

# **OU3 – Valfri applikation**

**5DV209**

**Jakob Fridesjö**

[tfy17jfo@cs.umu.se](mailto:tfy17jfo@cs.umu.se)

[jafr0025@student.umu.se](mailto:jafr0025@student.umu.se)

**2021-09-03**

**version 2**

## Innehållsförteckning

1	Introduktion .....	3
2.	Applikationens målgrupp .....	3
3.	Säkerhet och etik .....	3
4.	Användarhandledning .....	4
5.	Tekniska detaljer.....	5
6.	Reflektion och diskussion.....	7

## **1. Introduktion**

Denna rapport behandlar en egenutvecklad applikation för Android som är utvecklad i Kotlin och fungerar på Android 5.1 och senare.

Rapporten behandlar ämnen som målgrupp, etik, säkerhet, användning samt applikationens tekniska aspekter.

En app utvecklad för att hjälpa till ute i naturen, detta genom att tillhandahålla verktyg för navigation, anteckningar och diverse annan funktionalitet, allt detta utan internetuppkoppling, reklam eller spårning.

Uppgiften är ämnad för godkänt betyg.

## **2. Applikationens målgrupp**

Den målgrupp som applikationen främst är tänkt till är människor som är ute i naturen och vill använda telefonen som hjälpreda utan att behöva att nödvändigtvis vara uppkopplade.

Applikationen är även praktisk till många andra användningsområden utanför detta då de verktyg som ingår även är användbara i andra situationer.

Applikationen är tänkt att lösa en knepig situation där folk vill vara ute i naturen och känna sig nedkopplade, samtidigt som dom vill kunna använda telefonen som hjälpreda och verktyg.

Därför är applikationen tänkt att användas utan nätverk och är optimerad så att sensorer endast används då dom behövs och inga tunga beräkningar utförs i onödan, detta innebär att

applikationen drar mindre batteri samtidigt som användaren kan slå av mobildata och Wifi för att spara ytterligare batteri och även undvika distraktioner.

## **3. Säkerhet och etik**

Då applikationen inte använder sig av mobildata eller Wifi så är detta en relativt bra början från ett säkerhets och etiskt perspektiv. Applikationen använder sig inte heller av Googles tjänster eller liknande alternativ, detta innebär att applikationen i sig inte heller förmedlar vidare data till dessa samtidigt som den blir kompatibel med fler enheter, exempelvis Huawei eller ren AOSP.

Moderna telefoner använder satellit och nätverksbaserade platstjänster för att bestämma position. Telefonen skickar inte upp någon data till de satellitbaserade systemen så därför är de i sig

anonyma. Men detta innebär inte att platser är helt anonyma då applikationen använder sig av platstjänster och platser kan fångas in av Googles tjänster även om applikationen inte aktivt delar med sig av platser.<sup>1</sup>

Det går att lagra känsliga data i anteckningar, men dessa är lätta att ta bort antingen inne i applikationen eller genom att rensa applikationens data, något som då tar bort den lagrade datan i databasen, inte heller dessa data delas på något sätt vidare av applikationen.

#### **4. Användarhandledning**

Applikationen distribueras som en signerad APK-fil, denna kan installeras genom att öppnas med en filhanterare i Android.

Gränssnittet är uppdelat i de tre huvuddelarna Notes, Navigation och Utilities. Notes implementerar funktionalitet för att göra anteckningar. Att trycka på knappen i nedre högra hörn visar en vy där användaren kan skriva en ny anteckning. Om användaren trycker på en existerande anteckning så redigeras denna. Slutligen så kan användaren hålla inne på en anteckning varvid en dialog som frågar om användaren vill ta bort anteckningen visas.

Vyn för navigation visar koordinater överst samt kompass och nivåverktyg i mitten. I nedre högra hörn så finns en knapp som kan tryckas för att hämta koordinaterna för den plats användaren befinner sig på.

Slutligen så finns det en vy för övriga verktyg kallad *Utilities*. I denna vy så kan alarm och ficklampa användas genom att trycka på deras respektive knappar.

Navigation mellan olika vyer sker genom att använda navigationsfältet i den nedre delen av skärmen.

---

1 Google. Privacy & Terms. Google. <https://policies.google.com/technologies/location-data?hl=en-US> (Hämtad 2021-09-03)

## 5. Tekniska detaljer

Applikationen är programmerad i Kotlin för Android 5.1 och senare. Det designmönster som använts är främst MVVM, Model, View, ViewModel. Detta syftar att separera modell från data som endast behövs för menyn samtidigt som användandet av en ViewModel förenklar att spara av tillstånd för vyn, vilket gör det enklare att åstadkomma att aktiviteter kan stängas ner utan att viktiga tillstånd förloras.

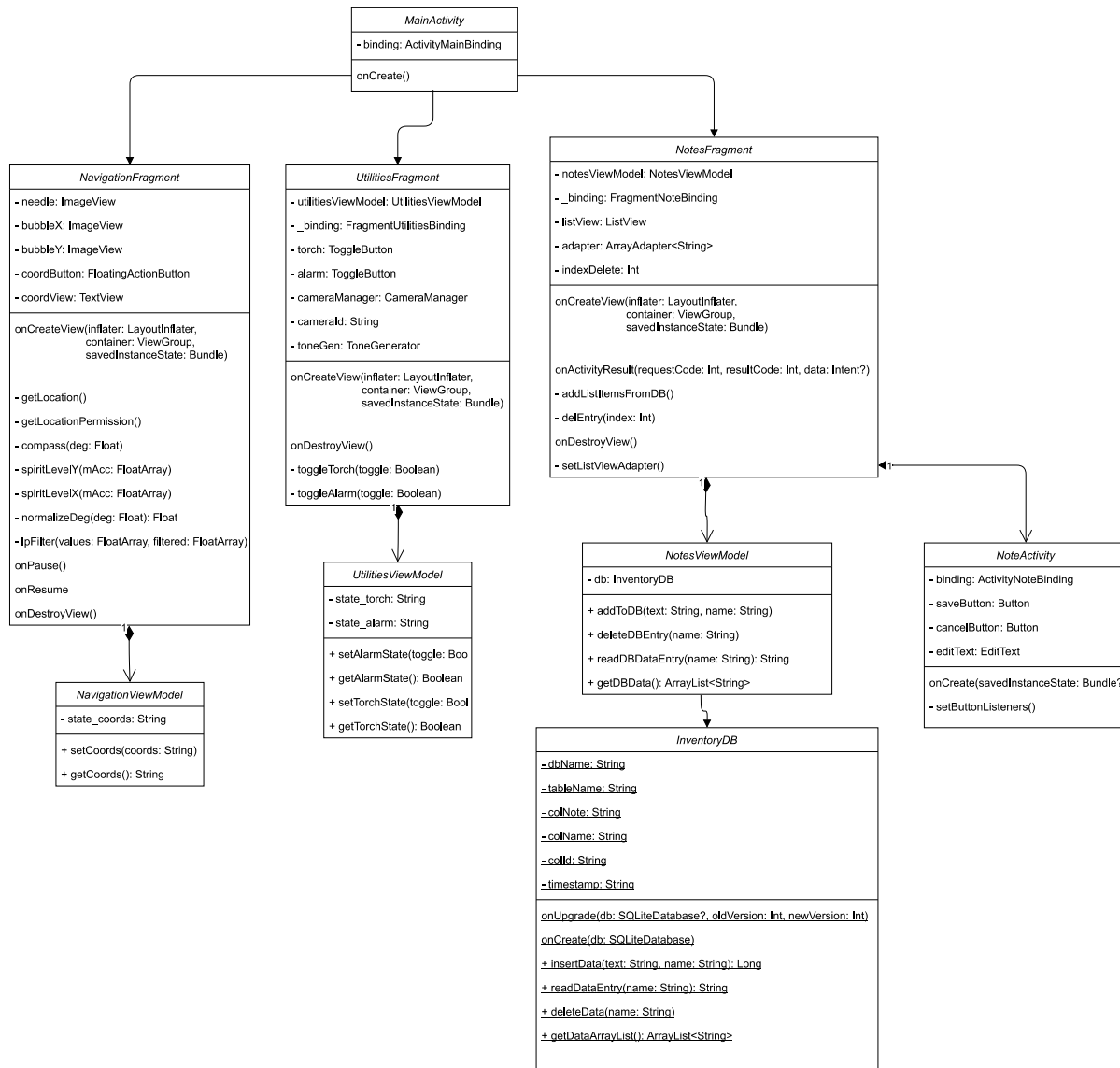
Exempelvis kan användaren byta till andra appar, slå av skärmen eller rotera telefonen utan att viktiga data förloras. UML-diagram för applikationen kan ses i figur 1.

Sensorer som hanteras är magnetometer, accelerometer och positionering. Utöver sensorerna används även ficklampa genom api till kamera, samt högtalare för alarm.

Magnetometer, accelerometer, positionering kräver alla speciella rättigheter som användaren måste godkänna för att använda dessa funktioner i applikationen.

Applikationen använder även en SQLite databas för att spara anteckningar som användaren gör. Denna databas gör så att applikationen kan spara anteckningar trots att applikationen stängs ner, dessa anteckningar sparas endast för applikationen och tas därför bort ifall applikationens data rensas.

Klassen *NotesViewModel* är kopplad till databas, klassen *NavigationFragment* använder platstjänster, magnetometer och accelerometer, klassen *UtilitiesFragment* använder Androids *ToneGenerator* för alarm och kamera api för ficklampa.



**Figur 1:** UML-diagram över applikationens arkitektur

## **6. Reflektion och diskussion**

Det som tog längst tid att implementera var anteckningsdelen av applikationen då denna krävde både aktiviteter och fragment samtidigt som kommunikation med databas krävdes för att spara gjorda anteckningar.

Att implementera kompass kräver även många beräkningar för att kompensera hur magnetometern påverkas av vilken vinkel som telefonen är i. Därför användes även accelerometer för att korrigera detta så bra som möjligt för bästa resultat.

Ytterligare något som tog tid att få till var navigation i botten av app. Detta var problematiskt eftersom att tillstånd förlorades varje gång navigation skedde trots att ViewModel användes. Lösning blev slutligen att använda alfaversioner av de paket som behövdes till denna komponent, vilket direkt löste problemet då en ViewModel användes för att spara tillstånd.