

MOBILE DEVELOPMENT 1

Project Dart OOP

Introduction

You will design and implement a **Hospital Management System** in **Dart** using **Object-Oriented Programming (OOP)** principles and a **layered architecture** (Domain, Data, UI).

The main objective is to demonstrate your ability to analyze, design, and implement a real-world problem using good software engineering practices.



Agenda & Deliverables

DATE	DELIVERABLE / EVENT
NOV 02	UML Diagram Project code
NOV 03	Project Jury

1 - Specifications

Your system should be able to manage 1 or several key hospital operations listed below.

 **Managing staff** (doctors, nurses, and administrative personnel)

 **Managing rooms** and beds allocated to patients

 **Managing patients**, their **appointments**, and **doctor** meetings

 **Supporting prescription** management and safe medication tracking

 **Note:** You are encouraged to select one feature or functional area (e.g., appointments, room management, or prescriptions) and develop it in depth — rather than trying to cover everything superficially !

Your implementation **must follow the layered architecture** seen in class:

Layer	Description
UI	Handles user interaction (console).
Domain	Contains the main business classes and logic.
Data	Manages data storage and retrieval (optional file I/O).

2- Deliverables

UML Diagram

- Model **only the DOMAIN layer** (no UI or Data classes).
- Include:
 - Classes, attributes, methods
 - Associations, multiplicities, inheritance
 - Aggregation / composition where appropriate

Code

Implement the system in Dart, organized by layers:

```
/lib
  /domain
  /data
  /ui
  /test
```

- ✓ Apply **good OOP design**: encapsulation, inheritance, and polymorphism.
- ✓ Ensure **clear separation** between layers.

Unit Tests

- Provide **unit tests** for the domain layer.
- Example: test scheduling an appointment, assigning a room, validating a prescription.
- Use the Dart test framework (package:test/test.dart).

Optional File Persistence

- You may implement save/load features (JSON, text, CSV...).

⚠ Any **AI-generated code** must be explicitly commented:

```
// AI generated
```

3- Jury Timeline

Criterion	Description	TIME
Presentation	Present the Users and the Use Cases (UML Use Case) of your application	1 min
UML Design	Present the UML diagrams and how we navigate on your Domain classes	2 min
Implementation	Present your Layered Architecture and services	1 min
Unit Tests	Present and run the unit tests	1 min
Jury Question	Able to answer to the jury question (what if....)	5 min

4- Evaluation

Criterion	Description	%
Presentation	Clear and concise presentation	10 %
UML Design	Completeness, correctness, and clarity of class diagram	20 %
Implementation	Correct use of OOP and domain logic Layered Architecture	30%
Unit Tests	Quality, coverage, and execution of tests	10 %
Jury Question	Able to answer to the jury question (what if....)	30 %