**User’s Manual**

To install web application in your system. It assumes you have, to begin with, the following dependencies installed:

1. Ubuntu 16

2. Python 2.7

3. Flask

4. Virtualenv (and, optionally, virtualenvwrapper)

5. Install all requirements from /dicom/requirements.txt

6. Install dcmtk on ubuntu using this command: sudo-apt-get install dcmtk

**A. Database Setup**

Flask has support for several relational database management systems, including [SQLite](https://sqlite.org/), [MySQL](https://www.mysql.com/), and [PostgreSQL](https://www.postgresql.org/). For this tutorial, we will be using MySQL. It's popular and therefore has a lot of support, in addition to being scalable, secure, and rich in features.

We will install the following (remember to activate your virtual environment):

1. [Flask-SQLAlchemy](http://flask-sqlalchemy.pocoo.org/2.1/): This will allow us to use [SQLAlchemy](http://www.sqlalchemy.org/), a useful tool for SQL use with Python. SQLAlchemy is an Object Relational Mapper (ORM), which means that it connects the objects of an application to tables in a relational database management system. These objects can be stored in the database and accessed without the need to write raw SQL. This is convenient because it simplifies queries that may have been complex if written in raw SQL. Additionally, it reduces the risk of [SQL injection attacks](https://www.w3schools.com/sql/sql_injection.asp) since we are not dealing with the input of raw SQL.
2. [MySQL-Python](https://pypi.python.org/pypi/MySQL-python/1.2.5): This is a Python interface to MySQL. It will help us connect the MySQL database to the app.

Install command:

pip install flask-sqlalchemy mysql-python

1. We'll then create the MySQL database. Ensure you have MySQL installed and running, and then log in as the root user:

mysql -u root

mysql> CREATE USER 'dt\_admin'@'localhost' IDENTIFIED BY 'dt2018';

Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE dicom\_db;

Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON dicom\_db . \* TO 'dt\_admin'@'localhost';

Query OK, 0 rows affected (0.00 sec)

1. We have now created a new user dt\_admin with the password dt2018, created a new database dicom\_db, and granted the new user all database privileges. Next, let's edit the instance/config.py.

# instance/config.py

SECRET\_KEY = 'p9Bv<3Eid9%$i01'

SQLALCHEMY\_DATABASE\_URI = 'mysql://dt\_admin:dt2018@localhost/dicom\_db'

**B. Migrations**

After you have configured database and add new database on mysql server, you could migrate the database using flask-migrate.

**For installation:**

pip install flask-migrate

**For migrate database:**

flask db init

flask db migrate

flask db upgrade

**To check whether your migrate successful or not, type it in your terminal:**

mysql -u root

mysql> use dicom\_db;

mysql> show tables;

**C. Running Website**

In your terminal (dicom folder), type this code after you set the virtual environment

export FLASK\_CONFIG=development

export FLASK\_APP=run.py

flask run

\* Serving Flask app "run"

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)