DISTRIBUTED SYSTEMS AND IOT

Ordering system “E Fuori Nevica”

Simone Montella M63001566
Angelo Andrea Nozzolillo M63001602

TABLE OF CONTENTS

01

THE PROBLEM

Brief discussion of the problem and definition of the objectives set.

01 THE PROBLEM

The problem faced concerns the management of **orders** in a pizzeria.

When a waiter takes an order, it is crucial to ensure that the requested pizzas are achievable, considering the **availability** of ingredients.

01 THE PROBLEM

Our idea

It is necessary to implement a **distributed system** capable of verifying and updating the availability of ingredients, in a "coordinated" way, preventing inconsistencies in the management of orders that can compromise the customer experience.

TABLE OF CONTENTS

02

THE DISTRIBUTED ARCHITECTURE

Description of the network components

02 THE DISTRIBUTED ARCHITECTURE

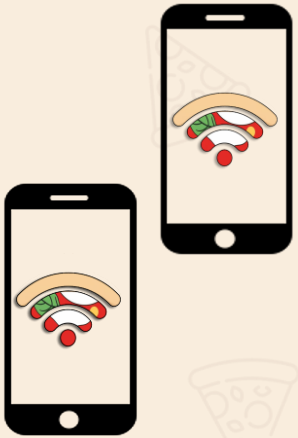
Web server

The web server hosts a **database** with 2 tables:
one for the management of ingredients and one for the management of nodes (waiters).

The waiters access the server:

- To record your information
- To get the information of the other nodes
- To check the availability of ingredients

02 THE DISTRIBUTED ARCHITECTURE

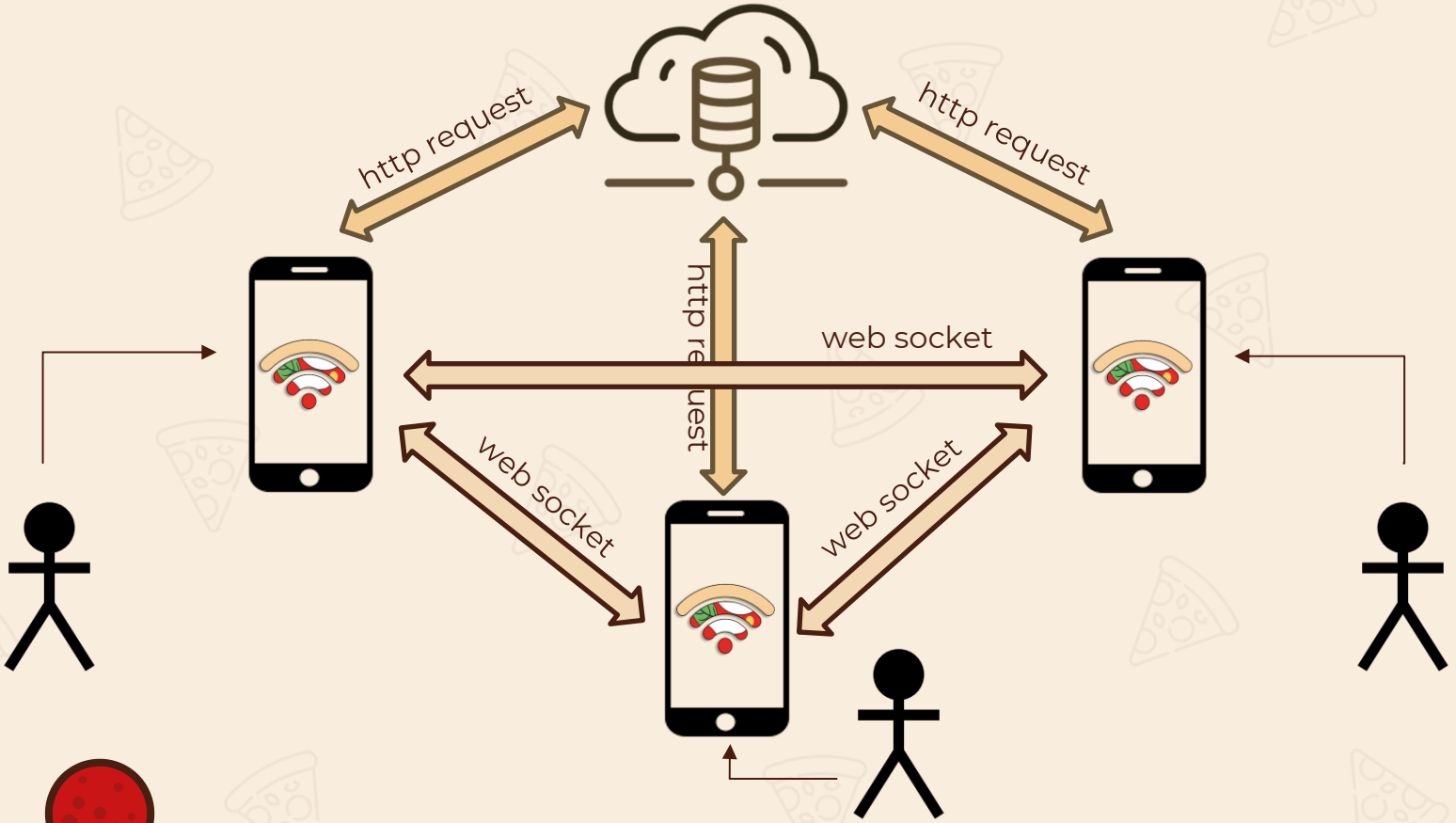


Smartphone

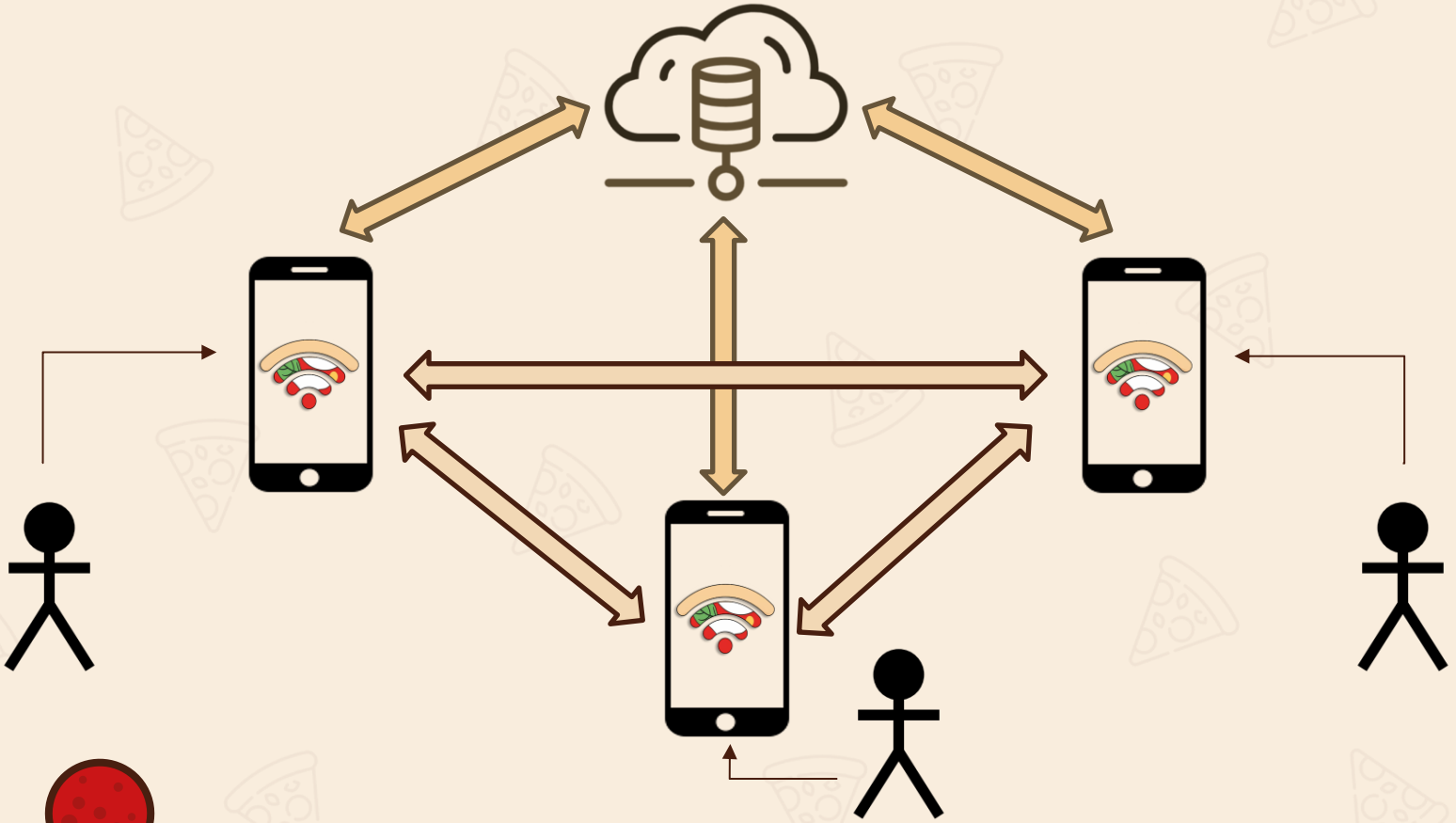
The smartphones supplied to the waiters are used to capture orders.

The devices are interconnected in a **local network**, to access the web server and communicate with each other, coordinating in mutually exclusive access to the database of ingredients.

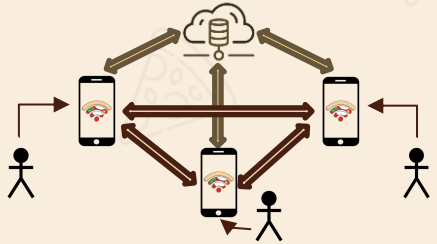
02 THE DISTRIBUTED ARCHITECTURE



02 THE DISTRIBUTED ARCHITECTURE



02 THE DISTRIBUTED ARCHITECTURE



- The system is **asynchronous**.
- Processes are **not** subject to failures.
- Channels are **reliable**, and messages are delivered intact once and only once.
- Processes spend a **finite time** in the critical section and release the common resources after a limited time

TABLE OF CONTENTS

03

THE MUTUAL EXCLUSION

Description of the "Ricart-Agrawala" mutual exclusion algorithm and in-app implementation

Distributed Mutual Exclusion: Requirements

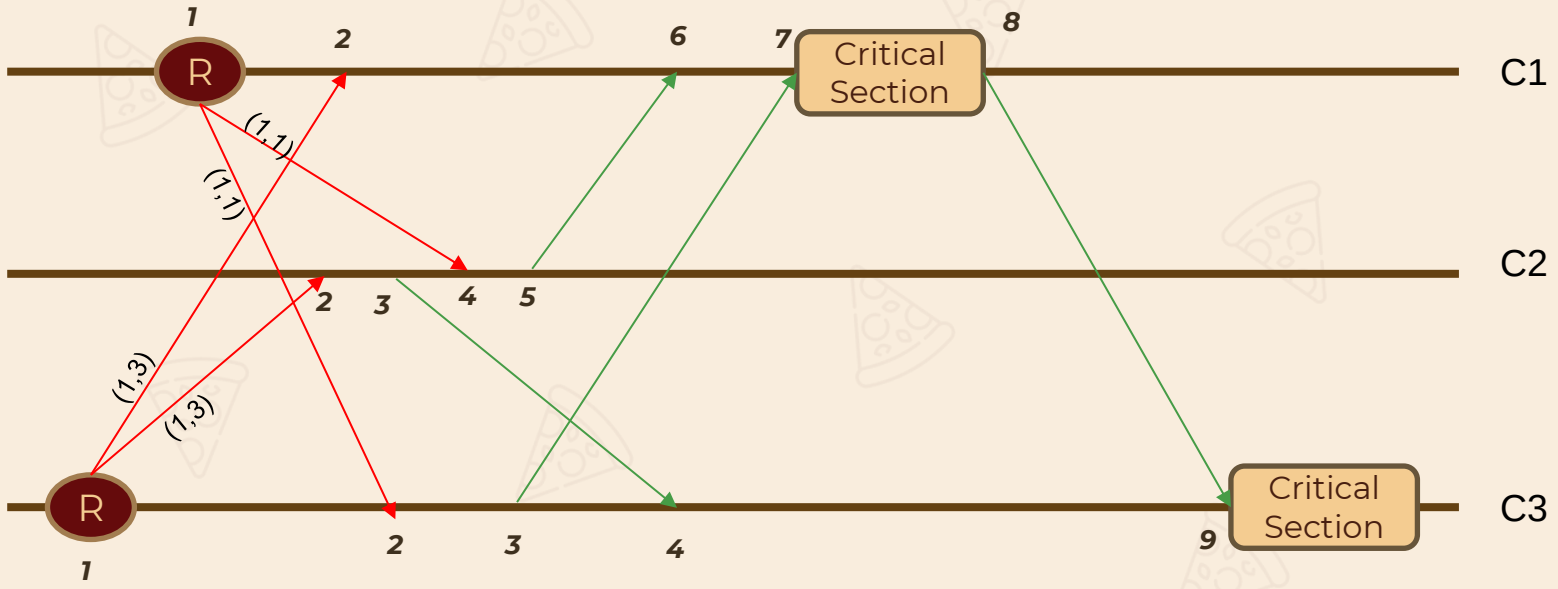
- **Safety**: At most one process can execute in the critical section at a time;
- **Liveness**: Requests to enter and exit the critical section eventually succeed.
- **Ordering**: if the requiest xis HB y(x→y), access to the critical section by x must occur before access by y.

RICART-AGRAWALA

The Ricart and Agrawala's algorithm is based on the principle that a process can enter the critical section only when its request is **stable**, i.e. it has received N-1 grant from the other processes.

The process that is using the resource simply **delays** sending ACKs until its critical section is finished.

03 THE MUTUAL EXCLUSION



03 THE MUTUAL EXCLUSION

WHY RICART-AGRAWALA?

1. Efficiency in communication

•Only 2(N−1) messages per critical section access.

•Better than Lamport.

2. Tokenless

•The algorithm does not consume network bandwidth when no process requires access to common resources.

3. Easy to deploy

•Based on logical timestamps and ordered queue.

•Clear management of request priorities.

4. Fairness guaranteed

•Equitable access to the critical section thanks to timestamps.

•Avoid starvation.

03 THE MUTUAL EXCLUSION

HOW IT WAS INTEGRATED INTO THE APPLICATION

The waiter captures the order and confirms the summary

The system forwards the request for access to the WS to everyone, starting the mutual exclusion algorithm

In the critical section, the system checks the feasibility of each pizza ordered

The system confirms the order or notifies the waiter of any pizzas that cannot be made

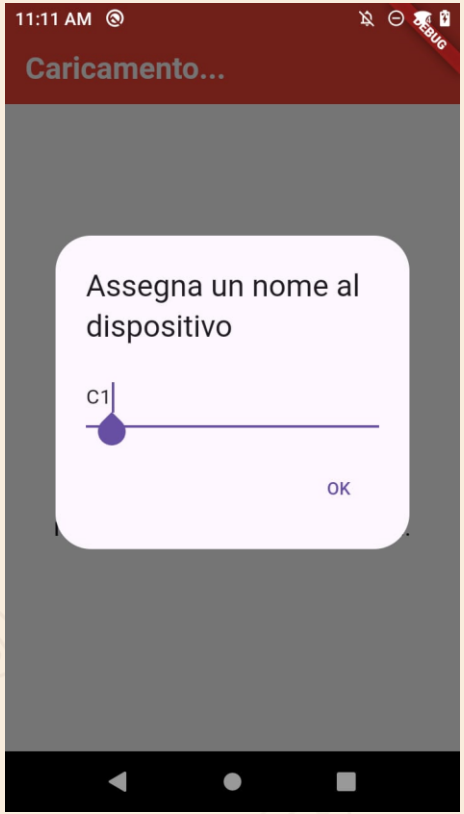
TABLE OF CONTENTS

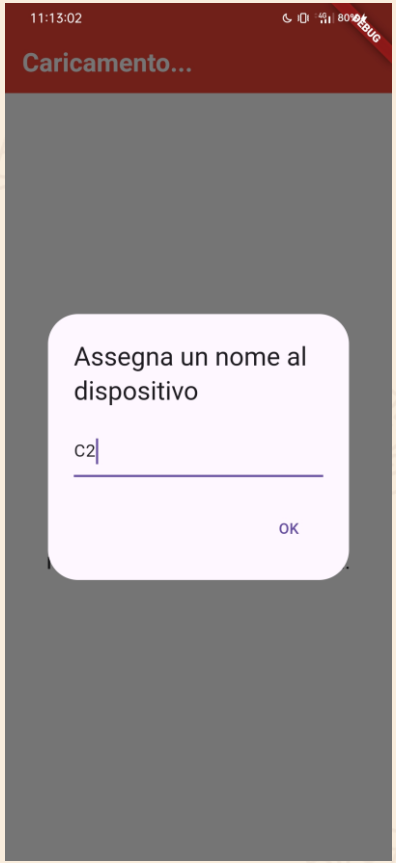
04

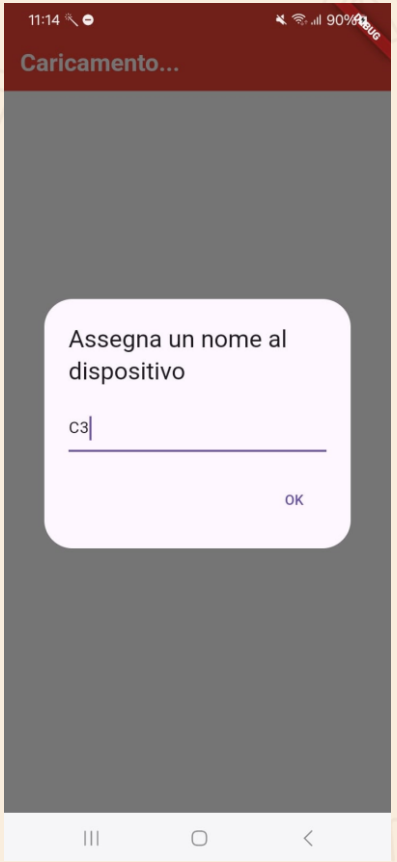
TESTING AND EVALUATION

We present the application, the tests conducted, and the results obtained.

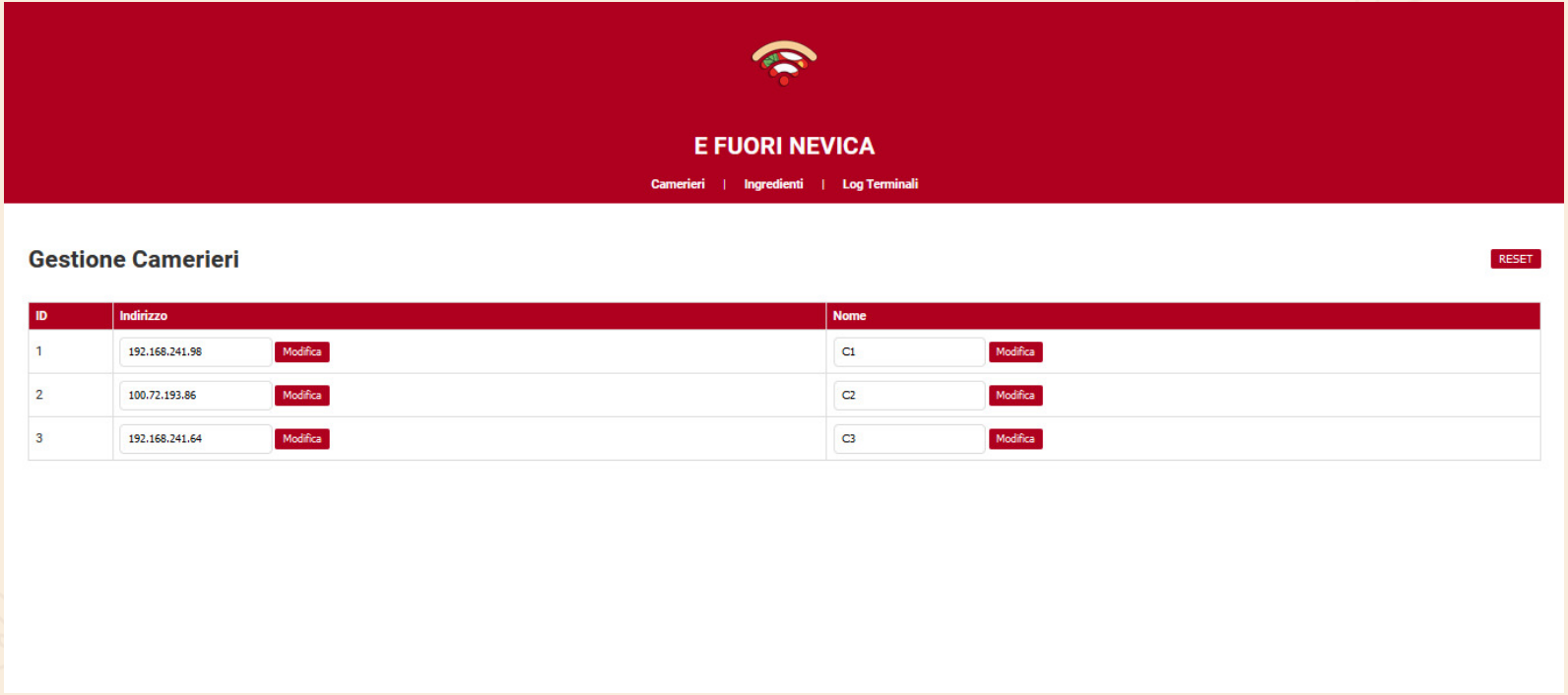
04 TESTING AND EVALUATION



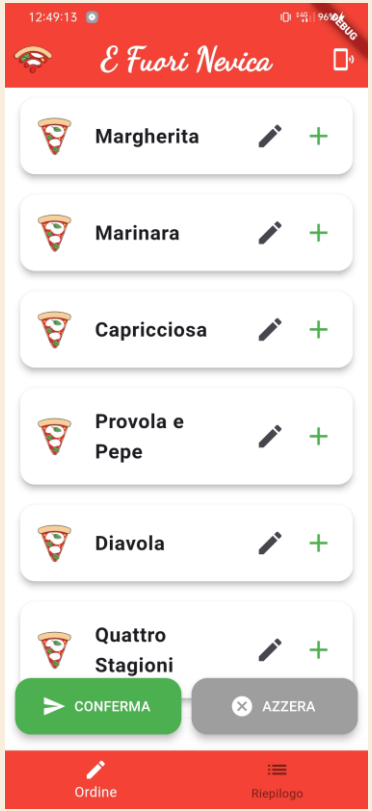


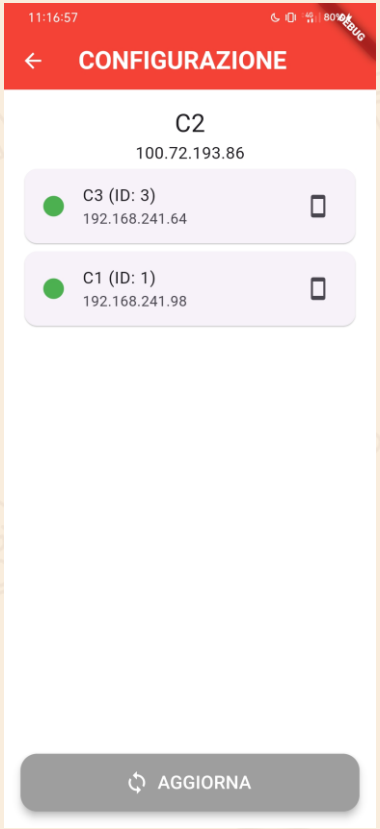


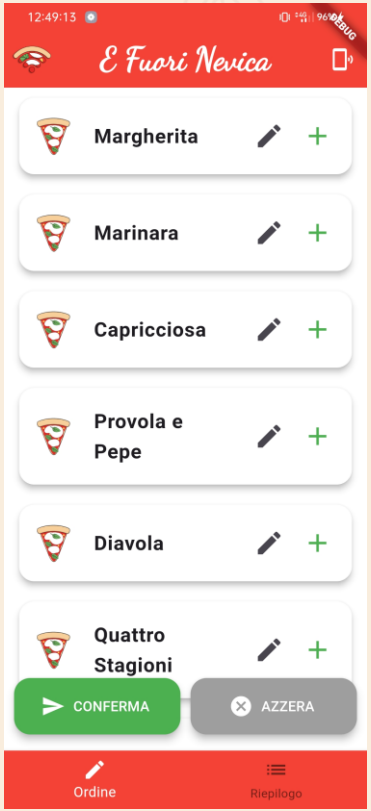
04 TESTING AND EVALUATION



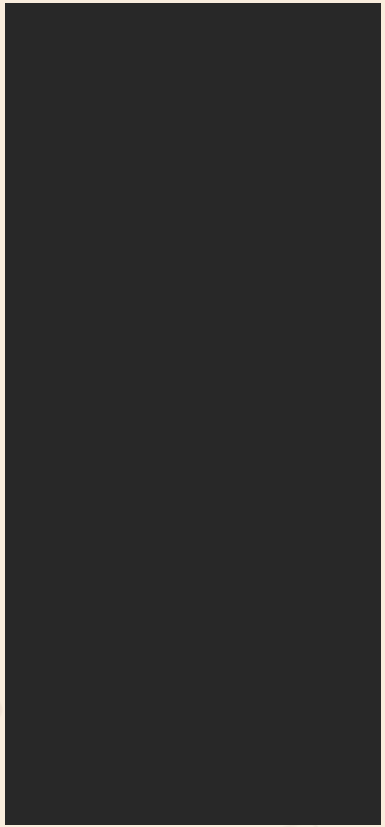
04 TESTING AND EVALUATION

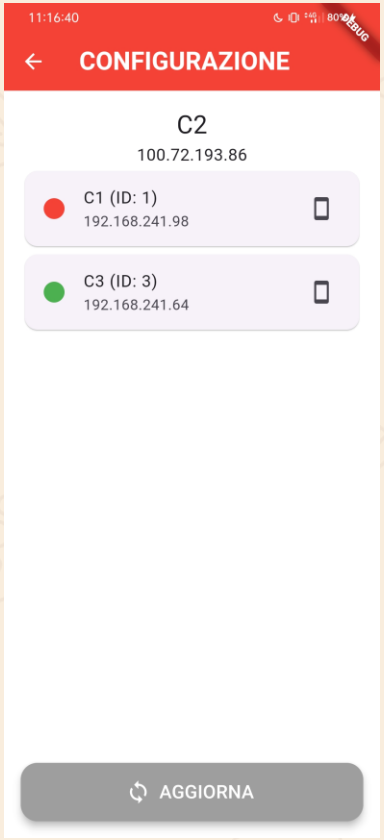


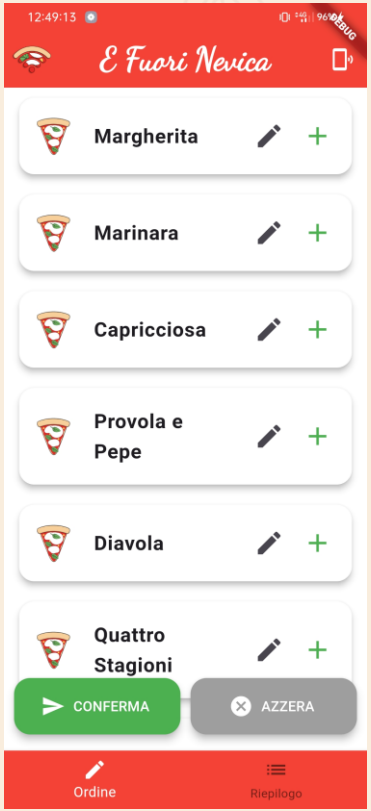




04 TESTING AND EVALUATION

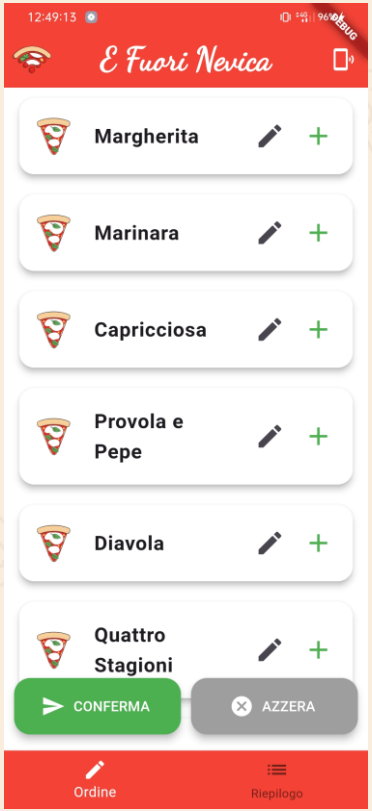


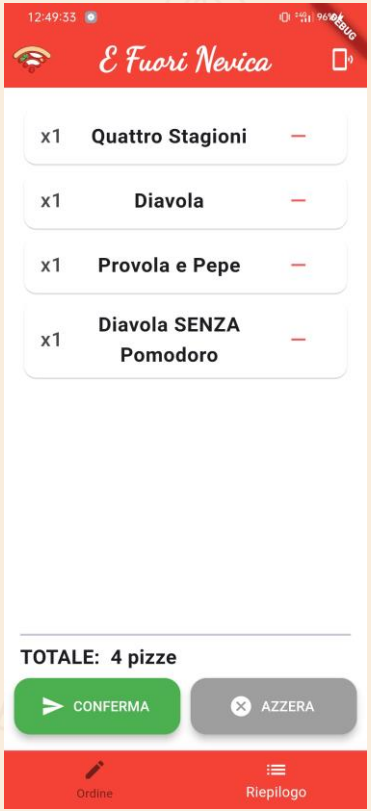




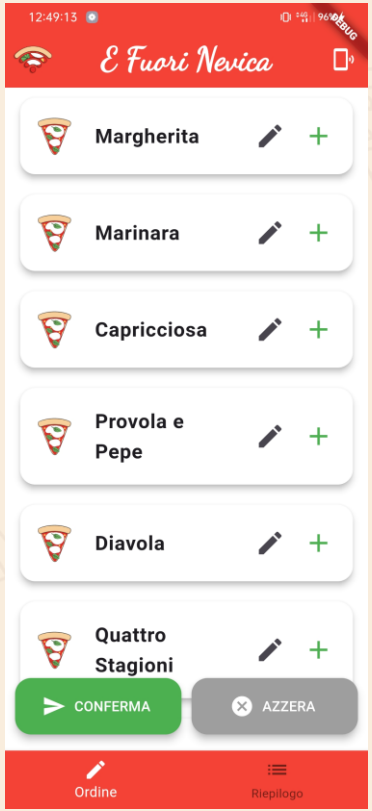
04 TESTING AND EVALUATION

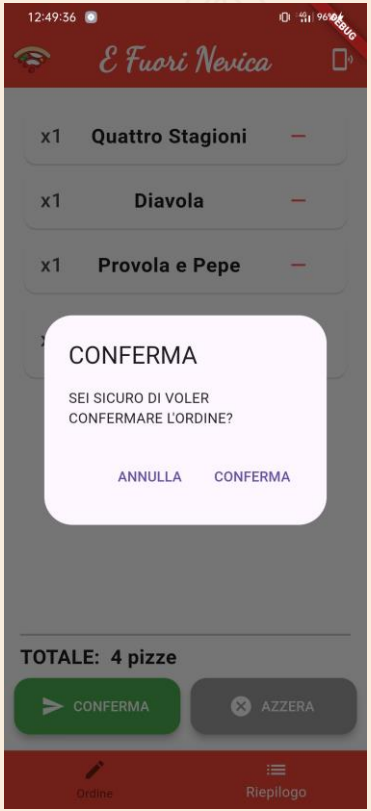
04 TESTING AND EVALUATION



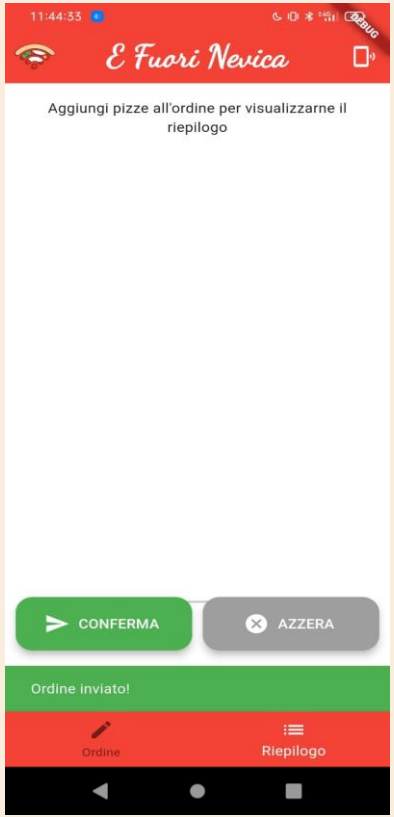


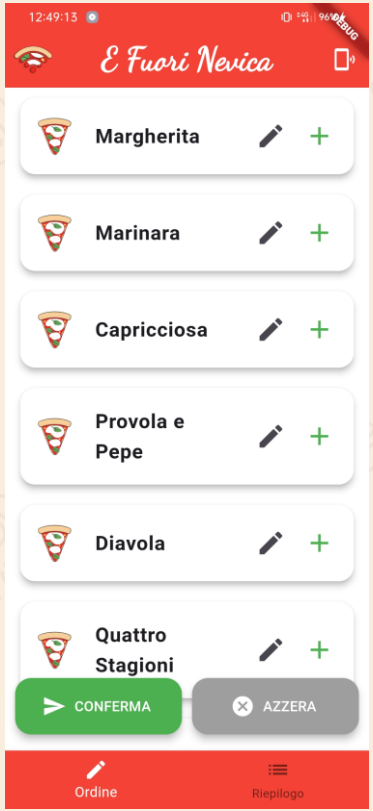
04 TESTING AND EVALUATION

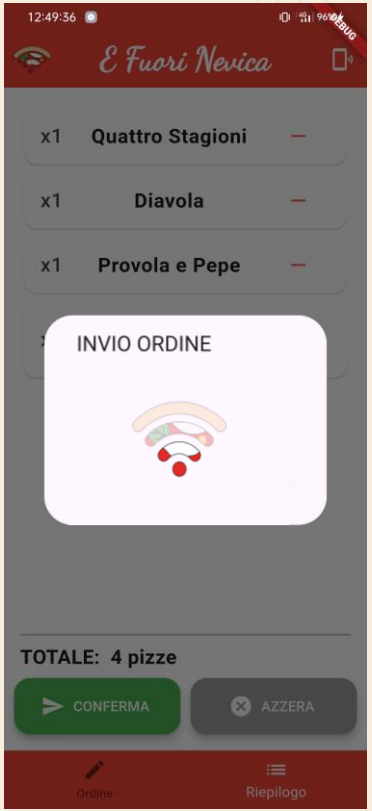




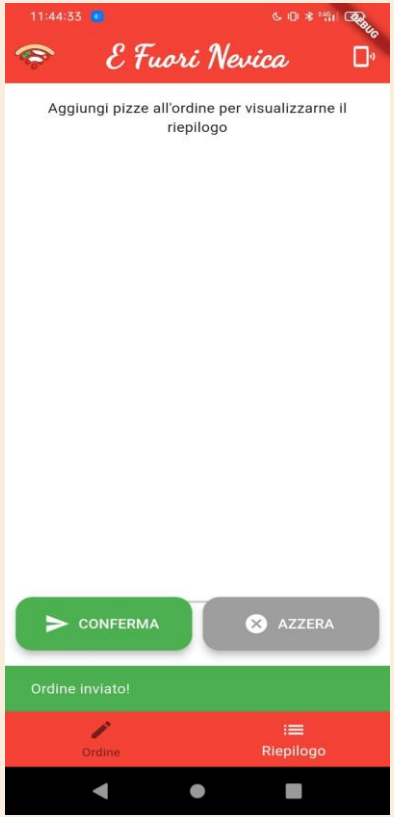
04 TESTING AND EVALUATION

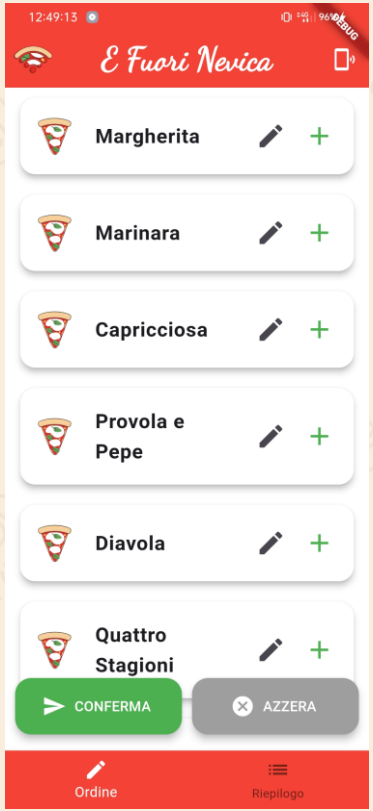


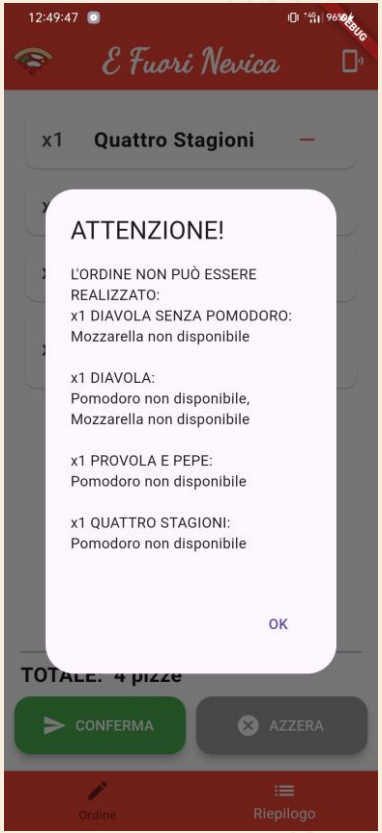




04 TESTING AND EVALUATION







04 TESTING AND EVALUATION

THANK YOU FOR YOUR ATTENTION!

References:
- G. Coulouris et al.: Distributed Systems: Concepts and Design, V ed., 2012.
- Slides of the Course

Simone Montella M63001566
Angelo Andrea Nozzolillo M63001602