

SMART VENDING MACHINE

TEAM DI SVILUPPO

LUCIANO MESSINEO MATRICOLA: ***0739954***

INDEX

GENERAL DESCRIPTION	2
SYSTEM ARCHITECTURE	2
TECNOLOGIE UTILIZZATE	2
FUNCTIONAL REQUIREMENTS	3
GESTIONE DELL'UTENZA	3
FUNZIONALITA'	5
PATTERN ARCHITETTURALE E STRUTTURA DEL PROGETTO	8
DATABASE	9
VERSIONING DEL CODICE E GESTIONE DEL PROGETTO	12

GENERAL DESCRIPTION

IL SEGUENTE PROGETTO HA LO SCOPO DI MIGLIORARE IL SISTEMA DI VENDITA DI PRODOTTI TRAMITE LE CLASSICHE MACCHINETTE AUTOMATICHE, INTRODUCENDO UN SISTEMA SMART DI COLLEGAMENTO TRAMITE APP ALLE STESSE, TRAMITE L'APP L'UTENTE SARA' IN GRADO DI COLLEGARSI AD UNA MACCHINETTA PER EFFETTUARE GLI ACQUISTI, GESTIRE IL PROPRIO WALLET VIRTUALE E PERMETTE AGLI AMMINISTRATORI DI AVERE UNA VISTA A 360° SU TUTTE LE COMPONENTI DEL SISTEMA TRAMITE DELLE APPOSITE VISTE.

SYSTEM ARCHITECTURE

TECNOLOGIE UTILIZZATE

IL CLIENT E' STATO SVILUPPATO UTILIZZANDO:

1. HTML, CSS, JAVASCRIPT
2. JQUERY
3. PER QUANTO RIGUARDA LE CHIAMATE AL BACK END SONO STATE UTILIZZATE LE FUNZIONI \$.get E \$.post
4. JAVA SERVER PAGES
5. SCRIPTLET

IL SERVER E' STATO SVILUPPATO UTILIZZANDO:

1. SERVLET
2. MYSQL CONNECTOR
3. APACHE TOMCAT (SERVER ENGINE)
4. NETBEANS
5. ORG.JSON (LIBRERIA JAVA PER LA GESTIONE DI RICHIESTE E CREAZIONE DI RISPOSTE IN FORMATO JSON)
6. JAVA.SECURITY (LIBRERIA JAVA PER LA CIFRATURA DELLA PASSWORD TRAMITE ALGORITMO DI HASHING SHA256)
7. QUARTZ : LIBRERIA CHE PERMETTE DI GESTIRE I JOB UNA VOLTA CHE IL SERVLET CONTEXT VIENE AVVIATO.
8. GIT (PER GESTIRE IL VERSIONING DEL CODICE)

FUNCTIONAL REQUIREMENTS

GENERALI

GESTIONE DELL'UTENZA

AMMINISTRATORE : A QUESTA FIGURA SONO CONCESSE TUTTE LE POSSIBILI OPERAZIONI, OVVERO:

1. GESTIONE DATI PERSONALI
2. GESTIONE WALLET
3. VISUALIZZAZIONE ACQUISTI
4. GESTIONE UTENTI
5. GESTIONE MACCHINETTE
6. GESTIONE PRODOTTI

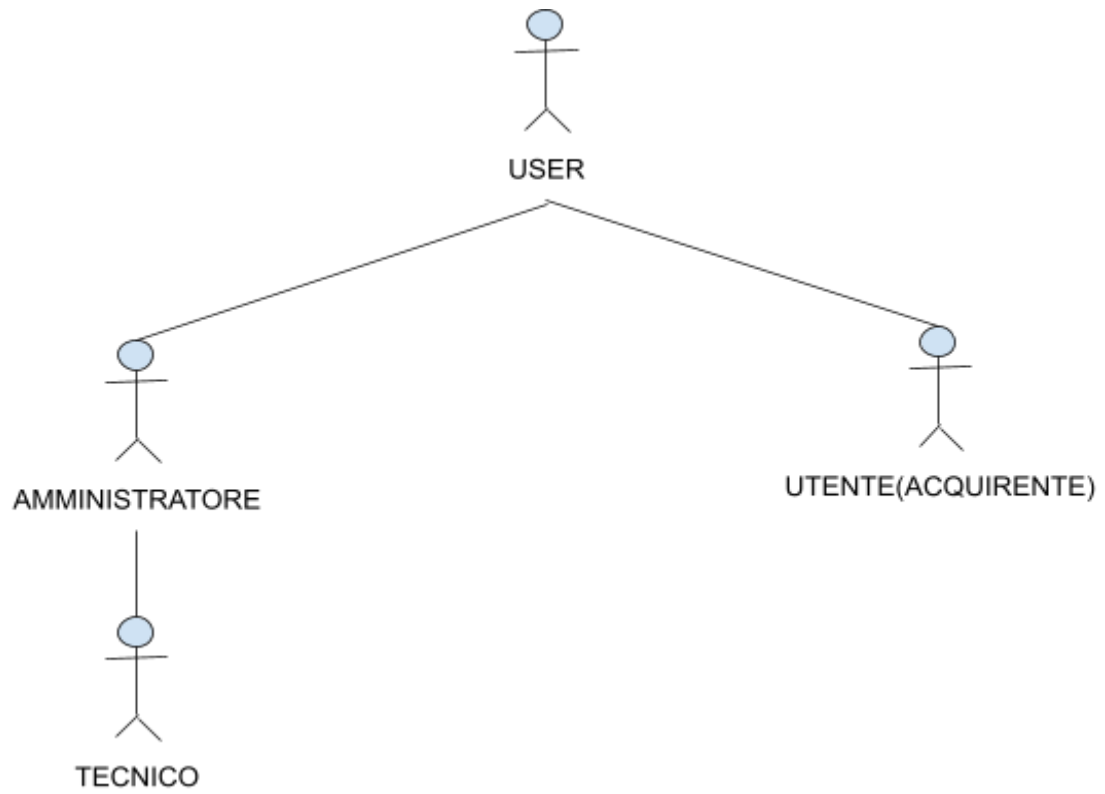
TECNICO: SVOLGE DELLE FUNZIONI LIMITATE RISPETTO ALL'ADMIN

1. GESTIONE DATI PERSONALI
2. GESTIONE WALLET
3. VISUALIZZAZIONE ACQUISTI
4. GESTIONE MACCHINETTE
5. GESTIONE PRODOTTI

UTENTE:

1. GESTIONE DATI PERSONALI
2. GESTIONE WALLET
3. VISUALIZZAZIONE ACQUISTI

SCHEMA:



FUNZIONALITA'

HOME PAGE E REGISTRAZIONE

AL PRIMO AVVIO L'APPLICATIVO MOSTRERA' ALL'UTENTE LA HOME-PAGE, LA QUALE PERMETTERA' DI EFFETTUARE IL LOGIN TRAMITE IL FORM, E ,TRAMITE LA NAV-BAR, POTRA' REGISTRARSI E RICARICARE LA HOME-PAGE.

PAGINA PERSONALE

UNA VOLTA EFFETTUATO IL LOGIN L'UTENTE VERRA' REINDIRIZZATO VERSO LA PAGINA PERSONALE, LA QUALE MOSTRERA' UNA SERIE DI FUNZIONALITA' GESTITE TRAMITE UN MENU LATERALE COSI' COMPOSTO:

1. GESTIONE DATI PERSONALI

QUI L'UTENTE PUO' VISUALIZZARE E MODIFICARE NOME, COGNOME, EMAIL E PASSWORD , INOLTRE PUO' ELIMINARE IL PROPRIO ACCOUNT

2. GESTIONE DEL WALLET

AD OGNI UTENTE, APPENA REGISTRATO, VERRA' ASSEGNATO UN WALLET VIRTUALE SUL QUALE EGLI POTRA' EFFETTUARE LE RICARICHE DI UN IMPORTO MASSIMO DI 100 EURO.

INOLTRE E' PRESENTE UNA SEZIONE RELATIVA ALLA VISUALIZZAZIONE DI TUTTI I MOVIMENTI , OVVERO RICARICHE ED ACQUISTI EFFETTUATI.

3. VISUALIZZAZIONE DEGLI ACQUISTI

VISTA SEMPLIFICATA CHE PERMETTE ALL'UTENTE DI POTER VISUALIZZARE TUTTI I PROPRI ACQUISTI

4. GESTIONE DEGLI UTENTI

LA SEGUENTE VISTA SARA' VISUALIZZABILE SOLTANTO DALL'ADMIN, IL QUALE POTRA' INSERIRE E CANCELLARE UN UTENTE, MODIFICARE LA TIPOLOGIA DI UTENZA.

5. GESTIONE DELLE MACCHINETTE

LA SEGUENTE VISTA SARA' VISUALIZZABILE SIA DALL'ADMIN CHE DAL TECNICO.

QUI SARA' POSSIBILE INSERIRE/CANCELLARE UNA MACCHINETTA, E

MODIFICARNE LO STATUS.

logica di stato della macchinetta:

A. STATO ACTIVE

LA MACCHINETTA **DOPO ESSERE STATA RICARICATA** VIENE ATTIVATA PER ESSERE INSERITA NELLA LISTA DELLE MACCHINETTE DISPONIBILI PER GLI ACQUISTI.
SE LA SUA CAPACITA' ATTUALE FOSSE PARI A ZERO ALLORA NON POTRA' ESSERE SETTATA COME ATTIVA.

B. STATO DISABLED

LA MACCHINETTA SI TROVA NEL SEGUENTE STATO QUANDO VIENE CREATA PER LA PRIMA VOLTA, QUANDO LA SUA CAPACITA' ATTUALE SARA' PARI A ZERO, OPPURE QUANDO IL TECNICO IMPOSTERA' IL SEGUENTE STATO PER LAVORI DI MANUTENZIONE.

C. STATO OCCUPIED

LA MACCHINETTA SI TROVERA' IN TALE STATUS FIN QUANDO UN QUALSIASI UTENTE LA STARA' UTILIZZANDO.

D. STATO FREE

LA MACCHINETTA SI TROVERA' IN TALE STATO QUANDO L'UTENTE CHE LA STAVA UTILIZZANDO LA RILASCIA PERCHE' EFFETTUA IL LOGOUT O PERCHE' TORNA ALLA PROPRIA PAGINA PERSONALE PER EFFETTUARE ALTRE OPERAZIONI, COSI' DA EVITARE DI OCCUPARLA PER TROPPO TEMPO.

Ricarica della macchinetta

UNA VOLTA PREMUTO IL PULSANTE DI RICARICA, VERRA' MOSTRATO UN FORM CONTENENTE LO STATO DEI PRODOTTI DELLA MACCHINETTA, MOSTRANDO LE QUANTITA' PER OGNUNO DI ESSI.

QUANDO VIENE INSERITO L'ID DI UN PRODOTTO IL SISTEMA CONTROLLERA' CHE ESSO ESISTA, E IN CASO NEGATIVO VERRA' MOSTRATA UNA NOTIFICA. LA QUANTITA' MASSIMA DI OGNI PRODOTTO E' PARI A 5 UNITA'.
(PER SEMPLICITA' IMPLEMENTATIVA OGNI MACCHINETTA CONTIENE AL MASSIMO 4 PRODOTTI E AVRA' UNA CAPACITA' MASSIMA DI 20 UNITA', CIOE' 5 PER OGNI PRODOTTO).

6. GESTIONE DEI PRODOTTI

TRAMITE LA SEGUENTE VISTA SIA ADMIN CHE TECNICO POSSONO INSERIRE, CANCELLARE E MODIFICARE UN SINGOLO PRODOTTO.

7. ACQUISTO

TRAMITE LA SEGUENTE VISTA QUALSIASI UTENTE PUO' VISUALIZZARE TUTTE LE MACCHINETTE DISPONIBILI; SARA' POSSIBILE CONNETTERSI AD UNA DI QUESTE SEMPLICEMENTE SELEZIONANDO IL PULSANTE "CONNETTITI". UNA VOLTA PREMUTO IL PULSANTE L'UTENTE VERRA' REINDIRIZZATO VERSO LA PAGINA DEGLI ACQUISTI, DOVE VERRANNO MOSTRATI I PRODOTTI DELLA MACCHINETTA.

PAGINA DEGLI ACQUISTI

UNA VOLTA ESSERSI CONNESSO ALLA MACCHINETTA L'UTENTE POTRA' ACQUISTARE UN PRODOTTO DELLA STESSA.

SE IL SALDO DEL WALLET FOSSE NON SUFFICIENTE, IL SISTEMA NOTIFICHERA' ALL'UTENTE CHE DOVRA' EFFETTUARE UNA RICARICA.

SE LA QUANTITA' DI UN PRODOTTO NON FOSSE SUFFICIENTE IL SISTEMA NOTIFICHERA' ALL'UTENTE CHE ESSO NON SARA' DISPONIBILE E DISABILITERA' IL RELATIVO PULSANTE DI ACQUISTO.

UNA VOLTA EFFETTUATO L'ACQUISTO L'UTENTE PUO' EFFETTUARE IL LOGOUT OPPURE ANDARE ALLA PAGINA PERSONALE.

PATTERN ARCHITETTURALE E STRUTTURA DEL PROGETTO

E' STATO SCELTO DI UTILIZZARE IL PATTERN MVC PER LA GESTIONE DEL PROGETTO, IN PARTICOLARE:

A. VIEW

1. META-INF
2. VIEW

CONTIENE LE PAGINE PRINCIPALI, LE JSP CHE IMPLEMENTANO LE COMPONENTI HTML, I FILE CSS E LE IMMAGINI UTILIZZATE NEL SITO

3. WEB-INF
4. INDEX.JSP

B. MODEL

CONTIENE TUTTE LE CLASSI CHE FORNISCONO UNA RAPPRESENTAZIONE DELLE TABELLE PRESENTI NEL DATABASE E LE CLASSI SERVICES CHE IMPLEMENTANO I SERVIZI CHE PERMETTONO DI MANIPOLARE I DATI.

INOLTRE SARA' PRESENTE UNA CLASSE CHE IMPLEMENTA IL CONNETTORE AL DATABASE.

C. CONTROLLER

CONTIENE LE CLASSI JAVA CHE IMPLEMENTANO LE SERVLET PER LA GESTIONE DELLE RICHIESTE GET E POST

SARANNO INOLTRE PRESENTI DEGLI ULTERIORI PACKAGE:

D. SECURITY

CONTIENE LA CLASSE CHE PERMETTE DI GESTIRE L'ALGORITMO DI CRITTOGRAFIA DELLA PASSWORD. IN PARTICOLARE VIENE UTILIZZATO L'ALGORITMO DI HASHING SHA-256 PER OTTENERE UN HASH CHE VERRA' MEMORIZZATO SUL DATABASE.

F. FILTERS

CONTIENE LE CLASSI PER LA PROTEZIONE DELLE ROTTE DEL SITO, PER QUANTO RIGUARDA TUTTE LE FUNZIONALITA' LEGATE ALLA SESSIONE E ALLA FASE DI AUTENTICAZIONE.

G.JOBS

CONTIENE TUTTE LE CLASSI CHE IMPLEMENTANO I VARI JOB, IN PARTICOLARE TROVIAMO IL JOB “ModifyStatusJob” CHE CONTROLLA OGNI MINUTO SE LO STATO DI OGNI MACCHINETTA RIMANE IN PENDING SU Occupied. INFATTI SE LA MACCHINETTA E' IN TALE STATO PER PIU' DI 10 MINUTI ALLORA VIENE SETTATA ALLO STATO free.

I FILE DI CONFIGURAZIONE PER I JOBS SI POSSONO TROVARE SOTTO LA CARTELLA /src/main/resources, IN PARTICOLARE TROVIAMO IL FILE quartz-properties, OVVERO IL FILE DI CONFIGURAZIONE DI QUARTZ E IL FILE quartz-jobs.xml, CHE VIENE FORNITO AL PRECEDENTE E CONTIENE LA DICHIARAZIONE DEI VARI JOBS E DEI TRIGGER CON LE RELATIVE IMPOSTAZIONI.

H.UTILS

CONTIENE TUTTE LE CLASSI CHE IMPLEMENTANO LE FUNZIONI DI SUPPORTO PER LE SERVLET.

DATABASE

PER QUESTO PROGETTO LA SCELTA E' RICADUTA NELL'UTILIZZO DI UN DATABASE RELAZIONALE, OVVERO MYSQL.

CREAZIONE CONCETTUALE DATABASE <SMARTVENDINGMACHINEDB>

USERS : ID, NAME, SURNAME, EMAIL, PASSWORD, TYPE, WALLET BALANCE

PRODUCTS : ID , NAME, TYPOLOGY,PRICE

MACHINES: *ID*, STATUS, NAME, MAX CAPACITY, ACTUAL CAPACITY, OCCUPIEDSINCE

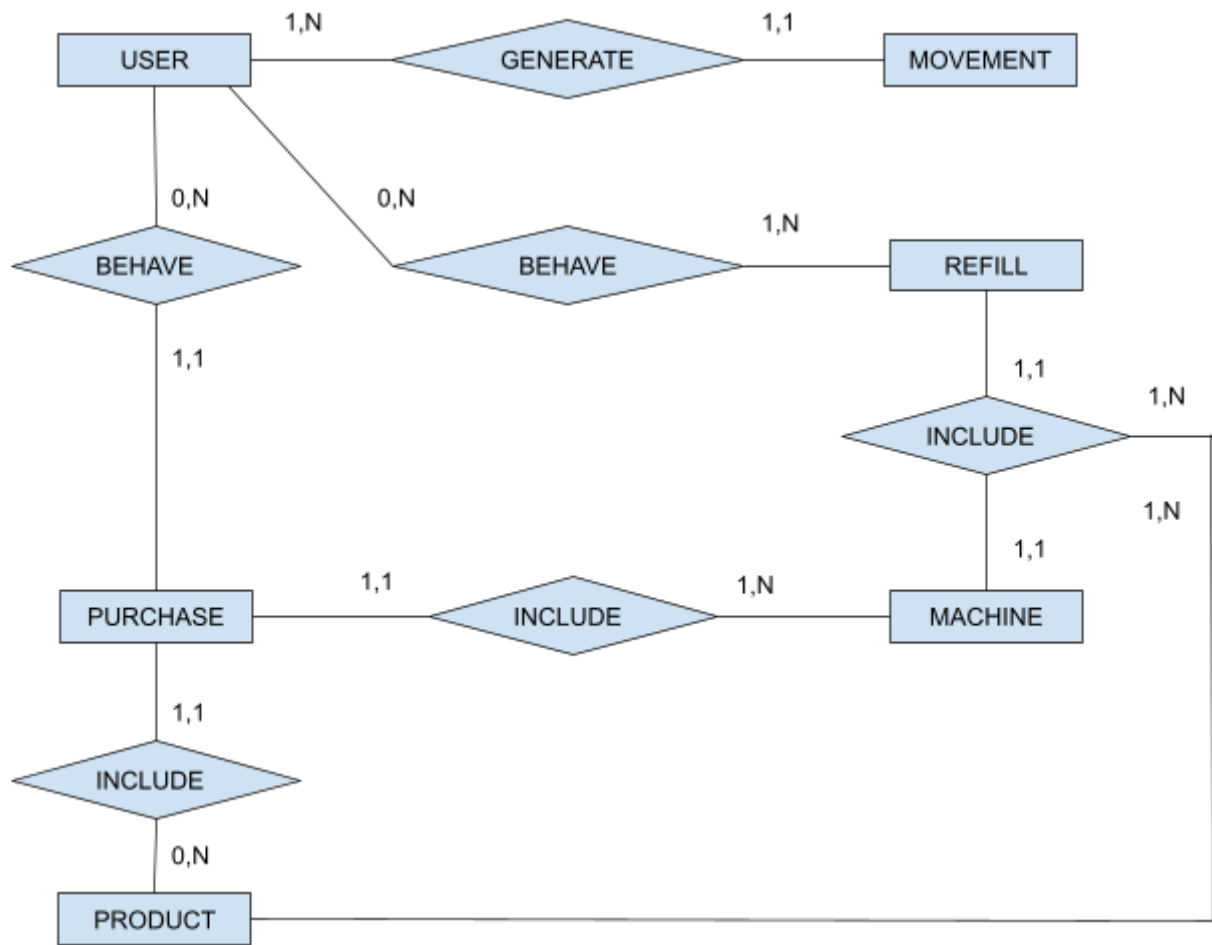
PURCHASES: ID,DATE,COST,USERID,PRODUCTID,MACHINEID

FK(USERID,PRODUCTID,MACHINEID)

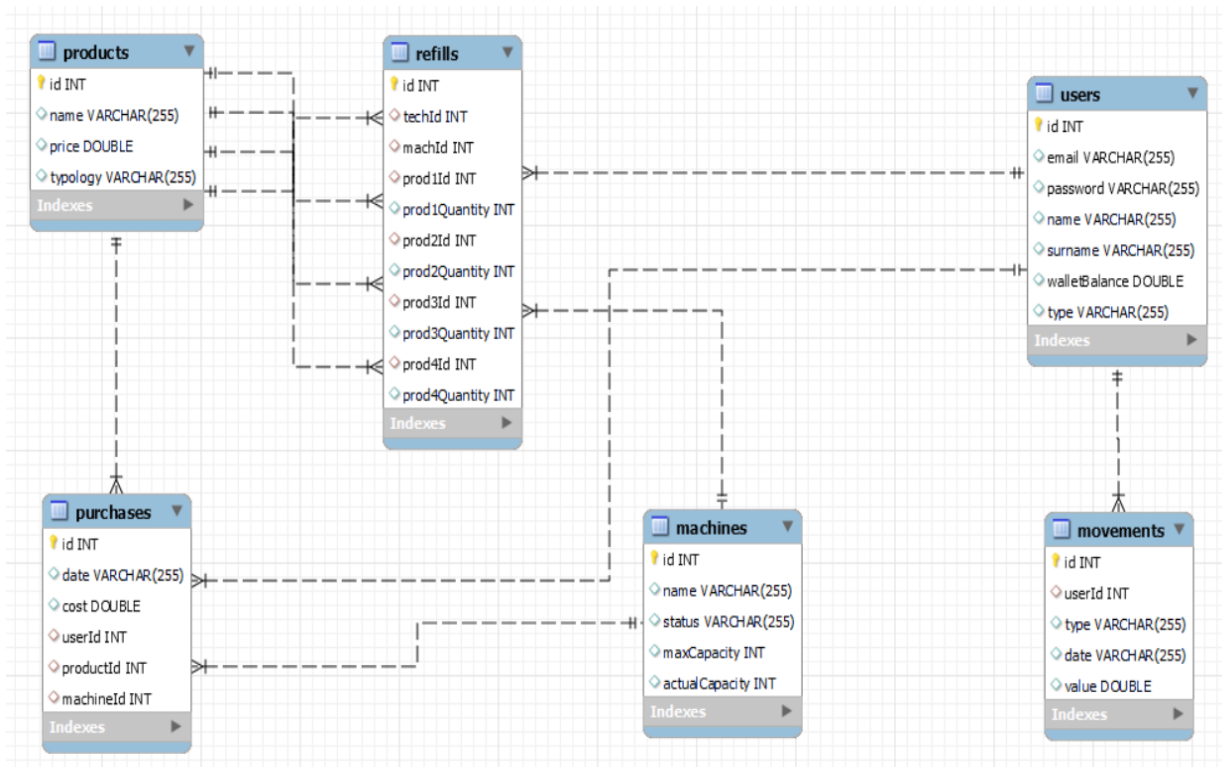
MOVEMENTS: ID,USERID,TYPE,DATE,VALUE FK(USERID)

REFILLS: ID, TECHID,MACHINEID,PROD1ID,PROD1QUANTITY,
PROD2ID,PROD2QUANTITY,
PROD3ID,PROD3QUANTITY,
PROD4ID,PROD4QUANTITY

FK(TECHID,MACHINEID,PROD1ID,PROD2ID,PROD3ID,PROD4ID)



CREAZIONE LOGICA DEL DB



VERSIONING DEL CODICE E GESTIONE DEL PROGETTO

IL VERSIONING DEL CODICE SORGENTE E DI TUTTI I DOCUMENTI E' STATO GESTITO TRAMITE GIT, EFFETTUANDO I VARI COMMIT IN DIVERSI STEP:

1. FASE DI DEV

CREAZIONE DELLE COMPONENTI PRINCIPALI DEL SITO E CREAZIONE DEI PRIMI MOCKUP

2. FASE DI TESTING

GESTIONE DEGLI ERRORI E DELLE ECCEZIONI, INTRODUZIONE DI ULTERIORI CONTROLLI E PRIMA FASE DI BUG-FIX.

CAMBIAMENTI DELLO STILE E DELL'INTERFACCIA GRAFICA.

PRIMO CONTROLLO SULLA CONSISTENZA DEI DATI

3. FASE DI STAGING

ULTIMA FASE DI BUG-FIX E CONTROLLO DELLA CONSISTENZA DEI DATI.

4. RILASCIO FINALE

E' POSSIBILE VISUALIZZARE IL CODICE SORGENTE E I DOCUMENTI ANNESSI ALLA SEGUENTE REPOSITORY SU GITHUB:

https://github.com/LucianoMessineo096/Vending_Machine/tree/master