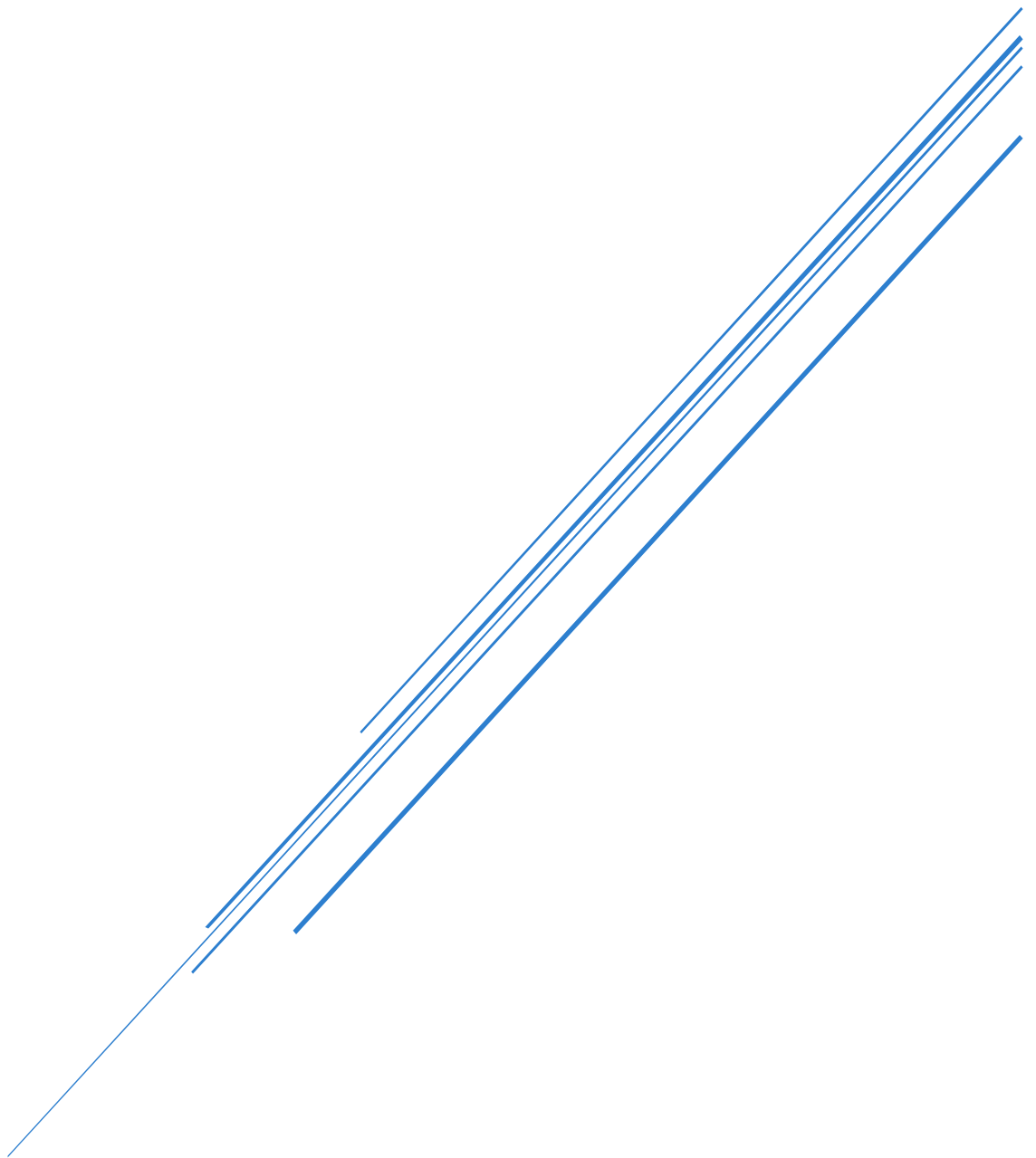


APLIKÁCIA PRE POZOROVANIE ZVIERAT

Semestrálna práca; Predmet VAMZ



**Adam Pangrác
5ZYS33**

1. Popis a analýza riešeného problému

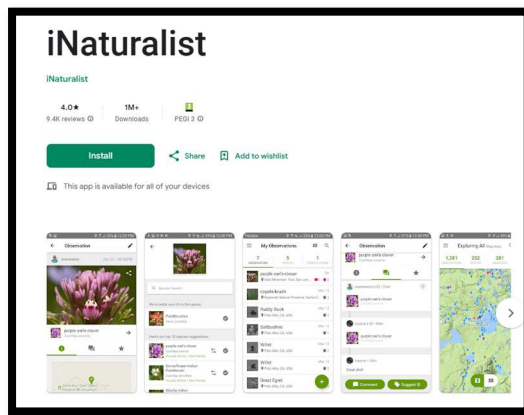
1.1. Špecifikácia zadania

- Zadaním tejto semestrálnej práce bolo vytvoriť mobilnú aplikáciu pre systém Android v jazyku Kotlin, za použitia Toolkitu Jetpack Compose UI.
- Aplikácia umožní používateľovi prezerať si záznamy pridané inými používateľmi a pridávať do systému nové záznamy.

1.2. Podobné aplikácie

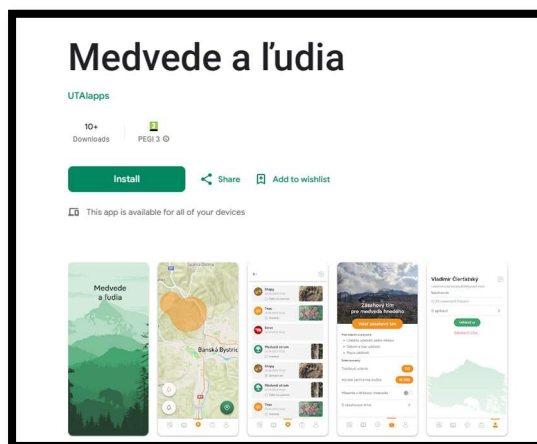
- **iNaturalist**

- Aplikácia iNaturalist je dostupná v Google Store pre android a je nápomocná pri identifikácii rastlín a zvierat v okolí.
- <https://play.google.com/store/apps/details?id=org.inaturalist.android&hl=en&pli=1>



-
- **Medvede a ľudia**

- Aplikácia vyvíjaná štátnou ochranou prírody SR
- Poskytuje informácie o bezpečnosti a možnostiach prevencie v územiach s výskytom medveďa hnedého a vkladať aktuálne údaje
- Dostupná na Apple zariadenia a už aj na Androide (Keď som robil checkpoint1 tak som ju v Google Play nenašiel)
- <https://apps.apple.com/sk/app/medvede-a-%C4%BEudia/id647443983>
- <https://play.google.com/store/apps/details?id=sk.utai.medvedealudia>



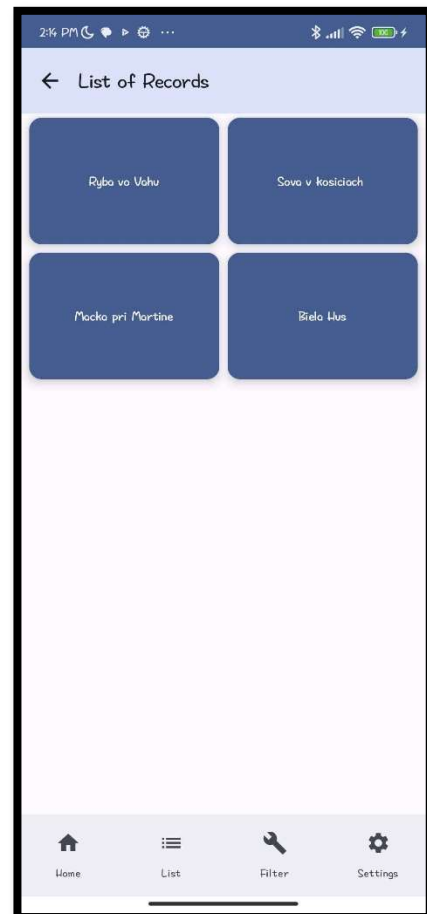
2. Návrh riešenia problému

2.1. Prípady použitia



Domovská obrazovka

- Obsahuje mapu so všetkými dostupnými záznamami, na ktoré keď sa klikne tak sa prepne na obrazovku s detailmi o jednotlivom zázname
- Obsahuje Floating button, ktorý prepne na obrazovku s možnosťou pridania nového záznamu



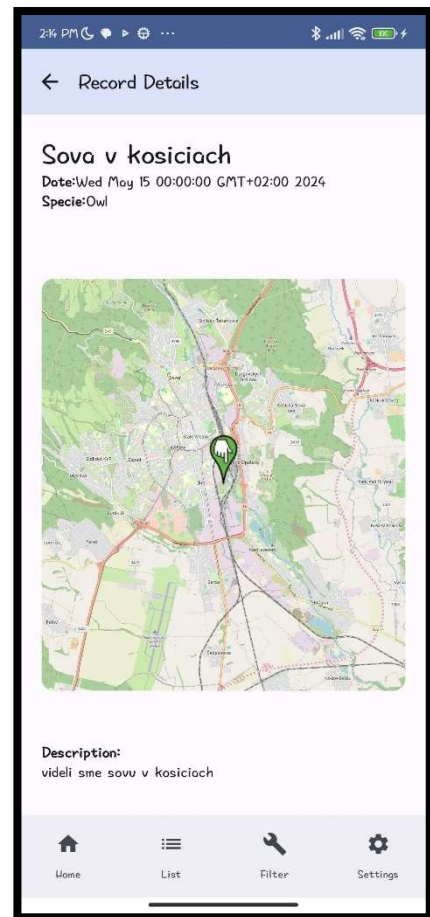
Obrazovka s Listom záznamov

- Obsahuje list všetkých dostupných záznamov zobrazený v mriežke
- Po kliknutí na tlačidlo konkrétneho záznamu sa prejde obrazovku s detailmi o danom zázname



Obrazovka pre pridávanie záznamov

- Obsahuje textové polia pre zadanie názvu záznamu a popisu záznamu
- Obsahuje mapu, na ktorej je možné ťuknutím vybrať polohu nového záznamu
- Obsahuje Dropdown menu, kde je možné vybrať druh zvierat'a (To na tomto obrázku nevidno)

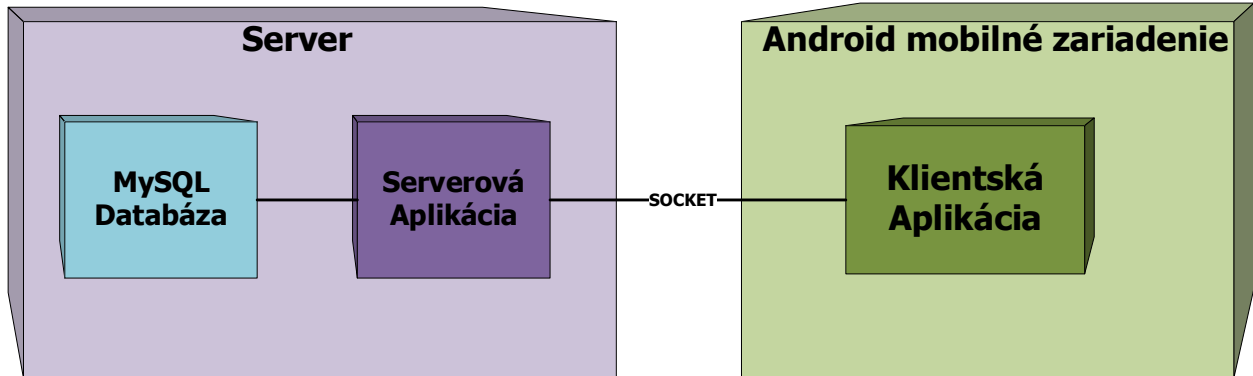


Obrazovka s detailmi o zázname

- Obsahuje všetky dostupné informácie o danom zázname:
 - Čas pridania
 - Druh zvierat'a
 - Názov záznamu
 - Popis záznamu
 - Mapa s markerom, ktorý určuje polohu markeru

3. Návrh aplikácie

- Aplikácia je plne funkčná len s pripojením na server, na ktorom beží potrebná aplikácia a MySQL databáza s potrebnými tabuľkami a procedúrami
- Architektúra systému:



- Aplikácia komunikuje so serverovou aplikáciou pomocou socketov
- Príklad na ukážke kódu:

```
override suspend fun getSpecies(): List<Specie> = withContext(Dispatchers.IO) {  
    var navrat: MutableList<Specie> = mutableListOf()  
  
    val socket = Socket(serverIP, 55557)  
    socket.setSoTimeout(100)  
    val writer = PrintWriter(socket.getOutputStream(), true)  
    val reader = BufferedReader(InputStreamReader(socket.getInputStream()))  
  
    /////////// Send request message to server  
    val newMessage = Message("Requesting all species", Calendar.getInstance().time, "Android client",  
MsgType.RequestAllSpecies)  
    writer.println(newMessage.toJsonString())  
  
    var response: String? = null  
    do {  
        try {  
            response = reader.readLine()  
        } catch (e: Exception) {  
            break  
        }  
    }  
    if (response != null) {  
        val message = Gson().fromJson(response, Message::class.java)  
        //val text = message.Text  
        val values = message.Text.split(";").toTypedArray()  
  
        val id = values[0].toInt()  
        val name = values[1]  
        val specie = Specie(id, name)  
        navrat.add(specie)  
    }  
} while (response != null)  
//Log.d(TAG, "RECEIVED: ${message.Text}")  
  
writer.close()  
reader.close()  
socket.close()  
  
return@withContext navrat  
}
```

- Táto ukážka popisuje získanie zoznamu druhov zvierat z databázy na servery
 - Klient pošle správu na server (newMessage), v ktorej je v tomto prípade významný iba MsgType, čo je enum ktorý hovorí o aký typ správy sa jedná.

- Server túto správu prijme, spracuje, získa z databázy žiadané údaje a pošle ich klientovi.
- Metóda vráti list
- **Mapa**
 - Pre vykresľovanie mapy a markerov som použil OpenStreetMap, lebo som zistil, že Google si za použitie API svojich Google Maps pýta peniaze
 - Musel som pri tom použiť AndroidView, čo sa robí pri používaní prvkov, ktoré nie sú dostupné v Compose (viac tu: <https://developer.android.com/develop/ui/compose/migrate/interoperability-apis/views-in-compose>)
 - Príklad použitia:

```
@Composable
fun OsmMap(
    markers: List<MapMarker>,
    navController: NavHostController,
    mapViewModel: MapViewModel,
    modifier: Modifier = Modifier,
) {
    val currentLat by remember {
        mapViewModel.currentCentreLat
    }
    var currentLon by remember {
        mapViewModel.currentCentreLon
    }
    mapViewModel.mapView = MapView(LocalContext.current)

    Column(modifier = modifier.fillMaxSize()) {
        AndroidView(
            modifier = Modifier.weight(1f),
            factory = { _ ->
                mapViewModel.mapView?.getMapCenter()
                mapViewModel.mapView?.apply {

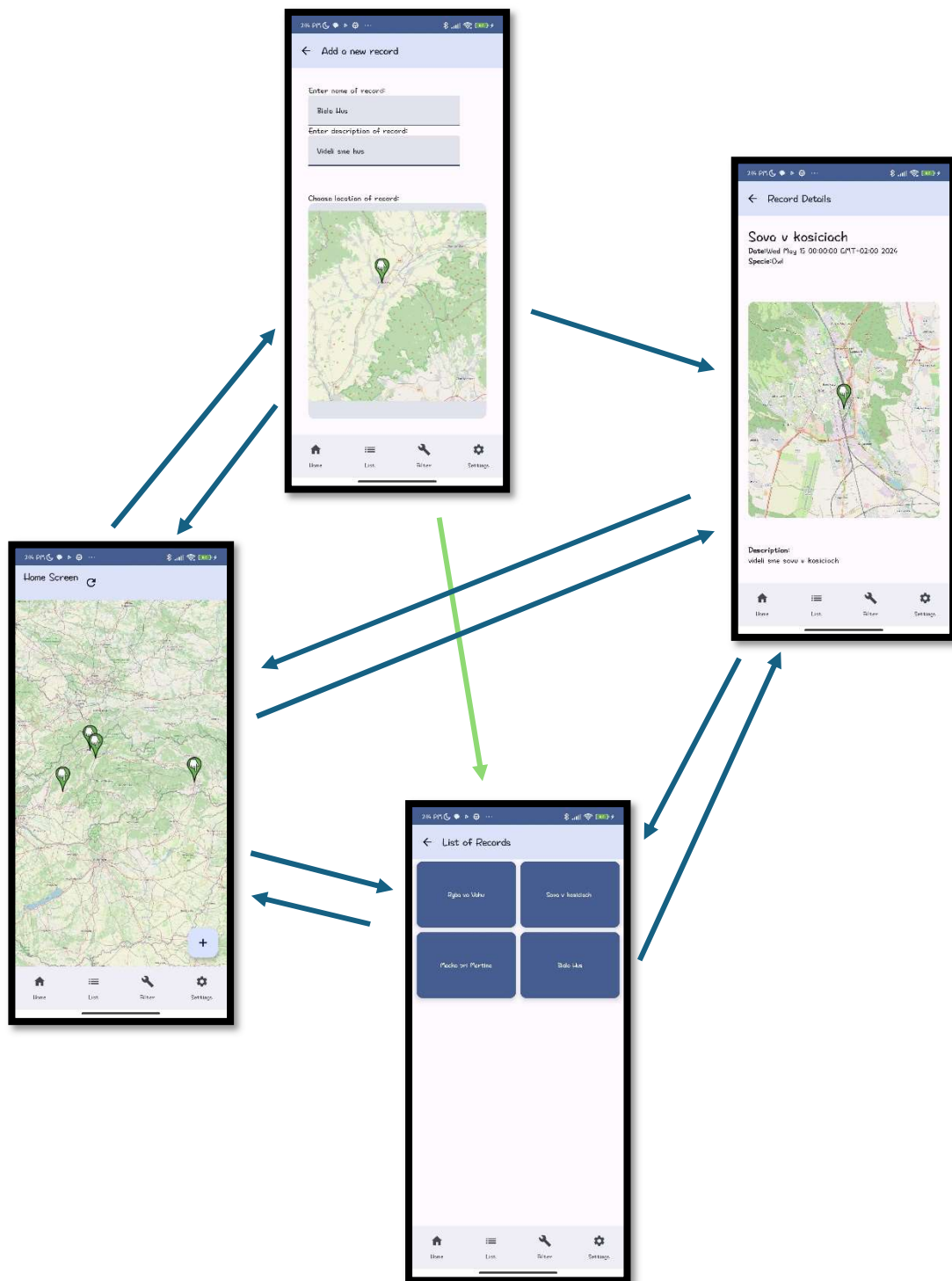
                    setMultiTouchControls(true)
                    setBuiltInZoomControls(false)
                    setTileSource(TileSourceFactory.MAPNIK)
                    setOnClickListener { }

                }!!
            },
            update = { view ->
                val geoPoint = GeoPoint(currentLat, currentLon)
                view.controller.setCenter(geoPoint)
                view.controller.setZoom(9)
                markers.forEach {
                    val newMarker = Marker(mapViewModel.mapView)
                    val position = GeoPoint(it.getLatitude(), it.getLongitude())
                    newMarker.position = position

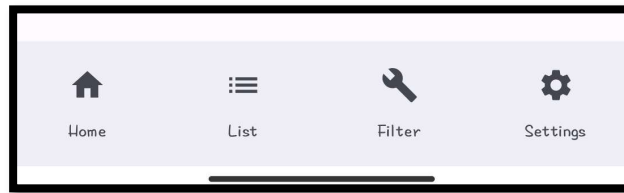
                    newMarker.setOnMarkerClickListener { marker, _ ->
                        Log.d(TAG, "CLICKED: ${it.getLabel()}")
                        navController.navigate("${AppScreen.RecordDetails.name}/${it.getID()}")
                        true
                    }
                    mapViewModel.mapView?.overlays?.add(newMarker)
                }
            }
        )
    }
}
```

- Update lambda
 - Používa sa na aktualizáciu MapView pri zmene dát
- Factory lambda
 - Používa sa pri inicializácii a konfigurácii View

▪ Navigácia v aplikácii



- Prepínanie obrazoviek prebieha v rámci jedného Scaffoldu, kde sa v spodnej lište nachádzajú navigačné tlačidlá.

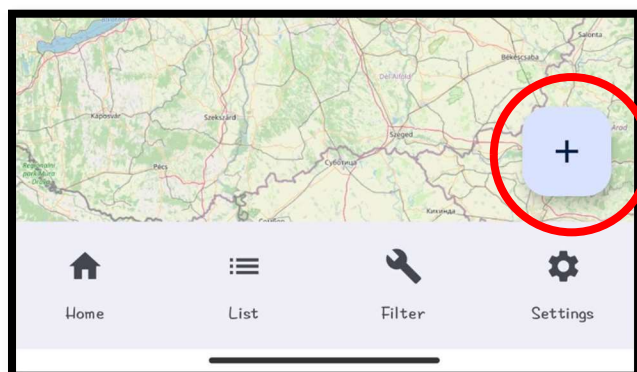


```
bottomBar = {
  NavigationBar {
    val selectedItem = null
    NavigationBarItem(
      icon = { Icon(Icons.Filled.Home, contentDescription = "A") },
      label = { Text(stringResource(R.string.home)) },
      selected = selectedItem == "B",
      onClick = {
        if (currentScreen != AppScreen.Home) {
          navController.popBackStack(AppScreen.Home.name, inclusive = false)
          currentScreen = AppScreen.valueOf(
            backStackEntry?.destination?.route?.split("/")?.get(0) ?:
AppScreen.Home.name
          )
        }
      }
    )
  }
}
```

- Pri kliknutí na daný prvok sa pomocou navControlleru mení obrazovka

```
NavHost(
  navController = navController,
  startDestination = AppScreen.Home.name,
  modifier = Modifier
    .fillMaxSize()
    .padding(innerPadding)
) {
  composable(route = AppScreen.Home.name) {
    HomeScreen(
      mapViewModel = mapViewModel,
      navController,
      retryAction = mapViewModel::getMapMarkers,
      modifier = Modifier.fillMaxSize()
    )
  }
}
```

- V NavHoste je definovaný navigačný graf
- **Navigácia do okna pre vytvorenie nového záznamu**
 - Pre navigáciu do obrazovky ktorá poskytuje možnosť pridať nový záznam sa dá dostať kliknutím na floating button ktorý sa zobrazuje iba na Domovskej obrazovke.



4. Príprava systému pre spustenie

- Odkazy:
 - Repozitár Klientskej aplikácie na githube:
 - <https://github.com/adapan005/AnimalObservingApp>
 - Repozitár Serverovej aplikácie na githube:
 - <https://github.com/adapan005/U8-Klient-Server>
- Pred spustením treba správne nastaviť IP Adresy a Port v rámci aplikácií, a to nasledovne:
 - V Klientskej Android aplikácii:
 - V súbore AppContainer.kt, treba nastaviť adresu servera na ktorom beží serverová aplikácia
 - V Serverovej aplikácii sa treba uistiť, že žiadna iná aplikácia nevyužíva port **55557**
- **Dôležité:**
 - Taktiež treba pripraviť MySQL databázu na localhoste port 4723; databáza s názvom animals, používateľ root a heslo Heslo123 (Prípadne sa dá zmeniť v AnimalObservingServer.cs)
 - MySQL príkazy pre vytvorenie potrebných tabuliek, procedúr atď. sa pokúsim pridať k repozitáru so serverovou aplikáciou, dopredu sa ospravedlňujem ak tam nie je.

5. Možné vylepšenia Android aplikácie do budúcnosti

- Dorobenie možnosti filtrovania záznamov podľa druhu zvierťa
- Pridať možnosť pridávať a zobrazovať obrázky k záznamom
- Dorobiť obrazovku s nastaveniami, kde sa bude dať zmeniť na aký server sa aplikácia pripája

6. Záver

- Myslím si, že Aplikácia v momentálnom stave poskytuje minimálnu funkčnosť, no tiež by bolo vhodné časti kódu prerobiť, a dokončiť vyššie spomenuté možné vylepšenia.
- Pri práci na tejto aplikácii mňa osobne najviac zaujala práca a posielanie dát cez sockety.
- Pri vytváraní aplikácie som sa tiež inšpiroval android codelabmy, ktoré sme robili na cvičeniach.

7. Zoznam použitých zdrojov

- Okrem code-labov a dokumentácie, som tiež získaval informácie z článkov na stránke <https://medium.com/> a otázok na stack overflow, ako napríklad táto: <https://stackoverflow.com/questions/77297775/create-a-map-using-openstreetmap-with-jetpack-compose-in-kotlin-programming-lang>