Additional service to realize

Like Service

Interaction of LIKE SERVICE of photos among photographers.

Implementation

- **REST**
- We define several classes in our code, shown as below

  LikeDesc

  ```
  class LikeDesc(BaseModel):
      liker_name: str = Field (None, title = Lname.STR, max_length = Lname.MAX_LENGTH)
      display_name: str = Field (None, title = Dname.STR, max_length = Dname.MAX_LENGTH)
      photo_id: int = Field (None, title = Pid.STR)
      date: str = Field (None, title = Date.STR, max_length = Date.MAX_LENGTH)
  ```

  Like

  ```
  class Like(Document, LikeDesc):
      pass
  ```

  LIKE_EXAMPLE

  ```
  LIKE_EXAMPLE = {
      "liker_name": "rdoisneau",
      "date": "25-04-2000"
      }
  ```

  LIKE_BODY

  ```
  LIKE_BODY = Body(..., example = LIKE_EXAMPLE)
  ```

  LikeDigest

  ```
  class LikeDigest(BaseModel):
      liker_name: str
      display_name: str
      photo_id: int
      link: str
  ```

  Likers

  ```
  class Likers(BaseModel):
      items: List[LikeDigest]
  ```

  Count

  ```
  class Count(BaseModel):
      count: int
  ```

- API descriptions (Endpoints with verbs and parameters)

```
# A liker gives a like to a photo
@app.post("/like/{display_name}/{photo_id}", status_code = 201)
async def give_like(response: Response, photo_id: int, display_name: str,
like: LikeDesc = LIKE_BODY)
```

```
# Get all the likers who give like to a photo
@app.get("/like/{display_name}/{photo_id}", response_model = Likers,
status_code = 200)
async def get_photo_all_likers(photo_id: int, display_name: str)
```

```
# Get all the photos liked by a liker
@app.get("/like/{liker_name}", response_model = Likers, status_code = 200)
async def get_liker_all_photos(liker_name : str)
```

```
# Get the detailed like information of a photo
@app.get("/like/{display_name}/{photo_id}/{liker_name}/information",
status_code = 200)
async def get_like(display_name: str, photo_id: int, liker_name: str)
```

```
# Count the total amount of like of a photo
@app.get("/like/{display_name}/{photo_id}/count", response_model = Count,
status_code = 200)
async def count_like(display_name: str, photo_id: int)
```

```
# A liker removes a like to a photo
@app.delete("/like/{display_name}/{photo_id}/{liker_name}", status_code = 200)
async def remove_like(display_name: str, photo_id: int)
```