UNIVERSITÉ
Concordia
UNIVERSITY

Software Engineering

SOEN – 6441

Advanced Programming Practices

Instructor: Dr. C. Constantinides

Project By:

Ruchit Dobariya - 40232238

Rutvik Jakasaniya - 40220251

Coding Standards – Python

1. Naming Conventions as camelCase

   Class name in CamelCase starts with capital letter.

2. Code Layout

   Consistent layout throughout the code

3. Exception Handling for Critical Situation

   There are many try and except cases where some of the method may give runtime error where exception handling is necessary.

4. Object Oriented Code

   The design pattern is based on object-oriented methods where classes are created.

5. String Formatting

   The string's placeholder is filled with the formatted value(s) inserted by the format() method.

6. Import Modules in Order

   The order of importation is be standard libraries first, followed by third party libraries, and then local libraries.

7. Indentation with Tabs

   All the indentation in the code is properly given with tab only.

Software Architecture Document

Architectural Pattern: Table Data Gateway

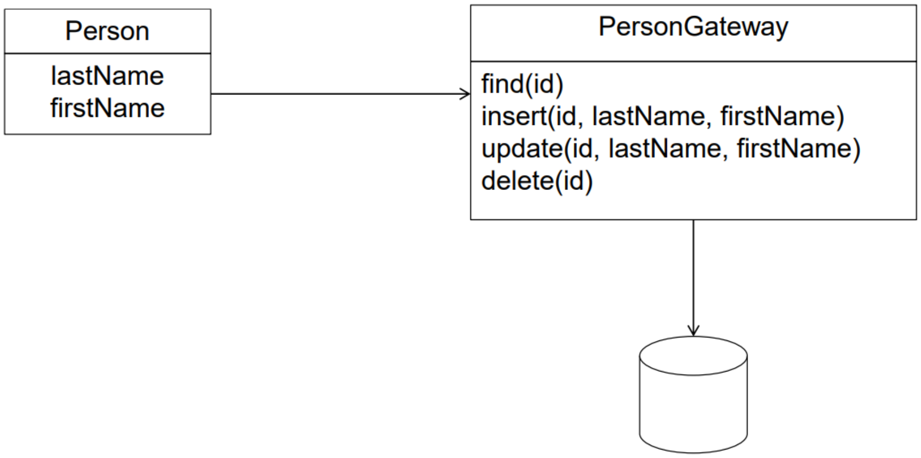
Table Data Gateway is a design pattern which belongs to Data Source Architectural Pattern contains an object which functions as a gateway to all the database tables.

A Table Data Gateway holds interface for accessing a table or view by applicable SQL queries: selects, inserts, updates, and deletes. The objective is to decouple the responsibility for retrieving things from a database from the actual usage of those objects.

For all database interactions, the main object sends messages to the TDG.Table Data Gateway is stateless, as it only pushes data back and forth.

In the project, there are total of four tables which operates as per the request made by the user. The main file imports TDG file as well as connection file, where connection is use to establish and maintain connection with the local server, where as in TDG the database is created and all the Insert, Delete, Update, find operations are implemented. Later, as per the user requirement, the methods are called using singleton object, and results are out shown in the database.

The pattern behaves as per the diagram shown below:

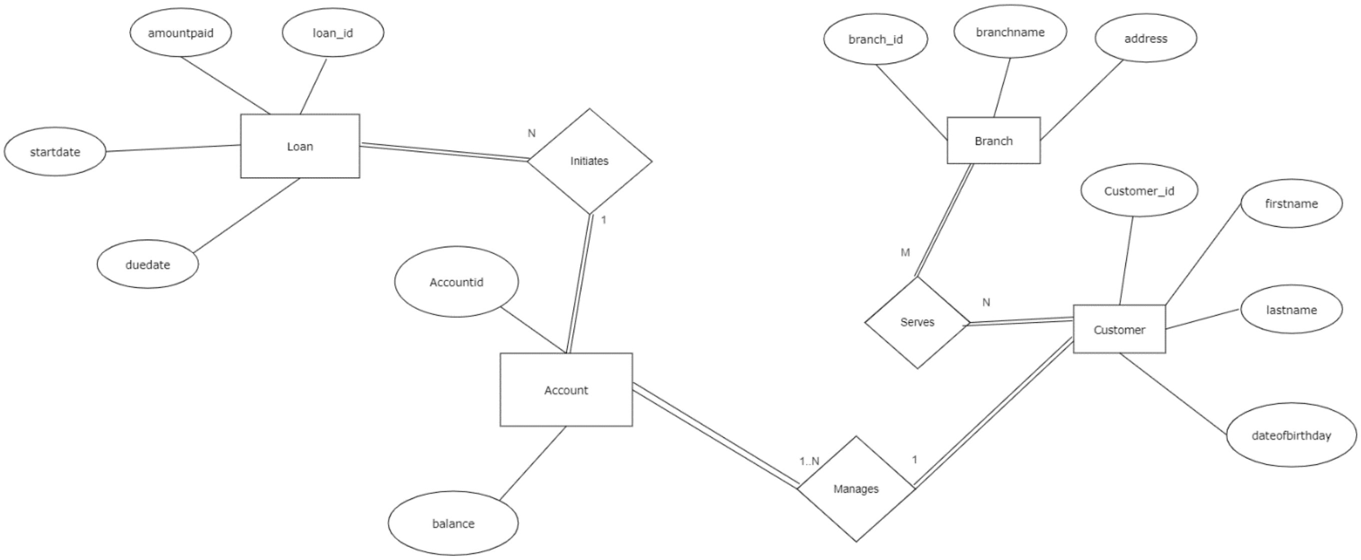

Person
lastName
firstName
PersonGateway
find(id)
insert(id, lastName, firstName)
update(id, lastName, firstName)
delete(id)

Refactoring Strategies

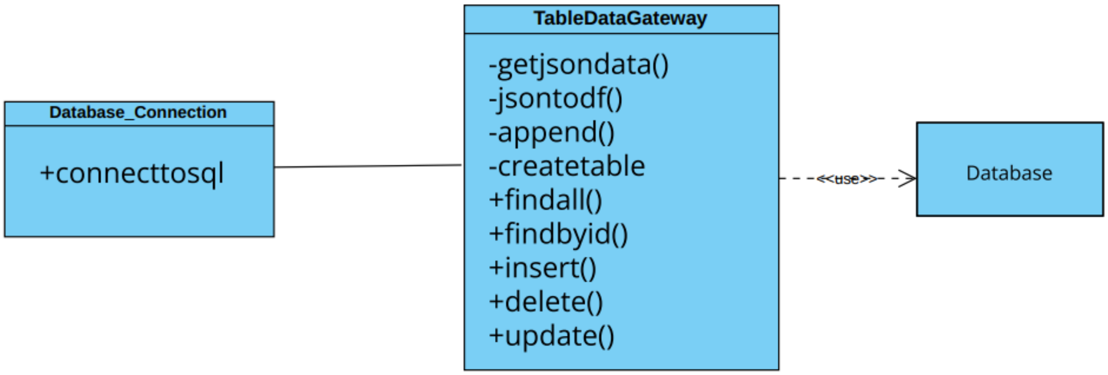
Refactoring code enhances the code we have written and also helps us to understand it better.

1. Defining functions for repetitive tasks.
2. Reduced the size of the code
3. Clean the code to enhance readability
4. DRY Don’t Repeat Yourself
5. Created reusable code
6. Pulled Unnecessary statements out of for/while loops

ER Diagram



UML Diagram



Testing Tool

Python Testing using Mock Database (Unit Testing)

Libraries Used: mock

Sources:- https://pypi.org/project/mock/,
https://docs.python.org/dev/library/unittest.mock.html

mock is a library for testing in Python. It allows you to replace parts of your system under test with mock objects and make assertions about how they have been used.

Unit testing has been used in our program. Unit testing determines whether separate pieces of code can successfully run in order to deliver the desired outcome, without interfering with one another.

To assure desired functioning, testing must be done continuously as the software is built or upgraded. The data in the database may be responsive to methods or functions that can change them Furthermore, you should definitely avoid changing production data during your test if it involves writing to a database table. It's preferable to utilize a different database while testing these features. Therefore, a mock database is necessary.

Every time a test is run, a temporary database is created, and it is deleted after the tests are finished.

MockDB class inherits four methods. These methods will enable us to "setup" and "teardown" a temporary database. Within a single class or test case, the SetUp() function is called prior to each test. In other words, SetUpClass() will execute prior to each test in the relevant test case, and TearDownClass() will execute following the completion of each test. The details of the test database are stored in testconfig. The config variable in the utils file is patched with testconfig using mock db config.

The test database is deleted by the tearDownClass method.After that we have inherit this classes to create test cases. To test the utils.py functions, create tests. The test utils code will run a few sample tests.

GITHUB LINK:- https://github.com/rutvikjakasaniya/APPproject

References:

1. MySQL Connector:- https://dev.mysql.com/doc/connector-python/en/connector-python introduction.html
2. Mock :- https://pypi.org/project/mock/
3. https://www.uml-diagrams.org/class-reference.html
4. https://www.fullstackpython.com/object-relational-mappers-orms.html