

NekoDev

Daniele Cafagno, Ilaria Ranieri, Luca di Tria

Project Work

Auriga & ITS Apulia Digital Maker



Progettazione del piano di sviluppo del progetto

Quali sono le varie fasi del nostro progetto



Ricerca



Implementazione



Presentazione

RICERCA DEL PROBLEMA

Studio e analisi dei dati e delle richieste del progetto

Progettare e realizzare un sistema composto dalle seguenti componenti:

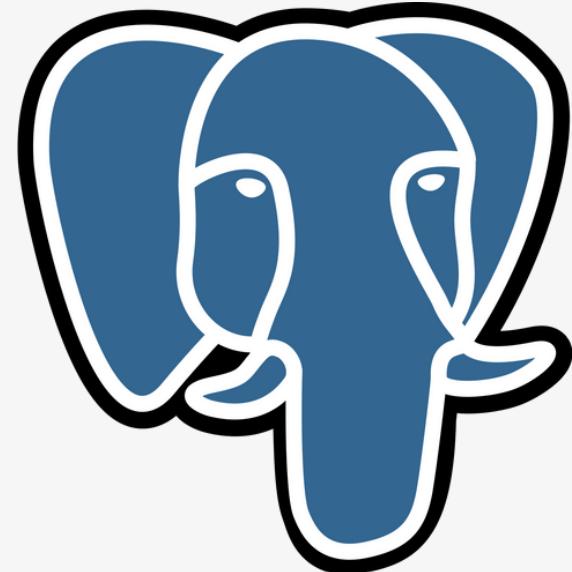
- Un servizio **Rest** sviluppata in **JAVA** che permetta:
 - la gestione delle funzionalità (inserimento, modifica, cancellazione e lettura);
 - la gestione delle abilitazioni della carta;
 - la possibilità di richiedere l'elenco delle funzionalità attive su un terminale considerato il PAN della carta inserita.
- Un **WEB Client** (HTML, CSS e Javascript o React) in grado simulare l'inserimento della carta all'interno dell'ATM.
 - Il client deve richiedere e mostrare a video le funzionalità attive. I dati di input (bankCode, peripherals, PAN) possono essere inseriti utilizzando un form.
- Dettagliare il progetto motivando le scelte tecniche
- Creare un **database** per la persistenza dei dati (opzionale)

IMPLEMENTAZIONE

Strutture e domini applicativi utilizzati



spring[®]



JS

CSS

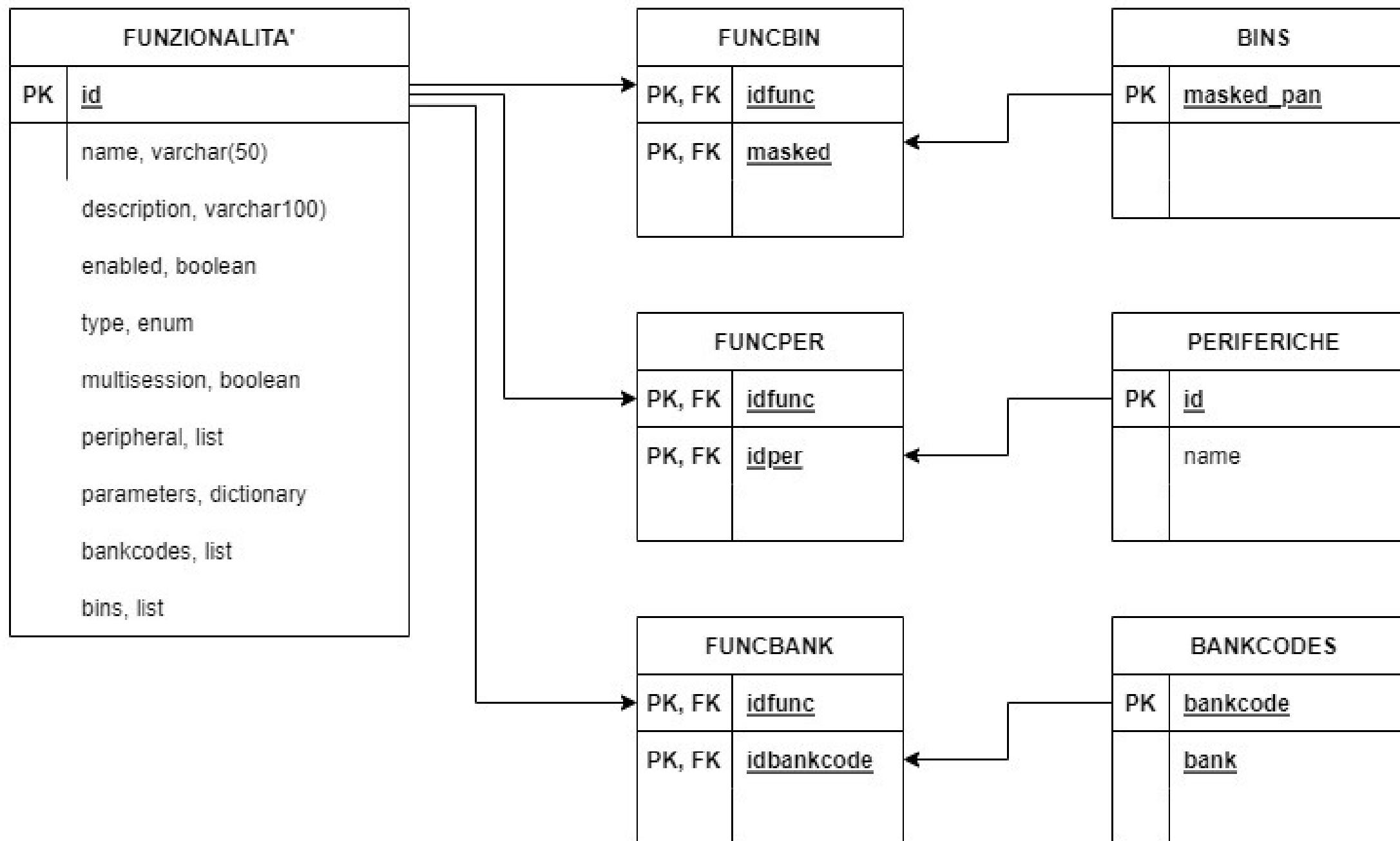
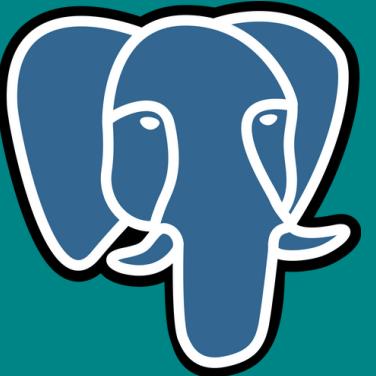


HTML



IMPLEMENTAZIONE

Strutture e domini applicativi utilizzati



È stato utilizzato un sistema SQL in quanto garantisce la funzione rollback in caso di fallimento di un'operazione della transaction che coinvolge più tavole

IMPLEMENTAZIONE

Strutture e domini applicativi utilizzati



A screenshot of the Spring Tool Suite 4 IDE. The code editor shows a Java class named `Functionality.java` with annotations like `@Entity`, `@Id`, and `@OneToMany`. The terminal window at the bottom shows Hibernate SQL queries being executed, such as `select functionality0_.id as id1_4_, functionality0_.bin_masked_pan as bin_mask2_4_, functionality0_.functionality_id as function3_4_ from functionality`.

```
workspace - ATMapi/src/main/java/it/its/atmapi/domain/Functionality.java - Spring Tool Suite 4
Edit Source Refactor Navigate Project Run Window Help
File Explorer × Functionalities × Functionalities × Functionalities × Functionalities × Functionalities × Parameter.java Peripheral.java
ApulianSongsAPI [boot] [devtools]
ATMapi [boot] [devtools]
src/main/java
  it.its.atmapi
    domain
      BankCode.java
      Bin.java
      Functionality.java
      FunctionalityBankCode.java
      FunctionalityBin.java
      FunctionalityPeripheral.java
      Parameter.java
      TypeFunc.java
    dto
      it.its.atmapi.dto
    exceptions
    iservices
    repositories
    restcontrollers
    services.impl
      BankCodeServiceImpl.java
      BinServiceImpl.java
      FunctionalityServiceImpl.java
      PeripheralServiceImpl.java
    utils
  it.its.atmapi.utils
  Javadoc Declaration Console × Terminal Gradle Tasks Gradle Executions
ATMapi - AtmApiApplication [Spring Boot App] C:\Users\U-20\Desktop\sts-4.14.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (27 ott 2022, 09:29:58) [pid: 8964]
Hibernate: select bankcode0_.bank_code_id as bank_code_id0_0_0_, bankcode0_.bank_name as bank_name0_0_0_ from bank_code bankcode0_ where bankcode0_.bank_code_id=?
Hibernate: select functionality0_.id as id1_4_, functionality0_.bin_masked_pan as bin_mask2_4_, functionality0_.functionality_id as function3_4_ from functionality
Hibernate: select bin0_.masked_pan as masked_p1_1_0_ from bin bin0_ where bin0_.masked_pan=?
Hibernate: delete from functionality_peripheral where id=?
Hibernate: delete from functionality_bank_code where id=?
Hibernate: delete from functionality_bin where id=?
Hibernate: delete from functionality where id=?
Hibernate: SELECT functionality.id, functionality.description, functionality.enabled, functionality.type_func, functionality.multisession, functionality.
```

È stato utilizzato
Spring in quanto
framework, quindi
agevolando la scrittura
del codice e
l'approccio *CodeFirst*

IMPLEMENTAZIONE

Strutture e domini applicativi utilizzati



sts-workspace - ATMapi/src/main/java/it/its/atmapi/repositories/FunctionalityRepository.java - Spring Tool Suite 4

```
1 package it.its.atmapi.repositories;
2
3
4
5 import java.util.List;
6
7
8 public interface FunctionalityRepository extends JpaRepository<Functionality, Integer>{
9     @Query(value = "SELECT functionality.id, functionality.description, functionality.enabled, functionality.type_func, functionality.multisession, fu
10         + "INNER JOIN functionality_bank_code ON functionality.id = functionality_bank_code.functionality_id "
11         + "INNER JOIN functionality_bin ON functionality.id = functionality_bin.functionality_id "
12         + "INNER JOIN functionality_peripheral ON functionality.id = functionality_peripheral.functionality_id "
13         + "WHERE functionality.enabled AND functionality_bank_code.bankcode_bank_code_id = :bankCodeId AND functionality_bin.bin_masked_pan = :ma
14         , nativeQuery = true)
15     public List<Functionality>findCustom(@Param(value ="bankCodeId") String bankCodeId, @Param(value ="maskedPan")String maskedPan);
16
17 }
18
19 }
```

Java code for the `FunctionalityRepository` interface, which extends `JpaRepository`. It includes a custom query method `findCustom` that joins three tables: `functionality`, `functionality_bank_code`, and `functionality_bin`, filtering by `enabled` status and specific IDs.

Boot Dashboard X

Console X

```
ATMapi - AtmApiApplication [Spring Boot App] C:\Users\U-20\Desktop\sts-4.14.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64-17.0.2.v20220201-1208\jre\bin\javaw.exe (27 ott 2022, 09:29:58) [pid: 8964]
Hibernate: select bankcode_.bank_code_id as bank_code1_0_0_, bankcode_.bank_name as bank_name2_0_0_ from bank_code bankcode_ where bankcode_.bank_code_id=?
Hibernate: select functionality0_.id as id1_4_, functionality0_.bin_masked_pan as bin_mask2_4_, functionality0_.functionality_id as function3_4_ from functionality_
Hibernate: select bin0_.masked_pan as masked_p1_1_0_ from bin bin0_ where bin0_.masked_pan=?
Hibernate: delete from functionality_peripheral where id=?
Hibernate: delete from functionality_bank_code where id=?
Hibernate: delete from functionality_bin where id=?
Hibernate: delete from functionality where id=?
Hibernate: SELECT functionality.id, functionality.description, functionality.enabled, functionality.type_func, functionality.multisession, functionality.name
```

Writable Smart Insert 1:1:0

IMPLEMENTAZIONE

Strutture e domini applicativi utilizzati



The image displays a user interface for an ATM system, divided into two main sections:

- Left Section (I ATTESA DI OPERAZIONE...):** This section features the Apulia Digital Maker logo at the top. Below it is a large white rectangular area containing the text "I ATTESA DI OPERAZIONE...". In the center of this area is the Apulia Digital Maker logo again. At the bottom are two buttons: a red-outlined "RIPRISTINA" button on the left and a blue "CONFERMA" button on the right.
- Right Section (INSEIRE I CAMPI NECESSARI ALLA VALIDAZIONE):** This section features a large red 'A' logo at the top. Below it is the text "INSEIRE I CAMPI NECESSARI ALLA VALIDAZIONE". It includes a placeholder for a profile icon with a shield containing a checkmark. To the right of this are two input fields: "BankCode" with the placeholder "Inserire il BankCode" and "PAN" with the placeholder "Inserire il PAN". Below these fields is a section titled "SELEZIONARE LE PERIFERICHE" with five options listed as grey buttons: "DISPENSATORE BANCONOTE", "MODULO DI VERSAMENTO", "STAMPANTE SCONTRINO", "SCANNER ASSEGNI", and "TASTIERA".

Interfaccia

IMPLEMENTAZIONE

Strutture e domini applicativi utilizzati

JS

ID	Nome	Descrizione	Abilitata	Tipo	Multisessione	Operazioni
13	Prelievo	Operazione per ritirare banconote	true	DEPOSIT	true	<button>delete</button>
19	Prelievo	Operazione per ritirare banconote	true	DEPOSIT	true	<button>delete</button>
25	Prelievo	Operazione per ritirare banconote	true	DEPOSIT	true	<button>delete</button>
31	Prelievo	Operazione per ritirare banconote	true	DEPOSIT	true	<button>delete</button>
37	Prelievo	Operazione per	true	DEPOSIT	true	<button>delete</button>

RIPRISTINA

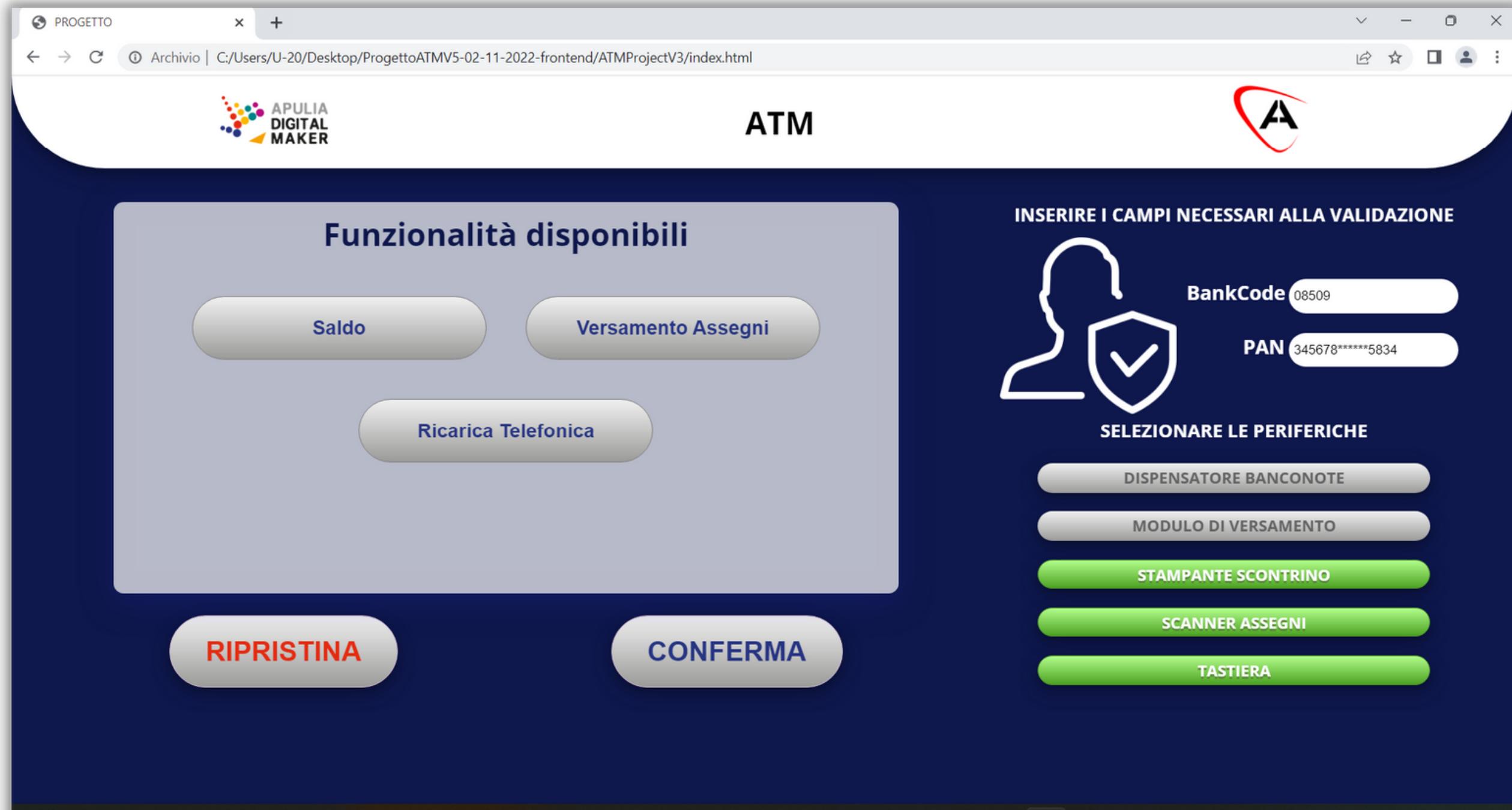
CONFERMA

MODIFICHE IMPLEMENTATE



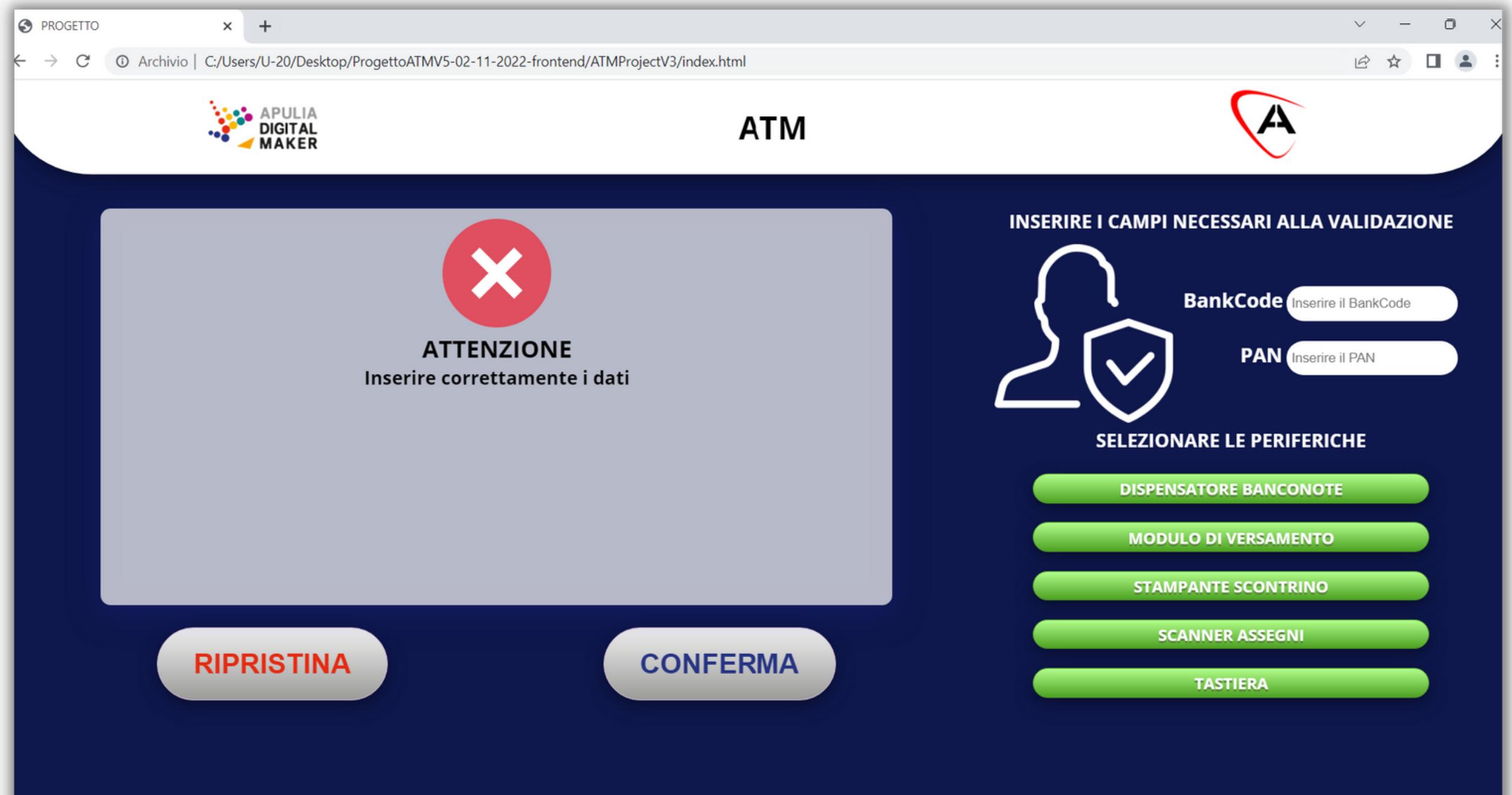
*Apportate modifiche
all'interfaccia tramite
sistemi Front-end*

MODIFICHE IMPLEMENTATE



Ora il sistema può riconoscere le periferiche selezionate e mostrare le operazioni possibili con la disponibilità di esse

MODIFICHE IMPLEMENTATE



Ora il sistema riconosce l'errato inserimento dei dati e notifica visivamente l'errore

Il team NekoDev



Daniele Cafagno
Back-end



Ilaria Ranieri
Front-end



Luca di Tria
Database admin

NekoDev

Daniele Cafagno, Ilaria Ranieri, Luca di Tria

Grazie!

