Masab Anees
Masab.anees019@gmail.com

# Api Documentation

This is an API written in Python Flask with SQLAlchemy Database under the strong bridge of Design Patterns

## Following Design Patterns use in this project

- Singleton
  - This is used to create a single instance of database, Flask App.

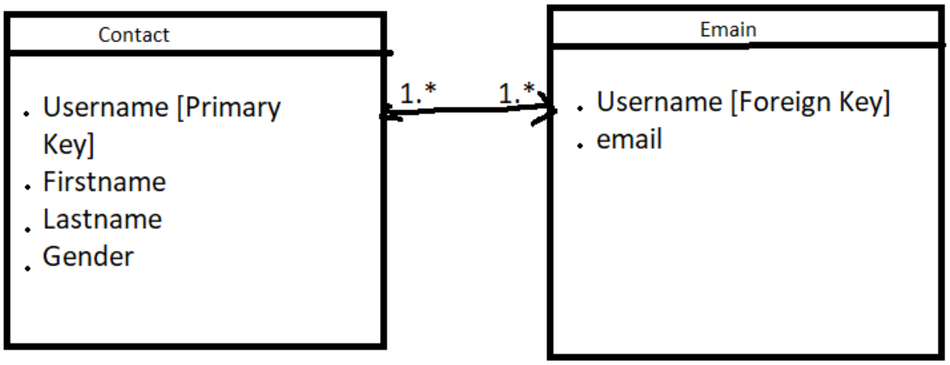
## Coding Files System

- There are total of 3 .py files are used in order to reduce the complexity of switching and increase the understanding of how things are working.
- Index.py contains the Factory and DB Module
- Function.py contains the functions used on this project
- Test.py created to run the endpoints of Apis (Use For testing Purpose)
- Initial Testing is performed for you to check, all sample testing is present in text.py

## Project Explanation

- For the Model extension, a new table called **Email** is created with the foreign key of user name which link the Primary key of the **Contact** Table, hence a Primary key & Foreign key relation was created in order to updating the existing table, which will be time taking.
- **Data Integrity & Data Redundancy** is maintained. When a username of Contact Table is updated or deleted, the following changes occur in the Email table .

## Relation between 2 tables

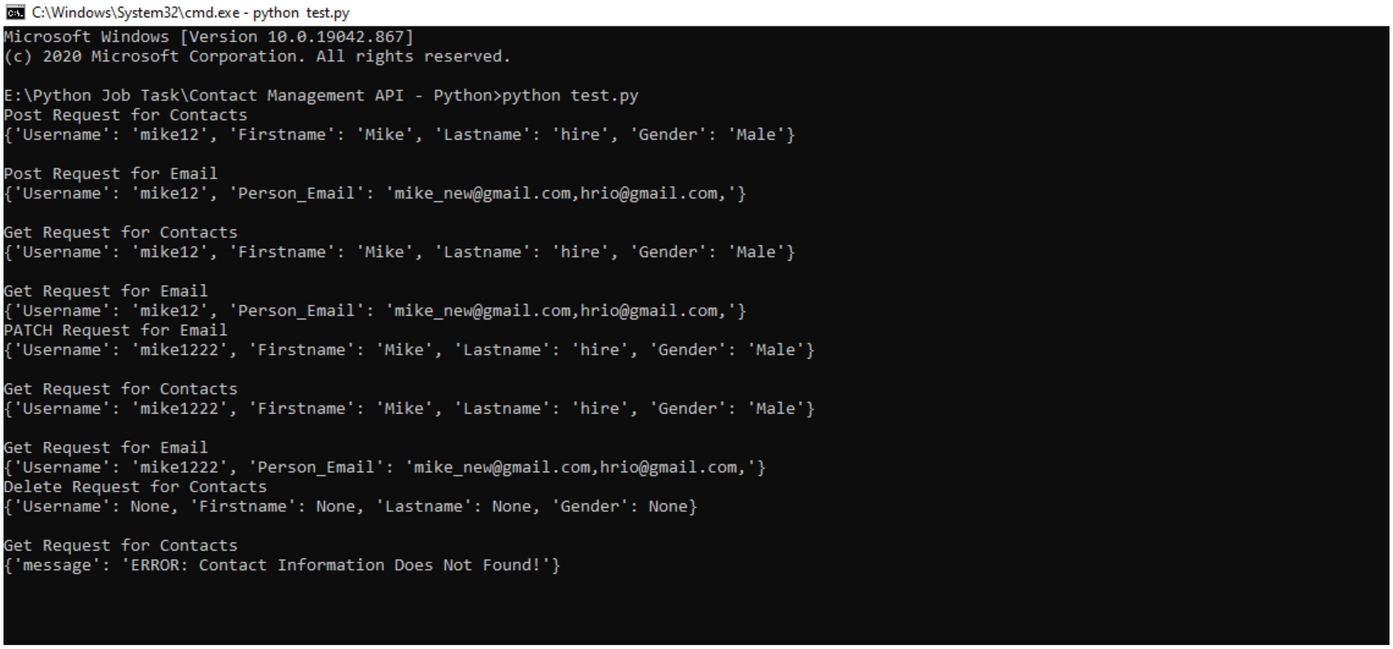


Masab Anees
Masab.anees019@gmail.com

## Relation between 2 tables

- In order to run this project, kindly install all recommended libraries given in **requirement.txt**

### Screenshots of Output from Api.



```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

E:\Python Job Task\Contact Management API - Python>python test.py
Post Request for Contacts
{'Username': 'mike12', 'Firstname': 'Mike', 'Lastname': 'hire', 'Gender': 'Male'}

Post Request for Email
{'Username': 'mike12', 'Person_Email': 'mike_new@gmail.com,hrio@gmail.com,'}

Get Request for Contacts
{'Username': 'mike12', 'Firstname': 'Mike', 'Lastname': 'hire', 'Gender': 'Male'}

Get Request for Email
{'Username': 'mike12', 'Person_Email': 'mike_new@gmail.com,hrio@gmail.com,'}
PATCH Request for Email
{'Username': 'mike1222', 'Firstname': 'Mike', 'Lastname': 'hire', 'Gender': 'Male'}

Get Request for Contacts
{'Username': 'mike1222', 'Firstname': 'Mike', 'Lastname': 'hire', 'Gender': 'Male'}

Get Request for Email
{'Username': 'mike1222', 'Person_Email': 'mike_new@gmail.com,hrio@gmail.com,'}
Delete Request for Contacts
{'Username': None, 'Firstname': None, 'Lastname': None, 'Gender': None}

Get Request for Contacts
{'message': 'ERROR: Contact Information Does Not Found!'}
```

# Example of API Requests

## Samples of Post Request

**END POINT:** = http://127.0.0.1:5000/Person_Contact

**END POINT**: http://127.0.0.1:5000/Person_Email

### EXAMPLE

**# POST OPERATION** = http://127.0.0.1:5000/Person_Contact + {Username , Firstname, Lastname, Gender}

**# POST OPERATION** = http://127.0.0.1:5000/Person_Email + {Username , email} **Samples of Get Request**

Masab Anees
Masab.anees019@gmail.com

## Sample Patch (Update) Request

**END POINT:** = http://127.0.0.1:5000/Person_Contact/Username

**END POINT**: http://127.0.0.1:5000/Person_Email/Username

**EXAMPLE**

# PATCH/UPDATE OPERATION = http://127.0.0.1:5000/Person_Contact/Username + {Username , Firstname, Lastname, Gender}

# PATCH/UPDATE OPERATION = http://127.0.0.1:5000/Person_Email/Username + {Username , Email}

## Sample Delete Request

**END POINT:** = http://127.0.0.1:5000/Person_Contact/Username

**EXAMPLE**

# DELETE OPERATION = http://127.0.0.1:5000/Person_Contact/Username + {Username , Firstname, Lastname, Gender}

## Sample GET Request

**END POINT:** = http://127.0.0.1:5000/Person_Contact/Username

**END POINT:** = http://127.0.0.1:5000/Person_Email/Username

**EXAMPLE**

# GET OPERATION = http://127.0.0.1:5000/Person_Contact/Username

# GET OPERATION = http://127.0.0.1:5000/Person_Email/Username