



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Farm Zone Manager

Διαχείριση Τμημάτων Αγροκτήματος

ΔΙ.ΠΑ.Ε. - Πανεπιστημιούπολη Σερρών
Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
Κατεύθυνση Μηχανικών Λογισμικού

Όνοματεπώνυμο:

Σίμων-Ουμπέρτο Μόσχος

Αριθμός Μητρώου:

3741

Επιβλέπων Καθηγητής:

Δρ. Θεόδωρος Λάντζος

Θεματολογία Διπλωματικής Εργασίας

Χαρτογράφηση αγροκτήματος με συμμετοχή θέσης, ορίων και παρεμβολής με χάρτες

Σκοπός της πτυχιακής αυτής είναι σχεδίαση μια εφαρμογής σε κινητό τηλέφωνο η οποία θα επιτρέπει στους αγρότες να χαρτογραφούν τα αγροκτήματά τους. Η Εφαρμογή θα επιτρέπει τον χρήστη όχι μόνο να ορίσει την τοποθεσία στο google maps, αλλά να ορίσει τα σύνορα και την δυνατότητα παρεμβολής χαρτών με αγροτική υπηρεσία και ΤΟΕΒ.

Αποτέλεσμα αυτής της εφαρμογής θα είναι ένα προϊόν το οποίο θα δίνει μια ξεκάθαρη εικόνα στον αγρότη για το χαρτογραφικό υπόβαθρο του αγροκτήματος του, καθώς και η μεταφορά δεδομένων σε άλλα συστήματα όπως εξαγωγή χάρτη σε άλλες εφαρμογές, που θα του επιτρέψει την έγκυρη επίλυση προβλημάτων με οργανισμούς και διαχείρισης αγροκτήματος.

Η εφαρμογή θα επιτρέπει την έτοιμη εισαγωγή και εξαγωγή χωραφιών από το κτήμα καθώς και τις δυνατότητες διαίρεσης και πρόσθεσης γειτονικών κτημάτων.

Ανάλυση Απαιτήσεων Εφαρμογής

Για ανάπτυξη της εφαρμογής θα χρειαστούμε:

- Ένα ολοκληρωμένο προγραμματιστικό περιβάλλον IDE για ανάπτυξη εφαρμογών σε πλατφόρμα Android.
- Ένα γεωγραφικό πληροφοριακό σύστημα (GIS) για την απεικόνιση χαρτών σε Android.
- Μια τοπική βάση δεδομένων για Android, όπου θα αποθηκεύονται τα δεδομένα του χρήστη.
- Ανάλυση δομής βασικών γεωγραφικών αρχείων.
- Βοηθητικές βιβλιοθήκες ανάγνωσης και εγγραφής αρχείων για Android.
- Μια βιβλιοθήκη πράξεων γεωγραφικών συνόρων, όπως ένωση, τομή και διαφορά μεταξύ δύο γεωγραφικά σύνορα.

Περιβάλλον Εργασίας και Βιβλιοθήκες Εφαρμογής

- **Android Studio IDE**, για την δημιουργία της εφαρμογής σε πλατφόρμα Android.
- **Room Database**, ως βάση δεδομένων εφαρμογής.
- **Maps SDK**, ως κύριο GIS της εφαρμογής.
- **ArcGIS Runtime API**, για την υλοποίηση των πράξεων γεωγραφικών συνόρων.
- **Proj4J** της OSGeo, για μετατροπή μεταξύ γεωγραφικών συστημάτων συντεταγμένων.
- **Java ESRI Shapefile Reader** του olenus, για την ανάλυση των αρχείων ShapeFile.
- **JDOM2**, για ποιο εύκολη ανάλυση αρχείων τύπου XML.

Δομή των βασικών γεωγραφικών αρχείων

Keyhole Markup Language (KML)

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <Polygon>
      <extrude>1</extrude>
      <altitudeMode>relativeToGround</altitudeMode>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            lng1,lat1,alt1 lng2,lat2,alt2 lng3,lat3,alt3 ... lng1,lat1,alt1
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
      <innerBoundaryIs>
        <LinearRing>
          <coordinates>
            lng1,lat1,alt1 lng2,lat2,alt2 lng3,lat3,alt3 ... lng1,lat1,alt1
          </coordinates>
        </LinearRing>
      </innerBoundaryIs>
    </Polygon>
  </Placemark>
</kml>
```

Geography Markup Language (GML)

```
<wfs:FeatureCollection
  xmlns="http://www.opengis.net/wfs"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:myns="http://some.example.url/myns">
  <wfs:member> <!-- or <gml:featureMember> -->
    <myns:geomProperty>
      <myns:name></myns:name>
      <gml:Polygon>
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates decimal="." cs=";" ts=" " >
              lng1,lat1 lng2,lat2 lng3,lat3 ... lng1,lat1
            </gml:coordinates>
          </gml:LinearRing>
        </gml:outerBoundaryIs>
        <gml:innerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates decimal="." cs=";" ts=" " >
              lng1,lat1 lng2,lat2 lng3,lat3 ... lng1,lat1
            </gml:coordinates>
          </gml:LinearRing>
        </gml:innerBoundaryIs>
      </gml:Polygon>
    </myns:geomProperty>
  </wfs:member>
</wfs:FeatureCollection>
```

GeoJSON

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [lng1,lat1],
            [lng2,lat2],
            [lng3,lat3],
            [lng4,lat4],
            [lng1,lat1]
          ]
        ]
      }
    }
  ]
}
```

Μετατροπή μεταξύ συστημάτων συντεταγμένων

Για των λόγο αποθήκευσης γεωγραφικών αρχείων σε άλλα γεωγραφικά συστήματα συντεταγμένων θα χρησιμοποιήσουμε την βιβλιοθήκη Proj4J για την μετατροπή των συντεταγμένων σε γεωγραφικό σύστημα συντεταγμένων EPSG 4326 (WGS84) που χρησιμοποιεί το Maps SDK.

Η μετατροπή γίνεται ως εξής:

- Δημιουργούμε ένα αντικείμενο τύπου `CoordinatesHelper`.
- Ορίζουμε των κωδικό `crs` που θέλουμε να μετατρέψουμε με την μέθοδο `setCrs`.
- Κάνουμε μετατροπή το `LatLng` σε `WGS84` με την βοήθεια της μεθόδου `convertPoint`.

```
public class CoordinatesHelper {
    private BasicCoordinateTransform transform;
    public CoordinatesHelper(){
        transform = null;
    }
    public boolean setCrs( String crs ){
        try{
            CoordinateReferenceSystem srcCrs = new CRSFactory()
                .createFromName(crs);
            CoordinateReferenceSystem dstCrs = new CRSFactory()
                .createFromName("EPSG:4326");
            transform = new BasicCoordinateTransform(srcCrs, dstCrs);
            return true;
        } catch ( Exception e ){
            transform = null;
            return false;
        }
    }
    public LatLng convertPoint( LatLng point ) {
        if( transform == null ) return point;
        ProjCoordinate srcCoord = new ProjCoordinate(point.longitude, point.latitude);
        ProjCoordinate dstCoord = new ProjCoordinate();
        transform.transform( srcCoord, dstCoord );
        return new LatLng(dstCoord.y,dstCoord.x);
    }
}
```

Βασικές πράξεις γεωγραφικών συνόρων

Ένωση συνόρων

```
public static List<LatLng> union(
    List<LatLng> p1, List<LatLng> p2
){
    Polygon polygon1 = convert(p1);
    Polygon polygon2 = convert(p2);
    if( polygon1 == null || polygon2 == null ) {
        return new ArrayList<>();
    }
    try{
        if( GeometryEngine.intersects(polygon1,polygon2) )
            return extractList(
                GeometryEngine.union(polygon1,polygon2)
            );
    }catch (Exception e){
        e.printStackTrace();
    }
    return new ArrayList<>();
}
```



Τομή συνόρων

```
public static List<List<LatLng>> intersections(
    List<LatLng> p1, List<LatLng> p2
){
    Polygon polygon1 = convert(p1);
    Polygon polygon2 = convert(p2);
    if( polygon1 == null || polygon2 == null ) {
        return new ArrayList<>();
    }
    try{
        if( GeometryEngine.intersects(polygon1,polygon2) )
            return extractLists(
                GeometryEngine.intersections(polygon1,polygon2)
            );
    }catch (Exception e){
        e.printStackTrace();
    }
    return new ArrayList<>();
}
```



Διαφορά συνόρων

```
public static List<List<LatLng>> differences(
    List<LatLng> p1, List<LatLng> p2
){
    Polygon polygon1 = convert(p1);
    Polygon polygon2 = convert(p2);
    if( polygon1 == null || polygon2 == null ) {
        return new ArrayList<>();
    }
    try{
        if( GeometryEngine.intersects(polygon1,polygon2) )
            return extractLists(
                GeometryEngine.difference(polygon1,polygon2)
            );
    }catch (Exception e){
        e.printStackTrace();
    }
    return new ArrayList<>();
}
```



Οι γεωγραφικές πράξεις συνόρων πραγματοποιήθηκαν με την βοήθεια της κλάσης GeometryEngine του ArcGIS Runtime API

Προβολή γεωγραφικών συνόρων στο Maps SDK

Σύνορα ως πολύγωνο

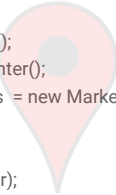
```
List<LatLng> border = getBorder();
List<List<LatLng>> holes = getHoles();
PolygonOptions options = new PolygonOptions();
options.addAll(border);
for(List<LatLng> hole : holes){
    if(hole.size()>0)
        options.addHole(hole);
}
Polygon polygon = googleMaps.addPolygon(options);
```

Σύνορα ως κύκλος

```
LatLng center = getCenter();
double radius = getRadius();
CircleOptions options = new CircleOptions();
options.center(center);
options.radius(radius);
options.position(center);
Circle circle = googleMaps.addCircle(options);
```

Σημείο ως marker

```
String title = getTitle();
String desc = getDesc();
LatLng center = getCenter();
MarkerOptions options = new MarkerOptions();
options.title(title);
options.snippet(desc);
options.position(center);
Marker marker = googleMaps.addMarker(options);
```



Σύνορα ως πολυγραμμικό διάνυσμα

```
List<LatLng> border = getBorder();
float width = getWidth();
PolylineOptions options = new PolylineOptions();
options.addAll(border);
options.width(width);
Polyline polyline = googleMaps.addPolyline(options);
```

Καθαρισμός σχήματος από το GIS.

Για καθορισμό συγκεκριμένων σχημάτων απο των χάρτη:

```
polygon.remove();
circle.remove();
marker.remove();
polyline.remove();
```

Καθαρισμός σχημάτων από το GIS.

Για καθορισμό όλων των σχημάτων απο των χάρτη:

```
googleMaps.clear();
```


Αλληλεπιδράσεις του χρήστη με το Maps SDK

Αλληλεπιδράσεις με το GIS

```
googleMaps.setOnMapClickListener( point -> {  
    //code...  
});
```

To LatLng point είναι το σημείο του χάρτη όπου ο χρήστης έκανε click στον GIS.

```
googleMaps.setOnMapLongClickListener( point -> {  
    //code...  
});
```

To LatLng point είναι το σημείο του χάρτη όπου ο χρήστης έκανε παρατεταμένο click στο GIS.

Αλληλεπιδράσεις με τα σύνορα

```
googleMaps.setOnPolygonClickListener( polygon -> {  
    //code...  
});
```

To Polygon polygon είναι το πολύγωνο που ο χρήστης έκανε click στο GIS.

```
googleMaps.setOnMarkerClickListener( marker -> {  
    //code...  
    return true;  
});
```

To Marker marker είναι το marker που ο χρήστης έκανε click στο GIS.

```
googleMaps.setOnCircleClickListener( circle -> {  
    //code...  
});
```

To Circle circle είναι ο κύκλος που ο χρήστης έκανε click στο GIS.

Τοπική Βάσης Δεδομένων με Room Database

Πίνακας Αγρών

```
@Entity(tableName = "Lands")
public class LandEntity{
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "ID")
    private long id;
    @ColumnInfo(name = "Title")
    private String title;
    @ColumnInfo(name = "Border")
    private final List<LatLng> border =
        new ArrayList<>();
    @ColumnInfo(name = "Holes")
    private final List<List<LatLng>> holes =
        new ArrayList<>();
    public LandEntity(
        long id, String title,
        List<LatLng> border,
        List<List<LatLng>> holes
    ){
        this.id = id; // if set to 0 it will auto generate
        this.title = title;
        if(border != null) this.border.addAll(border);
        if(holes != null) this.holes.addAll(holes);
    }
    //getters and setters
}
```

Dao Αγρών

```
@Dao
public interface LandDao {
    @Query(
        "SELECT * FROM Lands "
    )
    List<LandEntity> getLands();
    @Query(
        "SELECT * FROM Lands "+
        "WHERE ID = :id LIMIT 1"
    )
    LandEntity getLand(long id);
    @Insert(
        onConflict = OnConflictStrategy.REPLACE
    )
    long insert(LandEntity land);
    @Delete
    void delete(LandEntity land);
}
```

Κλάση βάσης δεδομένων

```
@Database(
    version = 1,
    entities = {
        LandEntity.class
    }
)
@TypeConverters({
    DBTypesConverter.class
})
public abstract class LocalDatabase
    extends RoomDatabase
{
    public abstract LandDao landDao();
}
```

Κλάση μετατροπής σύνθετων ιδιοτήτων

```
public class DBTypesConverter {
    @TypeConverter
    public static String borderToJson(
        List<LatLng> border
    ){
        return new Gson().toJson(border);
    }
    @TypeConverter
    public static List<LatLng> jsonToBorder(
        String border
    ){
        Type listType =
            new TypeToken<List<LatLng>>().getType();
        return new Gson().fromJson(border, listType);
    }
    //other custom attrs
}
```

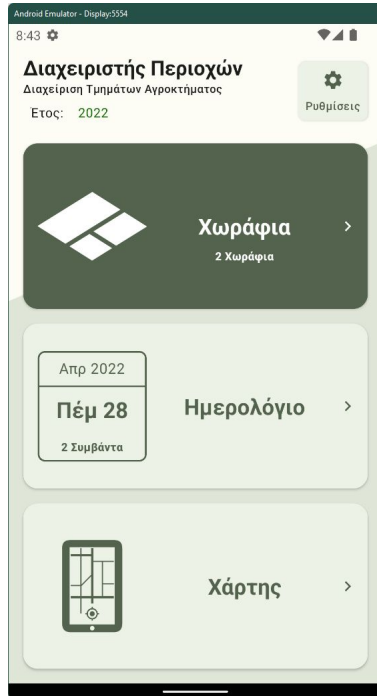
| Lands | |
|---------|---------|
| id | int |
| title | varchar |
| borders | json |
| holes | json |

Σύνοψη Παρουσίασης Εργασίας

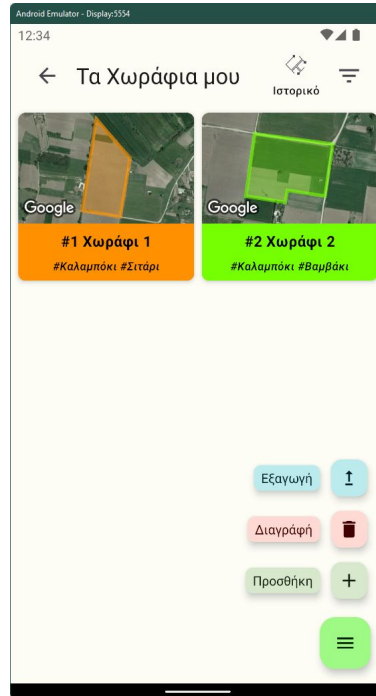
Μετά από αυτήν την παρουσίαση έχουμε μια πιο σαφή εικόνα για πως υλοποιήθηκαν οι εξής λειτουργίες:

- Την δομή μερικών βασικών γεωγραφικών αρχείων (KML, GML, GeoJSON).
- Πώς υλοποιήθηκε η μετατροπή μεταξύ συστημάτων συντεταγμένων.
- Την υλοποίηση των βασικών πράξεων γεωγραφικών συνόρων.
- Πώς προβάλουμε γεωγραφικά σύνορα στο Maps SDK.
- Πώς διαβάζουμε τις αλληλεπιδράσεις του χρήστη με το Maps SDK.
- Την υλοποίηση της τοπικής βάσης δεδομένων Android σε Room Database.

Διαφάνειες Εφαρμογής - Κύριο Μενού και Οθόνες Αγρών



Οθόνη Κύριου Μενού



Οθόνη Μενού Αγρών

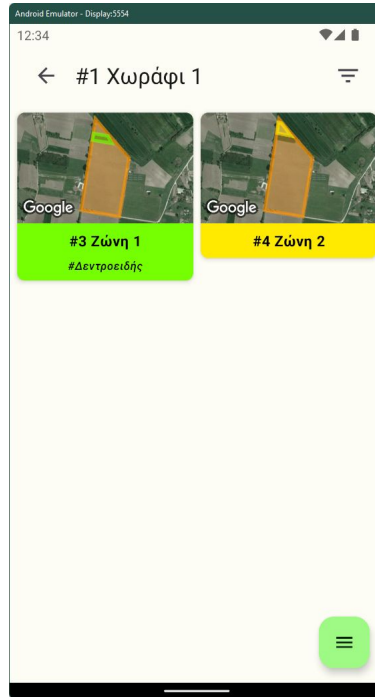


Οθόνη Συνόρων Αγρού

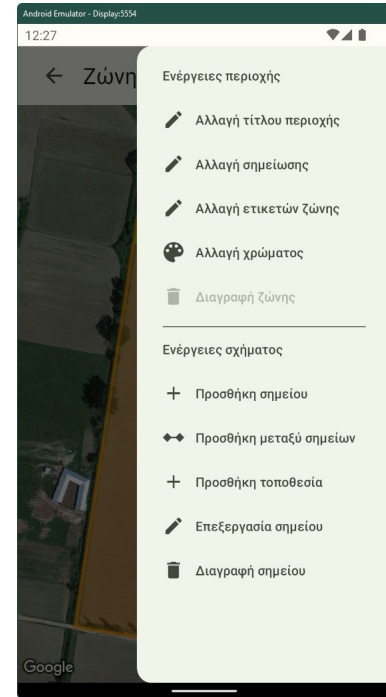
Διαφάνειες Εφαρμογής - Οθόνες Τμημάτων Αγρού



Οθόνη Προεπισκόπηση Αγρού

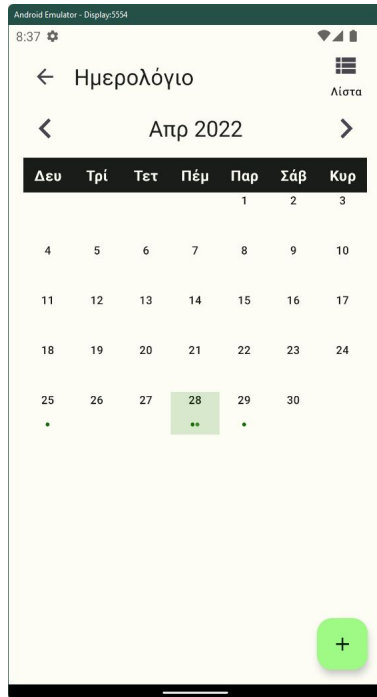


Οθόνη Μενού Τμημάτων Αγρού



Οθόνη Συνόρων Τμήματος Αγρού

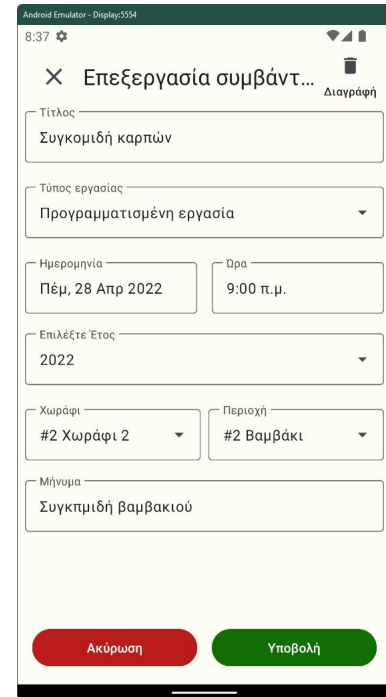
Διαφάνειες Εφαρμογής - Οθόνες Ημερολογίου



Οθόνη Ημερολογίου



Οθόνη Λίστας Εργασιών

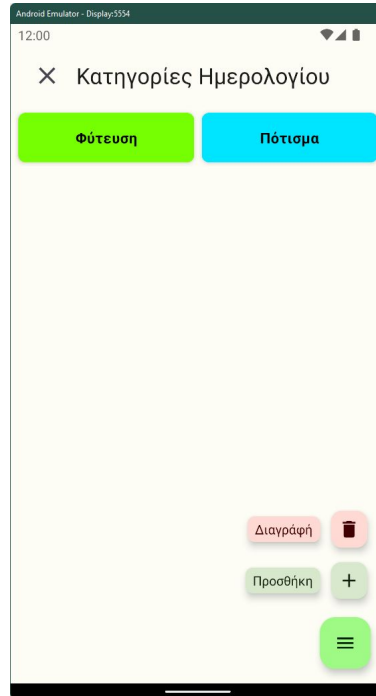


Οθόνη Εργασίας

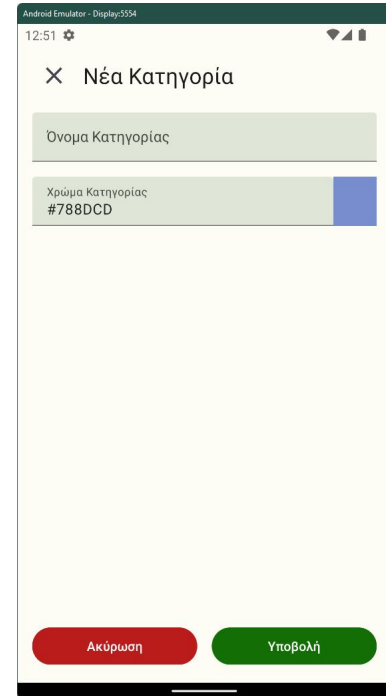
Διαφάνειες Εφαρμογής - Οθόνες Κατηγοριών Εργασίας



Οθόνη Ρυθμίσεων Εφαρμογής



Οθόνη Μενού Κατηγοριών Εργασίας

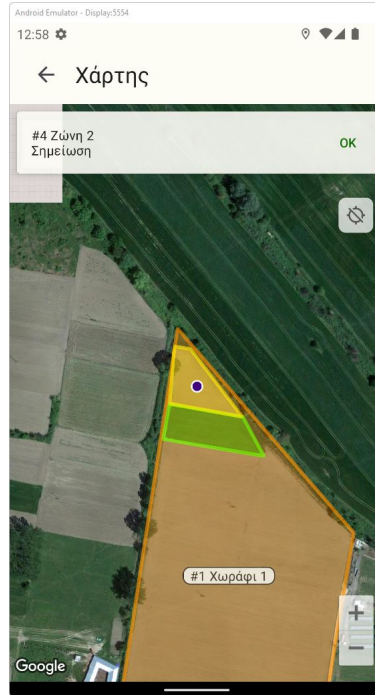


Οθόνη Κατηγορίας Εργασίας

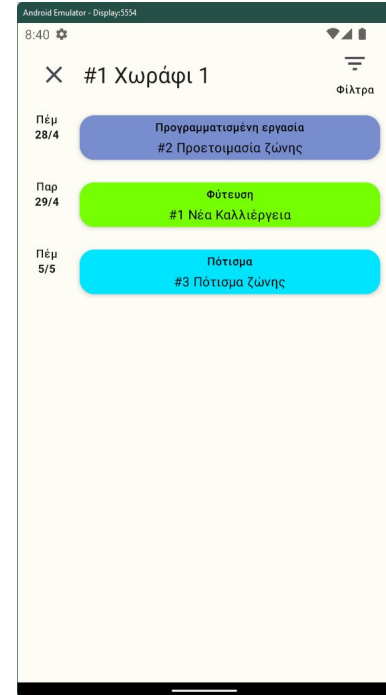
Διαφάνειες Εφαρμογής - Οθόνες Προεπισκόπησης Αγρών



Οθόνη Προεπισκόπησης Αγρών



Οθόνη Παρακολούθησης Τοποθεσίας



Οθόνη Λίστας Εργασιών Αγρού