CADT IDT

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?

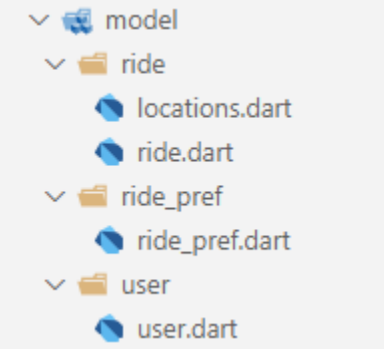
- ✓ Make sure you follow the COMMIT standards (see below)
- ✓ **Push the start code** into your repository **(**commit: `BLA-000- Start Code`)
- ✓ Once finished, submit to MS Team:
  - GitHub URL
  - This document

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

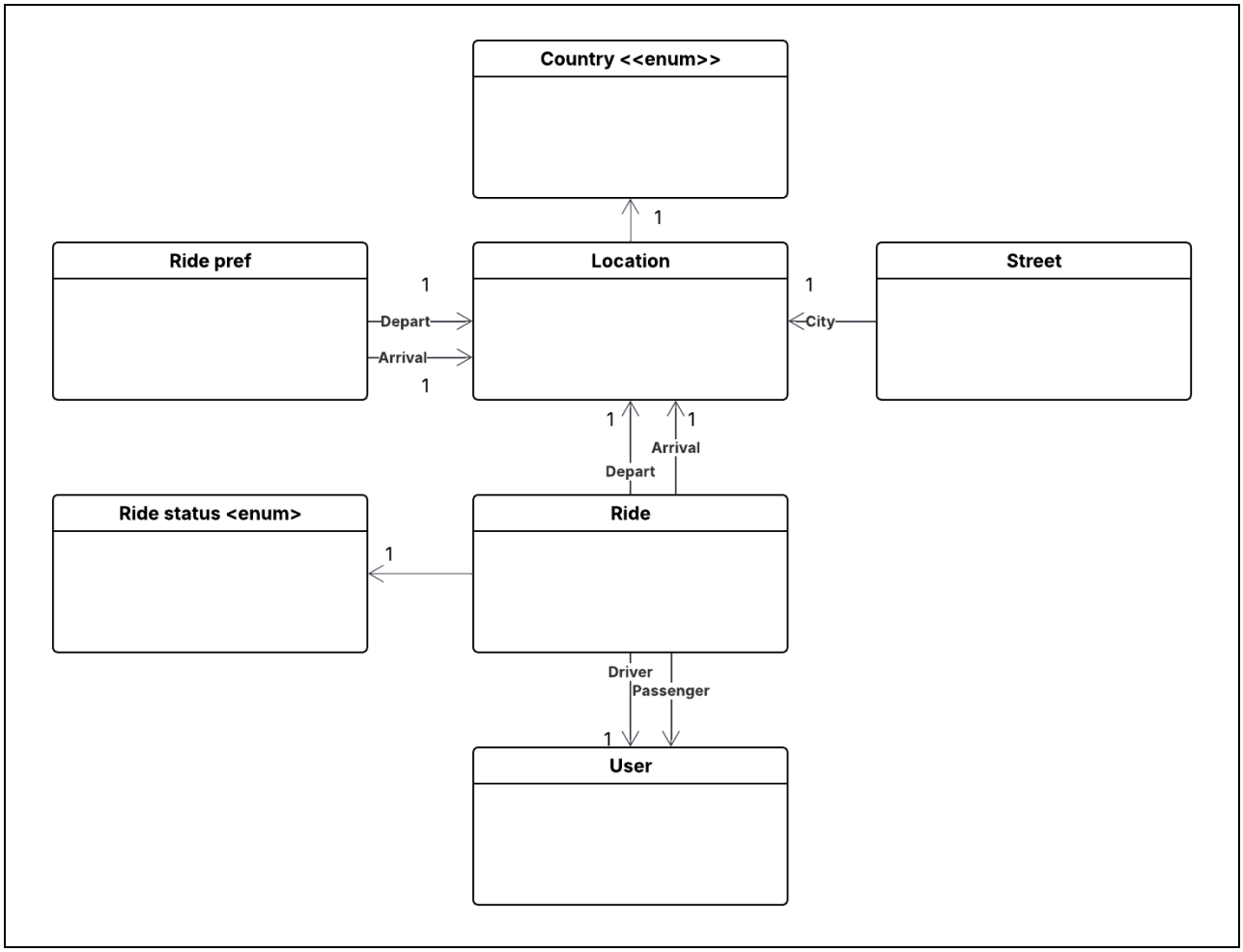
ACTIVITY 1 – Analyze the **model layer**

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.

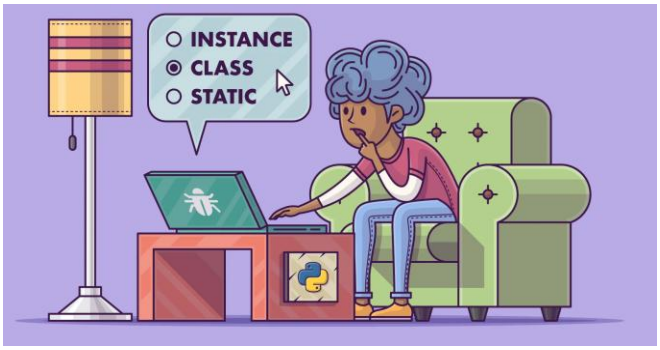


(no attribute, methods, just associations relationships)

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

The goal of the operator == is to compare the location country and name so if we done use the override in this operator dart only compare the memories address not the values.
The goal of the hash code method in location class is to make sure that two location with the same name and the same country are the same (have the same hash code) so it prevent duplicate.

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

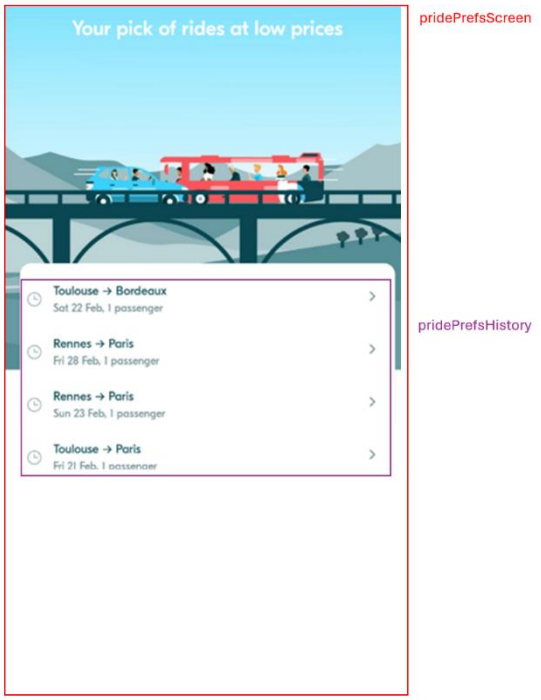


Answer the quiz in class

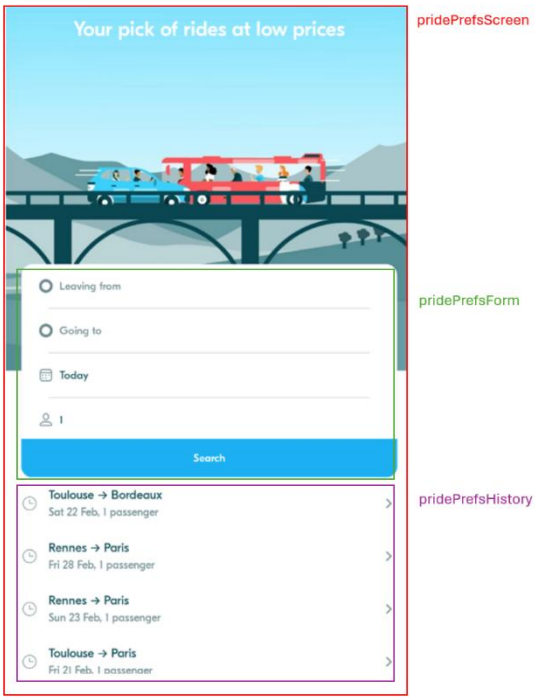
ACTIVITY 3 – Understand and analyze a first view**: Ride Preferences screen**

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.

So the Eesti font is is store in assets/fonts and it declare in pubspec.yaml and after that it register in global theme(theme.dart) and it applied to all widget through ThemData.

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

History Tile	BlaTextStyle	BlaColor
title	BlaTextStyle.heading	BlaColor.textnormal
subtile	BlaTextStyle.label	BlaColor.textlight

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

In the Ridepref model, the data is store as a DateTime in object and the model keep the data in raw format. but when the date is display it is formatting using the utility method **DateTimeUtils.formatDateTime.** and this method have in file Date_time_util that convert the raw data into user-friendly and readable label. And this method compares the given date with current date and return “Today”, “Yesterday” and “Tomorrow”. And if the date does not match these case it return and formatting using DateFormat(‘E, D, MMM’) to produce a readable string.

✏️ Analyze the Ride Preferences screen and complete the table.

Widget	Screen / Screen Widget /App Widget	Parameters	Callbacks
RidePrefScreen	Screen Widget	None	None
BlaBackground	App Widget	None	None
RidePrefHistoryTile	Screen	Ridepref ridepref	onPressed

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