

Uppgift 3: GarageLog

5DV209 - Utveckling av mobila applikationer

Jonatan Wincent
c22jwt@cs.umu.se

30 september 2025

Innehåll

1	Introduktion	1
2	Etik och säkerhet	1
3	Användarmanual	2
3.1	Garage	2
3.2	Fordon	3
3.3	Påminnelser	4
3.4	Servicehistorik	5
4	Applikationens uppbyggnad	6
5	Reflektion	7

1 Introduktion

GarageLog är en applikation som hjälper dig att hålla koll på dina fordon, vilket arbete som har utförts på dem och när det är dags för nästa service.

Målgruppen för applikationen är fordonsentusiaster som tar hand om service och underhåll av sina fordon själva, men kan användas av alla som har ett fordon eller något annat som kräver regelbundet underhåll.

En kort Google Play Butik beskrivning följer nedan:

GarageLog är för dig som enkelt vill hålla koll på alla dina fordon på ett och samma ställe. Spara servicehistorik och viktiga anteckningar samt ställ påminnelser för kommande underhåll eller andra händelser. Oavsett om du är entusiast eller bara vill sköta det regelbundna underhållet gör GarageLog det enkelt att hålla varje fordon i toppskick.

GarageLog är tillgängligt som ett öppet källkodsprojekt på <https://github.com/jonwin1/GarageLog>

Jag siktar på betyget VG.

2 Etik och säkerhet

I sin helhet finns det få säkerhetsmässiga och etiska problem eftersom all data endast lagras i appens interna lagringsutrymme där ingen annan ska kunna komma åt det. De enda data som lagras som skulle kunna anses känsligt är registreringsnumren, eftersom detta är en personuppgift, men de är aldrig åtkomliga utanför applikationen och användaren kan välja att inte ange ett riktigt registreringsnummer. Applikationen använder aldrig nätverk och därför finns det inga sårbarheter i den aspekten.

Det finns möjlighet både att ta nya bilder och att välja befintliga bilder att visa för fordon, detta hanteras genom intents som gör att applikationen aldrig får tillgång till mer än de bilder som användaren väljer. Detta förhindrar även spionering på användaren via kameran eftersom applikationen endast får de bilder som skickas tillbaka från kameraappen och aldrig har tillgång till kameran direkt.

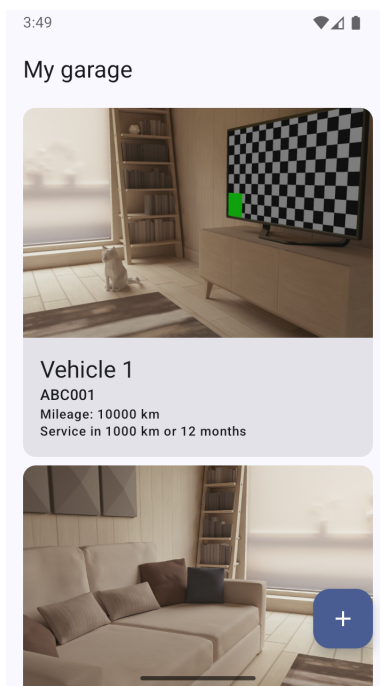
Angående de etiska aspekterna finns det inga problem som jag kan komma på. Ingen kan komma till skada av att använda applikationen och ingen

behandlas orättvist av den. Applikationen bidrar till "the common good" genom att vara ett användbart verktyg som vem som helst kan använda.

3 Användarmanual

Denna sektion beskriver hur GarageLog används och förklarar all funktionalitet i applikationen. Nedanstående undersektioner handlar om en del av applikationen var och förklarar all funktionalitet kopplat till den delen.

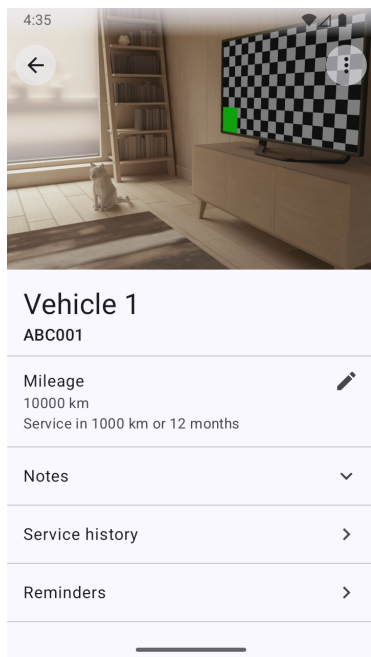
3.1 Garage



Figur 1: Förstasidan i applikationen, visar en lista med användarens fordon.

Det första som man möts av när man öppnar applikationen är garaget som innehåller en lista av alla fordon som man har i sitt garage, se Figur 1. Varje fordon representeras av ett kort i listan och har en bild av fordonet, dess namn, registreringsnummer, miltal och hur långt det är kvar till nästa servicepåminnelse. Längst ned i högra hörnet finns en knapp med en plussymbol för att lägga till nya fordon.

3.2 Fordon



Figur 2: Sidan för ett fordon, visar information om fordonet och knappar för att navigera vidare.



Figur 3: Sidan för att redigera ett fordon med textfält och en bild som kan ändras genom att trycka på den.

Om man trycker på ett av fordonskorterna i garagevyn navigeras man till sidan för det fordonet, se Figur 2. Från denna sida kan man komma åt all information om fordonet. Man kan se hur långt fordonet har gått och hur långt det är kvar till nästa servicepåminnelse. Trycker man på miltalesrutan dyker det upp en dialog där man kan uppdatera hur långt fordonet har gått. När man trycker OK i dialogen kontrolleras att det nya miltalet är högre än det gamla och om det inte är det frågar applikationen om man är säker på att man vill fortsätta, så att man inte råkar ange fel.

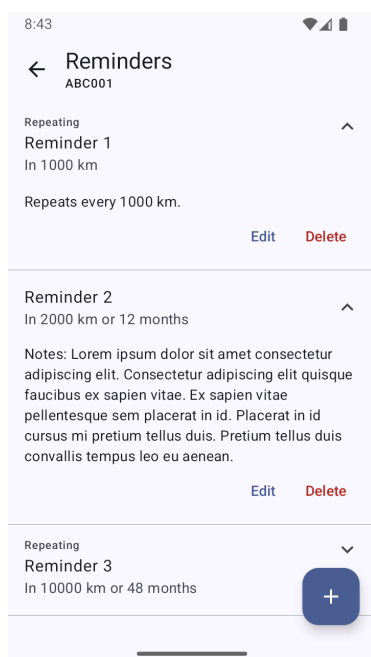
Nedanför detta finns ett fält för övriga anteckningar om fordonet som går att expandera genom att trycka på det för att visa anteckningarna. Längst ned finns knappar för att navigera vidare till servicehistoriken och servicepåminnelser för fordonet.

Till sist finns en knapp i övre högra hörnet som öppnar en meny där man kan välja att antingen redigera eller ta bort fordonet. Det fanns även planer för att lägga till knappar här för att exportera eller skriva ut all information om ett fordon vid eventuell försäljning eller liknande, men detta har inte hunnits med.

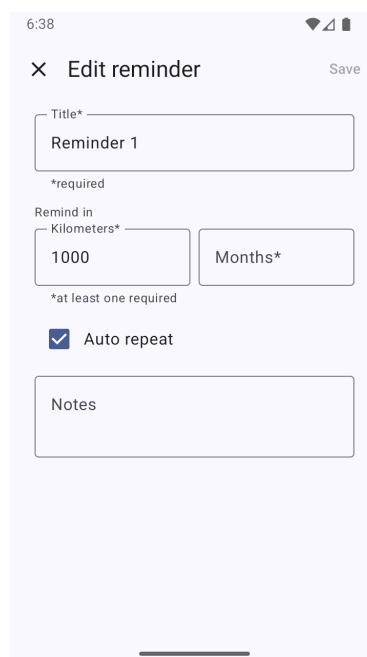
Väljer man att redigera ett fordon eller skapa ett nytt fordon så tas man till en skärm som ser ut som i Figur 3. I figuren är fältet för registreringsnummer avaktiverat eftersom det inte går att ändra i efterhand. Två av fälten, namn och registreringsnummer, måste anges för att kunna spara annars blir de röda med en varningsikon och spara-knappen avaktiveras. Övriga fält kan lämnas tomma om man vill. Trycker man på bilden längst upp dyker det upp en meny där man kan välja att ta en ny bild med kameraappen eller välja en bild ur sitt galleri, man kan även välja att ta bort bilden om en bild har valts.

När man har gjort sina ändringar så kan man spara med knappen längst upp till höger och för att man inte ska råka slänga bort sina ändringar så dyker det upp en dialogruta som frågar om man är säker på att man vill slänga ändringarna om man försöker gå tillbaka utan att spara. Spara-knappen avaktiveras om det av någon anledning inte går att spara. Detta skulle kunna vara för att inga ändringar har gjorts, något av de nödvändiga fälten är tomma eller att registreringsnumret inte är unikt.

3.3 Påminnelser



Figur 4:



Figur 5:

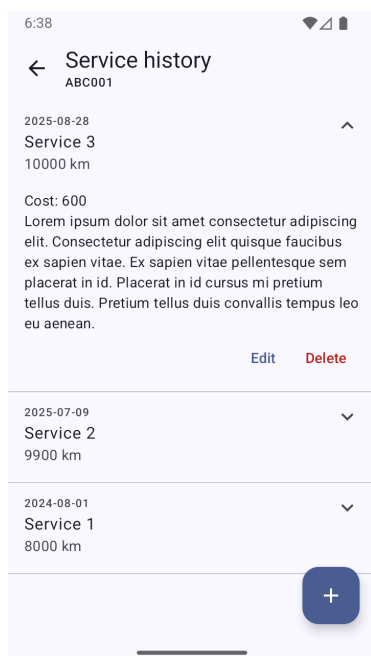
I påminnelsevyn finns en lista med alla påminnelser för det valda fordonet. Varje listobjekt går att expandera för att visa mer information som anteck-

ningar och knappar för att hantera påminnelsen. I det kollapsade tillståndet visas endast den viktigaste informationen som om påminnelsen automatiskt kommer att upprepas när en service kopplad till den slutförs och hur långt det är kvar till påminnelsen. Texten som beskriver hur långt det är kvar till påminnelsen kan visa både kilometer och månader beroende på vilka intervall som är valda och texten blir röd när det börjar vara kort tid kvar, vilket även de motsvarande texterna på fordons- och garage-sidorna blir.

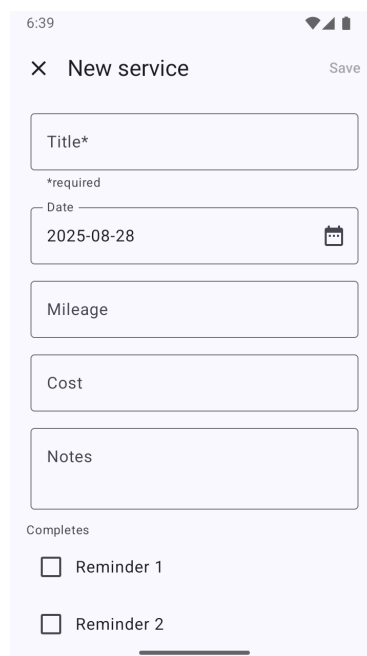
För varje påminnelse finns det knappar för att redigera och ta bort den, väljer man att ta bort en påminnelse öppnas en ruta som frågar om man är säker eftersom det inte går att ångra. Väljer man istället att redigera så öppnas gränssnittet som visas i Figur 5.

När man redigerar eller skapar en ny påminnelse krävs det att man anger en titel och hur långt det är tills påminnelsen ska ges, man kan ange det i både kilometer och månader men det räcker att ange en av dem. Utöver de nödvändiga fälten kan man välja om påminnelsen ska upprepas efter att en service kopplad till den har utförts och man kan ange övriga anteckningar längs ned. Bakåtnavigering och spara-knappen beter sig på samma sätt som beskrevs för fordon i sektion 3.2.

3.4 Servicehistorik



Figur 6:



Figur 7:

Den sista delen av applikationen handlar om servicehistoriken för fordon, se Figur 6. Denna sida liknar den för påminnelser väldigt mycket men visar information om servicetillfällen istället. Varje servicetillfälle representeras av ett listobjekt som anger titeln på servicen och vid vilket datum och miltal som den utfördes. Listobjekten kan även här expanderas för att visa mer information som kostnad, anteckningar och knappar för att hantera tillfället. Servicetillfällena sorteras efter datum med den senaste överst för att göra det lättare att hitta det man söker efter.

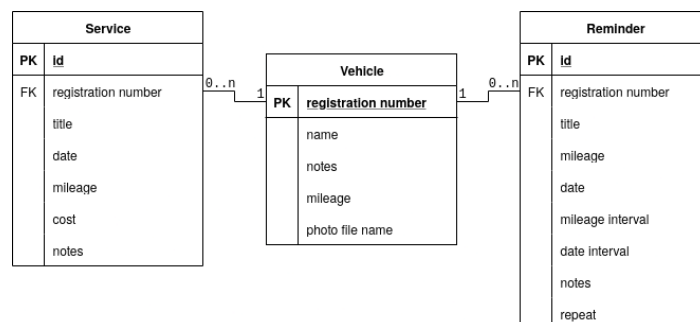
I redigeringsvyn för en service, se Figur 7, gäller samma saker som tidigare med bakåtnavigering, spara-knappen, och nödvändiga fält. Trycker man på kalendersymbolen i datumfältet öppnas en standarddialogruta för datumväljare, vilket gör att man slipper skriva in datum i rätt format manuellt. Om man skapar en service som har utförts vid ett miltal som är högre än vad fordonet har gått så uppdateras fordonets miltal automatiskt. När man skapar en ny service listas alla påminnelser längst ned med kryssrutor där man kan välja vilka händelser en service slutför, detta går inte att ändra i efterhand. När en påminnelse som inte upprepas slutförs så tas den bort från listan av påminnelser och för de som ska upprepas återställs intervallen för hur långt det är kvar till händelsen.

4 Applikationens uppbyggnad

GarageLog är helt uppbyggd av Compose med Compose navigering och använder inga aktiviteter eller fragment bortsett från `MainActivity`. Gränssnittet är uppbyggt av olika Compose skärmar och komponenter som finns under ui-paketet, varje underpaket till ui hanterar en distinkt del av applikationen eller innehåller återanvändbara komponenter.

Utöver ui-paketet finns det ett paket för dataklasser och ett för Room databasen, se Figur 8 för ett diagram över entiteterna i databasen och relationerna mellan dem. Som man kan se i diagrammet är informationen i applikationen representerad med tre tabeller i databasen: en för fordon, en för servicetillfällen och en för påminnelser. Varje servicetillfälle och påminnelse är kopplad till ett fordon genom det registreringsnummer, och ett fordon kan ha flera av båda.

Applikationen använder sig av MVVM (Model View ViewModel) arkitektur med Dagger Hilt för förenklad hantering av ViewModels med beroendelinjektion. Dagger Hilt injekterar automatiskt `VehicleRepository` instansen, som används för dataåtkomst, till vymodellerna och förenklar användningen av vymodell-fabriker.



Figur 8: Ett entitetsrelationsdiagram över dataklasserna i applikationen.

5 Reflektion

Jag valde att arbeta helt och hållet med Compose även fast vi inte gått igenom det lika mycket på kursen för att det kändes mest intressant och som det mer moderna arbetssättet. Detta skapade vissa utmaningar eftersom det var mycket nytt som jag inte kunde men som tur var finns det många resurser och bra dokumentation att ta hjälp av. I slutändan är jag nöjd med valet, jag känner att jag lärde mig mycket av det och föredrar att arbeta med Compose över aktiviteter och fragment.

Ett problem som uppstod under utvecklingen var inläsningen och visningen av bilder. Eftersom bilder från kameran ofta är stora kan det ta ett tag att läsa in och skala ned dem till rätt storlek, även om detta gjordes med coroutines så kändes gränssnittet långsamt när en bild lästes in. För att lösa problemet hittade jag ett bibliotek som heter Coil3 som har en composable `AsyncImage` som läser in och skalar bilden till rätt storlek i bakgrunden. Den visar en platshållarbild medan bilden blir inläst så att sidan kan visas för användaren direkt även om det tar lång tid.

Det finns flera saker jag skulle vilja förbättra eller lägga till om jag hade mer tid. En av dessa är struktureringen av vymodellerna, till exempel `VehicleViewModel` som från början var tänkt att användas vid visning och redigering av fordon används även på flera andra ställen och gör kanske mer än den borde. Istället hade man kunnat separera ut vymodellerna från ui-paketet och ge dem mer väldefinierade ansvar.

Några saker som fanns med i planen från början men som inte hanns med är att stötta flera bilder per fordon och låta användaren bläddra mellan dessa på fordonssidorna. Att exportera och/eller skriva ut all information inklusive servicehistorik för fordon om man ska sälja och vill skicka med servicehistoriken. Till sist att skicka en notis när en påminnelse när noll och det är dags för service.

Resurser

Nedan följer en lista av användbara resurser och exempel som har använts under utvecklingen.

- <https://saurabhjadhavblogs.com/compose-mvvm-roomdb-with-few-low-and-di>
- <https://proandroiddev.com/hilt-viewmodels-assisted-injection-aca2d6ee581d>
- <https://medium.com/@cgaisl/how-to-pass-arguments-to-a-hilt-viewmodel-from-compose-97c74a75f772>
- <https://medium.com/@fahadhabib01/basic-navigation-with-arguments-in-jetpack-compose-a-beginners-guide-8b43fb95e8c5>
- <https://medium.com/@riztech.dev/navigation-in-jetpack-compose-handling-the-back-stack-and-passing-data-8fb9a3ccfe9e>
- <https://m3.material.io/>
- <https://developer.android.com/reference/kotlin/androidx/compose/material3/package-summary>

Användning av AI

AI har använts för förbättring av språket i rapporten. Hela grundtexten och allt arbete är utfört av mig själv. AI har endast använts för korrigering av stavfel och grammatik.