جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

TIE 204
MVP Studio Lab 3

1. Lab Purpose

This lab translates the lecture *Design 101: Interfaces & Errors* into concrete system-engineering practice.
By the end of this lab, each team will:

- Define **clear system interfaces** before implementation
- Model **data and ownership** explicitly
- Detect **interface-level errors** (blocking, deadlock, starvation, silent failure)
- Prepare the system for **clean team split and MVP development**

Key principle: Interfaces are system contracts. Bad interfaces create bugs before code exists.

Task 1: Functional Decomposition

Break the system into **functional blocks** (not HW/SW yet).
Typical functions:

- Sense
- Communicate
- Process
- Decide
- Act

Deliverable:

- One functional block diagram (should be ready form Last week)

Task 2: Identify Interfaces

For every interaction between blocks, define an interface.
Mandatory Interface Table:

Interface Name	Producer	Consumer	Data Exchanged	Direction
IF-1				
IF-2				

Rule: If two blocks exchange *anything*, that is an interface.

Task 3: Interface Contract

For **each interface**, define the contract.

Template:

Interface Name:
Data:
Update Rate:
Timing Constraints:
Validity Duration:
Accuracy / Precision:

Rules:

- No protocols yet
- No implementation details

Task 4: Define Data Structures

For every interface, model the data explicitly.

Example:

- Temperature:
  - Unit: °C
  - Range: -20 → 80
  - Update: 1s
  - Owner: Sensor block
  - Consumers: Control algorithm
  - Classify each data item as:
    - Measurement
    - Command
    - State
    - Event
    - Configuration
    - Log

Task 5: Ownership Mapping

Define responsibility for each data item.

Data	Owner	Who Updates	Who Reads	Who Can Modify

Rule:

Ownership defines responsibility. Undefined ownership creates deadlocks and silent failures.

Task 6: Blocking vs Non-Blocking

For each interface, answer:

Is the interface blocking?

What is it waiting for?

Can both sides wait at the same time? If **yes** → mark **deadlock risk**.

Task 7: Interface Failure Modes

Each interface must define failure behavior.

Failure Type	How Detected	System Reaction
Timeout		
Invalid data		
Missing data		
Overflow / Backpressure		

Rule:

An interface without a failure path is an incorrect interface.

Task 9: Interface Dependency Graph

Draw a lightweight dependency graph:

- Nodes = subsystems
- Arrows = “waits for”

Check for Cycles

Task 9: Error Classification

Answer the following for your system:

Possible deadlock? Yes / No
Possible livelock? Yes / No
Possible starvation? Yes / No

Justify each answer in 1–2 sentences.

Task 11: Interface-Driven Team Split

Assign ownership based on subsystems.

Team Member	Subsystem Owned	Interfaces They Must Respect

Rule:

No one owns the whole system. Each person owns a subsystem and its interfaces.

Final Lab Deliverables

Add Your Interface design to your Product Specification Document and should be including:

1. Functional block diagram
2. Interface list and contracts
3. Data models and ownership tables
4. Interface error analysis
5. Deadlock / livelock reasoning
6. Team split table