import random

import string

import sqlite3

from fpdf import FPDF

class User:

    '''Represents a user that buys a cinema seat'''

    def \_\_init\_\_(self, name):

        self.name = name

    def buy(self, seat, card):

        '''Buys the ticket if the card is valid'''

        if seat.is\_free():

            if card.validate(price = seat.get\_price()):

                seat.occupy()

                ticket = Ticket(user = self, price = seat.get\_price(), seat\_number = seat.seat\_id)

                ticket.to\_pdf()

                return 'Seat purchase was successful!'

            else:

                return 'There was a problem with your card.'

        else:

            return 'Seat is taken.'

class Seat:

    '''Represents a cinema seat that can be taken by the user'''

    database = 'cinema.db'

    def \_\_init\_\_(self, seat\_id):

        self.seat\_id = seat\_id

    def get\_price(self):

        '''Get the price of a certain seat'''

        # connect to the database

        connection = sqlite3.connect(self.database)

        # create a cursor

        cursor = connection.cursor()

        # create our sql statement

        query = 'SELECT price FROM seat WHERE seat\_id = ?' # Questionmark is a place holder for something that will be gotten from the user

        # execute our query

        cursor.execute(query, [self.seat\_id])

        # get the price from the database and store it from the variable

        price = cursor.fetchall()[0][0]

        return price

    def is\_free(self):

        '''Checks in the databse if the seat is taken or not'''

        # connect to the database

        connection = sqlite3.connect(self.database)

        # create a cursor

        cursor = connection.cursor()

        # create our sql statement

        query = 'SELECT taken FROM seat WHERE seat\_id = ?' # Questionmark is a place holder for something that will be gotten from the user

        # execute our query

        cursor.execute(query, [self.seat\_id])

        # fetch the results

        result = cursor.fetchall()[0][0]

        # check if the seat is available or not

        if result == 0: # =0-> means is true

            return True

        else:

            return False

    def occupy(self):

        '''Changes the value of taken in the database from 0 to 1 if seat is free'''

        if self.is\_free():

            # connect to the database

            connection = sqlite3.connect(self.database)

            # create a cursor

            cursor = connection.cursor()

            # create our sql statement

            query = 'UPDATE seat SET taken = ? WHERE seat\_id = ?'

            # execute the querry

            cursor.execute(query, [1, self.seat\_id])

            # Commit changes to the database

            connection.commit()

            # Close the database connection

            connection.close()

class Card:

    '''Represents a bank card needed to finalise a seat purchase'''

    database = 'cinema.db'

    def \_\_init\_\_(self, type, number, cvc, holder):

            self.holder = holder

            self.cvc = cvc

            self.number = number

            self.type = type

    def validate(self, price):

            '''Checks if card is valid and has balance, subtracts price from balance'''

             # connect to the database

            connection = sqlite3.connect(self.database)

            # create a curson

            cursor = connection.cursor()

            # create our sql statement

            query = 'SELECT balance FROM card WHERE number = ? AND cvc = ?' # Questionmark is a place holder for something that will be gotten from the user

            # execute our query

            cursor.execute(query, [self.number, self.cvc])

            # fetch the results

            result = cursor.fetchall()

            if result:

                balance = result[0][0]

                if balance >= price:

                    query = 'UPDATE card SET balance = ? WHERE number = ? AND cvc = ?'

                    cursor.execute(query, [balance - price, self.number, self.cvc])

                    connection.commit()

                    connection.close()

                    return True

class Ticket:

    '''Represents a cinema ticket purchased by a user'''

    def  \_\_init\_\_(self, user, price, seat\_number):

        self.user = user

        self.price = price

        self.id = ''.join([random.choice(string.ascii\_letters) for i in range(8)])

        self.seat\_number = seat\_number

    def to\_pdf(self):

        '''Creaates a PDF ticket'''

        pdf = FPDF(orientation = 'P', unit = 'pt', format = 'A4')

        pdf.add\_page()

        # pdf.set\_font('Times', 'B', size = 24)

        pdf.set\_font(family="Times", style ="B", size = 24)

        pdf.cell(w = 0, h = 80, txt = 'Your digital ticket', border = 1, ln = 1)

        pdf.set\_font(family="Times", style="B", size=14)

        pdf.cell(w=100, h=25, txt="Names", border=1)

        pdf.set\_font(family="Times", style="B", size=12)

        pdf.cell(w=0, h=25, txt=self.user.name, border=1, ln=1)

        pdf.cell(w=0, h=5, txt="", border=0, ln=1)

        pdf.set\_font(family="Times", style="B", size=14)

        pdf.cell(w=100, h=25, txt="Ticket ID", border=1)

        pdf.set\_font(family="Times", style="B", size=12)

        pdf.cell(w=0, h=25, txt=self.id, border=1, ln=1)

        pdf.cell(w=0, h=5, txt="", border=0, ln=1)

        pdf.set\_font(family="Times", style="B", size=14)

        pdf.cell(w=100, h=25, txt="Price", border=1)

        pdf.set\_font(family="Times", style="B", size=12)

        pdf.cell(w=0, h=25, txt=f"{self.price}", border=1, ln=1)

        pdf.cell(w=0, h=5, txt="", border=0, ln=1)

        pdf.set\_font(family="Times", style="B", size=14)

        pdf.cell(w=100, h=25, txt="Seat Number", border=1)

        pdf.set\_font(family="Times", style="B", size=12)

        pdf.cell(w=0, h=25, txt=self.seat\_number, border=1, ln=1)

        pdf.cell(w=0, h=5, txt="", border=0, ln=1)

        pdf.output(f"{self.user.name}-{self.seat\_number}.pdf")

name = input('Enter your full names: ')

seat\_id = input('Preferred seat number: ')

card\_type = input('Your card type for payment: ')

card\_number = int(input('Your card number: '))

card\_cvc = int(input('Your card cvc: '))

card\_holder = input('Card holder name: ')

user = User(name)

seat = Seat(seat\_id)

card = Card(card\_type, card\_number, card\_cvc, card\_holder)

print(user.buy(seat, card))