جامعة مؤتة
كُليّة الدّراسات العُليا

حلول الجغرافيا المكانية بنظام الأندرويد

إعداد الطالب
يوسف سالم العصيات

إشراف
الأستاذ الدكتور إبراهيم العرود

رسالة مُقدّمة إلىٰ كليّة الدّراسات العُليا استكمالًا
لمتطلبات الحصول على درجة الماجستير في نظم
المعلومات الجغرافية (GIS)/ قسم الجغرافيا

جامعة مؤتة، 2021

الآراء الواردة في الرسالة الجامعية
لا تُعبّر بالضرورة عن وجهة نظر جامعة مؤتة

MUTAH UNIVERSITY

College of Graduate Studies

جامعة مؤتة

كلية الدراسات العليا

قرار إجازة رسالة جامعية

تقرر إجازة الرسالة المقدمة من الطالب يوسف سالم سليمان العصيات

والموسومة بـ: حلول الجغرافيا المكانية بنظام الاندرويد

استكمالاً لمتطلبات الحصول على درجة الماجستير في نظم المعلومات الجغرافية

القسم: نظم المعلومات الجغرافية في تاريخ ٢٠٢١/٠٥/٣١

من الساعة ١٢ إلى الساعة ٢ قرار رقم ٣/٢٠٢١

أعضاء اللجنة:	التوقيع	
أ.د ابراهيم مطيع ابراهيم العرود		مشرفا ومقررا
د. احمد حمد حمود العبادله		عضوا
د. سطام سالم مقبل الشقور		عضوا
د. زياد مفضي ايوب المخامرة		عضو خارجي

عميد كلية الدراسات العليا

أ.د عمر المعايطة

الإهداء

يا من أحمل اسمك بكل فخر، يا من علّمتني أن حدودي السماء، وأنني قمر أشعّ ضياء
...علّمتني الأبجدية لأخطّ هذه الكلمات الذهبية، فأنا منك وإليك، شكرًا لك أبي
إلى أجمل نساء الأرض وأرقى عقول النساء، إلى جنّة وضعت في الأرض تمشي باستحياء
...إلى تاج يوضع على الرأس بكل فخر، شكرًا لكي أمّي
... إلى أخوتي الذين وقفوا معي وساندوني، شكرًا لكم
...إلى زوجتي العزيزة التي سهرت معي وتحمّلت معي عناء رحلة العلم، شكرًا لكِ
...إلى ابني ريان الذي أشكر الله على أنه رزقني بك، شكرًا لوجودك في حياتي
...الشكر لجميع أصدقائي وأحبّتي، ولكل من أسهم في نجاحي

الشّكر والتقدير

أولًا الشكر والثناء، دائمًا، لله تعالى على إنجاز هذا العمل...
يسرني أن أوجه شكري لكل من نصحني أو أرشدني أو وجهني أو ساهم معي في إعداد هذا البحث بإيصالي للمراجع والمصادر المطلوبة في أي مرحلة من مراحله، وأخص بالشكر والتقدير مشرفي و أستاذي الفاضل الأستاذ الدكتور (ابراهيم العرود) على مساندتي وإرشادي بالنصح و التوجية.
كما أتقدم بأسمى آيات الشكر والتقدير إلى لجنة المناقشة الدكتور زياد المخامرة و الدكتور احمد العبادله و الدكتور سطام الشقور على قبول مناقشتي لرسالة الماجستير.
كما أتقدم بالشكر والتقدير إلى الهيئة التدريسية في قسم الجغرافيا وجامعة مؤتة المؤسسة التعليمية الرائدة لما قدمته لنا من تسهيلات وخدمات خلال فترة الدراسة التي يسرت لنا مسيرة إكمال دراستنا.

قائمة المحتويات

الموضوع	الصفحة
الشّكر والتقدير	أ
الإهداء	ب
قائمة المحتويات	ج
قائمة الجداول	هـ
قائمة الأشكال	و
المُلخّص باللغة العربية	ل
المُلخّص باللغة الإنجليزية	م
الفصل الأول: خلفيّة الدّراسة وأهميّتها	**1**
1.1 المُقدّمة	1
2.1 مُشكلة الدّراسة	2
3.1 أهميّة الدّراسة	3
4.1 أهداف الدّراسة	3
الفصل الثاني: المنهجيّة والتصميم	4
1.2 منهج الدّراسة	4
2.2 مصادر البيانات	4
3.2 البرامج المستخدمة في الدّراسة	4
4.2 إجراءات الدّراسة	7
الفصل الثالث: الإطار النظري والدراسات السابقة	15
1.3 لغة البرمجة	15
2.3 أنظمة التشغيل (Operating Systems)	16
3.3 نظام تشغيل أندرويد (Android)	16
4.3 الهاتف الذكي (Smart phones)	17
5.3 تطبيقات الهواتف الذكية (Mobile Apps)	17

الموضوع	الصفحة
6.3 برنامج أندرويد استوديو (Android Studio)	18
7.3 بعض الأدوات المهمة في برنامج أندرويد استوديو (Android Studio)	18
8.3 المكوّنات العامة لبرنامج أندرويد استوديو (Android Studio)	21
9.3 برمجة الأدوات الجغرافية داخل برنامج أندرويد استوديو (Android Studio)	25
10.3التطبيق المعدّ لهذه الدّراسة والبنية الهيكلية للحلول الجغرافية (Green Land)	29
11.3 المكوّنات والأدوات الإضافية داخل تطبيق (Green Land)	56
12.3 الخطوات العملية على التطبيق في العمل الميداني	68
13.3 التطبيقات العملية والدّراسات الميدانية الممكنة على التطبيق	81
14.3 مقارنة دقة الرصد الميداني بين تطبيقات (أندرويد) الجغرافية، وأجهزة (GPS) الدقيقة	97
15.3 برمجة أدوات (Arc GIS)داخل نظام (أندرويد)	99
16.3 الترميز الجغرافي (GeoCoding)	100
17.3 حلول الجغرافيا المكانية في التطبيقات الحكومية	100
18.3 حلول الجغرافيا المكانية في التطبيقات الخاصه	106
الفصل الرابع: النتائج والتوصيات	109
1.4 عرض النتائج	109
2.4 التوصيات	111
قائمة المصادر والمراجع	112

قائمة الجداول

الرقم	العنوان	الصفحة
1	بيانات الرصد الميداني للنقاط	98

قائمة الأشكال

الرقم	العنوان	الصفحة
1.	مخطط الدّراسة	6
2.	واجهة إدارة المشاريع كما تظهر في تطبيق(Green Land).	8
3.	واجهة الأدوات كما تظهر في تطبيق (Green Land).	8
4.	جزء من المعادلات الرياضية للعمليات الإحصائية على برنامج (Android Studio) المستخدمة في تطبيق (Green Land).	10
5.	جزء من الأكواد المستخدمة في عملية برمجة قواعد البيانات على برنامج (Android Studio) الموجودة داخل تطبيق (Green Land).	11
6.	العمل الميداني باستخدام تطبيق (Green Land).	12
7.	الميداني باستخدام تطبيق (Green Land)، وجهاز (RTK).	13
8.	الخرائط التي توفرها شركة (Google)	19
9.	النشاط (Main Activity)	23
10.	الواجهة التفاعلية (Layout)	24
11.	(Android Manifest).	25
12.	خرائط (Google) داخل برنامج (الأندرويد استوديو).	26
13.	الرمز التسلسي المميّز والموقع الإلكتروني الخاص بالمنصة.	27
14.	الكود البرمجي الخاص باستدعاء الإحداثيات الجغرافية من نظام الملاحة العالمي .	28
15.	واجهة تطبيق (Green Land)	29
16.	الكود البرمجي الخاص برفع البيانات	30
17.	الواجهة التفاعلية (Layout) الخاصة برفع البيانات.	31

الرقم	العنوان	الصفحة
18.	الكود البرمجي الخاص باستدعاء الإحداثيات الجغرافية من نظام الملاحة العالمي .	32
19.	الكود البرمجي الخاص بإضافة الإحداثيات الجغرافية من نظام الملاحة العالمي على الخريطة.	33
20.	واجهة تفاعليه (Layout) خاصة بالمعلومات الوصفية .	34
21.	الكود البرمجي الخاص بالمعلومات الوصفية (استدعاء الإحداثيات).	35
22.	الكود البرمجي الخاص بتحويل الصورة إلى نصوص والعكس .	36
23.	الأيقونة الزرقاء الخاصة بتحديد المسار على خرائط (Google) .	37
24.	تطبيق خرائط(Google) كما تظهر عليه المسارات الخاصة بالوصول إلى النقطة .	38
25.	واجهة تفاعليه (Layout) خاصة بالعمليات الإحصائية .	39
26.	الكود البرمجي الخاص بالأزرار الإحصائية في العمليات الإحصائية.	40
27.	الكود البرمجي الخاص بملف (Java Class) الذي يحتوي على قوالب جاهزة من العمليات الإحصائية.	41
28.	الظاهرة المركزية	42
29.	الكود البرمجي الخاص بالظاهرة المركزية داخل ملف (Java Class)	43
30.	الكود البرمجي الخاص بالظاهرة المركزية داخل ملف (Java Class) .	44
31.	الكود البرمجي الخاص بالظاهرة المركزية داخل ملف النشاط (Activity).	44

الرقم	العنوان	الصفحة
.32	الكود البرمجي الخاص بالظاهرة المركزية داخل الزر.	45
.33	المركز المتوسط.	45
.34	الكود البرمجي الخاص بالمركز المتوسط داخل ملف (Java Class).	46
.35	الكود البرمجي الخاص بالمركز المتوسط داخل ملف النشاط (Activity).	47
.36	المسافة المعيارية	48
.37	الكود البرمجي الخاص بالمسافة المعيارية داخل ملف (Java Class).	49
.38	الكود البرمجي الخاص بالمسافة المعيارية داخل ملف النشاط (Activity).	50
.39	الكود البرمجي الخاص بالمسافة المعيارية داخل الزر.	50
.40	الانحراف المعياري.	51
.41	الكود البرمجي الخاص بالمسافة المعيارية داخل الزر.	51
.42	الكود البرمجي الخاص بالتباين داخل الزر.	52
.43	الخريطة الحرارية	52
.44	الكود البرمجي الخاص بالخريطة الحرارية داخل ملف النشاط (Activity)	53
.45	الواجهة التفاعلية (Layout) لقاعدة البيانات داخل تطبيق (Green Land).	54
.46	الكود البرمجي الخاص بقواعد البيانات داخل ملف (Java Class) .	55
.47	الكود البرمجي الخاص بأداة الطبوغرافيا داخل ملف النشاط (Activity) .	58
.48	الكود البرمجي الخاص بأداة الطبوغرافيا (إنشاء فترة كنتورية	59

الرقم	العنوان	الصفحة
	بين النقاط المرصودة) داخل ملف النشاط (Activity) .	
49.	الكود البرمجي الخاص بأداة طبقات (KMZ) داخل الزر في ملف النشاط (Activity) .	61
50.	الكود البرمجي الخاص بأداة تحويل الإحداثيات داخل ملف (Java Class).	63
51.	الكود البرمجي الخاص بأداة تصدير النقاط في ملف النشاط (Activity).	65
52.	الكود البرمجي الخاص بأداة تصدير النقاط (تعديل نوع الإحداثيات) في ملف النشاط (Activity).	65
53.	الكود البرمجي الخاص بأداة تصدير النقاط (داخل زر التصدير) في ملف النشاط (Activity).	66
54.	الكود البرمجي الخاص بأداة تصدير النقاط (داخل زر المشاركة) في ملف النشاط (Activity).	66
55.	ملف (Note Pad) الذي تم تصديرة.	67
56.	الكود البرمجي الخاص بأداة استيراد النقاط (داخل الزر) في ملف (Activity).	68
57.	شكل المسار لمنطقة الدّراسة على برنامج (Google Earth).	69
58.	الواجهة التفاعلية لإدارة المشاريع في تطبيق (Green Land).	70
59.	الواجهة التفاعلية (Layout) لرصد النقاط في تطبيق (Green Land)، والتي يظهر فيها مسار منطقة الدّراسة.	71
60.	الواجهة التفاعلية (Layout) لرصد النقاط في تطبيق (Green Land)، والتي يظهر فيها النقطة المرصودة.	72
61.	الواجهة التفاعلية (Layout) لحفظ النقاط في تطبيق	73

الرقم	العنوان	الصفحة
	(Green Land)، والتي تظهر فيها المعلومات الوصفية للنقطة.	
.62	الواجهة التفاعلية (Layout) لقاعدة البيانات في تطبيق (Green Land)، والتي تظهر فيها البيانات.	74
.63	الواجهة التفاعلية (Layout) للعمليات الإحصائية في تطبيق (Green Land)، والتي تظهر فيها الأزرار الخاصة بالعمليات الإحصائية وخرائط (Google).	76
.64	الواجهة التفاعلية (Layout) لتصدير النقاط في تطبيق (Green Land)، والتي تظهر فيها النقاط المراد تصديرها.	77
.65	الواجهة التفاعلية (Layout) لاستيراد النقاط في تطبيق (Green Land)، والتي تظهر فيها النقاط المراد استيرادها.	78
.66	الواجهة التفاعلية (Layout) لرصد النقاط الطبوغرافية في تطبيق (Green Land) .	80
.67	الواجهة التفاعلية (Layout) لرصد النقاط الطبوغرافية (خيارات الرصد الطبوغرافي) في تطبيق (Green Land).	80
.68	منطقة الدّراسة حسب المخطط التنظيمي.	82
.69	مخطط تنظيمي لمنطقة الروضة للأراضي داخل حدود التنظيم	82
.70	ملف (KMZ)كما يظهر على الخريطة لموقع الدّراسة	84
.71	نقاط الرصد كما تظهر على الخريطة للأبنية السكنية.	85
.72	المركز المتوسط كما تظهر على الخريطة	86
.73	الظاهرة المركزية كما تظهر على الخريطة.	87
.74	الخريطة الحرارية كما تظهر على الخريطة.	88
.75	موقع بناء المدرسة الفعلي بالنسبة للمركز المتوسط كما تظهر على الخريطة.	90

الرقم	العنوان	الصفحة
.76	لائحة تصدير النقاط المرصودة	91
.77	نقاط الرصد كما تظهر داخل برنامج (Arc Map).	92
.78	نتائج العملية الإحصائية (Mean Center) داخل برنامج (Arc Map).	93
.79	نتائج العملية الإحصائية (Central Feature) داخل برنامج (Arc Map).	94
.80	نقاط منطقة الدّراسة حسب الصورة الجوية	95
.81	ملف (KMZ) للنقاط المراد رصدها لمنطقة الدّراسة	95
.82	نقاط الرصد الطبوغرافية كما تظهر على برنامج (Auto Cad).	96
.83	جهاز (RTK) من نوع (Stonex)	97
.84	الموقع الإلكتروني للحكومة الإلكترونية	101
.85	واجهة تطبيق دائرة الأراضي والمساحة	103
.86	واجهة تطبيق الخارطة الصحية	104
.87	نطاق الخدمة كما يظهر في تطبيق الخارطة الصحية	105
.88	فلترة النتائج كما تظهر في تطبيق الخارطة الصحية.	105
.89	واجهة تطبيق أوبر.	106
.90	واجهة تطبيق (Calibrated).	107
.91	واجهة تطبيق (Jordanian Converter).	108

المُلخّص

حلول الجغرافيا المكانية بنظام الأندرويد

يوسف سالم العصيات

جامعة مؤتة،2021

هدفت هذه الدّراسة إلى ربط برمجة نظام (أندرويد) مع حلول الجغرافيا المكانية المرتبطة بالعمل الميداني لدى المستخدمين الجغرافيين، بالإضافة إلى إظهار الإمكانات التي توفرها هذه البرمجة من حلول جغرافية مكانية لمعالجة البيانات.

وتمثّلت مشكلة الدّراسة في وجود فجوة بين البرمجة، أو تعلّم البرمجة، والجغرافيين والتي تحتاج إلى طريقة ربط معينة مع مجال الجغرافيا، بالإضافة الى الحاجة الماسّة لتصميم وتطوير أدوات يمكن من خلالها الوصول إلى نتائج، ربما لا يمكن الوصول إليها من خلال برامج متخصصة في التحليل المكاني كما تمثلت مشكلة الدراسة في الحاجة لتوفير تقنيات تساعد أو تسهّل القيام بالأعمال الميدانية والتي تساهم في رفع مستوى العمل الميداني.

كما لخّصت منهجية الدراسه في :

1- تقديم فهم عام عن موضوع البرمجة بلغة الأندرويد والتحليل المكاني.

2- الأمور المتعلقة، أو التي تختصّ بالجغرافيا داخل بيئة الأندرويد.

3- تصميم تطبيق يجسّد فهم أدوات التحليل المكاني، وتحليل النتائج التي يعطيها.

كما لخّصت نتائج الدراسه في أن لغة البرمجة (أندرويد) يمكن استخدامها في علوم الجغرافيا بشكل واسع، حيث إن الأدوات التي توفرها هذه اللغة تساعد بشكل كبير الجغرافيين في فهم الظواهر، سواء في الشق التحليلي للنتائج، أو الشق الميداني ، كما تعدّ لغة برمجة (أندرويد) من اللغات المفتوحة المصدر والمرنة، والتي يمكن تطوير الأدوات بواسطتها بشكل مستمر، كما أظهرت الدّراسة مطابقة النتائج التي تم العمل عليها في التطبيق، مع النتائج التي تم الحصول عليها من البرامج المتخصصة لنفس نقاط الرصد التي تم رصدها من أجل العمليات الإحصائية ، كما أظهرت نتائج الدّراسة أن دقة النتائج تعتمد على دقة الأدوات المستخدمة في العمل الميداني.

بعض التوصيات التي قام الباحث بطرحها:

1- يوصي الباحث بضرورة إضافة مواد لتعلّم البرمجة في أقسام الجغرافيا، وذلك لضرورة مواكبة الجغرافيين لهذا المجال.

2- ربط البرامج والأدوات المتخصصة في التحليل المكاني مع الأقسام الجغرافية.

3- فتح قنوات عمل مشترك مع الشركات المتخصصة في عمل تطبيقات حلول مكانية لدمج الأفكار المشتركة، وإثراء العلوم الجغرافية بهذا المجال.

استخدمت في هذه الدّراسة برمجية (أندرويد استوديو) التي صمم من خلالها تطبيق (Green Land)، والذي تم فيه تجسيد الحلول المكانية، حيث استخدم التطبيق في الأعمال الميدانية لبيان مدى فعالية الحلول المكانية في الواقع الحقيقي.

Abstract

Geospatial solutions using the Android system

Yousef Salem Alasayyat

Mutah University، 2021

This study aimed to link the programming of the Android system with geospatial solutions related to field work among geographical users, in addition to this, it shows the capabilities that are provided by this programming of geospatial solutions for data processing.

The challange of the study were in the a gap between programming, or learning programming, and geographers, which need a certain method of alignment with the field of geography, in addition to the necessity to design and develop tools through to reache to the results, possibly not accessible through programs specialized in spatial analysis. The addtional challange of the study was represented in the necessity to provide techniques that facilitate the work in the field and boost the producitivity.

The study methodology summarized below:

1. Provide explanation about programming in the Android language and spatial analysis
2. related topics to geography within the Android environment
3. Design an application that simulate tools of spatial analysis, and analyzes the results it gives.

The results of the study also summarized the programming language (Android) that can be widely used in geography, as the tools provided by this language greatly to help geographers in understanding phenomena, whether in the analytical part of the results, or the field part, as it is considered a programming language (Android) of the open source and flexible languages, through the tools that can be developed continuously, and the study also showed matching the results were deployed in the application, with the results obtained from specialized programs for the same monitoring points that were monitored for statistical operations, and the results showed The study stated that the accuracy of the results depends on the accuracy of the tools used in the field work.

Some recommendations made by the researcher

1. The researcher recommends the necessity of adding topics for learning programming in the geography departments, due to the necessity of keeping pace with the geographers in this field.
2. Linking programs and specialized tools in spatial analysis with geographical departments.
3. Launching joint working channels with companies that specialized in creating spatial solutions applications to integrate common ideas and enrich geographical sciences in this field.

In this study, the (Android Studio) software was used, through designed the (Green Land) application that simulated in spatial solutions were embodied. The application were used in fieldwork to demonstrate the effectiveness of spatial solutions in real life.

الفصل الأول

خلفيّة الدّراسة وأهميّتها

1.1 المُقدّمة:

يعدّ نظام التشغيل (أندرويد) من التقنيات الحديثة التي تساعد في ابتكار تطبيقات على الجوّالات التي تسهّل علينا أمورًا كثيرة في حياتنا العملية واليومية؛ إذ يعدّ (الأندرويد) نظام تشغيل مجاني مفتوح المصدر صُمم أساسًا للأجهزة ذات شاشات اللّمس، كالهواتف الذكية، والحواسب اللوحية .

ساهمت التطوّرات التي حصلت لنظام (أندرويد) على ربط الوظائف الجغرافية ، من خلال الخرائط، والصور الجوية، ونظام الملاحة العالمي (GPS) التي توفرها الأجهزة المحمولة وتدعمها، والتي تمكّننا من الدخول إلى هذا النظام من أوسع أبوابه، مع ضرورة ممارسته وفهم كيفية عمله من ناحية جغرافية.

يعتمد نظام الأندرويد في أساسه على فكره كتابة النصوص البرمجية (Code) داخل برامج مخصصة، مثل أندرويد استوديو (Android Studio)، من أجل برمجة وتصميم الأدوات والخدمات، وكيفية التفاعل والربط بينها، وتنفيذ المهام للوصول إلى الخدمة المطلوبة و يمكن حصر أعمال الجغرافيين داخل هذه النظام (الأندرويد) في فهم كيفية التعامل مع الخرائط وصور الأقمار الصناعية، وتحديد الإحداثيات الجغرافية، وأعمال رفع وتوقيع نقاط الإحداثيات، فمن خلالها نستطيع الوصول إلى بقية التقنيات، مثل: فكرة استدعاء إحداثيات الموقع، أو الوصول إلى إحداثيات في موقع ما، أو الوصول إلى أقرب نقطة، أو اتّباع أقصر مسار، أو تحويل الإحداثيات، أو إنشاء الطبقات لعمل ترقيم للظواهر، أو تحديد نطاق انتشار ظاهره معينة، أو تخزين بيانات وصفية لظواهر معينة، أو ربطها بقواعد بيانات على خوادم شبكية.

تساعد هذه العمليات في إنشاء تطبيقات عملية مثل تطبيق (أوبر) الذي يعتمد على طلب إحداثيات الموقع، ولتحديد أقرب موقع لتكسي، وتمكينه من الوصول إلى نقطة ما، من خلال فكرة أقرب مسار، أو مثل تطبيق الدفاع المدني الذي من خلاله ترسل موقعك في موقع الحادث ليتم تحديد موقع الحادث بشكل دقيق، وتسهيل عملية الوصول إليه في أقل وقت ممكن، بالإضافة إلى معرفة أقرب مركز إطفائية.

كل هذه الأمور يتم توضيبها للتسهيل على المستخدم للوصول إلى المعلومة والتعامل معها، وتوفير الوقت والجهد، وتحرّي الدقة فيها، أما عمليات المعالجة والتخزين والتحديث، فهي مهمة التطبيق نفسه.

ستقوم هذه الدّراسة على برمجة هذه التطبيقات للوصل إلى تطبيق يخدم الجغرافيين من ناحية فنية، مثل تسجيل النقاط من الواقع، وتثبيت النقاط والوصول إليها وتخزينها وتحويلها إلى النظام المحلي (JTM) و(Cassini)، وقياس المسافات، والتعامل مع الظواهر على الصور الفضائية وترقيمها، والتعامل مع ملفّات (KMZ) وكيفية إضافتها، وكيفية تحديد الاتجاهات بالبوصلة، وكيفية التعامل مع قواعد البيانات وملفّات (CSV)، والكثير من الأمور المتعلقة التي ربما يحتاجها الجغرافيون، في الميدان.

2.1 مُشكلة الدّراسة:

1- إن المجال الجغرافي من المجالات التي تتعامل مع برامج التحليل المكاني بشكل واسع، والتي تختصّ بمعالجة البيانات التي يتم الحصول عليها من الواقع الحقيقي، مثل برامج: (Erdas) و(ArcGIS) للوصول إلى نتائج يمكن من خلالها اتخاذ القرارات المناسبة بطريقة سهله و سريعه، ولكن لا يكفي هنا التعامل فقط مع هذه البرامج في العمل الجغرافي لوجود حاجة ماسّة لتصميم وتطوير أدوات غير موجوده في التطبيقات التجاريه و يمكن من خلالها الوصول إلى نتائج ربما لا يمكن الوصول إليها من خلال برامج متخصصة في التحليل المكاني، لذلك وجب هنا الاطّلاع وتعلّم لغة البرمجة التي تمكّن المطوّرين من توفير أدوات جديدة تساعد الجغرافيين في تحليل الظواهر الأكثر تعقيدًا.

2- أكثر المشكلات التي تواجه الجغرافيين، الذين يحاولون الحصول على البيانات من الواقع الحقيقي أو الميدان، هو توفير تقنيات تساعد أو تسهّل القيام بالأعمال الميدانية، مثل: الرفع، والتوقيع، والقياس، والترقيم، وتحديد الموقع والاتجاه، والاستعلام، وتخزين البيانات وتنظيمها داخل جداول

وصفية، والحصول على إحداثيات محلية، مثل (JTM)، وغيرها من الأمور.

3- الصعوبات التي تواجه جامع البيانات في الميدان بحاجه الى تطوير الآلية للتعامل معها لإعطاء ابعاد ديناميكية جديدة.

3.1 أهميّة الدّراسة:

تكمن أهميّة البحث الرئيسة في:

1-القدرة على تسهيل التعامل مع البيانات في الميدان، من خلال تصميم تطبيق (أندرويد) يساعد في عملية الحصول على البيانات من الواقع بشكل أقرب للدقة لحفظها، ومعالجتها، وتفسيرها، واستدعائها، وتقليل الوقت والجهد المبذولين في هذه العملية.

2-توضيح طريقة برمجة (أندرويد) في الأمور المختصة بالجغرافيا لفتح الباب على تطوير هذا المجال بما يخدم الجغرافيين.

3-التمكين من التعامل مع البيانات والمعلومات لتسهيل نقلها، ومعالجتها، وتخزينها، وعرضها، بشكل يخدم الجغرافيين دون تكلفة مادية.

4-نظرًا لندرة الكتب العربية المتحدثة بموضوع الدّراسة، تعدّ هذه الدّراسة إسهامًا جادًّا وحقيقيًّا في تحقيق إضافة محلية باللغة العربية، وتسهم في تطوير مهارات وخبرات المصممين والمهتمين بتطوير مهاراتهم في هذا المجال.

4.1 أهداف الدّراسة:

يهدف البحث، بشكل رئيس، تحري إمكانية نظام (أندرويد) بما يخصّ التطبيقات الجغرافية، وإنشاء تطبيق يخدم الجغرافيين في الميدان بشكل كبير، ويوفر لهم تقنيات متعددة تغنيهم عن الأدوات التقليدية لتقليل الجهد والوقت في التحليل مع البيانات ، بالإضافة إلى تقليل الفجوه بين الجغرافيين ولغة البرمجة. فمن خلال إنشاء تطبيق مدعم بهذه الأمور، سوف يغني ذلك عن استخدام أدوات تقليديه، مثل: الورق، والقلم لتسجيل الملاحظات، أو كاميرا تصوير، أو أدوات مرتفعة الثمن كجهاز (GPS) لتحديد الموقع، أو إعادة ترتيب البيانات في المكتب بعد الحصول عليها من الواقع، أو حتى استخدام حاسب لمعالجة البيانات.

الفصل الثاني

المنهجيّة والتصميم

1.2 منهجية الدّراسة:

تم الاعتماد في هذه الدّراسة على:

1-برنامج (Android Studio)، حيث تم تصميم تطبيق (Green Land) بوساطة هذا البرنامج الذي يوفر تقنيات وأدوات تسهّل عملية البرمجة، بالإضافة إلى معاينة التطبيق عليه، وإجراء التجارب على التطبيق للتأكد من عمل التطبيق بشكل صحيح.

2-تم الاعتماد على إجراء أعمال ميدانية ضمن بعض المناطق، من خلال رصد نقاط في الواقع الحقيقي للتأكد من قدرة التطبيق على العمل ميدانيًّا.

3-تم الاعتماد على المنهج التحليلي لمقارنة النتائج التي يستطيع التطبيق إيجادها، مع النتائج التي قدّمتها البرامج والأدوات الأكثر تخصصًا.

2.3 مصادر البيانات:

اعتُمد في الحصول على بيانات الدّراسة على المصادر التالية:

البيانات المكانية:

1-الخرائط التنظيمية لعام (2020)، من مديرية الشؤون البلدية للواء القصر.

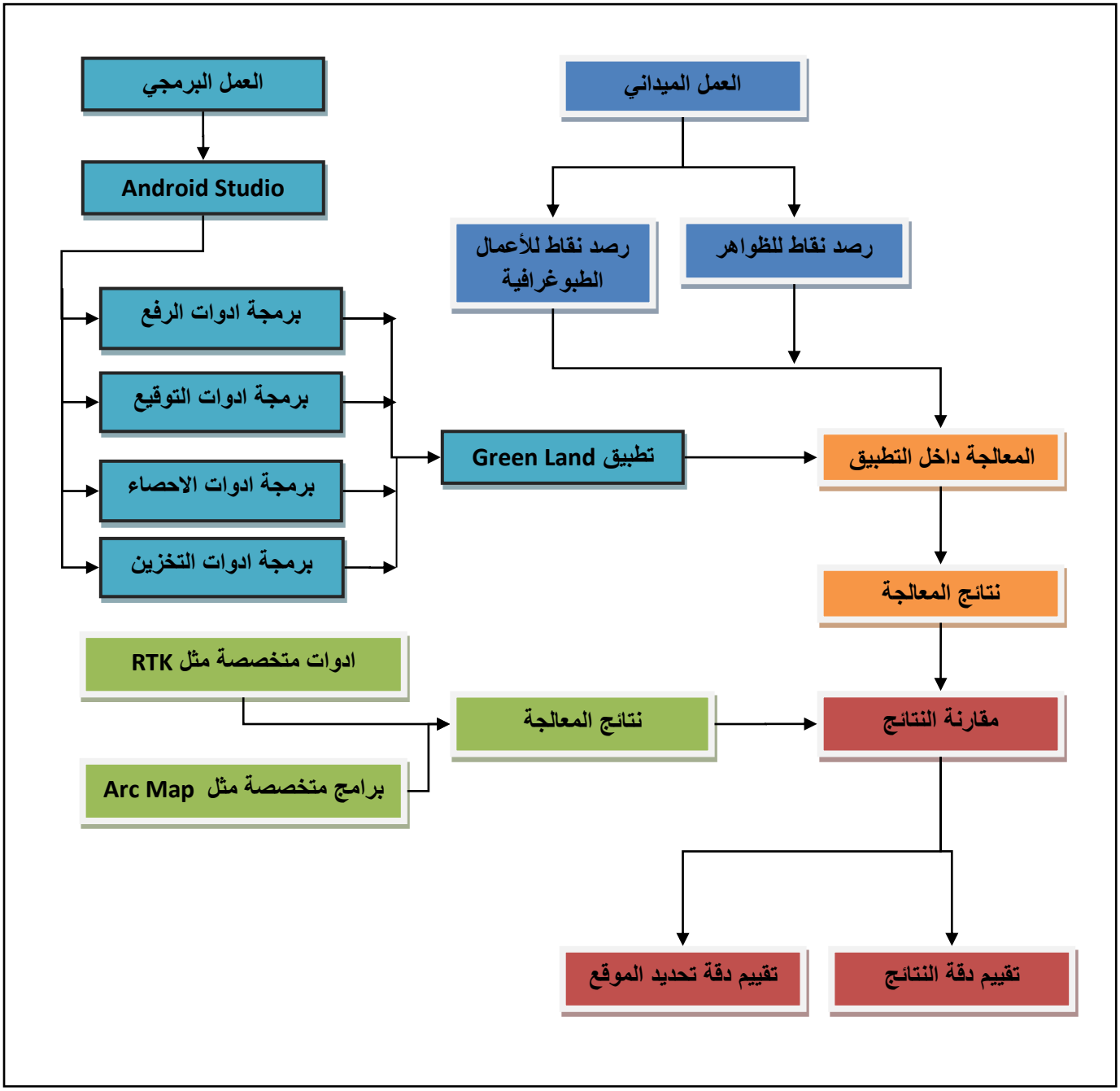
3.3 البرامج المستخدمة في الدّراسة :

1-برنامج (Arc Map)، والذي تم استخدامه من أجل مقارنة النتائج التي تم الحصول عليها من التطبيق الذي تم تصميمه (Green Land) لنقاط الرصد الميداني، مع نتائج برنامج (Arc Map) .

2-برنامج (Google earth)، وذلك لتحديد مناطق الدّراسة التي طرحها كأمثلة داخل البحث، حيث يتم استخراج ملف (KMZ) من قبل برنامج (Google Earth) لتتم إضافته إلى التطبيق.

3-برنامج (Android Studio)، حيث تم استخدامه من أجل تصميم برنامج (Green Land).

4-محرّر النصوص، أو ما يعرف بالمفكرة (Notepad)، تم استخدامه لعرض نتائج الملفّات التي يتم استخراجها من التطبيق، والتي تحتوي على إحداثيات النقاط المرصودة، بالإضافة إلى المعلومات الوصفية.



الشكل (1): مخطط الدّراسة

المصدر: عمل الباحث.

4.2 إجراءات الدّراسة:

أولًا:

1-تحديد العنوان وموضوع الدّراسة.

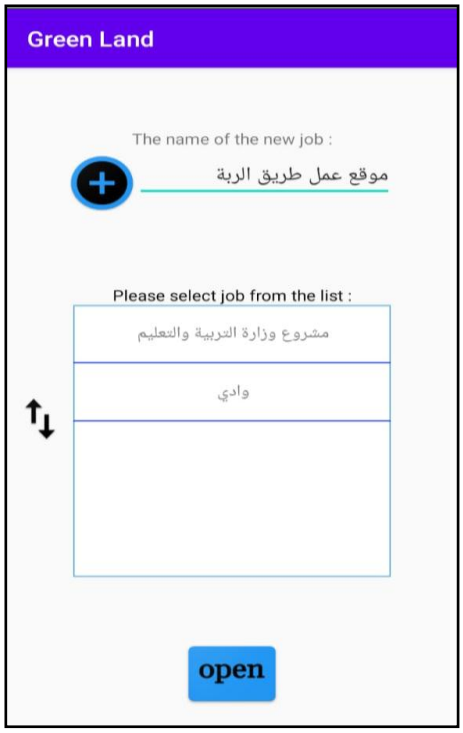
2- تقديم خلفيّة الدّراسة وأهميّتها، من خلال عرض مشكلة الدّراسة، وأهميّتها، وأهدافها، وحدودها، ومحدّداتها.

3-التعريف بالبرمجة، وأنظمة التشغيل، والهاتف الذكي، والتطبيقات، وبرنامج (أندرويد استوديو).

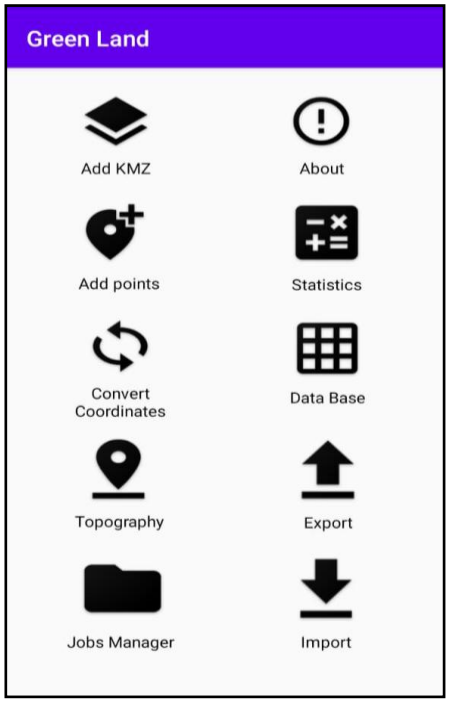
ثانيًا:

1-برمجة تطبيق (Green Land) باستخدام برنامج (Android Studio) للقيام بإنتاج أدوات توفر حلول مكانية للجغرافيين، حيث تم تقسيم العمل البرمجي إلى عدة أقسام، وهي:

أ- تصميم الواجهات البسيطة التي يمكن فهمها دون تعقيدات، حيث تم تصميم واجهة إدارة المشاريع في بداية عمل التطبيق، الشكل(2)؛ لتحديد المشروع المراد العمل عليه، بالإضافة إلى ذلك، إمكانية إنشاء مشروع جديد باسم يحدّده المستخدم، حيث يوصي الباحث بعدم تكرار تسمية المشروع لتجنّب حدوث أخطاء في عمل التطبيق، ويمكن، أيضا، حذف المشروع الذي تم إضافته، وذلك بالنقر المطوّل على اسم المشروع داخل الجدول، ثم النقر على خيار حذف، كما تم تصميم واجهة الأدوات داخل التطبيق برسومات بسيطة، كما في الشكل(3)، بالإضافة إلى تسميات تحدّد نوع العمل الذي تقوم به هذه الأدوات.



الشكل (2): واجهة إدارة المشاريع كما تظهر في تطبيق(Green Land).



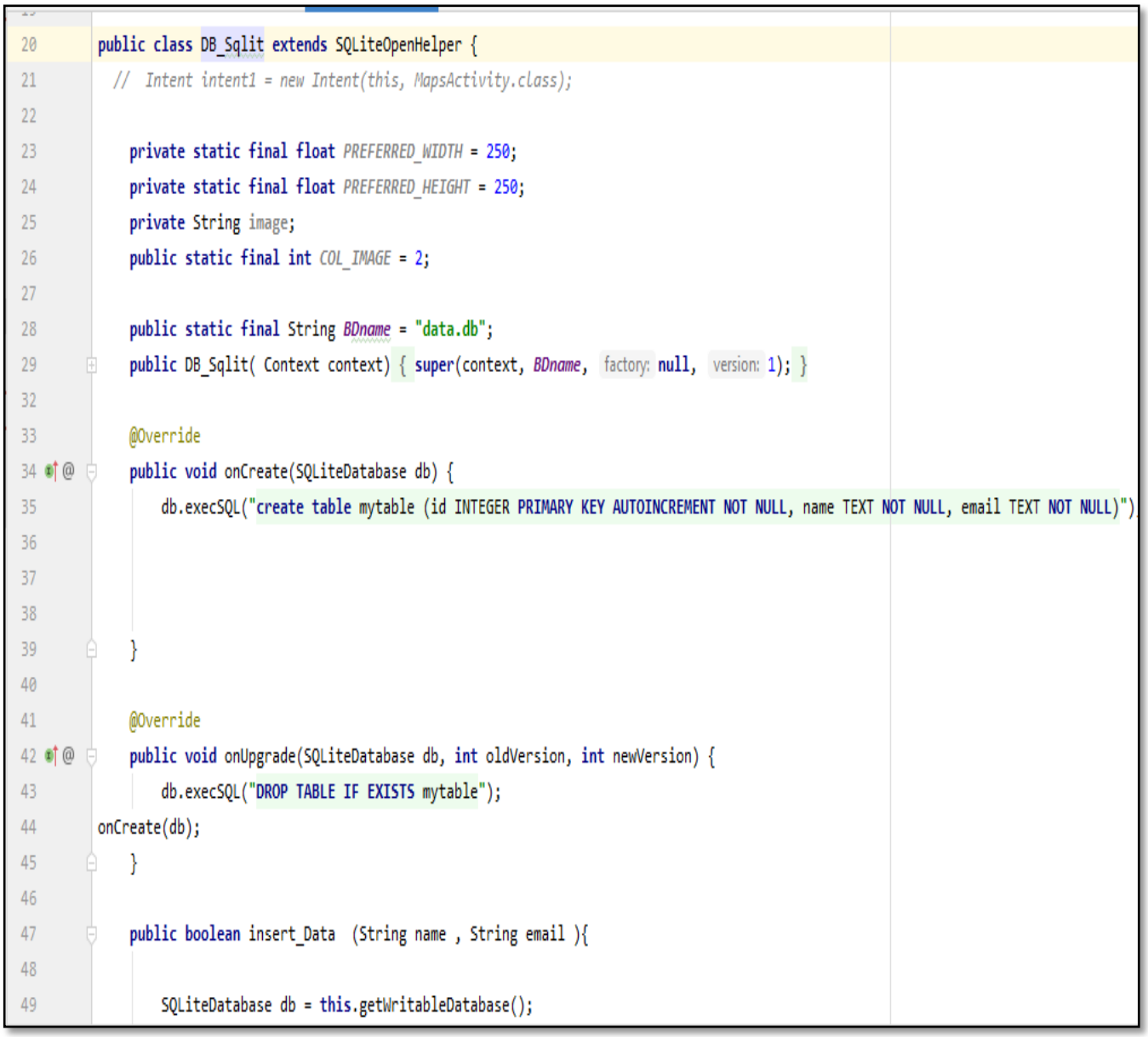
الشكل (3): واجهة الأدوات كما تظهر في تطبيق (Green Land).

ب- برمجة الأدوات المختصة بعمليات الرفع والتوقيع، وذلك من خلال ربط الميزات التكنولوجية الذي يوفره الجهاز الذكي، مثل: الخرائط، ونظام (GPS)، والأزرار، وقواعد البيانات مع بعضها بعضًا للقيام بالإعمال المطلوبة، حيث تقسّم عملية برمجة الأدوات إلى ثلاثة أقسام، وهي: برمجة أدوات الرفع، وبرمجة أدوات التوقيع، وبرمجة أدوات الإحصاء، حيث تتم عملية برمجة أدوات الرفع من خلال ربط أوامر الأزرار مع خاصية (GPS)، وتتم تخزين الإحداثيات بشكل مؤقت إلى حين النقر على زر الحفظ النهائي ليتم حفظها في قاعدة البيانات، بالإضافة إلى تخزين الملاحظات والصور الملتقطة، أيضا، في قاعدة البيانات، وتكون جميع هذه البيانات مرتبطة بقيمة فريدة داخل قاعدة البيانات لتسهيل عمليه البحث والاستدعاء، كما تتم برمجة أدوات التوقيع من خلال مقارنة الإحداثيات المدخلة مع الإحداثيات الملتقطة من (GPS) في الوقت الحقيقي لتحديد اتجاه المسار والمسافة المراد قطعها. أما بالنسبة للأدوات المختصة بعمليات الإحصاء، فهي أدوات تقوم بوظائف معينة على النقاط المرصودة للوصول إلى نتائج يمكن فهمها وتحليلها، حيث تعتمد وظائف العمليات الإحصائية على المعادلات الرياضية، كما في الشكل (4)، والتي يتم تطبيقها على النقاط المرصودة المحفوظة في قواعد البيانات، كما تتم برمجة قواعد البيانات بشكل يمكن فيه فهم القيم المراد حفظها فيه، وتحديد أسلوب الاستدعاء والبحث والحذف داخل قواعد البيانات، كما تتم برمجة قواعد البيانات، كما في الشكل(5)، مع بقية الأدوات للتمكّن من الوصول إلى النقاط المخزنة داخل القواعد وإجراء العمليات عليها.

activity_main.xml | Statistics.java

```java
package com.example.g_r;

import android.text.method.ScrollingMovementMethod;

public class Statistics {


    public String computeCentroid(String[] E, String[] N) {
        String[] points_E = E;
        String[] points_N = N;

        Double xxx = 0.0;
        Double yyy = 0.0;
        int n = points_E.length;

        for (int x1 = 0; x1 < points_E.length; x1++) {

            xxx += Double.parseDouble(points_E[x1]);
            yyy += Double.parseDouble(points_N[x1]);
        }

        //String.format("%.3f",wtocas.No);
        String cc1 = String.format("%.3f", xxx / n) + " , " + String.format("%.3f", yyy / n);
        // xxx/n +" , "+ yyy/n;
        // String cc1;
        return cc1;
    }

```

الشكل (4): جزء من المعادلات الرياضية للعمليات الإحصائية على برنامج (Android Studio) المستخدمة في تطبيق (Green Land).



```
public class DB_Sqlit extends SQLiteOpenHelper {
  // Intent intent1 = new Intent(this, MapsActivity.class);

    private static final float PREFERRED_WIDTH = 250;
    private static final float PREFERRED_HEIGHT = 250;
    private String image;
    public static final int COL_IMAGE = 2;

    public static final String BDname = "data.db";
    public DB_Sqlit( Context context) { super(context, BDname,  factory: null,  version: 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table mytable (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, name TEXT NOT NULL, email TEXT NOT NULL)")



    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS mytable");
onCreate(db);
    }

    public boolean insert_Data  (String name , String email ){

        SQLiteDatabase db = this.getWritableDatabase();
```

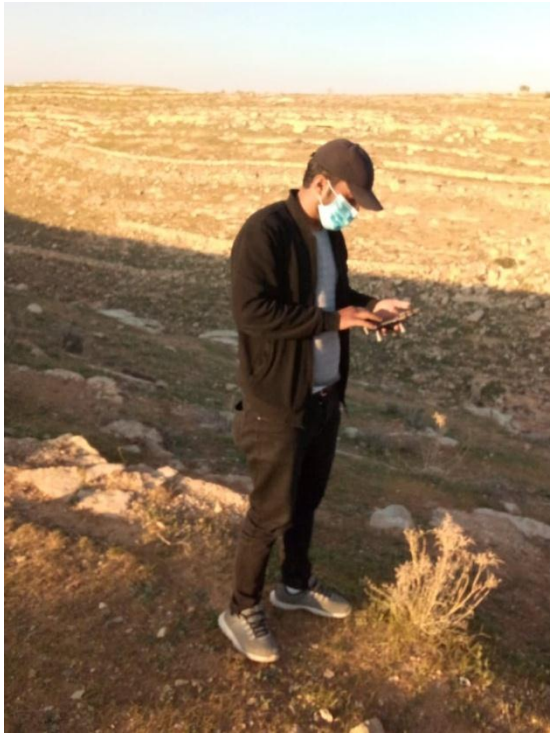
الشكل (5): جزء من الأكواد المستخدمة في عملية برمجة قواعد البيانات على برنامج (Android Studio) الموجودة داخل تطبيق (Green Land).

ثالثًا:

1-العمل الميداني، كما في الشكل(6)، حيث اعتمد العمل الميداني (بشكل أساس) على رصد النقاط من الواقع لبعض المواقع، والحصول على نقاط يتم عليها عملية الدّراسة واستخراج النتائج، حيث تم تقسيم العمل الميداني إلى قسمين:

أ- العمل الميداني من أجل رصد الظواهر، والتي يتم فيها إهمال منسوب أو ارتفاع النقاط عن السطح الجيوئيدي، وتستعمل هذه النقاط من أجل إجراء الدّراسات الإحصائية عليها، مثل: حساب الظاهرة المركزية، أو المركز

المتوسط، أو الانحراف المعياري، والتي لا يكون فيها ضرورة وجود منسوب للنقاط المرصودة؛ إذ يتم في هذه العملية الوقوف على الظاهرة المطلوبة، ثم إتمام عملية الرصد.



الشكل (6): العمل الميداني باستخدام تطبيق (Green Land).

ب-العمل الميداني من أجل رصد النقاط الطبوغرافية، الشكل(7)، والتي تحتوي على منسوب أو ارتفاع ، وتستعمل هذه النقاط في إنشاء مخطّطات طبوغرافية، أو كنتورية، أو إيجاد فروق الارتفاعات بين النقاط لحساب الانحدار أو الميول لسطح الأرض، ولا يمكن هنا إضافة أي ملاحظات أو اسم للنقاط الطبوغرافية في المعلومات الوصفية للنقطة؛ لأن ما يهم في عملية الرصد الطبوغرافي هو منسوب النقطة فقط، ولا يعني ذلك إهمال الإحداثيات الجغرافية، حيث يتم تخزين هذه النقاط في قواعد بيانات مفصولة عن قواعد البيانات الخاصة برصد الظواهر؛ وذلك لتجنّب تداخل عمل الأدوات في ما بينها.

الشكل (7): العمل الميداني باستخدام تطبيق (Green Land)، وجهاز (RTK).

رابعًا:

مقارنة النتائج التي تم الحصول عليها من التطبيق، مع برامج متخصصة، مثل (Arc Map) للتأكد من مدى دقة النتائج، حيث تم إضافة النقاط التي تم رصدها من الواقع إلى برنامج (Arc Map)، وذلك بعد أن تم تصدير هذه النقاط من التطبيق عبر أداة التصدير، حيث كانت النتائج متطابقة عند مقارنة بعض العمليات الإحصائية عليها، وبعد ذلك تم مقارنة دقة رصد التطبيق مع أجهزة متخصصة، مثل جهاز (RTK) الذي تصل فيه دقة الرصد إلى واحد سنتيمتر، بينما كانت دقة الرصد في التطبيق مقبولة نوعًا ما.

خامسًا:

تحليل التطبيقات الحكومية وعلاقتها مع الحلول المكانية، بالإضافة إلى كيفية عملها والأهداف المرجوة من عملها، بالإضافة إلى توفير هذه التطبيقات الوقت والجهد على المستخدمين، حيث ساعدت هذه التطبيقات على تقليل طلب الخدمة مباشرة من الدوائر الحكومية، كما وفرت الحلول المكانية في هذه التطبيقات سهولة في الحصول على الخدمات بشكل دقيق وفي وقت قليل، من خلال استخدام الخرائط وأنظمة

(GPS)، وقواعد البيانات الخاصة بها، بالإضافة إلى ذلك تم التطرق لبعض التطبيقات المملوكة لشركات خاصة، من خلال تفسيرها لفهم الفكرة منها، وإظهار الحلول المكانية الممكنة فيها.

سادسًا:

1- الوصول إلى النتائج ومناقشتها، من خلال إظهار مدى فائدة هذه الدّراسة، ومقدار إثراء المجال الجغرافي بالمعرفة البرمجية اللازمة لتطويرها، بالإضافة إلى إمكانية تطوير هذا الجانب لدى الجغرافيين للارتقاء بهذا المجال الواسع الذي لا يمكن الاستغناء عن البرمجة فيه؛ لأن كمية البيانات الضخمة بحاجة إلى تقنيات تكنولوجية لإدارتها، وتحليلها، ومعالجتها، وتخزينها.

2- التوصيات، تم التطرق في هذا الجزء إلى التوصيات الخاصة بهذه الدّراسة، والتي من شأنها تطوير المجال الجغرافي.

الفصل الثاني

الإطار النظري والدراسات السابقة

1.3 لغة البرمجة:

تعدّ لغة البرمجة لغة التخاطب بين الإنسان والآلة، تتكوّن هذه اللغة من مجموعة من الأوامر والتراكيب، ولها قواعد وأسس يجب مراعاتها عند الكتابة بها، شأنها شأن اللغات العادية، كاللغة العربية والإنجليزية (الأسطل، 2009).

مستويات لغة البرمجة:

1- لغات متدنية المستوى، سميت بذلك؛ لأنها قريبة من لغة الحاسوب (الآلة)، وبعيدة عن لغة الإنسان، حيث تعدّ لغة الآلة لغة متدنية المستوى، وهي اللغة التي تمثل لغة الحاسوب؛ أي اللغة التي يستخدمها الحاسوب لتنفيذ عملياته، وتعتمد تعليمات هذه اللغة على نظام العدد الثنائي، حيث تتكوّن تعليمات هذه اللغة من (1،0)، وتعدّ هذه اللغة صعبة جدًّا على الإنسان من حيث التعامل معها، بالإضافة إلى أن عملية كتابة البرامج باستخدام لغة الآلة يتطلب وقتًا وجهدًا كبيرين من المستخدمين، كما أن صعوبة تتبّع الأخطاء واحتمالية التعرّض لها كبيرة.

2- لغات عالية المستوى، سميت عالية المستوى؛ لأنها قريبة من لغة الإنسان، وبعيدة عن لغة الحاسوب (الآلة)، حيث يتم كتابة تعليماتها بلغة قريبة من لغة الإنسان، ويخاطب الإنسان من خلالها الحاسوب؛ أي أنه ليس هناك داعٍ لتعلّم الإنسان لغة الحاسوب (الآلة) كي يخاطبه، لكن لا بدّ من وجود مترجم (Compiler)، وقد جاءت هذه اللغة لتساعد مبرمجي الحاسوب بتركيز انتباههم على حلّ المشكلة دون التركيز على كيفية كتابة البرنامج وعمل الآلة، ومن الأمثلة على هذه الغات: لغة بيسك (Basic)، ولغة كوبول (Cobol)، ولغة سي (C)، ولغة باسكال (Pascal)، ولغة فورتران (Fortran)، ولغة جافا (Java)، ولغة (Visual Basic)، ولغة (Python)، وغيرها من اللغات الأخرى، حيث يسهُل التعامل مع هذه اللغات من ناحية التعلّم والكتابة،

بالإضافة إلى أنه يمكن تتبّع الأخطاء وتصحيحها، وتوفير الوقت والجهد على المبرمج في كتابة البرامج بجمل قليلة وبسيطة.

2.3 أنظمة التشغيل (Operating Systems) :

نظام التشغيل (Operating System) يشار إليه بالاختصار (OS)، وهو أهم برنامج يتم تشغيله على الكمبيوتر أو الموبايل، وهو الوسيط بين المستخدم وأجهزة الكمبيوتر، ومن خلاله يتم تنفيذ التعليمات والأوامر التي تعطيها لجهاز الكمبيوتر، ويعمل في جميع الأوقات عندما يكون جهاز الكمبيوتر أو الهاتف الذكي قيد التشغيل، ويجب لكل كمبيوتر أو هاتف ذكي أن يكون له نظام تشغيل يقوم بتشغيل وإدارة جميع المكوّنات العامة والبرامج والتطبيقات على الهواتف الذكية.

وبتعريف آخر، يمثّل نظام التشغيل جسرًا لتشغيل برامج المستخدم، ويقوم بالمهام الأساسية، مثل: إدارة وتخصيص مصادر الحاسوب : (الذاكرة، والقرص الصلب، والوصول إلى الأجهزة الملحقة ...)، وترتيب أولوية التعامل مع الأوامر، والتحكم في أجهزة الإدخال والإخراج، مثل لوحة المفاتيح، وكذلك لتسهيل التعامل مع الشبكات، وإدارة الملفّات.

3.3 نظام تشغيل أندرويد (Android) :

هو نظام تشغيل خاص بشركة جوجل (Google) الأمريكية، يدعم الكثير من الهواتف الذكية التي تنتجها شركات مختلفة، مثل: (Samsung) و(HTC)، وهو من النظم مفتوحة المصدر (open source) ، وقد أنتجته (Google) عام (2008)، وتعني كلمة (أندرويد) الروبوت على هيئة إنسان، ويخضع لتحديثات مستمرة، حيث يطلق على كل تحديث اسم مختلف، مثل: (Marshmello) ، (Lollipop)، وأما المتجر الخاص بالتطبيقات فيسمّى بلاي ستور (Play Store). (زهر،2016)

تستند واجهة مستخدم نظام أندرويد أساسًا إلى المعالجة المباشرة، باستخدام الإيماءات اللمسية التي تتوافق إلى حدّ كبير مع الحركات الواقعية، مثل: النقر، والمسح، وضمّ الأصابع، من أجل التعامل مع الأشياء على لوحة الشاشة، بالإضافة

إلى لوحة المفاتيح الافتراضية لإدخال النص، كما قامت شركة جوجل (Google) بتطوير أنظمة الأندرويد التي في أجهزة التلفزيون و(أندرويد أوتو) للسيارات، وأندرويد وير لساعات اليد، وذلك لتوسيع استخدامات نظام الأندرويد في وقتنا اليوم.

3.3 الهاتف الذكي (Smart phones) :

هو مصطلح يطلق على فئة من الهواتف المحمولة الحديثة التي تستخدم نظام تشغيل متطوّر، ومعظمها يستخدم شاشة اللمس واجهة للمستخدم، ويقوم على تشغيل تطبيقات الهاتف المحمول.

يعمل الهاتف الذكي على أنظمة التشغيل، منها: (أندرويد)، و(IOS)، و(ويندوز فون)، وتحتوي الهواتف علي جزئين أساسيين مكمّلين لبعضهما، وهما: (Hardware)، و(Software).

يمثّل الجزء المادي (Hardware) في الهاتف الذكي المكوّنات المادية أو الملموسة فيه، أما (Software) فهو عبارة عن مصطلح حاسوبي يطلق على مجموعة البرمجيات التي يتم إجراؤها لغايات حلّ مسائل جغرافية أو رياضية، أو تنفيذ عملية إحصائية.

5.3 تطبيقات الهواتف الذكية (Mobile Apps):

تعدّ التطبيقات برامج حاسوبية صمّمت لتشغيل الهواتف الذكية وأجهزة التابلت، وهي منتشرة كثيرًا في المجتمع، وتلعب دورًا مهمًّا في حياتنا اليومية، وتمكّن المستخدم من إنجاز مهماته اليومية، وكأنه على جهازه الشخصي، مثل: تصفّح واستخدام البريد الإلكتروني الشخصي، وتكون هذه التطبيقات متاحة على متاجر تطبيقات الهواتف الذكية، مثل: جوجل بلاي وأبل ستور وغيرها، وتطوير تطبيقات الهواتف الذكية وتصميمها يعتمد على نوع الجهاز نفسه (Taba, 2014).

تعدّ تطبيقات الهواتف الذكية مجالًا مستقلًّا عن أنظمة البرامج المتكاملة الموجودة عمومًا في أجهزة الكمبيوتر، حيث يوفر كل تطبيق وظائف محدودة ومعزولة،

مثل: لعبة، أو آلة حاسبة، أو تصفّح الويب على الهاتف الذكي، بالإضافة إلى ذلك تجنّبت التطبيقات تعدّد المهام بسبب موارد الأجهزة المحدودة للأجهزة المحمولة.

6.3 برنامج أندرويد استوديو (Android Studio) :

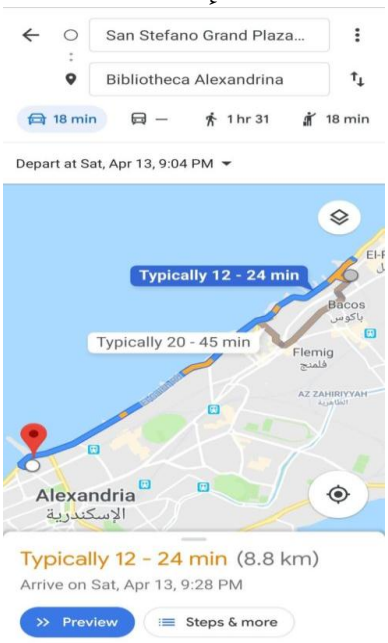
أندرويد استوديو هي منصة لكتابة التطبيقات تسهّل على المطوّرين كتابة الشيفرة المصدرية لتطبيقات (أندرويد)، كما تسمح للمطوّر بمعاينة هيئة تطبيقه على مختلف قياسات الشاشات بشكل فوري أثناء التطوير، وتسهّل عملية تطوير التطبيقات متعدّدة اللغات .(ويكيبديا،2021)، كما يعدّ برنامج أندرويد استوديو (Android Studio) بيئة تطوير متكاملة وجديدة تم إطلاقها مؤخّرًا من قبل نظام (Google) لتشغيل تطبيقات (الأندرويد)، وقد تم تصميمها لتوفير أدوات جديدة في تطوير التطبيقات، بالإضافة إلى تسهيل معاينتها قبل الشروع في نشرها على متجر (بلاي).
وهناك ميزة أخرى في برنامج أندرويد استوديو (Android Studio)، وهي الأدوات التي تتعامل مع تعبئة وتوسيم التعليمات البرمجية، والتي تتيح لك التحكم الكلّي في المشروع الخاص بك، في حال التعامل مع عدد كبير من التعليمات البرمجية، ويستخدم البرنامج، أيضًا، نظام السحب والإفلات لنقل المكوّنات في جميع أنحاء واجهة المستخدم، بالإضافة إلى ذلك، فهذه البيئة الجديدة متوفرة مع سحابة جوجل (Google) للتراسل، وهي الميزة التي تمكّنك من إرسال البيانات من الخادم إلى أجهزة (الأندرويد) من خلال السحابة.

7.3 بعض الأدوات المهمة في برنامج أندرويد استوديو (Android Studio) :

خرائط (Google) :

خرائط (Google) الشكل(8)، هي خدمة مجانية مقدمة من موقع البحث (Google) على الشبكة العالمية، حيث كانت هذه الخدمة توفر عند بدايتها خرائط لشوارع أربع دول، هي: الولايات المتحدة، وكندا، والمملكة المتحدة، وإيرلندا، بالإضافة إلى أن الشخص يستطيع مشاهدة صور عالية الوضوح بوساطة الصور التي توفرها الأقمار الصناعية لمئات من المدن في العالم، أما الآن فلقد توسّعت شبكة الطرق والشوارع لتشمل معظم مدن العالم، وليس للدول الأربع التي بدأ فيها المشروع فقط.

توفر هذه الخرائط في الوقت الحالي مزايا عديدة إضافية، مثل: مشاهدة خطوط الكنتور، وشبكة الطرق، وأسماء المعالم، وإمكانية الاستعلام عن المواقع من خلالها.



الشكل(8): الخرائط التي توفرها شركة (Google).

تعدّ الخرائط الإلكترونية منظومة تكنولوجية متكاملة تعبّر عن حالة تطورية لأدوات الجيل الثاني للإنترنت، وتتسم قدراتها بالجاهزية، والتوفر الدائم، والحداثة، والانتشار، والمجانية، والتفاعلية، والإتاحة عبر شبكة الإنترنت وتطبيقات الهواتف النقالة والأجهزة اللوحية، وتزداد فاعليتها ووظيفيتها في مجالات حياتية مختلفة يوميًّا.

تمكّن هذه التقنية الجغرافيين (في يومنا هذا) من تسهيل دراسة الظواهر الطبيعية من خلال تحديد مواقعها بوساطة الإحداثيات الجغرافية التي توفرها لاستخراج البيانات المطلوبة، ودراسة العلاقات بينها، أو طريقة انتشارها.

نظام التموضع العالمي (Global Positioning System) :

ويرمز له بـ(GPS) ، وهو نظام ملاحة عبر الأقمار الصناعية، يقوم بتوفير معلومات عن الموقع والوقت في جميع الأحوال الجوية، وفي أي مكان على الأرض أو بالقرب منها، من خلال منظومة الأقمار الصناعية (GPS) .

وهو نظام أمريكي للملاحة اللاسلكية يتخذ من الفضاء قاعدة له، وهو نظام يوفر لجميع مستخدميه المدنيين (في جميع أنحاء العالم على نحو مستمر ودون

انقطاع) خدمات مجانية لتحديد الموقع وتحديد الوقت؛ إذ باستطاعة أي شخص لديه جهاز استقبال لنظام التموضع العالمي أن يحصل على معلومات تحدّد له الموقع والتوقيت، كما يوفر النظام هذه المعلومات ليلًا ونهارًا في أي مكان من العالم بغض النظر عن الظروف الجوية.

ملفّات (Notepad) المفكرة:

تعدّ المفكرة من الأساليب المستخدمة في تسجيل الإحداثيات الجغرافية والمعلومات الوصفية بطريقة تنسيقية معينة تستطيع البرامج من خلالها فهم هذه الملفّات، من أجل استخراج المعلومات وعرضها، أو معالجتها، أو حفظها، حيث تحفظ هذه الملفات بالامتداد (txt) ، كما يمكن مشاركة هذه الملفّات بشكل سهل وبسيط بين البرامج والتطبيقات.

قواعد البيانات :

قاعدة البيانات هي مجموعة منظّمة من المعلومات المهيكلة، أو البيانات المخزنة عادةً بصيغة إلكترونية أو في نظام كمبيوتر، وعادةً ما تكون قاعدة البيانات تحت تحكّم نظام إدارة قاعدة بيانات(DBMS)، وتتم الإشارة، معًا، إلى البيانات ونظام إدارة قواعد البيانات جنبًا إلى جنب مع التطبيقات المرتبطة بهما باعتبارها نظام قواعد بيانات، وغالبًا ما يتم اختصاره إلى قاعدة بيانات فقط.

تتم صياغة البيانات ضمن الأنواع الأكثر شيوعًا من قواعد البيانات المستعملة اليوم على هيئة صفوف وأعمدة في سلسلة من الجداول لإضفاء الفاعلية على المعالجة والاستعلام عن البيانات، ويمكن، حينئذٍ، الوصول إلى البيانات، وإدارتها، وتعديلها، وتحديثها، والتحكم فيها، وتنظيمها بسهولة، حيث تستخدم غالبية قواعد البيانات لغة الاستعلام المركّبة (Sequential Query Language - SQL) لكتابة البيانات والاستعلام عنها.

يمكن من خلال قواعد البيانات تخزين المعلومات المكانية والوصفية داخلها واستدعاء هذه المعلومات في أي وقت، حيث تمكّننا قواعد البيانات من إدارة البيانات بشكل سلس وبسيط، وسرعة في الاستدعاء والتعديل والبحث، وتضمن لنا، أيضًا، عدم

التداخل بين البيانات، أو فقدان أي من المعلومات الجغرافية المتعلقة بالظواهر الجغرافية.

8.3 المكوّنات العامة لبرنامج أندرويد استوديو (Android Studio) :

يعدّ الهدف الرئيس، أو القاعدة الأساسية من عمل الدّراسة، القيام بتجسيد الحلول المكانية في أكواد برمجية يمكن من خلالها الوصول إلى تطبيق يستطيع التعامل مع الظواهر الموجودة في العالم الحقيقي .

تعدّ الأكواد البرمجية، أو الشيفرة البرمجية، هي البنية الأساسية لتكوين أي تطبيق مراد تصميمه، حيث تعرّف الأكواد على أنها مجموعة من التعليمات المكتوبة، أو الأوامر التي نتمكّن من خلالها من تنفيذ العمليات المطلوبة وتكوين الأشكال والتصاميم لعمل التطبيق.

تتكوّن الأكواد من مجموعة من المفاهيم المشتركة بين جميع لغات البرمجة، مثل:

1- الثوابت: وهي قيم يتم تعريفها في بداية العمل البرمجي، ولا تتغيّر في أي مرحلة من مراحل تنفيذ التطبيق، مثل الثوابت الرياضية.
2- المتغيّرات: عكس الثوابت؛ فهي قيم قابلة للتغيّر في الذاكرة أثناء عمل التطبيق حين يتطلّب الأمر ذلك.
3- الجمل الشرطية: وهو اشتراط يتطلّب تحقيق الشرط المطلوب لتنفيذ العملية المطلوبة.
4- الدوال: هي مجموعة أوامر أشبه ببرنامج صغير يقوم بمهمة واحدة فقط.
5-الحلقات التكرارية: هي عملية تنفيذ أجزاء محدّدة من الكود البرمجي عدة مرات، محدّدًا لها متى تبدأ، ومتى تنتهي، وعدد مرات التكرار.

تحتاج عملية البرمجة داخل بيئة (أندرويد) إلى تعلّم بعض اللغات للقدرة على التعامل مع الأدوات وبرمجتها لصناعة التطبيقات، مثل: لغة جافا، ولغة (XML)، و(CSS)، حيث تجتمع جميع هذه اللغات لتكوين التطبيق اللازم للعمل.

تقسم العمليات داخل بيئة (أندرويد استوديو) إلى عدة أقسام، وهي:

1- النشاط (Activity): وهي عبارة عن نافذة متعددة قد تكون صغيرة، أو ملء الشاشة لتوضع الكائنات عليها، وهي تمثل واجهة المستخدم، حيث تتعامل مع أربع وضعيات: (عرض بشكل سطري، ونموذجي، وعلائقي، وجدولي).

2- الحدث (Intent): هي عملية لاستدعاء نشاط (Activity) جديد آخر، وهي بمثابة عملية ربط بين أكثر من نشاط .

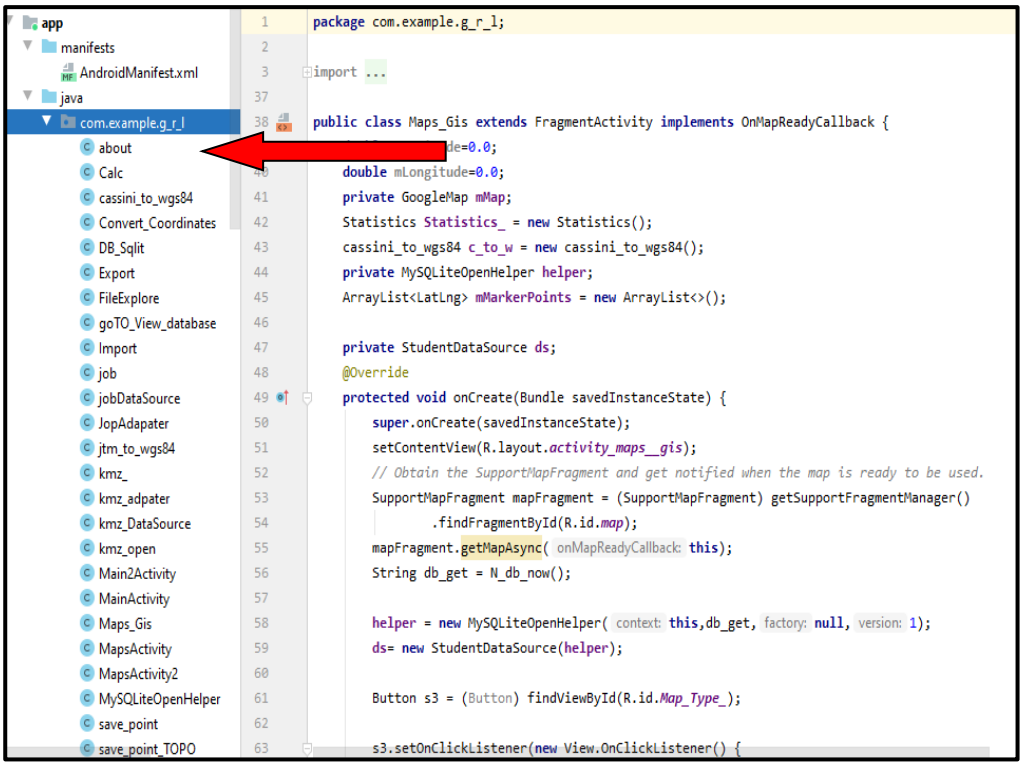
3- الخدمة (Service): تستخدم لوضع خدمات معينة في الخلفيّة أثناء العمل على نشاط آخر، مثل تشغيل الموسيقى في الخلفيّة أثناء تصفح البريد الإلكتروني.

4- تزويد وتبادل المحتوى (Provider Content): يمكن من خلال هذه العملية مناداة أي محتوى خارجي، أو تبادل المحتويات مع تطبيق آخر، مثل استدعاء قائمة العناوين من خلال الاتصال مع تطبيق (Contacts).

يوفر برنامج (أندرويد استوديو) أسلوب عمل مميّز وبسيط نوعًا ما للقيام بتصميم التطبيقات وتطويرها، من خلال مجموعة من الطرق المستخدمة في عملية البرمجة، مثل: الترتيب، والربط، وسهولة استخراج الأخطاء، لذلك يتم التعامل داخل برنامج (الأندرويد استوديو) مع عدة أمور واجب تعريفها لتسهيل فهم عملية البرمجة، ومنها:

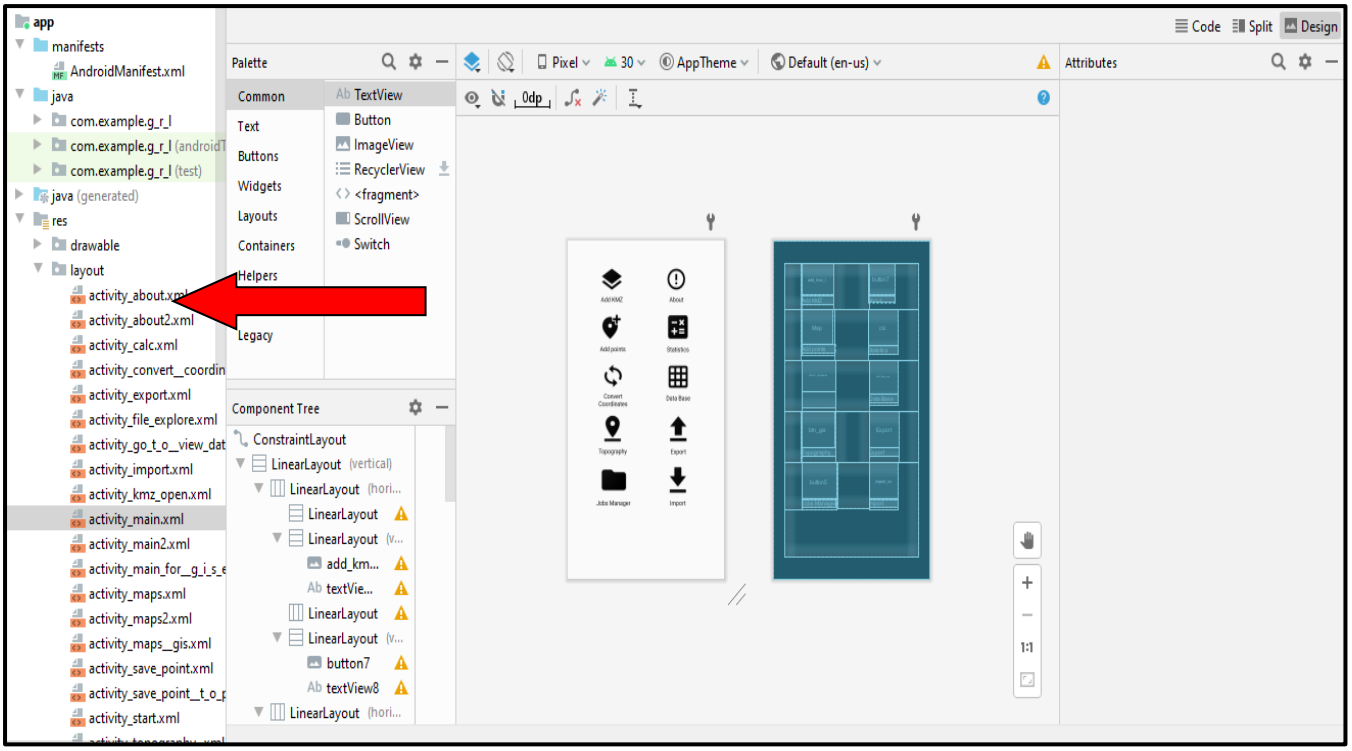
1- (Main Activity): وهي عبارة عن (Class)، أو ملفّات (Java)، وهي الأكواد أو النصوص البرمجية المكتوبة، أو التي يتم كتابتها بلغة (الجافا)، الشكل (9)، والتي من خلالها يتم تكوين العمليات والأوامر داخل التطبيقات، والتي تكون مرتبطة مع الواجهة التي تظهر للمستخدم.

2- (Layout): وهي الواجهة التي تظهر للمستخدم ويتم التفاعل معها، والمبنية على لغة (XML) التي تستخدم للتصميم والتنسيق ، الشكل (10).



الشكل(9): النشاط (Main Activity).

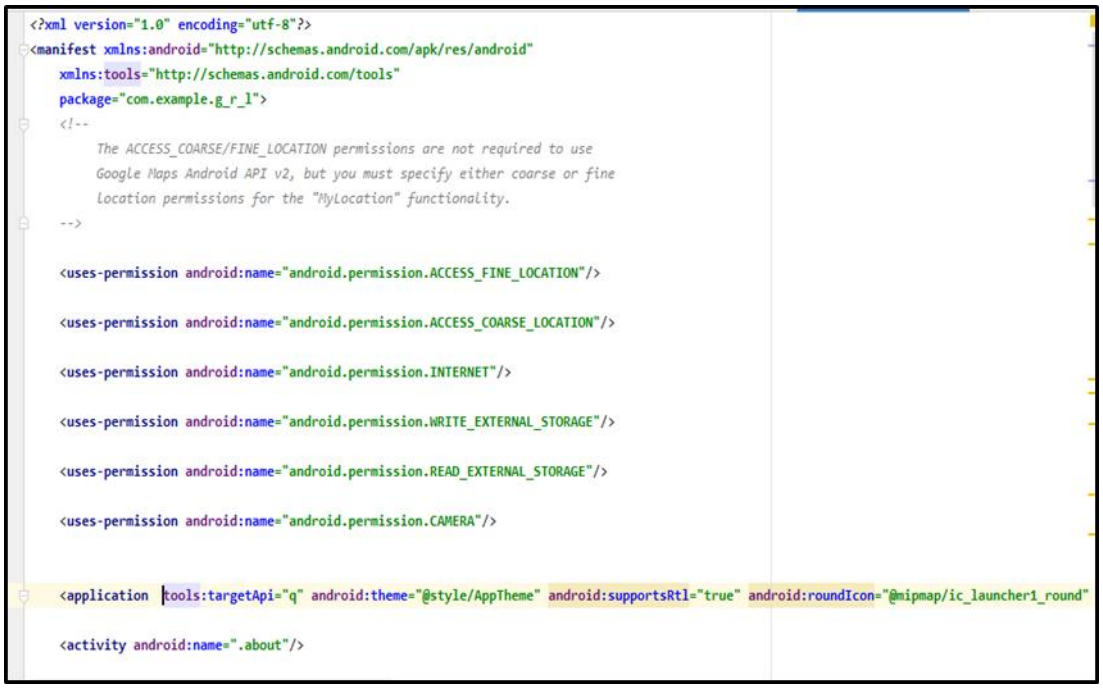
المصدر : برنامج (الأندرويد استوديو).



الشكل(10): الواجهة التفاعلية (Layout).

المصدر : برنامج (الأندرويد استوديو).

3- (Android Manifest): هو الجزء الذي يصف المكوّنات المختلفة للتطبيق، بما في ذلك الأنشطة، والخدمات، وأجهزة استقبال البثّ، ومزودو المحتوى الذين يشكّلون التطبيق، كما يقوم بتسمية الفئات التي تنفذ كل مكوّن وتنشر وظائفها، مثل رسائل (Intent) التي تمكّنهم من التعامل معها، كما تخبر هذه البيانات نظام (Android) عن المكوّنات والظروف التي يمكن إطلاقها في ظلها، بالإضافة إلى الأذونات التي يجب أن يمتلكها التطبيق للوصول إلى الأجزاء المحمية من واجهة برمجة التطبيقات والتفاعل مع التطبيقات الأخرى، الشكل(11).



```
<?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.example.g_r_l">
    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the "MyLocation" functionality.
    -->

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>

    <uses-permission android:name="android.permission.INTERNET"/>

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <uses-permission android:name="android.permission.CAMERA"/>


    <application tools:targetApi="q" android:theme="@style/AppTheme" android:supportsRtl="true" android:roundIcon="@mipmap/ic_launcher1_round"

    <activity android:name=".about"/>
```

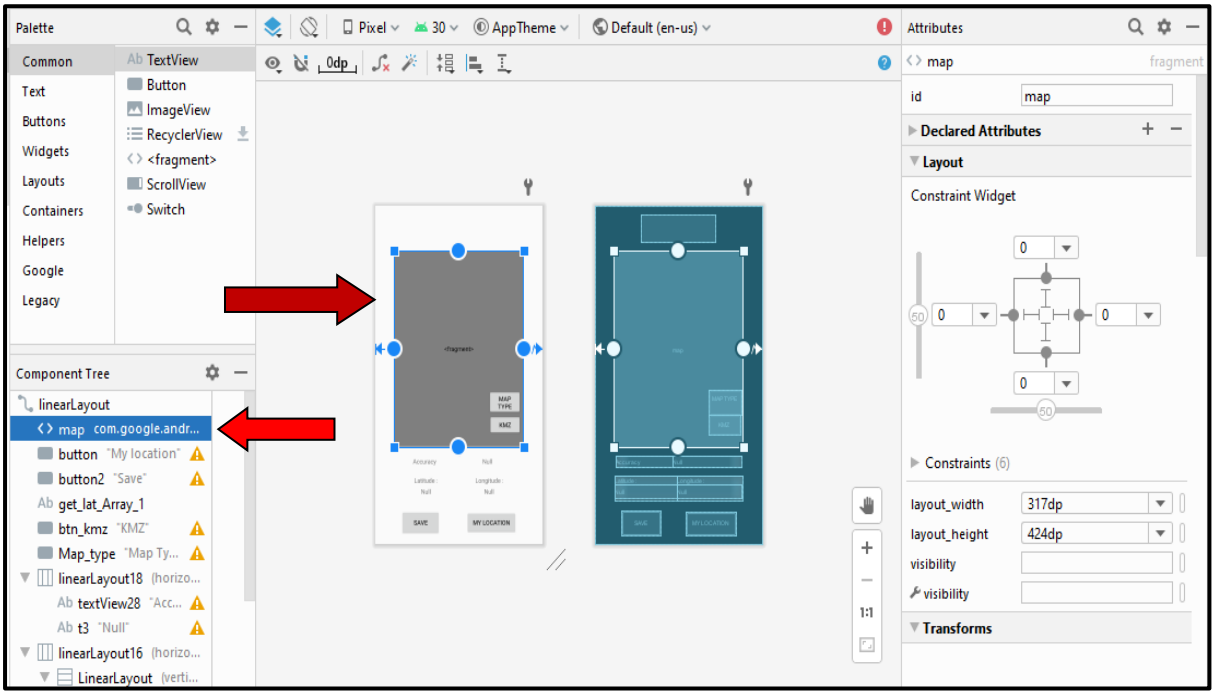
الشكل(11): (Android Manifest).

المصدر : برنامج (الأندرويد استوديو).

9.3 برمجة الأدوات الجغرافية داخل برنامج أندرويد استوديو (Android Studio):

يتم تكوين الحلول المكانية داخل برنامج (الأندرويد استوديو) بوساطة الأكواد البرمجية التي تقوم بدورها بالتعامل مع المتغيّرات والثوابت والجمل الشرطية، وجميع الأمور اللازمة لتكوين التطبيق، حيث يمكن ربط الأدوات التي يوفرها برنامج (أندرويد استوديو)، مثل: الخرائط، ونظام الملاحة العالمي، وقواعد البيانات، والأزرار، من خلال الأكواد البرمجية.

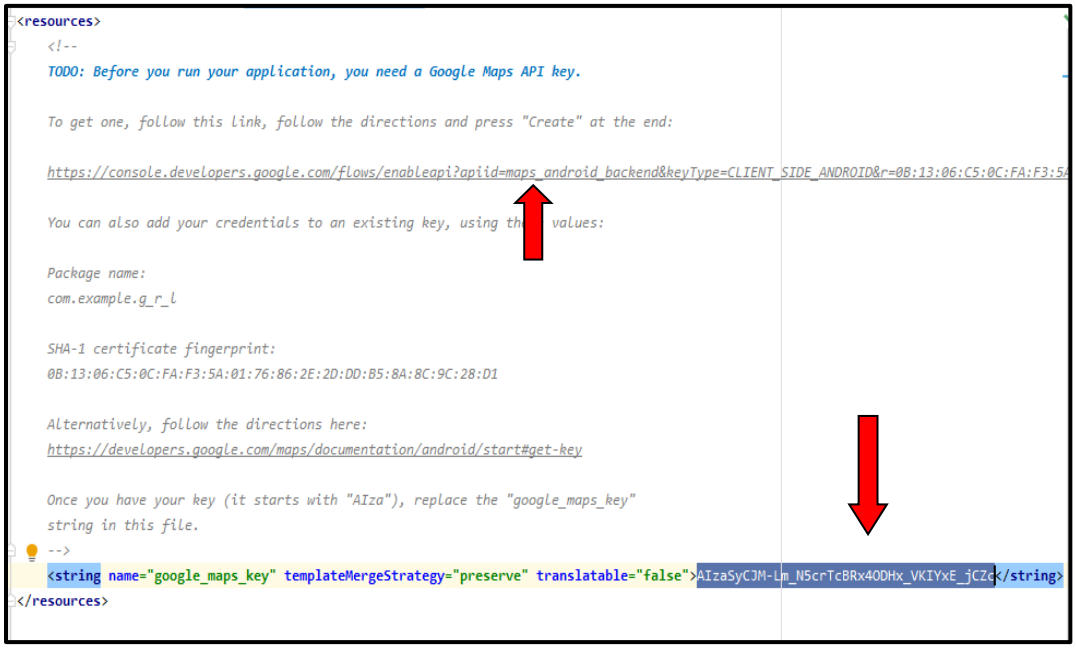
تعدّ خرائط (Google)، الشكل (12)، من الأدوات المهمة داخل التطبيق التي تظهر لنا تصوّرًا واقعيًّا للعالم الحقيقي، حيث لا يخلو أي تطبيق يتعامل مع الحلول المكانية من هذه الخرائط، لذلك يوفر برنامج (أندرويد استوديو) مكتبة خاصة بالخرائط تسمّى (Google Maps API) التي يمكن من خلالها التعامل مع الخرائط داخل التطبيق.



الشكل(12): خرائط (Google) داخل برنامج (الأندرويد استوديو).

المصدر : برنامج (الأندرويد استوديو).

يمكن إدارة الخرائط من خلال المنصة أو الموقع الإلكتروني الذي يوفره برنامج (أندرويد استوديو)، حيث يتطلّب تفعيل الخرائط رمز تسلسلي مميّز يتم تزويده من خلال الموقع الإلكتروني الذي يظهر في الشكل (13).



الشكل(13): الرمز التسلسي المميّز والموقع الإلكتروني الخاص بالمنصة .

المصدر : برنامج (أندرويد استوديو).

يعدّ الرمز التسلسلي مميّزًا لكل تطبيق، ويختلف عن غيره من تطبيق لآخر؛ لتجنّب التداخل في استخدام الميزات الخاصة بالخرائط على المنصة، وتجنّب التداخل بين التطبيقات الأخرى.

أما في ما يخص نظام الملاحة العالمي (GPS)، فتتم برمجة هذه الأداة بطريقة الاستعلام عن الموقع، من خلال برمجة زر يمكن من خلاله استدعاء الإحداثيات الجغرافية الحالية لموقع الجهاز النقال ليتم إضافة نقطة على الخريطة، الشكل (14).

```
@SuppressLint("MissingPermission")
private void getLastLocation() {
    if (checkPermissions()) {
        if (isLocationEnabled()) {
            mFusedLocationClient.getLastLocation().addOnCompleteListener(
                    new OnCompleteListener<Location>() {
                        @Override
                        public void onComplete(@NonNull Task<Location> task) {
                            Location location = task.getResult();
                            if (location == null) {
                                requestNewLocationData();
                            } else {
                                // latTextView.setText(location.getLatitude()+"");
                                ///  lonTextView.setText(location.getLongitude()+"");
                                // findNearest(places, new LatLng(location.getLatitude(), location.getLongitude()));
                                s11 =  location.getLatitude();
                                s22 =  location.getLongitude();



                            }
```

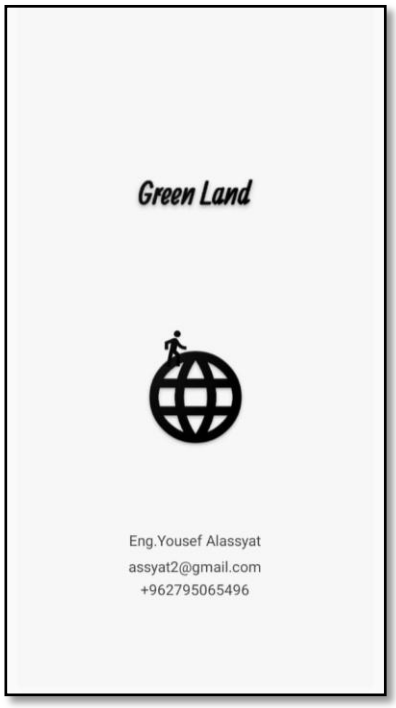
الشكل(14): الكود البرمجي الخاص باستدعاء الإحداثيات الجغرافية من نظام الملاحة العالمي .

المصدر: برنامج (أندرويد استوديو).

حيث يتم تخزين دائرة العرض في المتغيّر (s11)، وخط الطول في المتغيّر (s22) .

10.3التطبيق المعدّ لهذه الدّراسة والبنية الهيكلية للحلول الجغرافية (Green Land):

تطبيق(Green land) الأرض الخضراء الشكل(15) ، تم تصميم هذا التطبيق من أجل استكمال الأفكار المراد توريدها في هذه الدّراسة، حيث تم تصميم التطبيق بلغة (أندرويد) بواجهات بسيطة من أجل إيجاد حلول جغرافية للجغرافيين في الأعمال الميدانية، أو حتى المكتبية، ويمكن تحميل التطبيق على الأجهزة التي تدعم نظام (أندرويد) وخدمات سوق (بلاي)، من أجل الاستفادة من الخدمات التي يقدّمها التطبيق.



الشكل(15): واجهة تطبيق (Green Land).

يحتوي التطبيق على أربعة أقسام رئيسة، وهي: (رفع البيانات، وتوقيع البيانات، والإحصاء، وقواعد البيانات):

1-رفع البيانات:

تعدّ عملية جمع البيانات من العالم الحقيقي أو الميدان من الأمور الرئيسة والمهمة أثناء العمل على التطبيق، حيث تقوم عملية الرفع على تسجيل البيانات الوصفية والمكانية للظواهر الجغرافية أثناء عملية التجوّل الميداني.

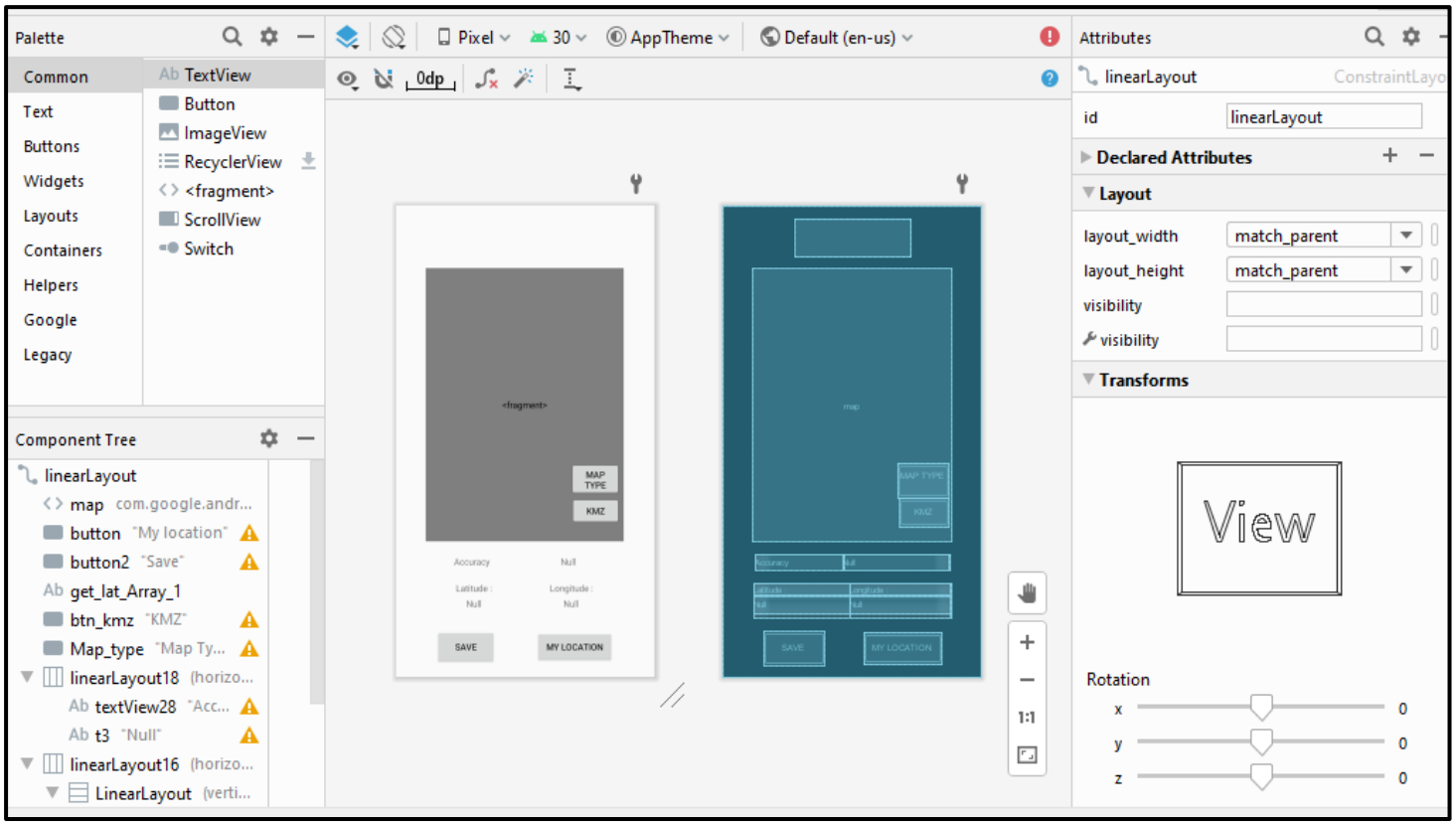
تتم برمجة عملية رفع البيانات من خلال تصميم نشاط (Activity) لكتابة الأكواد، بالإضافة إلى واجهة تفاعليه (Layout) خاص بالخريطة يتم تصميم فيها

زر تحديد الموقع، وزر حفظ النقطة، وزر خاص بشكل الخريطة المراد عرضها، حيث يتم في البداية تصميم الواجهة التفاعلية (Layout)، الشكل(17)، ثم تتم كتابة الأكواد البرمجية في ملف النشاط، كما يظهر في الشكل (16).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
    mapFragment.getMapAsync( onMapReadyCallback: this);
    //  DB_Sqlit db = new DB_Sqlit(this);
    pd = new ProgressDialog( context: MapsActivity.this);
    pd.setMessage("loading");

    String db_get = N_db_now();


    helper = new MySQLiteOpenHelper( context: this,db_get, factory: null, version: 1);
    ds= new StudentDataSource(helper);
    ds1= new kmz_DataSource(helper);

    mFusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this);
    getLastLocation();

    t11 = (TextView)findViewById(R.id.t1);
    t22 = (TextView)findViewById(R.id.t2);
    t33 = (TextView)findViewById(R.id.t3);
```

الشكل(16): الكود البرمجي الخاص برفع البيانات .

يتم تصميم الواجهة التفاعلية (Layout) من خلال إضافة وترتيب مواقع الأزرار ومكان ظهور النصوص وموقع الخريطة عليها، الشكل (17).



الشكل(17): الواجهة التفاعلية (Layout) الخاصة برفع البيانات.

يمكن تقسيم عملية الرفع إلى عدة أقسام فرعية، وهي:

أ- البيانات المكانية:

هي معلومات تحدد الموقع الجغرافي او المكاني للكائن على سطح الأرض وتخضع لعمليات التحليل المكاني(داود،2019).

وتتكوّن البيانات المكانية في هذا التطبيق من إحداثيات جغرافية (خط طول ودائرة عرض)، وتعدّ الإحداثيات المرفوعة من الواقع إحداثيات جغرافية مقاسة بنظام الزاوية، والسبب في ذلك يعود إلى أن نظام (GPS)، الذي يوفره الهاتف الذكي، نظام جغرافي، وليس محليًّا مقاس بالأمتار، وكذلك بسبب ارتباطها بالأقمار الصناعية التي توفر إحداثيات جغرافية، إما بالدرجات والدقائق والثواني، وإما بالدرجات العشرية.

حيث تعتمد الفكرة، هنا، على الوقوف في مكان الظاهرة في العالم الحقيقي، فيقوم التطبيق بتحديد الموقع بناء على موقع الوقوف بالنسبة لموضع الأقمار الصناعية، بعد ذلك يتم تسجيل الإحداثية داخل التطبيق.

يتم تكوين المعلومات المكانية من خلال الأكواد البرمجية الخاصة بنظام الملاحة العالمي (GPS)، الشكل (18)، حيث يتم استدعاء الإحداثيات الجغرافية لموقع الهاتف الذكي وتخزينها داخل متغيّرات ليتم ربطها مع المعلومات الوصفية.

```
@SuppressLint("MissingPermission")
private void getLastLocation() {
    if (checkPermissions()) {
        if (isLocationEnabled()) {
            mFusedLocationClient.getLastLocation().addOnCompleteListener(
                    new OnCompleteListener<Location>() {
                        @Override
                        public void onComplete(@NonNull Task<Location> task) {
                            Location location = task.getResult();
                            if (location == null) {
                                requestNewLocationData();
                            } else {
                                // latTextView.setText(location.getLatitude()+"");
                                ///  lonTextView.setText(location.getLongitude()+"");
                                // findNearest(places, new LatLng(location.getLatitude(), location.getLongitude()));
                                s11 =  location.getLatitude();
                                s22 =  location.getLongitude();



                            }
```

الشكل(18): الكود البرمجي الخاص باستدعاء الإحداثيات الجغرافية من نظام الملاحة العالمي .

في بداية الأمر، يتم التأكد من الأذونات المطلوبة لاستخدام تقنية (GPS)، حيث تظهر هذه الرسالة عند بداية تثبيت التطبيق على الهاتف الذكي، والتي يطلب فيها التطبيق استخدام تقنية الملاحة العالمية لتحديد المواقع، بعد ذلك يتم التأكد من أن زر تقنية الملاحة العالمي قد تم تشغيله بالفعل أم لا، ثم يتم طلب الإحداثيات الجغرافية لموقع الهاتف الذكي داخل جملة شرطية، وإذا تحقّق الشرط بأنه لا يوجد إحداثيات، يتم طلب الإحداثيات مرة أخرى، وإذا كان غير ذلك فيتم تخزين دائرة العرض داخل المتغيّر (s11)، وخط الطول داخل المتغيّر (s22).

وفي المرحلة الثانية، تتم إضافة هذه النقطة على الخريطة بوساطة الكود البرمجي في الشكل (19).

```
    @Override
    public void onSuccess(Location location) {
        // mMap.clear();
        getLastLocation();
        // d.show();

        if (location != null) {

            Double k1 = location.getLatitude();
            Double k2 = location.getLongitude();
            // Double altitude = location.getAltitude();

            t11.setText(String.valueOf(k1));
            t22.setText(String.valueOf(k2));
            t33.setText(String.valueOf(location.getAccuracy()));
            LatLng sydney = new LatLng(k1, k2);
            mMap.addMarker(new MarkerOptions().position(sydney).title("Marker"));
            mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(k1, k2),  v: 17));


        }else {
            Toast toast = Toast.makeText( context: MapsActivity.this,  text: "Please click again ... ", Toast.LENGTH_LONG);
            toast.getView().setBackgroundColor(Color.RED);
            toast.show();
            //  d.cancel();
        }
    }
});
```

الشكل(19): الكود البرمجي الخاص بإضافة الإحداثيات الجغرافية من نظام الملاحة العالمي على الخريطة.

حيث يتم تخزين الإحداثيات في المتغيّرين: (k1، k2)، ثم تتم إضافتها إلى الخريطة بوساطة الأمر (addMarker)، مع تعديل خصائص هذه الإحداثية، مثل: مشاهدة اسم النقطة عند النقر عليها، وتحديد موضع الشاشة ومقدار تركيز موضع الخريطة عليها، بالإضافة إلى إظهار الإحداثيات على شكل نصوص من خلال عارض النصوص (Textview)، مثل: (t11 ، t22)، أما في ما يخصّ (t33)، فيظهر من خلاله دقة تحديد الموقع للنقطة.

ب-البيانات الوصفية:

البيانات الوصفية هي الصفات المميّزة للكائن، أو هي المعلومات الواجب تخزينها عن كائن محدد، وتمثّل بالأعمدة في قاعدة البيانات (داود،2019).

يتم العمل على إضافة البيانات الوصفية للظاهرة في خانتين، هما: خانة اسم الظاهرة، وخانة الملاحظات، حيث تتم إضافة اسم مبدئي كرقم عشوائي داخل الخانة المخصصة للاسم، أما خانت الملاحظات، فتتم إضافة وصف عن الظاهرة في حال

إضافة أي ملاحظات عنها، ويكون الاسم فريدًا أو مميزًا لهذه الظاهرة، ويختلف عن بقية الظواهر الموجودة، ويكون مرتبطًا بالإحداثية الجغرافية أو البيانات المكانية لهذه الظاهرة في حال الاستعلام عنها لاحقًا؛ لتسهيل فهم الظاهرة أو تمييزها عن بقية الظواهر.

كما تتم إضافة خاصية تصوير الظاهرة كبيانات وصفية لهذه الظاهرة، من خلال التقاط صورة لها، ومن ثم ربطها مع البيانات المكانية للقدرة على تمييزها في حال عدم فهم الظاهرة، من خلال الاسم أو الملاحظات المرتبطة بها.

يتم تكوين المعلومات الوصفية من خلال الأكواد البرمجية في نشاط (Activity) منفصل عن السابق، ويتم الانتقال إلية من خلال النقر على زر (Save)، حيث يتم نقل المعلومات المكانية إلى النشاط الخاص بالمعلومات الوصفية، حيث يحتوي النشاط الخاص بالمعلومات الوصفية على واجهة تفاعلية (Layout) خاصة به يمكن من خلالها إضافة المعلومات الوصفية للنقطة التي تم رصدها، الشكل (20).

الشكل(20): واجهة تفاعليه (Layout) خاصة بالمعلومات الوصفية .

أما في ما يخصّ الأكواد البرمجية داخل النشاط (Activity)، كما في الشكل (21)، فتتم إضافة الإحداثيات التي تم نقلها بشكل تلقائي من النشاط السابق الخاص بالبيانات المكانية إلى النشاط الحالي الخاص بالمعلومات الوصفية بوساطة الأمر (Bundle)، ويتم إضافتها داخل محرّر النصوص (Edit Text)، والتي تم تعريفها على أنها (txt_lon ، txt_lat) .

```
    Bundle k = getIntent().getExtras();
    assert k != null;
    String m3 = k.getString( key: "z3");
    String m4 = k.getString( key: "z4");
    EditText txt_lat = (EditText)findViewById(R.id.ed_lat) ;
    EditText txt_lon = (EditText)findViewById(R.id.ed_lon) ;
    EditText txt_notes = (EditText)findViewById(R.id.ed_notes) ;
    EditText txt_name_P = (EditText)findViewById(R.id.ed_name_P) ;
     int min = 1;
     int max = 1000;
     int random = new Random().nextInt( bound: (max - min) + 1) + min;
    txt_name_P.setText(String.valueOf(random));
    txt_lat.setText(m3);
    txt_lon.setText(m4);

}
```

الشكل(21): الكود البرمجي الخاص بالمعلومات الوصفية (استدعاء الإحداثيات).

يتم تسمية النقاط بشكل تلقائي وبرقم عشوائي داخل محرّر النصوص (Text Edit)، والتي تم تعريفها على أنها (txt-name-P)، أما في ما يخصّ الملاحظات، فيتم تسجيلها داخل محرّر النصوص (txt-notes) .

يمكن التقاط صورة للظاهرة من خلال استخدام الكاميرا الخاصة بالهاتف الذكي، وإضافتها للمعلومات الوصفية من خلال النقر على زر التصوير(Take a picture) الذي يقوم بدوره بتشغيل كاميرا التصوير من أجل أخذ صورة للظاهرة، ثم يتم تحويل الصورة إلى نصوص ليتمكّن التطبيق من حفظها من خلال الكود البرمجي في الشكل (22).

```
private static String bitmapToString(Bitmap bitmap) {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.PNG,  quality: 70, baos);
    byte[] b = baos.toByteArray();
    return Base64.encodeToString(b, Base64.DEFAULT);
}

private static Bitmap stringToBitmap(String encodedString) {
    try {
        byte[] encodeByte = Base64.decode(encodedString, Base64.DEFAULT);
        return BitmapFactory.decodeByteArray(encodeByte,  offset: 0, encodeByte.length);
    } catch (Exception e) {
        e.getMessage();
        return null;
    }
}
```

الشكل(22): الكود البرمجي الخاص بتحويل الصورة إلى نصوص والعكس .

إن عملية تخزين الصورة داخل قواعد البيانات يحتاج إلى تحويلها إلى نصوص مشفّرة، ويحتاج عرضها إلى إعادة تحويلها من نصوص مشفّرة إلى صورة، كما يظهر في الشكل (22).

وعند الانتهاء من عملية إضافة المعلومات الوصفية للنقطة يتم تخزينها داخل قواعد البيانات من خلال النقر على زر (Add)، والتي سوف يتم شرح عملية التخزين في قواعد البيانات لاحقًا في هذه الدّراسة.

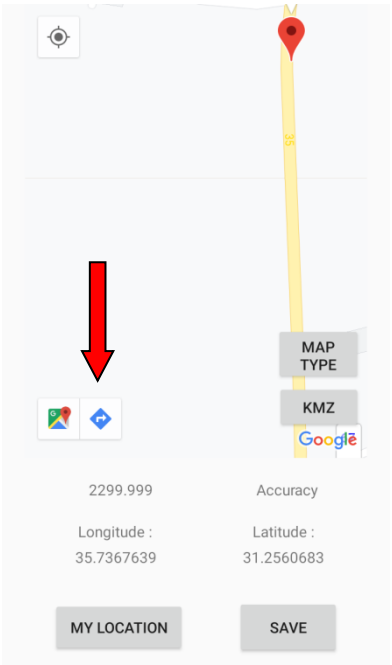
2-توقيع البيانات:

توقيع البيانات هو عملية مساحية يقوم فيها المختص بتنزيل الإحداثيات على الواقع الحقيقي وتثبيتها، بمعنى آخر، هو تحويل البيانات المكتوبة على الأوراق أو الخرائط إلى بيانات أو إحداثيات على الواقع.

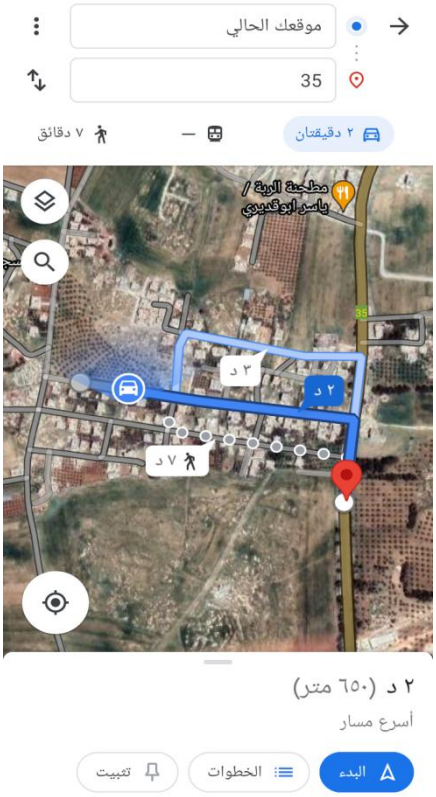
تعدّ عملية التوقيع من العمليات المهمة لتحديد موقع ما، أو الاستدلال عليه للوصول إلى مكان الظاهرة في العالم الحقيقي، من خلال البيانات المكانية بتحديد كونها تحتوي على الإحداثيات الجغرافية التي يستطيع من خلالها التطبيق تحديد موقع الظاهرة، مع العلم أنه لا يمكن الوصول إلى مواقع الظاهرة في العالم الحقيقي إلا من خلال نظام إحداثيات عالميه، وليست إحداثيات محلية؛ كون خرائط (Google) لا

يمكنها التعامل مع الإحداثيات المحلية، ولكن يمكن تحويل هذه الإحداثيات إلى إحداثيات جغرافية من خلال بعض الأدوات التي سوف يتم ذكرها لاحقًا ليتمكّن المستخدم من الوصول إلى الموقع المطلوب.

تتم عملية توقيع النقاط من خلال النقر المطول على النقاط التي تظهر على الخريطة، كما تظهر في الشكل (23)، حيث تظهر أسفل الخريطة أيقونة زرقاء عند النقر عليها، ويتم الانتقال إلى تطبيق خرائط (Google) الذي تظهر فيه النقطة التي تم النقر عليها، بالإضافة إلى المسار المطلوب للوصول إلى النقطة، الشكل(24).



الشكل(23): الأيقونة الزرقاء الخاصة بتحديد المسار على خرائط(Google) .



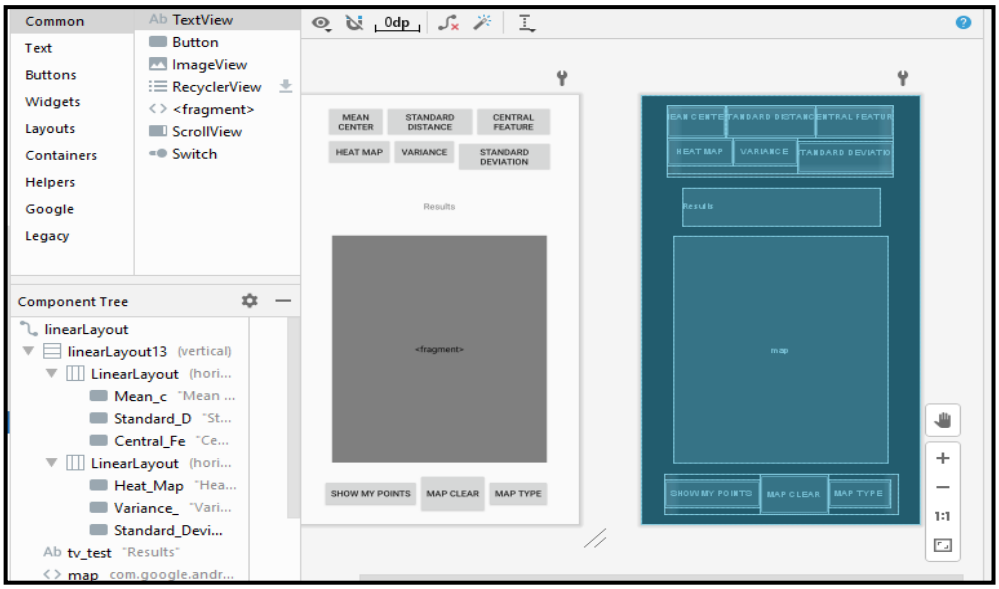
الشكل(24): تطبيق خرائط(Google) كما تظهر عليه المسارات الخاصة بالوصول إلى النقطة .

3-الإحصاء:

الاحصاء هو مجموعة النظريات والطرق العلمية التي تبحث في جمع البيانات وعرضها وتحليلها واستخدام النتائج في التنبؤ أو التقرير واتخاذ القرار (طبيه،2008).

يهتم علم الإحصاء بجمع، وتلخيص، وتمثيل، وإيجاد استنتاجات من مجموعة البيانات المتوفرة ، محاولًا التغلّب على المشكلات، مثل عدم تجانس البيانات وتباعدها، ويمكن، هنا، فهم علاقة الإحصاء، أو أدوات الإحصاء للتحليل المكاني بالحلول الجغرافية، من خلال تفسيريها للظواهر أو العلاقة بين الظواهر، وفهم كيفية انتشارها وسلوكها، ودراسة تشتّتها.

يدعم التطبيق العمليات الإحصائية من خلال واجهة الأدوات الخاصة بالإحصاء. الشكل (25)، حيث تتم برمجة النشاط (Activity) الخاص بالعمليات الإحصائية بوساطة الأكواد البرمجية التي يتم ربطها مع الواجهة التفاعلية (Layout). الشكل (26)، لتكوين الأزرار الخاصة بالأدوات الإحصائية، بالإضافة إلى وجود خريطة (Google) يمكن من خلالها عرض النتائج .



الشكل(25): واجهة تفاعليه (Layout) خاصة بالعمليات الإحصائية .

وتتم إضافة الأزرار الخاصة بالأدوات الإحصائية في الواجهة التفاعلية (Layout) الخاصة بالعمليات الإحصائية، بالإضافة إلى الأزرار الخاصة باستدعاء النقاط التي تم رصدها، وعملية مسح النقاط من على الخريطة، بالإضافة إلى تعديل نوع الخريطة المراد عرضها، كما تتم إضافة محرّر النص (TextView) لعرض النتائج على شكل نصوص.

بعد أن تتم إضافة جميع العناصر إلى الواجهة التفاعلية (Layout) الخاصة بالعمليات الإحصائية، يتم ربطها مع الأكواد البرمجية الموجودة داخل النشاط (Activity) .

```
public void Mean_c_(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Mean Center :"+"\n"+"Easting , Northing"+"\n"+ String.valueOf(Statistics_.
            computeCentroid(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N())));
    view_mean_center();
}
public void Standard_D(View view) {
    view_Standard_Distance();
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Standard Distance _done..");
}
public void Central_Fe(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Central Feature :"+"\n"+"Easting , Northing"+"\n"+
            String.valueOf(Statistics_.Central_Feature_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N())));
    view_Central_Feature();
}
public void Heat_Map_(View view) {
    addHeatMap();
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Heat Map _done..");
}
public void Variance__(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    Double vari = Math.pow((Double.parseDouble(Statistics_.Standard_Distance_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N()))),2);
    tv_clc_.setText("Variance :"+"\n"+ String.valueOf(vari)+" M");
}
public void Standard_Deviation(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Standard Deviation :"+"\n"+ String.valueOf(Statistics_.Standard_Distance_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N()))+" M");
```

الشكل(26): الكود البرمجي الخاص بالأزرار الإحصائية في العمليات الإحصائية.

في بداية الأمر، يتم إنشاء ملف (Java Class) يحتوي على أكواد برمجية خاصة بالعمليات الرياضية، حيث يعدّ ملف (Java Class) قوالب جاهزة من المعادلات الرياضية على شكل أكواد برمجية يتم استدعاؤها داخل النشاط (Activity) الخاص بالعمليات الإحصائية، كما في الشكل (27).

```
public class Statistics {

    public String computeCentroid(String[] E, String[] N) {
        String[] points_E = E;
        String[] points_N = N;
        Double xxx = 0.0;
        Double yyy = 0.0;
        int n = points_E.length;
        for (int x1 = 0; x1 < points_E.length; x1++) {

            xxx += Double.parseDouble(points_E[x1]);
            yyy += Double.parseDouble(points_N[x1]);
        }
        String cc1 = String.format("%.3f", xxx / n) + " , " + String.format("%.3f", yyy / n);
        return cc1;
    }

    public String Standard_Distance_(String[] E, String[] N) {
        String[] points_E = E;
        String[] points_N = N;

        Double xxx = 0.0;
        Double yyy = 0.0;
        Double xxx_sum = 0.0;
        Double yyy_sum = 0.0;
        Double Standard_Distance_Fi = 0.0;
        int n = points_E.length;

        for (int x1 = 0; x1 < points_E.length; x1++) {
```

الشكل(27): الكود البرمجي الخاص بملف (Java Class) الذي يحتوي على قوالب جاهزة من العمليات الإحصائية.

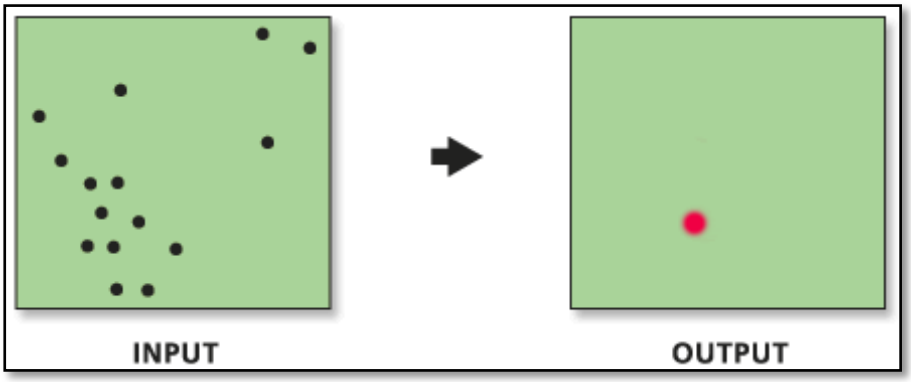
ويمكن فهم انتشار الظواهر التي يتم رفعها من الواقع وتخزينها في قواعد البيانات الخاصة بالتطبيق من خلال مجموعة من الأدوات الإحصائية، وهي:

أ- الظاهرة المركزية (Central Feature):

وهي الظاهرة التي تحدد أقرب ما يكون لمركز توزيع نقاط الظاهرة، كما في الشكل(28).(هلول،2018).

يكمن عمل هذه الأداة داخل التطبيق في نفس المبدأ (سابقاً)، ولكن ضمن النقاط أو الظواهر التي تم تخزينها داخل قواعد البيانات للتطبيق ليتمكّن التطبيق من القيام بالعملية الإحصائية المطلوبة، وإظهار النتائج ليستطيع المستخدم فهم هذه

النتائج، حيث تظهر هذه النتائج مباشرة على الخرائط المدعومة في التطبيق لتسهيل توضيح الظاهرة التي تم تحديدها كظاهرة مركزية، بالإضافة إلى إحداثيات هذه الظاهرة المركزية، حيث يستطيع الجغرافي الاستغناء عن العمل المكتبي لتحليل البيانات وتوفير الوقت والجهد، من خلال استخدام التطبيق في الميدان، وإظهار النتائج مباشرة حين الانتهاء من العمل في الموقع مباشرة .



الشكل(28): الظاهرة المركزية.

تتم برمجة الكود البرمجي الخاص بالظاهرة المركزية في البداية داخل ملف (Java Class)، كما في الشكل (29)، حيث يتم تجهزيه كقالب يتم استقبال الإحداثيات فيه على شكل مصفوفة، ويتم استقبال الإحداثيات في الكود البرمجي داخل مصفوفة (String[ ] E , String [ ] N)، وتعدّ المصفوفات وسيلة لتخزين عدة قيم من نفس النوع داخل متغيّر واحد، وتتميّز عناصر المصفوفة عن بعضها من خلال رقم محدد يعطى لكل عنصر يسمّى (index) أول عنصر فيها يتم تخزينه في (index) رقم (0).

```
blic String Central_Feature_(String[] E, String[] N) {
 String[] points_E = E;
 String[] points_N = N;
 int n = points_E.length;
 double su[] = new double[points_E.length];
 for (int x = 0 ; x < points_E.length;x++) {
     Double ttE_f = Double.valueOf(points_E[x]);
     Double ttN_f= Double.valueOf(points_N[x]);
     Double xxx = 0.0;
     for (int xc = 0; xc < points_E.length; xc++) {
         Double ttE = Double.valueOf(points_E[xc]);
         Double ttN = Double.valueOf(points_N[xc]);
         Double clc = Math.abs(Math.pow(Math.pow((ttE_f - ttE), 2) + Math.pow((ttN_f - ttN), 2), 0.5));
         xxx += clc;
     }
     su[x] = xxx;
 }
 Double first = Double.MAX_VALUE;
 for (int i = 0; i < su.length; i++) {
     if (su[i] < first) {
         first = su[i];
     }
 }
 Double ttE1=0.0;
 Double ttN1=0.0;
 for (int x = 0; x < su.length; x++) {
     if (first == su[x]) {
         ttE1 = Double.parseDouble(points_E[x]);
         ttN1 = Double.parseDouble(points_N[x]);
```

الشكل(29): الكود البرمجي الخاص بالظاهرة المركزية داخل ملف (Java Class) .

في المتغيّر (n) يتم تخزين طول المصفوفة أو عدد القيم الموجودة فيها، ثم يتم إنشاء جملة تكرار (for) يكون عدد التكرارات فيها مساويًا لطول المصفوفة التي تم استقبالها، ثم يتم تكوين جملة (for) أخرى داخل جملة (for) التي تم إنشاؤها في البداية، والتي تعمل بعدد دورات مساوٍ لعدد القيم في المصفوفة في كل دورة للمصفوفة الأولى، والهدف من هذه العملية هو حساب أقل مجموع مسافات تراكمية بين النقاط لكل نقطة، حيث يتم جمع هذه المسافات التراكمية داخل مصفوفة جديدة (su[ ]) لكل نقطة، بعد ذلك يتم تحديد أقل قيمة داخل المصفوفة (su[ ]) في جملة (for)، ويتم تخزين هذه القيمة في المتغيّر (first)، ثم يتم إنشاء جملة (for) تقوم على المقارنة بين القيمة المخزنة في المتغيّر (first) والمصفوفة (su[ ]) لتحديد الإحداثيات التي سوف

تحقق شرط المطابقة في جملة (if)، حيث يتم استدعاء الإحداثيات بناء على (index) الذي تم تحقق الشرط عندها، بعد ذلك يتم تخزين الإحداثيات داخل المتغيّر (cc1) ثم إرجاع القيم أو إرسالها إلى النشاط (Activity)، بوساطة الأمر (return)، كما في الشكل (30).

```
        }
    }
    String cc1 = String.format("%.3f", ttE1) + " , " + String.format("%.3f", ttN1);
    return cc1;
}
```

الشكل(30): الكود البرمجي الخاص بالظاهرة المركزية داخل ملف (Java Class) .

بعد أن يتم استدعاء الإحداثيات الخاصة بالظاهرة المركزية من ملف (Java Class) إلى ملف النشاط (Activity)، كما في الشكل(31)، يتم استخراج الإحداثيات المترية وتحويلها إلى إحداثيات جغرافية بوساطة الملف (Java Class) الخاص بتحويل الإحداثيات (cassini to wgs48)، بعد ذلك يمكن إضافتها إلى الخريطة بوساطة الأمر (addMarker) .

```
public void view_Central_Feature(){
    String cor =(Statistics_.Central_Feature_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N()));
    String cor_split = cor;

    String[] splitted = cor_split.split( regex: ",");

    Double E_s = Double.parseDouble(splitted[0]);
    Double N_s = Double.parseDouble(splitted[1]);
    c_to_w.cassini_to_wgs48_p(N_s,E_s);
    Double lat_s=c_to_w.a10;
    Double lon_s=c_to_w.a11;
    LatLng sydney = new LatLng(lat_s, lon_s);
    mMap.addMarker(new MarkerOptions().position(sydney).title("Central_Feature"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(lat_s, lon_s),  v: 13));


}
```

الشكل(31): الكود البرمجي الخاص بالظاهرة المركزية داخل ملف النشاط (Activity).

يتم ربط جميع الأكواد السابقة الخاصة بالظاهرة المركزية بالأكواد البرمجية الخاصة بالزر المخصص للظاهرة المركزية، بالإضافة إلى عرض الإحداثيات على شكل نصوص داخل عارض النصوص (Text View)، كما في الشكل (32).

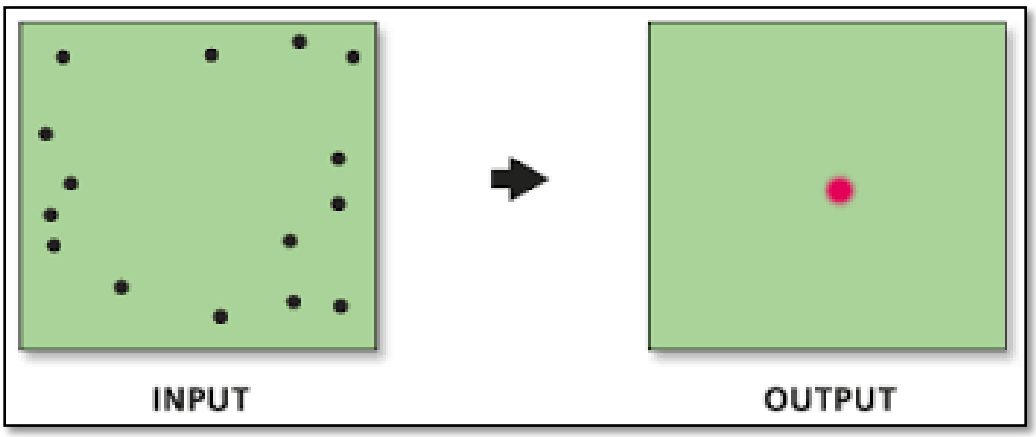
```
public void Central_Fe(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Central Feature :"+"\n"+"Easting , Northing"+"\n"+
            String.valueOf(Statistics_.Central_Feature_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N())));
    view_Central_Feature();
}
```

الشكل(32): الكود البرمجي الخاص بالظاهرة المركزية داخل الزر.

ب-المركز المتوسط (Mean Center):

ويقصد به الموقع الذي يتوسط جغرافياً مواقع باقي الظاهرات المنتشرة المدروسة في منطقة ما، كما في الشكل(33).(هلول،2018).

تعدّ هذه الوظيفة إحدى وظائف النزعة المركزية الهادفة للكشف عن الأنماط النقطية، بغرض إيجاد المركز المتوسط الذي يمثل مركز الثقل للتوزيع المكاني للنقاط، كما تمكّن هذه الأداة الجغرافيين في تحديد المركز الجغرافي للنقاط التي رصدها، أو التي تم رفعها من الواقع لتسهيل عملية التحليل المباشر أثناء العمل الميدان، فمن خلال التطبيق يمكن تحديد المركز المتوسط، وإظهار إحداثياته على الخريطة المدعومة لدى التطبيق.



الشكل(33): المركز المتوسط.

يتم برمجة المركز المتوسط (Mean Center) في البداية داخل ملف (Java Class)، كما في الشكل(34)، حيث يتم استقبال الإحداثيات في الكود البرمجي داخل مصفوفة (String[ ] E , String [ ] N) .

```
public String computeCentroid(String[] E, String[] N) {
    String[] points_E = E;
    String[] points_N = N;
    Double xxx = 0.0;
    Double yyy = 0.0;
    int n = points_E.length;
    for (int x1 = 0; x1 < points_E.length; x1++) {

        xxx += Double.parseDouble(points_E[x1]);
        yyy += Double.parseDouble(points_N[x1]);
    }
    String cc1 = String.format("%.3f", xxx / n) + " , " + String.format("%.3f", yyy / n);
    return cc1;
}
```

الشكل(34): الكود البرمجي الخاص بالمركز المتوسط داخل ملف (Java Class).

يتم تخزين طول المصفوفة أو عدد القيم فيها داخل متغيّر يسمى (n)، ثم يتم إنشاء جملة تكرار (for) يكون عدد التكرارات فيها مساويًا لطول المصفوفة التي تم استقبالها، وفي كل دورة يتم جمع القيم الموجودة في المصفوفة بالترتيب بدءًا من أول عنصر فيها في بداية الدورة الأولى، ووصولًا إلى آخر عنصر في نهاية الدورة، ويتم تخزينها داخل المتغيّر (xxx) للإحداثيات الشرقية، و(yyy) للإحداثيات الشمالية، وبعد أن يتم الانتهاء من الجملة التكرارية يقوم الكود البرمجي على قسمة مجموع الإحداثيات الشرقية على طول المصفوفة التي تمثّل المتغيّر (n)، وقسمة مجموع الإحداثيات الشمالية على طول المصفوفة التي تمثل المتغيّر (n)، وتخزين النتائج داخل المتغيّر (cc1)، ثم إرجاع النتائج بوساطة الأمر (return) إلى ملف النشاط (Activity) الخاص بالعمليات الإحصائية.

بعد أن يتم استقبال القيمة داخل ملف النشاط (Activity)، يقوم الكود البرمجي على تحويل الإحداثيات المترية الخاصة بالمركز المتوسط إلى إحداثيات جغرافية؛ لتتم إضافتها إلى الخريطة، كما في الشكل (35)، من خلال الأمر (addMarker).

```
public void view_mean_center(){
    String cor =(Statistics_.computeCentroid(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N()));
    String cor_split = cor;

    String[] splitted = cor_split.split( regex: ",");

    Double E_s = Double.parseDouble(splitted[0]);
    Double N_s = Double.parseDouble(splitted[1]);
    c_to_w.cassini_to_wgs48_p(N_s,E_s);
    Double lat_s=c_to_w.a10;
    Double lon_s=c_to_w.a11;
        LatLng sydney = new LatLng(lat_s, lon_s);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Mean_center"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
   // mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(lat_s, lon_s),  v: 13));
    }
```

الشكل(35): الكود البرمجي الخاص بالمركز المتوسط داخل ملف النشاط (Activity).

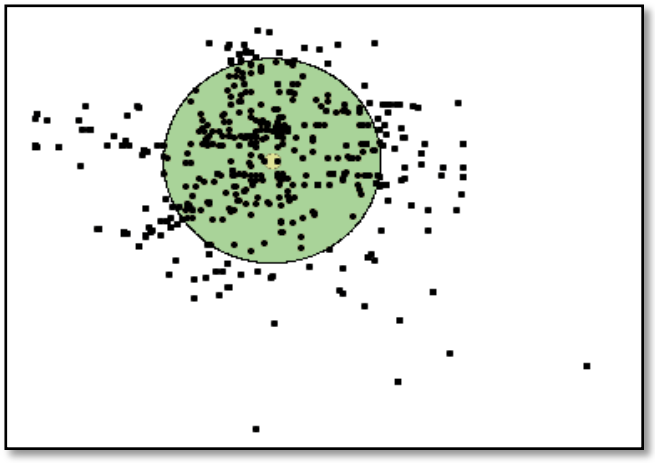
يتم ربط الأكواد (سالفة الذكر الخاصة بالمركز المتوسط) داخل الأكواد البرمجية المرتبطة بالزر الخاص بالمركز المتوسط، بالإضافة إلى عملية إظهار النتائج على شكل نصوص داخل عارض النصوص (Text View)، كما في الشكل (36).

```
public void Mean_c_(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Mean Center :"+"\n"+"Easting , Northing"+"\n"+ String.valueOf(Statistics_.
            computeCentroid(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N())));
    view_mean_center();
}
```

الشكل(36): الكود البرمجي الخاص بالمركز المتوسط داخل الزر.

ج- المسافة المعيارية (Standard Distance):

تعد المسافة المعيارية احدى مقاييس التشتت المكاني وهي تساوي الجذر التربيعي لمربع انحرافات احداثيات مواقع التوزيع المكاني للظواهر الجغرافية عن المتوسط المكاني، الشكل (36).(هلول،2018)



الشكل(36): المسافة المعيارية.

تتم برمجة المسافة المعيارية (Standard Distance) داخل الأكواد البرمجية في ملف (Java Class)، كما في الشكل(37)، حيث يتم استقبال الإحداثيات في الكود البرمجي داخل مصفوفة (String[ ] E , String [ ] N) .

```
public String Standard_Distance_(String[] E, String[] N) {
    String[] points_E = E;
    String[] points_N = N;

    Double xxx = 0.0;
    Double yyy = 0.0;
    Double xxx_sum = 0.0;
    Double yyy_sum = 0.0;
    Double Standard_Distance_Fi = 0.0;
    int n = points_E.length;

    for (int x1 = 0; x1 < points_E.length; x1++) {
        xxx += Double.parseDouble(points_E[x1]);
        yyy += Double.parseDouble(points_N[x1]);
    }
    Double E_Mc = xxx / n;
    Double N_Mc = yyy / n;
    Double[] d_E = new Double[points_E.length];
    Double[] d_N = new Double[points_N.length];
    for (int x11 = 0; x11 < points_E.length; x11++) {
        d_E[x11] = Math.pow(Double.parseDouble(points_E[x11]) - E_Mc, 2);
        d_N[x11] = Math.pow(Double.parseDouble(points_N[x11]) - N_Mc, 2);
        xxx_sum += d_E[x11];
        yyy_sum += d_N[x11];
    }
    Standard_Distance_Fi = Math.pow(((xxx_sum / n) + (yyy_sum / n)), 0.5);
    return String.valueOf(Standard_Distance_Fi);
}
```

الشكل(37): الكود البرمجي الخاص بالمسافة المعيارية داخل ملف (Java Class).

في البداية، يتم إيجاد مجموع الإحداثيات من خلال جملة (for) التي يكون مقدار التكرار فيها مساويًا لطول المصفوفة، ثم تتم قسمة المجموع على عدد القيم في المصفوفة، والتي تمثل إحداثيات المركز المتوسط، وبعد ذلك يتم إنشاء جملة (for) أخرى، حيث يتم من خلالها حساب المسافة المعيارية. وبعد الانتهاء من هذه العملية، يتم تخزين القيمة الناتجة في المتغيّر (Standad- Distance-Fi) التي تمثل المسافة المعيارية، أو نصف قطر الدائرة التي يتم تصميمها على الخريطة، والتي يكون مركزها إحداثيات المركز المتوسط، ثم يتم إرجاع أو إرسال القيمة إلى النشاط (Activity) من خلال الأمر (return).

بعد الانتهاء من الحسابات للمسافة المعيارية داخل ملف (Java Class)، يتم استقبال النتائج في ملف النشاط (Activity)، كما في الشكل (38)، والذي بدوره يقوم

بإسقاط النتائج على الخريطة على شكل دائرة يكون نصف قطرها المسافة المعيارية، وإحداثيات مركزها تكون إحداثيات المركز المتوسط.

```
public void view_Standard_Distance(){
    Double dis_C = Double.parseDouble(Statistics_.Standard_Distance_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N()));
    String cor =(Statistics_.computeCentroid(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N()));
    String cor_split = cor;

    String[] splitted = cor_split.split( regex: ",");

    Double E_s = Double.parseDouble(splitted[0]);
    Double N_s = Double.parseDouble(splitted[1]);
    c_to_w.cassini_to_wgs48_p(N_s,E_s);
    Double lat_s=c_to_w.a10;
    Double lon_s=c_to_w.a11;
    Circle circle = mMap.addCircle(new CircleOptions()
            .center(new LatLng(lat_s, lon_s))
            .radius(dis_C)
            .strokeColor(Color.RED)
            .fillColor(Color.TRANSPARENT));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(lat_s, lon_s),  v: 13));


}
```

الشكل(38): الكود البرمجي الخاص بالمسافة المعيارية داخل ملف النشاط (Activity).

يتم ربط الأكواد (سالفة الذكر الخاصة بالمسافة المعيارية) داخل الأكواد البرمجية المرتبطة بالزر الخاص بالمسافة المعيارية، كما في الشكل (39).

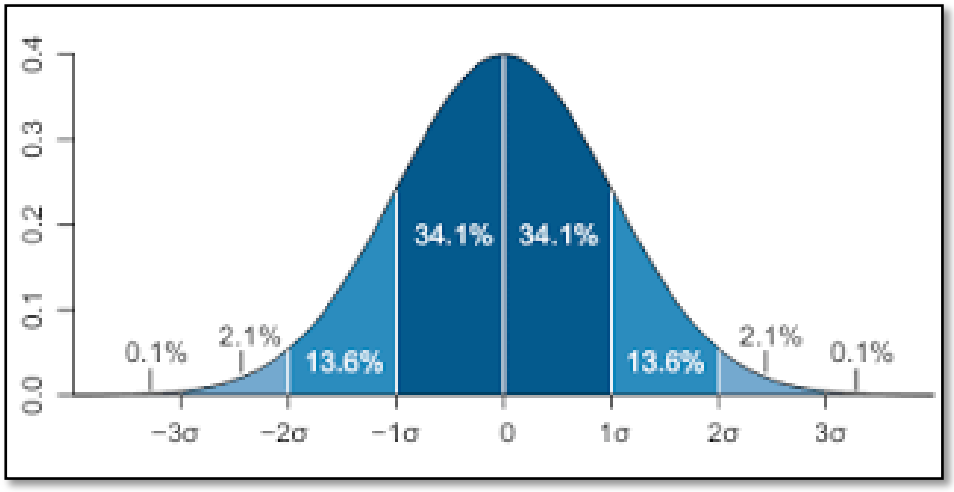
```
public void Standard_D(View view) {
    view_Standard_Distance();
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Standard Distance _done..");
}
```

الشكل(39): الكود البرمجي الخاص بالمسافة المعيارية داخل الزر.

د- الانحراف المعياري(Standard Deviation):

يمكن تعريف الانحراف المعياري، كما في الشكل (40)، على أنه مدى امتداد مجالات القيم ضمن مجموعة البيانات الإحصائية.

يتم استعمال هذه الأداة للتعبير عن مقدار تشتت القيم التي تم رصدها، أو رفعها من الواقع في الميدان، من خلال التطبيق المزود بهذه الأداة الإحصائية.



الشكل(40): الانحراف المعياري.

تتم برمجة الانحراف المعياري (Standard Deviation) كما في الشكل (41)، مباشرة داخل الزر، حيث تعدّ المسافة المعيارية هي نفسها الانحراف المعياري، كما يتم عرض قيمة الانحراف المعياري من خلال عارض النص (Text View).

```
public void Standard_Deviation(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    tv_clc_.setText("Standard Deviation :"+"\n"+ String.valueOf(Statistics_.Standard_Distance_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_lat

}
```

الشكل(41): الكود البرمجي الخاص بالمسافة المعيارية داخل الزر.

هـ- التباين (Variance):

وهو معدل مربعات انحرافات العلامات في التوزيع عن الوسط الحسابي، ويكون الانحراف المعياري عندها الجذر التربيعي للتباين بالنسبة لمجموعة البيانات الإحصائية.

يتم استعمال هذه الأداة للتعبير عن مقدار تشتت القيم التي تم رصدها، أو رفعها من الواقع في الميدان، من خلال التطبيق المزود بهذه الأداة الإحصائية.

تتم برمجة التباين (Variance)، كما في الشكل (42)، مباشرة داخل الزر، حيث يمثل التباين القيمة المربعة للانحراف المعياري، كما يتم عرض قيمة التباين من خلال عارض النص (Text View).

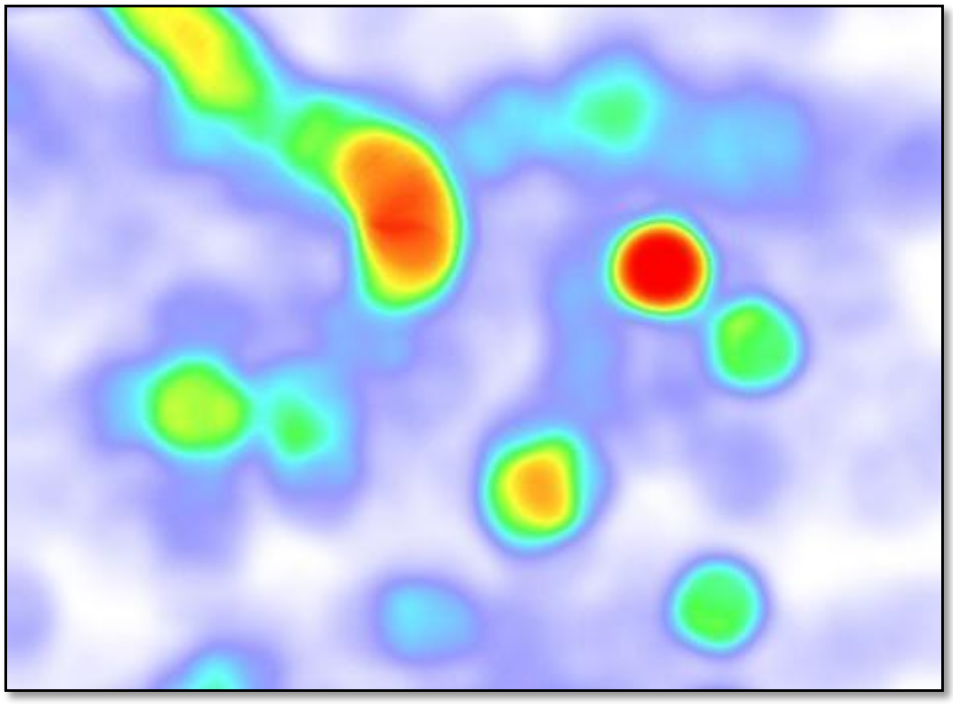
```
public void Variance__(View view) {
    TextView tv_clc_ = (TextView)findViewById(R.id.tv_test);
    Double vari = Math.pow((Double.parseDouble(Statistics_.Standard_Distance_(ds.Get_data_latlon_cassini_clc_E(),ds.Get_data_latlon_cassini_clc_N
    tv_clc_.setText("Variance :"+"\n"+ String.valueOf(vari)+" M");
}
```

الشكل(42): الكود البرمجي الخاص بالتباين داخل الزر.

و- الخريطة الحرارية (Heat map):

هي تقنية تصوّر مدى إنتشار الظواهر على شكل تدرج في الألوان، وقد يكون الاختلاف في اللون حسب درجة الكثافة أو الشدّة؛ ما يعطي إشارات بصرية واضحة للقارئ حول كيفية تجميع الظاهرة أو اختلافها الشكل(43).

تعكس هذه العملية بشكل واضح مقدار كثافة تجمع الظواهر أو تباعدها، حيث يتم تدريج الألوان فيها من المناطق ذات الكثافة العالية إلى الأقل كثافة، وتساعد هذه العملية في فهم انتشار الظواهر على الخريطة أو تجمعّها .



الشكل(43): الخريطة الحرارية.

يتم تكوين الكود البرمجي للخريطة الحرارية داخل ملف النشاط (Activity)، كما في الشكل (43)، والذي يقوم بدوره باستقبال الإحداثيات وتكوين الخريطة الحرارية.

```
private void addHeatMap() {
    ArrayList<LatLng> list_H = new ArrayList<~>();
    list_H = (ArrayList<LatLng>) ds.getAll_HotMap();
    HeatmapTileProvider provider = new HeatmapTileProvider.Builder()
            .data(list_H)
            .build();
    TileOverlay overlay = mMap.addTileOverlay(new TileOverlayOptions().tileProvider(provider));

}
```

الشكل(44): الكود البرمجي الخاص بالخريطة الحرارية داخل ملف النشاط (Activity).

ويتم ربط الكود البرمجي الخاص بالخريطة الحرارية مع الزر المخصص لها لإسقاطها على خرائط (Google).

4-قواعد البيانات (Data Base):

هي مجموعة من عناصرِ البيانات المنطقية المرتبطة مع بعضها بعضًا بعلاقة رياضية، حيث تتكوّن هذه القواعد من جدول أو أكثر، يتم تخزين البيانات فيها بشكل منظّم، وتكون علاقة الربط بين البيانات واحدًا لواحد، أو عنصرًا لعنصر.

إن ما يميّز قواعد البيانات داخل التطبيق، كما في الشكل(45)، هو سهولة البحث، أو إضافة، أو تعديل، أو حذف البيانات التي تم تخزينها أثناء العمل الميداني، كما يتم ربط مجموعة البيانات المخزنة بالخرائط المدعومة في التطبيق لسهولة عرضها وتوضيحها عليها.

الشكل(45): الواجهة التفاعلية (Layout) لقاعدة البيانات داخل تطبيق (Green Land).

تتكوّن قواعد البيانات بشكل عام من نوعين؛ قواعد بيانات تعتمد في عملها على خادم (Server)، مثل (Oracle)، وقواعد بيانات عبارة عن ملف واحد غير معتمد على خادم (Server)، مثل (SQLite).

(SQLite) هي مكتبة مكتوبة بلغة (C) تتيح إنشاء قواعد بيانات على القرص الصلب، ولا تحتاج إلى خوادم خاصة لمعالجتها، وتتيح الوصول إلى قاعدة البيانات باستخدام نسخة غير قياسية من لغة الاستعلام (SQL).

```
public class MySQLiteOpenHelper extends SQLiteOpenHelper {
    public MySQLiteOpenHelper(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table student (_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, lat DOUBLE , lon DOUBLE , notes TEXT NOT NULL ," +
                " img TEXT , nameP TEXT  )");
        db.execSQL("create table job (_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, name TEXT   )");
        db.execSQL("create table topo (_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, lat DOUBLE , lon DOUBLE , z DOUBLE  )");
        db.execSQL("create table kmz_data (_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, Br TEXT   )");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS student");
        db.execSQL("DROP TABLE IF EXISTS jop");
        db.execSQL("DROP TABLE IF EXISTS topo");
        db.execSQL("DROP TABLE IF EXISTS kmz_data");
        onCreate(db);
    }

}
```

الشكل(46): الكود البرمجي الخاص بقواعد البيانات داخل ملف (Java Class) .

تتم برمجة قواعد البيانات في ملف (Java Class)، كما في الشكل (39)، في الكود البرمجي الخاص بقواعد البيانات حيث تم تصميم عدة جداول لاستخدامها في مجموع أدوات، مثل الجدول الخاص بالإحداثيات المرصودة، وتم إنشاء ستة أعمدة فيها، وهي: عمود خاص برقم المعرف أو المميّز (Id)، والعمود الخاص بدائرة العرض (lat)، والعمود الخاص بخط الطول (lon)، والعمود الخاص بالملاحظات (notes)، والعمود الخاص بالصور (img)، والعمود الخاص باسم النقطة (nameP)، وتم إنشاء الجدول الخاص بإدارة المشاريع، والذي تم بداخله إنشاء عمود خاص بالرقم المميّز أو المعرف (Id)، وعمود خاص باسم المشروع (name)، وتم

إنشاء الجدول الخاص بنقاط الرصد الطبوغرافي، والذي تم بداخله إنشاء عمود خاص بالرقم المميّز أو المعرف (Id)، والعمود الخاص بدائرة العرض (lat)، والعمود الخاص بخط الطول (lon)، والعمود الخاص بالمنسوب (z)، وتم إنشاء الجدول الخاص بملفّات (KMZ)، والذي تم بداخله إنشاء عمود خاص بالرقم المميّز أو المعرف (Id)، وعمود خاص برابط موقع الملف على ذاكرة تخزين الجهاز(Br)، كما يقوم الكود البرمجي بإنشاء هذه الجداول مرة أخرى في حال كانت مفقودة أو غير موجودة في الجهاز.

يتم ربط الجداول الموجودة في قواعد البيانات مع الواجهات التفاعلية الخاصة بكل عنصر من الجداول السالفة، مثل الواجهة التفاعلية الخاصة برصد البيانات أو الطبوغرفي أو أداة طبقات (KMZ).

11.3 المكوّنات والأدوات الإضافية داخل تطبيق (Green Land):

يتّسم تطبيق الأرض الخضراء (Green Land) بمجموعة من الأدوات الإضافية التي تساعد في استكمال العمل الميداني، والتي توفر على المستخدم أدوات باهظة الثمن، بالإضافة إلى توفير الوقت والجهد وسهولة التعامل مع البيانات، مثل أداة رفع النقاط مع منسوبها من أجل عمل مخطّطات طبوغرافية أو كنتورية بعد معالجتها مكتبيًّا على برامج متخصصة، بالإضافة إلى إمكانية تصدير النقاط، أو استيرادها من التطبيق وإليه، وإمكانية مشاركتها من خلال البريد الإلكتروني، أو (البلوتوث)، أو حتى (الواي فاي)، بالإضافة إلى وجود أدوات تحويل الإحداثيات من النظام العالمي إلى النظام المحلي، بالإضافة إلى إظهارها على الخريطة لتسهيل المعاينة والقدرة على الوصول إلى الإحداثية المحولة، كما يمكن من خلال التطبيق إضافة ملفّات(KMZ) التي يتم تحويلها إلى برامج متخصصة، مثل: (Global Mapper)، و(Arc GIS)، و(Google Earth)، وإظهارها على الخريطة المدعومة من التطبيق، كما يمكن من خلال التطبيق التحكم في المشروع المراد العمل عليه من قائمة يمكن تحديد فيها المشروع المراد العمل عليه لسهولة التحكم في البيانات وترتيبها وسهولة الاستدعاء.

تعدّ الأدوات الأساسية في التطبيق مع الأدوات الإضافية مكملة لبعضها لتسهيل عملية الاستخدام، والوصول إلى الفائدة المرجوّة من التطبيق.

1-أداة الطبوغرافيا (Topography) :

الطبوغرافيا، أو سمات سطح الأرض، أو علم التضاريس، هو تمثيل دقيق لسطح الأرض بعناصره الطبيعية، حيث يمكن من خلال أداة الطبوغرافيا تكوين خرائط طبوغرافية توضّح تضاريس سطح الأرض، وتبيّن ارتفاعات النقاط بالنسبة لبعضها بعضًا.

تحتوي النقاط التي يتم رصدها من خلال هذه الأداة على منسوب النقطة بالنسبة للجيوئيد العالمي، وليس لمنسوب سطح البحر؛ كون هذه النقاط يتم تحديد مناسيبها بوساطة الأقمار الصناعية التي تعتمد في حساباتها للارتفاع عن سطح الجيوئيد، لذلك لا يمكن مقارنتها مع منسوب سطح الأرض لوجود فرق في الارتفاع بينها.

ويمكن من خلال النقاط التي تم رصدها تكوين أسطح متصلة بما يعرف بنموذج الارتفاعات الرقمية، أو يمكن عمل خطوط كنتورية للمنطقة لتحديد أماكن الارتفاعات والانخفاضات لتسهيل فهم طبيعة الأرض، ولتحديد مجاري الأودية، أو السفوح، أو الأودية، أو السهول لعمل الدّراسات المطلوبة من قبل المستخدمين، ولكن ما يعيب هذه الأداة هو أنه لا يمكن إيجاد فروق دقيقة بين النقاط للأماكن ذات المساحات الصغيرة نوعًا ما ، حيث يمكن استعمال هذه الأداة في المساحات الواسعة التي يمكن للتطبيق فيها إيجاد فروقات واضحة بين ارتفاعات النقاط، والسبب في ذلك يعود إلى الدقة المنخفضة في الإشارة التي يستقبلها نظام (GPS) في الهاتف الذكي.

تتم برمجة أداة الطبوغرافيا (Topography)، كما في الشكل(47)، بنفس الطريقة التي تمت فيها برمجة أداة رصد النقاط، مع وجود اختلافات بسيطة داخل ملف النشاط (Activity)، والتي يتم تخزين منسوب النقطة فيها، بالإضافة إلى وجود كود برمجي، كما في الشكل (48)، يقوم على رصد النقاط بشكل تلقائي بفترة كنتورية مقدارها متر واحد.

بعد ذلك، يتم تخزين النقاط مباشرة إلى قاعدة البيانات دون الحاجة إلى إعطاء أمر للتخزين داخل الأزرار الموجودة في الواجهة التفاعلية الخاصة بأداة الطبوغرافيا.

```
radioSexGroup33.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener()

{
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        Button btn88 = (Button) findViewById(R.id.button2_t);
        RadioButton rb=(RadioButton)findViewById(checkedId);
        Toast.makeText( context: Topography_.this, rb.getText().toString(), Toast.LENGTH_LONG).show();
        if (rb.getText().toString().equals("Auto TOPO (2 second)")){
            btn88.setEnabled(false);
            handler.removeCallbacks(runnable);
            handler.postDelayed( runnable = new Runnable() {
                public void run() {
                    go_run();
                    handler.postDelayed(runnable, delay);
                }
            }, delay);
            refresh_point();
            //  Longitude and latitude

        }else if (rb.getText().toString().equals("Manual TOPO")){
            btn88.setEnabled(true);
            refresh_point();
            handler.removeCallbacks(runnable);

        }

    }
});
```

الشكل(47): الكود البرمجي الخاص بأداة الطبوغرافيا داخل ملف النشاط (Activity) .

```
public boolean Min_dist_check(String[] E, String[] N,LatLng g) {
    wgs84_to_cassini wtocas = new wgs84_to_cassini();
    LatLng now = g;
    boolean boo = false;
    String[] points_E = E;
    String[] points_N = N;
    int n = points_E.length;
    wtocas.wgs84_to_cassini_p(now.latitude,now.longitude);
    Double N_con = Double.parseDouble((String.format("%.3f",wtocas.No)));
    Double E_con = Double.parseDouble((String.format("%.3f",wtocas.Ea)));
    Double E_Mc = E_con ;//now
    Double N_Mc = N_con;//now
    Double su[] = new Double[points_E.length];
    for (int x = 0; x < points_E.length; x++) {
        Double ttE = Double.parseDouble(points_E[x]);
        Double ttN = Double.parseDouble(points_N[x]);
        Double clc = Math.abs(Math.pow(Math.pow((E_Mc - ttE), 2) + Math.pow((N_Mc - ttN), 2), 0.5));
        su[x] = clc;
    }
    Double first = Double.MAX_VALUE;
    for (int i = 0; i < su.length; i++) {
        if (su[i] < first) {
            first = su[i];
        }
    }if (first >= 1.0) {
            boo = true;
    }else {
            boo = false;
        }
    return boo;
```

الشكل(48): الكود البرمجي الخاص بأداة الطبوغرافيا (إنشاء فترة كنتورية بين النقاط المرصودة) داخل ملف النشاط (Activity) .

حيث يتم فحص النقاط التي تم رصدها من خلال الكود البرمجي، كما في الشكل (48)، من خلال فحص المسافة بين النقطة المرصودة وآخر نقطة تم رصدها، وفي حال انطباق شرط المسافة (واحد متر) بين النقطتين، يتم تخزين النقطة الجديدة في قاعدة البيانات.

2-أداة طبقات (KMZ):

تعد أداة طبقات (KMZ)، كما تم ذكرها سلفًا، بأنها ملف يحتوي على علامات موضعية تتضمن اسمًا مخصصًا، وإحداثيات جغرافية للمواقع، وبيانات للنموذج الثلاثي الأبعاد، حيث تستخدم هذه الملفّات لعرض الإحداثيات على خرائط (Google) التي يمكن عرض النقاط والخطوط والمضلعات من خلالها على هذه الخرائط .

يمكن من خلال هذه الأداة إضافة طبقات(KMZ) التي يتم إنشاؤها على برامج، مثل، (Google Earth) ، أو (Arc Map) ، والتي يتم تحديد على سبيل المثال منطقة الدّراسة المراد العمل عليها ميدانيًّا بمضلع، وإضافة هذا المضلع إلى التطبيق لحصر مكان العمل الميداني الذي يظهر أثناء استخدام الخرائط داخل التطبيق، كما يمكن إضافة طبقات (Shape file) من نقاط أو خطوط إلى الخرائط، أو حتى ملفّات من صيغة (AutoCAD. DWG)، ولكن يشترط تحويلها قبل الإضافة إلى ملف (KMZ) ليستطيع التطبيق إضافتها.

تعدّ هذه الأداة من الأدوات التي تمكّن الجغرافي من الربط بين الملفّات التي يمكن العمل عليها على برامج أخرى والتطبيق (Green Land) لتسهيل عملية العرض والتفاعل داخل العمل الميداني، وتمكين الجغرافيين من مقارنة هذه البيانات مع الواقع لفهم هذه الظواهر وتفسيرها، أو حسب الحاجة المرجوّة من ذلك.

وكمثال بسيط على هذه الأداة، يمكن للمستخدم رسم مضلعات داخل برنامج (Google Earth) على الأبنية، مثل المدارس داخل منطقة معينة، ثم حفظ هذه الطبقة، ومن ثم إضافتها إلى التطبيق، بعد ذلك يمكن للمستخدم مشاهدة هذه المضلعات على التطبيق داخل الخرائط المدعومة، والوصول إليها من خلال تقنية (GPS)، فمن خلال ذلك يتم التسهيل على المستخدم من الربط بين البيانات والواقع للوصول إلى تحليل سليم لهذه الظواهر.

تتم برمجة أداة طبقات (KMZ) من خلال كود برمجي، كما في الشكل(49)، حيث يتم من خلاله استقبال ملفّات (KMZ)، وإضافتها إلى خرائط (Google) والتعامل معها، بالإضافة إلى ربط ملفّات (KMZ) مع قاعدة البيانات الخاصة بأداة

(KMZ) التي تقوم بحفظ مسارات مكان التخزين على قرص التخزين الخاص بالجهاز المحمول.

```
public void kmz_go_(View view) {
    mMap.clear();
    go_points();
    Cursor cursor1=  helper.getReadableDatabase().
            query( table: "kmz_data",new String[]{"_id","Br"}, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: "_id", limit: null);
    int i1 = 1;
    cursor1.moveToFirst();
    while (!cursor1.isAfterLast()){
        int id =cursor1.getInt( columnIndex: 0);
        String Br =cursor1.getString( columnIndex: 1);

        KmlLayer kmlLayer = createLayerFromKmz(Br);
        kmlLayer.addLayerToMap();
        cursor1.moveToNext();
        i1++;
    }
}
```

الشكل(49): الكود البرمجي الخاص بأداة طبقات (KMZ) داخل الزر في ملف النشاط (Activity) .

في بداية الأمر، يتم استدعاء روابط الملفّات الخاصة بطبقات (KMZ) المخزّنة في قاعدة البيانات لتحديد أماكن تخزينها في قرص التخزين على الهاتف المحمول، ثم تتم إضافتها إلى الخريطة من خلال الأمر (addLayerToMap) .

3-أداة تحويل الإحداثيات (Coordinates Conversion):

يكمن عمل هذه الأداة في عملية تحويل الإحداثيات من النظام الجغرافي العالمي للمرجع الجغرافي (WGS1984) إلى النظام المحلي، مثل، النظام التربيعي الفلسطيني، أو النظام الأردني، أو العكس، وإظهارها على الخريطة المدعومة على الخريطة، ويعود السبب في وجود هذه الأداة هو عدم توفر إحداثيات محلية في أجهزة الرصد المحمولة (Handheld GPS) في بعض الدّراسات الميدانية التي لا يمكن فيها إجراء العمليات الإحصائية والحسابية إلا من خلال إحداثيات محلية مترية، وفي بعض الأحيان تكون هناك حاجة لتحويل الإحداثيات المحلية إلى إحداثيات جغرافية عالميه يمكن عرضها على برامج أخرى، مثل (Google Earth).

ويقوم مبدأ تحويل الإحداثيات على تحويل الإحداثية الجغرافية من المرجع الجغرافي (WGS1984) إلى النظام الكارتيزي (الفراغي) الخاص بنفس المرجع، ثم التحويل إلى النظام الكارتيزي الخاص بالمرجع الآخر، مثل المرجع الجغرافي (Clarke 1880) الخاص بنظام التربيع الفلسطيني، ثم يقوم بتحويله إلى النظام الجغرافي للمرجع الجغرافي نفسه (Clarke 1880)، ثم يتم استخدام المسقط الخاص بنظام الإحداثيات المحلي (Cassini) الذي يستعمل المسقط الأسطواني المستعرض، وبذلك يكون قد تم تحويل الإحداثيات إلى النظام المتري أو الإسقاطي.

عند إدخال الإحداثيات إلى التطبيق أو إلى الأداة سالفة الذكر، يتم تحديد إذا كانت الإحداثية المدخلة بنظام جغرافي أو متري بشكل تلقائي داخل التطبيق.

يتم استخدام ملف (Java Class) الخاص بتحويل الإحداثيات في كثير من ملفّات النشاط (Activity) الموجودة في التطبيق، ويعود السبب في ذلك إلى كون الإحداثيات التي يتم إجراء الحسابات عليها هي إحداثيات مترية أو مسقطة وليست جغرافية، ومن هنا تأتي الحاجة إلى تحويل الإحداثيات الجغرافية إلى إحداثيات مترية أو مسقطة، من خلال الكود البرمجي الموجود، كما في الشكل (50).

```
public void wgs84_to_cassini_p(Double lat_cassini, Double lon_cassini) {
    lu = lat_cassini;
    lo = lon_cassini;
    a101 = 31.7340969444444;
    x = 6378137 * Math.cos(lu * 0.0174532925) * Math.cos(lo *
            0.0174532925);
    y = 6378137 * Math.cos(lu * 0.0174532925) * Math.sin(lo *
            0.0174532925);
    z = (6378137 * 0.99330562) * Math.sin(lu * 0.0174532925);
    a1 = (1.00001278) * x;
    a2 = (1.00001278) * y;
    a3 = (1.00001278) * z;
    a4 = (1 * a1 + (-0.00005731) * a2 + (0.00002143) * a3);
    a5 = ((0.00005731) * a1 + 1 * a2 + (-0.00003879) * a3);
    a6 = ((-0.00002143) * a1 + (0.00003879) * a2 + 1 * a3);
    a7 = (a4 + (275.722));
    a8 = (a5 + (-94.7824));
    a9 = (a6 + (-340.894));
    a10 = (Math.atan((a9 / Math.pow((Math.pow((a7), 2) + Math.pow((a8),
            2)), (0.5))) / (1 - Math.pow((0.082483215), 2)))) / 0.01745329252;
    a11 = (Math.atan(a8 / a7)) / 0.01745329252;
    A = ((a11 * 0.0174532925) - (35.2120805555555 * 0.0174532925)) *
            Math.cos(a10 * 0.0174532925);
    T = Math.pow(Math.tan(a10 * 0.0174532925), 2);
    C = (0.006803481 * Math.pow(Math.cos(a10 * 0.0174532925), 2)) / (1
            - 0.006803481);
    V = 6378300.789 / Math.pow((1 - 0.006803481 * Math.pow(Math.sin(a10 * 0.0174532925), 2)), 0.5);
    M = 6378300.789 * (1 - 0.006803481) * 1.005135373 * (a10 *
            0.0174532925) + 6378300.789 * (1 - 0.006803481) * (-0.002573165) * Math.sin(2 * a10
            * 0.0174532925) + 6378300.789 * (1 - 0.006803481) * 0.00000274473 * Math.sin(4 *
                a10 * 0.0174532925) + 6378300.789 * (1 - 0.006803481) * -0.0000000036434 *
                Math.sin(6 * a10 * 0.0174532925);
        S = 6378300.789 * (1 - 0.006803481) * 1.005135373 * Math.sin(a10 *
                0.0174532925);
        M0 = 6378300.789 * (1 - 0.006803481) * 1.005135373 * (a101 *
                0.0174532925) + 6378300.789 * (1 - 0.006803481) * (-0.002573165) * Math.sin(2 *
                a101 * 0.0174532925) + 6378300.789 * (1 - 0.006803481) * 0.00000274473 * Math.sin(4
                * a101 * 0.0174532925) + 6378300.789 * (1 - 0.006803481) * -0.0000000036434 *
                Math.sin(6 * a101 * 0.0174532925);
        Ea = 170251.555 + V * (A - T * Math.pow(A, 3) / 6 - (8 - T + 8 * C)
                * T * Math.pow(A, 5) / 120);
        No = 1126867.909 + M - M0 + V * Math.tan(a10 * 0.0174532925) *
                (Math.pow(A, 2) / 2 + (5 - T + 6 * C) * Math.pow(A, 4) / 24);

    }

}
```

الشكل(50): الكود البرمجي الخاص بأداة تحويل الإحداثيات داخل ملف (Java Class).

4-أداة تصدير البيانات (Export) :

تقوم هذه الأداة على تصدير البيانات الموجود داخل التطبيق على شكل ملف (Notepad) بامتداد (txt)، وتعدّ هذه الملفّات من الملفّات التي يمكن العمل عليها داخل برامج أخرى، مثل (Arc Map)، أو (Google Earth)، أو أي برامج أخرى تدعم ملفّات محرّر النصوص.

وينظر إلى هذه الأداة على أنها من الأدوات التي تعدّ انفتاحًا على البرامج الأخرى؛ بسبب إمكانية العمل على الملفّات التي يتم تصديرها منها، كما تلزم الحاجة للعمل على برامج أخرى لمعالجة النقاط، أو دراستها على برامج متخصصة.

يقوم التطبيق على تصدير البيانات أو الإحداثيات المطلوبة، بالإضافة إلى بقية البيانات المرتبطة بهذه الإحداثية بشكل يمكن للبرامج الأخرى أن تفهمه، من خلال ترتيبها في صفوف فوق بعضها بعضًا لتجنّب حصول أي أخطاء أثناء عملها على البرامج الأخرى، حيث إن عملية تنسيق الإحداثيات، بالإضافة إلى البيانات المرتبطة بها، هي طريقة موحدة بين جميع ملفّات محرّر النصوص التي تصدرها بقية البرامج الأخرى.

تتم برمجة أداة تصدير البيانات (Export) لاستخرج النقاط التي تم رصدها، إن كانت نقاط لظواهر أو نقاط طبوغرافية بوساطة الكود البرمجي، حيث يتم إنشاء نشاط (Activity)، كما في الشكل (51)، والذي يحتوي على بعض الأوامر، مثل: استيراد النقاط من قاعدة البيانات، وتحديد نوع الإحداثيات المراد تصديرها، كما في الشكل (52)، بالإضافة إلى كود زر المشاركة، كما في الشكل (53) ،من خلال إحدى وسائل المشاركة، مثل البريد الإلكتروني، حيث يقوم الكود البرمجي، كما في الشكل (54)، على تحويل النقاط المخزّنة في قاعدة البيانات إلى نصوص يتم حفظها في محرّر نصوص (Note Pad)، كما في الشكل (55)، ليتم استخدامه في برامج أخرى.

```
        if (rb.getText().toString().equals("Observed Points")){
            btn88.setEnabled(false);
            radioGroup1.check(ggg.getId());
            tv_sl.setText("Observed Points");
            view_export.setText("");
            String s = "id, Latitude, Longitude, Name, Notes\n";
            view_export.append(s);
            String[] latlon_now =ds.Get_data_latlon();
            for (int x1 = 0; x1 < latlon_now.length; x1++) {
                String r = latlon_now[x1] + "\n";
                view_export.append(r);
            }
        }else if (rb.getText().toString().equals("Topographic Points")){
            tv_sl.setText("Topographic Points");
            radioGroup1.check(ggg.getId());
            btn88.setEnabled(false);
            view_export.setText("");
            String s = "id, Latitude, Longitude, Elevation\n";
            view_export.append(s);
            String[] latlon_now =ds1.Get_data_latlon();
            for (int x1 = 0; x1 < latlon_now.length; x1++) {
                String r = latlon_now[x1] + "\n";
                view_export.append(r);
            }
        }

    }
});
```

الشكل(51): الكود البرمجي الخاص بأداة تصدير النقاط في ملف النشاط (Activity).

```
radioSexGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener()

{
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        RadioButton rb=(RadioButton)findViewById(checkedId);
    Button btn_sto = (Button)findViewById(R.id.btn1_update);

        Toast.makeText( context: View_data_Base.this, rb.getText().toString(), Toast.LENGTH_LONG).show();
        if (rb.getText().toString().equals("WGS1984")){

          refreshListView();
            btn_sto.setEnabled(true);

        }else if (rb.getText().toString().equals("Cassini")){
           refreshListView_cassini();
            btn_sto.setEnabled(false);

        }else if (rb.getText().toString().equals("JTM")){
            refreshListView_jtm();
            btn_sto.setEnabled(false);

        }

    }
});
```

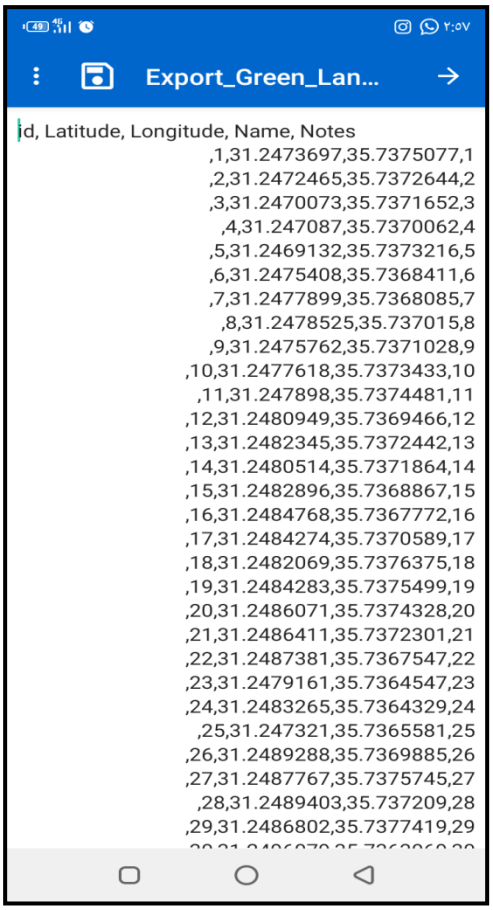
الشكل(52): الكود البرمجي الخاص بأداة تصدير النقاط (تعديل نوع الإحداثيات) في ملف النشاط (Activity).

```
private void saveTextAsFile (String content,String Ex,String Na,String ro){

    File root = new File(Environment.getExternalStorageDirectory(), ro);
    if (!root.exists()) {
        root.mkdirs();
    }

    try {
        File gpxfile = new File(root,  child: Na+Ex);
        FileOutputStream fos = new FileOutputStream(gpxfile);
        fos.write(content.getBytes());
        fos.close();
        Toast toast = Toast.makeText( context: this,  text: "Done ..", Toast.LENGTH_SHORT);
        toast.getView().setBackgroundColor(Color.GREEN);
        toast.show();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
        Toast.makeText( context: this,  text: "Error // app ", Toast.LENGTH_SHORT).show();
    } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText( context: this,  text: "Error saving", Toast.LENGTH_SHORT).show();

    }

}
```

الشكل(53): الكود البرمجي الخاص بأداة تصدير النقاط (داخل زر التصدير) في ملف النشاط (Activity).

```
public void share(View view) {

    StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();
    StrictMode.setVmPolicy(builder.build());
    File pdfFile = new File(Environment.getExternalStorageDirectory(),  child: "Green Land/Export/Export_Green_Land.txt");
    Uri uri = Uri.fromFile(pdfFile);
    Intent shareIntent = new Intent();
    shareIntent.setAction(Intent.ACTION_SEND);
    shareIntent.setType("application/pdf");
    shareIntent.putExtra(Intent.EXTRA_STREAM, uri);
    startActivity(Intent.createChooser(shareIntent,  title: "Share via"));

}
```

الشكل(54): الكود البرمجي الخاص بأداة تصدير النقاط (داخل زر المشاركة) في ملف النشاط (Activity).



الشكل (55) : ملف (Note Pad) الذي تم تصديرة.

5-أداة استيراد البيانات (Import):

تقوم هذه الأداة على استيراد ملفّات (Notepad) بامتداد (txt) التي تحوي البيانات داخل التطبيق ليتم إضافتها إلى قاعدة البيانات وحفظها فيها، حيث يجب أن تتكوّن هذه البيانات من إحداثيات ومعلومات وصفية عنها، ومرتبة بشكل يمكن للتطبيق فهمها وإضافتها بسلاسة دون حدوث أخطاء؛ ليتم عرضها على الخرائط، مع إمكانية إجراء العمليات الإحصائية عليها.

يكون مصدر هذه البيانات في العادة من برامج أخرى يتم تصديرها منها، مثل: (Google Earth)، أو(Arc Map) ، أو حتى من أجهزة قياس أخرى أكثر دقة في القياس، مثل أجهزة (RTK- Real time kinematic) التي تعمل بقياسات دقيقة للرفع الميداني للبيانات، ولكن يشترط هنا لإضافة هذه البيانات أن تكون بنظام إحداثيات جغرافي وليس محلي، مع العلم أن التطبيق يقوم بتحويل هذه الإحداثيات

الجغرافية إلى إحداثيات محلية ليتمكّن من إجراء العمليات الحسابية عليها داخل التطبيق.

تتم برمجة أداة استيراد البيانات (Import) من خلال أكواد برمجية داخل ملف نشاط (Activity) التي تقوم بدورها بالتعامل مع ملفّات محرّر النصوص (Note Pad)، حيث تقوم بقراءة النصوص داخل ملف محرّر النصوص، وإعادة صياغتها داخل التطبيق من أجل تخزينها داخل قاعدة البيانات، بالإضافة إلى عرضها على الخرائط.

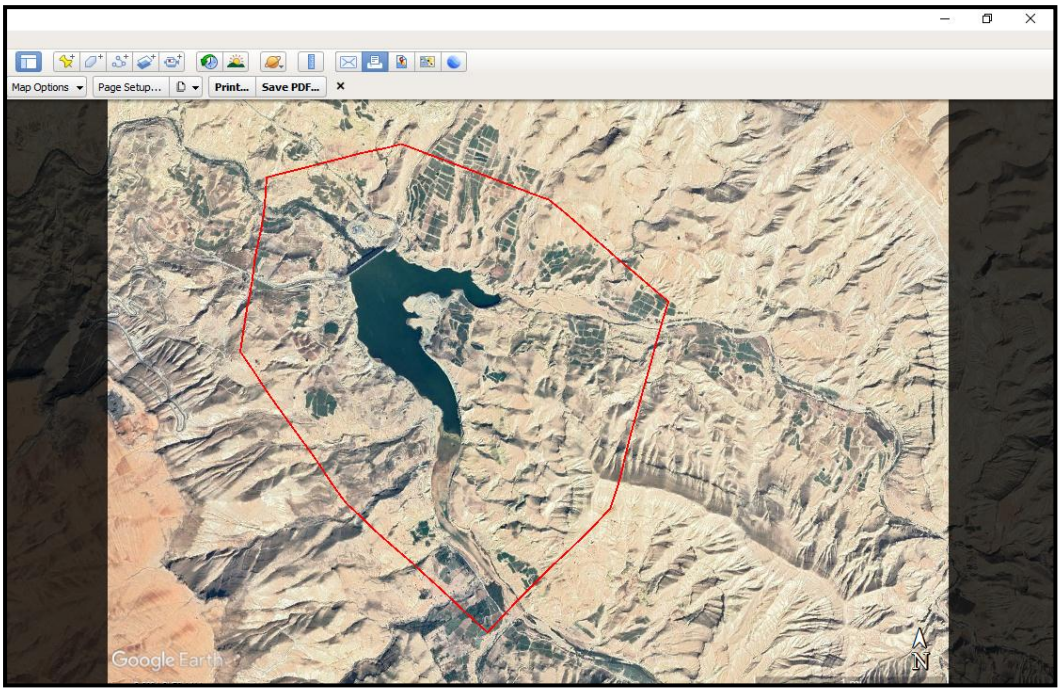
يتم في البداية تحميل ملف محرّر النصوص (Note Pad) المطلوب، من خلال الكود البرمجي المرتبط بالزر المخصص للاستيراد (Import)، كما في الشكل (56)، بعد ذلك يتم تخزين النقاط التي تم استيرادها من ملف محرّر النصوص إلى قواعد البيانات.

```
public void import_(View view) {
    Button d = (Button)findViewById(R.id.Import1_im);
    d.setEnabled(false);
    loadFileList();
    showDialog(DIALOG_LOAD_FILE);
    Log.d(TAG, path.getAbsolutePath());
    d.setEnabled(true);

    }
```

الشكل (56) : الكود البرمجي الخاص بأداة استيراد النقاط (داخل الزر) في ملف (Activity).

12.3 الخطوات العملية على التطبيق في العمل الميداني :

1. في البداية، يقوم المستخدم بتحديد منطقة الدّراسة المراد العمل عليها ميدانيًّا، من خلال عمل مضلع أو مسار على برنامج (Google Earth)، كما في الشكل (50)، وبعد القيام بذلك يتم حفظ ملف المضلع أو المسار الموجود بامتداد (KMZ)، ومن ثم نقله إلى التطبيق والقيام بعملية الاستيراد للملف من خلال أداة إضافة طبقات (KMZ) في التطبيق .



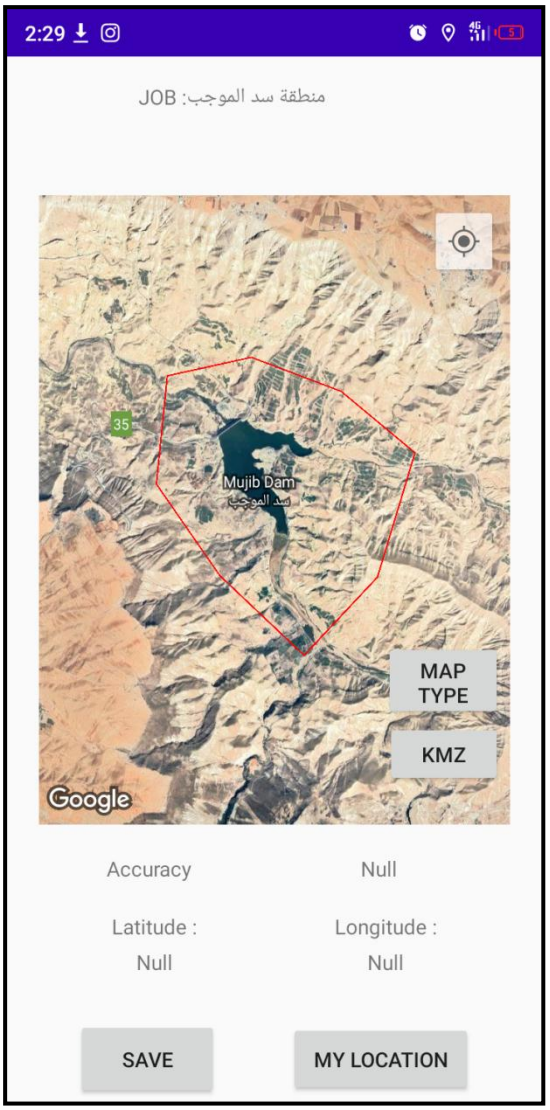
الشكل (57): شكل المسار لمنطقة الدّراسة على برنامج (Google Earth).

يعدّ الهدف من هذه العملية هو تحديد موقع الدّراسة في الميدان لضمان العمل داخل هذه المنطقة، وتجنّب العمل خارج هذا المضلع الذي تم تحديده مسبقًا على برنامج (Google Earth)؛ كون العمل خارج منطقة الدّراسة المراد العمل عليها يسبّب جهدًا بدنيًّا أكبر، وضياعًا للوقت، وربما الحصول على بيانات غير مطلوبة من خارج منطقة الدّراسة قد تؤثر في النتائج المرجوّة من الدّراسة.

ويجب التنويه، هنا، قبل عملية إضافة ملفّات (KMZ) الخاصة بمنطقة الدّراسة، أنه يجب علينا تسمية المشروع حسب اسم الدّراسة المطلوبة في بداية عمل التطبيق، كما في الشكل (58)، حيث إن عملية تسمية المشروع تبقي جميع عمليات الدّراسة، مثل: الرفع الميداني، وعمليات المعالجة داخل التطبيق مرتبطة باسم المشروع الذي تمت تسميته؛ لتجنّب تداخل البيانات أثناء القيام بعمل أكثر من دراسة منفصلة، حيث تظهر في بداية التطبيق لائحة يظهر فيها مكان لتسمية اسم المشروع، وقائمة في الأسفل بأسماء المشاريع السابقة التي تم العمل عليها في هذا التطبيق، وتكون النقاط فيها مرتبطة باسم المشروع لكل دراسة مفصولة عن الأخرى.

الشكل (58): الواجهة التفاعلية لإدارة المشاريع في تطبيق (Green Land).

2. في المرحلة الثانية (وهي بداية العمل الميداني) يقوم المستخدم بالوصول إلى موقع الدّراسة كما تم تحديده مسبقًا داخل المضلع أو المسار الذي يظهر على الخرائط المدعومة من قبل التطبيق، كما في الشكل (60) .

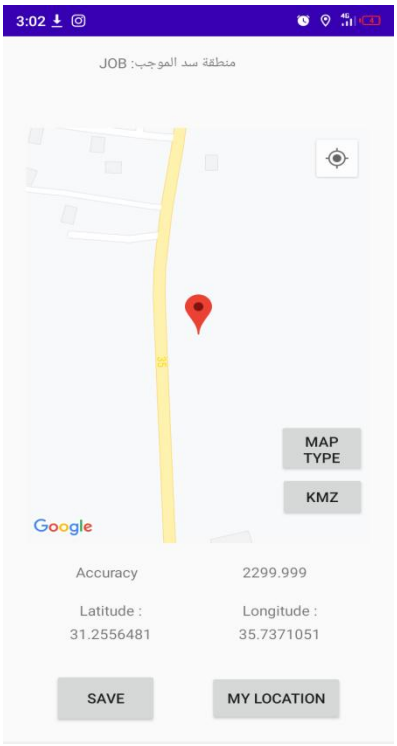


الشكل (59): الواجهة التفاعلية (Layout) لرصد النقاط في تطبيق (Green Land)، والتي يظهر فيها مسار منطقة الدّراسة.

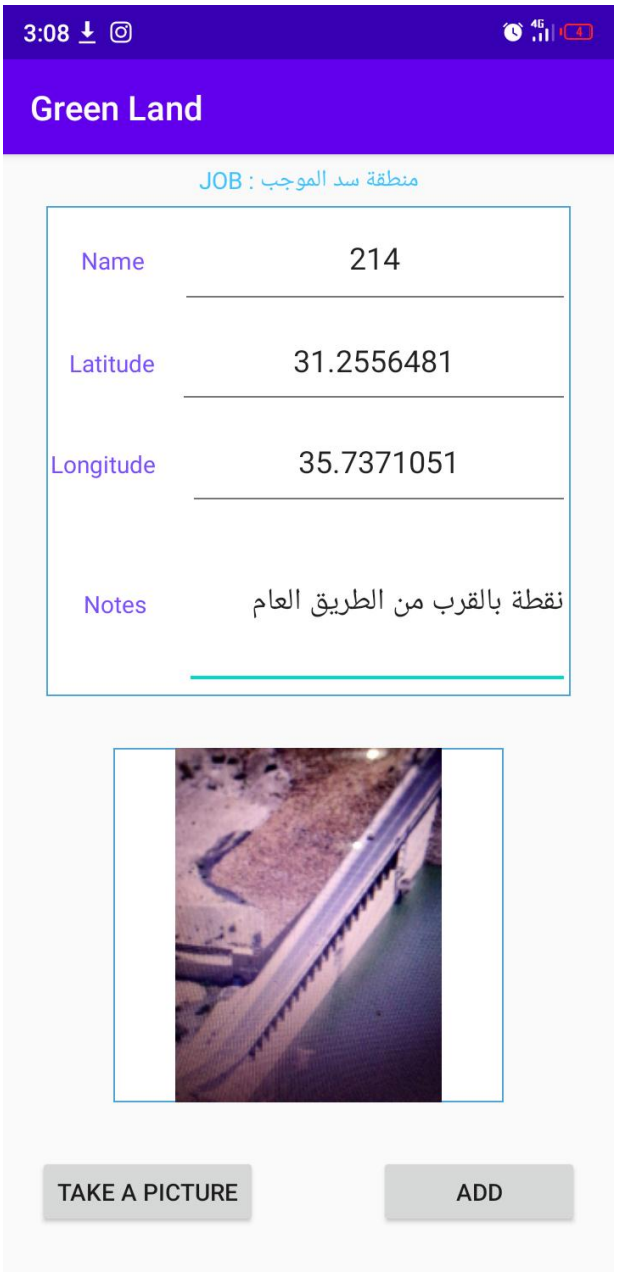
3. يتم تحديد نوع الظواهر المراد رصدها على الواقع من خلال الدّراسة المسبقة لنوع الدّراسة والهدف منها، حيث يتم الوقوف عند الظاهرة المطلوبة، ومن ثم يتم رصدها بوساطة أداة (Add points) الموجودة في التطبيق؛ أي إضافة نقطة أو ظاهرة على الخريطة وقاعدة البيانات حسب الموقع الذي يتم الوقوف فيه، من خلال النقر على زر(My Location)، كما في الشكل (60)، والذي يقوم على تحديد موقعك الجغرافي بوساطة تقنية (GPS)، فتظهر إحداثيات الموقع المطلوب، بالإضافة إلى دقة رصد الموقع التي تعتمد على وضع الأقمار الصناعية وعدد الأقمار الظاهرة له، حيث يجب أن تكون دقة رصد الموقع لاتزيد عن أربعة أمتار للحصول على نتائج مقبولة.

بعد ذلك، يتم النقر على زر (Save)، كما في الشكل (60)، لحفظ النقطة المطلوبة أو موقع الظاهرة، ومن ثم تظهر لائحة جديدة، كما في الشكل(61)، تظهر فيها تفاصيل النقطة المراد حفظها، حيث يتم من خلالها إدخال بقية المعلومات الوصفية للظاهرة، مثل: اسم الظاهرة، والملاحظات المراد إضافة المعلومات إليها، وإمكانية التقاط صورة لهذه الظاهرة، بعد ذلك يتم النقر على زر (Add) لإضافة النقطة إلى قاعدة البيانات وتخزينها.

يتم تكرار الأمر نفسه على بقية الظواهر داخل منطقة الدّراسة التي تم تحديدها مسبقًا.



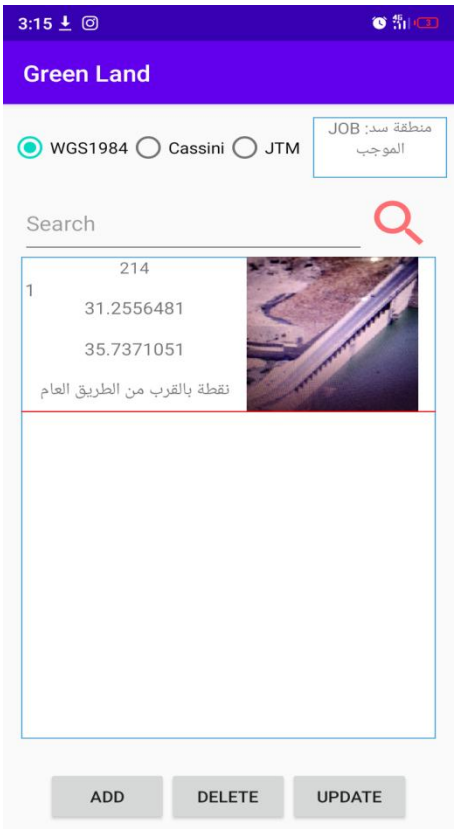
الشكل (60): الواجهة التفاعلية (Layout) لرصد النقاط في تطبيق (Green Land)، والتي يظهر فيها النقطة المرصودة .



الشكل (61): الواجهة التفاعلية (Layout) لحفظ النقاط في تطبيق (Green Land)، والتي تظهر فيها المعلومات الوصفية للنقطة.

4. يمكن مشاهدة النقاط التي تم رصدها، والتي يتم إضافتها، بشكل تلقائي على الخريطة المدعومة في التطبيق، حيث تساعد عملية إضافة النقاط على الخريطة في معرفة النقاط التي تم رصدها على الواقع، ولتجنّب رصدها مرة أخرى، بالإضافة إلى ذلك، يمكن التعديل على النقاط المرصودة من خلال الدخول على أداة (Data Base) التي توفر إمكانية البحث عن النقاط من خلال الاسم، بالإضافة إلى إمكانية حذفها، أو تعديلها، أو مشاهدة تفاصيل النقطة أو الظاهرة، مثل: الملاحظات، أو إحداثيات النقطة، أو الصورة التي تم التقاطها لها، كما يمكن

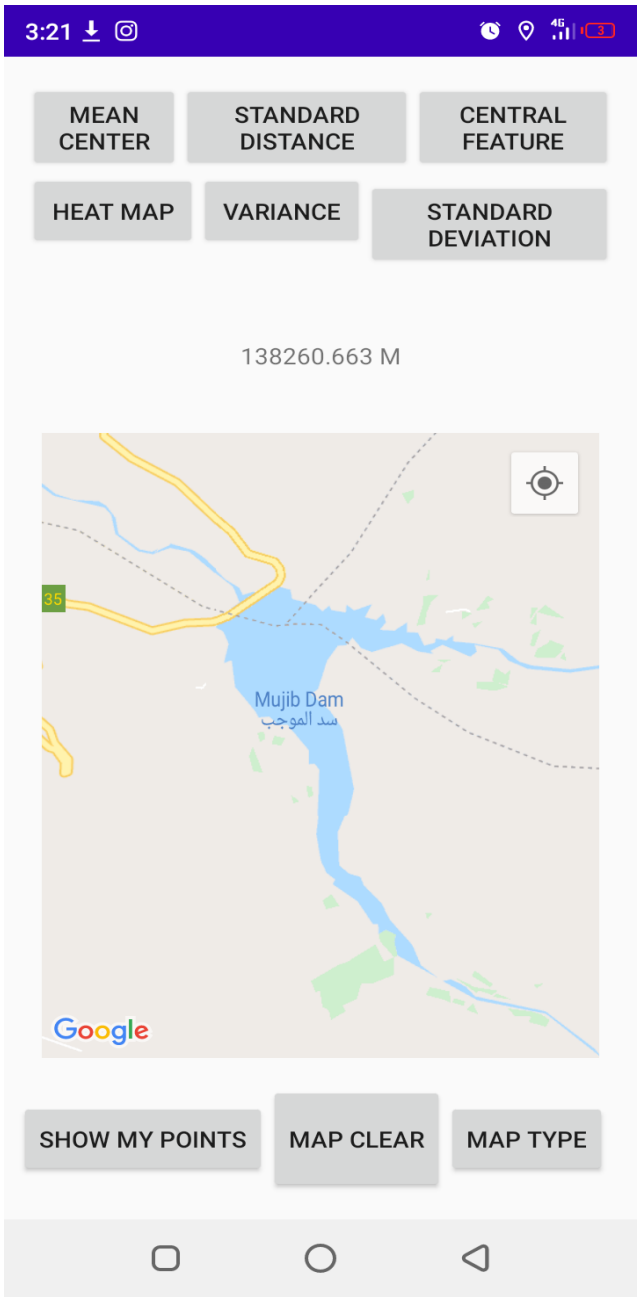
تغير نظام الإحداثيات لهذه النقاط أو الظواهر المرصودة، كما في الشكل (52)، حيث تم توفير ثلاثة أنظمة إحداثيات لتسهيل التعامل معها، في حال اتّباع نظام إحداثيات معين للدّراسة، مثل: نظام التربيع الفلسطيني، أو النظام الأردني، أو النظام الجغرافي العالمي.



الشكل (62): الواجهة التفاعلية (Layout) لقاعدة البيانات في تطبيق (Green Land)، والتي تظهر فيها البيانات.

5. بعد أن تم رصد النقاط المطلوبة داخل منطقة الدّراسة، يمكن لنا إجراء العمليات الإحصائية مباشرة من خلال أداة (Statistics)؛ إذ تحتوي هذه الأداة على مجموعة من الأزرار، كما في الشكل (63)، والتي تمثل الأوامر الإحصائية أو العمليات الحسابية التي تتم على النقاط أو الظواهر التي تم رصدها من الواقع. ففي البداية يتم النقر على زر(Show My Points) من أجل إظهار النقاط التي تم رصدها على الخرائط المدعومة من التطبيق، بالإضافة إلى وجود زر (Map Clear) من أجل إخفاء هذه النقاط، كما يمكن تعديل نوع الخريطة المعروضة من خلال النقر على زر (Map Type) التي تظهر فيها قائمة مكونة من أنواع الخرائط، مثل، خريطة دون معالم، أو خريطة تظهر الطرق أسماء المناطق، أو خريطة تحوي صورة فضائية للواقع الحقيقي. وبالرجوع إلى الأزرار الإحصائية،

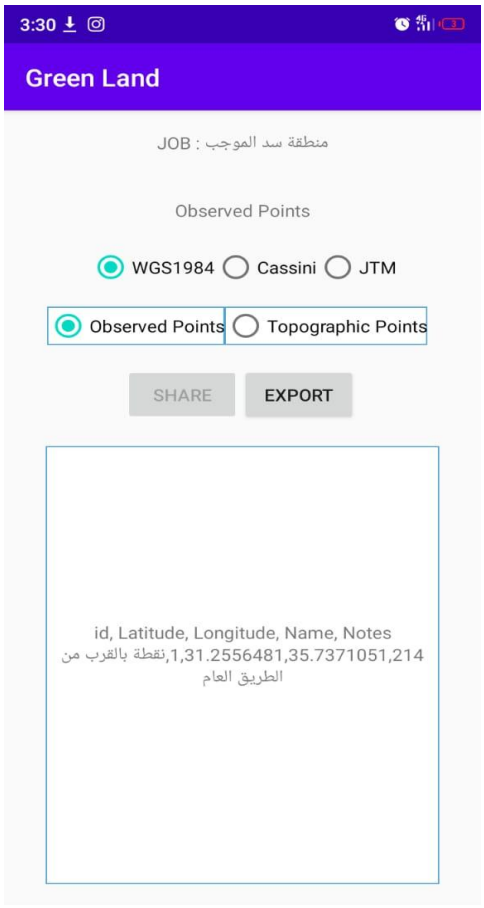
يمكن من خلالها إجراء العمليات الحسابية على النقاط، مثل: إيجاد الظاهرة المركزية (Central Feature)، أو المركز المتوسط (Mean Center)، أو المسافة المعيارية(Standard Distance)، أو الانحراف المعياري (Standard Deviation)، أو التباين (Variance)، أو الخريطة الحرارية (Heat map)، حيث إن جميع هذه الأوامر يتم العمل عليها للوصول إلى نتائج مباشرة دون الحاجة إلى استخدام برامج متخصصة، فعند استخدام أي أمر من هذه الأوامر الإحصائية، يتم عرض النتائج على الخريطة مباشرة، مثل: إظهار نقطة مميزة بين هذه النقاط، أو إظهار ألوان مميّزة تبيّن الكثافة، أو الانتشار لهذه النقاط، بالإضافة إلى وجود حيّز تظهر فيه النتائج مكتوبة أسفل الأزرار الإحصائية.



الشكل (63): الواجهة التفاعلية (Layout) للعمليات الإحصائية في تطبيق (Green Land)، والتي تظهر فيها الأزرار الخاصة بالعمليات الإحصائية وخرائط (Google).

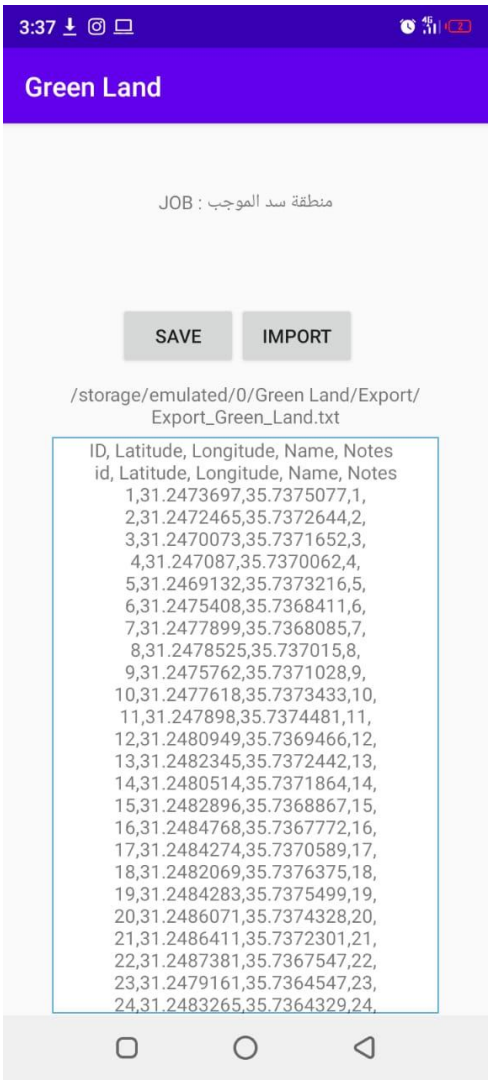
6. يمكن استخراج أو تصدير النقاط التي تم رصدها في العمل الميداني من خلال أداة (Export)، كما في الشكل (64)، التي تقوم على تحويل هذه النقاط إلى ملف محرّر النصوص بامتداد (txt)، من أجل إمكانية إجراء عمليات متقدمة أو معالجتها على برامج أخرى متخصصة، كما يمكن تحديد نظام الإحداثيات المراد تصدير الإحداثيات إليه، مثل: النظام الجغرافي، أو التربيع الفلسطيني، أو النظام

الأردني، بالإضافة إلى ذلك، يمكن بعد تصدير هذه النقاط مشاركتها من خلال أكثر من وسيلة (كما تم ذكرها سلفًا).



الشكل (64): الواجهة التفاعلية (Layout) لتصدير النقاط في تطبيق (Green Land)، والتي تظهر فيها النقاط المراد تصديرها.

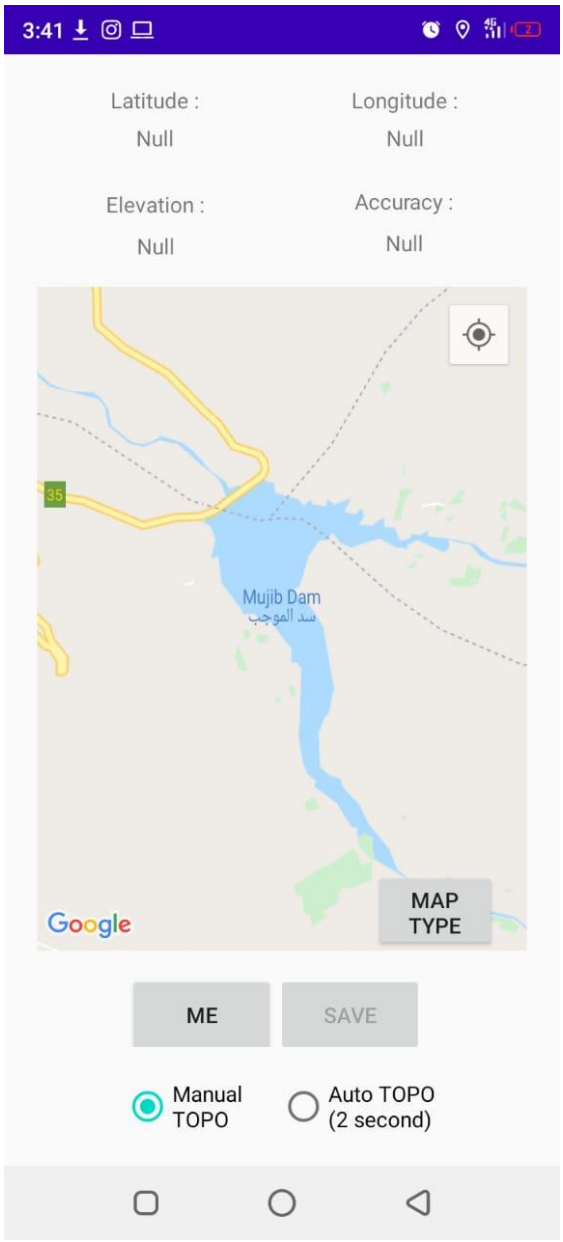
7. في حال الحصول على نقاط تم رصدها أو إيجادها من خلال برامج أخرى، ويراد إضافتها إلى التطبيق لإتمام العمل الميداني أو حتى إجراء عمليات إحصائية عليها، فيمكن من خلال أداة (Import) استيراد هذه النقاط إلى التطبيق ، كما في الشكل (65)، ولكن يشترط أن تكون بملف محرّر نصوص وبامتداد (txt)؛ لتجنّب حدوث أخطاء أثناء إضافتها، بالإضافة إلى أن تكون بنظام إحداثيات جغرافي، ومرتبة بالشكل المتعارف عليه.



الشكل (65): الواجهة التفاعلية (Layout) لاستيراد النقاط في تطبيق (Green Land)، والتي تظهر فيها النقاط المراد استيرادها.

8. إذا كانت الدّراسة المقصود منها عمل مخطّطات طبوغرافية، أو إنشاء نموذج ارتفاعات رقمية، فيمكن ذلك من خلال أداة (Topography) الطبوغرافيا ، كما في الشكل (65) ، كما يمكن أن توفر هذه الأداة أكثر من أسلوب للرصد، مثل أسلوب رصد النقاط بشكل يدوي، من خلال النقر على زر (Me)، ثم النقر على زر (Save) لحفظ النقطة التي تم رصدها، ويكون ذلك من خلال الوقوف على الموقع المطلوب رصد ارتفاعه، ثم الانتقال إلى موقع آخر، ثم استخدام نفس الأسلوب، ولا يشترط الترتيب فيها، كما يتم إثناء عملية الرصد إضافة النقاط إلى الخريطة المدعومة داخل التطبيق لمشاهدة هذه النقاط، ولعدم تكرار رصد الموقع، بالإضافة إلى التأكد من رصد المواقع المطلوبة كافة، كما توفر هذه الأداة أسلوبًا

آخر من الرصد وهو الرصد التلقائي للنقاط، كما في الشكل (67)، والتي تعمل على مبدأ رصد نقطة كل ثانيتين، بالإضافة إلى الرصد كل مسافة واحد متر، بمعنى آخر يجب تحقيق الشرطين، وهما: الوقت والمسافة لعدم تداخل مواقع النقاط، فعند تحرك المستخدم في الميدان مسافة واحد متر عن النقطة السابقة، بالإضافة إلى مهلة وقت ثانيتين لرصد النقطة التالية، في هذه الحالة يتم حفظ النقطة بشكل تلقائي وإضافتها إلى قاعدة البيانات، وفي حال عدم انطباق الشرطين لن يتم رصد النقطة، بالإضافة إلى ذلك، تظهر داخل الأداة نصوص مكتوب فيها الإحداثيات الجغرافية، ودقة رصد الموقع، وارتفاع النقطة المراد حفظها، كما يمكن تغيير نوع الخريطة المراد عرضها.



الشكل (66): الواجهة التفاعلية (Layout) لرصد النقاط الطبوغرافية في تطبيق (Green Land) .

الشكل (67): الواجهة التفاعلية (Layout) لرصد النقاط الطبوغرافية(خيارات الرصد الطبوغرافي) في تطبيق (Green Land) .

13.3 التطبيقات العملية والدّراسات الميدانية الممكنة على التطبيق:

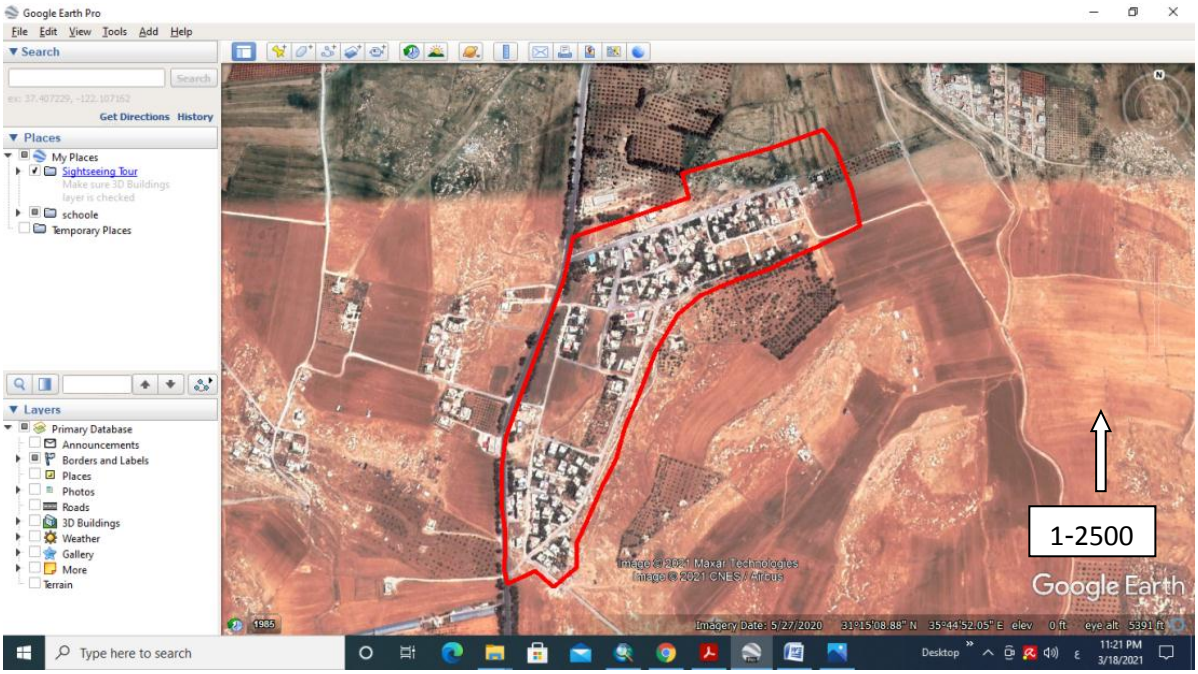
من أجل إتمام أفكار هذه الرسالة تم إجراء عمل ميداني على التطبيق من اجل الوصول الى الفائدة المرجوة منة ، وعرض النتائج وتحليلها وعرضها وإظهار مدى فوائد البرمجة في العمل الجغرافي الميداني للحصول على البيانات بأقل تكاليف وبوقت وجهد اقل.

في البداية يتم عرض بعض الدراسات التي يمكن القيام بها بواسطة التطبيق بشكل موجز ومنها :

احتاجت وزارة التربية والتعليم الى تحديد موقع لبناء مدرسة استنادا الى مواقع الأبنية التي تم إنشائها داخل حدود التنظيم لمنطقة الروضة التي تقع جنوب منطقة الربة في محافظة الكرك جنوب المملكة الأردنية الهاشمية ، حيث اشترطت وزراه التربية والتعليم اختيار قطعة ارض لا يوجد عليها أبنية لبناء المدرسة او اختيار بناء سكني جاهز قابل للاستملاك دون الحاجة الى إنشاء مدرسة في حال انطباق الشروط عليه.

وتم العمل على ذلك من خلال الخطوات التالية:

1- تم تحديد منطقة الدّراسة، وهي المناطق الموجودة داخل حدود التنظيم من منطقة الروضة، من خلال الرجوع إلى البلدية التي تتبع لها المنطقة، ثم طلب مخطّط لحدود التنظيم لتلك المنطقة، وبعد الحصول عليه استخدم برنامج (Google Earth) لرسم مسار أو مضلع منطقة الدّراسة المطلوبة الشكل(68)، حسب المخطّطات التنظيمية لها، كما في الشكل(69)؛ لتجنّب رصد الأبنية خارج حدود التنظيم، وبعد ذلك تم حفظ المسار أو المضلع داخل ملف(KMZ)، ثم نقله إلى التطبيق لتتم إضافته.



الشكل (68): منطقة الدّراسة حسب المخطط التنظيمي.

المصدر : عمل الباحث بالاعتماد على برنامج (Google Earth) لعام 2020

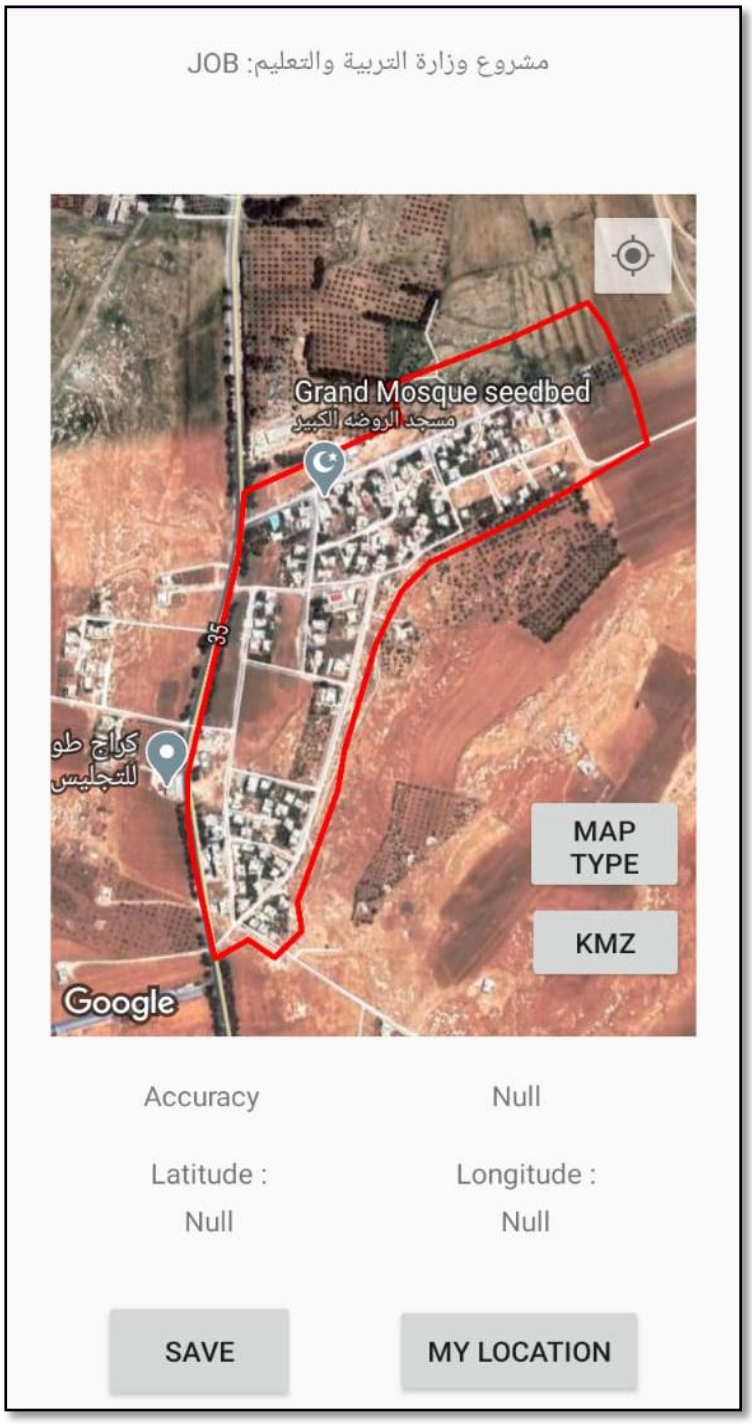
الشكل (69): مخطط تنظيمي لمنطقة الروضة للأراضي داخل حدود التنظيم

المصدر: مديرية الشؤون البلدية للواء القصر، 2020

2- بعد أن تمت إضافة ملف (KMZ) إلى التطبيق، كما في الشكل (70)، امكن معاينة الموقع على الخريطة للقيام بعملية رصد الأبنية، كما في الشكل(71)، المراد عمل الدّراسة عليها من خلال أداة (Add Point) التي نتمكّن من خلالها من تحديد أماكن هذه الأبنية كنقاط رصد، وحفظها داخل قاعدة البيانات.

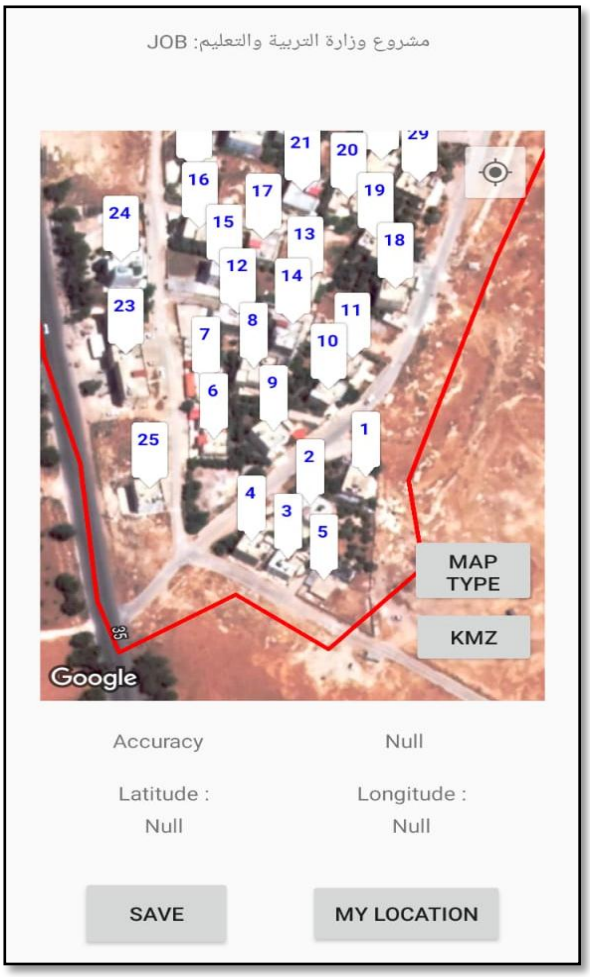
يجب أثناء العمل الميداني جعل الخريطة في الخلفيّة الخاصة بالواجهة التفاعلية من نوع (Hybrid)؛ للتمكّن من تحديد الأبنية المراد دراستها داخل حدود الدّراسة المطلوبة، واستثناء الأبنية الحكومية ودور العبادة من عملية الرصد؛ كونها لا تمثل جزءًا من الدّراسة، أو الهدف منها.

3- تم العمل على رصد الأبنية من خلال الوقوف على البناء أو بالقرب منه، ثم حفظ النقطة المطلوبة، وتجنّب الوقوف في أماكن تحجب إشارة الأقمار الصناعية التي يلتقطها الجهاز لتحديد إحداثية الموقع بوساطة تقنية (GPS) في الجهاز المحمول، بالإضافة إلى التأكد من دقة الموقع لتجنّب تسجيل إحداثيات لا تمثل مكان النقطة المراد عمل الدّراسة عليها، حيث يجب أن تكون الدقة (Accuracy) التي يلتقطها نظام (GPS) في حدود الأربعة متر .



الشكل(70): ملف (KMZ) كما يظهر على الخريطة لموقع الدّراسة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land) في الجهاز المحمول.



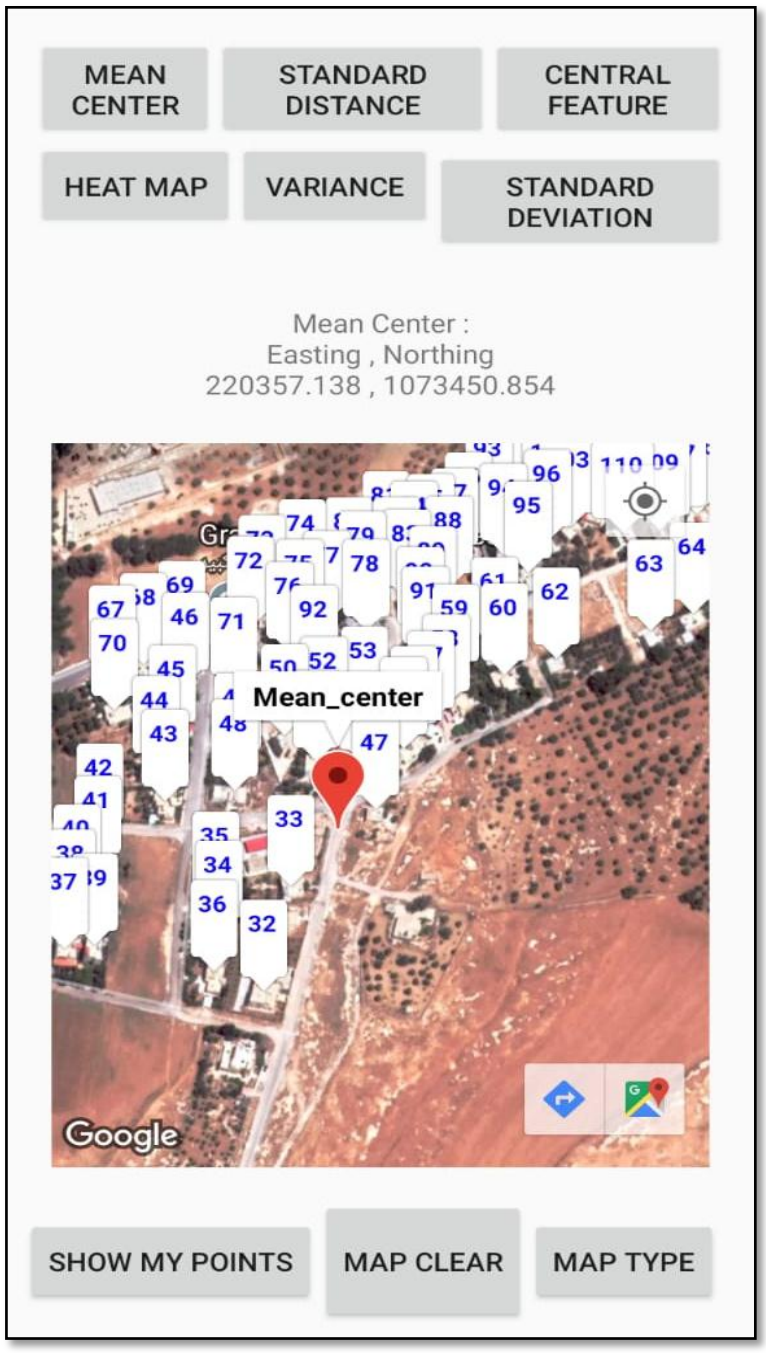
الشكل(71): نقاط الرصد كما تظهر على الخريطة للأبنية السكنية.

المصدر : عمل الباحث، بالاعتماد على تطبيق (Green Land) في الجهاز المحمول .

4-للوصول إلى نتائج سريعة ومباشرة، يتم الدخول إلى أداة (Statistics) لإجراء العمليات الإحصائية على النقاط التي تم رصدها ميدانيًّا، بالرجوع إلى الشروط التي طلبتها وزارة التربية والتعليم، وهي اختيار قطعة رض لا يوجد عليها أبنية لبناء المدرسة، أو اختيار بناء سكني جاهز قابل للاستملاك دون الحاجة إلى إنشاء مدرسة، في حال انطباق الشروط عليه.

من خلال الزر الخاص بأمر (Mean Center) الموجود داخل الأداة، يمكن الحصول على نقطة متوسطة لمجموعة النقاط التي تم رصدها، حيث يتم إظهار إحداثياتها وإضافتها إلى الخريطة، وهي النقطة التي تحمل الإحداثيات (1073450.854، 220357.138) الشكل(72)، علمًا أن هذه النقطة يتم

إظهارها بنظام إحداثيات متري (Palestine grid)، وتكون هذه النقطة مميّزة بلون أحمر لسهولة تمييزها عن بقية النقاط المرصودة.

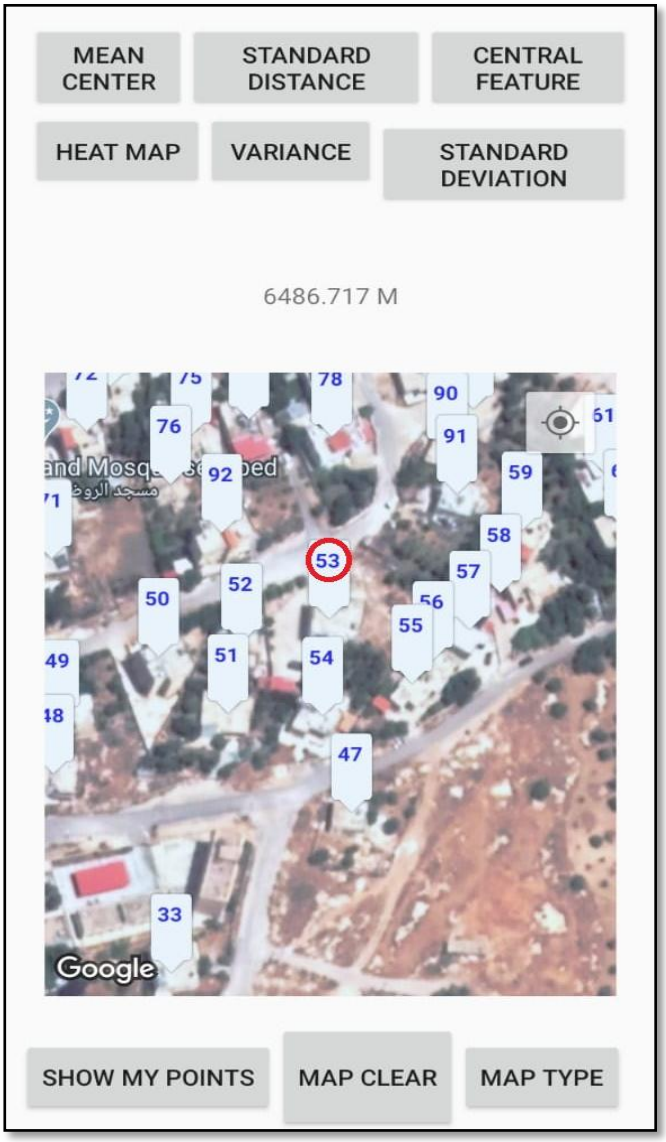


الشكل(72): المركز المتوسط كما تظهر على الخريطة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land).

أما في ما يخصّ اختيار بناء جاهز سكني قابل للاستملاك دون الحاجة إلى إنشاء مدرسة في حال انطباق الشروط عليه، فيتم ذلك عن طريق النقر على زر (Central Feature) التي يمكن من خلالها تحديد النقطة الأكثر تمركزًا بين النقاط

المرصودة، حيث تم تحديد البناء رقم (53)، كما في الشكل(72)، على أنه البناء الأكثر تمركّزًا بين الأبنية الأخرى، حسب التوزيع المكاني للنقاط.



الشكل(73): الظاهرة المركزية كما تظهر على الخريطة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land).

5-ولمشاهده مقدار كثافة تجمع الظواهر على الخريطة، كما في الشكل(74)، يمكن النقر على زر (Heat Map)، من خلال نفس الأداة، حيث يظهر الأختلاف في الألوان على الخريطة يبيّن كثافة أو تباعد النقاط عليها، على شكل خريطة حرارية .

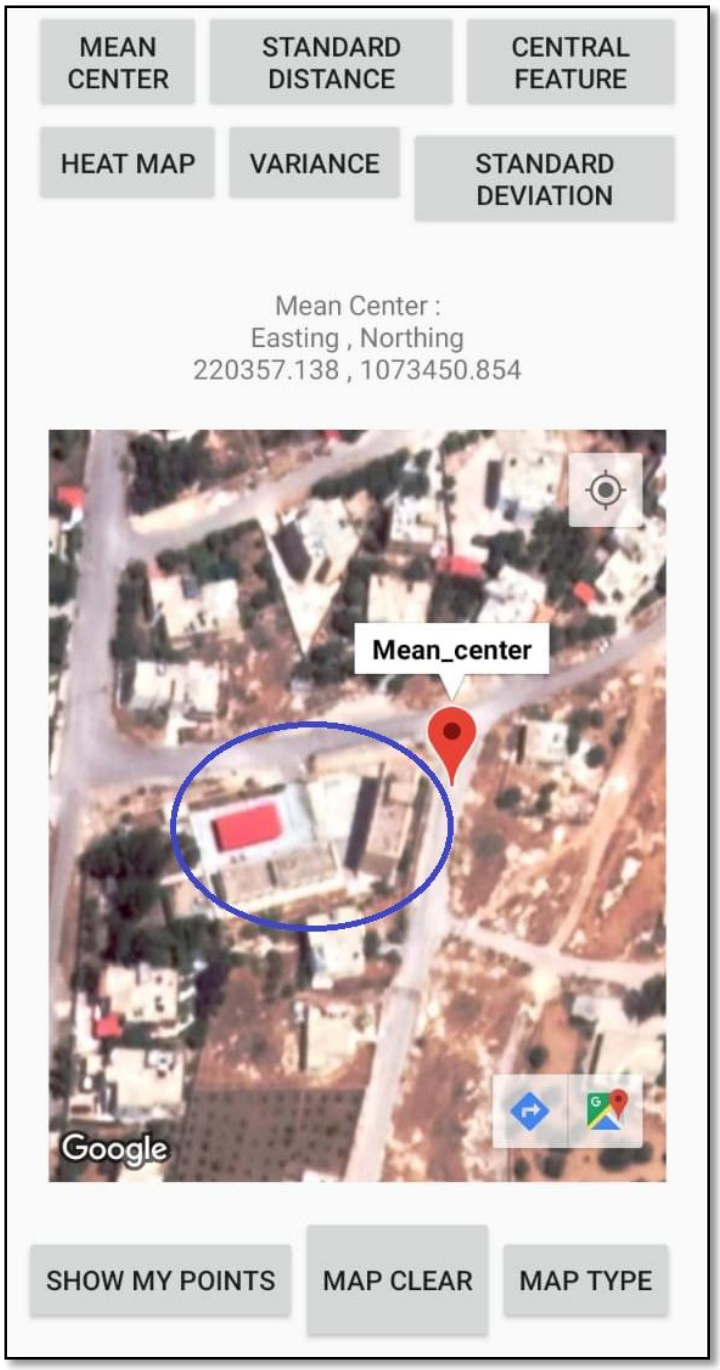
الشكل(74): الخريطة الحرارية كما تظهر على الخريطة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land).

إن النتائج التي تم الوصول إليها سابقًا، تمثل الحلول المطلوبة لاختيار موقع للمدرسة المراد إنشاؤها، حيث أظهرت النتائج ما يخصّ الحالة التي يشترط اختيار موقع غير مشغول من الأبنية لبناء مدرسة، فقد تم اختيار موقع على طريق معبّد، وليس على موقع أرض فارغ، ولكن يمكن اختيار قطعة أرض فارغة بالقرب من النقطة التي تم تحديها كمركز متوسط ، أما في الشرط الآخر وهو اختيار بناء جاهز سكني قابل لاستملاك دون الحاجة إلى إنشاء مدرسة في حال انطباق الشروط عليه، فقد تم

اختيار البناء رقم (53)؛ كونه البناء الأكثر تمركّزًا بالنسبة لبقية الأبنية، بالاستناد إلى العمليات الإحصائية السابقة الذكر.

عند مقارنة النتائج التي تم الوصول إليها مع الموقع الحقيقي للمدرسة التي تم إنشاؤها فعليًّا في منطقة الروضة، تبيّن أن البناء يقع في موقع مطابق لنتائج الدّراسة، الشكل(75)، وقد يكون اختيار موقع البناء الفعلي للمدرسة التي أنشأتها وزارة التربية والتعليم مبنيًا على دراسات واقعية للتوزيع السكاني لسكان القرية، مع العلم أن المدرسة موجودة قبل عملية التوسّع السكاني على أطراف القرية، حيث إن ما نسبته (20%) من الأبنية هي أبنية تم إنشاؤها بعد إنشاء المدرسة الموجود حاليًّا.

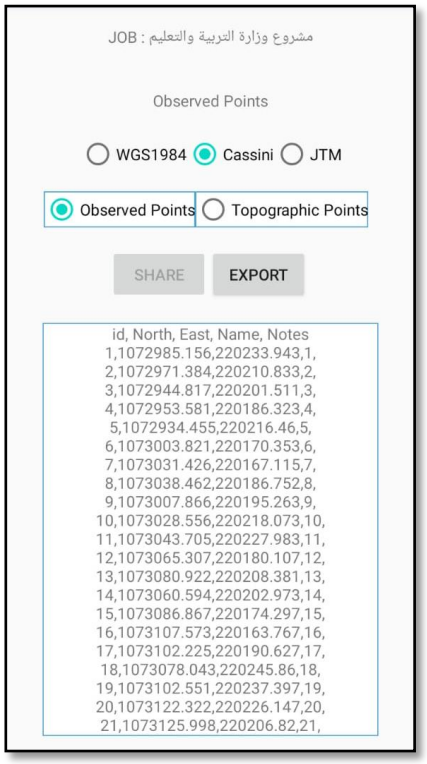


الشكل(75): موقع بناء المدرسة الفعلي بالنسبة للمركز المتوسط كما تظهر على الخريطة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land).

يمكن مقارنة النتائج التي تم الوصول إليها من خلال التطبيق(Green Land) مع النتائج التي يمكن استنتاجها من برامج متخصصة، مثل برنامج (Arc Map) المتخصص في نظم المعلومات الجغرافية، وتم القيام باستخراج النقاط بوساطة أداة (Export) على شكل ملف محرّر نصوص بامتداد (txt)، كما في الشكل(76)، ولكن ينبغي التنويه إلى أن النقاط التي يتم تصديرها يجب أن تكون بنظام إحداثيات محلي

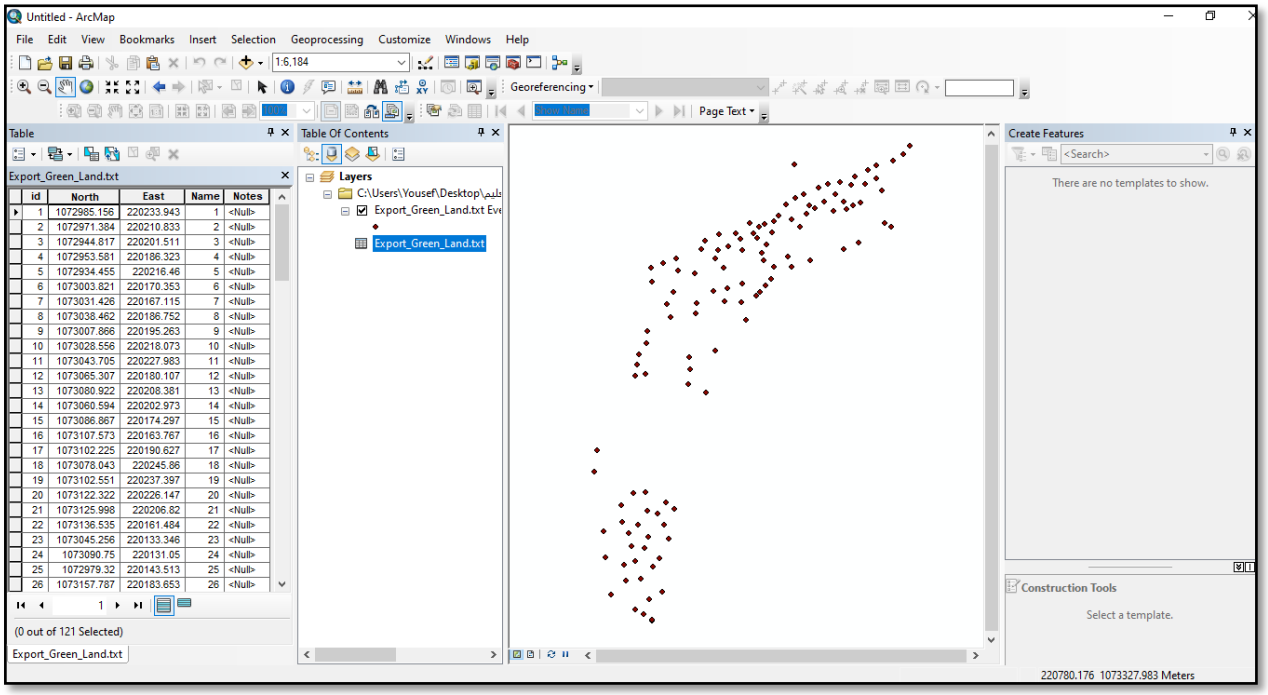
للتمكّن من إجراء الحسابات عليها داخل برنامج (Arc Map)، حيث توفر هذه الأداة مجموعة من أنظمة الإحداثيات المستعملة في المملكة الأردنية الهاشمية، والتي سوف نتعامل فيها مع نظام التربيع الفلسطيني (Cassini) .



الشكل(76): لائحة تصدير النقاط المرصودة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land).

بعد تصدير الملف ومشاركته باستخدام إحدى وسائل المشاركة، مثل البريد الإلكتروني، يتم إضافة ملف محرّر النصوص الخاص بالنقاط إلى برنامج (Arc Map)، كما في الشكل(77)، من أجل إجراء العمليات الإحصائية عليها.

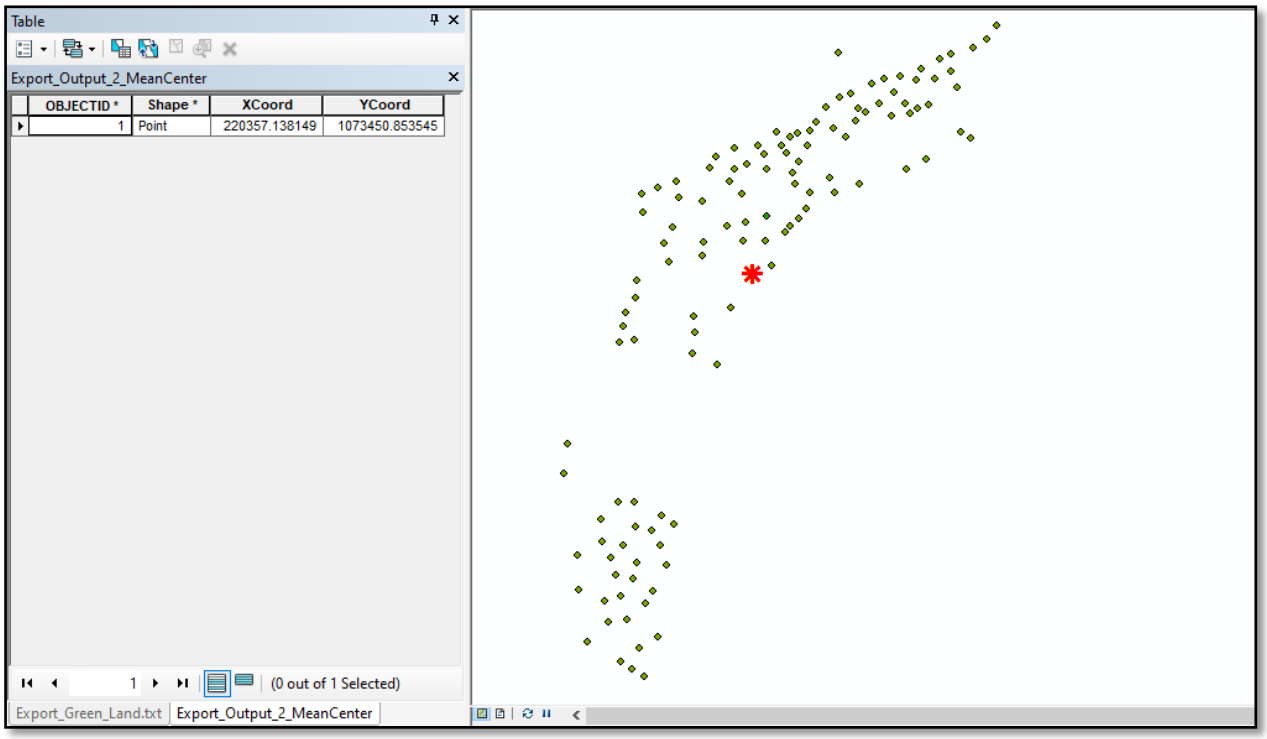


الشكل(77): نقاط الرصد كما تظهر داخل برنامج (Arc Map).

المصدر : عمل الباحث، بالاعتماد على برنامج (Arc Map).

بعد إضافة ملف محرر النصوص الخاص بالنقاط المرصودة، يتم عرض هذه النقاط على البرنامج بوساطة الأدوات المخصصة لذلك، ثم يتم تحديد نظام إحداثيات مطابق لنظام الإحداثيات الذي تم تصدير الملف إليه؛ لتجنّب حدوث أخطاء في عملية الإحصاء، وإظهار النتائج المطلوبة للمقارنة مع التطبيق.

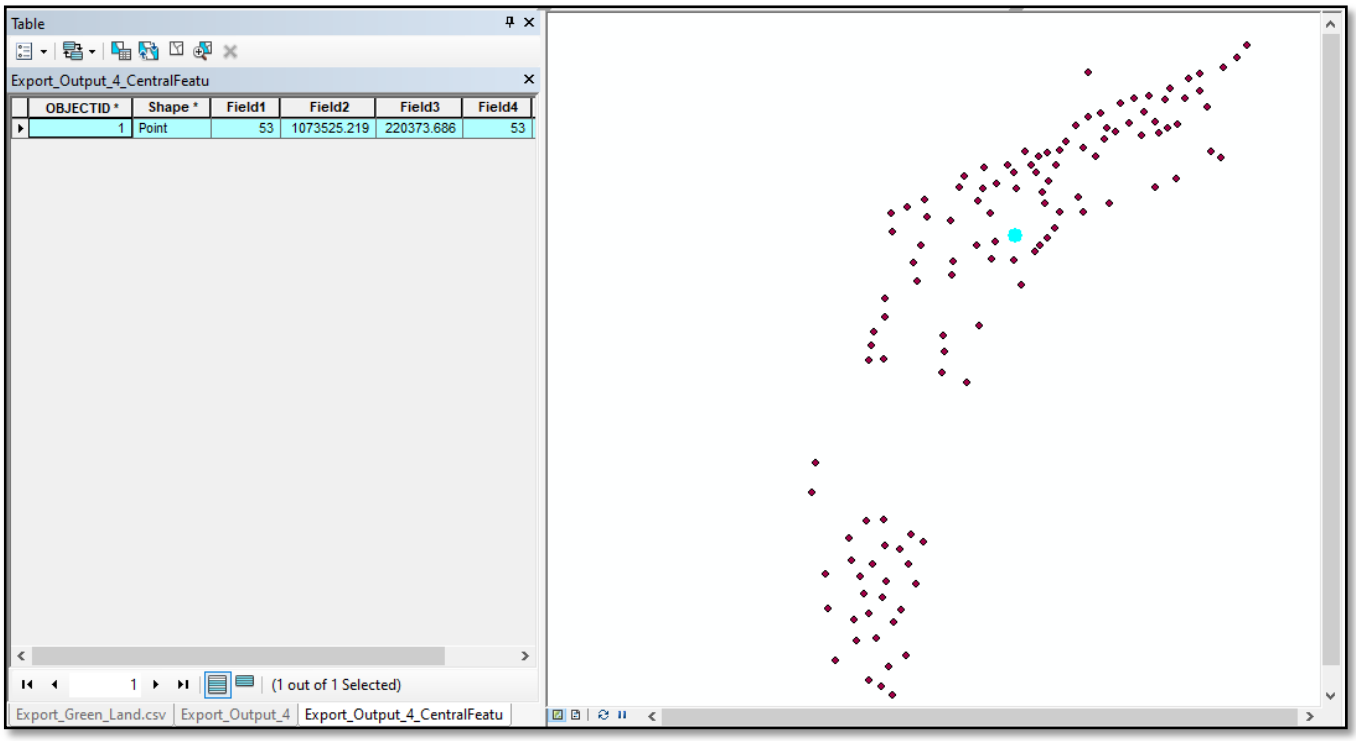
في البداية، سوف نقوم بمقارنة العملية الإحصائية للتطبيق (Mean Center) التي أظهرت أن المركز المتوسط للنقاط هي الإحداثية: (220357.138، 1073450.854)، وأظهرت النتائج داخل برنامج (Arc Map) أن إحداثيات المركز المتوسط هي: (220357.138، 1073450.854)، كما في الشكل(78)، وذلك يدلّ على تطابق النتائج للعملية الإحصائية بين تطبيق (Green Land)، وبرنامج (Arc Map).



الشكل(78): نتائج العملية الإحصائية (Mean Center) داخل برنامج (Arc Map).

المصدر: عمل الباحث، بالاعتماد على برنامج (Arc Map).

أما في ما يخصّ مقارنة العملية الإحصائية (Central Feature)، فقد أظهرت النتائج أن البناء رقم (53) هو البناء الأكثر تمركزًا بالنسبة لبقية الأبنية، وعند مقارنة هذه النتائج مع النتائج التي تم إظهارها في برنامج (Arc Map)، نجد أن النتائج متطابقة، حيث تم اختيار البناء رقم (53)، كما في الشكل(79)، حسب العمليات الإحصائية التي تمت على النقاط داخل برنامج (Arc Map).



الشكل(79): نتائج العملية الإحصائية (Central Feature) داخل برنامج (Arc Map).

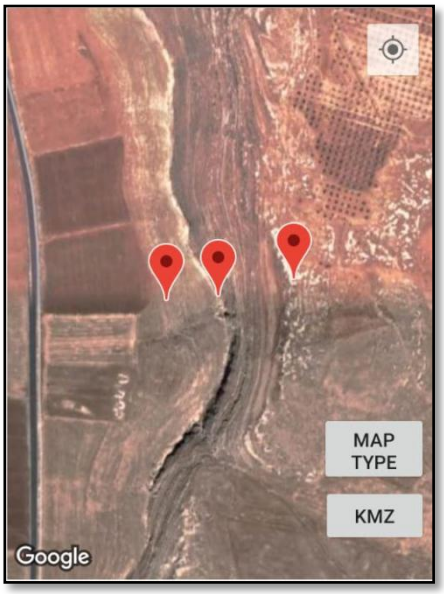
المصدر: عمل الباحث، بالاعتماد على برنامج (Arc Map).

أما في ما يخصّ الدّراسة الأخرى المراد عملها، فهي رصد طبوغرافية الأرض، حيث تم اختيار موقع لرصد النقاط الطبوغرافية لمنطقة تقع بالقرب من منطقة (راكين) التي تقع شمال محافظة الكرك، جنوب المملكة الأردنية الهاشمية لمعرفة عمق الوادي وتحديد درجة الانحدار لبناء جسر يوصل بين أطراف الوادي ويتصل مع الطريق المعبد، من خلال أخذ ثلاث نقاط؛ منها نقطتان على الأطراف، وواحدة في العمق، وذلك من خلال الخطوات التالية:

1-يتم تحديد موقع الدّراسة من خلال برنامج (Google Earth)، كما في الشكل(80)، وذلك بتحديد ثلاث نقاط تقريبية للمنطقة المراد الدّراسة فيها، ثم يتم حفظ هذه النقاط في ملف (KMZ) ، ومن ثم إضافة الملف إلى التطبيق بوساطة أداة (Add KMZ) لتظهر على الخرائط داخل التطبيق.

الشكل (80): نقاط منطقة الدّراسة حسب الصورة الجوية.

المصدر: عمل الباحث، بالاعتماد على برنامج (Google Earth).



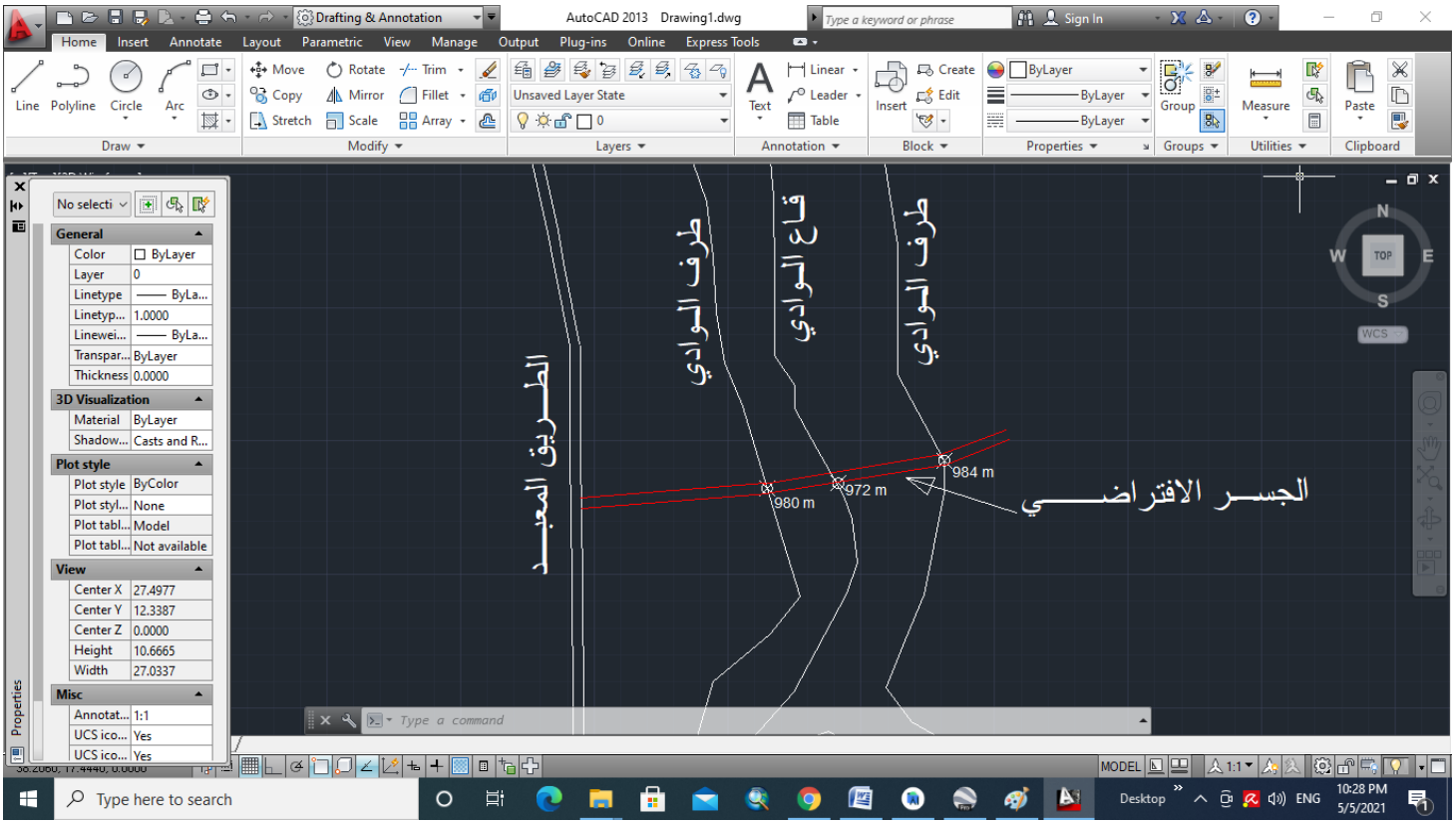
الشكل(81): ملف (KMZ) للنقاط المراد رصدها لمنطقة الدّراسة.

المصدر: عمل الباحث، بالاعتماد على تطبيق (Green Land).

2-في المرحلة التالية، كما في الشكل(81)، وهي الوصول إلى موقع النقاط، والتي تظهر على الخريطة، حيث يتم الموقوف عند تلك النقطة ورصده من خلال الدخول إلى الأداة المخصصة (Topography)، واختيار عملية الرصد اليدوية

(Manual Topo)، ثم النقر على زر (Me)، فتظهر إحداثيات النقطة المراد رصدها، بالإضافة إلى دقة رصد الموقع، ومنسوب النقطة، بعد ذلك يتم النقر على زر(Save) لحفظ النقطة المراد رصدها، ويتم تكرار العملية على بقية النقاط المطلوبة، مع العلم أن النقاط التي يتم رصدها أثناء العمل الميداني تكون بنظام إحداثيات جغرافي؛ بسبب نظام الإحداثيات الذي تبثّه الأقمار الصناعية، والتي يلتقطها الجهاز.

3-وبعد أن تم رصد هذه النقاط، يتم الذهاب إلى أداة (Data Base) لمعاينة النقاط التي تم رصدها، بالإضافة إلى ذلك، يمكن تصدير النقاط التي تم رصدها بنظام إحداثيات محلي بدلًا من الجغرافي لإجراء العمليات الحسابية عليها داخل برامج متخصصة، مثل (Auto Cad) الشكل(82).



الشكل (82): نقاط الرصد الطبوغرافية كما تظهر على برنامج (Auto Cad).

المصدر: عمل الباحث، بالاعتماد على برنامج (Auto Cad).

14.3 مقارنة دقة الرصد الميداني بين تطبيقات (أندرويد) الجغرافية، وأجهزة (GPS) الدقيقة:

تعدّ أجهزة (GPS)، أو أجهزة (RTK-Real time kinematic)، كما في الشكل (83)، من الوسائل الحديثة والتقنية التي يمكن من خلالها رصد الواقع الحقيقي بشكل دقيق جدًّا قد يصل مقدار الخطأ فيها إلى واحد سنتيمتر، والتي يكون فيها التقاط إشارة الأقمار الصناعية وتحليلها، يختلف كليًّا عن تقنية (GPS) الموجودة في الأجهزة الذكية المحمولة.

لا بد لنا، هنا، في هذا البحث من التطرّق إلى موضوع الدقة في رصد النقاط؛ لأن نوع الدّراسة المراد العمل عليها، والوصول فيها إلى نتائج، ربما تكون بحاجة إلى دقة كبيره لا يمكن توفيرها من خلال التطبيق (Green Land)، لذلك (قبل القيام بأي دراسة) يجب تحديد مقدار دقة العمل الميداني؛ ليتم اختيار الوسائل المطلوبة، ودراسة مدى دقتها في الرصد للوصول إلى النتائج المطلوبة.

الشكل (83): جهاز (RTK) من نوع (Stonex) .

المصدر: موقع شركة سيجما للأعمال المساحية.

لفهم المقارنة في دقة الرصد بين تطبيق (Green Land)، وجهاز (RTK)، سوف نقوم برصد ميداني لعدّة نقاط باستخدام التطبيق وجهاز (RTK) لنفس النقاط لمقارنة النتائج، وسوف تتم المقارنة في الدقة على المحور الأفقي والمحور العامودي،

والمقصود بذلك إيجاد مقدار قرب النتائج المسجّلة من النتيجة الفعلية لكل من الأدوات المستخدمة.

الجدول (1)

بيانات الرصد الميداني للنقاط:

رقم النقطة	الأداة المستخدمة	الإحداثيات X	الإحداثيات Y	الارتفاع	مقدار الخطأ الأفقي بوحدة المتر	مقدار الخطأ العامودي بوحدة المتر
1	Green Land	219009.1	1072504.2	986.517	3.86	2.83
	RTK Stonex	219006.2	1072501.6	983.687	0.02	0.03
2	Green Land	219119.2	1072641.1	965.01	1.57	5.81
	RTK Stonex	219117.7	1072641.4	970.82	0.01	0.025
3	Green Land	219118.4	1072662.4	966.01	1.12	2.81
	RTK Stonex	219119.3	1072661.7	968.82	0.02	0.03
4	Green Land	219120.1	1072643.1	995.01	3.14	3.76
	RTK Stonex	219121.7	1072640.4	998.77	0.01	0.02
5	Green Land	219132.2	1072650.1	949.01	2.75	4.18
	RTK Stonex	219133.4	1072652.6	944.83	0.01	0.025
6	Green Land	219160.2	1072644.1	954.21	3.66	5.11
	RTK Stonex	219162.7	1072641.4	959.32	0.02	0.03
7	Green Land	219218.3	1072751.7	968.01	3.93	4.71
	RTK Stonex	219219.7	1072755.4	972.72	0.01	0.025
8	Green Land	219111.2	1072818.1	985.41	1.99	3.8
	RTK Stonex	219112.4	1072819.7	981.61	0.03	0.02
9	Green Land	219502.3	1072889.2	980.51	2.49	3.11
	RTK Stonex	219504.7	1072888.4	983.62	0.01	0.03
10	Green Land	219908.5	1072546.1	977.91	2.97	2.47
	RTK Stonex	219909.8	1072543.4	975.44	0.01	0.025

المصدر: عمل الباحث من العمل الميداني 2020

بعد الحصول على النتائج السابقة، تبيّن أن نوع الدّراسة ومقدار الدقة المطلوبة هي التي تحدّد الوسائل المطلوب استخدامها للوصول إلى النتائج، فإذا كان المراد من العمل الميداني دقة مقبولة نوعًا ما، فيكون اختيار التطبيق (Green Land) هو الأفضل، أما إذا كان المراد من العمل الميداني دقة عالية تصل إلى واحد سنتيمتر،

فيجب اختيار أدوات أو وسائل عالية الدقة، مثل جهاز(RTK) للوصول إلى نتائج عالية الدقة.

ولكن (وعلى ما تم ذكره سابقًا) يعدّ التطبيق هو الأفضل للاستخدام الميداني للجغرافيين؛ كون أغلب الدّراسات الميدانية لا تحتاج إلى دقة عالية في الرصد، ولقلّة تكلفتها، وسهولة استخدامها، وسرعة إنجاز الإعمال فيها، حيث إن أجهزة (RTK) مرتفعة الثمن وبحاجة إلى أصحاب اختصاص للعمل عليها، بالإضافة إلى ذلك، تحتاج نقاط الرصد فيها إلى معالجة من خلال برامج متخصصة، ولا يمكن الوصول فيها إلى نتائج مباشرة خلال العمل الميداني.

15.3 برمجة أدوات (Arc GIS) داخل نظام (أندرويد):

توفر شركة (Esri) المصممة لبرنامج (Arc Gis) مكتبات رقمية يمكن من خلالها استخدام أدوات التحليل المكاني الجاهزة للعمل التي يمكن إضافتها أثناء القيام في برمجة التطبيقات الجغرافية، من خلال في ما يعرف بالـ(ArcGIS Runtime API for Android)، حيث يمكن أنشاء تطبيقات الأجهزة المحمولة التي تتضمن إمكانات، مثل: التعيين، والتشفير الجغرافي، والتوجيه، والمعالجة الجغرافية، والتحليلات المكانية الأخرى، بالإضافة إلى إمكانية استخدام الخرائط الخاصة بشركة (ESRI) .

تسهم هذه العملية في إيجاد أدوات تمكّن من رفع مستوى العمل الجغرافي في التطبيقات التي تتعامل مع التحليل المكاني، حيث إن وجود أدوات جاهزة ذات وظائف جغرافية داخل التطبيق تمكّن المطوّرين من تسهيل عملية برمجة أدوات التحليل المكاني، بالإضافة إلى توفيرها الدقة العالية في العمل التي يمكن الوصول من خلالها إلى نتائج دقيقة، في أقل وقت، وجهد، وتكلفة.

تعدّ هذه المكتبات أو الأدوات التي تقدّمها شركة (Esri) مجانية، ويمكن الحصول عليها من خلال الموقع الخاص بالشركة، بالإضافة إلى سهولة إضافتها إلى التطبيقات الخاصة بمجال التحليل المكاني.

توفر هذه المكتبات، أيضا، أدوات خاصة بمشاركة البيانات من خلال ما يسمّى بـ(السحابة) التي يمكن بوساطتها مشاركة نتائج عمليات الرفع والتحليل والمعالجة مع الآخرين، وذلك لتوفير بنية تحتية للبيانات الجغرافية، يمكن من خلالها تبادل البيانات من أي طرف يودّ الوصول إليها.

16.3 الترميز الجغرافي (GeoCoding) :

لترميز الجغرافي هو عملية برمجة مجموعة من إحداثيات خطوط الطول والعرض إلى عنوان يمكن قراءته والعكس صحيح؛ إذ يمكن، أيضًا، إدخال كلمات وتحويلها إلى عناوين وخطوط طول وعرض، وهذا ما يسمّى بالترميز الجغرافي العكسي.

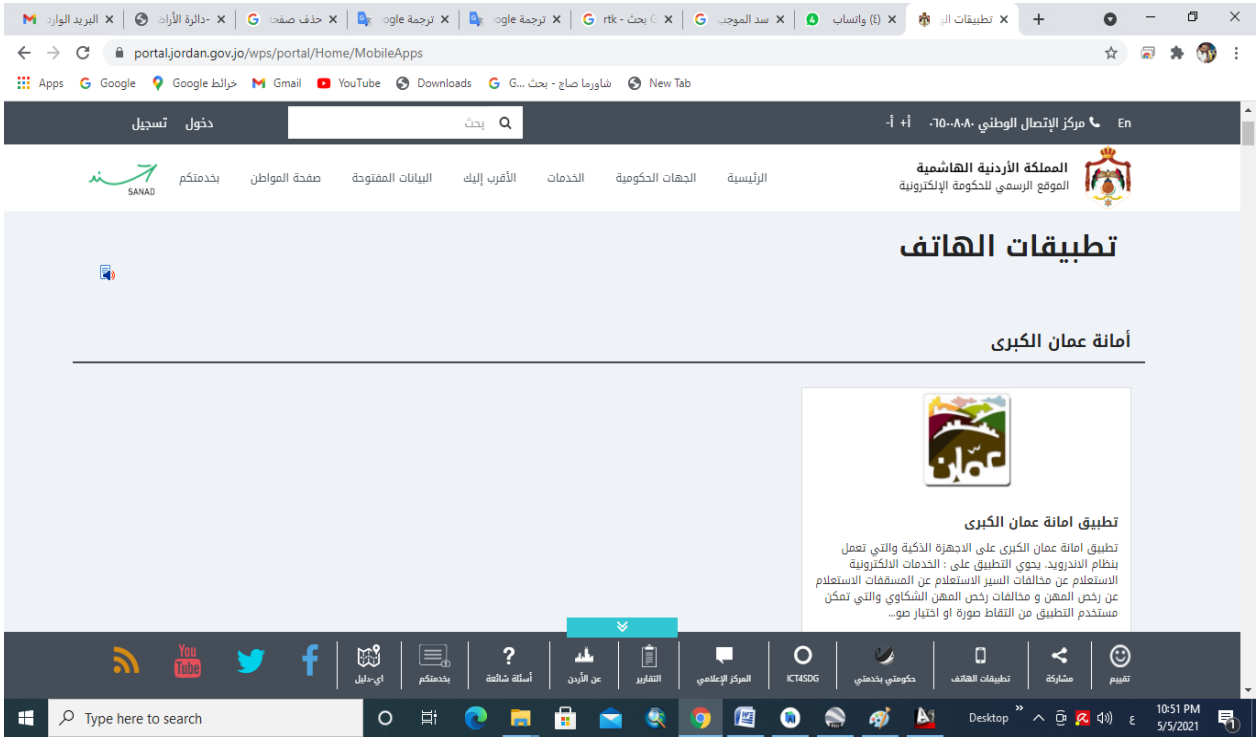
يعدّ الترميز الجغرافي من أهم الخدمات التي تقدّمها لنا منصة خرائط (Google) العالمية؛ كونها تمكّننا من الحصول على معلومات جغرافية مهمة، كإحداثيات موقع ما (Latitude & Longitude) انطلاقًا من العنوان أو اسم المنطقة، والعكس صحيح؛ أي أن (Geocoding Api) تمكّننا، كذلك من الحصول على عنوان نقطة ما على الخريطة انطلاقًا من إحداثياتها (Bouguer, 2015).

يمكن القول أن عملية الترميز الجغرافي هي أقرب ما تكون إلى عملية الربط بين المعلومات المكانية والمعلومات الوصفية، بالإضافة إلى ذلك إمكانية البحث عن معلومة مكانية بوساطة معلومة وصفية، أو البحث عن معلومة وصفية بوساطة معلومة مكانية، وكل ذلك يتم عن طريق الخرائط التي تدعمها التطبيقات والمرتبطة بخوادم تتبع لشركة (Google) العالمية.

17.3 حلول الجغرافيا المكانية في التطبيقات الحكومية:

تعدّ الحكومة في المملكة الأردنية الهاشمية من الحكومات الرائدة في التوجّه نحو استخدام تكنولوجيا المعلومات والاتصالات كأداة لتحقيق نظام حوكمه أفضل، حيث تقوم على تعزيز وصول المعلومات والخدمات الإلكترونية إلى الجمهور والهيئات والجهات الحكومية.

وتقوم باستخدام الهاتف الذكي ووسائل الاتصال المتنقلة لتوفير الخدمات الإلكترونية لتوفير خدمات عن بُعد دون الحاجة إلى مراجعة المكاتب الحكومية لتوفير الوقت والجهد في الحصول على المعلومات المطلوبة ضمن تطبيقات يتم تصميمها بشكل بسيط وسهل يمكن أن يستخدمه جميع شرائح المجتمع، والتي توفرها ضمن موقعها الإلكتروني، كما في الشكل (84).



الشكل (84): الموقع الإلكتروني للحكومة الإلكترونية.

تعدّ جميع التقنيات الموجوده في معظم التطبيقات الحكومية حلول مكانية بنظام (أندرويد) يتم توفيرها لخدمة المستخدم وتسهيل عملية الوصول إلى الأماكن المطلوبة، حيث يقوم التطبيق على استخدام خرائط (Google) وقواعد بيانات متصلة بالتطبيق، بالإضافة إلى استخدام تقنية (GPS) التي تعمل على تحديد موقع متلقي الخدمة، ويقوم التطبيق بإرسال إحداثيات مكان الوقوف، ثم إرسالها إلى الخوادم المرتبطة بالتطبيق، ثم تقوم الخوادم بمقارنة هذه البيانات، بعد ذلك تقوم بعرض النتائج على الخرائط.

تعدّ عملية تحديد نطاق الخدمة (Service Area) من الحلول المكانية التي يتم استخدامها في الكثير من التطبيقات والبرمجيات المتخصصة، حيث تقوم هذه العملية على تحديد أقرب مصدر خدمة بالنسبة لموقع الوقوف عن طريق تحديد أقصى

مسافة سير، أو أقصى وقت وصول للخدمة، بالإضافة إلى ذلك يمكن تحديد نطاق الخدمة لمركز الخدمة، مثل تحديد نطاق الخدمة لمحطات الدفاع المدني التي يشترط فيها الوصول إلى موقع الحادث خلال ثماني دقائق، حيث يتم احتساب الوقت المستغرق للمرور عبر الطرق المتصلة بمركز الخدمة، وعلى هذا الأساس يتم تحديد نطاق الخدمة.

تمكّن التطبيقات الحكومية المستخدم من الحصول إلى الخدمة في أقل وقت وجهد، ومن شأنها رفع مستوى الخدمات المقدمة ورفع مستوى التعامل، بالإضافة إلى تقليل الضغط الحاصل على الدوائر الحكومية بسبب المراجعين، وهذا ما تهدف إليه الحلول المكانية في التطبيقات الحكومية.

يمكن تلخيص عمل الحكومة الإلكترونية في التطبيقات ذات الحلول المكانية من خلال إنشاء بعض التطبيقات المختصة في أعمال بعض الدوائر الحكومية، ومنها:

1- تطبيق دائرة الأراضي والمساحة:

يعدّ تطبيق دائرة الأراضي من أنجح التطبيقات الحكومية، بالإضافة إلى استنادها إلى الحلول المكانية في تقديم الخدمات، من خلال مجموعة الأدوات البسيطة التي يمكن العمل عليها بشكل سهل وخالٍ من أي تعقيدات.

يعمل مبدأ تطبيق دائرة الأراضي والمساحة على عرض الخرائط وإظهار قطع الأراضي عليها، من خلال البحث عن معلومات القطعة المراد إظهارها، حيث تعدّ المعلومات الوصفية هي ذاتها معلومات القطعة التي يتم إدخالها، وأما المعلومات المكانية فهي المضلع الذي يظهر على الخريطة لتحديد مكانها وشكلها على الخريطة.

يتصل تطبيق دائرة الأراضي بقاعدة بيانات موجودة على خوادمها، فعندما يتم طلب خدمة يتم الاتصال بهذه الخوادم ومقارنة المعلومات الوصفية المدخلة مع المعلومات الوصفية المخزنة على الخوادم، ثم يقوم بعد بمطابقة هذه المعلومات بعرض شكل القطعة على الخريطة التي تم طلبها بوساطة المعلومات الوصفية، حيث تعدّ طريقة عرض قطعة الأرض عملية لعرض ملف (KMZ) على الخريطة.

كما يمكن عرض قطعة الأرض على الخرائط من خلال تقنية (GPS)، حيث يقوم التطبيق على تحديد موقع الوقوف، ثم إرسال هذا الموقع أو الإحداثية إلى

الخوادم، ثم يقوم بمقارنة هذه الإحداثية بالإحداثيات المحفوظة لديه، ثم يقوم بعرض قطعة الأرض على الخريطة في مكان الوقوف.

يهدف تطبيق دائرة الأراضي والمساحة، كما في الشكل (85)، إلى تسهيل تقديم خدمات الدائرة للمواطنين إلكترونيًّا على مدار الساعة، حيث يقدّم التطبيق مجموعة الخدمات، مثل:

البحث عن قطعة أرض عن طريق إدخال معلوماتها: (المحافظة، والمديرية، والقرية، والحوض، والحي، ورقم القطعة)، أو إحداثياتها، وإظهار كيفية الوصول إليها، بالإضافة إلى عرض إحداثيات القطعة عن طريق إدخال: (المحافظة، والمديرية، والقرية، والحوض، والحي، ورقم القطعة)، وعرض معلومات القطعة للموقع الحالي للمستخدم عن طريق الأقمار الصناعية.

الشكل (85): واجهة تطبيق دائرة الأراضي والمساحة .

تعدّ الأمور التي تم ذكرها في تطبيق دائرة الأراضي والمساحة حلولًا مكانية تستخدم بوساطة نظام (أندرويد)، حيث توفر الجهد والوقت للوصول إلى المعلومات المطلوبة من خلال ربط جميع هذه التقنيات، مثل الخرائط، وقواعد البيانات، وملفّات

(KMZ)، وتقنية (GPS) في تطبيق واحد تخدم المستخدمين للارتقاء بالخدمات الحكومية.

2-تطبيق الخارطة الصحية :

يعدّ تطبيق الخارطة الصحية، كما في الشكل(86)، من التطبيقات التي تستخدم الحلول المكانية في تقديم الخدمات الصحية، من خلال إظهار أماكن المستشفيات والمراكز الصحية المنتشرة في المملكة الأردنية الهاشمية على الخرائط، وذلك لتحديد أقرب مستشفى أو مركز صحي، وتسهيل عملية الوصول إليها في أقل مسافة ووقت.

كما يوفر التطبيق إمكانية البحث حسب موقع الوقوف عن أقرب مستشفى أو مركز صحي على نطاق مسافة معين (نطاق الخدمة)، كما في الشكل(87)، ويتم تحديده من قبل المستخدم، بالإضافة إلى إمكانية فلترة النتائج، الشكل(88)، بتحديد نوع المستشفى، إما خاص، أو حكومي، أو إذا كان مركزًا صحيًّا.

الشكل (86): واجهة تطبيق الخارطة الصحية.

الشكل (87): نطاق الخدمة كما يظهر في تطبيق الخارطة الصحية.

الشكل (88): فلترة النتائج كما تظهر في تطبيق الخارطة الصحية.

18.3 حلول الجغرافيا المكانية في التطبيقات الخاصه:

تعدّ الشركات الخاصة عملية الحوسبة أو البرمجة من الأسباب الرئيسة في زيادة الأرباح، وتنظيم الأعمال، والحفاظ على كفاءة المنتج، لذلك تبدي هذه الشركات اهتمامًا كبيرًا في هذا الجانب، فالكثير من الشركات تطرّقت لاستخدام الحلول المكانية في تطبيقاتها، مثل: تطبيق شركة أوبر، الشكل(89)، أو شركة كريم التي تعتمد على تقديم خدمة توصيل الزبائن عبر المركبات عن طريق طلب الخدمة الذي يعتمد على مكان الوقوف، وإيجاد أقرب سيارة توصيل استنادًا إلى موقع الوقوف، حيث تعدّ هذه الطريقة من الطرق الرائجة في البحث عن أقرب خدمة بالنسبة لمكان الوقوف، والتي تسمّى نطاق الخدمة، حيث يتم عرض خريطة يظهر فيها موقع المستخدم، بالإضافة إلى المركبات المجاورة، ومعلومات هذه المركبة، ومعلومة المستخدم، فعند اختيار المركبة المراد التوصيل بها، يصل إشعار إلى صاحب المركبة أن شخصًا ما يريد خدمة توصيل إلى مكان ما، بالإضافة إلى ذلك، تظهر لدى صاحب المركبة جميع المعلومات الخاصة بالراكب أو المستخدم، بعد ذلك تتم عملية التوصيل.

الشكل (89): واجهة تطبيق أوبر.

بعض التطبيقات التي قام بتصميمها الباحث لبعض الشركات، والتي تحتوي على حلول مكانية لعمل هذه التطبيقات، ومنها:

1-تطبيق (Calibrated)، الشكل(90)، الخاص بشركة البلقاء لفنّ المساحة والوكيل لشركة (سوكيا) اليابانية، حيث يوفر التطبيق حلولًا مكانية للبحث عن نقاط التحكم الأرضية، والتي يمكن إظهارها حسب مكان الوقوف في العمل الميداني، حيث إن المستخدم يحتاج أثناء عملة الميداني إلى معرفة نقاط التحكم التي توفرها دائرة الأراضي والمساحة، والتي يحتاج، من خلالها، إلى تثبيت حدود الملكيات العقارية، فالتطبيق يحتوي على خرائط تظهر عليه نقاط التحكم الأرضية المحيطة به، والتي يكون عددها أربع نقاط في أقل الأحوال، بالإضافة إلى ذلك، يتم تصدير هذه النقاط في ملف محرّر النصوص بامتداد (txt)، والتي تتم إضافتها إلى البرنامج الأساس الخاص بالشركة لإتمام عملية المعايرة المطلوبة لنقاط التحكم الأرضية.

الشكل (90): واجهة تطبيق (Calibrated).

2- تطبيق (Jordanian Converter)، كما في الشكل (91)، وهو تطبيق يقوم على تحويل الإحداثيات المحلية إلى إحداثيات جغرافية أو العكس، بالإضافة إلى وجود خرائط يمكن الوصول من خلالها إلى موقع هذه النقاط المدخلة، حيث يتم استعمال هذا التطبيق من قبل المستخدمين لصعوبة إظهار النقاط ذات الإحداثيات المحلية على برامج متخصصة، مثل (Google Earth)، بالإضافة إلى ذلك، إمكانية الوصول إلى هذه النقاط عبر الخرائط المدعومة في الواقع الحقيقي دون استخدام أجهزة (GPS) المحمولة، كما يوفر التطبيق نظامي إحداثيات محلي، وهما: نظام الإحداثيات الأردني، ونظام التربيع الفلسطيني، المستخدمين في المملكة الأردنية الهاشمية، والمعتمدين لدى دائرة الأراضي والمساحة.

الشكل (91): واجهة تطبيق (Jordanian Converter).

تعدّ جميع التطبيقات السالفة الذكر تطبيقات قائمة (بشكل أساس) على الحلول المكانية التي توفر أداة سهلة للمستخدمين لتسهيل عملية الحصول على الخدمة المطلوبة، والتي يتم فيها توفير الوقت والجهد.

الفصل الرابع

النتائج والتوصيات

1.4 النتائج :

1-أظهرت نتائج الدّراسة أن لغة البرمجة (أندرويد) يمكن استخدامها في علوم الجغرافيا بشكل واسع، حيث إن الأدوات التي توفرها هذه اللغة تساعد بشكل كبير الجغرافيين في فهم الظواهر، سواء في الشق التحليلي للنتائج، أو الشق الميداني، بالإضافة إلى القدرة عن الاستغناء عن الأدوات والبرامج المتخصصة، مثل: برنامج (Arc Map)، أو برنامج (Google Earth) أو جهاز (RTK).

2-تعدّ لغة برمجة (أندرويد) من اللغات المفتوحة المصدر والمرنة، والتي يمكن تطوير الأدوات بواسطتها بشكل مستمر، بالإضافة إلى إمكانية تصميم أي أداة عليها من شأنها القدرة على تحليل البيانات فيها؛ إذ يمكن إضافة أي معادلة حسابية داخل الأدوات لتوفير إمكانية تحليل بأسلوب معين للوصول إلى نتائج معينة، مثل القيام بإضافة أدوات إحصائية، كالمركز المتوسط، أو الظاهرة المركز، والكثير من الأدوات التي يمكن إضافتها للحصول على النتائج المطلوبة.

3-يمكن اعتبار حلول الجغرافيا المكانية على أنها مجموعة من الأدوات مرتبطة مع بعضها بعضًا تم برمجتها بأسلوب معين للقيام بالأعمال المطلوبة للوصول إلى نتائج معينة يتم من خلالها توفير الوقت والجهد المبذولين في العمل الميداني.

4-توفر الخرائط في التطبيقات نظرة حقيقية واقعية في العمل الميداني؛ لأنها تساعد في معاينة الظواهر وإظهار النتائج عليها، ولكون هذه الخرائط صورة للواقع ملتقطة بوساطة الأقمار الصناعية تكون فيها المعالم واضحة وبأبعاد حقيقية، كما أنها تحمل إحداثيات حقيقية يمكن مطابقتها مع برامج أخرى متخصصة، مثل برنامج (Google Earth).

5-أظهرت الدّراسة مطابقة النتائج التي تم العمل عليها في التطبيق، مع النتائج التي تم الحصول عليها من البرامج المتخصصة لنفس نقاط الرصد التي تم رصدها من أجل العمليات الإحصائية، حيث تم تصميم الأدوات الإحصائية في التطبيق بشكل

مطابق للأدوات الموجودة في البرامج المتخصصة، مثل برنامج (Arc Map) الذي يعدّ من البرامج الرائدة في عملية التحليل المكاني.

6- وجود أدوات إضافية، مثل أداة إضافة طبقات (KMZ)، تعدّ نقلة نوعية في التطبيق؛ كونها تساعد في عملية تحديد موقع الدّراسة، بالإضافة إلى ذلك، يمكن استعمالها لإضافة ملفّات من برامج متخصصة، مثل (Arc Map) كملفّات (Shape file)، وذلك بعد تحويلها إلى ملفّات (KMZ) ليتمكّن التطبيق من إضافتها دون حدوث أخطاء داخل التطبيق.

7- أظهرت نتائج الدّراسة أن دقة النتائج تعتمد على دقة الأدوات المستخدمة في العمل الميداني، حيث يحتاج الوصول إلى نتائج ذات دقة عالية إلى أدوات متخصصة، بينما الوصول إلى نتائج مقبولة نوعًا ما لا يحتاج إلى أدوات متخصصة، بل إلى ما يوفره التطبيق من الأدوات، ولكن يجب التنويه، هنا، إلى أن القصد من دقة النتائج هو مقدار دقة رصد النقاط؛ كون العمليات الإحصائية تكون متطابقة في عملياتها مع أي برنامج متخصص آخر أثناء عملها على نقاط الرصد، ولكن يختلف مقدار الخطأ في الرصد في تحديد مكان الظاهرة حسب الأدوات المستعملة في عملية الرصد، حيث تمت المقارنة بين الأدوات المتخصصة، مثل جهاز (RTK)، والتطبيق بالنسبة لرصد النقاط التي تحتوي على إحداثيات أفقية، أو النقاط التي تحتوي على إحداثيات أفقية وعمودية، كما تبيّن من الدّراسة التي أجريت عليها أن مقدار دقة النتائج يعتمد على دقة أدوات الرصد المستخدمة.

2.4 التوصيات :

1-يوصي الباحث بضرورة إضافة مواد لتعلّم البرمجة في أقسام الجغرافيا، من أجل مواكبة الجغرافيين لهذا المجال؛ كون المجال الجغرافي عامة، أصبح يعتمد (بشكل أساس) على أمور البرمجة التي من شأنها رفع مستوى العمل الجغرافي بشقيه التحليلي والميداني.

2-ربط البرامج المتخصصة ذات العلاقة مع الأقسام الجغرافية؛ لما توفره هذه البرامج من أدوات يمكن من خلالها توفير دقة عالية في النتائج، بالإضافة إلى توفير أدوات ذات دقة عالية في القياس من باب التوضيح لمبدأ الحصول على نتائج دقيقة من أدوات ذات دقة.

3-يوصي الباحث بفتح قنوات عمل مشترك مع الشركات المتخصصة في عمل تطبيقات حلول مكانية لتصميم تطبيقات من شأنها رفع مستوى المجال الجغرافي، من خلال دمج الأفكار المشتركة، وإثراء العلوم الجغرافية بهذا المجال.

قائمة المصادر والمراجع

أولًا: المراجع العربية.

الحجار، محمد بركات (2011)، "**أثر الإعلان التجاري على السلوك الشرائي لمستخدمي خدمات الهواتف الذكية المحمولة في مدينة عمّان- دراسة ميدانية**". (دراسة ماجستير منشورة).جامعة الشرق الأوسط، عمّان، الأردن.

داود، جمعة، (2019)، **نظم المعلومات الجغرافية وقواعد البيانات**، مصر، (عبر الإنترنت)، متاح: (13 شباط) :https://www.academia.edu

زهر ، سوزان (2016) ، "**مهارات الطلاب في استخدام الهواتف الذكية للوصول إلى مصادر المعلومات دراسة مقارنة بين كليتي الطب والآداب في جامعة بيروت العربية**" Cybrarians Journal، 43

العزاوي، علي،(2013)، "**التحليل الإحصائي المكاني في نظم المعلومات الجغرافية**"، مذكرة، جامعة الموصل، الموصل.

وزارة الاقتصاد الرقمي والريادة ،(2021)، " **الحكومة الإلكترونية**"،(عبر الإنترنت) ، متاح: (13 شباط):https://modee.gov.jo

الأسطل ، محمود ،(2009)، "**أثراء وحدة البرمجة في مقرر تكنولوجيا المعلومات في ضوء المعايير الأدائية للبرمجة وأثره على مهارة البرمجة لدى طلاب الصف الحادي عشر**" .(دراسة ماجستير منشورة).الجامعة الاسلامية، غزة، فلسطين.

طبيه ، احمد ،(2008)، "**مبادئ الإحصاء**"،عمان، الاردن.

هلول ، محمد ،(2018)، "**تقييم كفاءة طرق النقل البرية المعبدة في محافظة القادسية باستخدام برامج (GIS) والأساليب الاحصائية**"،جامعة القادسية ، العراق.

ويكيبيديا ،(2021)، "**أندرويد استوديو**"،(عبر الإنترنت)+، متاح: (13 شباط): https://ar.wikipedi.android

Bouguer ، Aissa،(2015)، **Google Maps Api**، مذكرة، (عبر الإنترنت)، متاح: (13 شباط) :https://github.com/aissa-bouguern

ثانيًا: المراجع الأجنبية:

Taba .S. E,(2014).**Empirical studies on the relation between user interface design and perceived quality of android applications**, (published master thesis), University Kingston, Ontario ,Canada.

Google Play Store، **"تطبيق دائرة الأراضي والمساحة"**,(On-Line),available: (Feb28):https://play.google.com/store/apps

Google Play Store، **"تطبيق الخارطة الصحية "**,(On-Line),available: (Feb28):https://play.google.com/store

Google Play Store, **"تطبيق اوبر"**,(On-Line),available: (Feb28):https://play.google.com/store/apps/

Al Balqa For Surveying Arts, "**Calibrated app**" ,(On-Line),available: (Feb28):https://ar-ar.facebook.com/Eng.organisation

Mediafire, "**Jordanian Converter**",(On-Line),available: (Feb28): https://www.mediafire.com

المعلومات الشخصية

الاسم: يوسف سالم العصيات

التخصص: الماجستير في نظم المعلومات الجغرافية (GIS)

الكلية: العلوم الاجتماعية

سنة التخرج: 2021