Dream Vault

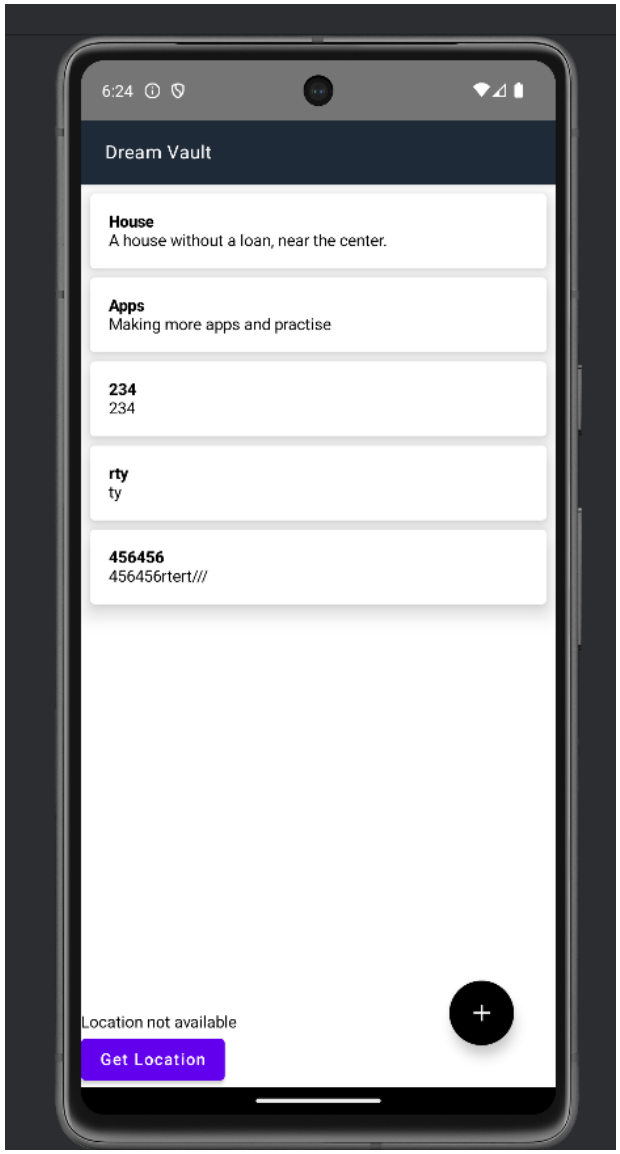
Το αποθετήριο των προσωπικών ευχών

Επισκόπηση

- Η εφαρμογή **Dream Vault** επιτρέπει στους χρήστες να αποθηκεύουν, να ενημερώνουν και να διαγράφουν "ευχές" (ή αντικείμενα που επιθυμούν να αποκτήσουν ή να πραγματοποιήσουν). Η εφαρμογή χρησιμοποιεί το **Jetpack Compose** για το UI και τη βάση δεδομένων **Room** για την αποθήκευση δεδομένων. Επιπλέον, αξιοποιεί τις υπηρεσίες τοποθεσίας για να παρέχει δεδομένα γεωγραφικής θέσης στους χρήστες.

Αρχική Οθόνη (Home Screen)

- **Σκοπός**: Παρουσιάζει τις καταχωρημένες ευχές του χρήστη με τη δυνατότητα αλληλεπίδρασης.
- **Διαμόρφωση**:
- **Πλαίσιο Εφαρμογής**: Εμφανίζεται ένας τίτλος, το "Dream Vault", με μία μπάρα πλοήγησης (AppBar) για την πλοήγηση.
- **Λειτουργία**: Ο χρήστης βλέπει όλες τις καταχωρημένες ευχές του σε μια **λίστα**. Η δυνατότητα να διαγράψει μια ευχή μέσω **swipe-to-dismiss** προσφέρει εύκολη αλληλεπίδραση.
- **Προσθήκη νέας ευχής**: Πατώντας το **Floating Action Button (FAB)** στην κάτω δεξιά γωνία, ο χρήστης μεταφέρεται στην οθόνη προσθήκης/επεξεργασίας για μια νέα ευχή.
- **Ληψη τοποθεσίας:** Αποκτά την τρέχουσα τοποθεσία του χρήστη και την εμφανίζει ως διεύθυνση μέσω της αντίστροφης γεωκωδικοποίησης.

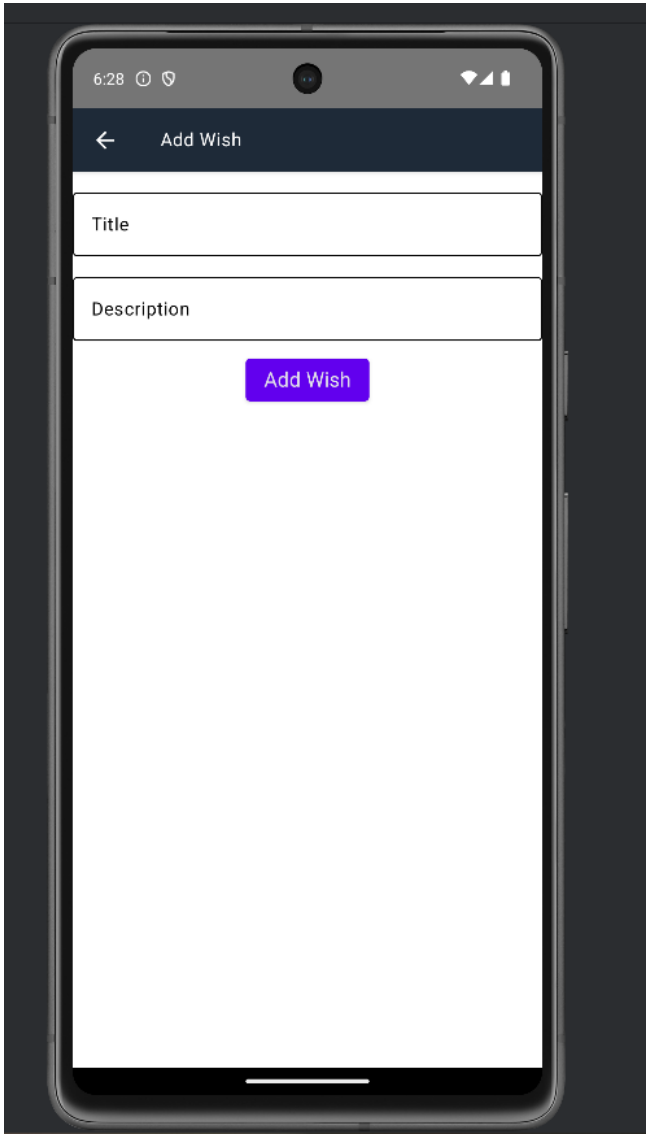



```
Scaffold(
    scaffoldState = scaffoldState,
    topBar = {AppBarView(title= "Dream Vault")},
    floatingActionButton = {
        FloatingActionButton(
            modifier = Modifier.padding(all = 20.dp),
            contentColor = Color.White,
            backgroundColor = Color.Black,
            onClick = {
                Toast.makeText(context, text: "FAButton Clicked", Toast.LENGTH_LONG).show()
                navController.navigate( route: Screen.AddScreen.route + "/0L")

            }) {
            Icon(imageVector = Icons.Default.Add, contentDescription = null)
        }
    }
```

Οθόνη Προσθήκης/Επεξεργασίας Ευχής (Add/Edit Screen)

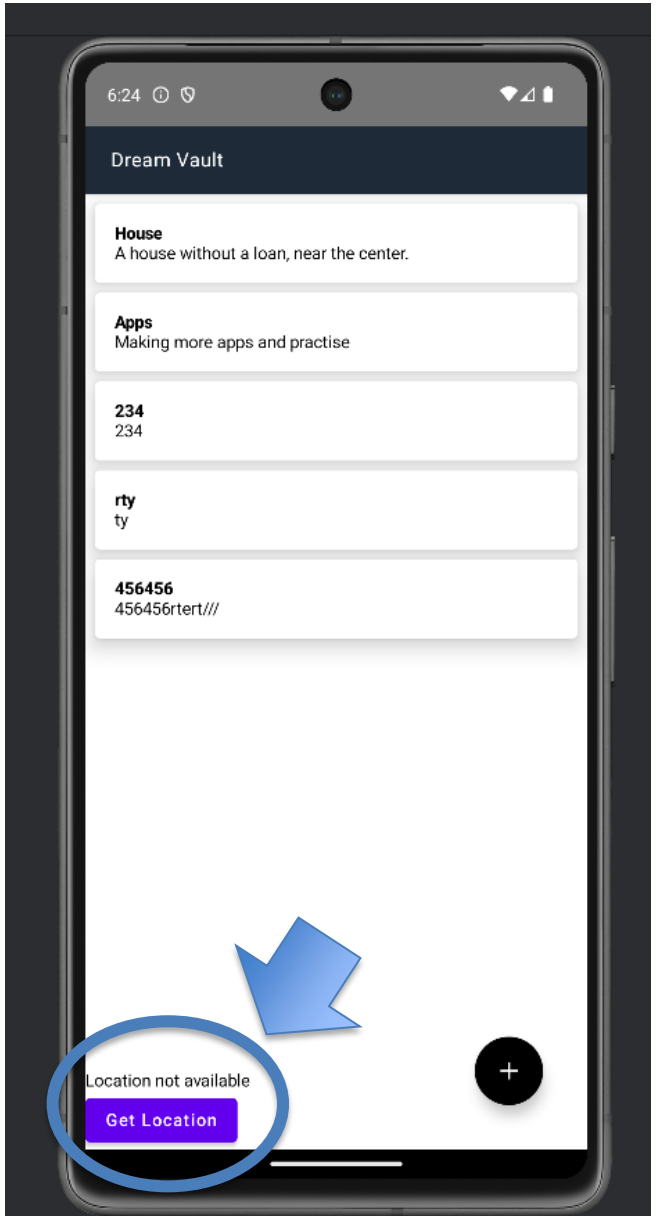
- **Σκοπός**: Επιτρέπει στον χρήστη να προσθέσει μια νέα ευχή ή να επεξεργαστεί μια υπάρχουσα.
- **Διαμόρφωση**:
- **Τίτλος & Περιγραφή**: Η οθόνη διαθέτει δύο βασικά πεδία εισαγωγής: **τίτλος** και **περιγραφή** της ευχής. Ο χρήστης μπορεί να πληκτρολογήσει ελεύθερα.
- **Λειτουργία**: Εδώ ο χρήστης μπορεί να καταχωρίσει ή να επεξεργαστεί την ευχή του. Πατώντας το κουμπί **Αποθήκευση**, οι αλλαγές αποθηκεύονται στην τοπική βάση δεδομένων (Room).




```
Spacer(modifier = Modifier.height(10.dp))
Button(onClick={
    if(viewModel.wishTitleState.isNotEmpty() &&
        viewModel.wishDescriptionState.isNotEmpty()){
        if(id != 0L){
            viewModel.updateWish(
                Wish(
                    id = id,
                    title = viewModel.wishTitleState.trim(),
                    description = viewModel.wishDescriptionState.trim()
                )
            )
        }else{
            //  AddWish
            viewModel.addWish(
                Wish(
                    title = viewModel.wishTitleState.trim(),
                    description = viewModel.wishDescriptionState.trim())
```

Οθόνη Τοποθεσίας (Location Screen)

- **Σκοπός**: Εμφανίζει την τοποθεσία του χρήστη (γεωγραφικές συντεταγμένες) και τη δυνατότητα εμφάνισης της διεύθυνσης μέσω geocoding.
- **Διαμόρφωση**:
- **Εμφάνιση Συντεταγμένων**: Παρουσιάζει τη γεωγραφική τοποθεσία (latitude, longitude) σε πραγματικό χρόνο.
- **Geocoding**: Χρησιμοποιώντας την υπηρεσία geocoder, η οθόνη εμφανίζει τη διεύθυνση της τοποθεσίας.
- **Λειτουργία**: Ο χρήστης μπορεί να αιτηθεί την τοποθεσία του πατώντας το κουμπί **Get Location**. Αν δεν έχει παραχωρηθεί η άδεια για την τοποθεσία, η εφαρμογή ζητά άδεια.

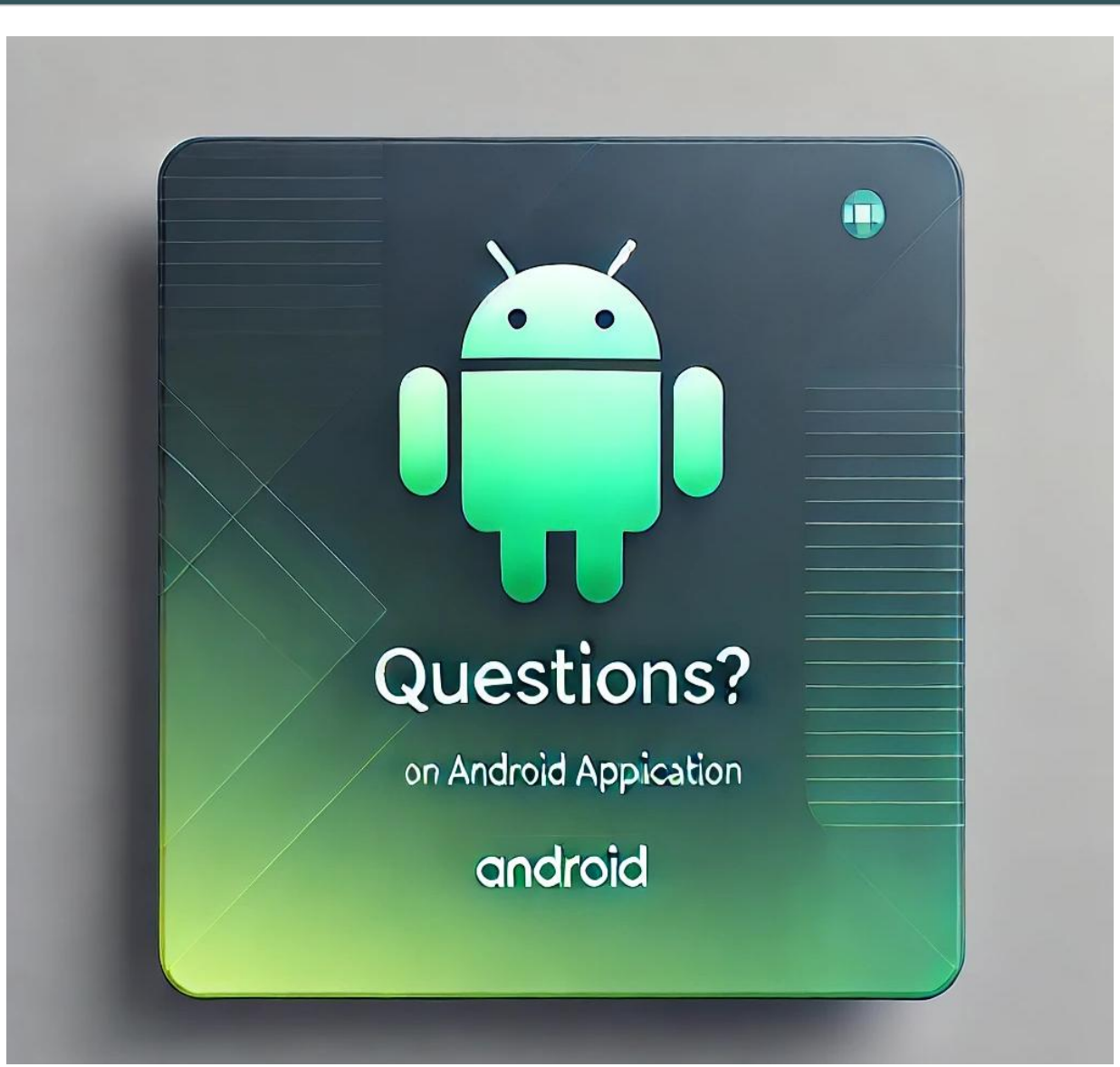



```
val requestPermissionLauncher = rememberLauncherForActivityResult(
    contract = ActivityResultContracts.RequestMultiplePermissions() ,
    onResult = { permissions ->
        if(permissions[Manifest.permission.ACCESS_COARSE_LOCATION] == true
            && permissions[Manifest.permission.ACCESS_FINE_LOCATION] == true){
            // I HAVE ACCESS to location

            locationUtils.requestLocationUpdates(viewModel = viewModel)
        }else{
            val rationaleRequired = ActivityCompat.shouldShowRequestPermissionRationale(
                context as MainActivity,
                Manifest.permission.ACCESS_FINE_LOCATION
            ) || ActivityCompat.shouldShowRequestPermissionRationale(
                context as MainActivity,
                Manifest.permission.ACCESS_COARSE_LOCATION
            )
```

Η Σχέση Μεταξύ των Οθονών και της Λειτουργικότητας

- **Πλοήγηση**: Χρησιμοποιούμε την **Navigation Component** του Jetpack για τη διαχείριση της πλοήγησης από την αρχική οθόνη στην οθόνη προσθήκης/επεξεργασίας και στην οθόνη τοποθεσίας.



Ερωτήσεις