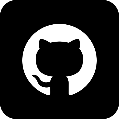
Ticketing System



Proiectul este o aplicatie pentru gestionarea biletelor de evenimente, care ofera functionalitati de:

* **Achizitie Ticket** (Trei categorii de tickete: “Premium”, “Mid”, “Low”).
* **Generare Bilet** (QR code, informatii client) in format .pdf.
* **Activare Ticket** (pe baza seriei ticket generate automat si CNP-ul clientului).
* **Editare/Salvare** date “telefon”, “email” client in baza de date.
* **Posibilitatea Clientului** de a isi sterge datele din baza de date.

Achizitia unui ticket este implementata pe baza unui Cod Numeric Personal (CNP) unic al clientului, si posibilitatea achizitiei unui singur ticket (1 buc.) din cele trei categorii.

Astfel pentru achizitia unui ticket de tip:

* ***Premium***: costul biletului este de : 200 lei/buc. Nr locuri disponibile: 10.
* ***Mid***: costul biletului este de : 100 lei/buc. Nr locuri disponibile: 20.
* ***Low***: costul biletului este de : 50 lei/buc. Nr locuri disponibile: 30.

Aplicatia in interfata grafica este structurata in doua parti (taburi):

# Tabul 1 – “Buy Ticket”.

Patru campuri marcate cu \* (obligatorii) pentru “Nume”, “Prenume”, “Cnp”, “Email”, si un camp optional “Telefon”. Afisarea unui nou panou ***(depinde de corectitudinea datelor introduse)*** cu selectia achizitiei unui tip de ticket, afisare cantitate/nr. locuri disponibile, buton cu afisare pretul ticketului, cu rolul de innregistrare a achizitiei in baza de date.

# Tabul 2 – “Edit/Activate”.

Doua campuri obligatorii pentru Serie Ticket si CNP, afisare un nou panou cu posibilitatea de a edite nr. telefon si email. Afisare informatii client cu nume , prenume, bilet validat/nevalidat si butoane de a modifica/sterge client, precum si buton de validare

# Tehnologii Utilizate

* Java (JDK 21)

Proiectul este dezvoltat in Java, un limbaj de programare orientat pe obiecte.

* Maven

Ca instrument de build si gestionarea dependentelor.

* MySQL

Ca si baza de date relationala, pentru stocarea datelor aplicatiei, inclusiv a informatiilor de la clienti, tranzactii si bilete.

* Python

Pentru a genera QR code, implementare date client intr-un fisier .pdf

# Structura Aplicatiei

***Pachetul org.example.database***

Clasa ***DataBase***: Gestiomneaza conexiunea la baza de date. Este utilizata pentru a executa interogari SQL, conexiuni si alte operatiuni legate de baze de date.

***Pachetul org.example.model***

Clasa ***InfoUser***: Reprezinta informatii despre utilizatori, interogari din baza de date.

Clasa ***StocBilete***: Modeleaza stocul, inventarul biletelor disponibile.

Clasa ***User***: Reprezinta un utilizator, contine atribute precum nume, email, si alte detalii personale.

***Pachetul org.example.services***

Clasa ***SystemDeliveryReceive***: Gestioneza serviciile de livrare sau primire a informatiilor, o interfata intre baza de date si clasele din ***Pachetul org.example.model.***

***Pachetul org.example.ui***

Clasa ***TicketsSystemManagement***: Componenta principala a interfetei grafice a utilizatorului (GUI), care gestioneaza logica aplicatiei bazata pe toate evenimentele aplicatiei, cum ar fi apasarea de butoane, completarea campurilor.

Clasa ***MainApp***: Punctul de intrare al aplicatiei (public static void main). Aceasta initializeaza componentele principale ale aplicatiei si afiseaza interfata grafica.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Dupa stergerere…

