SHOP DATABASE APPLICATION DOCUMENTATION

Project name: Shop Database Application

Author: Maksym Kuzma

Class: C4a

Email: kuzmamax78@gmail.com

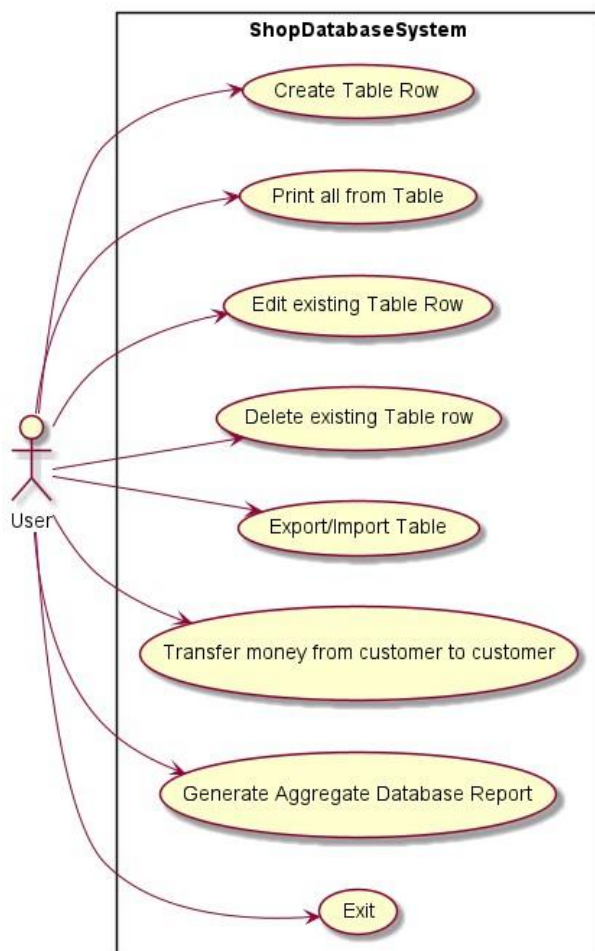Phone number: 771188642

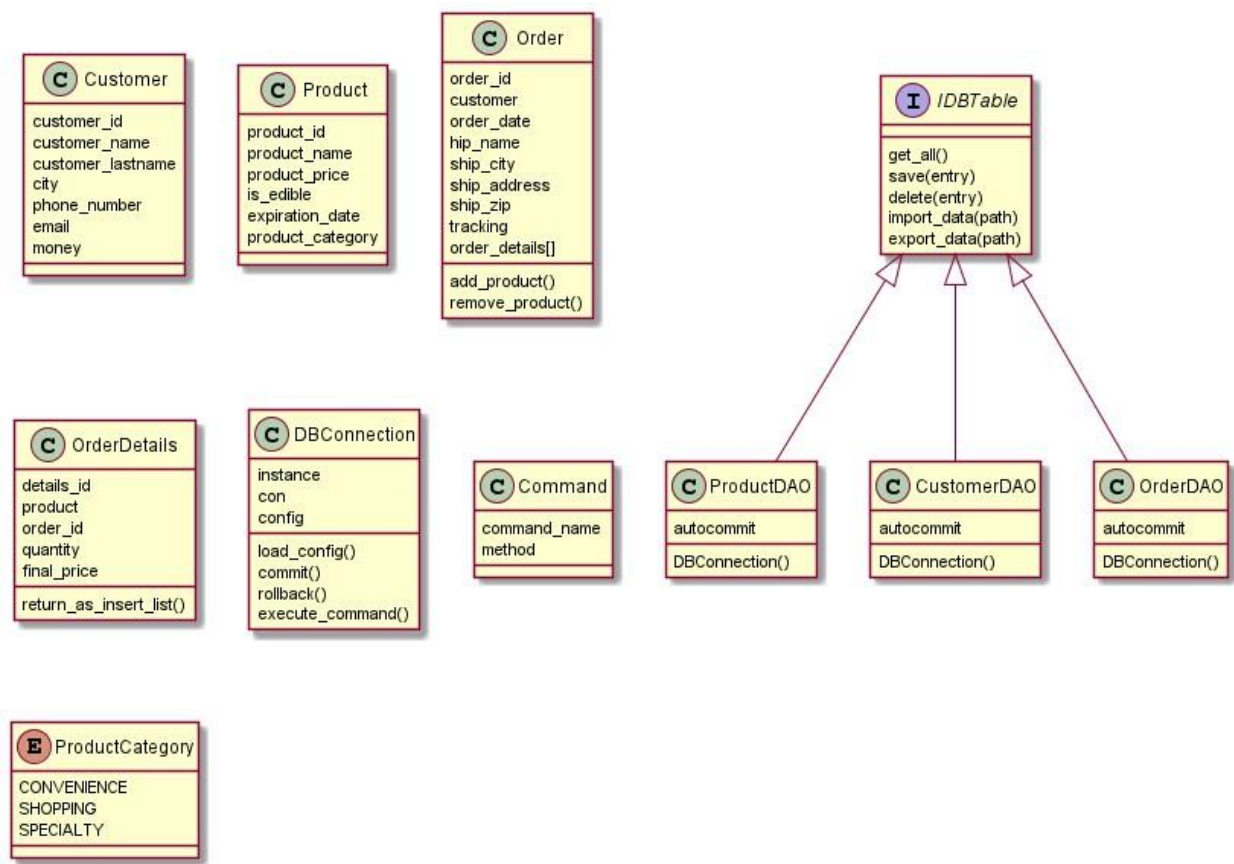School: SPŠE Ječná

Project is being used as school project

Application Requirements

- Python 3.9 with external libraries
    - pandas
    - pyodbc
    - numpy
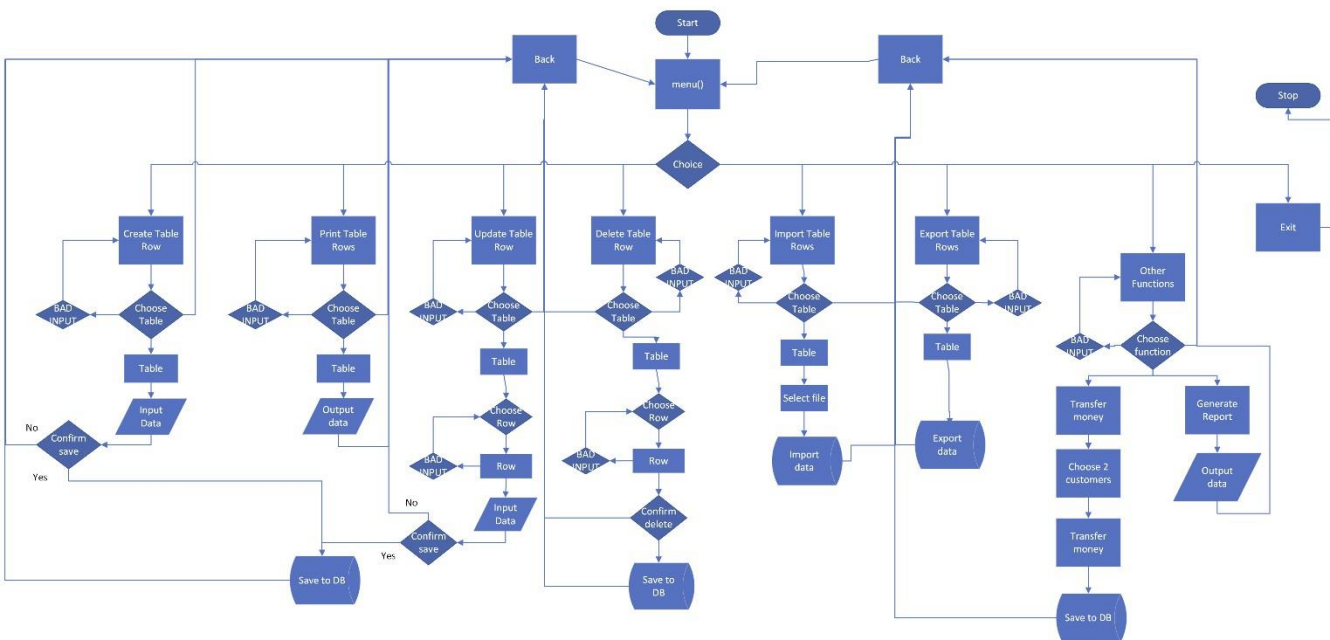- Microsoft SQL Management Studio 2019
- MS SQL Server
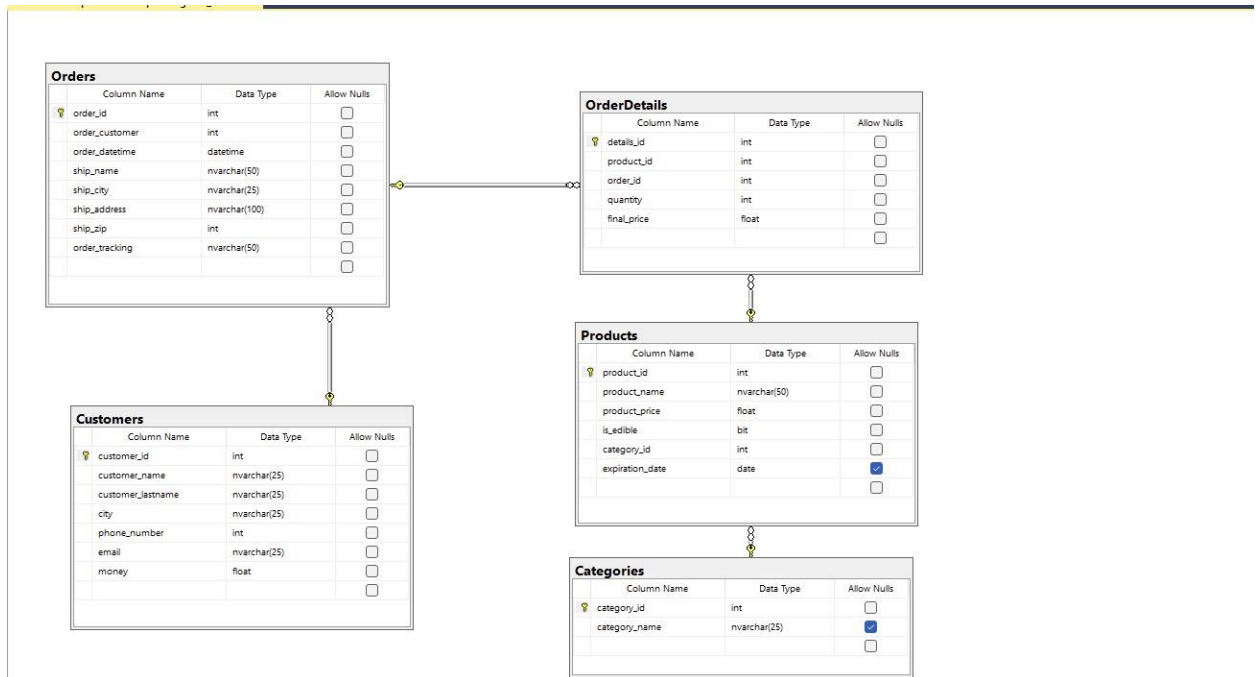
Application UML Use Case Diagram

Application UML Class Diagram

C Customer
customer_id
customer_name
customer_lastname
city
phone_number
email
money

C Product
product_id
product_name
product_price
is_edible
expiration_date
product_category

C Order
order_id
customer
order_date
hip_name
ship_city
ship_address
ship_zip
tracking
order_details[]

add_product()
remove_product()

I IDBTable
get_all()
save(entry)
delete(entry)
import_data(path)
export_data(path)

C OrderDetails
details_id
product
order_id
quantity
final_price

return_as_insert_list()

C DBConnection
instance
con
config

load_config()
commit()
rollback()
execute_command()

C Command
command_name
method

C ProductDAO
autocommit

DBConnection()

C CustomerDAO
autocommit

DBConnection()

C OrderDAO
autocommit

DBConnection()

E ProductCategory
CONVENIENCE
SHOPPING
SPECIALTY

Application UML Activity Diagram

Start
menu()
Choice
Back
Back
Stop
Exit

Create Table Row — Choose Table — BAD INPUT — Table — Input Data — Confirm save — No / Yes — Save to DB

Print Table Rows — Choose Table — BAD INPUT — Table — Output data

Update Table Row — Choose Table — BAD INPUT — Table — Choose Row — BAD INPUT — Row — Input Data — Confirm save — No / Yes

Delete Table Row — Choose Table — BAD INPUT — Table — Choose Row — BAD INPUT — Row — Confirm delete — Save to DB

Import Table Rows — Choose Table — BAD INPUT — Table — Select file — Import data

Export Table Rows — Choose Table — BAD INPUT — Table — Export data

Other Functions — Choose function — BAD INPUT — Transfer money — Choose 2 customers — Transfer money — Save to DB

Generate Report — Output data

Application Database ER Diagram

Application Export/Import Data Scheme

Export data from each is saved in separate .csv file and has this format:

- **products.csv**

Header: product_name,product_price,is_edible,category_id,expiration_date

Row example: Bread,25.0,True,1,2022-05-05

- **customers.csv**

Header: customer_name,customer_lastname,city,phone_number,email,money

Row example: Max,Kuzma,Prague,772731155,maxkuzma@gmail.com,3000.0

- **orders.csv**

Header: order_customer,order_datetime,ship_name,ship_city,ship_address,ship_zip,order_tracking

Row example: 1,2022-01-06 01:25:24.940,Maksym Kuzma,Prague,Address 12,14000,CZ13254223

- **order_details.csv**

Header: product_id,order_id,quantity,final_price

Row example: 2,1,3,3600.0

Files in mentioned format can also be used as import files.

Application configuration

All configuration and application installation is described in README.txt file in main folder directory.

Application Errors and their possible solutions

If any error occurs, it's saved into /log/error.log file

Known errors:

FileNotFoundError with value "Config not found"

Solution:

Step 1: create folder cfg in program directory if not exists

Step 2: create config.json file in this folder

Step 3: write configuration like this into this file

```
{
 "SERVER": "REPLACE WITH YOUR SERVER NAME",
 "DATABASE": "Shop",
 "UID": "REPLACE WITH YOUR USER THAT WAS SET IN SQL DATABASE IMPORT*",
 "PWD": " REPLACE WITH YOUR PASSWD THAT WAS SET IN SQL DATABASE IMPORT* "
}
```

*More information in README.txt

Step 4: The problem must be solved.

InterfaceError

Solution: This error appears when config file is not

correctly configured. Check config.

If any other error occurs contact with me and I'll help you fix it.

Third-party libraries

In order to launch this application on Linux or MacOS you have to have installed Python 3.9 or higher on your machine.

Also, you will need to download third-party libraries using **pip install** in CLI. Use these commands to install them:

- pip install pyodbc
- pip install pandas
- pip install numpy

Project resumé

Project has CRUD for Many to Many table.

Project has multi table transaction for transfering money between customers.

Project can generate aggregative report.

Project can import/export data from/to .CSV file.

Project has external config file that can be configured.

Project saves all errors/exceptions to a log file.