

---

**ITTS Plesso “Enrico Fermi” - PISTOIA**

**a.s. 2020/2021**

**Relazione di informatica:**

*Progetto monopattino*

**Proff.**

*Ing. Pasquale Silvestro*

*David Caramelli*

**Alunno:** *Giacomo Orsucci*

**Firma:**

*Giacomo  
Orsucci*

*Pistoia (PT), consegna per il 23/05/2021*

---

# Sommario

<b>Credenziali per accedere a gestione.php</b>	<b>1</b>
<b>Progettazione sistemistica</b>	<b>2</b>
Schema generale	2
Stazioni di noleggio	3
Infrastruttura server	3
Connettività	4
Tecnologie software scelte	4
<b>Progettazione informatica</b>	<b>5</b>
Modello concettuale	5
Modello logico	6
Vincoli imposti	6
<b>Normalizzazione</b>	<b>8</b>
Traccia	8
I forma normale	8
II forma normale	9
III forma normale	10

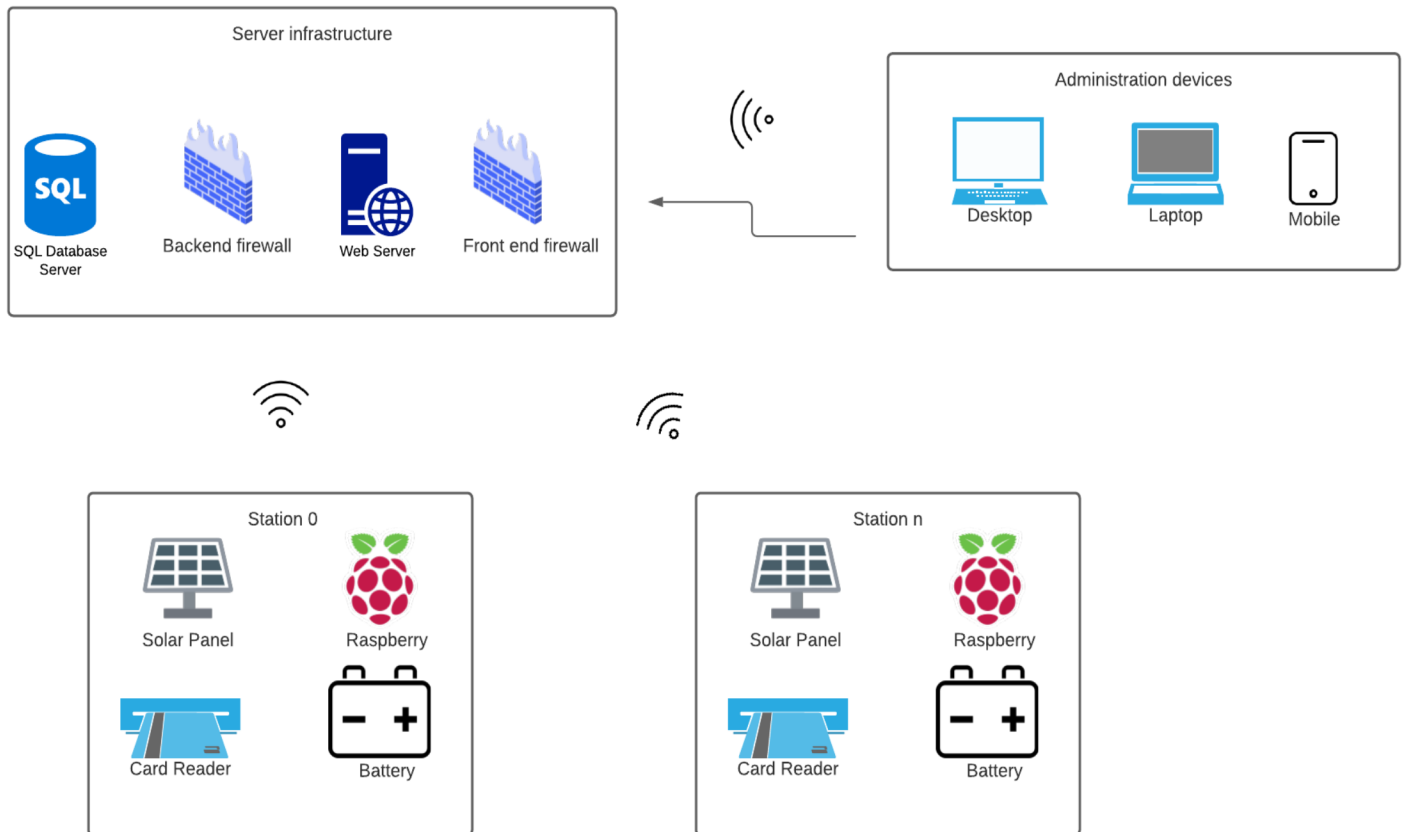
## Credenziali per accedere a gestione.php

- 1) username: Giacomo Orsucci    password: Giaco123    ruolo: superadmin
- 2) username: Mario                password: 123            ruolo: admin

I username non sono case sensitive, la password sì. Gli utenti superadmin possono accedere alla pagina di gestione degli account di amministrazione, gli altri no.

# Progettazione sistemistica

## Schema generale



---

## Stazioni di noleggio

Ogni stazione sarà dotata:

1. di un sistema di ricarica e di bloccaggio elettromeccanico dei monopattini (comprendente dei sensori di lettura dei loro codici RFID univoci)
2. di pannelli solari per la ricarica dei monopattini
3. di un Raspberry
4. di un lettore di smartcard

Il Raspberry di ogni stazione si occuperà del controllo delle ricarica dei monopattini e del loro conseguente rilascio per il noleggio una volta raggiunta la ricarica completa, della gestione dei pannelli solari, del lettore di smartcard e dei lettori dei codici RFID e dell'esecuzione delle query necessarie al momento del noleggio e della restituzione dei monopattini. Per garantire un'ulteriore sicurezza dei dati degli utenti l'hardware delle stazioni sarà a prova di scasso (in particolare i Raspberry).

La scelta dell'hardware è ricaduta su Raspberry perché offre una grande versatilità di utilizzo (per la gestione dei pannelli, dei monopattini, per l'esecuzione delle queries e quant'altro) e le prestazioni necessarie a prezzi contenuti.

## Infrastruttura server

Per quanto riguarda l'infrastruttura server si è deciso, (data anche la limitata potenza di calcolo richiesta per offrire il servizio), in controtendenza rispetto a ciò che avviene normalmente al giorno d'oggi, di non esternalizzare il servizio per permettere un maggiore controllo e contenimento dei dati trattati e dei costi.

Ed è composta da:

1. un distributore dei carichi
2. un server ospitante il database
3. un web server (anche gestore dei login e del bilanciamento dei carichi)
4. 2 firewall

I due firewall ed il collocamento dei server come descritto nello schema generale permettono una maggiore protezione del database contenente i dati degli utenti, tra cui dati sensibili come il numero della carta di credito e la password degli amministratori. In particolare, il web server si troverà nella DMZ (demilitarized zone) che si ha tra i due firewall così che sia accessibile dall'esterno per i servizi necessari, mentre il server ospitante il database, appunto, sarà posto "dietro" ad essi per garantire una maggiore sicurezza. Inoltre, si è deciso di far gestire i login ed il bilanciamento dei carichi al web server e non a server dedicati perché, considerati la natura del

---

servizio e del contesto, non si prevedono grossi carichi di lavoro tali da giustificare (anche economicamente) una tale complessità di infrastruttura.

## Connettività

Considerate le basse prestazioni richieste, gli elevati costi e la difficoltà di installazioni sotterranee dovute al rilievo artistico e culturale delle città italiane si è escluso di garantire la connettività via cavo. Si è scelto quindi di usare la tecnologia wireless. Si sfrutterà quella esistente se già presente sul territorio, altrimenti si metterà in piedi una nuova infrastruttura che avrà il doppio scopo di offrire connettività alle stazioni di noleggio ed ai cittadini del Comune di Pistoia. Considerata però l'intrinseca insicurezza della tecnologia si prevede la realizzazione di una VPN tra le stazioni di noleggio ed il database che garantisca l'autenticazione e la segretezza delle comunicazioni. Le stazioni di amministrazione invece non faranno parte delle VPN, ma saranno in grado di accedere alla piattaforma di gestione come gli utenti normali. In questo ultimo caso, dato che non è richiesto l'accesso diretto ai dati del database (gli amministratori contatteranno il web server che poi, a sua volta, contatterà il server con il database quando necessario. Al contrario le stazioni di noleggio hanno bisogno di dialogare direttamente con la base di dati) la sicurezza della connessione ai dati del database è garantita dai firewall presenti nella rete dei server.

## Tecnologie software scelte

### Tecnologie backend per la gestione del database

- SQL
- MySQL

### Tecnologie backend per la gestione della piattaforma web

- Apache
- PHP
- Software per la gestione dei login e dei carichi

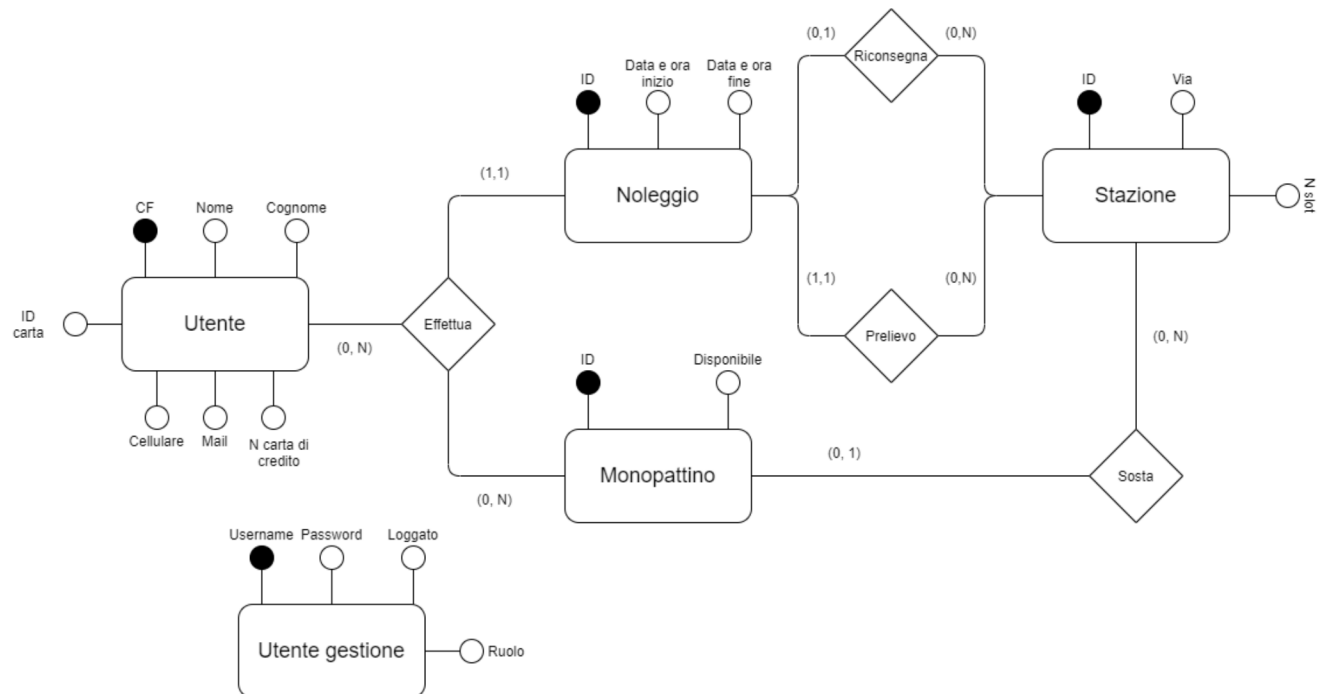
### Tecnologie lato client

- JavaScript
- HTML
- CSS (framework bootstrap per la Pubblica Amministrazione)

Si sottolinea che si è scelto il framework Bootstrap creato appositamente per la pubblica amministrazione per allineare, anche graficamente, la piattaforma web progettata alle altre offerte dalla PA.

# Progettazione informatica

## Modello concettuale



La progettazione del database è stata realizzata tenendo conto di tutte le richieste e le necessità evidenziata dalla traccia, inoltre possiamo notare che:

1. l'entità "Utente gestione" è distaccata dalla tabella "Utente". Questo perché si è deciso di separare gli utenti che usufruiscono del servizio da quelli autorizzati ad amministrare la piattaforma. In questo modo è sì necessario avere un account per ogni tipo di utilizzo che viene fatto della piattaforma (servizio di noleggio o servizio di amministrazione), ma si assicura anche una maggiore "pulizia" delle tabelle. Unendo le due tabelle si avrebbe infatti avuto, per gli utenti esclusivamente registrati per il noleggio, password uguale a "null" e sarebbe stato più complesso gestire i ruoli, in quanto sarebbe esistito anche un ruolo per il noleggio.
2. sono presenti due relazioni tra "Noleggio" e "Stazione", questo perché è necessario avere sia l'id della stazione di noleggio che quello della stazione di restituzione per poi permettere, in particolare, di vedere (lato amministrazione) da quali stazioni sono state effettuate più prenotazioni e quali monopattini sono ancora in uso e quindi non restituiti.

## Modello logico

Il modello concettuale precedentemente presentato viene tradotto nel seguente modello logico:

**Utente**(CE, Nome, Cognome, Cellulare, Mail, NCartaCredito, IDCarta)

**Monopattino**(ID, Disponibile, IDStazioneSosta\*)

**Noleggio**(ID, CFUtente\*, IDMonopattino\*, DataOrainizio, DataOraFine, IDStazioneNoleggio\*, IDStazioneRestituzione\*)

**Stazione**(ID, Via, NSlot)

**UtenteGestione**(Username, Password, Loggato, Ruolo)

## Vincoli imposti

Entità/ associazione	Nome campo	Chiave	Tipo	Vincoli espliciti	Default /attributi
<b>Utente</b>	CF	Primary	Varchar(16)	/	/
	Nome	/	Varchar(30)	Not null	/
	Cognome	/	Varchar(30)	Not null	/
	Cellulare	/	Varchar(10)	Not null	/
	Mail	/	Varchar(320)	Not null	/
	NCartaCredito	/	Varchar(32)	Not null	md5()
	IDCarta	/	Int	Unique	Auto_increment()
<b>Monopattino</b>	ID	Primary	Int	/	Auto_increment()
	Disponibile	/	Boolean	Not null	/
	IDStazioneSosta	Foreign	Int	/	/

<b>Noleggior</b>	ID	Primary	Int	/	Auto_increment()
	CFUtente	Foreign	Varchar(16)	Not null	/
	IDMonopattino	Foreign	Int	Not null	/
	DataOraInizio	/	DateTime	Not null	/
	DataOraFine	/	DateTime	/	/
	IDStazioneNoleggior	Foreign	Int	Not null	/
	IDStazioneRestituzione	Foreign	Int	/	/
<b>Stazione</b>	ID	Primary	Int	/	Auto_increment()
	Via	/	Varchar(38)	Not null	/
	NSlot	/	Int	Not null	/
<b>Utente Gestione</b>	Username	Primary	Varchar(30)	/	/
	Password	/	varchar(32)	Not null, unique	md5()
	Loggato	/	Boolean	Not null	/
	Ruolo	/	Enum("admin", "superadmin")	Not null	/

Sui tipi di dati scelti ed i vincoli imposti è necessario spiegare e motivare maggiormente le scelte fatte:

1. In generale si è cercato di evitare i valori "null" progettando il database a tal fine ed imponendo il vincolo "not null" quando possibile. I valori "null" sono accettati solamente quando strettamente necessari alle operazioni di gestione richieste.
2. Quando era necessario gestire delle numerazioni senza la necessità di permettere la definizione di numerazioni preesistenti è stato usato l'attributo "auto\_increment()" per demandare la loro gestione al sistema.
3. L'entità "Utente" presenta sia il campo "Mail" che "Cellulare" per permettere una maggiore possibilità di comunicazione con l'utente. La email, per esempio, potrebbe essere utilizzata per le comunicazioni periodiche e automatiche degli acconti per il servizio, mentre il numero di cellulare potrebbe essere usato se si necessita di contattare urgentemente l'utente.



4. Ad ogni utente è associato un id univoco (unique) della corrispondente smartcard per assicurare il bloccaggio della carta e la conseguente assegnazione di una nuova carta con un nuovo id in caso di smarrimento o di furto della stessa evitando la perdita dell'account.
5. Alcuni campi che potrebbero essere ritenuti "int" come il numero della carta di credito o il cellulare sono di tipo "varchar" per svincolarsi dal limite imposto dal tipo "int" e che potrebbe essere raggiunto involontariamente dal numero rappresentate le cifre inserite.
6. Il campo "Ruolo" dell'entità "Utente gestione" è di tipo "enum" per permettere l'inserimento di soli due ruoli specifici, ovvero "admin" e "superadmin". Il primo è autorizzato ad amministrare la piattaforma di gestione, ma non gli account degli amministratori, mentre il secondo entrambi.
7. L'entità "Utente gestione" presenta il campo "loggato" perché, al momento dell'accesso alla piattaforma di ogni amministratore, si controlla se esso è già loggato su altri browser o dispositivi così da impedire accessi multipli e garantire un maggiore livello di sicurezza per l'account.
8. La sicurezza dei dati sensibili come la password degli account di amministrazione o i numeri di carta di credito è garantita salvandoli sotto forma di dati criptati mediante la funzione md5().

## Normalizzazione

### Traccia

Data la seguente entità:

**QUADRO (Cod\_Quadro, Cod\_Museo, Titolo\_Quadro, Nome\_Museo, Citta\_Museo, Prezzo, DataInizioEsposizione, DataFineEsposizione)**

si verifichino le proprietà di normalizzazione e si proponga, eventualmente, uno schema equivalente che rispetti la terza forma normale, motivando le scelte effettuate.

### I forma normale

Affinché la relazione risulti in prima forma normale è necessario che:

- gli attributi siano tutti atomici (non composti o multivalore)
- siano presenti le primary key
- i valori della stessa colonna appartengano allo stesso dominio (siano dello stesso tipo)
- l'ordine con il quale le righe vengono inserite nella tabella sia irrilevante

presupponendo di utilizzare un DBMS (in questo caso MySQL) gli ultimi due punti sono implicitamente garantiti. Osservando l'entità fornita si può inoltre notare come anche il primo punto sia già soddisfatto; non sono presenti però chiavi primarie.

## Assegnazione delle chiavi primarie per la I forma normale

Si prende per assunto che il database analizzato operi in un contesto nazionale o internazionale, pertanto si considerano i codici dei musei e dei quadri in tale contesto e non si accettano numerazioni diverse a livello locale (municipale, provinciale, regionale ecc). Si pone inoltre l'attenzione sull'entità "Quadro" e sulle sue esposizioni pertanto, si elevano a chiavi primarie gli attributi "Cod\_Quadro", "Cod\_Museo" e "DataInizioEsposizione", permettendo così anche l'esposizione dello stesso quadro in musei diversi, ma in date diverse. Si sottolinea inoltre che si è deciso di elevare a chiavi primarie gli attributi "Cod\_Quadro", "Cod\_Museo" e non "Nome\_Museo" e "Titolo\_Quadro" per permettere, data anche la natura del contesto esaminato, l'esistenza di quadri con lo stesso titolo e musei con lo stesso nome, ma di fatto diversi. L'entità studiata, in I forma normale, si presenta nel seguente modo:

**QUADRO (Cod\_Quadro, Cod\_Museo, Titolo\_Quadro, Nome\_Museo, Citta\_Museo, Prezzo, DataInizioEsposizione, DataFineEsposizione)**

### Il forma normale

Affinché la relazione risulti in seconda forma normale:

- E' necessario che la tabella si trovi nella I forma normale
  - Se la relazione non presenta chiavi composte si trova già in II forma normale
  - Se sono presenti chiavi composte è necessario, se presenti, eliminare le dipendenze funzionali presenti tra le parti delle chiavi e gli attributi non chiave

Nel caso studiato troviamo una chiave composta da "Cod\_Quadro", "Cod\_Museo" e "DataInizioEsposizione". Quindi, verificando le dipendenze funzionali, troviamo le seguenti:

(Cod\_Quadro) → (Titolo\_Quadro, Prezzo);

(Cod\_Museo) → (Nome\_Museo, Città\_Museo);

A seguito dell'eliminazione delle dipendenze funzionali troviamo le seguenti entità e le seguenti relazioni:

**Quadro(Codice\_Quadro, Titolo\_Quadro, Prezzo);**

**Museo(Codice\_Museo, Nome\_Museo, Città\_Museo);**

**Esposizione(Codice\_Quadro\*, Codice\_Museo\*, DataInizioEsposizione, DataFineEsposizione);**

---

### III forma normale

Per portare la relazione in terza forma normale è necessario, a partire dalla II forma normale, verificare ed eliminare la presenza delle dipendenze funzionali transitive tra gli attributi non chiave. L'entità studiata (a partire dalla II forma normale), non presentandone, si trova già in III forma normale.