WEEK 3 -THEORY

UI Architecture - Theme, Widgets, Screens

Learning objectives

- ✓ **Structure Flutter widgets** for extendibility and consistency
- ✓ Comply with **coding conventions**
- ✓ Create a library of **generic widgets** aligned with a **design system**
- ✓ Understand the dart concepts of **static methods**, **attributes**

How to start?

- ✓ Create a **GitHub repository** for this project
- ✓ Get the **start code**, including the Figma Design System
- ✓ Ensure you can run the start project on your computer

How to submit?

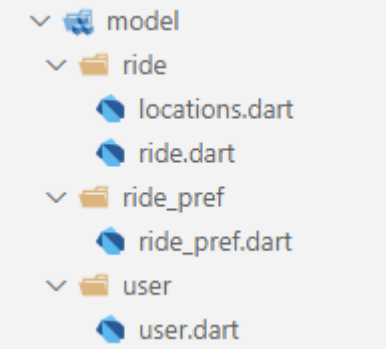
- ✓ Make sure you follow the COMMIT standards (see below)
- ✓ **Push the start code** into your repository **(**commit: BLA-000- Start Code)
- ✓ Once finished, submit to MS Team:
  - o GitHub URL
  - o This document

You are **not** allowed to **use AI** to submit this work

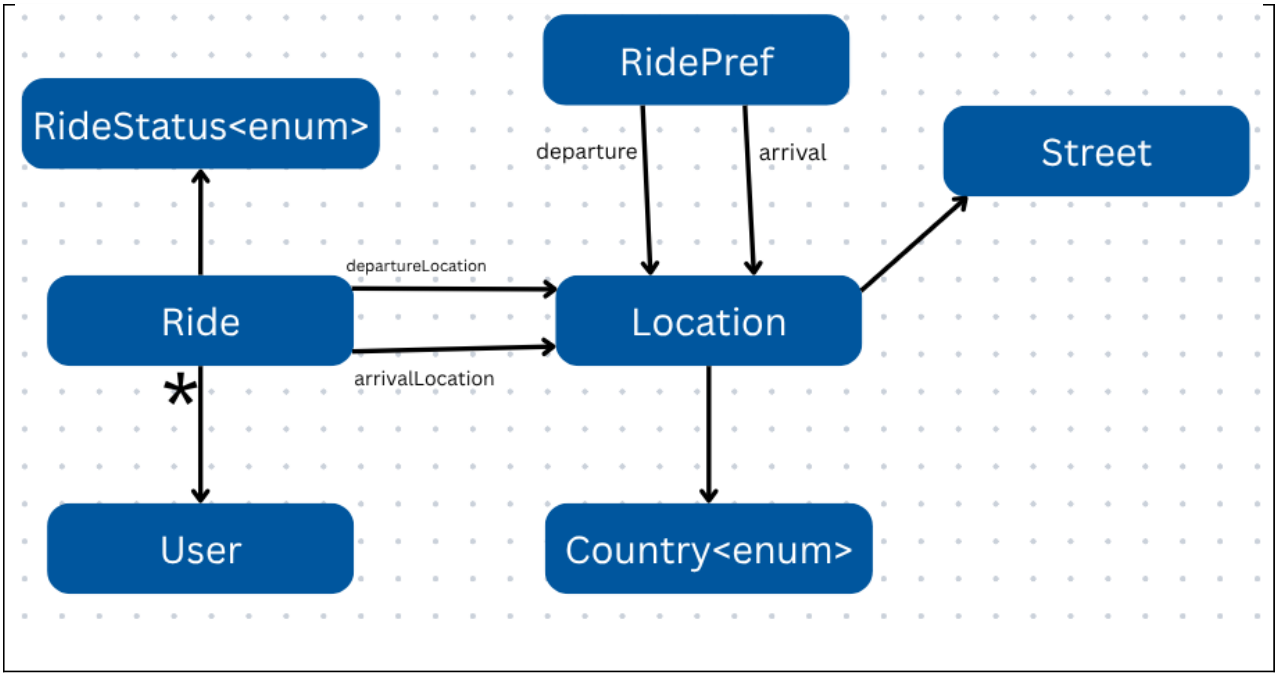
ACTIVITY 1 – Analyze the model layer

The model layer - so far - is composed of 6 classes:

Ride, Location, Country, DateTime, User, RidePref



Draw the **UML class diagram** representing the relationships between those 6 classes.

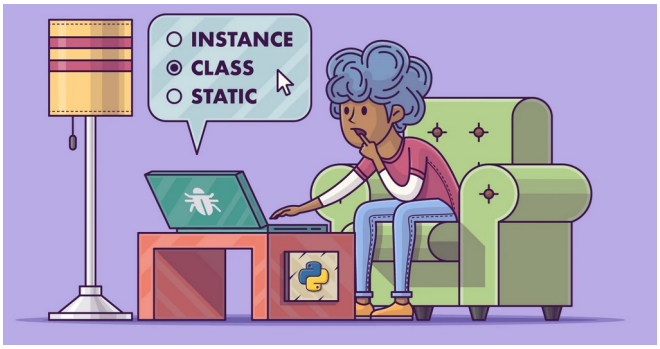


(no attribute, methods, just associations relationships)

What is the goal of the operator == and the hashCode method in Location class?

`operator ==` defines logical equality, and `hashCode` ensures that equality works correctly in collections like `Set` and `Map`.

ACTIVITY 2 – Understand the difference between **instance** and **static** access



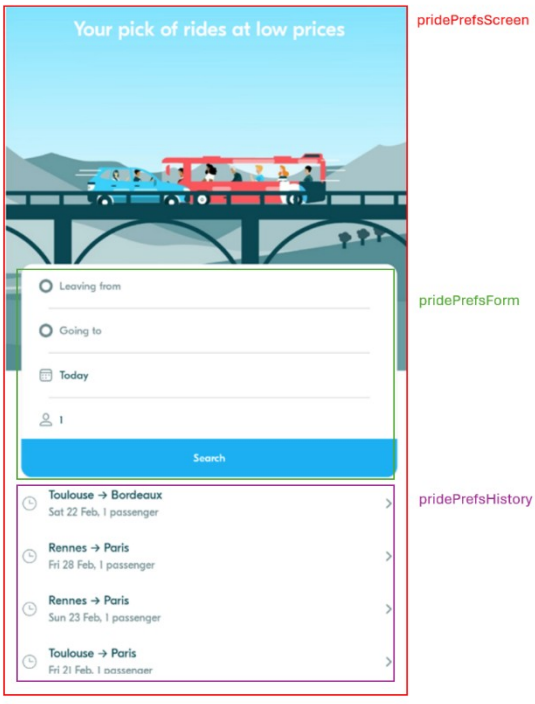
Answer the quiz in class

ACTIVITY 3 – Understand and analyze a first view**: Ride Preferences screen**

Let's build step by step a **Ride Preferences screen**, showing only the past entered ride preferences.

Understand the **methodology**, **steps**, and **coding standards** to implement this view

First version (for theory)



Final version (for practice)

Explain how the font (*Eesti*) is loaded from the assets and is applied to all widgets.

- The global theme sets fontFamily: 'Eesti' which makes it the default font for all text widgets
- The MaterialApp uses blaTheme, applying the Eesti font to every text widget in the app automatically

Analyze how the **history tile** interacts with **the App theme**.

History Tile	BlaTextStyle	BlaColor
title	BlaTextStyles.body(access by static method)	BlaColors.textNormal (access by static method)
subtile	BlaTextStyles.label(access by static method)	BlaColors.textLight(access by static method)

Explain how the **date** is converted into a readable label

```
if (targetDate == today) {
return 'Today';
} else if (targetDate == today.subtract(Duration(days: 1))) {
return 'Yesterday';
} else if (targetDate == today.add(Duration(days: 1))) {
return 'Tomorrow';
} else {
return DateFormat('E d MMM').format(dateTime); // Example: Wed 12 Feb
}
```

we have a targetDate as the current date so when it meet with the Yesterday it subtract with the duration 1 days and if it meet with the Tomorrow it will add duration 1 day.

Analyze the Ride Preferences screen and complete the table.

Widget	Screen / Screen Widget /App Widget	Parameters	Callbacks
RidePrefScreen	Screen (StatelessWidget)	RidePref ridePref	onRidePrefSelected()
BlaBackground	App Widget		No
RidePrefHistoryTile			

Note: The “RidePrefHistoryTile” doesn’t exist in the lib you provided but I see only RidePrefsService which is class not the widget it just a class.

At home, try to **reproduce it by yourself** (with the correction for help)