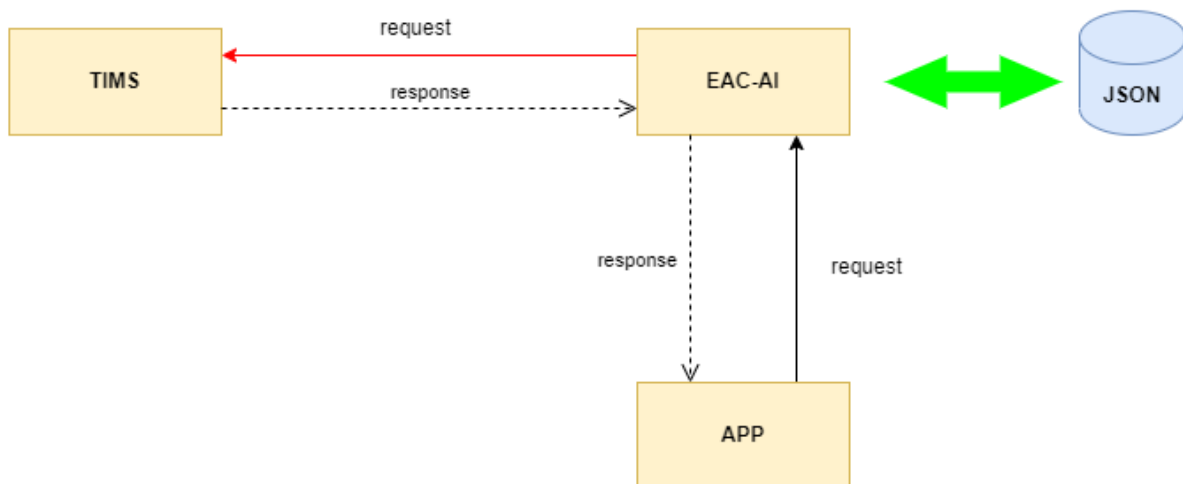# TIMS - EAC

1. POST : checking

- Follow Diagram

- Request from **APP** to **EAC-AI**

```
{
        image: string(base64),
        checking_type: string(checkin/checkout)
}
```

- Request from **EAC-AI** to **TIMS**

```
{
        email: string(email),
        checking_type: string(checkin/checkout)
}
```

- ● Response (App)
  - ☐ Case 200: Success

    ```
    {
            "message": "success",
            "data": {
                    "id": 1,
                    "email": "minhvn@tokyotechlab.com",
                    "name": "Vu Ngoc Minh"
            },
            "version": "1.0.0"
    }
    ```

  - ☐ Case 400: Bad Request

    ```
    {
            "message": "error",
            "errors": [
                    {
                    "key": "image",
                    "errorCode": 400,
                    "message": "can not recognize employee"
                    }
            ],
            "version": "1.0.0"
    }
    ```

  - ☐ Case 401: Unauthorized
  - ☐ Case 404: Not Found

    ```
    {
            "message": "NOT_FOUND",
            "errors": [ ],
            "version": "1.0.0"
    }
    ```
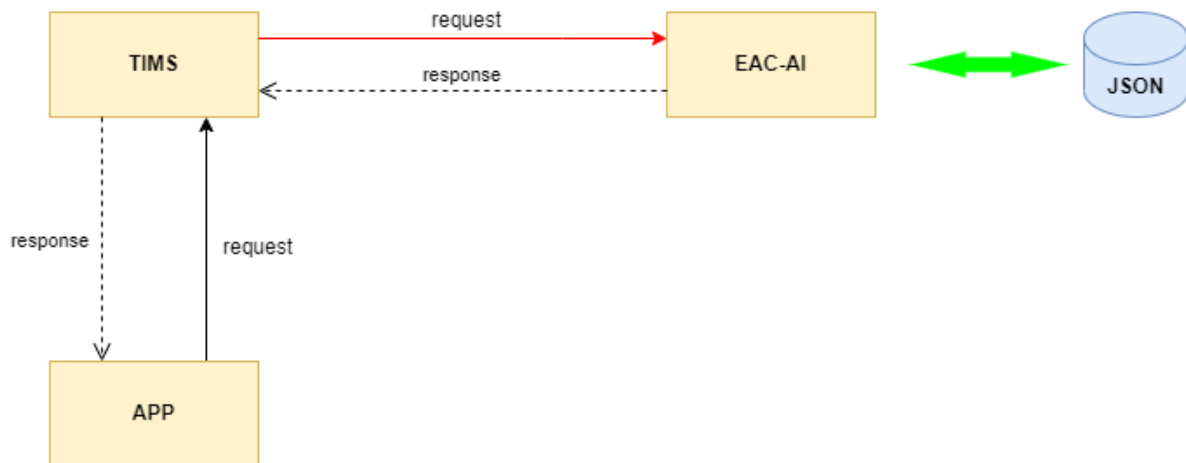
  - ☐ Case 500: Internal Server Error

    ```
    {
            "message": "INTERNAL_SERVER_ERROR",
            "errors": [ ],
            "version": "1.0.0"
    }
    ```

2. POST : user

- Follow Diagram

- Request from **APP** to **TIMS**

```
{
        "name": string(fullname),
        "email": string(email, validate),
        "images": Array<string(base64)>,
        "avatar": string(base64),
        "position": string
}
```

- Request from **TIMS** to **EAC-AI**

```
{
        "email": string(email),
        "images": Array<string(base64)>
}
```

- Response (App)
  - ☐ Case 200: Success

    ```
    {
            "message": "success",
            "data": {
                    "id": 1,
                    "email": "minhvn@tokyotechlab.com",
                    "name": "Vu Ngoc Minh"
            },
            "version": "1.0.0"
    }
    ```

  - ☐ Case 400: Bad Request

    ```
    {
            "message": "error",
            "errors": [
                    {
                    "key": "index",
                    "errorCode": 400,
                    "message": "images are not enough or can not recognize"
                    }
            ],
            "version": "1.0.0"
    }
    ```

  - ☐ Case 404: Not Found
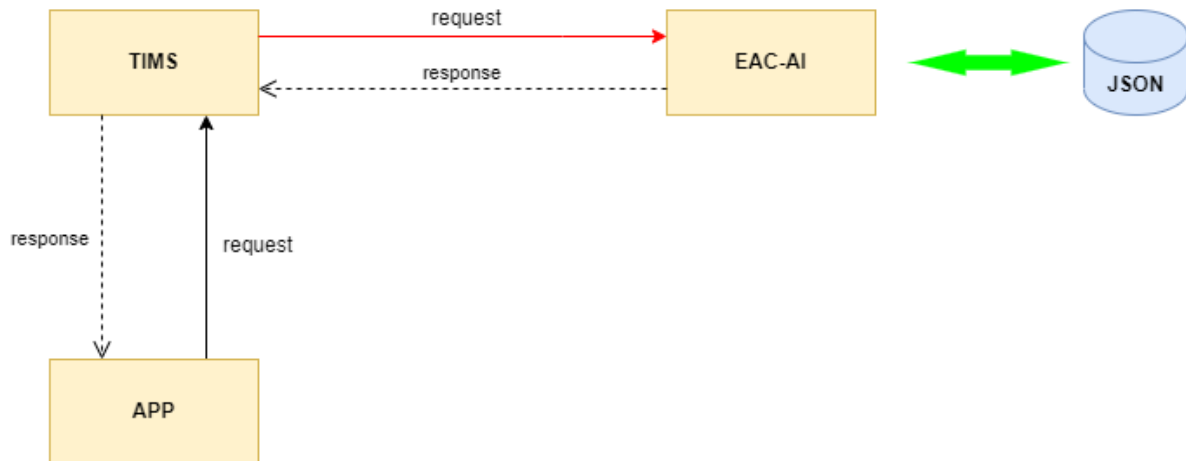
    ```
    {
            "message": "NOT_FOUND",
            "errors": [ ],
            "version": "1.0.0"
    }
    ```

  - ☐ Case 500: Internal Server Error

    ```
    {
            "message": "INTERNAL_SERVER_ERROR",
            "errors": [ ],
            "version": "1.0.0"
    }
    ```

3. DELETE : user

- Follow Diagram

- Request from **APP** to **TIMS**

```
{
        "email": string(email, validate),
}
```

- Request from **TIMS** to **EAC-AI**

```
{
        "email": string(email),
}
```

- Response (App)
  - ☐ Case 200: Success

```
{
        "message": "success",
        "data": {
                "id": 1,
                "email": "minhvn@tokyotechlab.com",
                "name": "Vu Ngoc Minh"
        },
        "version": "1.0.0"
}
```

  - ☐ Case 400: Bad Request

```
{
        "message": "error",
        "errors": [
                {
                "key": "email",
                "errorCode": 400,
                "message": "email is not exist or email is wrong"
                }
        ],
        "version": "1.0.0"
}
```

  - ☐ Case 404: Not Found
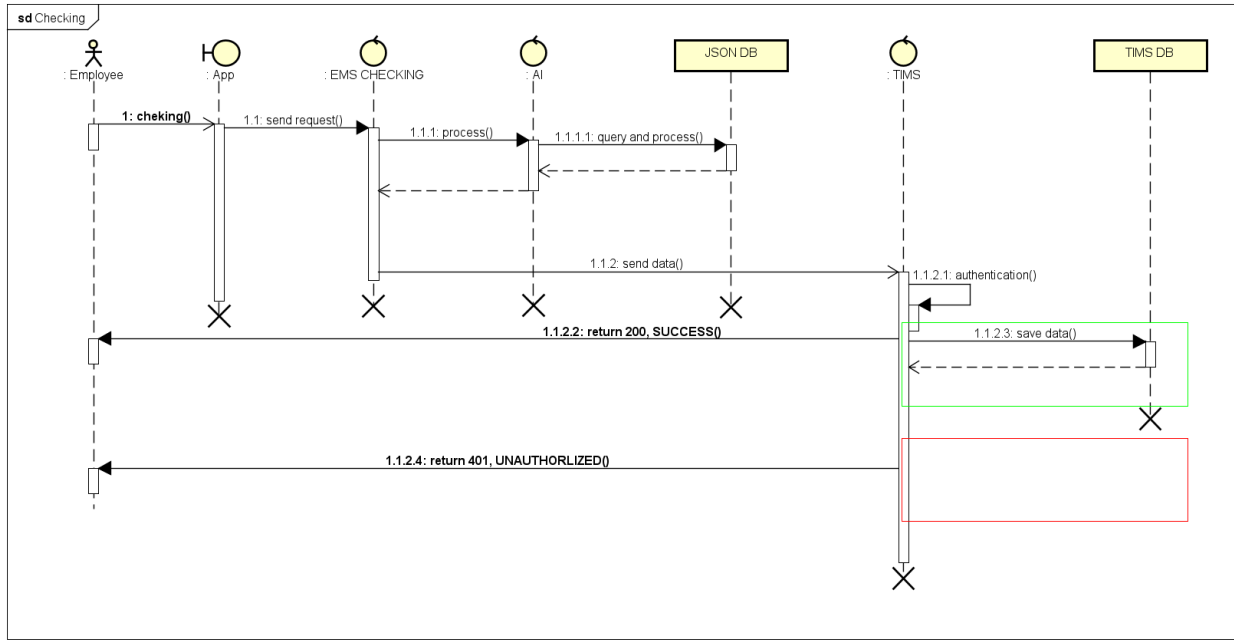
```
{
        "message": "NOT_FOUND",
        "errors": [ ],
        "version": "1.0.0"
}
```

  - ☐ Case 500: Internal Server Error

```
{
        "message": "INTERNAL_SERVER_ERROR",
        "errors": [ ],
        "version": "1.0.0"
}
```

4. Sequence Diagram

- Checking

● Add/ delete user

sd Add user

: Admin   : App   EMS CHECKING   : AI   JSON DB

1: login/ add user()

1.1: send request()

1.1.1: process()

1.1.1.1: authentication()

1.1.1.2: excute()

1.1.1.3: success()

1.1.2: return 200, SUCCESS()

1.1.1.4: unauthentication()

1.1.3: return 401, UNAUTHORLIZED()

sd Delete user

: Admin | : App | : EMS CHECKING | : AI | JSON DB

1: login/ delete user()

1.1: send request()

1.1.1: process()

1.1.1.1: authentication()

1.1.1.2: excute()

1.1.1.3: success()

1.1.2: return 200, SUCCESS()

1.1.1.4: unauthentication()

1.1.3: return 401, UNAUTHORLIZED()