

Flutter Parkeringsapplikation - Uppgift 3

Introduktion

Nu ska ni bygga vidare på er parkeringsapplikation genom att skapa en Flutter-klient som ersätter delar av ert CLI. Ni kommer att skapa:

1. En mobilapp för slutanvändare som vill parkera

Projektstruktur

parking_user (Mobilapplikation)

- **Målplattform:** Mobil (iOS/Android) och webb/desktop (landskapsläge)
- **Användare:** Personer som behöver parkera sina fordon
- **Navigation:** Använd `NavigationBar` eller `BottomAppBar` för bottennavigering, `NavigationRail` för sidnavigering.
- **Huvudfunktioner:**
 - Användarregistrering/inloggning
 - Fordonshantering
 - Val av parkeringsplats
 - Hantering av aktiva parkeringar

Krav för Godkänt (G)

Notera att kraven beskriver funktionalitet och inte val av implementationsstrategier. Detta är per design. Ni bestämmer själva vad ni vill ta med er från föreläsningarna och använder i uppgifterna. Troligen kommer ni använda en stor del av det som täcks dock.

Generella Krav

- Applikationen ska fungera som beskrivet ovan
- Lämplig felhantering och återkoppling till användaren
- Datavalidering där det är lämpligt

parking_user (Mobilapp)

1. Användarhantering:
 - Registrering av nya användare
 - In-/utloggning
2. Fordonshantering:
 - Lägga till/ta bort fordon
 - Lista egna fordon

3. Parkeringsfunktioner:

- Visa lediga parkeringsplatser
- Starta parkering
- Avsluta parkering
- Visa parkeringshistorik

Inlämningskrav

1. Kompletta källkod applikationen

2. Kort videodemo som visar:

- Huvudfunktioner i appen
- Navigation mellan olika vyer
- Exempelanvändning

3. Minimal dokumentation:

- Installationsanvisningar
- Lista över implementerade funktioner
- Kända begränsningar
- Du kan skriva dessa som kommentar till inlämningen eller README.md i ditt repo. Om det ligger i din README, var vänlig uppge det som kommentar i inlämningen.

Extra Utmaningar för VG (gör minst 2)

- Lägg till sökfunktioner för listor. T.ex. för att söka efter parkeringsplats.
- Lägg till funktionalitet för att redigera skapat data t.ex. för fordon eller användare.
- Lägg till sorteringsalternativ för listor som t.ex. parkeringsplatser eller pågående/avslutade parkeringar.
- Lägg till mörkt/ljust tema
- Eget med examinator kommunicerat och godkänt förslag.

Checklista för inlämning

Kategori	Krav	Status	Kommentar
Generella Krav			
	Båda applikationerna ska vara funktionella	<input type="checkbox"/>	
	Lämplig felhantering och återkoppling till användaren	<input type="checkbox"/>	
	Datavalidering där det är lämpligt	<input type="checkbox"/>	
parking_user			
Användarhantering	Registrering av nya användare	<input type="checkbox"/>	
	In-/utloggning	<input type="checkbox"/>	
Fordonshantering	Lägga till fordon	<input type="checkbox"/>	
	Ta bort fordon	<input type="checkbox"/>	
	Lista egna fordon	<input type="checkbox"/>	
Parkeringsfunktioner	Visa lediga parkeringsplatser	<input type="checkbox"/>	
	Starta parkering	<input type="checkbox"/>	
	Avsluta parkering	<input type="checkbox"/>	
	Visa parkeringshistorik	<input type="checkbox"/>	
VG Kriterier (2 krävs)			
	Sökfunktioner för listor	<input type="checkbox"/>	
	Redigering av skapat data (fordon+parkeringsplatser)	<input type="checkbox"/>	
	Sorteringsalternativ för listor (parkeringsplatser eller påbörjade/avslutade parkeringar)	<input type="checkbox"/>	
	Mörkt/ljust tema	<input type="checkbox"/>	
	Eget godkänt förslag	<input type="checkbox"/>	
Inlämningskrav			
	Komplett källkod	<input type="checkbox"/>	
	Videodemo	<input type="checkbox"/>	
	Dokumentation/README	<input type="checkbox"/>	

Instruktioner för användning:

1. Använd denna tabell för att hålla koll på din implementation
2. Markera status:
 - ☐ = Ej implementerat
 - ☒ = Under utveckling
 - ☒ = Implementerat och testat
3. Använd kommentarsfältet för att notera:
 - Kända problem eller begränsningar
 - Särskilda implementationsdetaljer
 - För VG-kriterierna: notera vilka 2 du har valt att implementera om du försökt för VG

Kom Igång Tips

1. Börja med UI/UX:
 - Skissa layouts på papper först
 - Börja med grundläggande widgets
 - Använd hårdkodad data initialt
 - Fokusera på navigationsflödet
2. Stegvis utveckling:
 - Implementera en feature i taget
 - Testa innan du går vidare.
 - Se till att programmet går att köra innan du fortsätter till nya uppgifter.
 - Håll koden organiserad
 - Överväg att skriva kommentarer för din egen skull eller för den som ska läsa koden senare (t.ex. kollega/examinator)

Angående inloggning/authenticering

- Vi har redan klämt in så mycket i kursen så vi kommer inte rulla vår egen authenticering med kryptering, salt, hash, session tokens eller annan vedertagen lösning.
- Därför kan ni leva med en väldigt simpel form av registrering/inloggning i er app.
- Förslagsvis:
 - Låt en registrering vara en simpel create av en person.
 - Låt en inloggning vara en check på klienten om ni t.ex. skrivit in namnet på en användare som faktiskt finns.
 - Det kan se ut som ett inloggningsflöde utan att bära någon form av säkerhet.
- Senare i kursen kommer vi nyttja Firebase för att implementera autentisering.
 - Vi kommer då koppla session tokens/firebase auth id:s från Firebase till skapade personer i databasen.
 - Vi tar det när det kommer!

Veckoplanering

Läs på Teams under "Classwork" för varje vecka (11-15) för ummaningar på hur ni tar er igenom uppgiften, några steg i taget. Dessa listar jag även nedan.

Vecka 11 - Grundläggande Setup och Widgets

- **Föreläsning 6:** Flutter installation, emulator, widgets, material, buildcontext
- **Att göra:**
 1. Skapa ett nytt Flutter-projekt:

```
flutter create parking_user
```

2. Sätt upp development environment
3. Experimentera med grundläggande Material widgets
 - <https://docs.flutter.dev/ui/widgets/material>
4. Skapa första utkast av huvudvyer

Vecka 12 - Layouts och Serverintegration

- **Föreläsning 7:** layouts, futurebuilder, dialog, kodning mot server
- **Att göra:**
 1. Implementera grundläggande layouts för alla vyer
 2. Börja integrera mot er server med:
 - HTTP-anrop via `http`-paketet
 - `FutureBuilder` för asynkron datahämtning
 - Felhantering och laddningsindikatorer

Vecka 13 - Navigation och Forms

- **Föreläsning 8:** navigering, modal, forms, expandable, scrolling
- **Avstämning E2.5** - CRUD-funktionalitet i mobilappen
- **Att göra:**
 1. Implementera navigation mellan vyer
 2. Skapa formulär för:
 - Användarregistrering
 - Fordonsregistrering
 - Parkeringsplatshantering
 3. Implementera scrolling lists och expandable widgets

Vecka 14 - State Management och Responsive Design

- **Föreläsning 9:** persist widget state, provider, responsive design, hero animation, animations
- **Att göra:**
 1. Implementera simpel state management med Provider
 2. Göra appen responsiv för olika skärmstorlekar och portrait/landscape
 3. Lägga till animationer för bättre användarupplevelse

Vecka 15 - Slutförande och Examination

- **Examination** - Fullständig implementation av appen samt inlämning