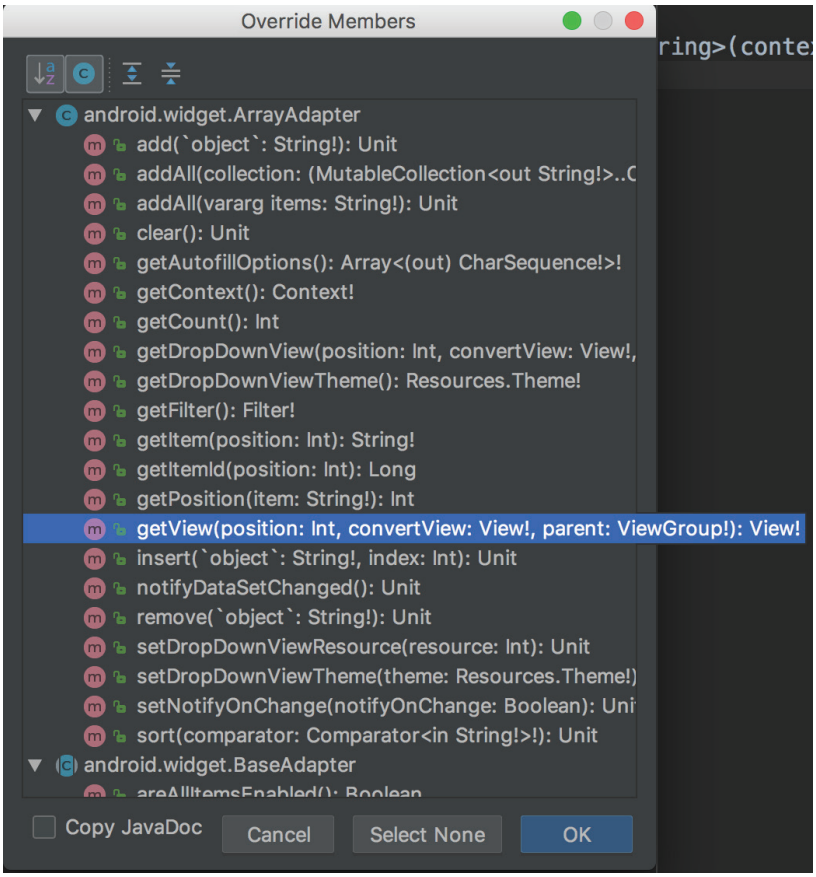
```
class CustomAdapter(context: Context?, resource: Int, objects: Array<out String>?) : ArrayAdapter<String>(context, resource, objects) {
}
```

أُجري بعض التغييرات كما يأتي:

```
context: Context?,private val objects: Array<out String>?,private val image: Array<out Int>?) : ArrayAdapter<String>(context, R.layout.item, objects)
```

1. أُضيف مصفوفة من نوع int لكي تستقبل الصور.
2. أُضيف private val قبل أسماء متغيّرات المصفوفات، حتى أتمكن من استخدامها داخل الصنف الحالي.
3. استبدل كلمة resource في صنف الأب إلى R.layout.item مكان تواجد واجهة التصميم التي أنشأتها سابقاً.

بعد إضافة الباني، أعمل استدعاء لدالّة اسمها getView لربط ملف item.xml الذي أنشأته سابقاً مع الصنف الحالي:




```kotlin
@SuppressLint( ...value: "ViewHolder")
override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {

    val l:LayoutInflater= LayoutInflater.from(context)
    val customView:View= l.inflate(R.layout.item,parent, attachToRoot: false)
    return customView
}
```

- أُنشئ كائناً من الصنف LayoutInflater وظيفته ربط الواجهة بالكود؛ لأنّ الواجهة ليست تابعة لهذا الصنف:
  1. LayoutInflater.from(context)، تعني أنّ قيمة l مرتبطة بالصنف الحالي (context تشير إلى الصنف الحالي).
  2. l.inflate(R.layout.item,parent,false)، بعد أن تمّ ربط المتغيّر l بالكود، أريد أن أربطه بملف التصميم، فتصبح l وسيطاً بين ملف الكود وملف التصميم.
- أُنشئ كائناً أسمّيه customView من الصنف View لأتمكّن من الوصول إلى العناصر التي بداخل ملف item.xml.
- أجعل الدالّة ترجع قيمه المتغيّر الذي عرّفته، وهو customView.

الشكل (3-2) يوضّح وظيفة layoutInflater:

الشكل (3-2)

بما أنّ النصّ والصورة مختلفان في كلّ عنصر، أُضيف الأكواد الآتية:

```
customView.textView.text = objects!![position]
customView.imageView.setImageResource(image!![position])
```

كلمة position هي متغيّر من نوع int معرّف في الباني الخاصّ في الدالّة getView.
objects وimage المصفوفات التي قمنا بتعريفها داخل الباني في الصنف CustomAdapter.

```
package com.muminkilani.lview

import ...

class CustomAdapter(context: Context?,private val objects: Array<out String>?,private val image

    @SuppressLint( ...value: "ViewHolder")
    override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {

        val l:LayoutInflater= LayoutInflater.from(context)
        val customView:View= l.inflate(R.layout.item,parent, attachToRoot: false)

        customView.textView.text = objects!![position]
        customView.imageView.setImageResource(image!![position])

        return customView
    }
}
```

أتذكّر: فائدة علامتَي التعجب !!.

أنتقل الآن إلى ملف الكود الرئيسي MainActivity، وأنفّذ الخطوات الآتية:

1. أُعرّف مصفوفتين: واحدة أضع فيها الأسماء التي أريدها، والأخرى تحتوي على الصور.

```
val name = arrayOf("Lion", "Baird", "Snake")
val image = arrayOf(R.drawable.lion, R.drawable.baird, R.drawable.snake)
```

2. أنشئ كائناً أسمّيه myAdapter من الصنف CustomAdapter الذي أنشأته سابقاً، الذي يستقبل ثلاث قيم: الأولى تشير إلى الصنف الحالي، والقيمتان الأخريان هما المصفوفتان name، image.

```
val myAdapter = CustomAdapter( context: this, name, image)
```

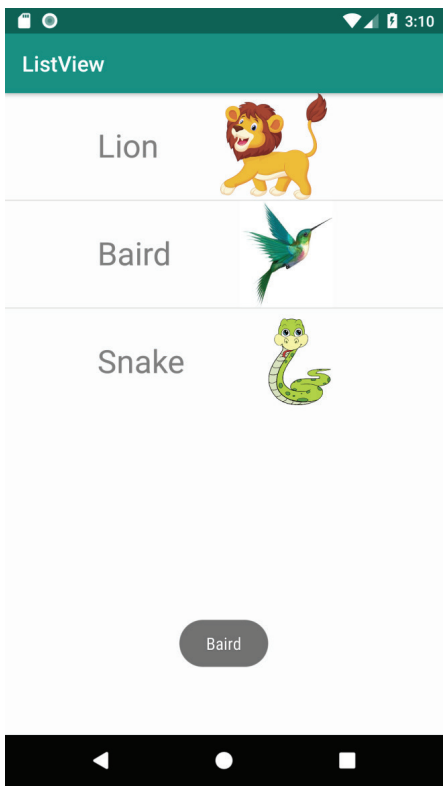
3. لإظهار البيانات على القائمة listView، أكتب id الخاصّ بالقائمة، ثمّ نقطة، وأكتب adapter وأعطيها قيمة myAdabter.

```
listView.adapter = myAdapter
```

يصبح الكود كما يأتي:

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)


        val name = arrayOf("Lion", "Baird", "Snake")
        val image = arrayOf(R.drawable.lion, R.drawable.baird, R.drawable.snake)


        val myAdapter = CustomAdapter( context: this, name, image)

        listView.adapter = myAdapter

        listView.setOnItemClickListener { adapterView, view, i, l ->
            val stringText = adapterView.getItemAtPosition(i).toString()
            Toast.makeText( context: this, stringText, Toast.LENGTH_LONG).show()
        }


    }
}
```

أُشغّل المشروع، وأفحص النتائج:



نشاط (3) عملي: أُغيّر مصدر العناصر من مصفوفة عادية (array) إلى مصفوفة القوائم (listArray).

Grid view:

عرض عناصر البيانات على شكل صفوف وأعمدة كالجداول، تُستخدَم لعرض عدد غير محدود من البيانات بشكل منظّم، قد تكون البيانات نصوصاًن او صوراً، أو فيديو، ...
عدد البيانات المعروضه يكون وفق حجم الشاشة المستخدمة، ولا تستهلك من الذاكرة (ram) إلّا وفق عدد العناصر المعروضة على الشاشة، وتعمل كقائمة listView تماماً، وتختلف في الشكل فقط.

- **أنواع Grid view:**

1. عادية: عرض البيانات بشكل العناصر الافتراضيّ.
2. مخصّصة: عرض البيانات بشكل مخصّص للعناصر.

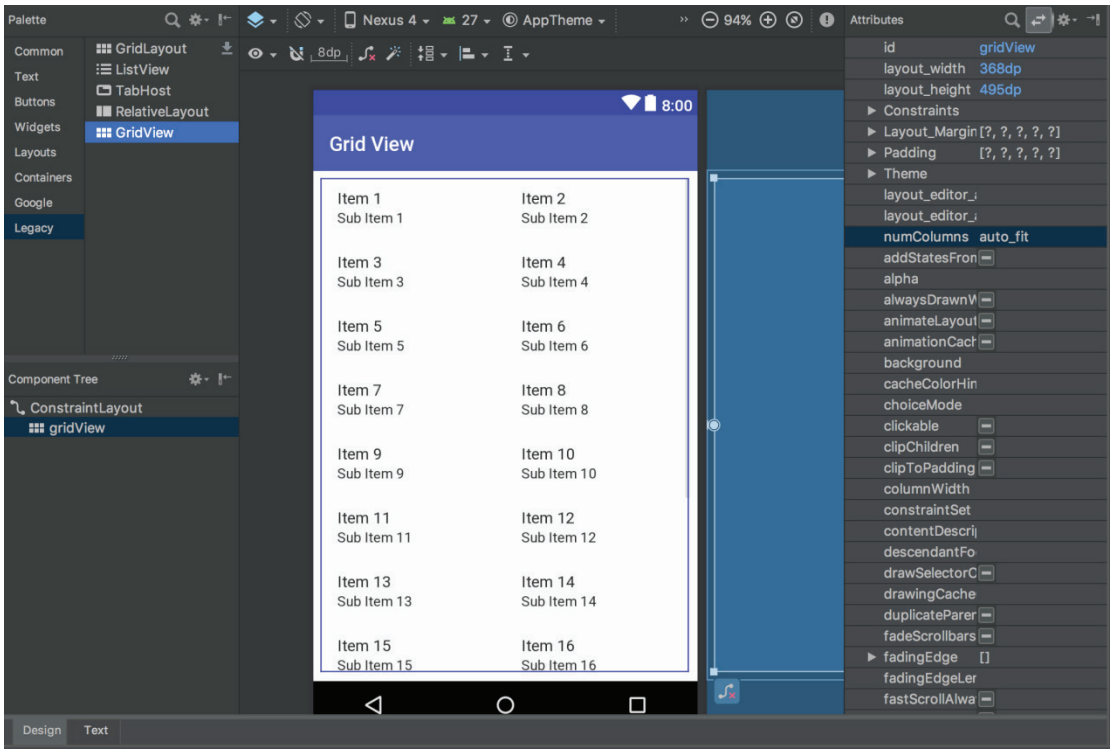
نشاط (4) للاطلاع: أبحث عن تطبيقات تستخدم grid view.

لا أريد أن أتطرّق إلى الكود البرمجيّ في هذا الموضوع؛ لأنّ gridView تستخدم أكواد listView، والاختلاف فقط في ملفّات التصميم.

- **إنشاء Grid view عاديّة:**

نشاط (5) عملي: أُنشئ مشروعاً يعرض أسماء حيوانات بطريقة gridView.

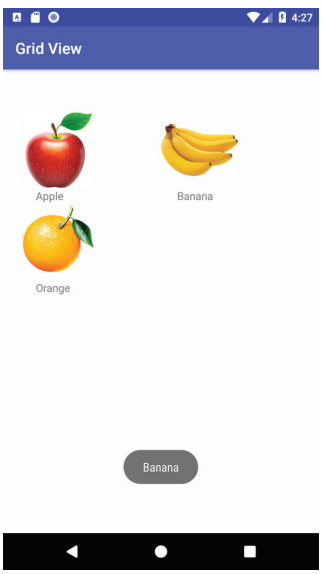
أُنشئ مشروعاً جديداً، أُضيف من قائمة العناصر gridView، ألاحظ عند إضافتها أنّها تعرض البيانات ك listView؛ لذلك أنتقل إلى قائمة خصائص الـ gridView، وأختار auto_fit من خيار numCulomn كما يأتي:



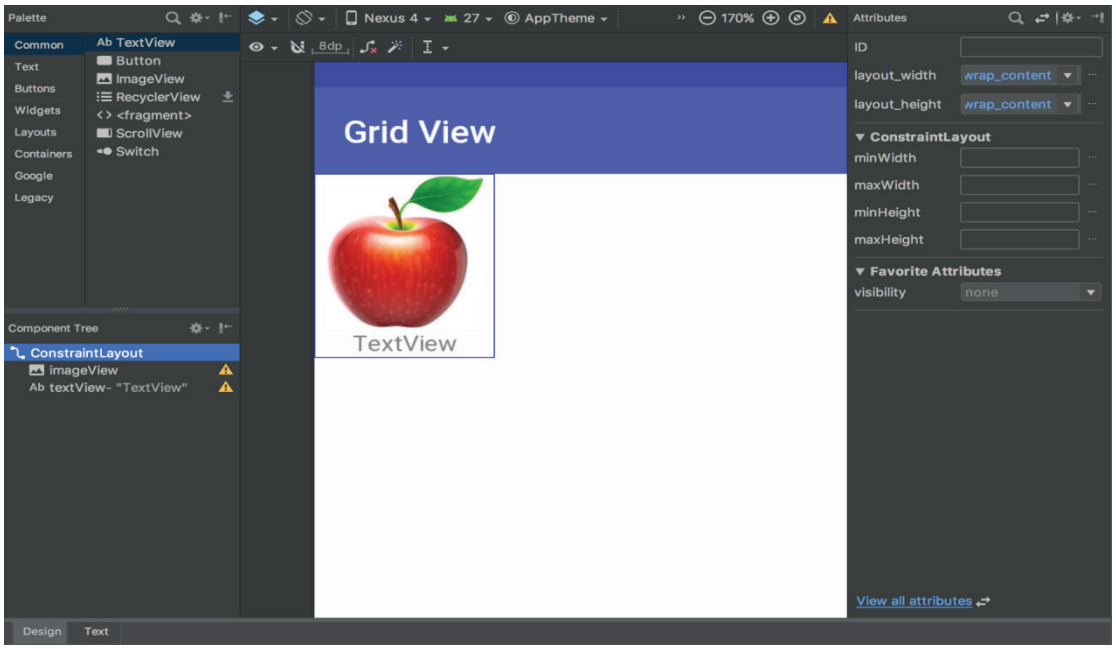
أنتقل إلى الكود البرمجيّ، وأكتب كما كتبت عندما أنشأت listView عاديّة، ثمّ أختبر النتائج.

- **إنشاء Grid view مخصّصة:**

نشاط (6) عملي: أُنشئ gridview تعمل كما في الصورة الآتية:

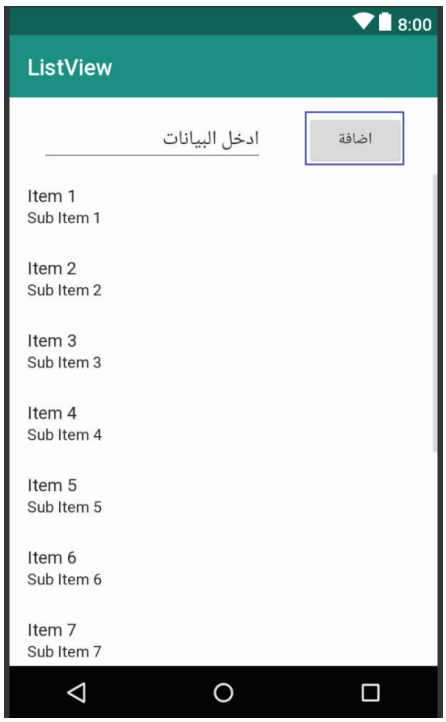


أنشئ مشروعاً جديداً، أُضيف gridView، ثمّ أُنشئ ملف layout أُسمّيه item، قبل ذلك أتأكّد من وضع الصور التي أريدها في ملف drawable، وأُضيف كما يأتي:



أنتقل إلى الكود، وأكتب كما كتبت عندما أنشأت listView مخصّصة، ثمّ أختبر النتائج.

نشاط (7) عملي: أُنشئ تطبيقاً يحتوي على قائمة بيانات فارغة، وأريد إدخال بيانات بعد ظهور التطبيق على الهاتف الذكيّ.
أنشئ مشروعاً جديداً، وأضع في واجهة التصميم كما يأتي:



أنتقل إلى الكود البرمجيّ، وأقوم ببعض التغييرات البسيطه:

التغييرات في ملف كود الصنف CustomAdapter:

```
class CustomAdapter(context: Context?,private val objects: ArrayList<String>?) : ArrayAdapter<String>(context, R.layout.item, objects) {

    @SuppressLint( ...value: "ViewHolder")
    override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {

        val l:LayoutInflater= LayoutInflater.from(context)
        val customView:View= l.inflate(R.layout.item,parent, attachToRoot: false)

        customView.textView.text = objects!![position]

        return customView
    }
}
```

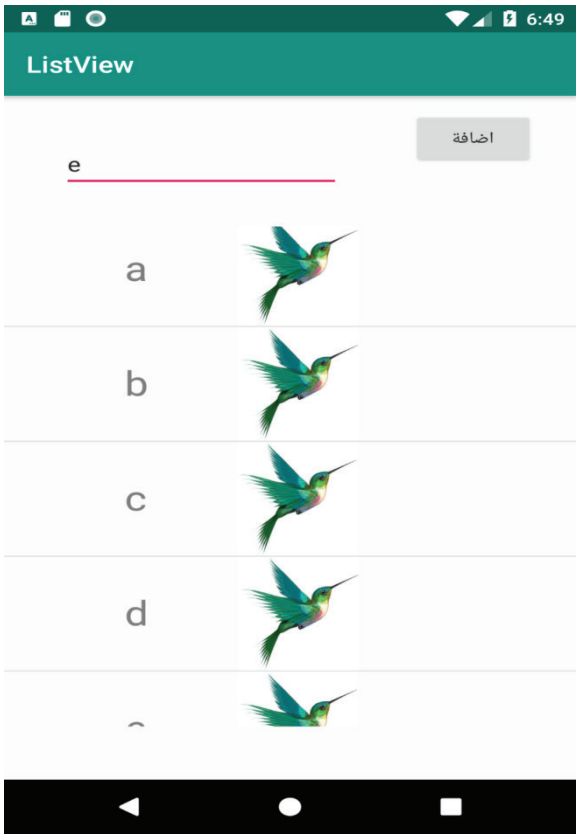
أُغيّر نوع المصدر من مصفوفة عاديّة إلى مصفوفة قائمة

```
private val objects: ArrayList<String>?
```

التغيير في الكود الرئيسيّ MainActivity كما يأتي:

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)


        val arrayList = ArrayList<String>()


        button.setOnClickListener { it: View!
            val name: String? = editText.text.toString()
            arrayList.add(name!!)
            val cA = CustomAdapter( context: this, arrayList)
            listView.adapter = cA
        }


    }
}
```

عمل تشغيلاً للمشروع، والاحظ النتائج:



الأسئلة

1. ما المقصود بـ listView؟
2. أوضّح بالرسم وظيفة layoutInflater.

3-1 الموقف التعليميّ التعلُّميّ

الوسائط المتعدّدة (multimedia).

وصف الموقف التعليميّ: طلبت وزارة التربية والتعليم العالي من المدارس المِهْنيّة الموجودة في فلسطين إنشاء تطبيق يعرض فيديو عن أهمّ نشاطات المدرسة وإنجازاتها السنويّة.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من الوزارة عن: - إصدار نظام الأندرويد. - حجم الفيديو وصيغته. - مدّة الفيديو. • أجمع البيانات عن : - العناصر(views) المستخدمة للإدخال والإخراج. - مكان ملف الفيديو. - الفيديو. - الجمل البرمجيّة المستخدمة.	• حوار ومناقشة. • التعلُّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب الوزارة، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّر	• أصنّف البيانات (نظام الأندرويد المستخدم). أحدّد: - المعطيات (حجم الفيديو، صيغة الفيديو (mp4، wmv،...)، مدة الفيديو). - العناصر (views) المستخدمة. - مكان ملف الفيديو. - الفيديو. - الجمل البرمجيّة. • إعداد خطّة للتنفيذ.	• التعلُّم التعاونيّ/ مجموعات. • حوار ومناقشة .	• وثائق (نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد) • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).

أُنفّذ	• اختيار عدد من الواجهات المقترحة قابلة للاستخدام في تطبيق الأندرويد. • اختيار واجهة التطبيق المناسبة وفق طلب الوزارة. • أُصمّم وأُبرمج واجهة التطبيق. • اختبار المشروع على جهاز افتراضي، أو على جهاز يعمل بنظام أندرويد.	• التعلُّم التعاونيّ/ مجموعات. حوار ومناقشة. • عصف ذهنيّ/ استمطار الأفكار.	• حاسوب. • الطلب الخاص بالوزارة • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.
أتحقّق	• أتحقّق من ملاءمة اختيار الواجهة المقترحة للتطبيق. • أتاكّد من ملاءمة التطبيق. • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل مناسب.	• التعلُّم التعاونيّ/ مجموعات. • حوار ومناقشة.	• الخطّة المعدّة سابقاً. • الطلب الخاص بالوزارة. • حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام الأندرويد.
أوثّق وأقدّم	• أوثّق البيانات التي تتعلّق بـ: - العناصر (views) المستخدمة للإدخال والإخراج. - مكان ملف الفيديو. - الفيديو. - الجمل البرمجيّة المستخدمة. • إعداد ملف بالحالة (إنشاء تطبيق يعرض فيديو عن أهمّ نشاطات المدرسة وإنجازاتها السنويّة). • إعداد عرض تقديميّ. • أُقدّم التطبيق للوزارة.	• حوار ومناقشة. • التعلُّم التعاونيّ/ مجموعات عمل.	• حاسوب. • شاشة عرض. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.
أقوّم	• رضا الوزارة عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات.	• حوار ومناقشة.	• طلب الوزارة. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد يهتم بواجهات التطبيق وعناصرها وإنشاء قوائم البيانات.

نشاط (1) للاطلاع: أبحث مع زملائي عن صِيَغ متعدّدة للفيديو والصوت.

نشاط (2) للاطلاع: أبحث عن استخدامات للوسائط المتعدّدة في حياتي اليوميّة.

الوسائط المتعدّدة: هو مصطلح واسع الانتشار يشير إلى استعمال عدة وسائط مختلفة مثل: (الصوت، والصور، والفيديو) لتقديم المحتوى بطريقة تفاعليّة؛ لتحقيق هدف أو عدّة أهداف محدّدة المعلومات.

من الوسائط التي أريد استخدامها في الأندرويد:

1. عارض الفيديو.
2. مشغّل الصوت.
3. عارض الصور (تعلمتها سابقاً في الصف الحادي عشر).

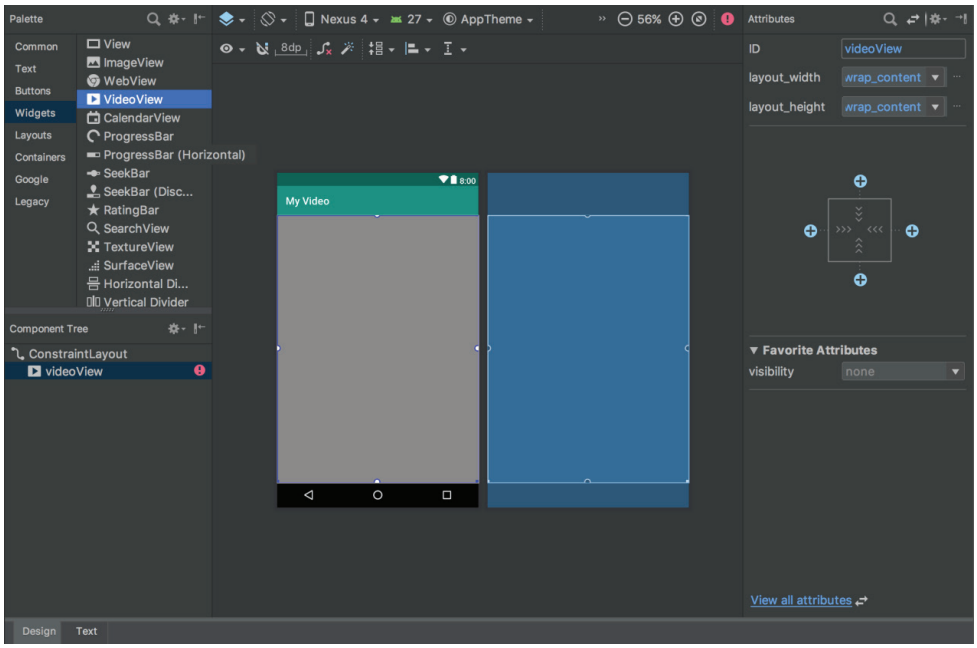
أوّلاً: عارض فيديو (videoView):

يعرض ملفّات الفيديو على جهاز الأندرويد، إمّا أن يكون الفيديو من موقع على الإنترنت، أو داخل ملف المشروع.

كيفيّة عمل عارض فيديو:

نشاط (3) عملي: عمل عارض فيديو (videoView) يعرض فيديو من الإنترنت:

أُنشئ مشروعاً جديداً، وأُضيف عارض فيديو (videoView) كالآتي:



أستطيع تغيير حجم عرض الفيديو، بتغيير أبعاد عارض الفيديو (videoView).

أنتقل إلى الكود البرمجيّ، وأكتب الأوامر الآتية داخل onCreate():

```
videoView.setVideoURI(Uri.parse( uriString: "https://video.wixstatic.com/video/80c05f_e2c524db1f264178a8558c92dbf76fb0/1080p/mp4/file.mp4"))
val mediaController = MediaController( context: this)
videoView.setMediaController(mediaController)
videoView.start()
```

1. أضفت رابط الفيديو الذي أريد أن أعرضه في عارض الفيديو، مع الانتباه عند إضافه رابط لفيديو يجب أن ينتهي الرابط باسم الفيديو وامتداده (file.mb4)، ويكون كاملاً مع https://، وإلّا لن يعمل الفيديو على عارض الفيديو. setVideoURI خاصّة لإضافة الرابط، كتبت بداخلها Uri.parse حتى يتمّ التعامل مع النصّ على شكل رابط إنترنت.
2. أنشات كائناً من الصنف MediaController وأعطيته الاسم نفسه، يشير إلى activity الحاليّة لتشغيل أدوات التحكُّم للفيديو (التشغيل، والإيقاف، والتقديم، والتاخير).
3. ربطت أدوات التحكُّم بالفيديو بواسطة setMediaController(mediaController).
4. أخيراً أعطيت أمراً لتشغيل الفيديو تلقائيّاً عندما يتم تشغيل التطبيق videoView.start()، ويمكن الاستغناء عنها عندما لا أريد تشغيل الفيديو تلقائيّاً.
5. بما أنّ الفيديو عن طريق الإنترنت أُضيف صلاحيّة الوصول إلى الإنترنت في ملف manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.muminkilani.myvideo">
<uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Video"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

manifest

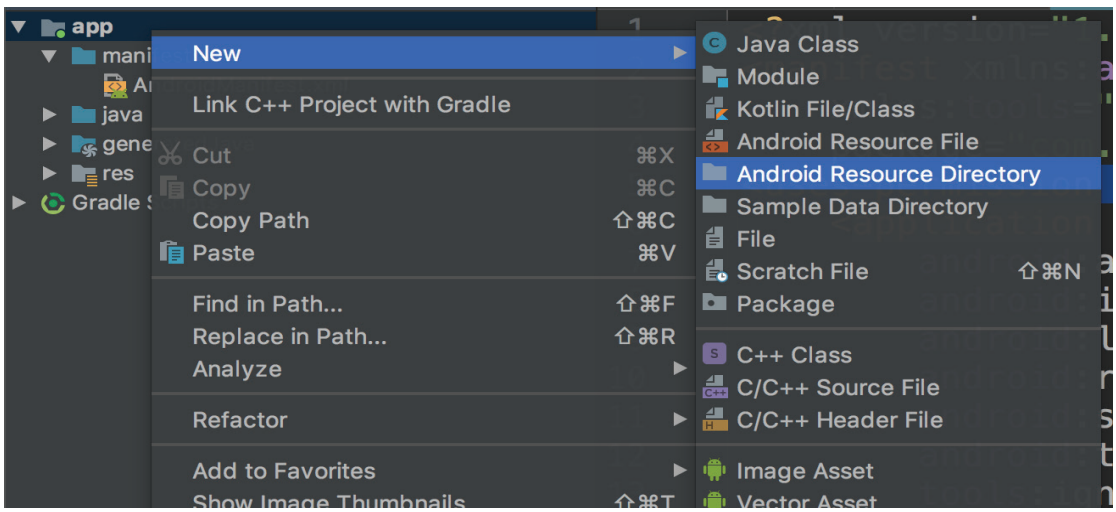
أشغّل المشروع، وأعمل اختباراً للتطبيق، وأفحص النتائج:

نشاط (4) عملي: عمل عارض فيديو (videoView) يعرض فيديو من داخل ملف المشروع:

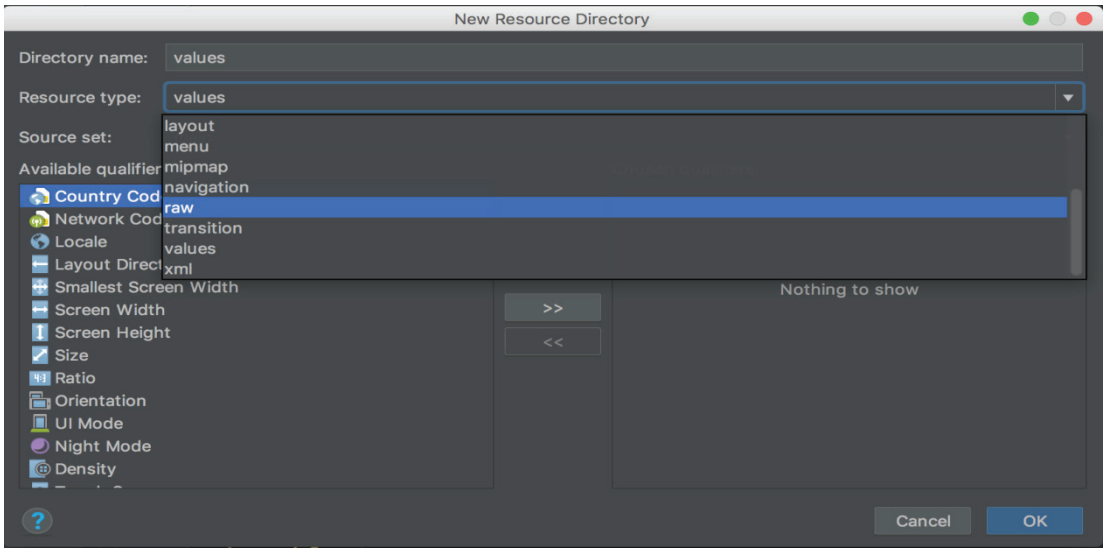
أُنشئ مشروعاً جديداً، وأُضيف عارض فيديو كما تعلمت سابقاً، قبل أن أنتقل إلى الكود البرمجيّ، أُنشئ ملفاً اسمه raw وأضع الفيديو الذي أريد عرضه داخله.

لإضافة الملف أتّبع الخطوات الآتية:

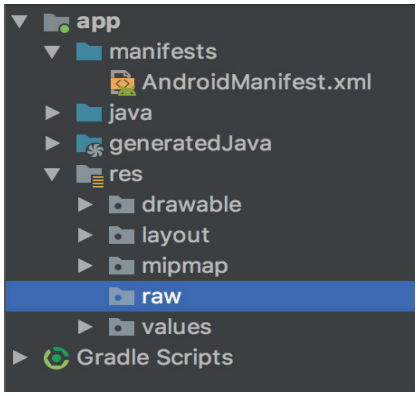
1. أضغط بزر الفأرة اليمين على ملف المشروع، وأختار new-> Android Resource Directory



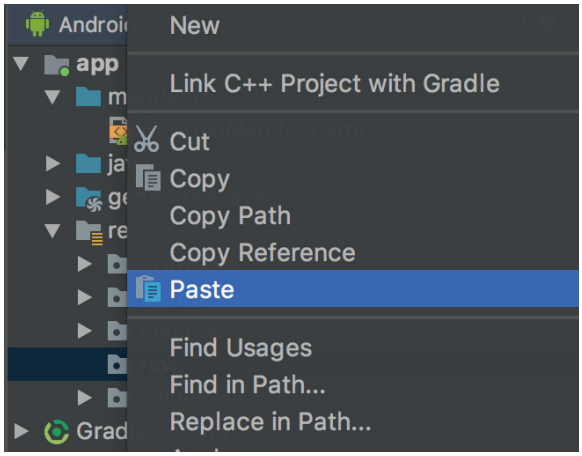
2. أختار من قائمة Resource type نوع الملف rew كالآتي:



3. تمّ إنشاء الملف ويظهر كالآتي:



بعد إنشاء ملف raw أنسخ الفيديو من جهازي، وألصقه في الملف:



أنتقل إلى الكود البرمجيّ، وأكتب الأوامر الآتية:

```
videoView.setVideoURI(Uri.parse( uriString: "android.resource://" + packageName + "/" + R.raw.file ))
val mediaController = MediaController( context: this)
videoView.setMediaController(mediaController)
videoView.start()
```

قمت بتغيير مكان رابط الفيديو فقط ، واستبدلته بِـ:

" android.resource://" + packageName + "/" + R.raw.file"

بما أنّ الفيديو موجود في ملف المشروع أكتب android.resource:// ثمّ packageName + (تجلب اسم حزمة التطبيق) ثمّ "/" ثمّ مكان تواجد الملف، ويمكن كتابة السطر كالآتي:

android.resource://com.muminkilani.myvideo/" + R.raw.file."

أُشغّل المشروع، وأعمل اختباراً للتطبيق، وأفحص النتائج.

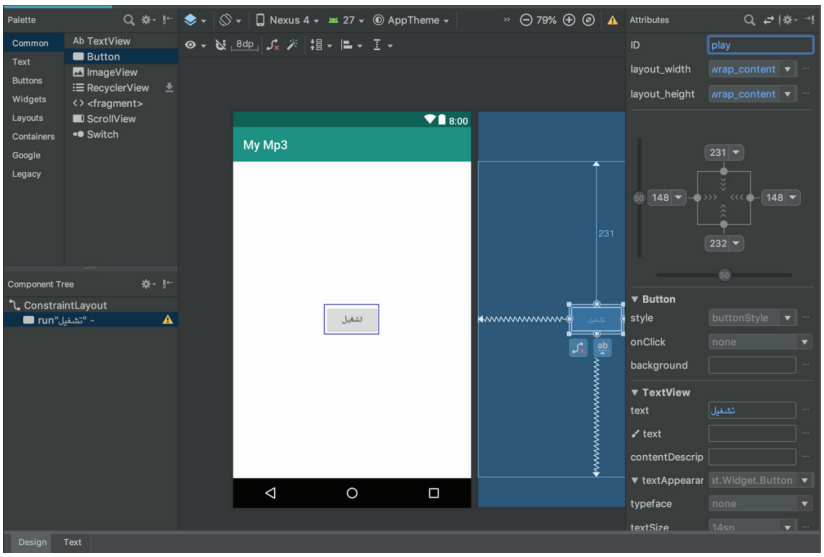
ثانياً: مشغّل الصوت:

يعمل على تشغيل الأصوات بصيغ مختلفة، مثل: الموسيقى، أو تسجيلات وغيرها، ويُستحسَن استخدام صوتيّات mp3، ويمكن استخدام هذه الملفّات من موقع، أو وضع ملف الصوت في التطبيق نفسه.

كيفيّة عمل مشغّل صوت بسيط:

نشاط (5) عملي: عمل تطبيق يقوم بتشغيل صوت من موقعٍ ما:

أحتاج لعمل مشغّل صوت إضافة عنصر من نوع button لعمل تشغيل وإيقاف تشغيل الصوت، أُنشئ مشروعاً جديداً، وأُضيف button، وأُعطيه اسم تشغيل وأعطيه id مناسباً كما يأتي:



أنتقل إلى الكود البرمجيّ، وأكتب ما يأتي:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

1.  val mediaPlayer = MediaPlayer()

    play.setOnClickListener { it: View!

        if (play.text=="تشغيل") {
2.          mediaPlayer.reset()
            mediaPlayer.setDataSource("https://server6.mp3quran.net/balilah/001.mp3")
            mediaPlayer.prepare()
            mediaPlayer.start()
            play.text= "ايقاف"
        }else if (play.text=="ايقاف"){
3.          mediaPlayer.stop()
            play.text= "تشغيل"
        }

    }
}
```

1. أُنشئ كائناً من الصنف MediaPlayer() يعمل مشغّلاً للصوت، ثمّ أنتقل إلى داخل button.
2. أكتب جملاً شرطيّة لكي أُغيّر وضع button من التشغيل للإيقاف والعكس، الجملة الشرطيّة الأولى تفحص إذا كان نصّ button "تشغيل"، وتنفّذ الأوامر الآتية:
   - mediaPlayer.reset(): تهيئة مشغّل الصوت لإعادته للوضع الافتراضيّ؛ أي لا يحتوي على ملف صوتيّ.
   - mediaPlayer.setDataSource(): أُضيف لها عنوان رابط المقطع الصوتيّ، والتأكّد من أنّه ينتهي بامتداد معيّن، مثل (.mp3)، ويبدأ بِـ (https أو http).
   - mediaPlayer.prepare(): وظيفتها تهيئة مشغّل الصوت لبدء التشغيل.
   - mediaPlayer.start(): مهمّتها بدء التشغيل.
   - play.text ="إيقاف": تغيير نصّ button من تشغيل إلى إيقاف.
3. إذا لم يتحقّق الشرط الأول هذا يعني أنّ مشغّل الصوت يعمل ونصّ button هو «إيقاف»؛ لذلك كتبت جملة شرطيّة أخرى لفحص نصّ button إذا كان "إيقاف"، وتنفّذ الأوامر الآتية:
   - mediaPlayer.stop(): إيقاف تشغيل الصوت.
   - play.text = "تشغيل": تغيير نصّ button من إيقاف إلى تشغيل.

قبل تشغيل المشروع أتأكّد من إضافة صلاحيّة استخدام الإنترنت كما تعلمت سابقاً، ثمّ أُشغّل المشروع وأختبر التطبيق، وأفحص النتائج.

نشاط (6) عملي: عمل تطبيق يقوم بتشغيل صوت من داخل ملف المشروع:

أُنشئ مشروعاً جديداً، وأُضيف button كما في نشاط (4)، أُنشئ ملفاً اسمه raw، وأضع ملف الصوت الذي أريده داخل ملف raw كما تعلّمت في نشاط (3)، ثمّ أنتقل إلى الكود البرمجيّ، وأكتب كما في النشاط السابق، مع وجود بعض التغييرات الموضّحة كالآتي:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)


1.  var mediaPlayer = MediaPlayer()

    play.setOnClickListener { it: View!

        if (play.text=="تشغيل") {
            mediaPlayer.reset()
2.          mediaPlayer = MediaPlayer.create( context: this,R.raw.m)
            mediaPlayer.start()
            play.text= "ايقاف"
        }else if (play.text=="ايقاف"){
            mediaPlayer.stop()
            play.text= "تشغيل"
        }

    }
}
}
```

1. تغيير نوع الكائن إلى var لكي أستطيع أن أعيد تعريفه بطريقه أخرى.
2. داخل الجمله الشرطيّة الأولى حذفت mediaPlayer.setDataSource()، و mediaPlayer.() prepare، ثمّ أضفت هذا السطر:

mediaPlayer = MediaPlayer.create(this, R.raw.(اسم الملف الصوتيّ)

الذي يعمل على تشغيل ملف الصوت من داخل ملف raw.

هنا لا أحتاج لإضافة صلاحيّة الوصول للإنترنت، أُشغّل المشروع، ثمّ أختبر التطبيق، وأفحص النتائج.

الأسئلة

1. ما المقصود بالوسائط المتعدّدة؟
2. كيف أنشئ ملف raw؟

4-1 الموقف التعليميّ التعلُّميّ

مُمَرِّر الصفحات (scroll view).

وصف الموقف التعليميّ: طلب شاعر مشغل تطبيقات الهاتف المحمول في مدرسة جنين المهنيّة، مساعدته في إنشاء تطبيق يعرض فيه ديواناً شعريّاً خاصّاً به.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من الشاعر عن: - إصدار نظام الأندرويد. - ديوان الشعر. • أجمع البيانات عن: - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - الجمل البرمجيّة المستخدمة.	• حوار ومناقشة. • التعلُّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب الشاعر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّر	• أُصنّف البيانات (نظام الأندرويد المستخدم). • أُحدّد: - المعطيات: (عنوان الديوان، عدد القصائد وأسماؤها). - العناصر (views) المستخدمة. - عدد الواجهات. - الجمل البرمجيّة. • إعداد خطّة للتنفيذ.	• تعلّم تعاونيّ/مجموعات. • حوار ومناقشة .	• وثائق (نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أنفّذ	• اختيار عدد من الواجهات المقترحة قابلة للاستخدام في تطبيق الأندرويد. • أتّفق مع الشاعر على اختيار واجهات التطبيق المناسبة. • أُصمّم وأُبرمج واجهات التطبيق. • أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد.	• تعلّم تعاونيّ/مجموعات. • حوار ومناقشة. • عصف ذهنيّ / استمطار الأفكار.	• حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.

أتحقّق	• أتحقّق من ملاءمة اختيار الواجهة المقترحة للتطبيق. • أتاكّد من ملاءمة التطبيق. • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز بشكل مناسب.	تعلّم تعاونيّ/مجموعات. حوار ومناقشة.	• الخطّة المعدّة سابقاً. • الطلب الخاصّ بالشاعر. • حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام الأندرويد..
أوثّق وأقدّم	• أوثّق البيانات التي تتعلّق بـ: - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - الجمل البرمجيّة المستخدمة. • إعداد عرض تقديميّ. • إعداد ملف بالحالة (إنشاء تطبيق يُعرَضُ فيه ديوان شعريّ خاصّ بالشاعر). • أُقدّم التطبيق للشاعر.	• حوار ومناقشة. • تعلّم تعاونيّ/مجموعات.	• حاسوب. • شاشة عرض. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.
أقوّم	• رضا الشاعر عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات.	• حوار ومناقشة.	• طلب الشاعر. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد يهتم بواجهات التطبيق وعناصرها وإنشاء قوائم البيانات. • نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد).

نشاط (1) للاطلاع: أكتشف مع زملائي استخدامات أخرى لممرِّر الصفحات (Scroll view).

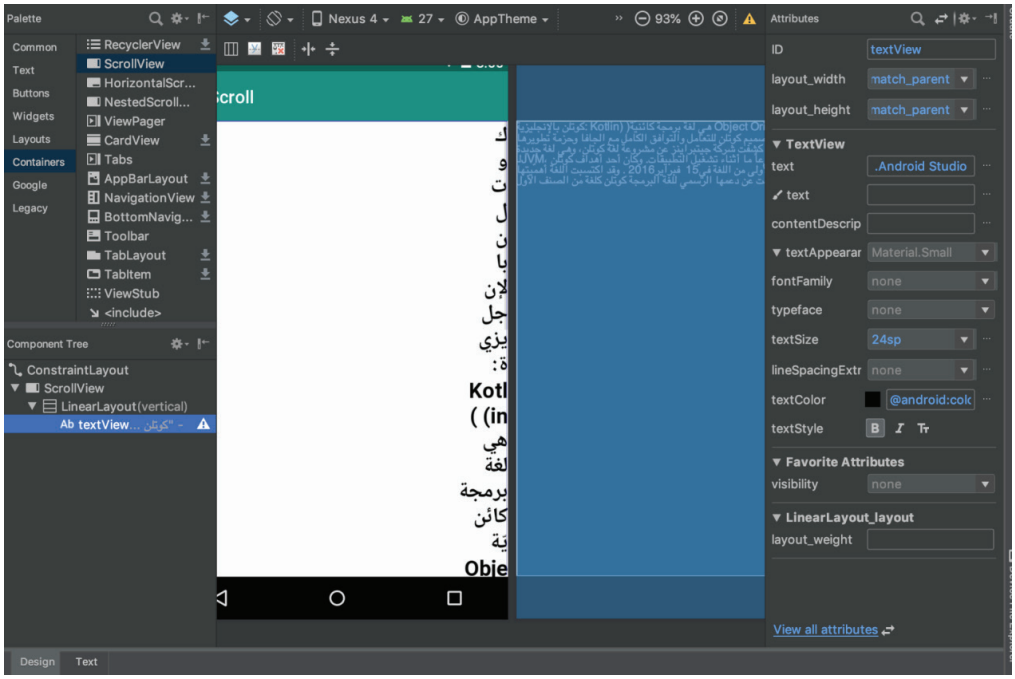
ممرِّر الصفحات (Scroll view):

غالباً ما يُستعمل مع النصوص الطويلة أو النوافذ التي تحتوي على عدّة عناصر، وتكون شاشة الجهاز غير كافية لعرض جميع المحتويات، ويمكن الاستعانة بممرِّر الصفحات للحصول على نافذة قابلة لزيادة الطول أو العرض.

أنواع ممرِّر الصفحات المستخدمة في الأندرويد:

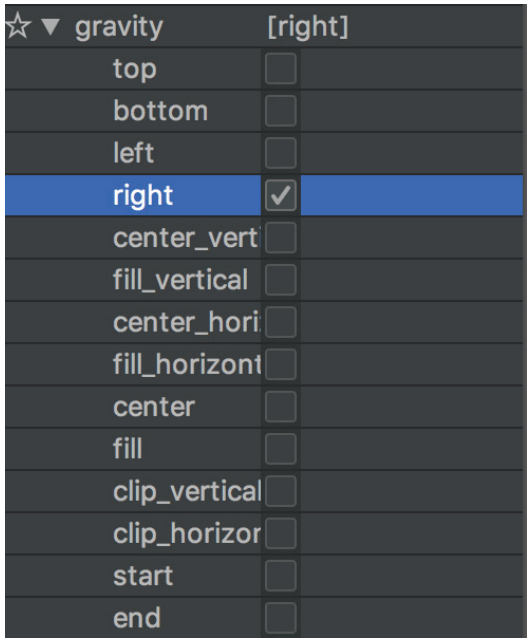
1. ممرِّر الصفحات الافتراضيّ (scrollView): يمرّر محتويات الصفحة بشكل عموديّ من أسفل إلى أعلى وبالعكس، مع ظهور عمود التمرير.
2. ممرّر الصفحات الأفقيّ (HorizontalScrollView): يمرّر محتويات الصفحة بشكلٍ أفقيّ من اليمين إلى اليسار وبالعكس دون ظهور عمود التمرير.
3. ممرّر الصفحات العموديّ المتداخل (NestedScrollView): يمرّر محتويات الصفحة بشكل عموديّ من أسفل إلى أعلى وبالعكس دون ظهور عمود التمرير.

نشاط (2) عملي: أنشئ تطبيقاً يعرض مقالاً عن الكوتلن بشكل منسّق، مستخدماً ممرّر الصفحات الافتراضيّ.
أُنشئ مشروعاً جديداً، وأُضيف ممرّر الصفحات scrollView ويكون بداخله LinearLayout أُضيف داخلها textView، وأُحدّد خصائص مناسبة للنصّ كما تعلّمت سابقاً، ثمّ أُضيف المقال الذي أُريده كما يأتي:



ألاحظ أنّ النصّ في التصميم ليس منسّقاً، لكن عند عمل اختبار للتطبيق يظهر بالشكل الصحيح.

عند إضافة نصّ عربيّ أنتقل إلى خصائص textView، وأضغط على gravity، وأختار right لتنسيق النصّ كالآتي:

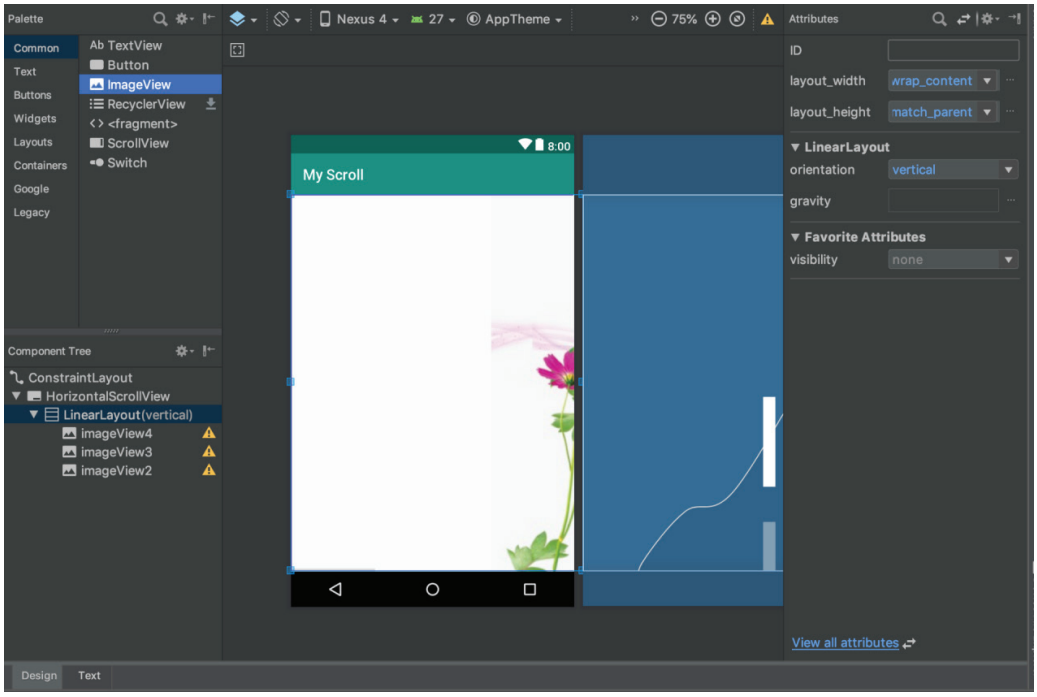


أشغّل المشروع، وأعمل اختباراً للتطبيق، وأفحص النتائج:

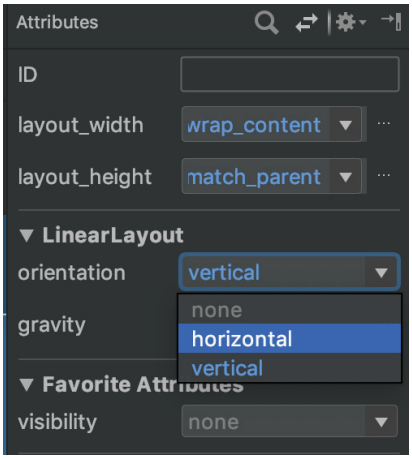
نشاط (3) عملي: أنشئ تطبيقاً يعرض مقالاً عن الأندرويد باستخدام ممرّر الصفحات العموديّ المتداخل (NestedScrollView).

نشاط (4) عملي: أُنشئ تطبيقاً يعرض ثلاث صور مرتّبة بشكل أفقيّ في الواجهة نفسها.

أنشئ مشروعاً جديداً، ثمّ أنسخ الصور الثلاث المراد إضافتها، وأُلصقها في مجلد drawable، أنتقل إلى واجهة التصميم، وأضيف ممرّر الصفحات الأفقيّ (HorizontalScrollView) ويكون بداخله LinearLayout، أُضيف داخلها ثلاثة عناصر imageView وأعطي كلّ عنصر صورة مختلفة، كما يأتي:



أحوّل LinearLayout التي داخل HorizontalScrollView من عموديّ vertical إلى أفقيّ horizontal كما يأتي:



أُشغّل المشروع وأختبر التطبيق:

الأسئلة

1. ما المقصود بممرّر الصفحات؟
2. أوضّح الفرق بين أنواع ممرّر الصفحات؟

أسئلة الوحدة

السؤال الأوّل: أضع دائرة حول رمز الإجابة الصحيحة فيما يأتي:

1. ما نوع القائمة التي تعرض عناصر البيانات بأكثر من عمود؟
   أ. listView
   ب. cardView
   ج. gridView
   د. menu

2. ما وظيفة ArrayAdapter في عمل listView؟
   أ. ربط listView مع مصدر البيانات.
   ب. تعمل كمصفوفة بيانات.
   ج. أمر يشير إلى وجود ضغط على عنصر listView.
   د. لا شيء ممّا ذُكر.

3. ما عمل loadUrl في مستعرض الإنترنت؟
   أ. تفعيل جافا سكربت.
   ب. عرض صفحة الويب.
   ج. صلاحيّات الإنترنت.
   د. إضافة نصّ.

4. ما اسم الملف الذي توضَع فيه ملفّات الصوت والفيديو؟
   أ. ملف res.
   ب. ملف drawable.
   ج. ملف layout.
   د. ملف raw.

5. ما وظيفة mediaPlayer.reset()؟
   أ. تهيئة مشغّل الصوت لإعادته للوضع الافتراضيّ؛ أي لا يحتوي على ملف صوتيّ.
   ب. إعادة تشغيل الملف الصوتيّ.
   ج. تشغيل الفيدبو.
   د. إيقاف تشغيل الصوت.

السؤال الثاني: أُعرّف كلّاً من: قوائم البيانات، ArrayAdapter، مستعرض الإنترنت، الوسائط المتعدّدة.

السؤال الثالث: أشرح باختصار عن ممرّر لصفحات (Scroll view).

السؤال الرابع: أُعدّد بعض الوسائط المتعدّدة المستخدمة في الأندرويد.

السؤال الخامس: أوضّح بالرسم وظيفة layoutInflater.

مشروع الوحدة:

أُنشئ تطبيقاً يحتوي على قائمة بيانات من ضمنها عنصر ينتقل إلى:

1. مشغّل صوتيّ.
2. عارض فيديو.
3. مستعرض ويب.
4. مقالة.

الوَحْدةُ الثّانية
(مواضيع خاصّة بالأندرويد)

أبحث مع زملائي عن مهارات إضافيّة في إنشاء التطبيقات.

يُتوقَّعُ من الطلبة بعد دراسة هذه الوَحْدة، والتفاعل مع أنشطتها أن يكونوا قادرين على امتلاك مهارات إضافيّة في برمجة تطبيقات الاندرويد من خلال تحقيق الآتي:

1. إنشاء fragments.
2. إنشاء مربّع الحوار dialog.
3. إنشاء الإشعارات (notifications).
4. التعامل مع الأخطاء.

1011011100001011101110011101
10010110111000010
0111011100111101

الكفايات المِهْنيّة

الكفايات المهنيّة المُتوقّع من الطلبة امتلاكها بعد الانتهاء من دراسة هذه الوحدة، والتفاعل مع أنشطتها:

أولاً: الكفايات الحرفيّة.

- عمل fragments.
- عمل مربّع الحوار dialog.
- عمل الإشعارات(notifications).
- كيفيّة التعامل مع الأخطاء وإصلاحها.
- تقويم النتائج.

ثانياً: الكفايات الاجتماعيّة والشخصيّة.

- العمل ضمن فريق.
- تقبُّل التغذية الراجعة.
- احترام رأي الآخرين.
- القدرة على إدارة الحوار وتنظيم النقاش.
- مصداقيّة التعامل مع الزبون.
- حفظ خصوصيّة الزبون.
- الاستعداد للاستجابة لطلبات الزبون.
- التحلّي بأخلاقيّات المهنة أثناء العمل.
- التمتُّع بالفكر الريادي أثناء العمل.
- استخلاص النتائج ودقّة الملاحظة.
- الاتّصال والتواصل الفعّال.
- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- الإفادة من مقترحات الآخرين.
- امتلاك مهارة التأمُّل الذاتيّ.
- الاستعداد التام لتقبُّل ملحوظات الزبون وانتقاداته.
- القدرة على الإقناع.

ثالثاً: الكفايات المنهجيّة.

- توظيف التعلُّم التعاونيّ.
- امتلاك مهارة البحث العلميّ، والقدرة على توظيف أساليبه.
- القدرة على التخطيط.
- القدرة على استمطار الأفكار.
- استخدام المصادر والمراجع المختلفة.
- توثيق نتائج العمل وعرضها.
- تقويم النتائج وعرضها.
- توظيف التكنولوجيا والبرامج الإحصائيّة وتمثيل البيانات.
- جمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:

- الالتزام بالجلسة الصحيّة أمام الحاسوب.
- إراحة العينين كلّ ربع ساعة.
- الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة.
- الابتعاد عن الحاسوب كلّ ساعة، والقيام ببعض التمارين الرياضيّة.

2-2 الموقف التعليميّ التعلُّميّ

مربّع الحوار (dialog).

وصف الموقف التعليميّ: استضاف طبيب أطفال، خبيراً من مشغل تطبيقات الهاتف المحمول في مدرسة سعد صايل المهنيّة؛ لمساعدته في إنشاء تطبيق تجريبيّ يقوم بإضافة مرضى، وتخزين بيانات كلّ مريض بشكل مؤقّت وعرضها في قائمة (باستخدام واجهة واحدة).

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلُّم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من ليث عن: - إصدار نظام الأندرويد. - المعلومات التي يريد إدخالها عن المريض. • أجمع البيانات عن: - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - الجمل البرمجيّة المستخدمة.	• حوار ومناقشة. • التعلُّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب الطبيب، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّر	• أصنّف البيانات (نظام الأندرويد المستخدم). • أحدّد: - المعطيات: (الاسم الرباعيّ، العمر، فصيلة الدم، المرض،العلاج). - العناصر (views) المستخدمة. - عدد الواجهات. - الجمل البرمجيّة المستخدمة. • إعداد خطّة للتنفيذ.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• وثائق (نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).

أنفّذ	• اختيار عدد من الواجهات المقترحة القابلة للاستخدام في تطبيق الأندرويد. • أتّفق مع الطبيب على اختيار واجهة التطبيق المناسبة. • أُصمّم واجهة التطبيق . • أكتب الكود البرمجيّ. • أعمل اختباراً للمشروع على جهاز افتراضيّ، أو على جهاز يعمل بنظام أندرويد.	• تعلم تعاوني/مجموعات. • حوار ومناقشة. • عصف ذهنيّ / استمطار الأفكار.	• حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.
أتحقّق	• أتحقّق من ملاءمة اختيار الواجهة المقترحة • للتطبيق. • أتاكّد من ملاءمة التطبيق. • أتحقّق من تنفيذ التطبيق وظهوره على الجهاز • بشكل مناسب.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• الخطّة المعدّة سابقاً. • الطلب الخاص بالطبيب. • حاسوب. • برنامج Android Studio. • جهاز افتراضيّ، أو جهاز يعمل بنظام • الأندرويد.
أوثّق وأقدّم	• أوثّق البيانات التي تتعلّق بـ: - العناصر (views) المستخدمة للإدخال والإخراج. - عدد الواجهات. - الجمل البرمجيّة المستخدمة. • إعداد ملف بالحالة (إنشاء تطبيق تجريبيّ يقوم بإضافة مرضى، وتخزين بيانات كلّ مريض بشكل مؤقّت، وعرضها). • إعداد عرض تقديميّ.	• حوار ومناقشة. • تعلُّم تعاونيّ/ مجموعات عمل.	• حاسوب. • شاشة عرض. • جهاز افتراضيّ، أو جهاز يعمل بنظام • أندرويد.
أقوّم	• رضا الطبيب عن التطبيق. • ملاءمة التطبيق للمعايير والمواصفات.	• حوار ومناقشة.	• طلب الطبيب. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد يهتم بواجهات التطبيق وعناصرها وإنشاء قوائم البيانات.

سؤال أُعدّد بعضاً من استخدامات مربّع الحوار (dialog)؟

مربّع الحوار (dialog):

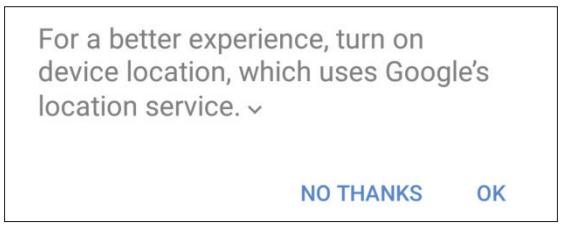
هو نافذة صغيرة تظهر معلومات للمستخدم، وله استخدامات كثيرة، مثل: رسائل التنبيه، أو طلب معيّن، أو إدخال بيانات، وغيرها...

أنواع مربّع الحوار (dialog):

- مربّع حوار عاديّ.
- مربّع حوار مخصّص.

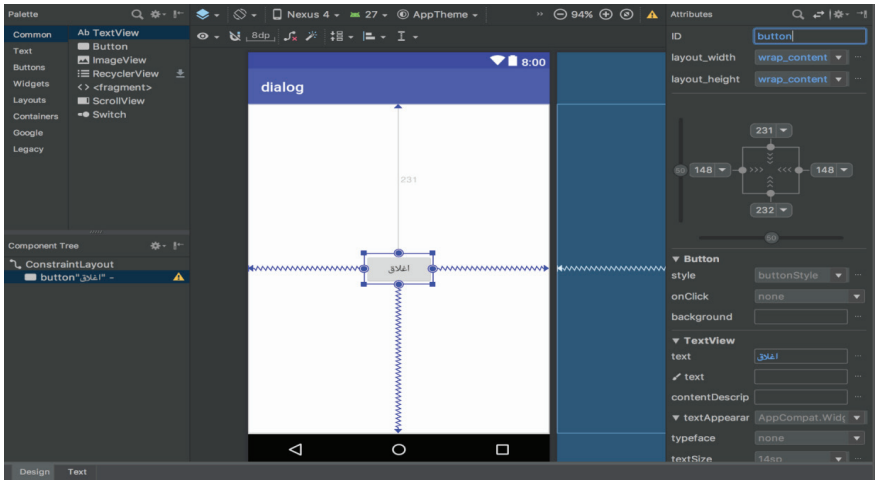
مربّع حوار عاديّ:

شكله افتراضيّ ولا أستطيع تغييره، أضيف إليه عناصر فقط، كما يأتي:



نشاط (1) عملي: أُنشئ مربّع حوار بسيط يظهر رسالة "هل تريد إغلاق التطبيق؟"، يحتوي على زر "إلغاء"، "موافق"، يظهر عند الضغط على عنصر button.

أُنشئ مشروع أندرويد جديد، عندما يفتح البرنامج بشكل كامل أُضيف عنصر button في منتصف واجهة التصميم، كما يأتي:



أنتقل إلى الكود البرمجيّ، وبما أنّ مربّع الحوار (dialog) يظهر عند الضغط على عنصر button، إذن أكتب أوامر إظهار مربّع الحوار داخل عنصر button:

```
button.setOnClickListener { it: View!

    val builder = AlertDialog.Builder( context: this)

    builder.setTitle("تنبيه")
    builder.setMessage("هل تريد اغلاق التطبيق؟")

    builder.setPositiveButton( text: "موافق") { _: DialogInterface, _: Int ->
        finish()
    }

    builder.setNeutralButton( text: "الغاء") { _: DialogInterface, _: Int ->

    }

    val dialog = builder.create()
    dialog.show()

}
```

1. أُنشئ كائناً أسمّيه builder من الصنف AlertDialog والوصول من خلاله إلى الدالّة Builder التي تستقبل this تشير إلى الصنف الحالي.
2. بدأت في بناء مربّع الحوار بإعطائه:
   - عنواناً أسميته ("تنبيه") builder.setTitle.
   - محتوى مربّع الحوار، وكتبت الجملة المطلوبة ("هل تريد إغلاق التطبيق؟")builder.setMassage.
3. أنشأت button إيجابيّاً أعطيته نصّ «موافق»، يحتوي على قيمتين يمكن استخدامهما داخل الدالّة، هنا لم أستخدم أيّاً منهما فوضعت هذه الاشاره (_)، ولا يجوز حذفهما من داخل الدالّة؛ لأنّهما قمّتان رئيسيّتان خاصّتان بعناصر button في مربّع الحوار. كلمه finish() عبارة عن دالّة خاصّة بصنف Activity() مهمّتها إغلاق التطبيق.
4. أنشأت button سلبيّاً لإغلاق مربّع الحوار دون عمل أيّ شيء، وأعطيته اسم "إلغاء"، وطريقة كتابته مثل button الإيجابيّ، ويمكن استخدام أيّ منهما بالطريقة نفسها والأداء نفسه، ولكن سبب تسمية كلّ واحد منهما باسم هو التسهيل على المبرمج في فهم الكود، وإنجاز العمليّات البرمجيّة بشكل أسرع.
5. أعرف متغيّراً اسمه dialog نوعه Dialog، أسندت له builder.create() التي تعني تجميع ما تمّ إنشاؤه في الخطوات السابقة، وتخزينها في مربّع الحوار dialog بالشكل المناسب، ويكون جاهزاً للظهور.
6. dialog.show(): تعني إظهار مربّع الحوار الذي تمّ تجهيزه في الخطوة السابقة.

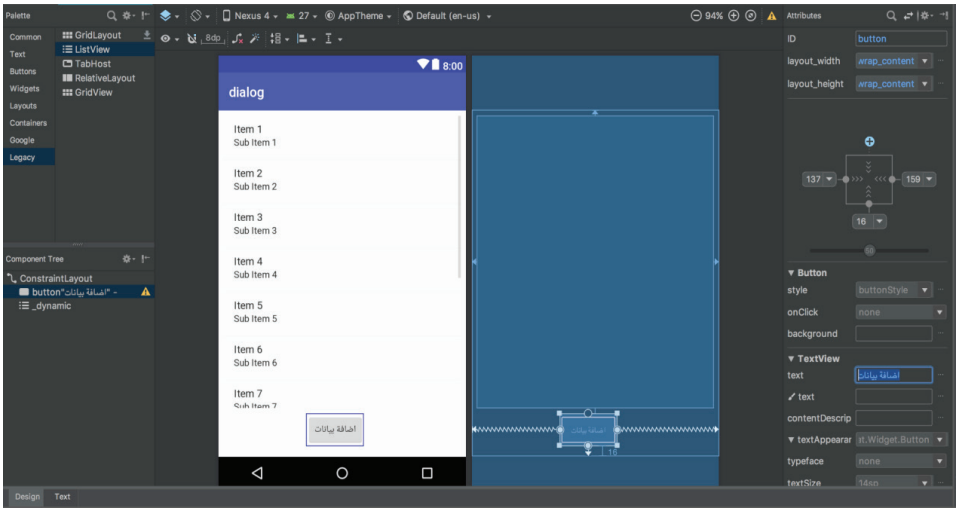
أعمل تشغيلاً للمشروع، ثمّ أختبر التطبيق، وأفحص النتائج:

مربّع حوار مخصّص:

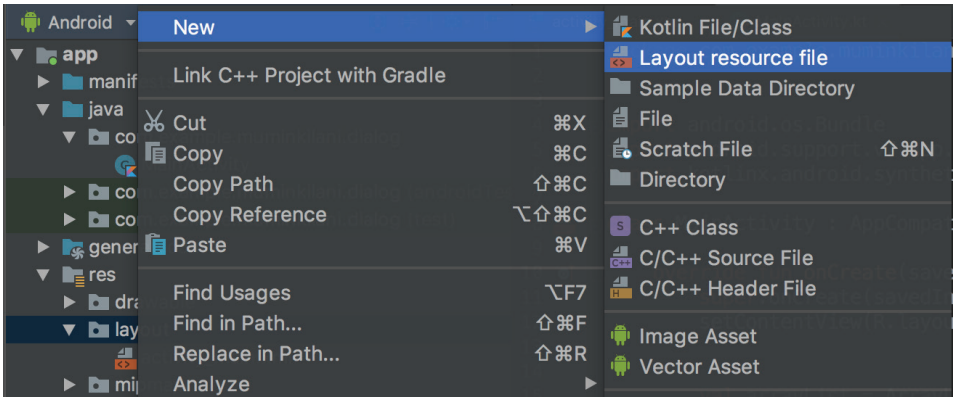
يظهر بالشكل الذي أريده، وأستطيع إضافة عناصر مختلفة بداخله كما يأتي:

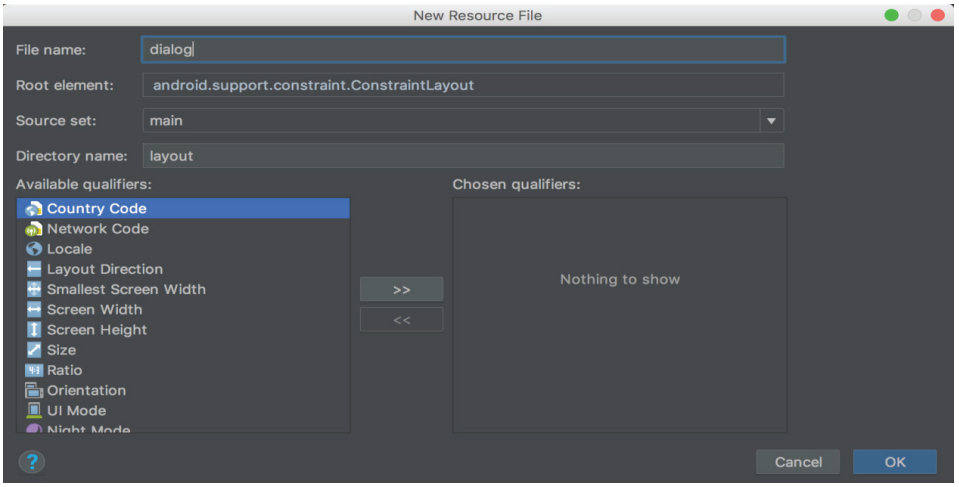
نشاط (2) عملي: أنشئ تطبيقاً يضيف بيانات لقائمة listView بواسطة مربّع الحوار.

أُنشئ مشروعاً جديداً، وعندما يفتح البرنامج بشكل كامل أُضيف العناصر كالآتي:

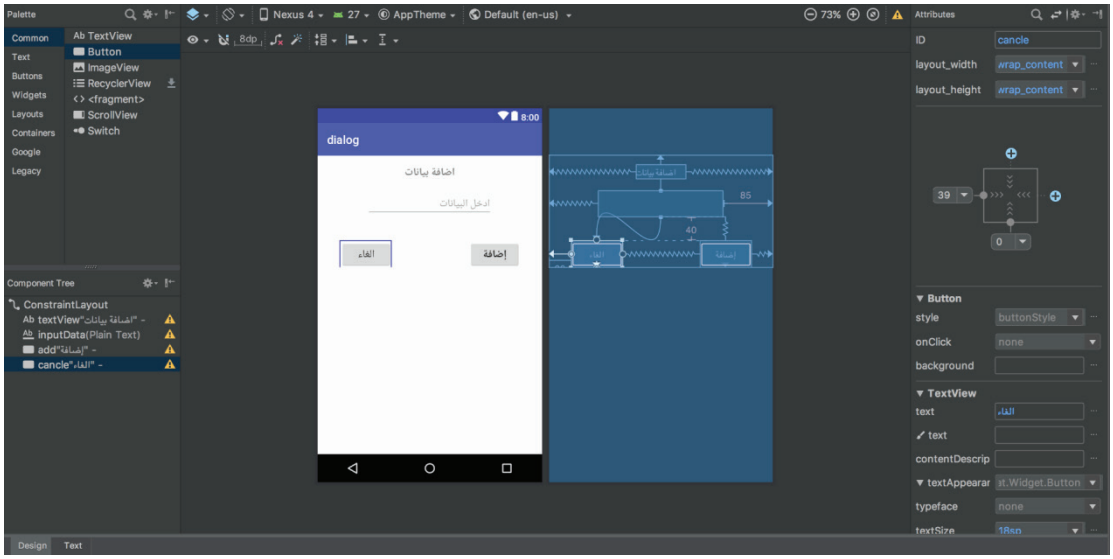


لعمل مربّع حوار (dialog) مخصّص أنشئ ملفّ تصميم واجهة (layout) جديداً وأسمّيه dialog، أضغط بزر الفأرة اليمين على مجلد layout -> new -> layout resourse file:





أضيف داخل واجهة التصميم textView، planText وعنصري button، وأعطي العناصر id مناسباً، يجب مراعاة ألّا يكون حجم مربّع الحوار على حجم الشاشة؛ لكي يظهر بشكل مناسب على التطبيق:



أنتقل إلى الكود البرمجيّ، وأكتب كما يأتي:

```
1- val arrayList = ArrayList<String>()
   button.setOnClickListener { _ ->

2-     val inflater = LayoutInflater.from( context: this)
       val view = inflater.inflate(R.layout.dialog,  root: null)

3-     val builder = AlertDialog.Builder( context: this)
       builder.setView(view)

4-     val dialog = builder.create()
       dialog.show()

5-     view.add.setOnClickListener { it: View!
           val textData = view.inputData.text.toString()
           arrayList.add(textData)

           val myAdapter = ArrayAdapter<String>( context: this, android.R.layout.simple_expandable_list_item_1, arrayList)
           listView.adapter = myAdapter
       }

6-     view.cancle.setOnClickListener { it: View!
           dialog.cancel()
       }

   }
```

استخدمت هذه الاشاره هنا لانه يوجد setOnClickListener متداخلة، يمكن الاستغناء عنها ولكن تظهر اشارة تنبيه تنوه لاضافتها.

1. أنشئ مصفوفة قائمة ArrayList لإضافة البيانات بداخلها، ثمّ أنتقل للكتابة داخل دالّة button الخاص بإظهار مربّع الحوار.
2. ربط ملف الواجهة التصميميّة (dialog.xml) بالكود البرمجيّ، كما تعلّمت سابقاً.
3. أُعرّف builder وأعطيه view، وهي الواجهة (dialog.xml) التي قمت بتصميمها، (builder.setView(view.
4. إنشاء مربّع الحوار وإظهاره.

5. للوصول إلى العنصر button الخاص بالإضافة أكتب view أوّلاً، ثمّ add (وهو id الخاص بالعنصر) داخل دالّة button إضافة، عرّفت متغيّراً من نوع string لجلب البيانات من planText، ثمّ أضفتها بمصفوفة القائمة arrayList، بعدها كتبت الأوامر اللازمة لإظهار محتوى مصفوفة القائمة arrayList على القائمة listView.
6. الأوامر الخاصّة بعنصر الإلغاء، dialog.cancle() تفيد إغلاق مربّع الحوار.

أُشغّل المشروع، ثمّ أختبر التطبيق، وأفحص النتائج:

نشاط (3) عملي: أُنشئ تطبيقاً يظهر مربّع حوارعند فتح التطبيق، يحمل عنواناً ومحتوًى من اختياري.

نشاط (4) عملي: أضيف الأمر اللازم عند الضغط على button إضافة في النشاط (3) يعمل على إغلاق مربّع الحوار بعد عمليّة الإضافة.

الأسئلة

1. ما هو مربع الحوار dialog؟
2. ما وظيفة الدال show() في عمل مربّع الحوار dialog.

أسئلة الوحدة

السؤال الأوّل: أختار رمز الإجابة الصحيحة فيما يأتي:

1. ما تصنيف الخطأ البرمجي الآتي؟

```
textView.text= 1234
```

أ. خطأ وقت تشغيل المشروع.
ب. خطأ منطقي.
ج. خطأ أثناء الكتابة.
د. لا شيء ممّا ذكر.

2. ما فائدة setContentText() في بناء الإشعارات؟

أ. كتابة عنوان للإشعار.
ب. كتابه محتوى الإشعار.
ج. إضافة صورة للإشعار.
د. اسم قناة الإشعار.

3. ما وظيفة onAttach ؟

أ. ربط الواجهة fragment بالواجهة الرئيسيّة activity.
ب. تهيئة الواجهة fragment للظهور على الواجهة الرئيسيّة activity.
ج. يبدأ عملها عند انتهاء عمل الدالّة onCreate.
د. تعمل عندما تكون الfragment مرئيّة.

4. ما هي commit؟

أ. دالة مسؤولة عن التبديل بين الواجهات fragments.
ب. تأكيد عمليّة تبديل الواجهات frsgments.
ج. تحذف الواجهة fragment القديمه.
د. لا شيء ممّا ذكر.

5. ماذا يحدث عندما ينفذ السطر البرمجيّ الآتي dialog.cancle()؟

أ. يظهر مربّع الحوار dialog.
ب. يتمّ إغلاق مربّع الحوار dialog.
ج. إضافة زر button في مربّع الحوار dialog.
د. لا يفيد بعمل أيّ شيء.

السؤال الثاني: أوضّح بالرسم دورة حياة الواجهة fragment.

السؤال الثالث: أعرّف ما يأتي:

- مربّع الحوار dialog.
- الواجهة fragment.
- الإشعارات notifications.
- القنوات.

السؤال الرابع: أوضّح الفرق بين الخطأ البرمجيّ والخطأ غير البرمجيّ.

مشروع الوحدة:

أُنشئ تطبيقاً يحتوي على واجهتين fragment، إحداهما تفيد في حساب مساحة مربّع، ويظهر النتيجة في مربّع حوار dialog مخصّص، والأخرى تفيد في حساب مساحة دائرة وتظهر النتيجة في إشعار notification.

الوَحْدةُ الثّالثة

قواعد بيانات SQLite في تطبيقات الهاتف المحمول

أُناقش: يتمّ تخزين البيانات بأكثر من طريقة، وفي بعض الحالات يُفضَّل استخدام قواعد البيانات لعمليّة التخزين.

يُتوقَّعُ من الطّلبة بعد دراسة هذه الوَحْدةِ، والتّفاعُلِ مع أنشطتها أن يكونوا قادرين على بناء تطبيق باستخدام SQLite من خلال ما يأتي:

1. التعرُّف إلى مفهوم قواعد البيانات العلائقيّة وعناصره المختلفة.
2. بناء قواعد البيانات ومعالجتها باستخدام نظام SQLite.

10110111000010111011100111010010110111000010

الكفايات المِهْنيّة

الكفاياتُ المِهْنيّةُ المُتوَقَّعُ من الطّلبة امتلاكُها بعد الانتهاء من دراسة هذه الوَحدةُ والتّفاعُل معَ أنشطتِها:

أولاً: الكفايات الحرفيّة.

القدرة على:

- تحليل قاعدة البيانات وتصميمها.
- بناء قاعدة بيانات تلبّي حاجة الزبون.
- استرجاع المعلومات المطلوبة من قاعدة البيانات.
- تحديث قاعدة البيانات.
- تقويم النتائج.

ثانياً: الكفايات الاجتماعيّة والشخصيّة.

- العمل ضمن فريق.
- تقبُّل التغذية الراجعة.
- احترام رأي الآخرين.
- القدرة على إدارة الحوار وتنظيم النقاش.
- مصداقيّة التّعامُل مع الزبون.
- حِفْظُ خصوصيّة الزبون.
- الاستعداد للاستجابة لطلبات الزبون.
- التحلّي بأخلاقيّات المهنة أثناء العمل.
- التمتُّع بالفكر الريادي أثناء العمل.
- استخلاص النتائج ودقّة الملاحظة.
- الاتصال والتواصل الفعّال.
- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- الإفادة من مقترحات الآخرين.
- امتلاك مهارة التأمُّل الذاتيّ.
- الاستعداد التام لتقبُّل ملحوظات الزبون وانتقاداته.
- القدرة على اقتراح البدائل والإقناع.

ثالثاً: الكفايات المنهجيّة.

- توظيف التعلُّم التعاونيّ.
- القدرة على إدارة الحوار وتنظيم النقاش.
- امتلاك مهارة البحث العلميّ، والقدرة على توظيف أساليبه.
- القدرة على التخطيط.
- القدرة على استمطار الأفكار.
- استخدام المصادر والمراجع المختلفة.
- توثيق نتائج العمل وعرضها.
- تقويم النتائج وعرضها.
- توظيف التكنولوجيا والبرامج الإحصائيّة وتمثيل البيانات.
- جمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:

- تجنُّب الأكل والشرب أمام الحاسوب.
- الجلوس بوضعيّة سليمة أمام الحاسوب؛ وذلك لإراحة الظهر.
- التأكُّد من عدم وجود أسلاك متعريّة.
- إبعاد النظرعن الشاشة بشكل دوريّ؛ لتجنُّب ألم وإجهاد العينين.
- تجنُّب الجلوس أمام الحاسوب لفترات طويلة.
- التأكُّد من وجود تهوية جيّدة وإضاءة مناسبة.

1-3 الموقف التعليميّ التعلُّميّ

مفهوم قواعد البيانات العلائقيّة وعناصرها المختلفة.

وصف الموقف التعليميّ: يرغب طلبة إحدى المدارس الأكاديميّة في إعداد مطويّة عن مفهوم قواعد البيانات العلائقيّة وعناصرها المختلفة؛ فزاروا شركة للتكنولوجيا للاستفسار عن مضمون المطويّة.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلّم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من طلبة المدرسة الأكاديميّة حول: - معرفتهم السابقة بمفهوم قواعد البيانات العلائقيّة وعناصره المختلفة. - محتوى المطويّة. - مواصفات المطويّة (عدد الصفحات، الصور، روابط تعليميّة...). • جمع البيانات عن : مفهوم قواعد البيانات العلائقيّة. - عناصر قواعد البيانات العلائقيّة (الجداول، الحقول، السجلّات، المفاتيح، العلاقات). - التكامل المرجعيّ.	• حوار ومناقشة. • التعلُّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب المدرسة الزائرة، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد، مجلّات عن قواعد البيانات). • التكنولوجيا (حاسوب، شاشة عرض، إنترنت).
أخطّط وأقرّر	• أُحدّد ما يأتي: - مفهوم قاعدة البيانات العلائقيّة. - مفهوم الجداول. عناصر الجدول: الحقول. السجلّات. - مفهوم المفاتيح. أنواع المفاتيح: المفتاح الرئيسيّ. المفتاح الثانويّ. المفتاح الأجنبيّ.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• وثائق (طلب المدرسة الزائرة، مكتبات، مجلّات حول قواعد البيانات).

	- مفهوم العلاقات. أنواع العلاقات: علاقة واحد إلى واحد. علاقة واحد إلى متعدّد. علاقة متعدّد إلى واحد. علاقة متعدّد إلى متعدّد. - مفهوم التكامل المرجعيّ. • إعداد خطّة العمل التي تتضمّن موعد تسليم المطويّة للزائرين.		
أُنفّذ	• كتابة مفهوم قواعد البيانات العلائقيّة. • كتابة مفهوم الجدول. • إنشاء جدول للمقارنة بين عناصر الجدول (الحقول والسجلّات). • كتابة مفهوم المفاتيح. • إنشاء جدول للمقارنة بين أنواع المفاتيح المختلفة. • كتابة مفهوم العلاقات. • إنشاء جدول للمقارنة بين أنواع العلاقات المختلفة. • كتابة مفهوم التكامل المرجعيّ. • إعداد المطويّة بحلّتها الأوّليّة.	• تعلُّم تعاونيّ/ مجموعات. • حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار.	• إنترنت، حاسوب. • البيانات التي تمّ جمعُها. • قرطاسيّة. • - الطلب الخاصّ بالزائرين.
أتحقّق	• التحقّق ممّا يأتي: تحديد مفهوم قواعد البيانات العلائقيّة. إجراء المقارنة بين عناصر الجدول (الحقول والسجلّات). تحديد مفهوم المفاتيح. إجراء المقارنة بين انواع المفاتيح المختلفة. تحديد مفهوم العلاقات. إجراء المقارنة بين أنواع العلاقات المختلفة. تحديد مفهوم التكامل المرجعيّ.	• تعلم تعاونيّ/ مجموعات. • حوار ومناقشة.	• البيانات التي تمّ جمعُها. • إنترنت، حاسوب. • الخطّة المعدّة سابقاً.

أوثّق وأقدّم	• توثيق البيانات التي تتعلّق بِـ: - مفهوم قواعد البيانات العلائقيّة. - عناصر قواعد البيانات العلائقيّة: (الجداول، والحقول، والسجلّات، والمفاتيح، والعلاقات). - التكامل المرجعيّ. • إعداد ملفّ بالحالة (طلب المدرسة الأكاديميّة). • إعداد عرض تقديميّ. • إعداد ملخّص يتضمّن مفهوم قواعد البيانات العلائقيّة، وعناصره ودور كلّ عنصر في قاعدة البيانات.	• حوار ومناقشة. • تعلُّم تعاونيّ/ مجموعات عمل.	• البيانات التي تمّ جمعُها سابقاً. • حاسوب. • شاشة عرض. • قرطاسيّة.
أقوّم	• رضا المدرسة الزائرة عن المطويّة. • ملاءمة المطوية للمواصفات والمعايير الخاصة بقواعد البيانات العلائقية وعناصرها.	• حوار ومناقشة.	• طلب المدرسة الزائرة. • نشرات للمواصفات والمعايير الخاصة بقواعد البيانات العلائقية وعناصرها.

أناقش: يجب أن يحتوي كلّ جدول على مفتاح رئيسيّ.

أتعلّم: مفهوم قواعد البيانات العلائقيّة وعناصره المختلفة

نشاط (1) للاطلاع: كان نظام الملفّات مستخدماً قديماً قبل ظهور قواعد البيانات، أبحث عن آليّة عمل هذا النظام، ومميّزاته وعيوبه، وأُناقش ذلك مع زملائي.

قواعد البيانات Database:

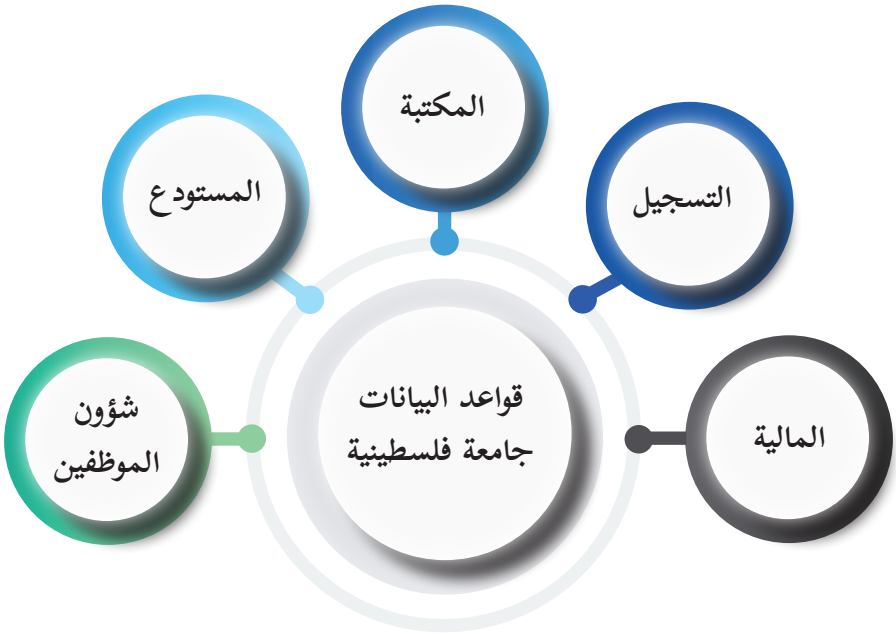
عبارة عن مجموعة من البيانات المنظّمة والمرتبطة بموضوع معيّن داخل جدول أو مجموعة من الجداول؛ بهدف استرجاعها لاتّخاذ القرارات المطلوبة.

من الأمثلة على قواعد البيانات ما يأتي:

- القبول والتسجيل الجامعيّ في المعاهد والجامعات.
- تسجيل ومحاسبة المرضى في المستشفيات العامّة أو الخاصّة.
- المبيعات والمشتريات في الشركات التجاريّة والمصانع وغيرها.
- إدارة المخازن والمستودعات في مؤسّسة ما.
- التأمين بأنواعه (التأمين الصحيّ، وتأمين السيارات، وتأمين البضائع).
- إدارة خدمات البلديّة (مياه، كهرباء، صرف صحيّ، ....الخ) .
- إدارة المكتبات العامّة والخاصّة (إعارة الكتب، بيع الكتب وشراؤها).
- العمليّات المصرفيّة في المصارف والبنوك ...الخ.

نشاط (2) للاطلاع: أبحث عن أمثلة إضافيّة لقواعد البيانات، وأناقشها مع زملائي.

الشكل (1-4) يوضّح بنية قواعد البيانات، حيث تتمّ مشاركة البيانات بين مستخدمي قاعدة البيانات كافّة، وبالتالي فإنّ أيّ تعديل يتمّ من قِبَل إحدى المستخدمين يكون مرئيّاً لدى بقية المستخدمين.



شكل (1-4): بنية قاعدة بيانات جامعة فلسطينيّة

قواعد البيانات العلائقيّة:

عبارة عن مجموعة من الجداول تربط بينها علاقات، وكلّ جدول يشتمل على صفوف (سجلّات)، وأعمدة (حقول).

من خصائص قواعد البيانات العلائقيّة ما يأتي:

1. تستخدم الجداول لتمثيل البيانات.
2. تستخدم أنواع العلاقات كافّة لربط الجداول فيما بينها، وهي:
   - علاقة واحد إلى واحد.
   - علاقة واحد إلى متعدّد.
   - علاقة متعدّد إلى واحد.
   - علاقة متعدّد إلى متعدّد.
3. يحتوي كلّ جدول على مفتاح رئيسيّ، وهو صفّة مميّزة (عمود أو حقل)، أو مجموعة من الصفات تستطيع تحديد كلّ سجلّ في الجدول بشكل مميّز.
4. ترتيب الصفوف والأعمدة غير مهمّ في الجدول.
5. القِيم المخزّنة في الحقول بسيطة وغير مركّبة، مثل: أن يخزّن مكان الميلاد في حقل وتاريخ الميلاد في حقل آخر منفصل، ولا يمكن تخزين الحقلين في حقل واحد معاً.
6. لا يسمح بوجود تكرار في السجلّات.

أنظمة إدارة قواعد البيانات العلائقيّة Database management system (DBMS):

هي مجموعة من البرامج التي تُمكِّن المستخدم من التعامل مع قاعدة البيانات من حيث تعريفها وإنشاؤها وتمثيلها وتعديلها.

يتم التعامل مع البيانات من خلال تعريفها، ووضع القيود عليها، وحذفها، وفرزها، وتخزينها، واسترجاعها بطرق سهلة، ومن أشهرها: SQLite، SQL Server، MySQL، Oracle، MS Access.

ملحوظة: تتميّز نظم إدارة قواعد البيانات العلائقيّة بقوّتها، وسهولة تصميمها و برمجتها، وتعامل المستخدمين معها؛ لذلك يتمّ استخدامها بشكل واسع.

مثال: قاعدة بيانات المستشفى

جدول المرضى

اسم المريض	رقم المريض	رقم الغرفة	الجنس	الطبيب
حسان	313	100	1	سيف
حنان	345	300	2	حسان
خالد	988	100	1	دعاء
منى	456	200	2	عزة

جدول الغرف

رقم الغرفة	عدد الأسرة
100	3
200	2
300	1

جدول الأدوية

رقم الدواء	اسم الدواء	المصنع
s123	FDG	HG
s153	PANADOL	AB
s173	FIFA	AB

جدول العلاج

رقم المريض	رقم الدواء	الكمية
313	s123	3
313	s153	2
988	s123	1

شكل (2-4): قاعدة بيانات مستشفى

قاعدة بيانات المستشفى في الشكل (2-4)، تحتوي على أكثر من جدول:

1. جدول المرضى.
2. جدول الغرف.
3. جدول الأدوية.
4. جدول العلاج.

تربط هذه الجداول علاقات، تساعد الحاسب في استرجاع استعلام معيّن من داخل قاعدة البيانات.

مثال: مريض يحمل الرقم (313)، للاستعلام عن بياناته (الاسم، رقم الغرفة، اسم الدواء)، كالآتي:

- أولاً: من جدول المرضى، باستخدام رقم المريض يُستخرج اسم المريض والغرفة التي يرقد فيها (اسم المريض حسان، رقم الغرفة 100).
- ثانياً: من جدول العلاج، باستخدام رقم المريض تُستخرج أرقام الأدوية التي يتناولها المريض (s123، s153).
- ثالثاً: من جدول الأدوية باستخدام أرقام الأدوية تُستخرج أسماء هذه الأدوية (PANADOL، FDG).

عناصر قواعد البيانات العلائقيّة:

1. الجداول:

في قواعد البيانات العلائقيّة يتمّ تجميع البيانات وتخزينها داخل جداول ثنائيّة الأبعاد.

يتكوّن الجدول من:

- أعمدة : تُستخدَم لإظهار صفات البيانات أو خصائصها، وهي تمثّل حقول الجدول.
- صفوف: تمثّل السجلّات، وكلّ سجلّ يحتوي على بيانات خاصّة بعنصر واحد .

مثال: جدول المرضى

اسم المريض	رقم المريض	رقم الغرفة	الجنس	الطبيب
حسان	313	100	1	سيف
حنان	345	300	2	حسان
خالد	988	100	1	دعاء
منى	456	200	2	عزة

شكل (3-4): جدول المرضى

عند إنشاء جدول المرضى وإدخال بيانات المريض الأوّل: اسمه حسّان، ورقمه 313، ورقم غرفته 100، وطبيبه المعالِج سيف. تُسمّى هذه البيانات الممثّلة لمريض واحد (صفّ واحد) سجلّاً (Record)، بينما خانة البيانات المفردة المكوّنة للسجلّ تسمّى حقلاً (Field)؛ اسم المريض الأوّل حسّان هو قيمة حقل (اسم المريض)، والقيمة 313 هي قيمة حقل (رقم المريض) وهكذا.

الاستنتاج: عدد السجلّات (المرضى) في جدول المرضى 4 سجلّات، وكلّ سجلّ يتكوّن من الحقول (البيانات) الآتية: اسم المريض، ورقم المريض، ورقم الغرفة، والجنس، والطبيب.

2. المِفتاح Key:

هو حقل أو أكثر من حقول الجدول، يُستخدَم في عمليّات البحث (الاستعلام) والمعالجة، مثل: الإضافة، والتعديل، والحذف.

تُقسم المفاتيح إلى نوعين:

- **من حيث تكرار القيم:**

- المفتاح الرئيسيّ (Primary Key): هو حقل (عمود) في جدول قاعدة البيانات يتضمّن قيماً استثنائيّة مميّزة، لا يجوز تكرارها في صفوف (سجلّات) هذا العمود. ويُستخدَم هذا المفتاح في عمليّة البحث للوصول إلى صفّ معيّن في الجدول. فمثلاً: حقل رقم المريض هو مفتاح رئيسيّ في جدول المرضى، لكلّ مريض رقم خاصّ به يميّزه عن المرضى الآخرين، (انظر إلى شكل 2-4).

> ملحوظة: يُستخدم المِفتاح الرئيسيّ لاسترجاع بيانات سجلّ واحد فقط؛ فلو أردنا استرجاع بيانات مريض معيّن فإنّنا نبحث عنه باستخدام رقمه وليس باستخدام حقول أخرى، وإذا استخدمنا رقم الغرفة في جدول المرضى فإنّه يقوم بإرجاع بيانات المرضى كافّة، المقيمين في الغرفة نفسها، وليس مريضاً بعينه.

- المِفتاح الأجنبيّ: (Foreign Key) وهو حقل يُستخدَم للربط بين جدولين، فيكون مفتاحاً رئيسيّاً في الجدول الأساسيّ ومفتاحاً أجنبيّاً في الجدول التابع، وقِيم حقل المفتاح الأجنبيّ في الجدول التابع تتكرّر بحيث تكون بعض القيم أو كلّها غير مميّزة، مثلاً: رقم الغرفة في جدول المرضى يُعدُّ مفتاحاً أجنبيّاً لجدول الغرف، (انظر إلى شكل 2-4).
- المِفتاح الثانويّ Secondary Key: وهو حقل تتكرّر فيه قيم البيانات، ويُستخدم لغايات استرجاع البيانات، فمثلاً جدول المرضى يحتوي على قائمة بالمرضى والمِفتاح الرئيسيّ هو رقم المريض، ولكن إذا نُسِيَ رقم المريض، في هذه الحالة يُستخدم اسم المريض في عمليّة البحث وليس رقمه لاسترجاع بياناته؛ لذلك يتمّ اختيار اسم المريض كمفتاح ثانويّ،(انظر إلى شكل 2-4).

- **من حيث التركيب:**

- المِفتاح البسيط Simple Key: وهو مفتاح يتكوّن من حقل واحد فقط من حقول الجدول، مثل: رقم المريض في جدول المرضى، ورقم الهويّة في جدول المواطنين، ورقم المساق في جدول المساقات...الخ.
- المِفتاح المركّب Compound Key: هو مفتاح يتكوّن من حقلين أو أكثر من حقول الجدول.

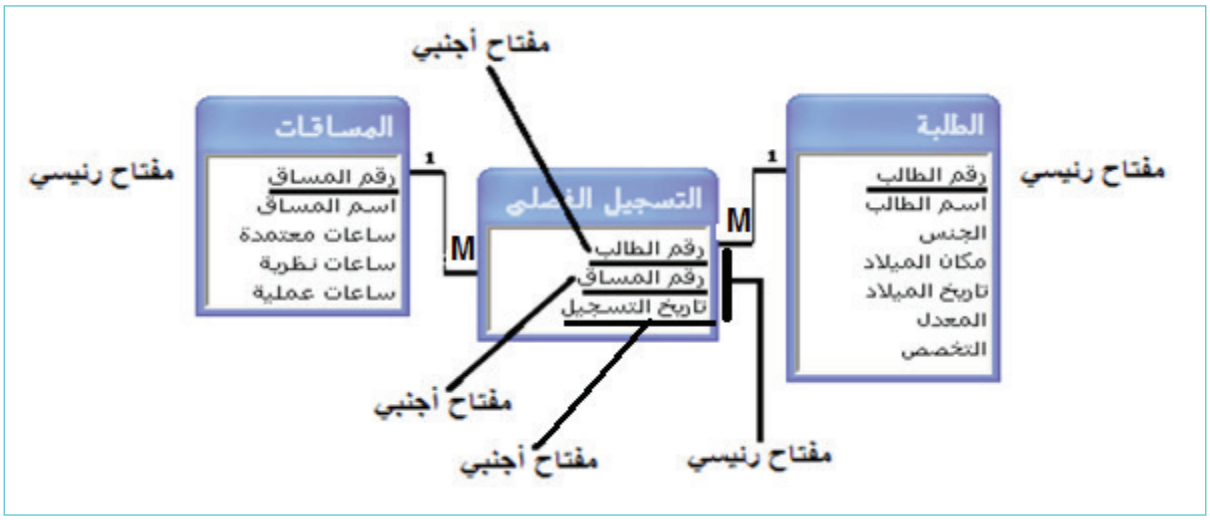
أمثلة:

- (رقم المريض، رقم الدواء) في جدول العلاج، (انظر إلى شكل 2-4).
- (رقم الطالب، رقم المساق، تاريخ التسجيل) في جدول التسجيل الفصليّ، (انظر إلى شكل 2-4).

- (رقم الموظّف، السنة، الشهر) في جدول الرواتب الشهريّة للموظّفين.
- (رقم الطالب، رقم المساق، السنة الدراسيّة، الفصل الدراسيّ) في جدول العلامات.

بناء على ما سبق تُصنَّف أنواع المفاتيح كما يأتي:

- **مفتاح رئيسيّ بسيط:** لا تتكرّر قيم البيانات فيه، ويتكوّن من حقل واحد، مثل: رقم المريض في جدول المرضى، (انظر إلى شكل 2-4).
- **مفتاح رئيسيّ مركّب:** لا تتكرّر قيم البيانات فيه، ويتكوّن من حقلين فأكثر، مثل: (رقم المريض، رقم الدواء) في جدول العلاج، (انظر إلى شكل 2-4).
- **مفتاح ثانويّ أو أجنبيّ بسيط:** تتكرّر قيم البيانات فيه، ويتكوّن من حقل واحد، مثل: رقم الغرفة في جدول المرضى، (انظر إلى شكل 2-4).
- **مفتاح ثانويّ أو أجنبيّ مركّب:** تتكرّر قيم البيانات فيه، ويتكوّن من حقلين فأكثر.



شكل (4-4): قاعدة بيانات خاصّة بالمساقات التي تمّ تسجيلها من قِبَل الطالب

نشاط (3) نظري: بناء على قاعدة بيانات المستشفى (الممثَّلة في الشكل 4-4) أكمل ما يأتي:

- حقل رقم الدواء يُعدُّ مفتاحاً أجنبيّاً في جدول _____، ومفتاحاً رئيسيّاً في جدول_____.
- حقل _____ يُعدُّ مفتاحاً اجنبيّاً في جدول العلاج، ومفتاحاً رئيسيّاً في جدول المرضى.
- حقل رقم الغرفة في جدول _____ يُعدُّ مفتاحاً أجنبيّاً لجدول _____.

3. العلاقات (Relations):

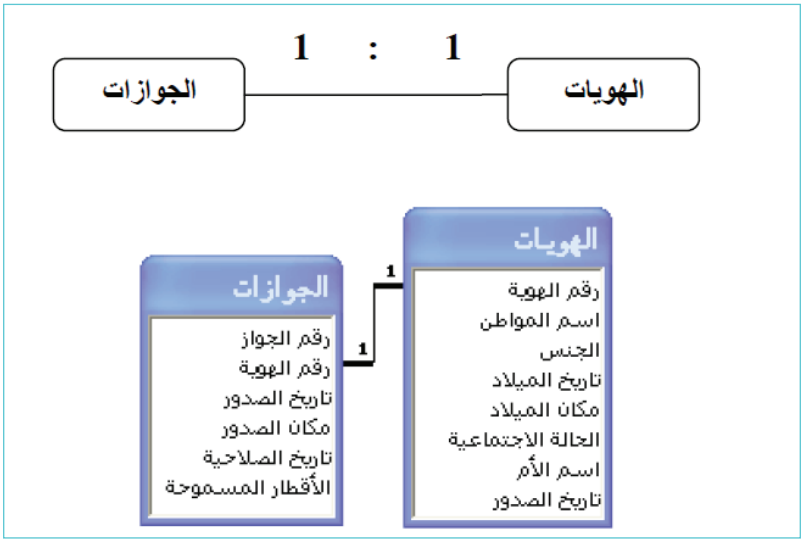
- العلاقة: هي الربط بين الجداول باستخدام حقل المفتاح الرئيسيّ primary key في الجدول الأول (الأساسيّ)، وحقل المفتاح الأجنبيّ foreign key في الجدول الثاني (التابع).
- الفائدة من العلاقات: تُستخدَم العلاقات لتجنُّب تكرار البيانات نفسها في عدة جداول أكثر من مرّة؛ لذلك يتمّ ربط الجداول للاستفادة من هذه البيانات بواسطة العلاقات.

- **أنواع العلاقات:**

أوّلاً: واحد إلى واحد One to One:

كلّ سجلّ في الجدول الأول يرتبط مع سجلّ واحد من الجدول الثاني.
وكلّ سجلّ من الجدول الثاني يرتبط مع سجلّ واحد من الجدول الأول.
ويُرمَز لها بـ: 1:1

مثال: المواطن الفلسطينيّ الواحد له هُويّة فلسطينيّة واحدة، وله جواز فلسطينيّ واحد فقط.
ولا يجوز أن يشترك أكثر من مواطن واحد في هُويّة واحدة، أو جواز سفر واحد أبداً.

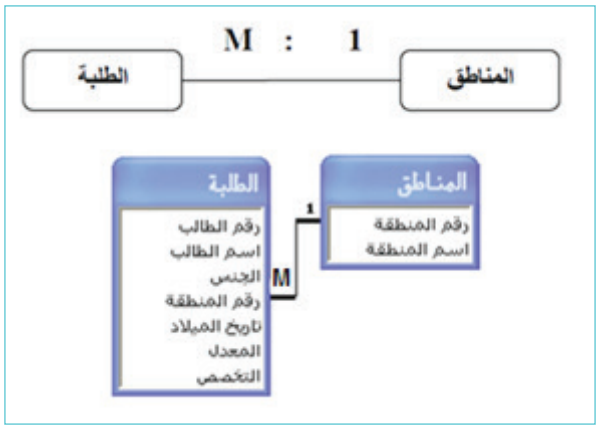


شكل 5-4: العلاقة بين الهويّات والجوازات

ثانياً: واحد إلى متعدّد One to Many

كلّ سجلّ في الجدول الأول يرتبط بعدّة سجلّات في الجدول الثاني، وكلّ سجلّ في الجدول الثاني يرتبط بسجلّ واحد فقط في الجدول الأول .
يُرمَز لها بـ : M:1

مثال: المنطقة الواحدة يولد فيها أكثر من طالب واحد في آن واحد.
والطالب الواحد يولد في منطقة واحدة فقط.



شكل 6-4: نوع العلاقة بين المناطق والطلبة

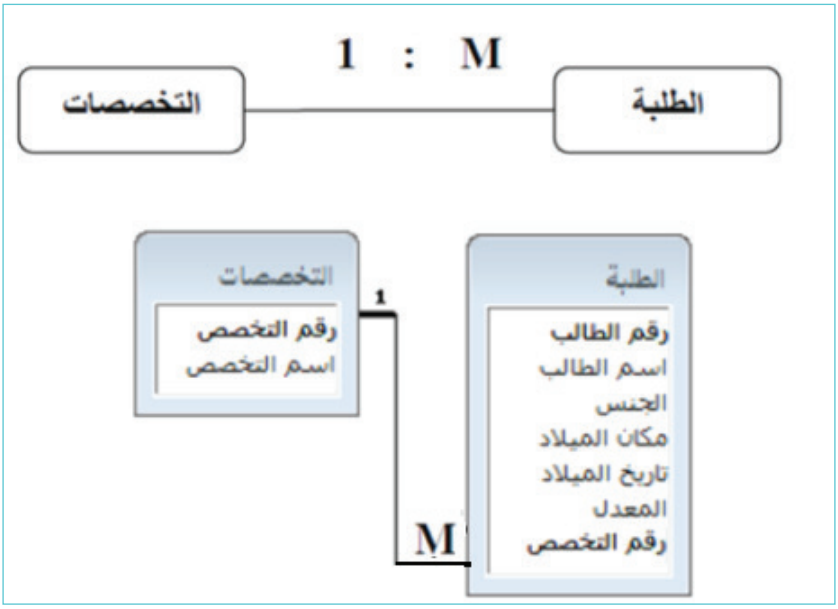
ثالثاً: متعدّد إلى واحد Many to One

كلّ سجلّ في الجدول الأول يرتبط بسجلّ واحد فقط في الجدول الثاني، وكلّ سجلّ في الجدول الثاني يرتبط بعدّة سجلّات في الجدول الأول.

يُرمز لها بـ : M:1

مثال: الطالب الواحد يدرس في تخصُّص واحد فقط.

والتخصّص الواحد يدرس فيه أكثر من طالب.



شكل 7-4: نوع العلاقة بين الطلبة والتخصصات

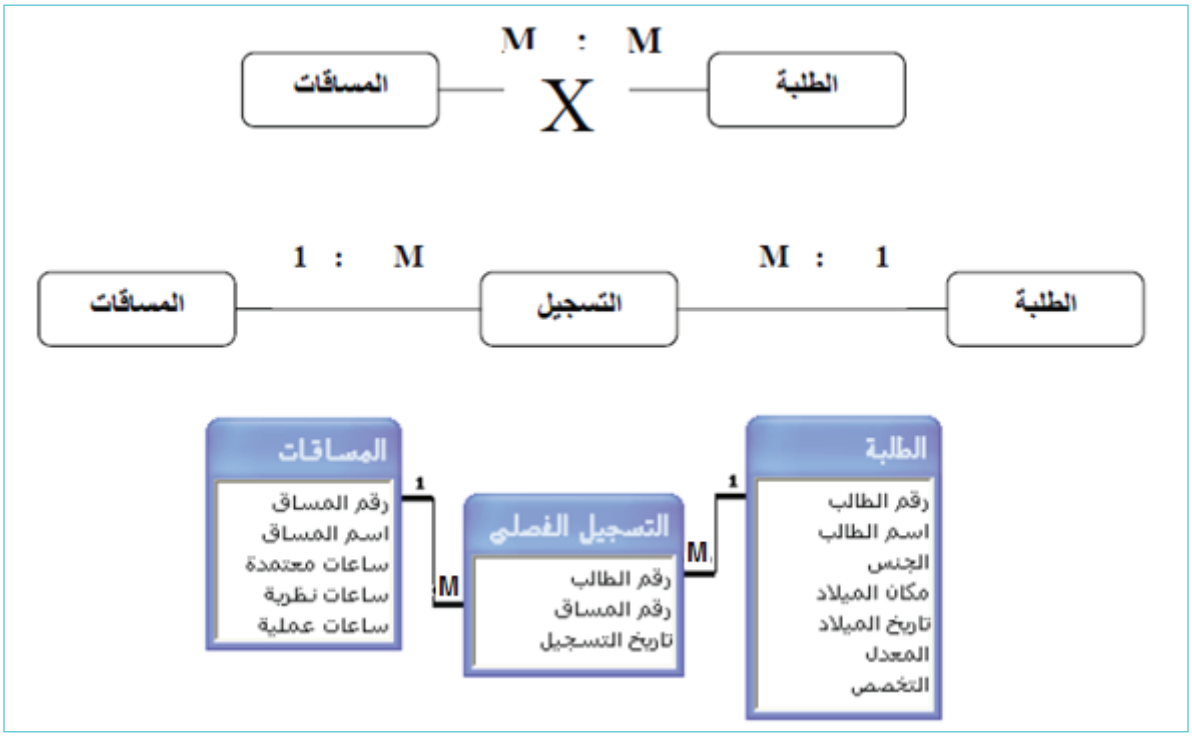
رابعاً: متعدّد إلى متعدّد Many to Many

كلّ سجلّ في الجدول الأول مرتبط بعدّة سجلذات في الجدول الثاني، وكلّ سجلّ في الجدول الثاني مرتبط بعدّة سجلّات في الجدول الأول .

يُرمز لها بـ: M:M

مثال: الطالب الواحد يسجّل اكثر من مساق واحد في آن واحد.

والمساق الواحد يسجّل فيه أكثر من طالب واحد في آن واحد.



شكل 8-4: نوع العلاقة بين الطلبة والمساقات

يوضّح الشكل (8-4) أنّ العلاقة بين الطلبة والمساقات علاقة متعدّد إلى متعدّد، وهي علاقة غير مرغوب فيها؛ لذلك تتمّ معالجتها بوضع جدول وسيط (التسجيل الفصليّ) يحتوي على المفتاح الرئيسيّ من كِلا الجدولين، وبالتالي تتحوّل العلاقة بين الجداول الأساسيّة والجدول الوسيط إلى علاقة واحد إلى متعدّد.

نشاط (4) نظري: أرسم العلاقات بين الجداول في قاعدة بيانات المستشفى مع بيان نوعها.

4. التكامل المرجعيّ: Referential Integrity

مجموعة من القواعد التي يتمّ تطبيقها على الجداول المرتبطة لضمان تطابق البيانات، وعدم حدوث أخطاء فيها.
القواعد التي يفرضها التكامل المرجعيّ على العلاقة:

1. عدم السماح بإضافة سجلّ جديد في حال عدم تطابق قيمة المفتاح الاجنبيّ مع قيمة موجودة ضمن المفتاح الرئيسيّ في الجدول المرتبط معه.

   مثال: عدم السماح بإدخال مريض في غرفة غير موجودة.

2. عدم السماح بحذف سجلّ، في حال تطابق قيمة المفتاح الرئيسيّ للسجلّ مع قيمة المفتاح الأجنبيّ في الجدول المرتبط معه.

   مثال: عدم إمكانيّة حذف بيانات غرفة في مستشفى في حال تواجد مرضى في هذه الغرفة.

3. عدم السماح بتعديل قيمة المفتاح الرئيسيّ، إذا وجدت هذه القيمة كمفتاح أجنبيّ في الجدول المرتبط معه.

   مثال: لا يمكن تعديل رقم الدواء في جدول الأدوية ما دام استخدامه يتمّ كعلاج في جدول العلاج.

نشاط (5) نظري: بالاعتماد على نشاط رقم (4)، إذا علمت أنّه تمّ فرض تكامل مرجعيّ على العلاقات كافّةً في قاعدة بيانات المستشفى (انظر شكل 2-4)، أُجيبُ عمّا يأتي:

- ما القواعد التي تمّ فرضها على كلّ علاقة؟
- أذكرُ السجلّات التي يمكن حذفها من قاعدة البيانات دون انتهاك قواعد التكامل المرجعيّ.
- أذكر أرقام المرضى التي يمكن تعديلها دون انتهاك قواعد التكامل المرجعيّ.

- **شروط العلاقة بين جدولين:**

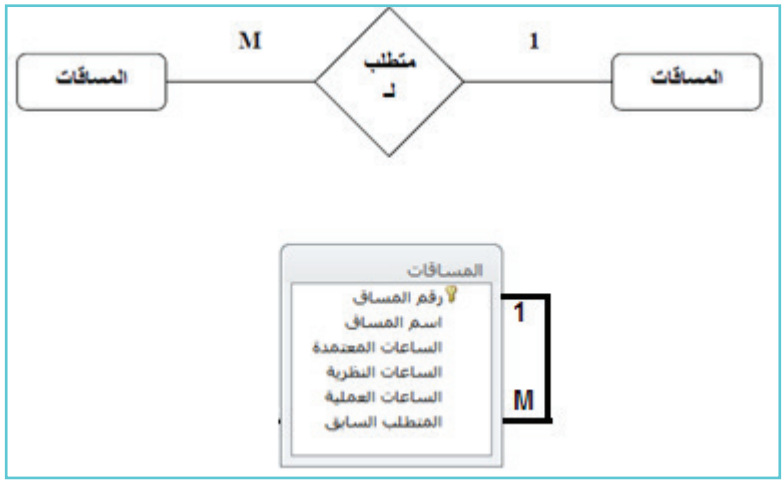
لربط علاقة تكامُل مرجعيّ صحيحة بين جدولين في قاعدة البيانات العلائقيّة، يلزم توفير الشروط الآتية:

1. أن تكون حقول الربط بين الجدولين من نوع البيانات نفسه، (مثال: رقميّة، نصيّة، تاريخ....).
2. أن تكون حقول الربط من حجم البيانات نفسه، (مثال: رقم صحيح مع رقم صحيح، أو نصّ طوله 5 أحرف مع نصّ طوله 5 أحرف....الخ)
3. ألّا يكون هنالك بيانات في احد الجدولين أو كليهما مخالفة لإحدى قواعد التكامل المرجعيّ،(مثال: إدخال المريض في غرفة رقمها غير موجود في جدول الغرف)؛ لذلك يُفضَّل دائماً إنشاء التكامل المرجعيّ على العلاقات التي تربط بين الجداول قبل إدخال أيّة بيانات فيها.
4. أن تدلّ حقول الربط على المعلومة نفسها ، كأن يدلّ رقم الغرفة في جدول المرضى ورقم الغرفة في جدول الغرف على المعلومة نفسها، وهي أرقام غرف المستشفى، وبالتالي يمكن الربط بينهما، ولا يُشترط أن تكون حقول الربط بين الجدولين بالمسمّى نفسه.

ملحوظة:في بعض الحالات تحتاج لربط الجدول مع نفسه في العلاقة، وتُسمّى العلاقة في هذه الحالة علاقة انعكاسيّة.

مثال(1): المساق ومتطلّبه السابق

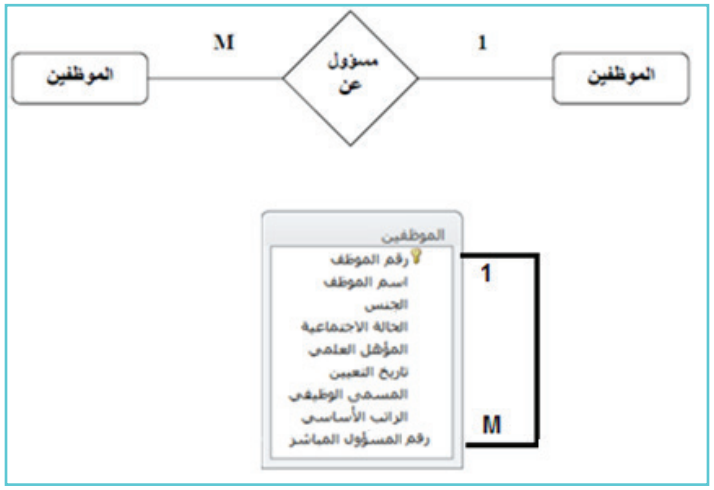
عندما يكون للمساق الواحد متطلّب سابق واحد فقط والمتطلب هو أيضاً مساق، والمساق الواحد متطلّب سابق لأكثر من مساق واحد. (مثال: مقدّمة في الحاسوب متطلّب سابق لأكثر من مساق).



شكل 9: نوع العلاقة بين المساق ومتطلّبه السابق

مثال(2): الموظّف ومسؤوله المباشر

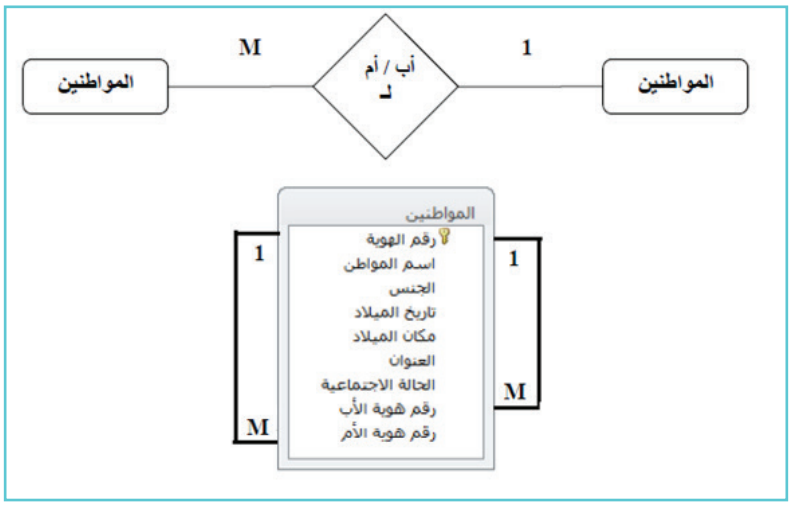
عندما يكون للموظف الواحد مسؤول مباشر واحد فقط والمسؤول هو أيضاً موظّف.



شكل 10: نوع العلاقة بين الموظّف ومسؤوله المباشر

مثال(3): المولود ورقم هويّة والده ووالدته

حيث يُسجَّل رقم هويّة الأب ورقم هويّة الأم في سجلّ المولود، فالمولود مواطن ووالده مواطن ووالدته أيضاً مواطنة.



شكل 11: نوع العلاقة بين المولود ووالده ووالدته

نشاط (6) نظري: أكمل الجدول الآتي بذكر نوع العلاقة التي تربط بين الجداول الآتية:

الجداول	نوع العلاقة
الطالب والمساقات	
الموظّفون والأقسام	
المشتركون في مسابقة والفائزون	
الموظفون والأبناء	
العملاء والحسابات البنكيّة	
الطبخة ومكوّناتها	
المريض والزيارات	
الأطفال والتطعيمات	
المالك والسيّارة	
الطالب والشهادة	
المدير والمدرسة	

الأسئلة

1. أُوضّح العلاقة بين الجداول التي تربط بين المفتاح الرئيسيّ في الجدول الاساسيّ ومفتاحه الأجنبيّ في الجدول التابع؟
2. أذكر مثالاً واحداً على كلّ نوع من أنواع العلاقات الآتية: (واحد إلى واحد، واحد إلى متعدّد، متعدّد إلى متعدّد)، وتوضيح ذلك برسم الجداول، مع توضيح الحقول التابعة لكلّ جدول، وتمييز المفتاح الرئيسيّ عن الأجنبيّ في العلاقة؟

2-3 الموقف التعليميّ التعلُّميّ

بناء قواعد البيانات ومعالجتها باستخدام نظام SQLite.

وصف الموقف التعليميّ: توجَّه صاحب شركة للخدمات اللوجستيّة إلى شركة تطوير برمجيّات؛ لمساعدته في إنشاء تطبيق يسمح بتتبُّع مخزون منتجات الشركة.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلّم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من صاحب الشركة حول: - منتجات الشركة. - إصدار نظام الأندرويد المستخدم في الشركة. • أجمع البيانات عن: - العناصر (views) المراد استخدامها. - عدد الشاشات المستخدمة في التطبيق. - نظام قواعد البيانات.	• حوار ومناقشة. • التعلم التعاونيّ/ مجموعات. البحث العلميّ.	• وثائق (طلب صاحب الشركة، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (حاسوب، إنترنت، هاتف يعمل بنظام الأندرويد).
أخطّط وأقرّر	• أحدّد: - عدد الواجهات. - عناصر واجهة المستخدم والتنسيقات المناسبة لها. - وضع مقترحات لواجهات التطبيق. - نوع قواعد البيانات (SQLite). - الجمل البرمجيّة المستخدمة. • إعداد خطّة العمل.	• تعلُّم تعاونيّ/ مجموعات. • حوار ومناقشة. • عصف ذهنيّ/ استمطار الأفكار.	• وثائق (طلب صاحب الشركة، البيانات التي تمّ جمعها سابقاً). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أنفّذ	• عقد جلسة مع صاحب الشركة لاختيار واجهات التطبيق المناسبة. • تصميم وبرمجة واجهات التطبيق، وإضافة قواعد البيانات. • عمل اختبار للتطبيق على جهاز يعمل بنظام أندرويد، أو جهاز افتراضيّ.	• تعلم تعاونيّ/ مجموعات. • - حوار ومناقشة.	• الطلب الشركة الزائرة. التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد)، برنامج android studio، جهاز يعمل بنظام أندرويد).

أتحقّق	• أتحقّق من: - تصميم واجهات التطبيق. - ملاءمة التطبيق لاحتياجات الشركة. - عدم ظهور أيّ خطأ أثناء التنفيذ.	• تعلم تعاونيّ / مجموعات. • حوار ومناقشة.	• طلب صاحب الشركة. • الخطّة المعدّة سابقاً. • البيانات التي تمّ جمعُها. • تكنولوجيا (إنترنت، حاسوب، جهاز يعمل بنظام أندرويد).
أوثّق وأقدّم	• توثيق البيانات التي تتعلّق بِـ: - العناصر (views) المراد استخدامها. - عدد الشاشات المستخدمة في التطبيق. - نظام قواعد البيانات. • إعداد عرض تقديميّ. • إعداد ملفّ بالحالة (إنشاء تطبيق يسمح بتتبُّع مخزون منتجات شركة خدمات).	• حوار ومناقشة. • تعلم تعاوني/ مجموعات عمل.	• البيانات التي تمّ جمعُها سابقاً. • تكنولوجيا (حاسوب، شاشة عرض).
أقوّم	• ملاءمة التطبيق للمعايير والمواصفات. • رضا صاحب الشركة عن التطبيق.	• حوار ومناقشة.	• طلب صاحب الشركة. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد يهتم بواجهات التطبيق وعناصرها وإنشاء قوائم البيانات.

نشاط (1) للاطلاع: أبحث عن أنواع البيانات (الحقول) التي تدعمهاSQLite.

أتعلّم: بناء قواعد البيانات ومعالجتها باستخدام نظام SQLite

نشاط (2) نظري: أُناقش: تُعدُّ قواعد البيانات SQLite الأمثل لتخزين البيانات الخاصة ومعالجتها بتطبيقات الهواتف المحمولة.

يُعَدُّ استخدام قواعد البيانات عنصراً أَساسيّاً في معظم تطبيقات الهواتف المحمولة. وتُقسم التطبيقات وفق طبيعة استخدامها للبيانات إلى قسمين:

1. تطبيقات تعتمد بشكل كُلّيّ على الكمّ الهائل من البيانات، مثل تطبيقات مِنصّات التواصل الاجتماعيّ.
2. تطبيقات تعتمد على كميّات قليلة من البيانات، وتقوم بتخزينها بشكل متواصل، مثل عدد النقاط التي تحصل عليها في لعبة معيّنة.

SQLite:

محرّك SQL مفتوح المصدر يتّسم بالسرعة والقوّة. ويمكن من خلال SQLite القيام بالعمليّات الأساسيّة لإدارة البيانات (CRUD): الإنشاء Create، والقراءة Read، والتحديث Update، والحذف Delete.

لماذا SQLite؟

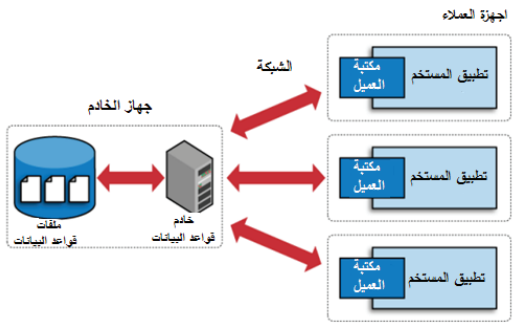
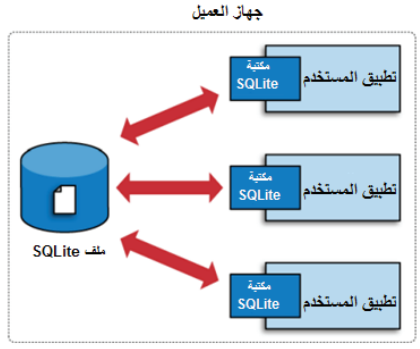
الهدف الأساسيّ من استخدام SQLite هو الاستغناء عن نظم إدارة قواعد البيانات التقليديّة (Database Management Systems أو DBMS) التي تحتاج لوجود خادم Server، وتحتاج لعمليّة إعدادات معقّدة، في حين أنّ قاعدة البيانات في SQLite تتكوّن فقط من ملف واحد صغير الحجم.

نشاط (3) للاطلاع: أُعدّد ميّزات قواعد البيانات التقليديّة التي لا تحقّقها SQLite.

ميّزات SQLite:

- صِغَر حجم ملف قاعدة البيانات (أقلّ من 600 كيلوبايت)؛ لذا يمكن استخدامها ضمن الأجهزة ذات الموارد المحدودة، كالهواتف المحمولة.
- معتمدة على الملفات: تتكوّن قاعدة البيانات بأكملها من ملف واحد على القرص؛ ما يجعلها محمولة تماماً.
- سهولة نقل البيانات من جهاز لآخر.
- سهولة التركيب، فلا نحتاج لإعدادات كثيرة، كما في نظم إدارة قواعد البيانات التقليديّة.
- تدعم الترميز في الكثير من اللغات، ولا سيّما اللغة العربيّة.
- تأتي دائماً مُتضَمَّنة في البرنامج الذي يتخاطب معها، بحيث تُمَكِّن البرنامج من استدعاء وظائفها بواسطة مناداة (call) بسيطة؛ ما يقلّل الزمن المطلوب للوصول إلى قاعدة البيانات.
- تدعم حجم قاعدة البيانات، وقد يصل إلى 2 تيرابايت.

توضيح الفرق بين أسلوب بناء نظم DBMS التقليديّة وSQLite:

نظام خادم/عميل في إدارة قواعد البيانات التقليديّة	نظام SQLite
	

شكل 1-4: التركيب البنائيّ لنظام قواعد البيانات التقليديّة ونظام SQLite

حزمة أوامر SQLite:

تتشابه الأوامر التي تستخدمها SQLite في التفاعل مع قواعد البيانات مع الأوامر التي تستخدمهاSQL القياسيّة، وهذه الأوامر هي: CREATE، وSELECT، وINSERT، وUPDATE، و DELETE، وDROP. ويمكن تصنيفها إلى ثلاثة مجموعات استناداً إلى طبيعتها الوظيفيّة.

1. الأوامر التي تُعنى بتعريف البيانات، وتُسمّى لغة التعريف (Data Definition Language) DDL.

الأمر	الهدف الوظيفيّ
CREATE	يُستخدم هذا الأمر لإنشاء: جدول جديد New table عرض جزئيّ أو كليّ من الجدول View of a table أيّ عنصر آخر من عناصر قواعد البيانات object
ALTER	يهدف هذا الأمر إلى تغيير أو تعديل العناصر في قاعدة البيانات.
DROP	يستخدم هذا الأمر لحذف: جدول table عرض جزئيّ أو كليّ من الجدول View of a table أيّ عنصر آخر من عناصر قواعد البيانات object

2. الأوامر التي تُعنى بمعالجة البيانات، وتُسمّى لغة المعالجة (Data Manipulation Language) DML

الأمر	الهدف الوظيفيّ
INSERT	إضافة السجلّات إلى الجداول.
UPDATE	تحديث السجلّات في الجداول.
DELETE	حذف السجلّات من الجداول.

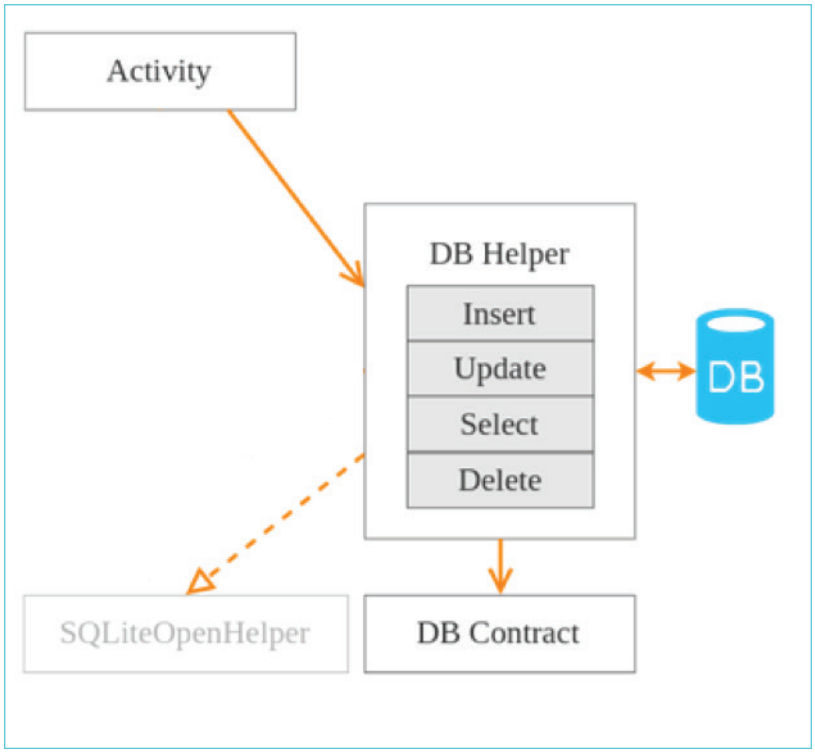
3. الأوامر التي تُعنى بالاستعلام عن البيانات، وتُسمّى لغة الاستعلام (Data Query Language) DQL

الأمر	الهدف الوظيفيّ
SELECT	الاستعلام عن السجلّات من الجداول.

الأصناف والدوال:

عند تصميم تطبيق يتعامل مع قواعد البيانات في كوتلن، لا بدّ من الاهتمام بثلاثة أصناف (Classes):

DB Contract Class	يحتوي على مخطّط (اسم الجدول وأسماء الأعمدة).
DB Helper Class	يحتوي هذ الصنف على الدّوال التي تقوم بعمليّات على قاعدة البيانات، مثل: الإضافة، والحذف، والتحديث، والاستعلام.
Activity Class	هذا الصنف يعبّر عن الكود البرمجي الذي من خلاله يتم استدعاء الدوال الموجودة في صنف DB Helper لإدارة العمليّات على قاعدة البيانات.



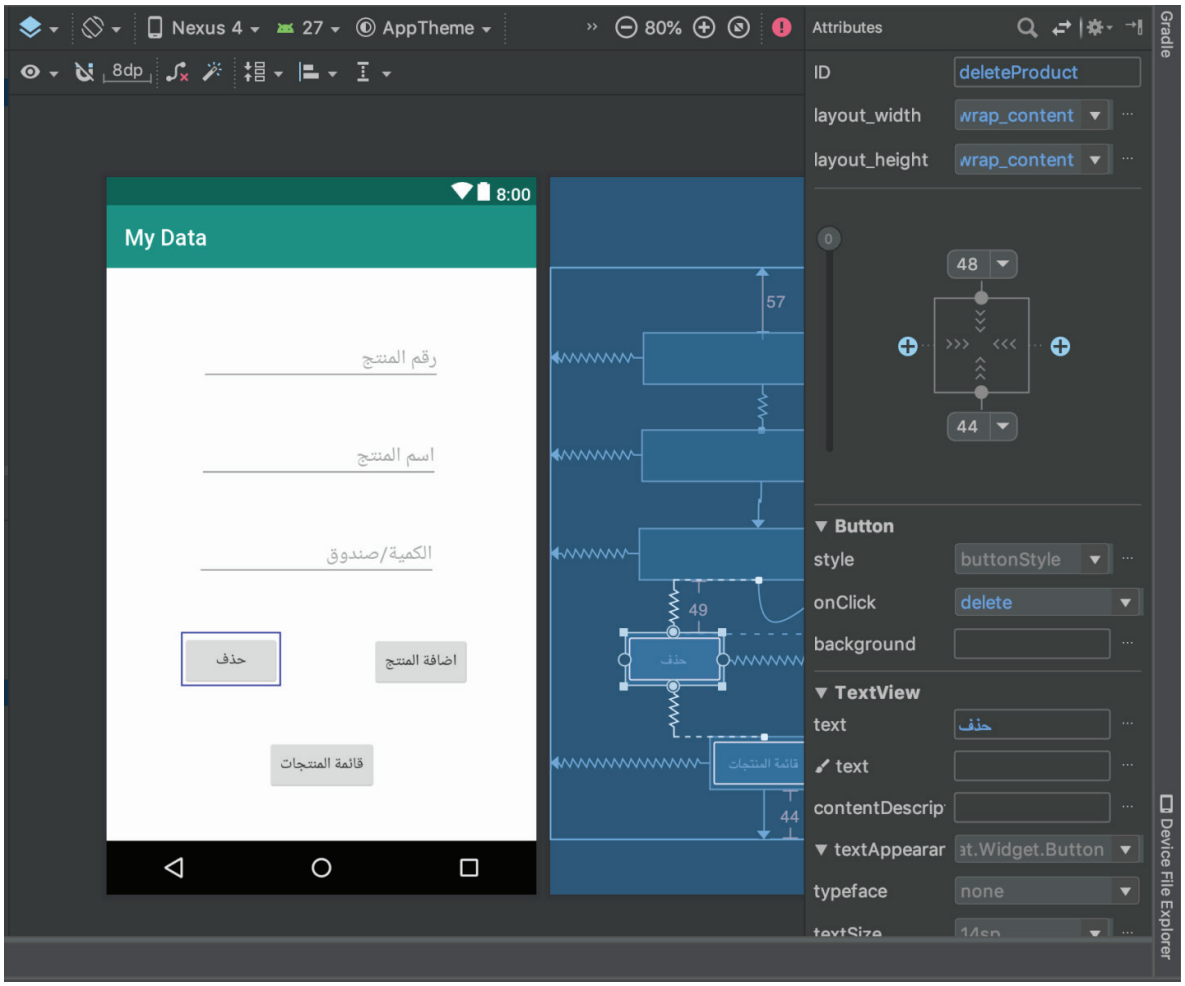
أهمّ الأصناف والدّوال في SQLite:

الصنف Class-	الغاية والوظيفة	أهمّ الدوال(Methods) المعرفة ضمن الصنف
المؤشّر-Cursor	توفير الوصول إلى نتائج الاستعلام من قاعدة البيانات.	• Close() تحرير جميع الموارد التي يستخدمها المؤشّر وإغلاقها. • getCount() دالّة تفيد بعدد السجلّات التي تمّ استرجاعها. • moveToFirst() الإشارة للسجلّ الأوّل في مجموعة السجلّات التي تمّ استرجاعها • moveToLast() الإشارة للسجلّ الأخير في مجموعة السجلّات التي تمّ استرجاعها. • moveToNext() الإشارة للسجلّ التالي في مجموعة السجلّات التي تمّ استرجاعها. • move() نقل المؤشّر من موقعه الحالي إلى مكان آخر بإزاحة محدّدة.
SQLiteDatabase	يعمل هذا الصنف كواجهة بين الكود البرمجيّ وقواعد بيانات SQLite، بما في ذلك القدرة على إنشاء وحذف وتنفيذ عمليات أخرى على قواعد البيانات.	• insert() إدراج صف جديد في جدول قاعدة البيانات. • delete() حذف الصفوف من جدول قاعدة البيانات. • query() تنفيذ عبارة استعلام محددة واسترجاع النتائج من خلال كائن من نوع Cursor. • execSQL() تنفيذ عبارة استعلام SQL واحدة لا تعيد بيانات النتائج. • rawQuery() تنفيذ عبارة استعلام SQL وإرجاع نتائج مطابقة على هيئة كائن Cursor.

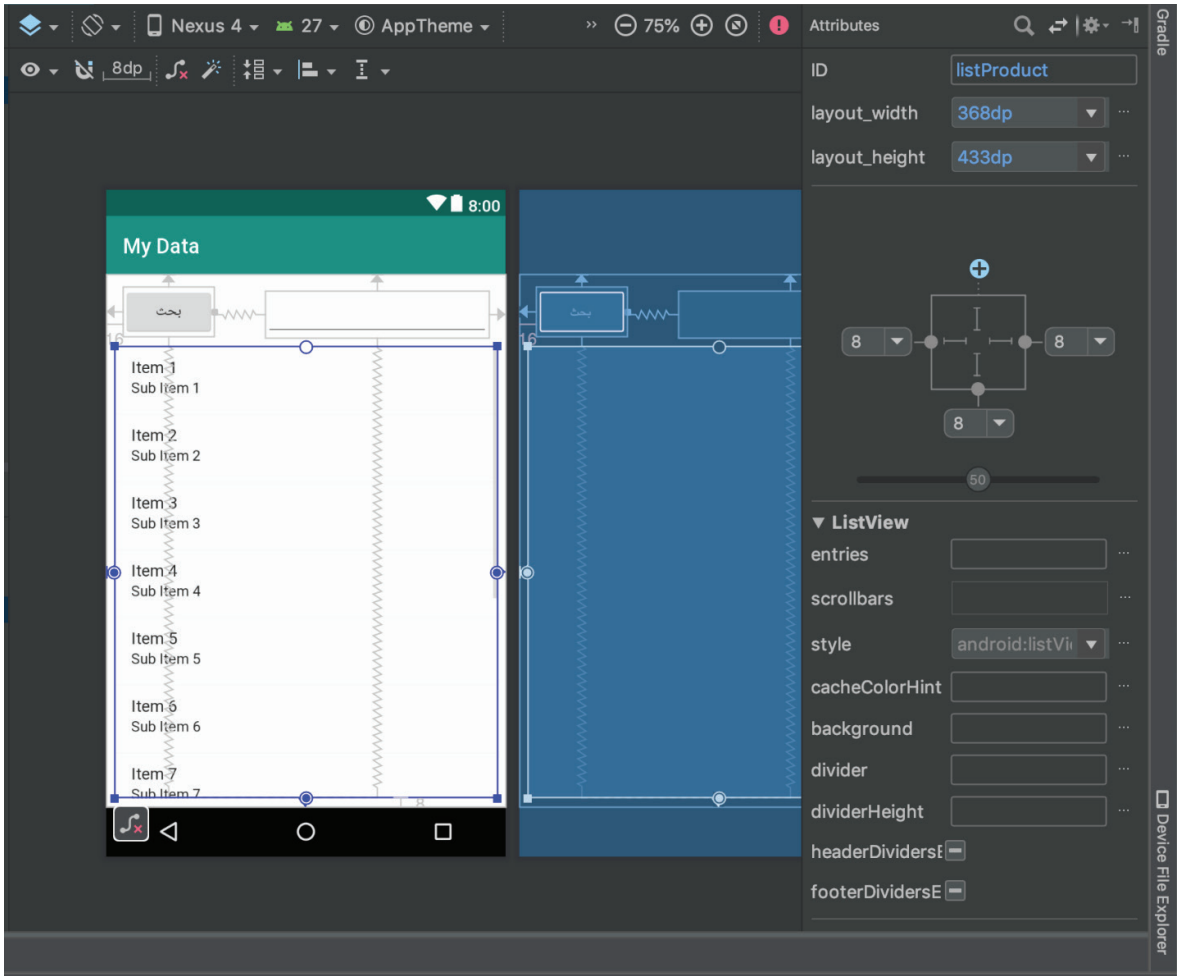
SQLiteOpen-Helper	لتسهيل إنشاء وتحديث قواعد البيانات	• onCreate() • يتمّ استدعاء الدّالّة عند إنشاء قاعدة البيانات للمرة الأولى، ويتمّ تمرير هذه الدالّة إلى كائن SQLiteDatabase • onUpgrade() يتمّ استدعاء الدّالّة في حالة احتواء الكود البرمجيّ على مرجع رقم إصدار قاعدة بيانات أحدث، ويُستخدم هذا عادة عند تحديث تطبيق على الجهاز، ويتطلّب تحديث مخطّط قاعدة البيانات أيضاً للتعامل مع تخزين البيانات الإضافيّة. • getWritableDatabase() يَفتح أو يُنشِئ قاعدة بيانات للقراءة والكتابة، ويعيد مرجعاً إلى قاعدة البيانات في شكل كائن SQLiteDatabase. • getReadableDatabase() يَفتح أو يُنشِئ قاعدة بيانات للقراءة فقط، ويعيد مرجعاً إلى قاعدة البيانات في شكل كائن SQLiteDatabase. • close() إغلاق قاعدة البيانات.

نشاط (4) عملي: أنشئ تطبيق إدارة مخزون لمنتجات معبّنة، يمكن من خلاله الإضافة والحذف والتعديل والبحث عن المنتجات.

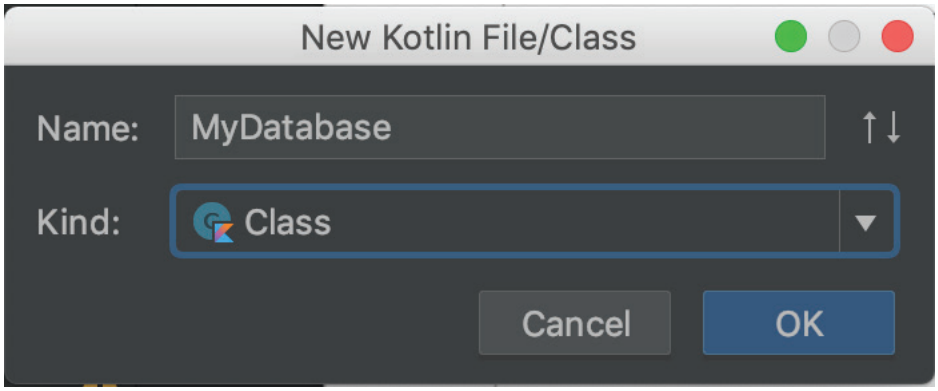
- **أوّلاً:** افتح الأندرويد ستوديو، وأُنشئ مشروعاً جديداً، ثمّ أُضيف العناصر اللازمة وأُعدّل الid لكلّ عنصر بما يناسب، كالآتي:



- **ثانياً:** أُنشئ activity جديدة، وأضيف في ملف واجهة التصميم كما يأتي:



- **ثالثاً:** أُنشئ صنفاً خاصّاً بقاعدة البيانات، أضغط بزر الفارة اليمين على app ، ثمّ new، ثمّ kotlin file/ class، كالآتي:



- **رابعاً:** أكتب في الصنف ما يأتي:

```
1-  class MyDatabase(context: Context) : SQLiteOpenHelper(context, MyDatabase.DATABASE_NAME, factory: null ,MyDatabase.DATABASE_VERSION ) {

3-      override fun onCreate(db: SQLiteDatabase?) {
            db!!.execSQL(DATABASE_CREATE)
        }

4-      override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
            db!!.execSQL( sql: " DROP TABLE IF EXISTS $DATABASE_CREATE" )
            onCreate(db)

        }

2-      companion object {
            val KEY_ID = "_ID"
            val P_NAME = "NAME"
            val P_QUANTITY = "QUANTITY"
            val DATABASE_TABLE_NAME = "products"

            private val DATABASE_NAME = "data.db"
            private val DATABASE_VERSION = 1


            private val DATABASE_CREATE =
                "CREATE TABLE $DATABASE_TABLE_NAME ($KEY_ID INTEGER PRIMARY KEY, " +
                        "$P_NAME TEXT NOT NULL, " + "$P_QUANTITY INTEGER);"

        }

    }
```

1. أجعل الصنف يرث SQLitOpenHelper الذي يستقبل ثلاث قيم:
   - context المعرفه في الباني الخاص بالصنف الرئيسيّ.
   - اسم ملف قاعدة البيانات الذي سأعرّفه في الخطوة التالية.
   - قيمة null.
   - إصدار قاعدة البيانات.
2. أُنشئ كائناً مساعداً يحتوي على عدّة متغيّرات حتى يتمّ استخدامها في أيّ مكان في الصنف الحالي:
   - أُعرّف متغيّراً يحمل اسم العمود الذي يخزّن فيه رقم المنتج.
   - أُعرّف متغيّراً يحمل اسم العمود الذي يخزّن فيه اسم المنتج.
   - أُعرّف متغيّراً يحمل اسم العمود الذي يخزّن فيه الكميّة.
   - أُعرّف متغيّراً يحمل اسم جدول قاعدة البيانات.
   - أُعرّف متغيّراً يحمل اسم ملف قاعدة البيانات، ويكون امتدادة (.dp).
   - أُعرّف متغيّراً يحمل إصدار قاعدة البيانات.
   - أُعرّف متغيّراً لإنشاء قاعدة البيانات، وتتمّ بكتابة ما يأتي:
     - علامة التنصيص ".
     - أكتب CREAT TABLE.
     - إشارة & واسم جدول قاعدة البيانات ثمّ أفتح قوساً).
     - أكتب أوّلاً اسم العمود الذي يكون مفتاحاً أساسيّاً، ثمّ أكتب INTEGER PRIMARY KEY وتعني أنّ هذا العمود هو مفتاح أساسيّ للجدول ونوع البيانات Integer، وهنا يجب إدخال قيمة المفتاح يدويّاً، أمّا إذا أردت أن أجعله يأخذ قيمة تلقائيّة أُضيف AOTUINCRIMENT، ثمّ فاصلة، ثمّ فراغ، ثمّ أُضيف الأعمدة المتبقيّة مع نوع البيانات.

ملحوظة: يجب الانتباه إلى كتابة الكلمات؛ إمّا أحرفاً كبيرة أو صغيرة، وإلى أماكن الفواصل والفراغات.

3. في onCreate يتمّ الاستعلام عن قاعدة البيانات المنشأة.
4. في onUpgrade يتمّ الاستعلام إذا كان يوجد إصدار جديد لقاعدة البيانات وتحديثها.

- **خامساً:** اكتبُ دالّتي الإضافة والحذف كما يأتي:

```
fun addProduct(productName: String ,name: String,productId :String,id: Int,productQ: String,q :Int ) {
    val db :SQLiteDatabase! = this.writableDatabase
    val values = ContentValues()
    values.put(productName, name)
    values.put(productId,id)
    values.put(productQ,q)

    db.insert(DATABASE_TABLE_NAME, nullColumnHack: null, values)
    db.close()

}

fun deleteProduct(productName: String) {
    val db :SQLiteDatabase! = writableDatabase
    db.execSQL( sql: "delete from $DATABASE_TABLE_NAME where $P_NAME=\"$productName\";")

}
```

1. دالّة الإضافة تستقبل ست قيم ثلاث قيم عبارة عن البيانات، وثلاث هي عناوين الجداول، ثمّ أكتب بداخلها كالآتي:
   - أُعرّف كائناً من الصنف SQLiteDatabase لفتح قاعدة البيانات، وجعلها قابلة للاستخدام.
   - أُعرّف كائناً من الصنف ContentValues() لإضافة البيانات.
   - أبدأ بإدخال البيانات باستخدام الدالّة put التي تأخذ قيمتين: القيمة الأولى عبارة عن المفتاح وهو اسم العمود، والثانية القيمة المراد إضافتها إلى العمود.
   - إضافة البيانات إلى قاعدة البيانات باستخدام الدالّة insert التي تستقبل ثلاث قيم: وهي ملف قاعدة البيانات، والثانية أضعها null ، والثالثة البيانات التي أريد إضافتها والتي قمت بتخزينها في الخطوة السابقة في الكائن values.
   - إغلاق قاعدة البيانات.
2. دالّة الحذف تستقبل قيمة من نوع string ليتمّ حذف بيانات الصف من خلالها، وأكتب داخلها ما يأتي:
   - أُعرّف كائناً من الصنف SQLiteDatabase لفتح قاعدة البيانات وجعلها قابلة للاستخدام.
   - أكتب استعلام الحذف delete from ، ثمّ اسم الجدول، ثمّ where، ثمّ اسم العمود، ثمّ = ، وأكتب القيمة التي تستقبلها الدالّة.

- **سادساً:** أكتب دالّة البحث كالآتي:

```
fun findPtoduct(productname: String):Array<String>{
    var dbString :Array<String>
    val db :SQLiteDatabase! = writableDatabase
    val query = "select * from $DATABASE_TABLE_NAME where $P_NAME=\"$productname\";"

    val c :Cursor! = db.rawQuery(query, selectionArgs: null)
    dbString = Array<String>(c.count){" "}
    c.moveToFirst()


    while (!c.isAfterLast) {
        for (i :Int in dbString.indices){
            if (c.getString(c.getColumnIndex(KEY_ID)) != null&&c.getString(c.getColumnIndex(P_NAME)) != null&&c.getString(c.getColumnIndex(
                    P_QUANTITY)) != null) {

                dbString[i] += c.getString(c.getColumnIndex(KEY_ID))+"_ "+ c.getString(c.getColumnIndex(P_NAME))+" , "+c.getString(c.getColumnIndex(
                    P_QUANTITY))

            }
            c.moveToNext()
        }
    }
    c.close()
    return dbString

}
```

1. تستقبل الدالة قيمة ليتمّ البحث عنها، ثمّ تعيد مصفوفة بالقيم التي تمّ العثور عليها.
2. أُنشئ مصفوفة أسمّيها dbString.
3. أُعرّف كائناً من الصنف SQLiteDatabase لفتح قاعدة البيانات، وجعلها قابلة للاستخدام.
4. أُنشئ متغيّر استعلام عن القيمة المدخلة.
5. أُعرّف مؤشّراً للإشارة إلى بيانات الاستعلام في العمود الذي أريد البحث فيه.
6. أعرّف المصفوفة التي أنشأتها سابقاً، حجمها يكون بقيمة المؤشّر، وأعطيه قيمة أوليّة فارغة.
7. أجعل المؤشّر يؤشّر على بداية القيم.
8. أكتب جملة while لفحص أنّ المؤشّر لا يؤشّر على null.
9. أكتب جملة for للمرور بجميع قيم المصفوفة.
10. أكتب جملة if لفحص وجود قيمة في الصفّ من خلال أسماء الأعمدة.
11. داخل جمله if أخزّن البيانات التي تمّ العثور عليها في المصفوفة مع رقم المنتج والكميّة، وليس فقط اسم المنتج.
12. أكتب دالّة لينتقل المؤشّر إلى القيمة التالية بعد كلّ خطوة تتمّ في جملة for.
13. إغلاق قاعدة البيانات.
14. إرجاع قيمة المصفوفة.

- **سابعاً:** أكتب دالّة لإظهار جميع البيانات كما يأتي:

```
fun databaseToString(): Array<String> {
    var dbString :Array<String>
    val db :SQLiteDatabase! = writableDatabase
    val query = "select * from $DATABASE_TABLE_NAME where 1"

    val c :Cursor! = db.rawQuery(query, selectionArgs: null)
    dbString = Array<String>(c.count){" "}
    c.moveToFirst()


    while (!c.isAfterLast) {
        for (i :Int in dbString.indices){
        if (c.getString(c.getColumnIndex(KEY_ID)) != null&&c.getString(c.getColumnIndex(P_NAME)) != null&&c.getString(c.getColumnIndex(
                P_QUANTITY)) != null) {

                dbString[i] += c.getString(c.getColumnIndex(KEY_ID))+"_ "+ c.getString(c.getColumnIndex(P_NAME))+" , "+c.getString(c.getColumnIndex(
                    P_QUANTITY))

            }
            c.moveToNext()
        }

    }
    c.close()
    return dbString
}
```

تختلف هذه الدالّة عن دالّة البحث في أنّها لا تستقبل قيمة، و يكون الاستعلام في كلّ الجدول في السطر الآتي:

```
select * from &DATABASE_TABLE_NAME where 1
```

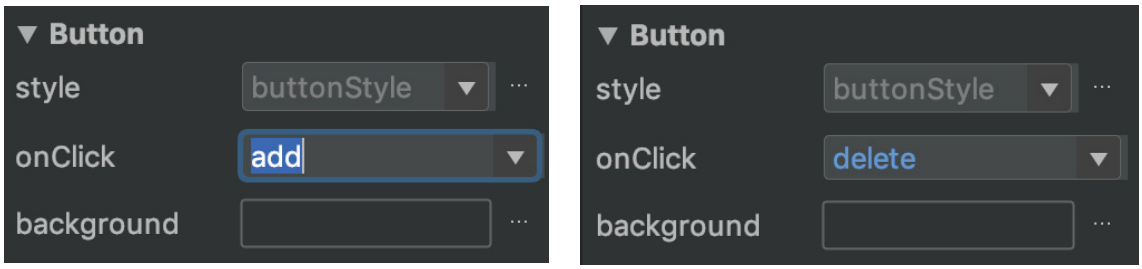
كتبت رقم واحد؛ أي في كلّ الجدول وليس بحثاً عن قيمة في عمود محدّد.

- **ثامناً:** أنتقل إلى صنف الواجهة الأولى، وأكتب كالآتي:

```
class MainActivity : AppCompatActivity() {
1-  lateinit var mydatabase:MyDatabase
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
2-  mydatabase =MyDatabase( context: this)
3-  listProductPage.setOnClickListener { it: View!
        val intent= Intent( packageContext: this,Main2Activity::class.java)
        startActivity(intent)
    }
}
4-  fun add(view: View) {
        val idProduct : String =idProduct.text.toString()
        val nameProduct : String = nameProduct.text.toString()
        val quantity : String = quantity.text.toString()
        mydatabase.addProduct(MyDatabase.P_NAME,nameProduct,MyDatabase.KEY_ID,idProduct.toInt(),MyDatabase.P_QUANTITY,quantity.toInt())

    }
5-  fun delete(view: View) {
        val nameProduct : String =nameProduct.text.toString()
        mydatabase.deleteProduct(nameProduct)

    }
}
```

1. أُنشئ كائناً من صنف قاعدة البيانات التي أنشأتها.
2. أُعرّف الكائن الذي أنشأته.
3. أكتب أمر الزر الخاص بالانتقال إلى الصفحة الثانية.
4. أكتب دالّة للإضافة add لربطها بالزر الخاص بالإضافة، أُعرّف ثلاثة متغيّرات من نوع String لتخزين كلّ قيمه مُدخلة في كلّ planText، ثمّ أستدعي الدالّة addProduct من خلال الكائن الذي أنشأته سابقاً، وأعطيها القيم: اسم العمود، والقيمة المدخلة، وهكذا.

5. أكتب دالّة للحذف delete لربطها بالزر الخاصّ بالحذف، أُعرّف متغيراً لتخزين القيمة المدخلة في planText، ثمّ أستدعي دالّة الحذف التي تستقبل القيمة المراد حذفها.
6. أربط الأزرار buttons بالدوال التي كتبتها كالآتي:



- **تاسعاً:** أنتقل لصنف الواجهة الثانية، وأكتب كالآتي:

```
class Main2Activity : AppCompatActivity() {
1-  lateinit var mydatabase:MyDatabase
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)

2-      mydatabase =MyDatabase( context: this)
        printData()

    }

3-  fun findPtoduct(view: View){
        val name : String  = editTextSearch.text.toString()

        val dbString : Array<String>  = mydatabase.findPtoduct(name)
        val adapter= ArrayAdapter<String>( context: this,android.R.layout.simple_expandable_list_item_1,dbString)
        listProductPage.adapter= adapter

    }

4-  fun printData(){
        val dbString : Array<String>  = mydatabase.databaseToString()
        val adapter= ArrayAdapter<String>( context: this,android.R.layout.simple_expandable_list_item_1,dbString)
        listProductPage.adapter= adapter
    }
}
```

1. أنشئ كائناً من صنف قاعدة البيانات التي أنشأتها.
2. أعرّف الكائن الذي أنشأته، وأكتب اسم الدالّة التي تعرض البيانات.
3. أكتب دالّة لربطها بزر البحث، أُعرّف متغيّراً لتخزين القيمة المُدخلة في planText، ثمّ أُعرّف مصفوفة لتخزين القيم المسترجعة من دالّة البحث، أنشئ كائناً من ArrayAdapter كما تعلمت سابقاً، ثمّ أعرض القيم في القائمة listView.
4. أكتب دالّة لعرض جميع البيانات، أُعرّف مصفوفة لتخزين جميع القيم الموجودة في الجدول داخلها، وعرضها في القائمة listView.

- **عاشراً:** أُشغّل المشروع وأختبر التطبيق، وأفحص النتائج.

الأسئلة

1. ما ميّزات Sqlite؟
2. ما هو الأمر الخاصّ في الاستعلام عن البيانات؟

أسئلة الوحدة

السؤال الأوّل: أضع دائرة حول رمز الإجابة الصحيحة فيما يأتي:

1. ماذا يُعدّ السجلّ في الجدول؟

أ. حقلاً. ب. صفّاً.

ج. عموداً. د. مفتاحاً رئيسيّاً.

2. ما الحقل الذي لا تتكرّر قيمته في الجدول؟

أ. مفتاح ثانويّ. ب. مفتاح رئيسيّ.

ج. مفتاح أجنبيّ. د. مفتاح مرشّح.

3. ما امتداد ملف قواعد البيانات الذي تمّ إنشاؤه في نظام SQLite؟

أ. .mdb ب. .db

ج. .fdb د. .sdb

4. لماذا يُستخدم الأمر execSQL؟

أ. تنفيذ عبارة استعلام SQL واحدة لا تعيد بيانات النتائج.

ب. تنفيذ عبارة استعلام محدّد واسترجاع النتائج.

ج. تنفيذ عبارة استعلام محدّد وحذف النتائج .

د. فتح أو إنشاء قاعدة بيانات للقراءة فقط.

5. تُسمّى الأوامر التي تُعنى بتعريف البيانات:

أ. DQL ب. SQL

ج. DML د. DDL

السؤال الثاني: أُعرّف ما يأتي:

- قواعد البيانات.
- أنظمة إدارة قواعد البيانات (SQLite).
- المفتاح الرئيسيّ.
- DDL

السؤال الثالث: ما هي الأصناف (Classes) التي يجب إنشاؤها عند عمل تطبيق يتعامل مع قواعد البيانات؟

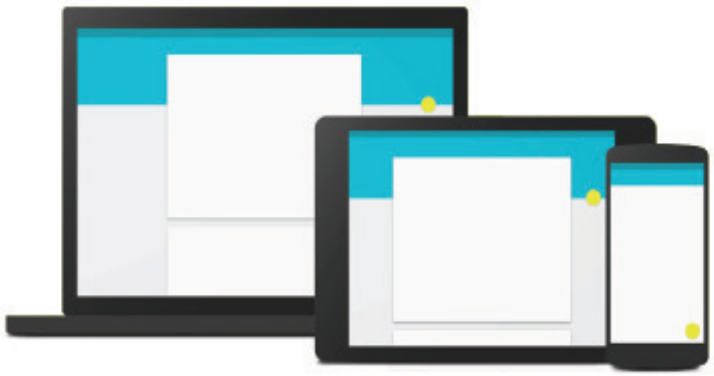
السؤال الرابع: أعلّل: تحديد مفتاح رئيسيّ لكلّ جدول من جداول قاعدة البيانات.

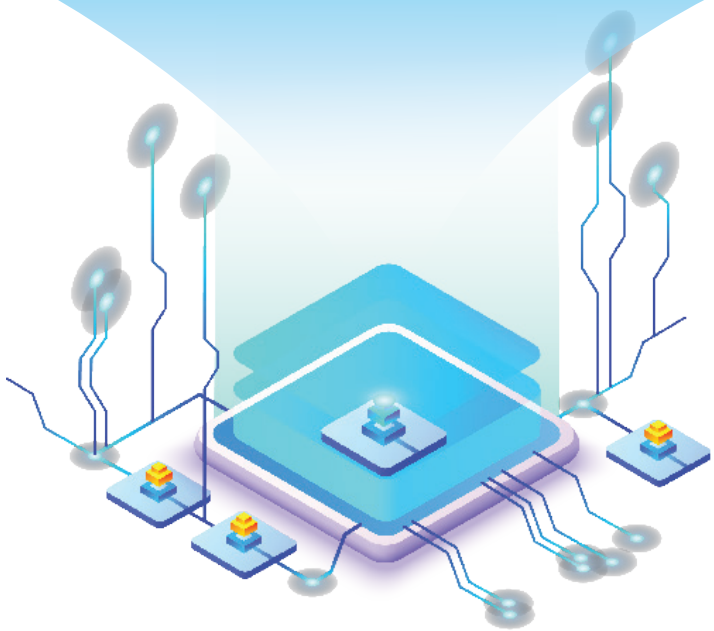
مشروع الوحدة:

أحلّل وأنشئ تطبيقاً يساعد في تخزين رواتب الموظّفين واسترجاعها، واحتساب أجورهم باستخدام قاعدة البيانات SQLite.

الوَحْدةُ الرابعة

تنسيق التطبيق





أُناقش: التصميمُ الجيّدُ من أبرزِ العواملِ التي تساعدُ على انتشارِ التطبيقِ.

يُتوقَّعُ من الطلبة بعد دراسة هذه الوَحدة، والتفاعُل مع أنشطتها أن يكونوا قادرين على إعداد تنسيقات لتطبيق الأندرويد من خلال تحقيق الآتي:

1. تعديل أيقونة التطبيق.
2. إعداد التطبيق بأكثر من لغة.
3. تحسين شكل التطبيق (style).
4. إعداد التطبيق للعمل على أجهزة الأندرويد المختلفة.

الكفايات المِهْنيّة

الكفايات المهنيّة المُتوقّع من الطلبة امتلاكها بعد الانتهاء من دراسة هذه الوحدة، والتفاعل مع أنشطتها:

أولاً: الكفايات الحرفيّة.

القدرة على:

- تعديل أيقونة التطبيق.
- إعداد التطبيق بأكثر من لغة.
- إعداد ملف للألوان للاستخدام في التطبيق.
- إعداد ملف للقالب (style) للاستخدام في التطبيق.
- إعداد التطبيق للعمل على أجهزة الأندرويد المختلفة.
- تقويم النتائج.

ثانياً: الكفايات الاجتماعيّة والشخصيّة.

- العمل ضمن فريق.
- تقبُّل التغذية الراجعة.
- احترام رأي الآخرين.
- القدرة على إدارة الحوار وتنظيم النقاش.
- مصداقيّة التعامل مع الزبون.
- حِفظ خصوصيّة الزبون.
- الاستعداد للاستجابة لطلبات الزبون.
- التحلّي بأخلاقيّات المهنة أثناء العمل.
- التمتُّع بالفكر الريادي أثناء العمل.
- استخلاص النتائج ودقّة الملاحظة.
- الاتصال والتواصل الفعّال.
- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
- الإفادة من مقترحات الآخرين.
- امتلاك مهارة التأمُّل الذاتيّ.
- الاستعداد التام لتقبُّل ملحوظات الزبون وانتقاداته.
- القدرة على اقتراح البدائل والإقناع.

ثالثاً: الكفايات المنهجيّة.

- توظيف التعلّم التعاونيّ.
- الحوار والمناقشة.
- امتلاك مهارة البحث العلميّ، والقدرة على توظيف أساليبه.
- القدرة على التحليل.
- القدرة على التخطيط.
- القدرة على استمطار الأفكار (العصف الذهنيّ).
- استخدام المصادر والمراجع المختلفة.
- توثيق نتائج العمل وعرضها.
- تقويم النتائج وعرضها.
- جمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:

- الالتزام بالجلسة الصحيّة أمام الحاسوب.
- إراحة العينين كلّ ربع ساعة.
- الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة.
- الابتعاد عن الحاسوب كلّ ساعة، أو القيام ببعض التمارين الرياضيّة.

1-4 الموقف التعليميّ التعلُّميّ

تعديل أيقونة التطبيق.

وصف الموقف التعليميّ: زار مطوّر تطبيقات أندرويد مشغل تطبيقات الهاتف المحمول في مدرسة اللّد المِهْنيّة؛ لمساعدته في تعديل أيقونة التطبيق الخاصّ به.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلّم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من المطوّر عن التطبيق من حيث: - ملف المشروع. - الأيقونة المراد استخدامها. • أجمع البيانات عن كيفيّة تغيير الأيقونة لتطبيق الأندرويد.	• حوار ومناقشة. • التعلّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب المطوّر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة، حاسوب، جهاز محمول).
أخطّط وأقرّر	• أصنّف البيانات الخاصّة بملف المشروع والأيقونة المراد تغييرها. • أحدّد آليّة تغيير الأيقونة (android studio) • أُعدّ خطّة للتنفيذ.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• وثائق (طلب المطوّر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أنفّذ	أُجهّز أيقونة التطبيق بعدّة نسخ. • أضيف الأيقونة للتطبيق. • أنفّذ التطبيق على جهاز أندرويد افتراضيّ، أو جهاز أندرويد حقيقيّ.	• تعلّم تعاونيّ/مجموعات. • حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار.	• الطلب الخاصّ بالمطوّر. • حاسوب، برنامج خاصّ (android studio).

أتحقّق	• أتحقّق من تعديل أيقونة التطبيق. • أتحقّق من ملاءمة أيقونة التطبيق لطلب المطوّر.	• تعلّم تعاونيّ/مجموعات. • حوار ومناقشة.	• الخطّة المعدّة سابقاً. • الطلب الخاصّ بالمطوّر. • حاسوب، إنترنت. • هاتف محمول.
أوثّق وأقدّم	• أوثّق آليّة تعديل الأيقونة. • إعداد العروض التقديميّة. • إعداد ملف بالحالة (تعديل أيقونة التطبيق).	• حوار ومناقشة. • تعلُّم تعاونيّ/ مجموعات عمل.	• حاسوب. • شاشة عرض.
أقوّم	• رضا المطوّر عن تعديل الأيقونة. • ملاءمة التطبيق بعد تعديل الأيقونة للمعايير والمواصفات.	• حوار ومناقشة.	• الطلب الخاص بمطور التطبيقات. • نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد.

أناقش مع زملائي فوائد الأيقونات للتطبيقات في الحواسيب والهواتف النقّالة.

أتعلّم: تعديل أيقونة التطبيق

نشاط (1) نظري: أبحث عن أبعاد الأيقونات المستخدمة في الأندرويد؟

أيقونة التطبيق:

تُعدّ أيقونة التطبيق عنصراً مهمّاً لإنجاح التطبيق وانتشاره؛ لأنّها تلخّص محتوى التطبيق، وتلفت انتباه الأشخاص خاصّة إذا كان هنالك اكثر من تطبيق بالاسم نفسه؛ لذلك من الأفضل أن تكون الأيقونة مميّزة تعبّر عن فكرة التطبيق بشكل إبداعيّ.

أحجام أيقونة التطبيق وفق دقّة شاشة الهاتف الذكيّ:

1. mdpi: تُستخدم للأجهزة ذات دقّة متوسّطة (160 dpi)، أبعاد 48*48 بيكسل.
2. hdpi: تُستخدم للأجهزة ذات دقّة عالية (240 dpi)، أبعاد 72*72 بيكسل.
3. xhdpi: تُستخدم للأجهزة ذات دقّة أعلى من hdpi (320 dpi)، أبعاد 96*96 بيكسل.

4. xxhdpi: تُستخدم للأجهزة ذات دقّة أعلى من xhdpi (480 dpi)، أبعاد 144*144 بيكسل.
5. xxxhdpi: تُستخدم للأجهزة ذات دقّة أعلى xxhdpi (640 dpi)، أبعاد 192*192 بيكسل.
6. nodpi: (anydpi) وتُستخدم مع جميع الأجهزة التي تعمل بنظام Oreo 8 والأنظمة الأعلى، وتعطي الحجم المناسب للأيقونة تلقائيّاً وفق دقّة الشاشة.

طُرق إعداد أيقونة التطبيق:

• أولاً: برامج معالج الصور

يوجد العديد من البرامج لتغيير حجم وامتداد الصور، ومعالجتها بشكل احترافيّ أو بسيط، مثل: الفوتوشوب،.....

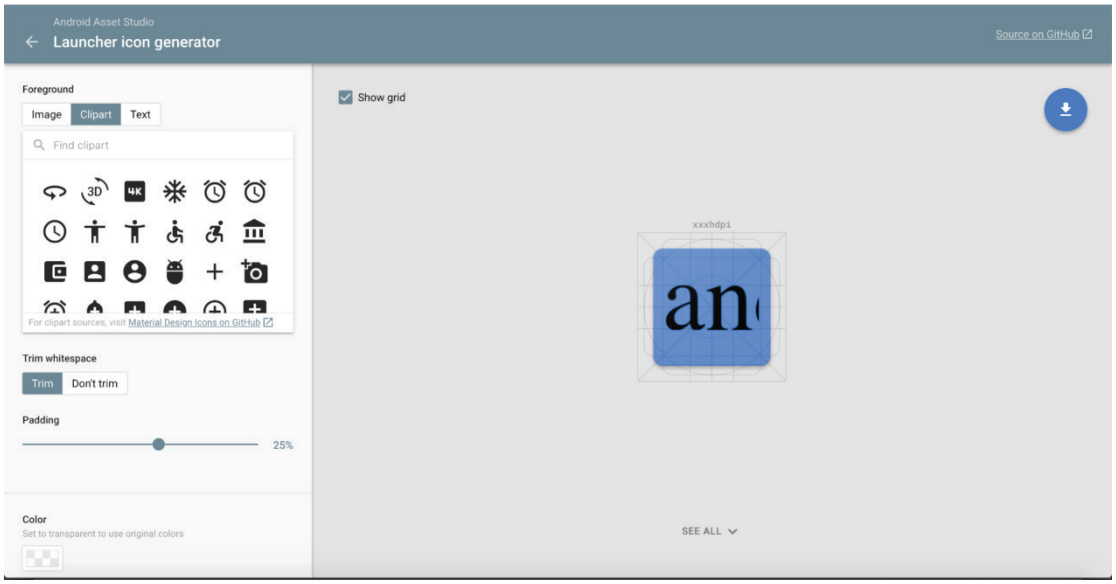
نشاط (2) للاطلاع: أبحث مع زملائي عن برامج معالجة الصور.

• ثانياً: مواقع إلكترونيّة لعمل أيقونة التطبيق

تختلف طريقة عمل الأيقونة من موقع لآخر، وهناك العديد من المواقع المتخصّصة بعمل أيقونات لتطبيقات الأندرويد، مثل:

https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html

الشكل (1-5) صورة توضيحيّة للموقع:



نشاط (3) للاطلاع: أبحث مع زملائي عن مواقع لصناعة أيقونات للتطبيق، وأحاول استخدام إحداها.

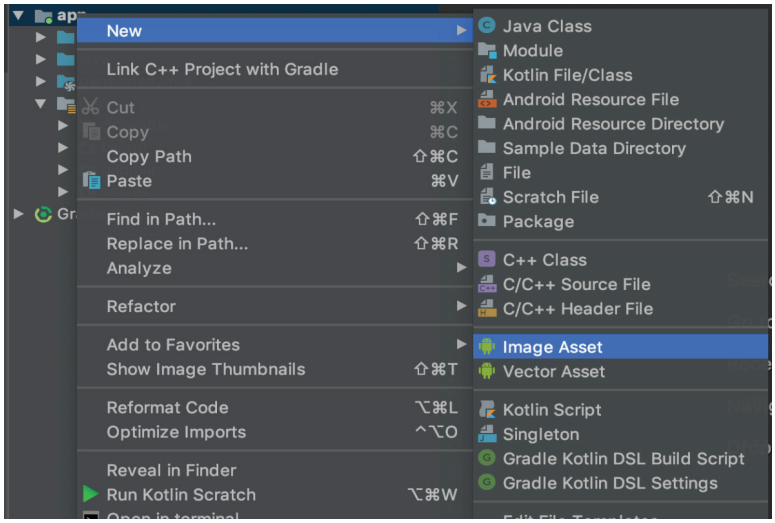
• **الثاً: استخدام الأندرويد ستوديو**

عند إنشاء مشروع على الأندرويد ستوديو، يعطي برنامج التطبيق أيقونة افتراضيّة؛ لأنّه لا يمكن أن يعمل التطبيق دون أيقونة، توجد الأيقونة في ملف المشروع داخل مجلّد اسمه mipmap يحتوي على ملفين من الصور، هما:

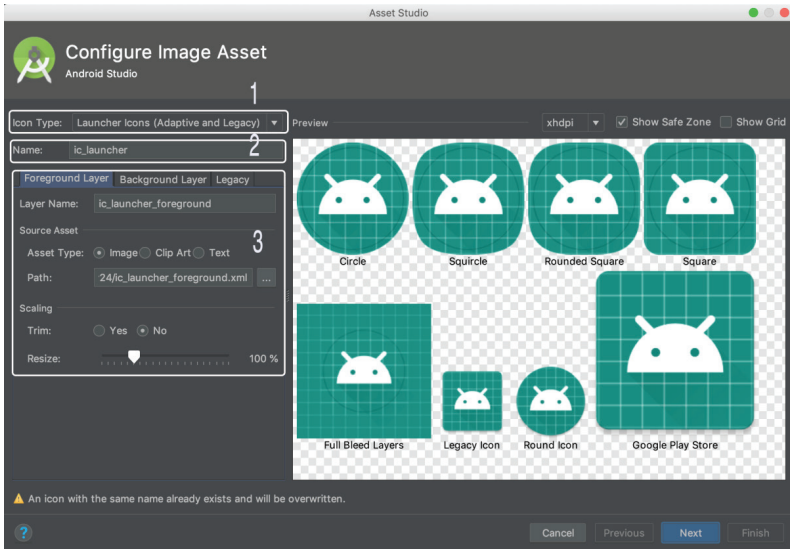
1. ملف للأيقونة بشكل اعتياديّ (مربّع)، ومسمّى باسم ic_launcher.
2. ملف للأيقونة بشكلّ دائري تظهر إذا كانت إعدادات الهاتف تُظهر الأيقونة بهذا الشكل، ومسمّى باسم ic_launcher_round.

لتعديل الأيقونة أو عمل أيقونة جديدة في الأندرويد ستوديو أتتبّع الخطوات الآتية:

- **أوّلاً:** أضغط بزر الفارة الأيمن على ملف المشروع (app) وأختار (New)، ثمّ (Image Asset) كالآتي:



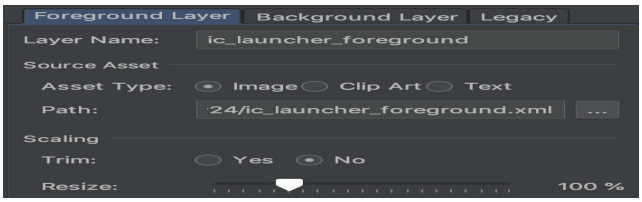
- **ثانياً:** تظهر النافذة الآتية توضّح أهمّ عناصر الأيقونة:



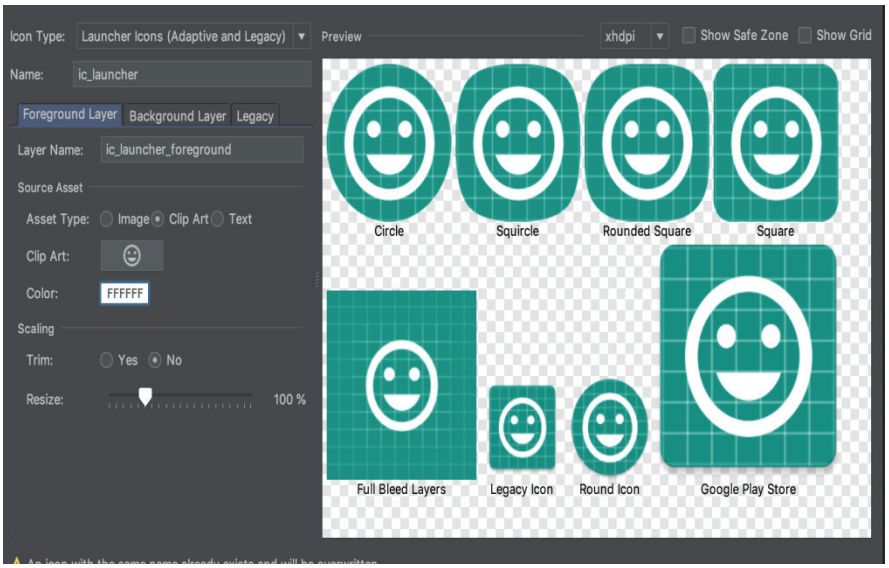
1. نوع الأيقونة (Icon Type) أترك الخيار كما هو: Launcher Icons (Adaptive and Legacy)
2. اسم الأيقونة (Name): يمكن تغيير اسم الأيقونة مع مراعاة عدم احتواء الاسم على أحرف كبيرة، وتجنُّب استخدام الأرقام.....، أو إبقاء الاسم كما هو، وعمل استبدال للأيقونة الافتراضيّة.
3. الشكل العام للأيقونة، ويتكوّن من قسمين رئيسيّين :

- المقدّمة Foreground إمّا أن تكون:

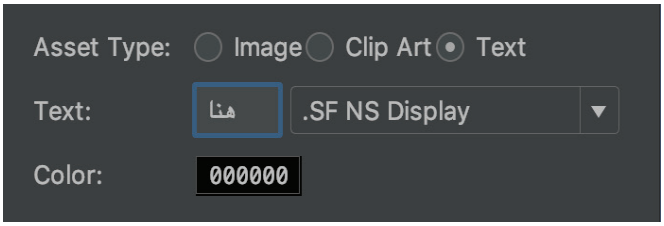
• صورة Image يتمّ تغييرها بالضغط على ... ، واختيار صورة من مكانٍ ما على الجهاز، مثل سطح المكتب:



• قصاصة فنيّة Clip Art، يتمّ تغييرها بالضغط على ، واختيار القصاصة المناسبة للتطبيق، ويمكن تغيير لون القصاصة بالضغط على 000000، واختيار اللون المناسب كما يأتي:

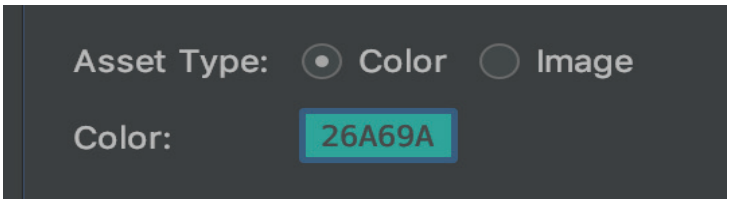


• نصّ Text، كتابة النصّ في الخانة المخصّصة، ويمكن تغيير نوع الخطّ من الخيارات التي بجانب مربّع النصّ، وتغيير اللون كما تمّ تغييره في النقطة السابقة:

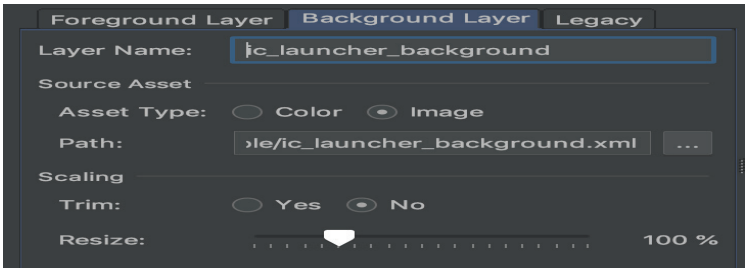


- الخلفيّة Background، إمّا أن تكون:

• لون Color ويمكن تغييره بالضغط على رمز اللون فتظهر نافذة، ثمّ اختيار اللون المناسب، كالآتي:

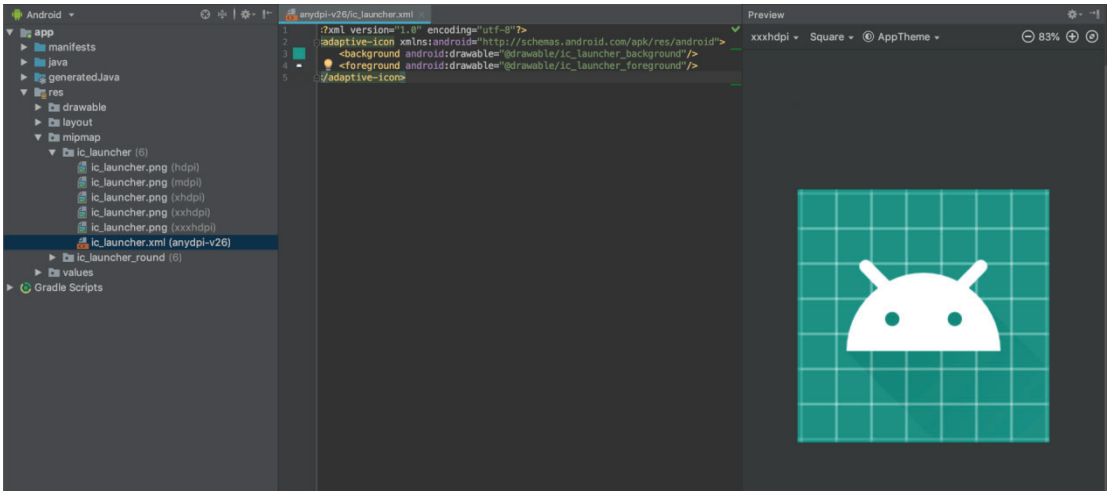


• صورة Image، ويمكن تغييرها بالضغط على ...، ثم اختيار الصورة:



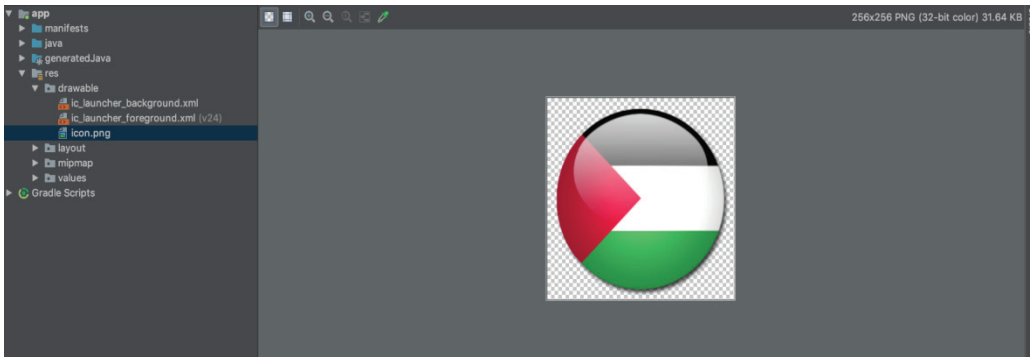
- **ثالثاً:** أضغط next ثم finish، ويتمّ تخزين الأيقونة في مجلّد mipmap بالاسم الذي تمّ اختياره سابقاً.

الهدف من إعداد الأيقونة بهذا الشكل في الأندرويد ستوديو؛ أي وجود مقدّمة وخلفيّة في الأيقونة، لتتوافق مع نظام التشغيل Oreo الذي تتحرّك فيه مقدّمة الأيقونة، مع ثبات الخلفيّة عند تحريك الجهاز المحمول؛ لهذا السبب ملف الأيقونة لنظام التشغيل Oreo ليس صورة إنّما ملف xml مكوّن من طبقتين المقدّمة والخلفيّة، كالآتي:

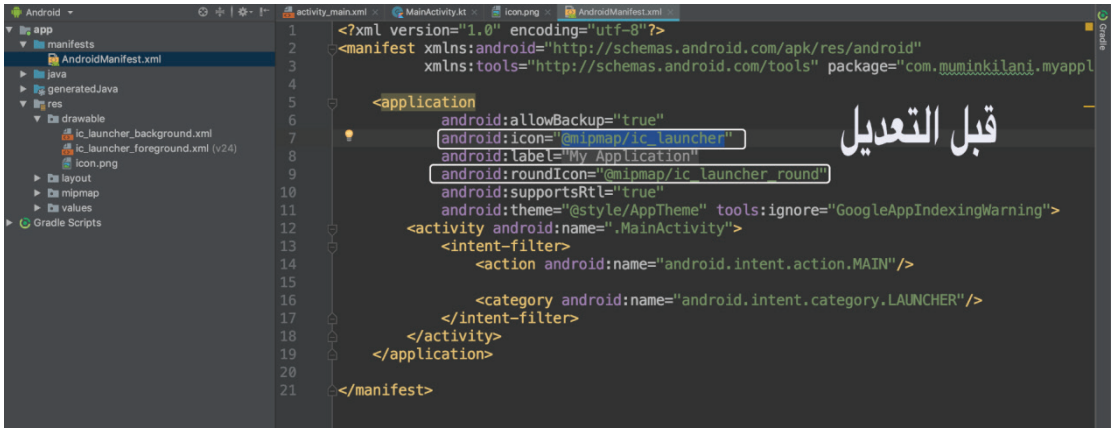


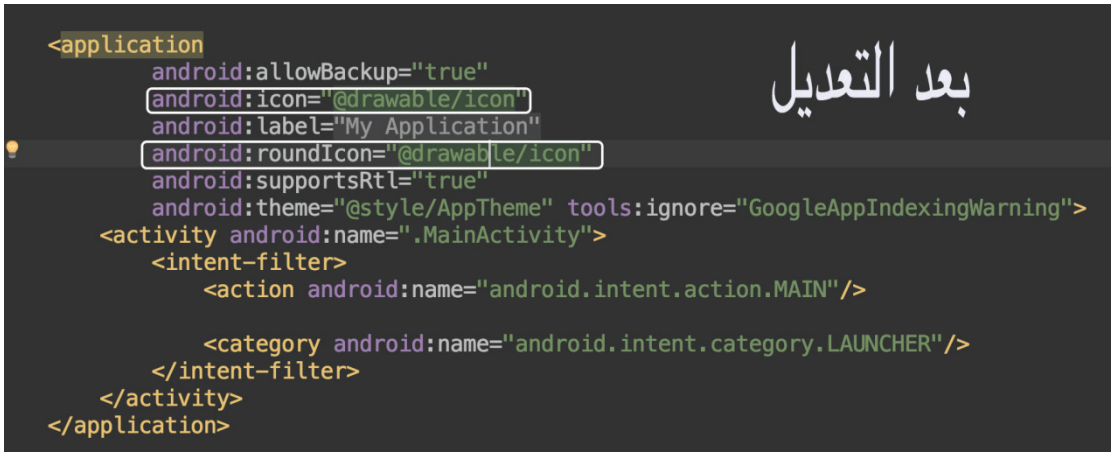
نشاط (4) عملي: أُنشئ مشروع أندرويد جديداً، ثمّ أُعدّل أيقونة التطبيق.

- **أولاً:** أُعدّل أيقونة التطبيق بإضافة صورة في مجلّد الصور drawable، ثمّ تغييرها من ملف manifests:
  - أنسخ الأيقونة المرادة، وألصقها في مجلّد drawable كالآتي:



  - أنتقل إلى ملف AndroidManifest.xml، ثمّ أعدّل أيقونة التطبيق كالآتي:

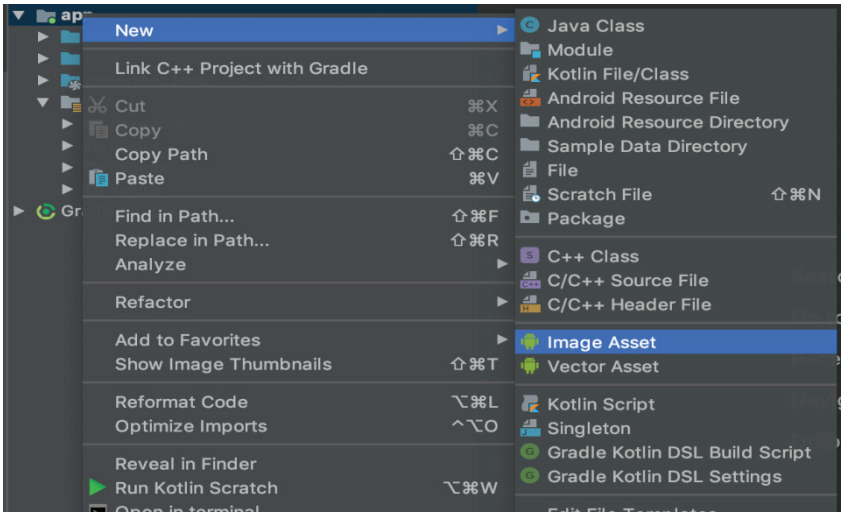




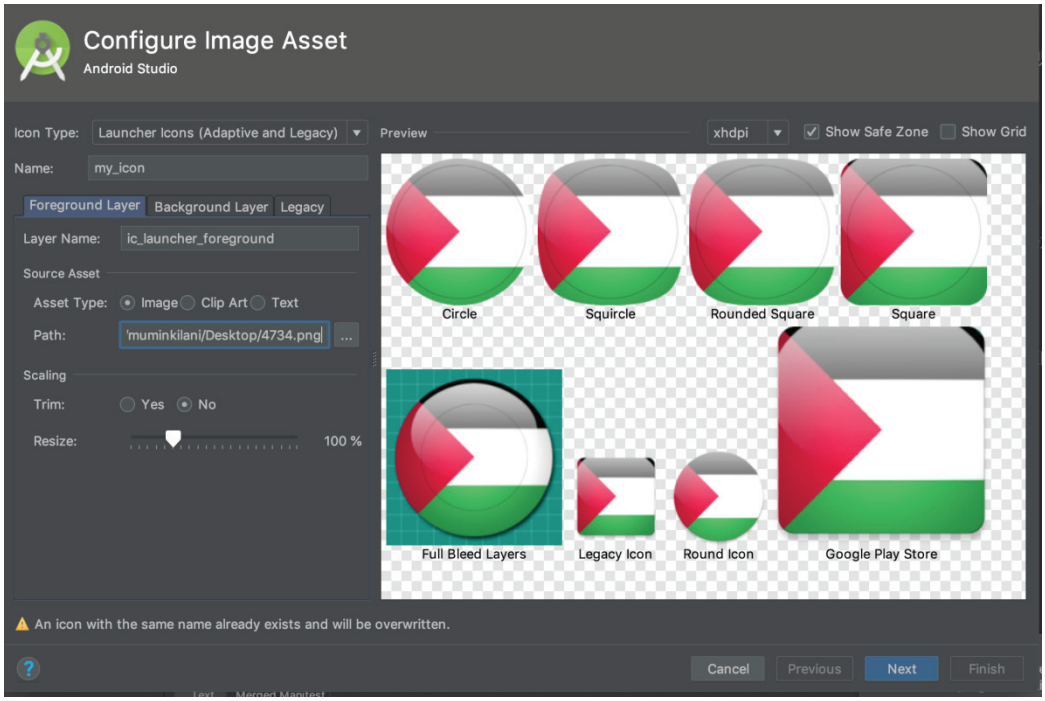
  - أختبر المشروع، وألاحظ تغيير الأيقونة:

- **ثانياً:** أعدّل أيقونة التطبيق من خلال مجلّد mipmap:

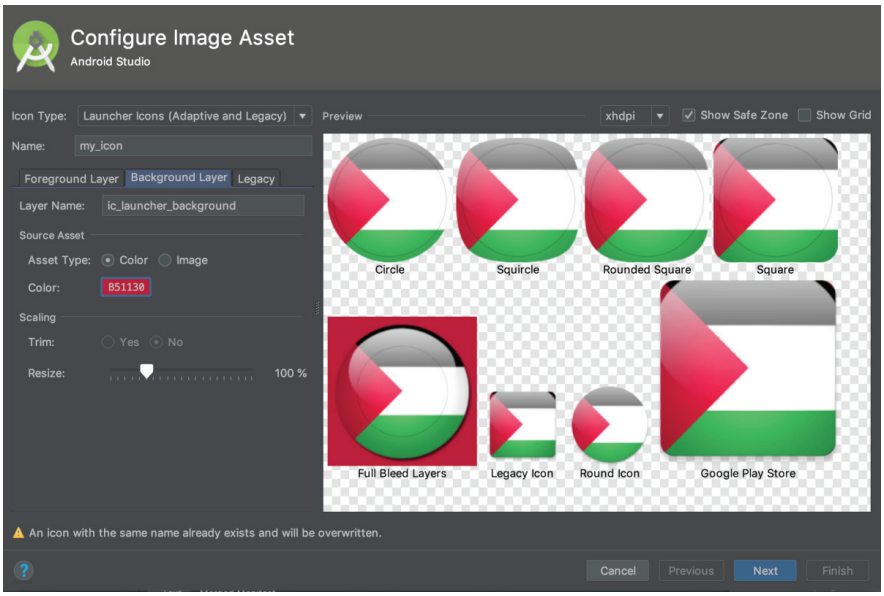
• أضغط بزر الفارة الأيمن على ملف المشروع (app)، وأختار (New)، ثمّ (Image Asset) كالآتي:



• أُعدّل اسم الأيقونة لِـ (my_icon)، ثمّ أعدّل مقدّمة الأيقونة باختيار الصورة التي أريد جعلها أيقونة للتطبيق، كالآتي:

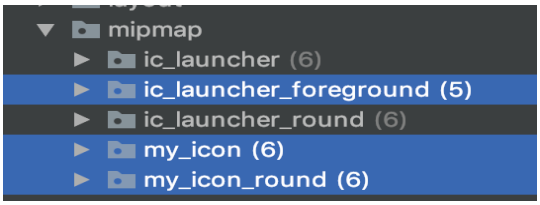


- أُعدّل خلفيّة الأيقونة بوضع لون مناسب، كالآتي:

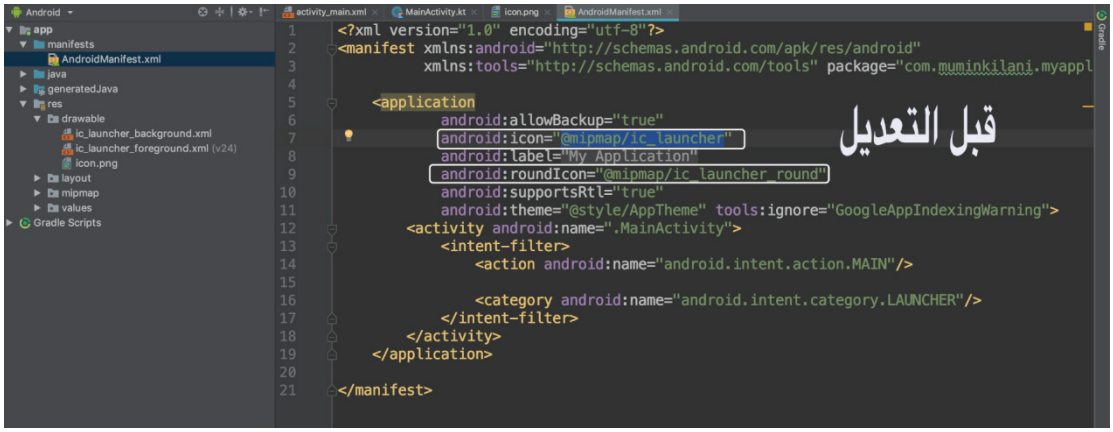


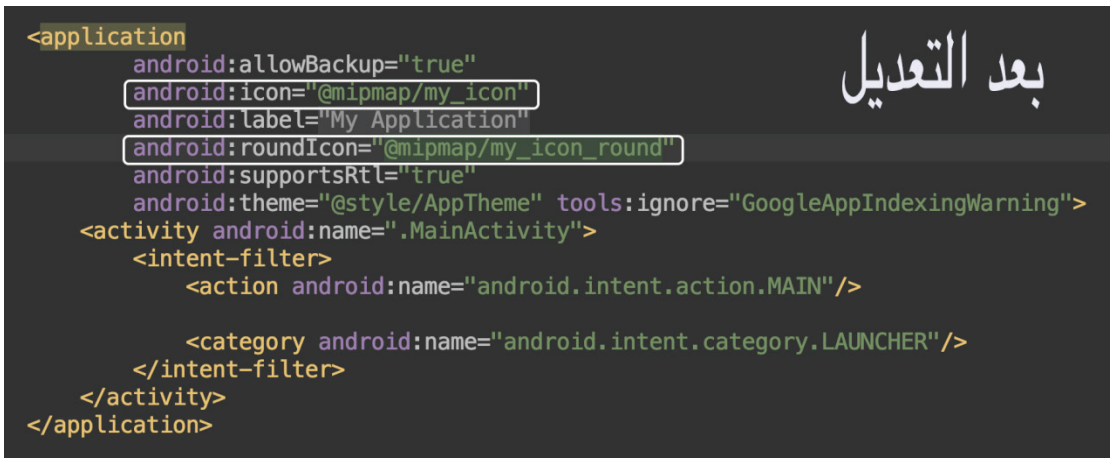
- أُلاحظ أنّ حجم الصور كبير، والخلفيّة غير ظاهرة؛ لذلك أُغيّر حجمها بواسطة التحكُّم بشريط الحجم(Resize)، أو باختيار yes من خيارات Trim، حتى تصبح مناسبة كالآتي:

- أضغط next ثمّ finish، وألاحظ إنشاء ملفّين جديدين باسم my_icon، وملف للصورة المقدّمة، كالآتي:

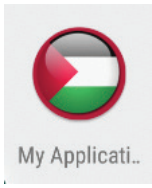


- أنتقل إلى ملف AndroidManifest.xml، ثمّ أُعدّل أيقونة التطبيق، كالآتي:





- أُشغّل المشروع، وأُلاحظ تغيير الأيقونة:



نشاط (5) عملي: أُنشئ مشروعاً جديداً، وأُعدّل أيقونة التطبيق بحيث تكون المقدّمة نصّاً والخلفيّة صورة.

الأسئلة

1. أُعدّد أحجام أيقونة التطبيق وفق دقّة شاشة الهاتف الذكيّ.
2. ما أهمّ عناصر الأيقونة؟

2-4 الموقف التعليميّ التعلُّميّ

إعداد التطبيق بأكثر من لغة.

وصف الموقف التعليميّ: استشار مطوّر تطبيقات أندرويد مشغل تطبيقات المحمول في مدرسة اللّد المِهنيّة؛ لمساعدته في جعل تطبيقه يدعم عدّة لغات.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلّم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من المطوّر عن التطبيق من حيث: - ملف المشروع. - اللغات المراد دعمها في التطبيق. • أجمع البيانات عن: - كيفيّة جعل التطبيق يدعم عدّة لغات. - الرموز الخاصّة باللغات.	• حوار ومناقشة. • التعلّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب المطوّر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة. حاسوب، جهاز محمول).
أخطّط وأقرّر	• أُصنّف البيانات الخاصّة بملفّ المشروع واللغات المراد دعمها. • أحدّد: - الكلمات والجمل المراد ترجمتها. - اللغات المراد دعمها. - الرموز الخاصّة باللغات المراد دعمها. - آليّة جعل التطبيق يدعم تعدُّد اللغات. • إعداد خطّة للتنفيذ.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• وثائق (طلب المطوّر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد، كتب). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أنفّذ	• أُجهّز الجمل المترجمة للّغات المطلوبة. • أضيف الملفّات المناسبة للمشروع (strings.xml). • أُضيف واجهات التصميم التي تدعم اللغات المختلفة. • أُعيد ترتيب عناصر الواجهات بما يتناسب مع اللغات. • أُنفّذ التطبيق على جهاز أندرويد يدعم اللغات المراد الترجمة إليها.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة. • عصف ذهنيّ/ استمطار الأفكار.	• الطلب الخاص بالمطوّر. • حاسوب، برنامج خاصّ (android studio)، جهاز افتراضيّ، أو جهاز يعمل بنظام أندرويد.

أتحقّق	• أتحقّق من تغيُّر لغة التطبيق عند تغيُّر لغة الجهاز. • أتحقّق من ملاءمة التغيير لطلب المطوّر.	• تعلّم تعاونيّ/ مجموعات. • حوار ومناقشة.	• الخطّة المعدّة سابقاً. • الطلب الخاص بالمطوّر. • حاسوب، إنترنت • هاتف محمول.
أوثّق وأقدّم	• أوثّق كيفيّة جعل التطبيق يدعم عدّة لغات. • أوثّق الرموز الخاصّة باللغات. • إعداد العروض التقديميّة. • إعداد ملف بالحالة (جعل التطبيق يدعم عدّة لغات).	• حوار ومناقشة. • تعلُّم تعاونيّ/ مجموعات عمل.	• حاسوب. • شاشة عرض. •
أقوّم	• رضا المطوّر عن التطبيق الناتج بما يتلاءم مع الطلب. • ملاءمة التطبيق بعد تطويره للمعايير والمواصفات.	• حوار ومناقشة.	• نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد يهتم بواجهات التطبيق وعناصرها التي تدعم أكثر من لغة.

أناقش مع زملائي أثر دعم التطبيقات للّغات المختلفة على انتشار التطبيق.

أتعلّم: إعداد التطبيق بأكثر من لغة

نشاط (1) للاطلاع: أتناقش مع زملائي حول تطبيقات عالميّة تدعم اللغة العربيّة، وأثر ذلك على استخدامنا لها.

غالبيّة الناس يعتزّون بلغتهم الأصليّة، ويفضّلون استخدامها على استخدام لغة أجنبيّة في شتّى مجالات الحياة: في المعاملات، وفي العمل، وفي مواقع التّواصل، وأيضاً في استخدامهم التطبيقات المختلفة؛ لذلك من المهمّ جدّاً لمبرمجي تطبيقات الأندرويد الأخذ بعين الاعتبار المنطقة الجغرافيّة التي يستهدفها التطبيق، لمعرفة أيّ اللغات إضافتها للتطبيق أنسب.

وبشكلٍ عام يمكن إنشاء التطبيق باللّغتين الرئيسيّتين العربيّة والإنجليزيّة، فالأولى تستهدف سكّان الوطن العربيّ والعرب في شتّى أنحاء العالم، أمّا الثانية فهي تستهدف فئة كبيرة من العالم؛ لأنّ معظم سكّان العالم إن لم تكن لغتهم الأساسيّة هي الإنجليزيّة، فإنّهم يتعاملون بهذه اللغة في المدارس والجامعات.

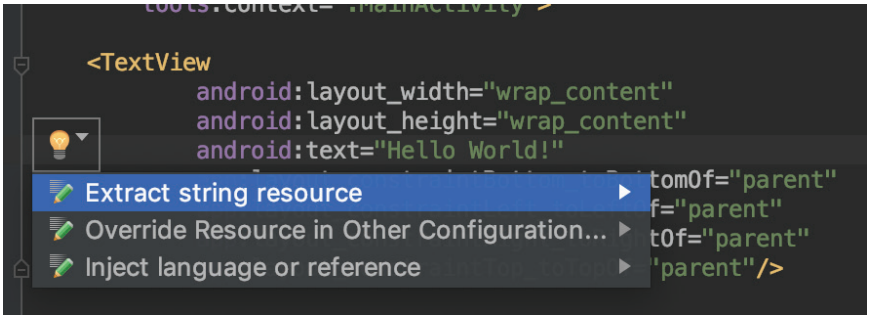
لإعداد التطبيق بأكثر من لغة، تُضاف ملفّات لدعم هذه اللغات على:

1. الملف الخاص بالكلمات المستخدمة في التطبيق، وهو ملف Strings.xml.
2. ملفّات التصميم الموجودة داخل مجلّد layout.

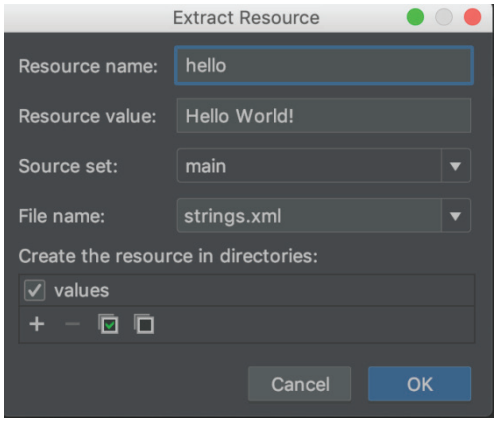
عمليّات الإضافة في ملفّ strings.xml:

- **أوّلاً:** إضافة كلمات من ملفّ تصميم الواجهة إلى ملف strings.xml:

1. التصميم: أفتح الكود xml الخاص بالتصميم.
2. أضغط على إشارة الموجودة عند النصّ المكتوب.
3. أختار الخيار الأول كما يأتي:



4. أكتب في النافذة داخل الخانة الأولى (Resource name) كلمة مفتاحيّة تُستخدَم للتعامل مع النصّ الموجود، والخانة الثانيه هي النصّ المكتوب الذي يظهر للمستخدم، والخانة الأخيرة هي مكان تخزين الكلمة في ملف srings.xml. الافتراضيّ، كالآتي:



5. أنتقل إلى ملف strings.xml، وألاحظ تخزين الكلمة، كما يأتي:

```
<string name="hello">Hello World!</string>
```

• **ثانياً:** إضافة الكلمات مباشرة في ملف strings.xml:

الطريقة الأولى من ملف كود xml:

1. أفتح ملف strings.xml.
2. أكتب كالآتي:

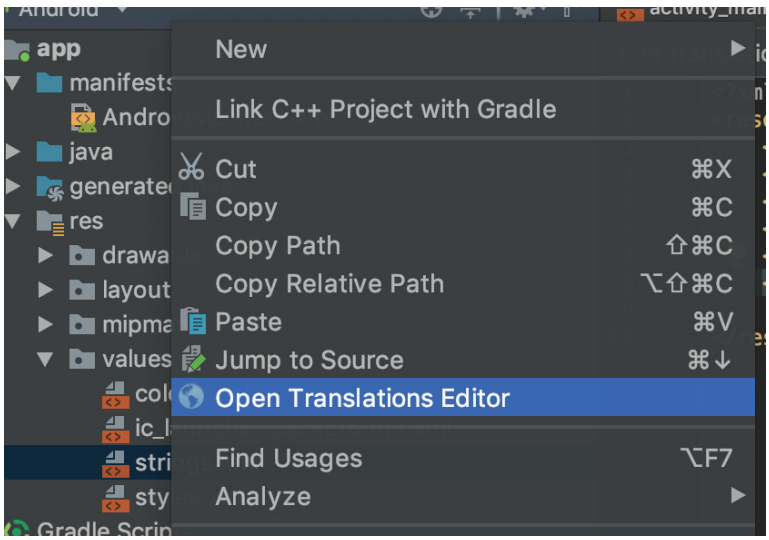
```
<string name="width">Width</string>
```

- name هي عباره عن مفتاح للوصول للنصّ من أماكن مختلفة.
- الكلمة التي تقع بين <string name= > ..... </string> هي الكلمة التي تظهر في التطبيق.

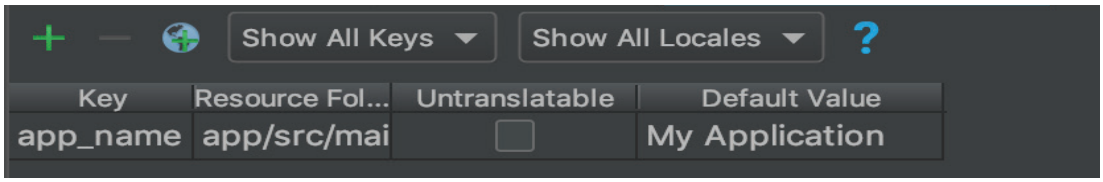
3. يمكن إضافة جميع الكلمات والعبارات بالطريقة نفسها مع اختلاف name.

الطريقة الثانية من محرّر النصوص:

1. أضغط بزر الفأرة اليمين على ملف strings.xml، وأختار Open Translations Editor، كالآتي:

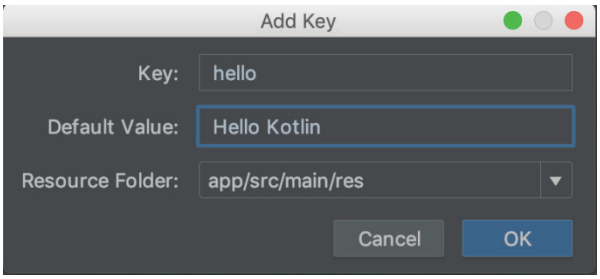


2. تظهر النافذة الآتية، وهي عبارة عن جدول يتكوّن من أربعة أعمدة:



- العمود الأول هو key مفتاح القيم الذي يُسمّى في كود xml (name).
- العمود الثاني مسار تخزين ملف string.
- العمود الثالث خيار يتيح عدم ترجمة النصّ إلى لغات أخرى.
- العمود الرابع وهو القيمة؛ أي النصّ الذي سيظهر في التطبيق.

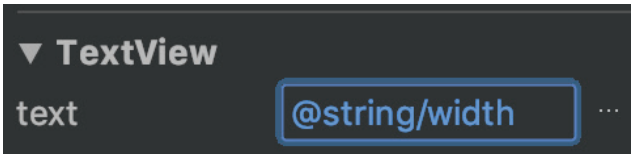
3. أضغط على إشارة + تظهر نافذة، أكتب المفتاح والقيمة، ثمّ ok كما يأتي:



4. تتمّ إضافة أيّ قِيم بالطريقة نفسها.

استدعاء الكلمات من ملف strings.xml:

- **أوّلاً:** في ملف التصميم، لاستخدام أي عبارة أو كلمة من ملف strings.xml، يُكتب كما يأتي:



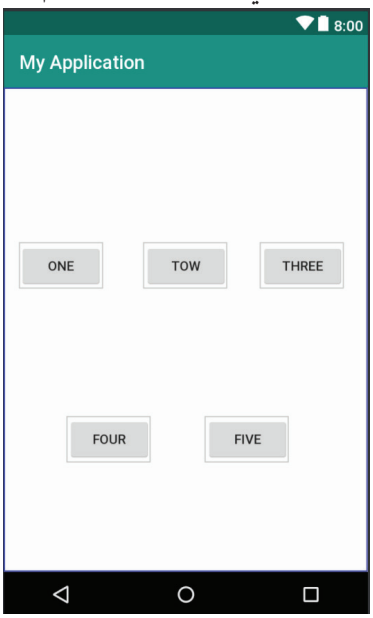
تُستخدم @ حتى أصل إلى ملفات resource، أكتب اسم الملف string ثمّ (/)، ثمّ اسم المفتاح (name) الذي يوصل إلى النصّ المطلوب.

- **ثانياً:** في ملف الكود، لاستخدام أي عبارة أو كلمة من ملف strings.xml، يُكتب كما يأتي:

```
textView.text = getText(R.string.app_name)
```

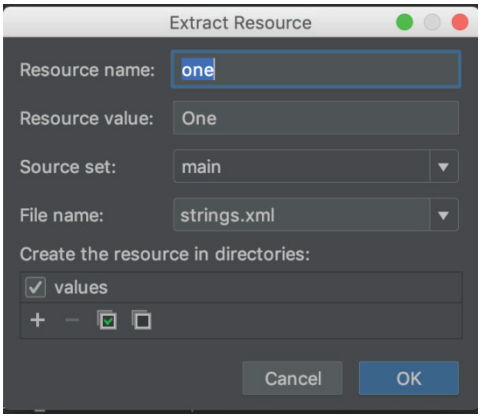
الدالّة getText تجلب النصّ من مصدره، وهنا المصدر هو (R تشير لمجلد resource، ثمّ نقطة(.)، ثمّ string الملف الذي يحتوي على الكلمات، ثمّ نقطة (.)، ثمّ اسم المفتاح (name) الذي يوصل إلى النصّ المطلوب.

نشاط (2) عملي: أصمّم واجهة تطبيق، كما في الصورة الآتية تدعم اللغتين العربيّة والإنجليزيّة:



- **أولاً:** أنشئ مشروعاً جديداً، ثمّ أضيف العناصر كما في الصورة، أكتب باللغة الإنجليزيّة؛ لأنّ ملف strings.xml الافتراضيّ هو للّغة الإنجليزيّة.

- **ثانياً:** أضيف الكلمات التابعة لكلّ عنصر في ملف strings.xml، أنتقل لكود التصميم، أضغط على الإشارة، ثمّ أكتب في خانة name ما يدلّ على الكلمة للتسهيل، كالآتي:



- **ثالثاً:** أفتح ملف strings.xml، وأتأكّد من الإضافة كما يأتي:

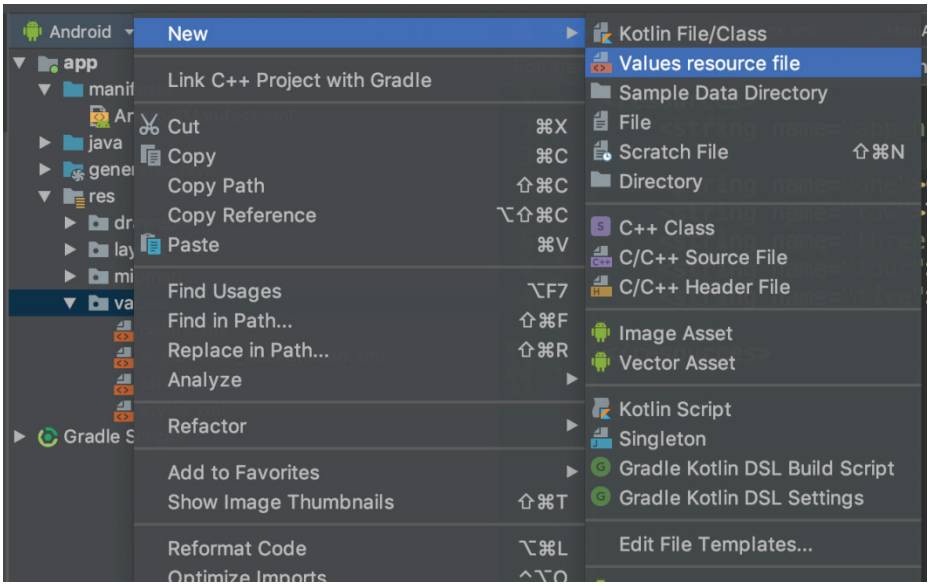
```
<resources>
    <string name="app_name">My Application</string>

    <string name="one">One</string>
    <string name="tow">TOW</string>
    <string name="three">THREE</string>
    <string name="four">FOUR</string>
    <string name="five">FIVE</string>

</resources>
```

My Application هو اسم التطبيق، ويمكن تغييره من هنا بحذف كلمة (My Application) واستبدالها.

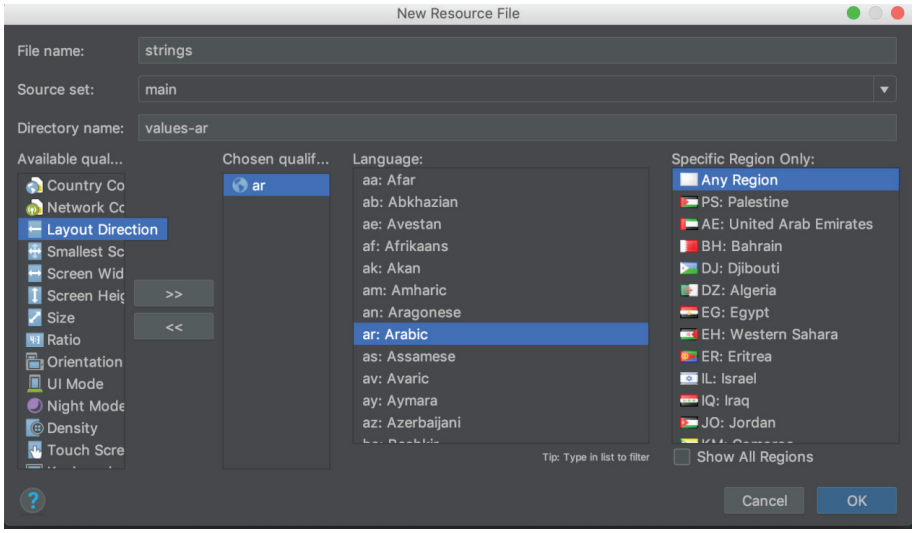
• **ثالثاً:** أنشئ ملف string جديداً لإضافة الكلمات باللغة العربيّة، ويمكن إنشاؤه بإحدى الطُّرق الآتية:

الطريقة الأولى من ملف كود xml:

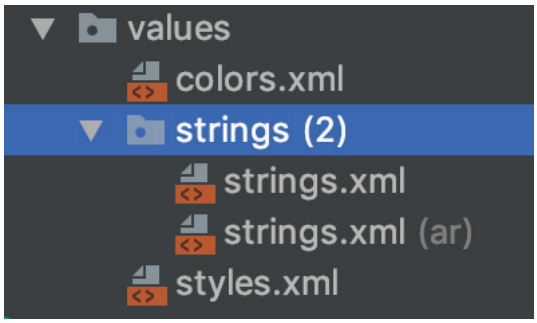
1. أضغط برز الفأرة اليمين على مجلّد values، ثمّ أختار new، ثمّ values resource file، كالآتي:



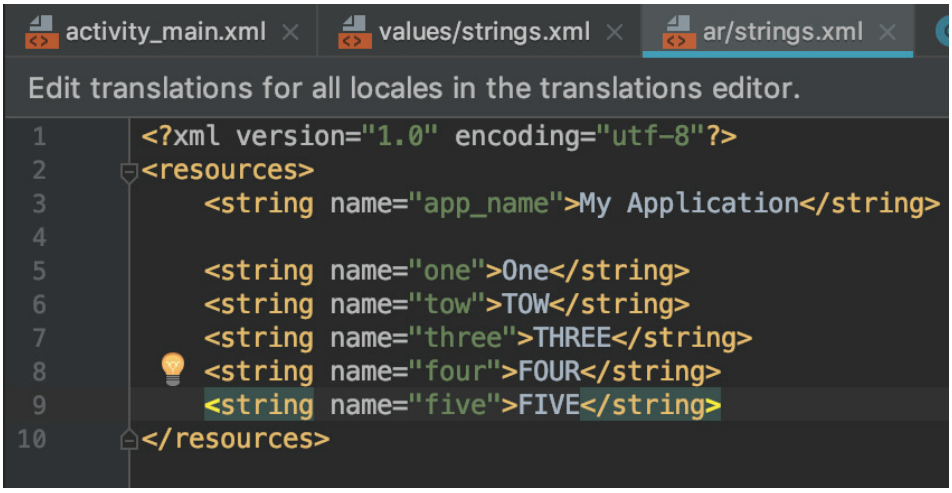
2. تظهر نافذة، أكتب اسم الملف strings ومن قائمة Available qualifiers أختار Locale، ثمّ أضغط على الإشارة >>، تظهر قائمة اللغات أختار اختصار اللغة العربيّة (ar:Arabic)، وأتأكّد من اختيار Any Region ليعمل التطبيق باللغة العربيّة في جميع البلدان، ثمّ أضغط ok، كما يأتي:



3. تمّ إنشاء ملف جديد باسم strings.xml (ar)، ووضعه مع الملف الافتراضيّ في مجلّد اسمه strings، كالآتي:



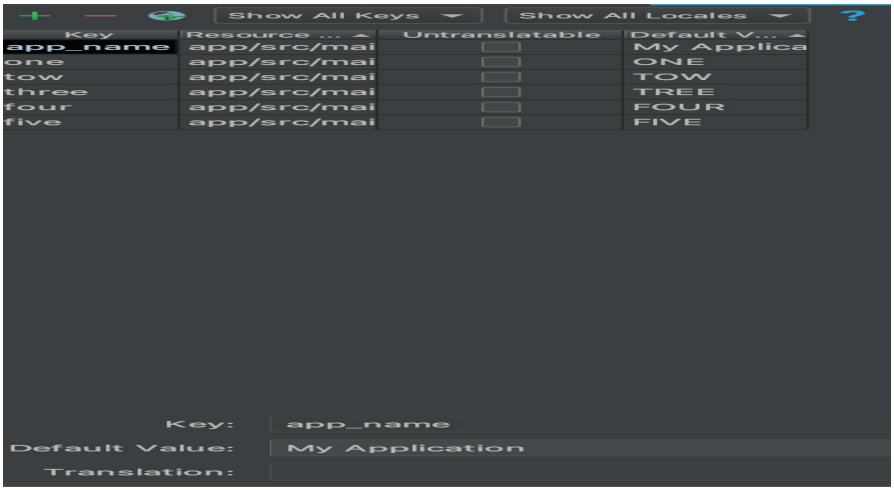
4. أفتح ملف strings.xml (ar) وأنسخ محتويات الملف الافتراضيّ بداخله كما يأتي:



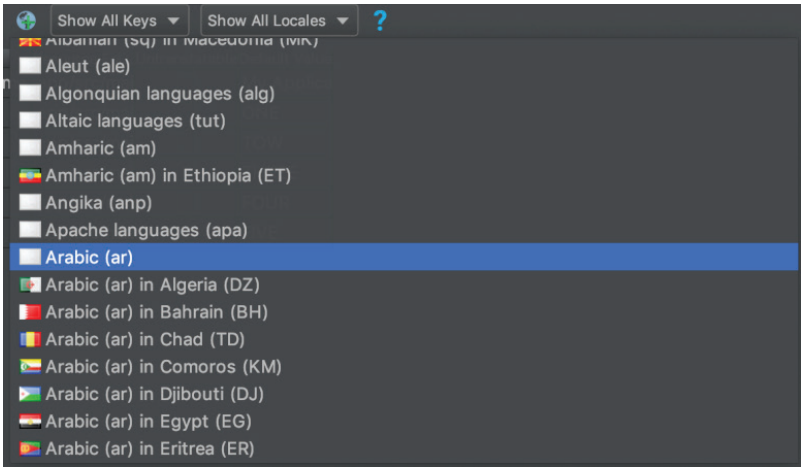
5. أعدّل على الكلمات لتصبح باللغة العربيّة، مع إبقاء المفتاح name كما هو، كالآتي:

الطريقة الثانية من محرّر النصوص:

1. أفتح محرّر النصوص كما تعلّمت سابقاً.
2. عند ظهور النافذة الآتية، أضغط على :



3. أختار من الخيارات التي تظهر Arabic (ar)، كالآتي:



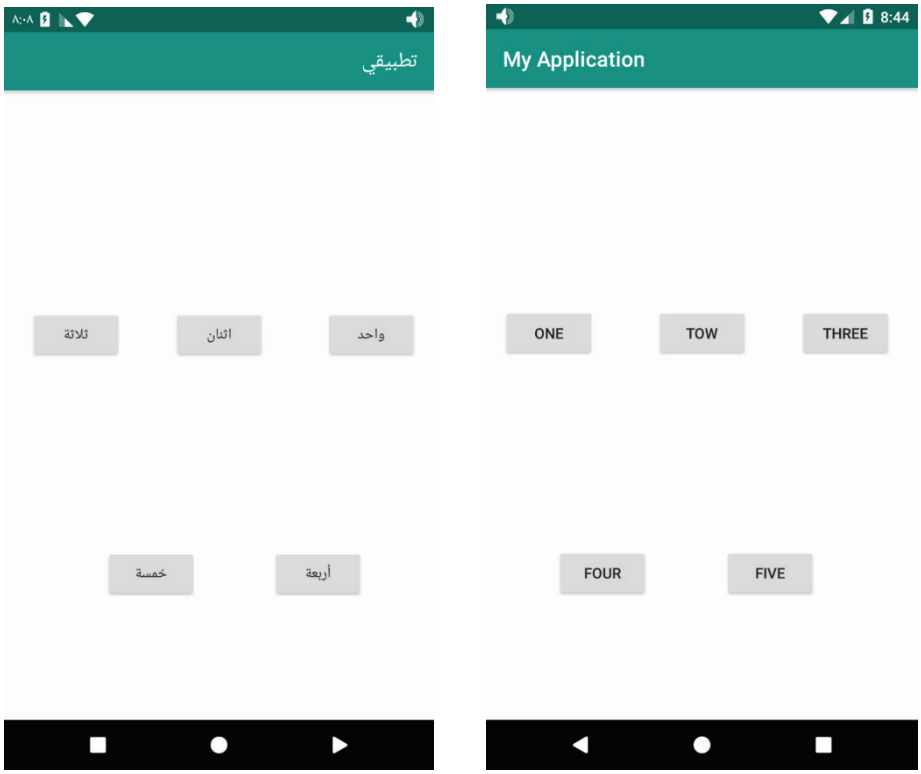
4. تمّ إنشاء ملف جديد باسم strings.xml (ar) ووضعه مع الملف الافتراضيّ في مجلّد اسمه strings، وإضافة عمود جديد للّغة العربية كما يأتي:

Key	Resource Fol...	Untranslatable	Default Value	Arabic (ar)
app_name	app/src/mai	☐	My Applica	My Applicat
one	app/src/mai	☐	ONE	
tow	app/src/mai	☐	TOW	
three	app/src/mai	☐	THREE	
four	app/src/mai	☐	FOUR	
five	app/src/mai	☐	FIVE	

5. أُضيف الكلمات العربيّة في عمود اللغة العربيّة بالضغط المزدوج في الفراغ، ثمّ أكتب فيصبح كما يأتي:

Key	Resource Fol...	Untranslatable	Default Value	Arabic (ar)
app_name	app/src/mai	☐	My Applica	تطبيقي
one	app/src/mai	☐	ONE	واحد
tow	app/src/mai	☐	TOW	اثنان
three	app/src/mai	☐	THREE	ثلاثة
four	app/src/mai	☐	FOUR	أربعة
five	app/src/mai	☐	FIVE	خمسة

• **رابعاً:** أُشغّل التطبيق، ثمّ أعدّل لغة الجهاز عربيّة تارةً، وإنجليزيّة تارةً أخرى، وألاحظ النتائج:

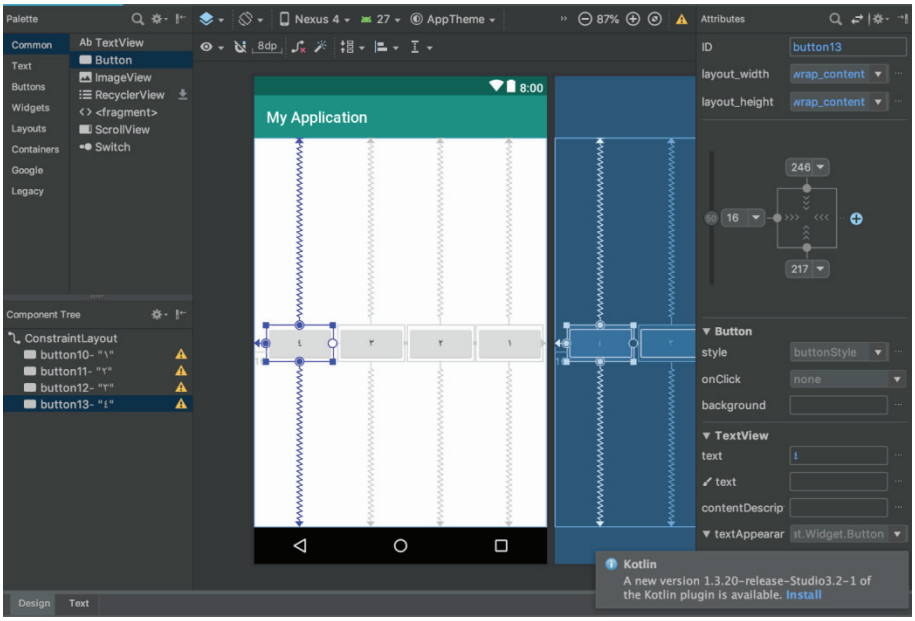


تعديل ملف layout ليتوافق مع عدّة لغات:

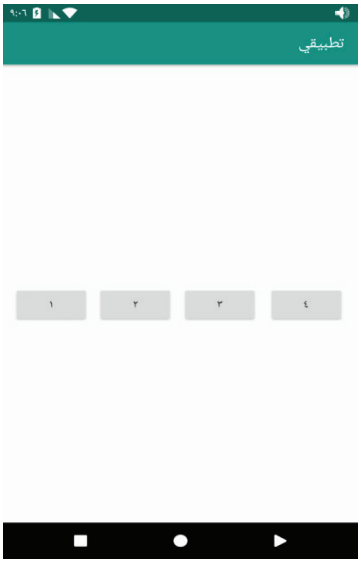
يوجد تطبيقات تُعرض بلغة واحدة مثل العربيّة، وإذا تمّ تشغيلها على جهاز لغته الأساسيّة الإنجليزيّة يتغيّر شكل التطبيق، وتتبدّل أماكن العناصر، فيصبح اليمين يساراً والعكس، ولتجنُّب هذه المشكلة يتمّ إنشاء layout بالاسم نفسه لدعم اللغة المطلوبة، ويُعاد ترتيب العناصر بالشكل الصحيح.

نشاط (3) عملي: أصمّم تطبيقاً يُظهر الأرقام من واحد إلى أربعة، مرتّبة تصاعديّاً من اليمين إلى اليسار، ويدعم الأجهزة التي تعمل بجميع اللغات.

- **أولاً:** أُنشئ مشروعاً جديداً، ثمّ أضيف كما يأتي:

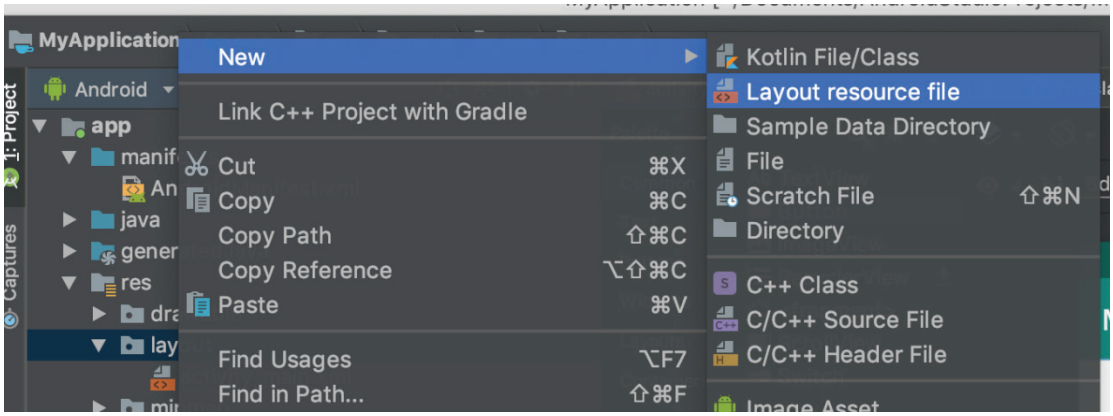


- **ثانياً:** أعمل اختباراً للمشروع على جهاز يعمل باللّغة العربيّة، وأُلاحظ أنّ العناصر مرتّبة تنازليّاً وليس تصاعديّاً، كما يأتي:

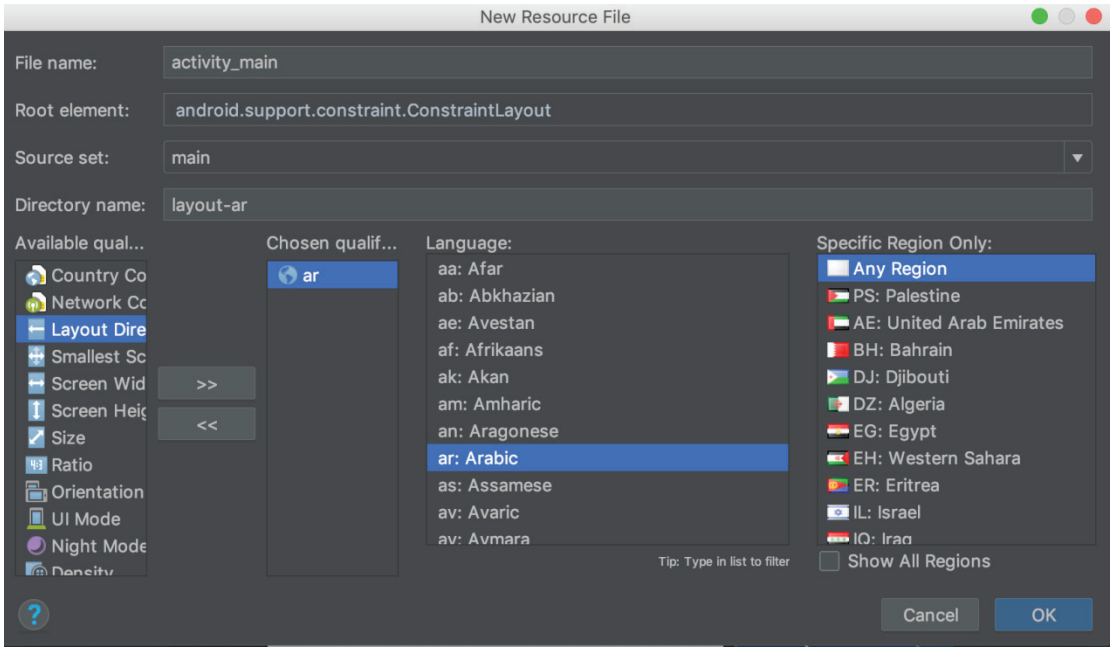


• **ثالثاً:** لتفادي هذه المشكلة:

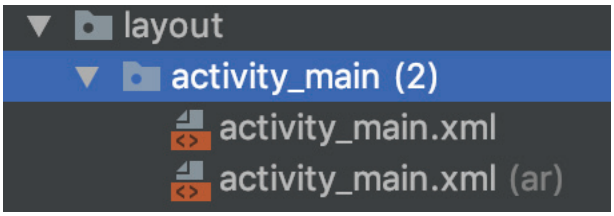
1. أنشئ ملف تصميمٍ جديداً بالضغط بزر الفارة اليمين على مجلّد layout، ثمّ new، ثمّ layout resource file كما يأتي:



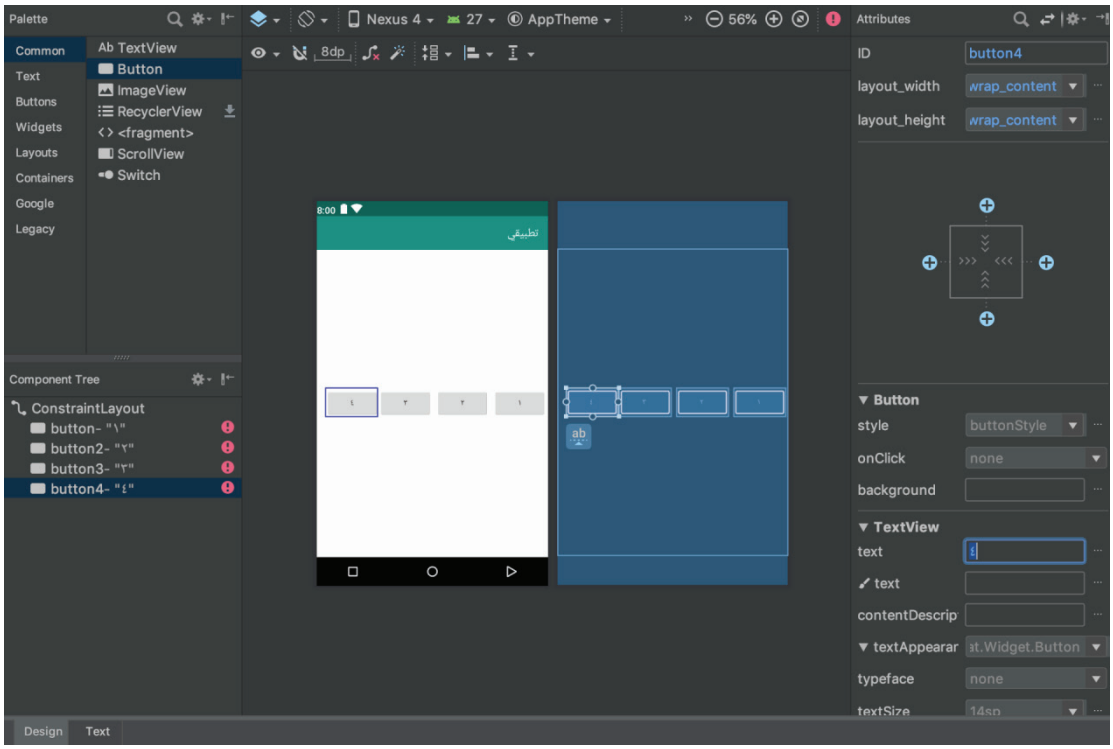
2. أكتب في النافذة اسم ملف التصميم نفسه الذي أريده أن يدعم اللغة العربيّة، ومن قائمة Available qualifiers أختار Locale ، ثمّ أضغط على الإشاره >> ، تظهر قائمة اللغات فأختار اختصار اللغة العربيّة (ar:Arabic)، أتأكّد من اختيار Any Region ليعمل التطبيق باللغة العربيّة في جميع البلدان، ثمّ أضغط ok، كما يأتي:



3. تمّ إنشاء ملف تصميم باسم الملف القديم نفسه، ووضع الملفين في مجلّد كما يأتي:



4. أفتح ملف التصميم الذي يدعم اللغة العربيّة، وأُضيف العناصر كما يأتي:



- **رابعاً:** أشغّل المشروع، وأختبر التطبيق عند تغيير لغة الجهاز لأكثر من لغة.

ملحوظة: يُرجى الانتباه عند عمل أكثر من ملف تصميم بالاسم نفسه ليدعم لغة معيّنة، وأن تأخذ العناصر الـ id نفسه، ليتمّ التعامل معها في الكود البرمجيّ باستخدام id واحد لكلّ عنصر، وعند عمل تشغيل على عدّة أجهزة بلغات مختلفة، لا تظهر مشاكل أو أعطال.

نشاط (4) نظري: أناقش مع زملائي فوائد تعديل ملف layout ليتوافق مع عدّة لغات.

الأسئلة

1. أين تُضاف الملفّات ليدعم التطبيق أكثر من لغة؟
2. أوضّح كيف يتمّ استدعاء نصّ من ملف strings.xml؟

4-4 الموقف التعليميّ التعلُّميّ

إعداد التطبيق للعمل على أجهزة الأندرويد المختلفة.

وصف الموقف التعليميّ: زار مطوّر تطبيقات أندرويد مشغل تطبيقات المحمول في مدرسة اللّد المِهنيّة، لمساعدته في جعل تطبيقه يدعم جميع الأجهزة التي تعمل بنظام الأندرويد.

العمل الكامل			
خطوات العمل	وصف الموقف الصّفيّ	المنهجيّة (استراتيجيّة التّعلّم)	الموارد
أجمع البيانات وأحلّلها	• أجمع البيانات من المطوّر عن التطبيق من حيث: - ملف المشروع. • أجمع البيانات عن: - أحجام الشاشات. - واجهات التصميم (layout) المراد إضافتها.	• حوار ومناقشة. • التعلُّم التعاونيّ/ مجموعات. • البحث العلميّ.	• وثائق (طلب المطوّر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا (روابط إلكترونية من مصادر موثوقة، حاسوب، جهاز محمول).
أخطّط وأقرّر	• أُصنّف البيانات الخاصّة بملف المشروع. • أحدّد: - أحجام الشاشات المختلفة. - كيفيّة التعامل مع أحجام الأجهزة المختلفة. - واجهات التصميم المراد إضافتها. - إعداد خطّة للتنفيذ.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• وثائق (طلب المطوّر، نشرات وكتب ومصادر عن كيفية تطوير تطبيق أندرويد). • التكنولوجيا: (روابط إلكترونية من مصادر موثوقة، حاسوب، هاتف يعمل بنظام الأندرويد).
أنفّذ	• أُضيف ملفات التصميم (layout) بأحجام شاشات مختلفة لكلّ واجهة. • أُجري التعديلات اللازمة على العناصر للتوافق مع أحجام الشاشات. • أُنفّذ التطبيق على أجهزة أندرويد مختلفة الحجم.	• تعلُّم تعاونيّ/ مجموعات. • حوار ومناقشة. • عصف ذهنيّ/استمطار الأفكار.	• الطلب الخاصّ بالمطوّر. • حاسوب، برنامج خاصّ (android studio)، أجهزة افتراضيّة، أو أجهزة تعمل بنظام أندرويد.

أتحقّق	• أتحقّق من توافق التطبيق مع جميع أحجام الأجهزة. • أتحقّق من ملاءمة التطبيق لطلب المطوّر.	• تعلُّم تعاونيّ/مجموعات. • حوار ومناقشة.	• الخطّة المعدّة سابقاً. • الطلب الخاصّ بالمطوّر. • حاسوب، إنترنت. • هاتف محمول.
أوثّق وأقدّم	• أوثّق كيفيّة جعل التطبيق يدعم أحجام الشاشات المختلفة. • أوثّق واجهات التصميم المراد إضافتها • إعداد العروض التقديميّة. • إعداد ملف بالحالة (جعل التطبيق يدعم عدّة لغات).	• حوار ومناقشة. • تعلم تعاونيّ/ مجموعات عمل.	• حاسوب. • شاشة عرض.
أقوّم	• رضا المطوّر عن التطبيق. • ملاءمة التطبيق بعد تحسينه للمعايير والمواصفات.	• حوار ومناقشة.	• نشرات عن المعايير والمواصفات الخاصة بتطوير تطبيقات الأندرويد يهتم بواجهات التطبيق وإعدادها بحيث تعمل على أجهزة أندرويد مختلفة.

نشاط (1) للاطلاع: أبحث عن أنواع الأجهزة التي تعمل بنظام الأندرويد.

نشاط (2) للاطلاع: أبحث عن تطبيق وأتأكّد من دعمه لأكثر من حجم شاشة.

بعض التطبيقات لا تتوافق مع جميع أحجام شاشات الأجهزة التي تعمل بنظام الأندرويد، وقد لا تظهر بالشكل المطلوب، كما هو موضّح في الشكل (1-6)، وقد لا تتوافق أيضاً مع وضعيّة قلب الشاشة؛ لذلك لا بدّ من جعل التطبيق المستخدم يتوافق مع عدد أكثر من أحجام الشاشات كي يزداد انتشاره بشكل أوسع؛ ولعمل ذلك تتمّ عمليّات الإضافة والتعديل على ملفّات واجهات التصميم (layout) من قبل المبرمج.

جهاز حجم كبير

جهاز حجم متوسط

ساعة يد

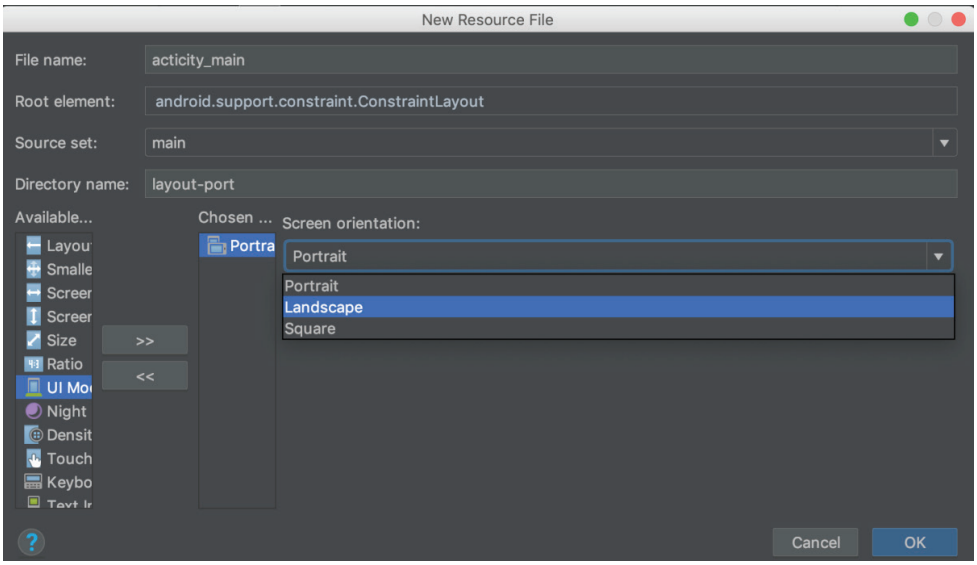
تُقسم أحجام الشاشات وفق دقّتها إلى:

1. شاشات ذات دقّة منخفضة بكثافة 120dpi ((ldpi (low)).
2. شاشات ذات دقّة متوسّطة بكثافة 160dpi ((mdpi (medium)).
3. شاشات ذات دقّة عالية بكثافة 240dpi ((hdpi (high)).
4. شاشات ذات دقّة أعلى من السابقة (hdpi) بكثافة 320dpi ((xhdpi (extra-high)).
5. شاشات ذات دقّة أعلى من السابقة (xhdpi) بكثافة 480dpi ((xxhdpi (extra-extra-high)).
6. شاشات ذات دقّة أعلى من السابقة (xxhdpi) بكثافة 640dpi ((xxxhdpi (extra-extra-extra-high)).

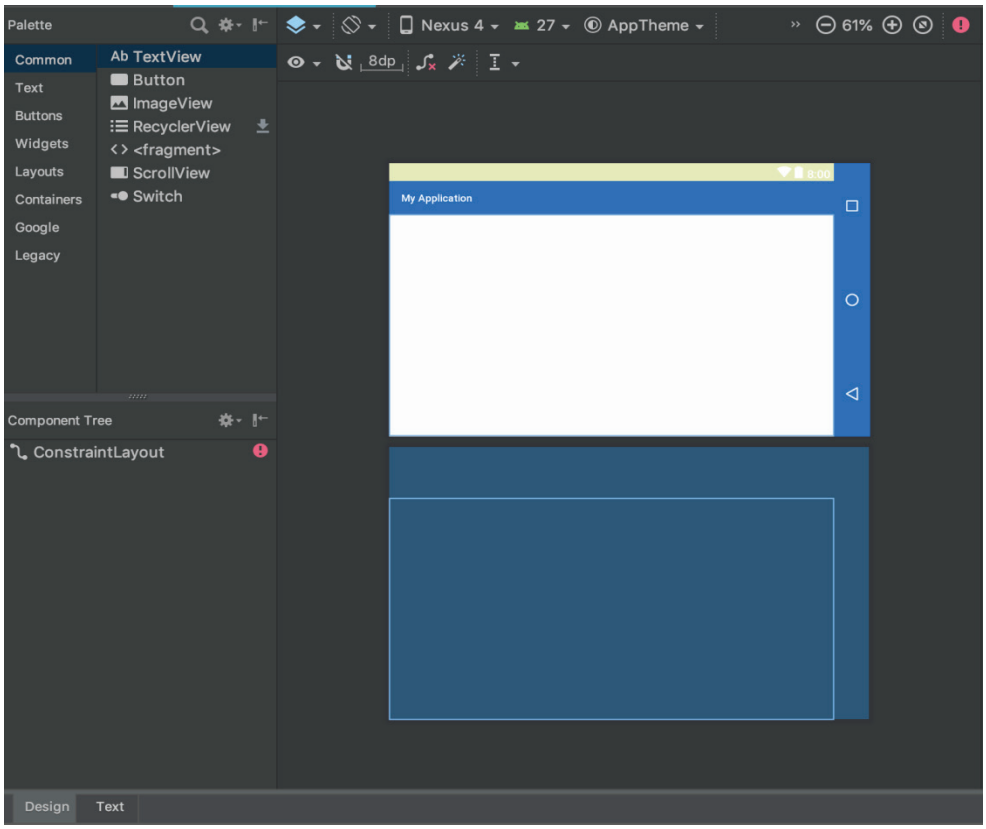
ملحوظة: يُقصد بِـ dpi (Dot Per Inch) أي نقطة (بيكسل) لكلّ بوصة (إنش).

نشاط (3) عملي: بالعودة إلى النشاط (2) الموقف التعليميّ (4-5) أجري التعديلات اللازمة على الواجهة لكي تتوافق مع جميع الأجهزة، ومختلف الوضعيّات.

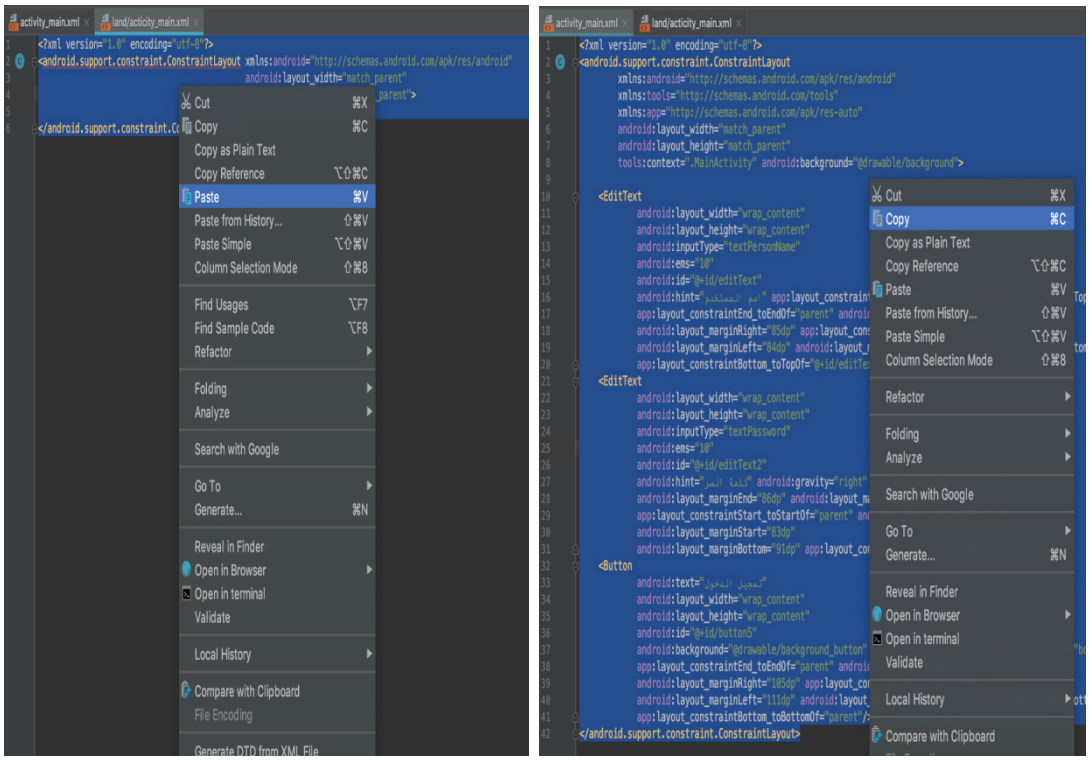
- **أوّلاً:** أبدأ بواجهة التصميم الافتراضيّة وأجعلها تتوافق مع وضعيّة قلب الشاشة، أنشئ واجهة تصميم جديدة بالاسم نفسه، وأختار Orientation، ثمّ >> ، ثمّ أختار Landspace كالآتي:



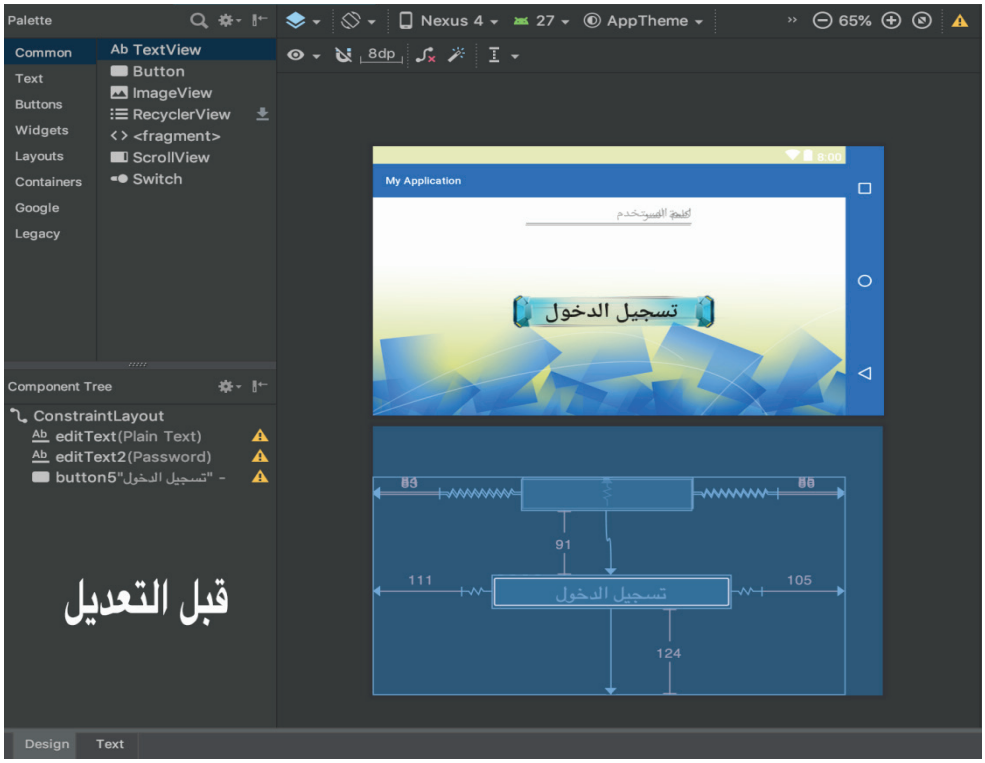
ثمّ يتمّ إنشاء ملف تصميم كالآتي:

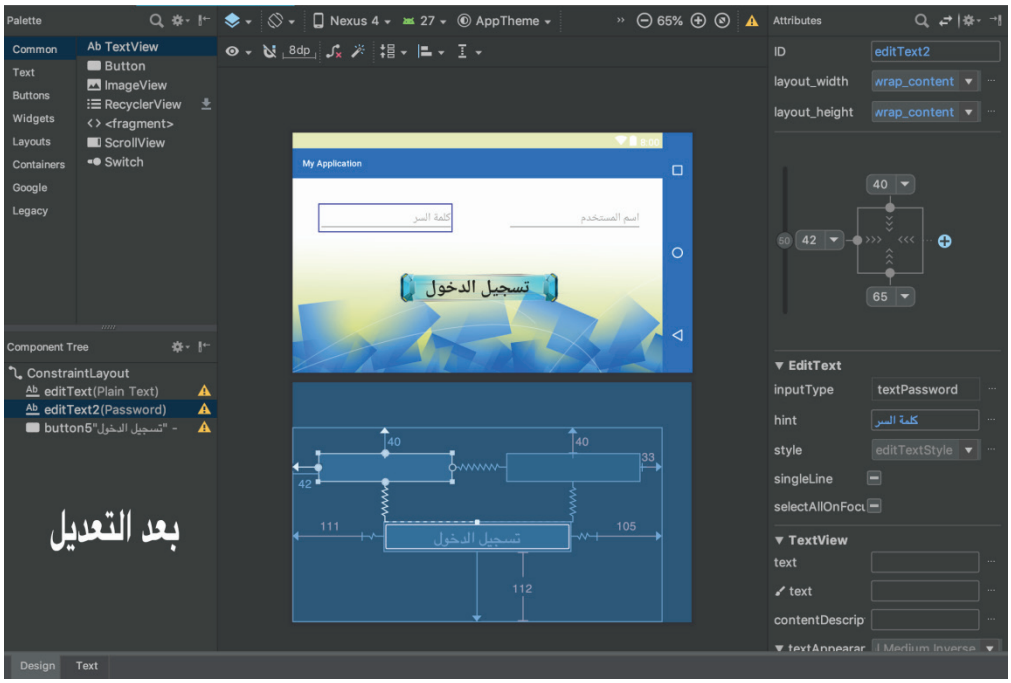


- **ثانياً:** أفتح كود xml الخاصّ بواجهة التصميم الرئيسيّة، وأنسخ الكود كما هو، ثمّ أنتقل إلى كود الواجهة الجديدة وأُلصقه هناك، كالآتي:

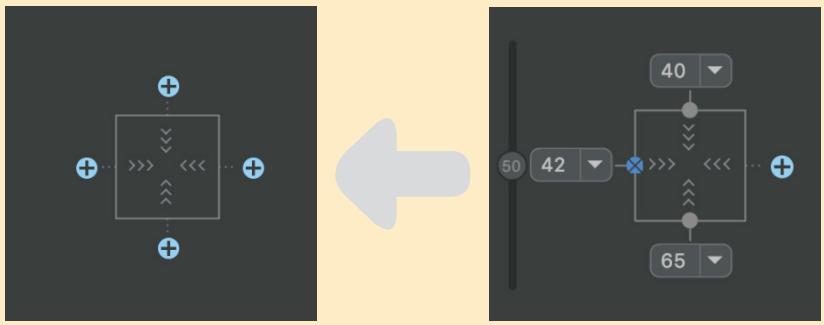


- **ثالثاً:** أفتح واجهة التصميم، وأُعدّل على العناصر لتكون متناسقة بشكل مناسب، كما يأتي:

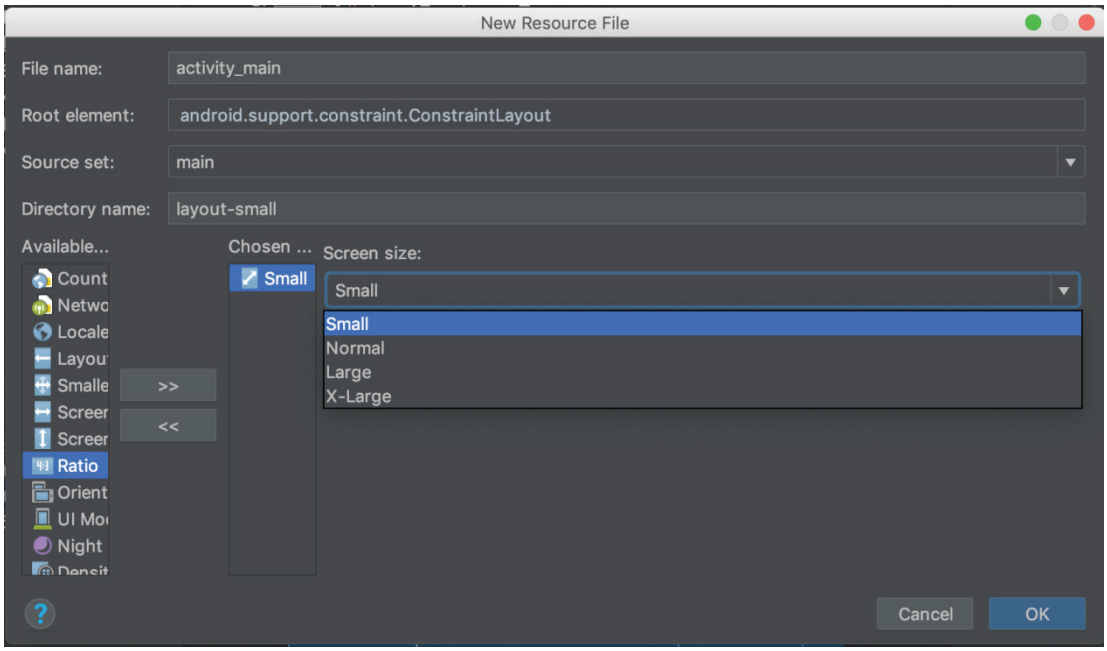




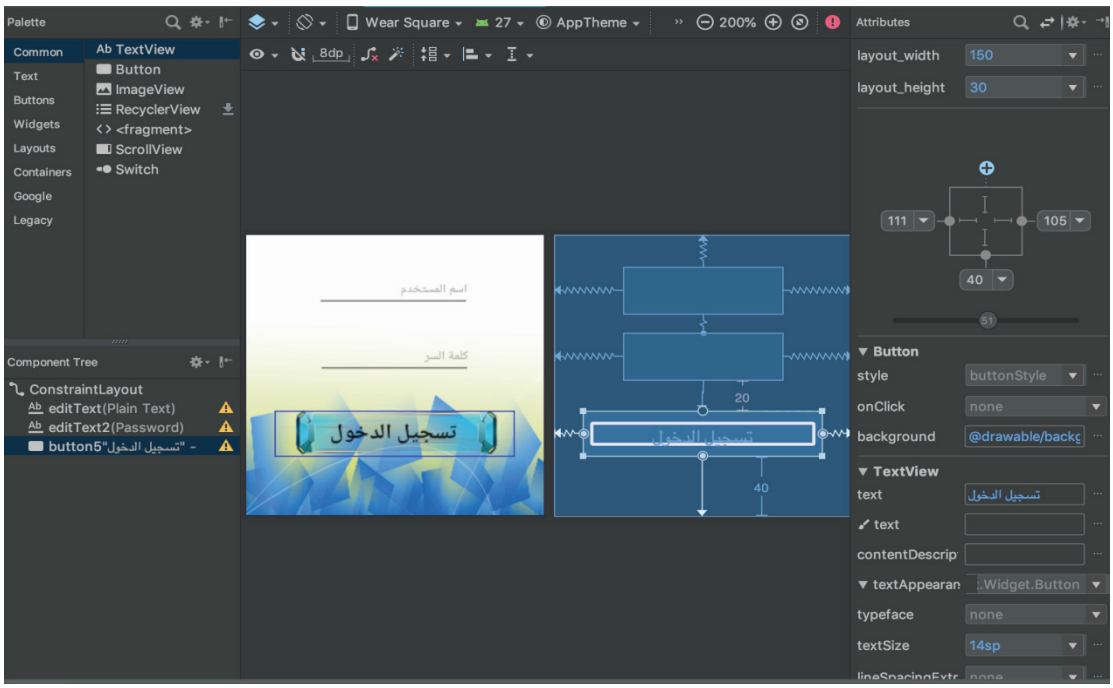
ملحوظة: قبل التعديل على مواقع العناصر أُلغي ارتباط العناصر من جميع الجهات كالآتي:



- **رابعاً:** أُنشئ واجهة جديدة بحجم شاشة مختلف وبالاسم نفسه، أختار size، ثمّ أضغط على [>>]، ثمّ أختار small، كالآتي:



- **خامساً:** أنسخ كود xml من الواجهة الأساسيّة، وأُلصقه في كود الواجهة المنشأة، ثمّ أُعدّل على العناصر من حيث الطول والعرض وحجم الخطّ، كالآتي:



بما أنّ الواجهة السابقة عبارة عن ساعة ذكيّة مربّعة الشكل أو دائريّة، فلا يلزم إضافة وضع قلب الشاشة.

- **سادساً:** أعمل تشغيل للمشروع، وأختبر النتائج.

أهمّ المعايير والإجراءات المتّبعة عند إنشاء أكثر من واجهة تصميم بأحجام مختلفة:

1. عند استخدام العناصرنفسها لا بدّ من استخدام ال id نفسه في كلّ الواجهات للعنصر الواحد.
2. إمكانيّة إضافة عناصر في واجهات، وعدم إضافتها في واجهات أخرى.
3. إمكانيّة تعديل ترتيب العناصر من واجهة لأخرى.

نشاط (4) عملي: أُكمل النشاط (3) لكي يتمّ التوافق مع الشاشات الأخرى، وبوضعيّة قلب الشاشة.

الأسئلة

1. ما المقصود بِـ dpi؟
2. ما أهمّ المعايير والإجراءات المتّبعة عند إنشاء أكثر من واجهة تصميم بأحجام مختلفة؟

أسئلة الوحدة

السؤال الأوّل: أختار رمز الإجابة الصحيحة فيما يأتي:

1. ما المجلّد الخاصّ بالأيقونة ؟

أ. layout. ب. manifests. ج. values. د. mipmap

2. كيف يتمّ استخدام كلمة في ملف strings من ملف xml ؟

أ. --.R.string. ب. --.R.strings. ج. --/@strings. د. --/@string.

3. ما هو status bar ؟

أ. الشريط الذي يظهر فيه اسم التطبيق. ب. الشريط السفليّ.

ج. الشريط الذي يظهر فيه نسبة البطّاريّة وغيرها. د. لا شيء ممّا ذُكر.

4. على ماذا يدلّ الاختصار dpi؟

أ. Device Pixel Inch. ب. Dot Per Inch.

ج. Dot Pixel Inch. د. لا شيء ممّا ذُكر.

السؤال الثاني: أوضّح: تُقسم أحجام الشاشات وفق دقّتها.

السؤال الثالث: أوضّح باختصار الملفّات التي تُعنى بتغيير نمط التطبيق.

السؤال الرابع: أذكر خطوات إضافة كلمات من ملف التصميم الواجهة إلى ملف strings.xml.

السؤال الخامس: ما هي ملفّات الأيقونة التي يحتويها مجلّد mipmap؟

مشروع الوحدة:

أُجري التعديلات اللازمة على مشروع الوَحدة الثانية من حيث:

1. اسم التطبيق.
2. دعم أكثر من لغة.
3. تعديل الأيقونة.
4. تعديل نمط التطبيق.
5. توافق التطبيق مع أكثر من جهاز، وبوضعيّات مختلفة.

تم بحمد اللّه

لجنة المناهج الوزارية:

أ. د. مروان عورتاني	د. بصري صالح	م. فواز مجاهد
أ. ثروت زيد	أ. عبد الحكيم أبو جاموس	م. وسام نخلة
د. سمية النخالة		

المشاركون في ورشات تطبيقات الهواتف الذكية للصف الثّاني عشر:

محمد الجوراني	جمال جودة	عاصم شراب
سماح بعلوشة	محمد قباجة	