
Cheating Detection On Online Tests

1조 B611137 이강복 | B611169 임정민 | B611210 하인혜

Contents

- 1. Introduction**
- 2. Related Work**
- 3. Proposed System**
 - 3.1 Webpage
 - 3.2 Authentication**
 - 3.3 Eye-Tracking
- 4. Plans**
- 5. Expected Result**

Introduction : Problems

Types of cheating on online test

Substitute examination



Cheating with
blind spot of camera



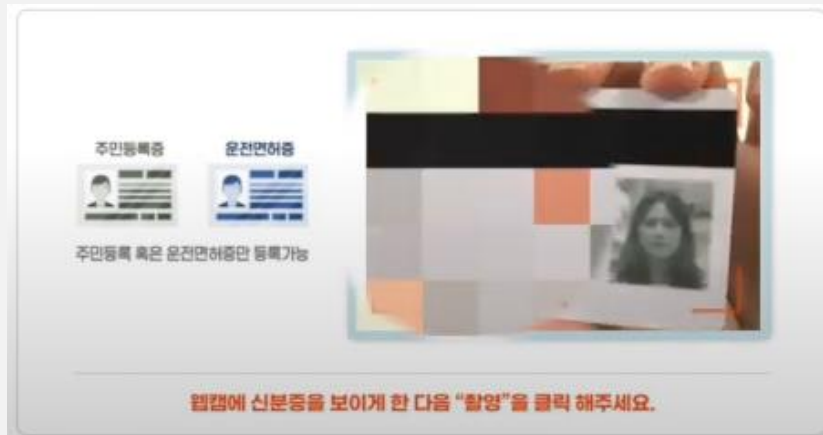
Web surfing



Introduction : Limitation of current program

How to prevent substitute examination

OnTest



- Take a picture of one's ID card
- Check the student's ID card with the information

Monito



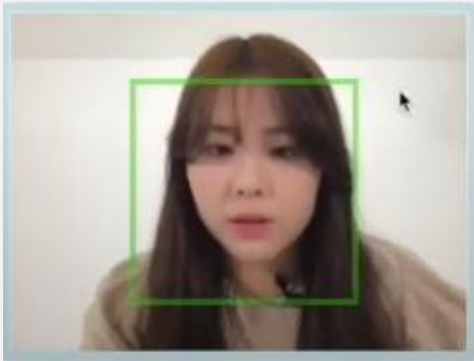
- Take a picture of one's ID card
- Check the birth date and name of the students

- **Supervisor must check the identity card against the person.**
 - Fails to detect face spoofing attacks

Introduction : Limitation of current program

How to prevent cheating using blind spot of camera

OnTest



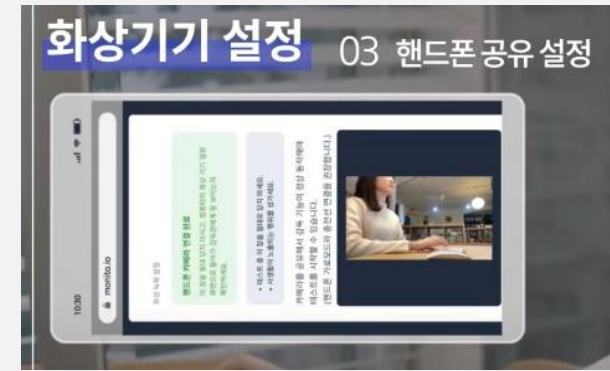
In the square box.



Not in the square box.

- Using a webcam
- A method of warning: when the user's face is out of the bounding box
- **Cheating is possible in the bounding box**

Monito



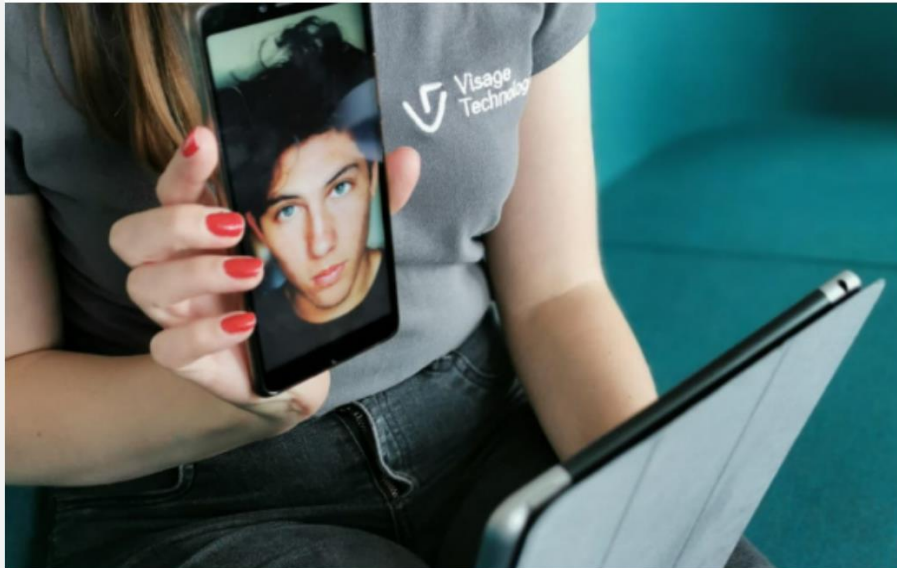
- Use both webcam and cell phone camera to reduce blind spots
- **A blind spot that can't be captured by two cameras**

Proposed system

Differentiation from Related Work

1. Anti-spoofing

: Prevent spoofing attacks by using anti-spoofing technology.



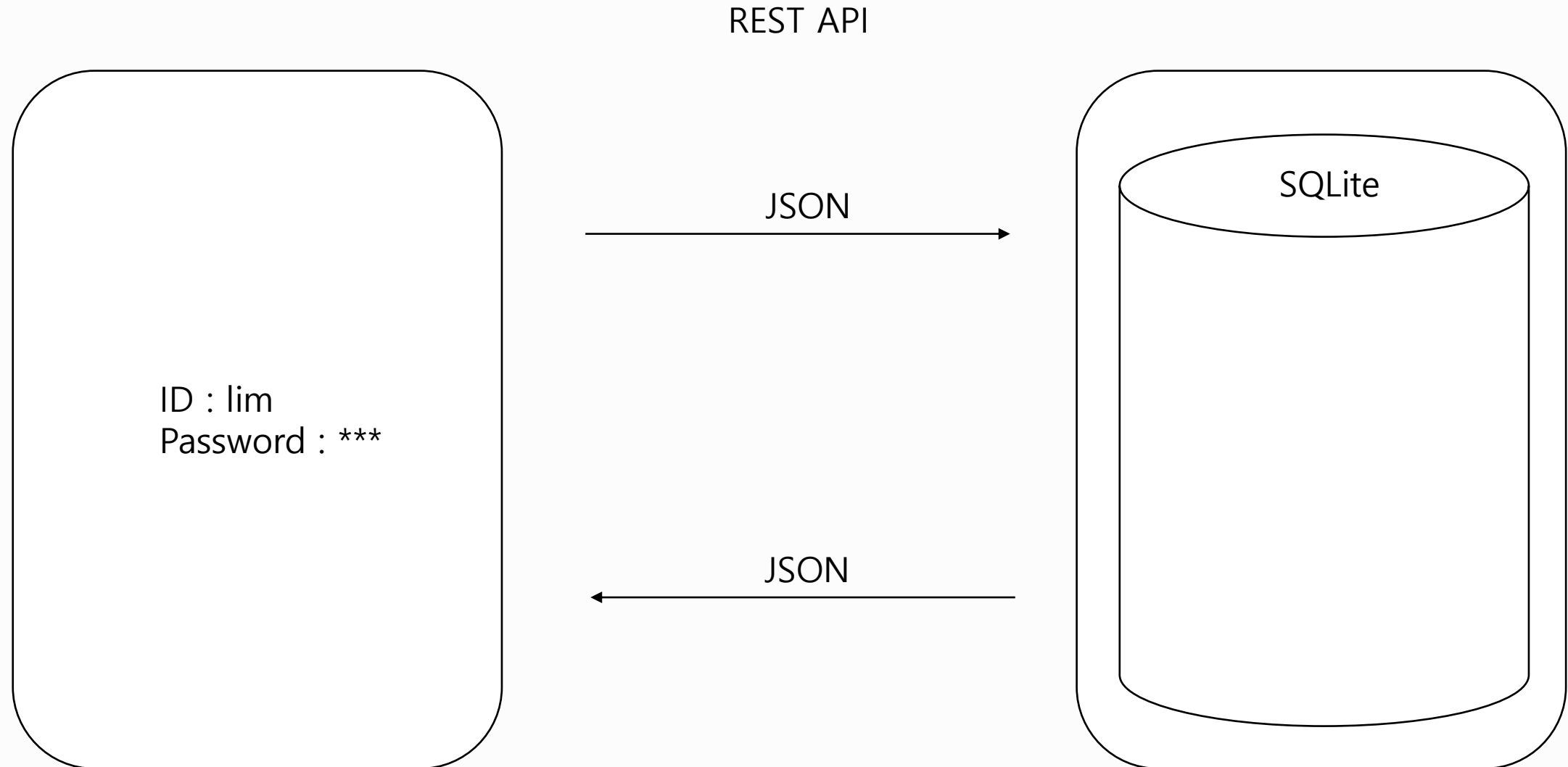
2. Eye tracking

: Use eye tracking to find cheating in camera blind spots.



Proposed System : Webpage

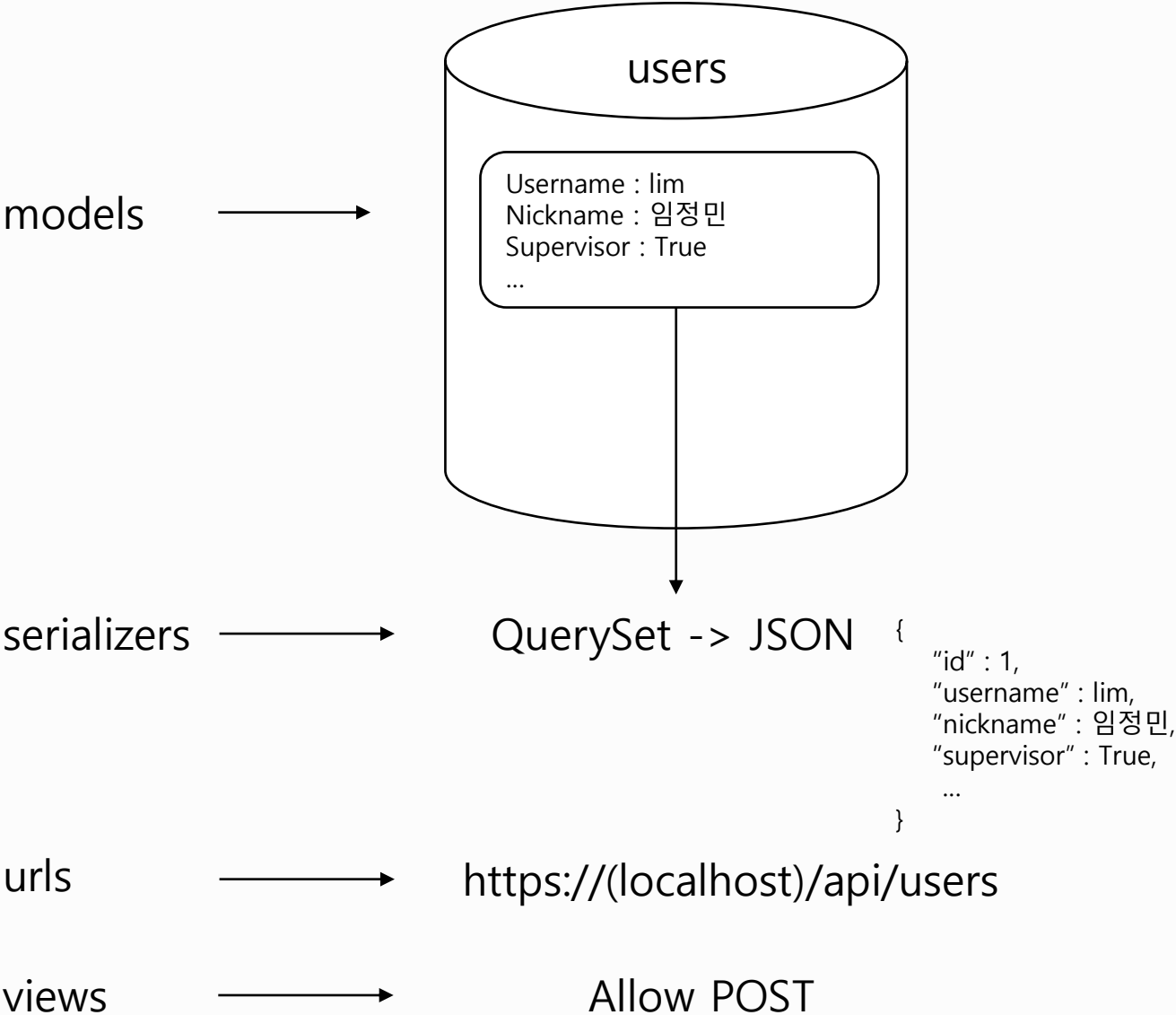
1. Communication



Proposed System : Webpage

2. Sign in Example

Django
└─ Users application
...



Proposed System : Webpage

3. Sign in Example

React

lim

JSON.stringify

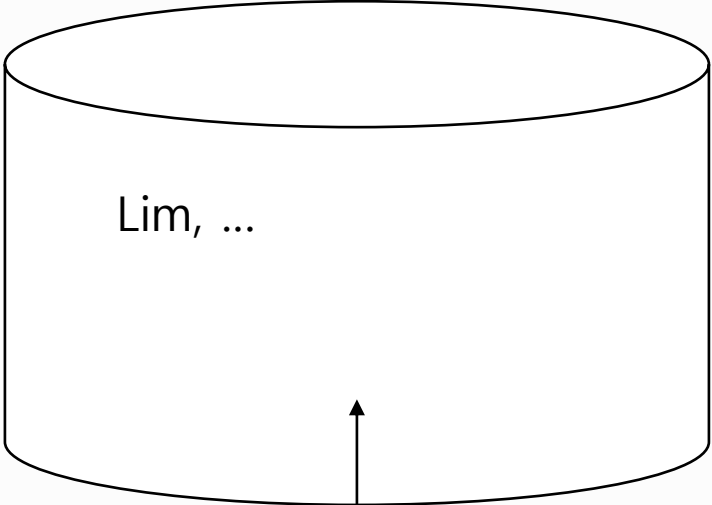


```
{
  method : "POST",
  headers : { "Content-Type" : "application/json",
  body: {
    "username" : "lim",
    ...
  }
}
```

Status Code: 200 OK



Django



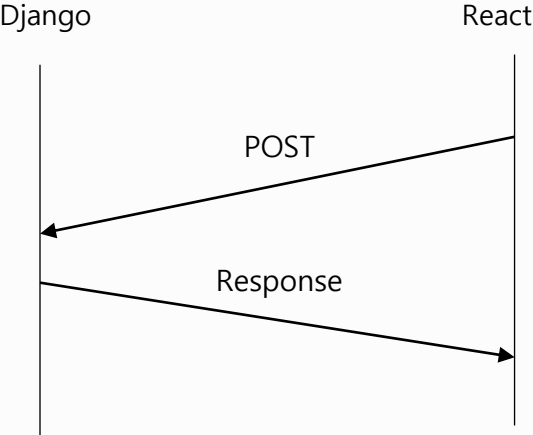
https://(localhost)/api/users



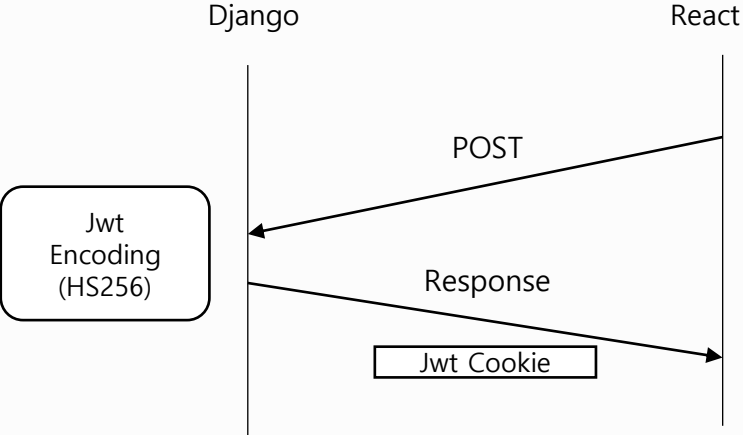
Proposed System : Webpage

4. Sign in(JWT)

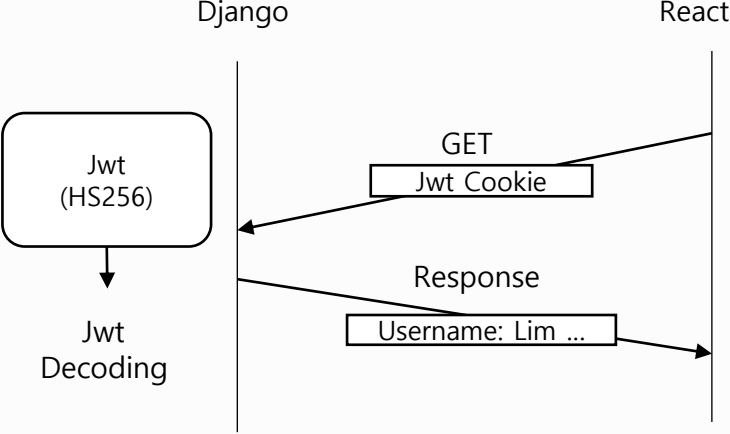
Register



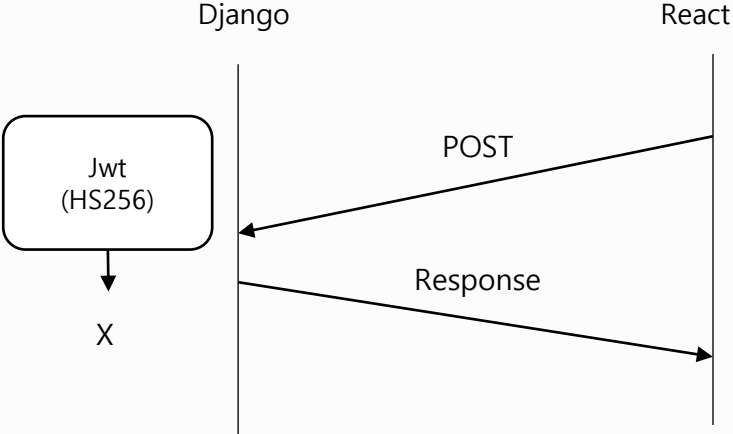
Login



User

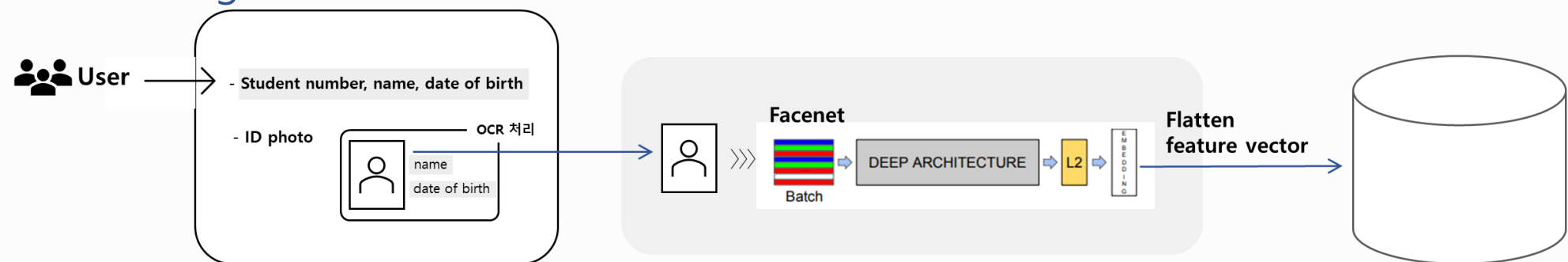


Logout

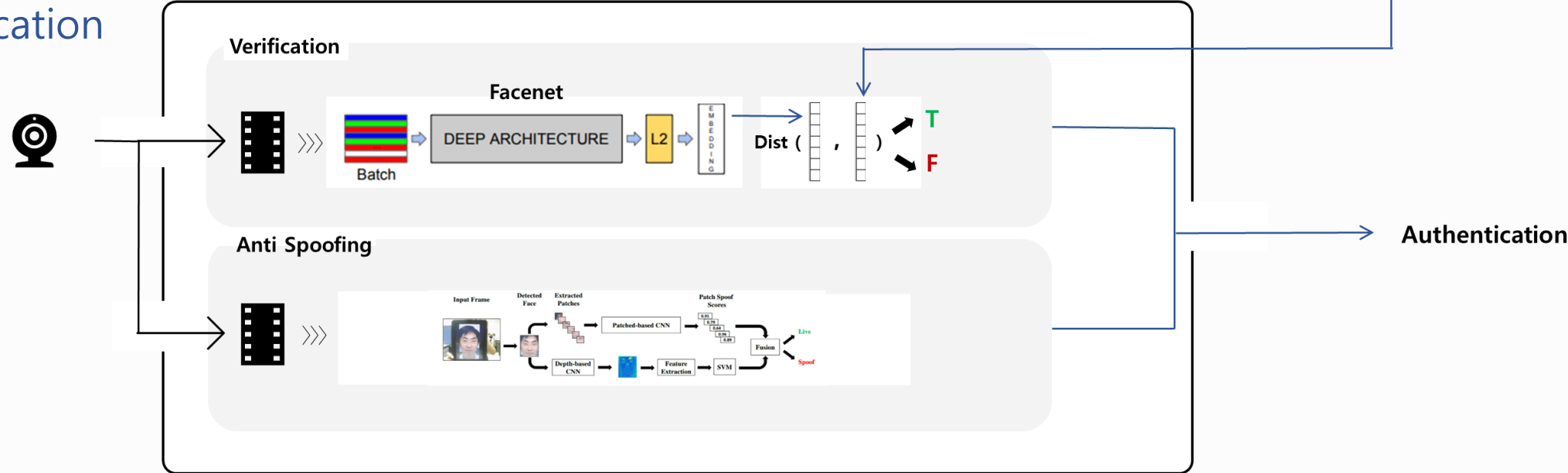


Proposed System : Face Authentication

Pre-information registration



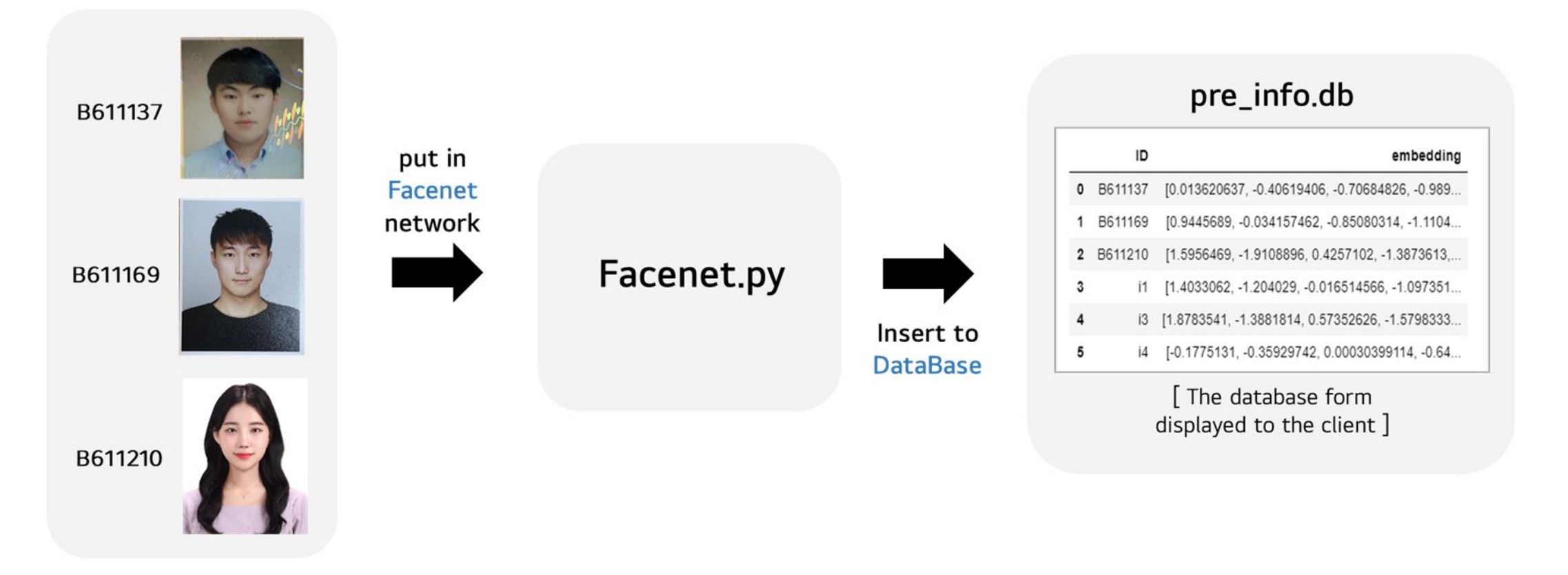
Authentication



Proposed System : Face Authentication

1. Pre-information registration

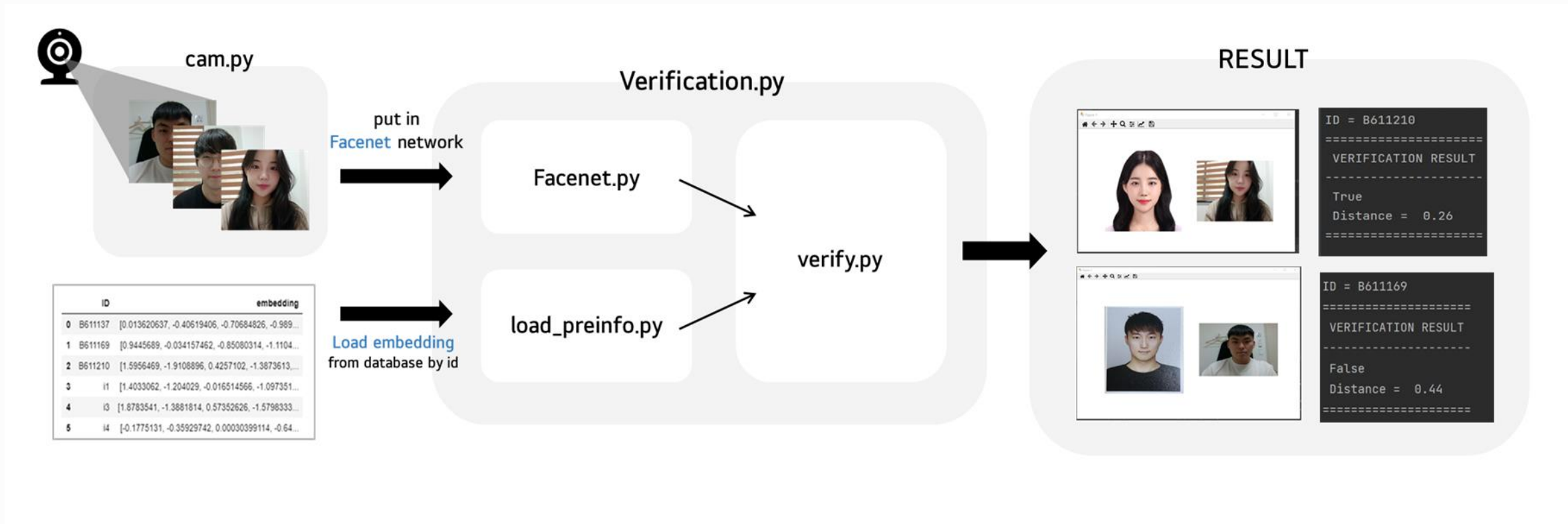
: Get pictures, puts then in the facenet network, and uploads the results to the database.



Proposed System : Face Authentication

2. Face verification

: Receive an image from the webcam and verify it.

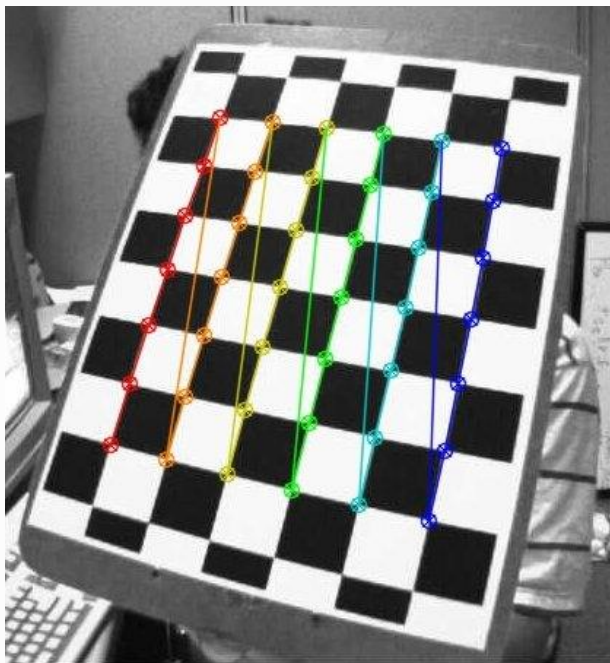


Proposed System : Eye-Tracking



Proposed System : Eye-Tracking

1. Camera Calibration



- Some pinhole cameras introduce significant distortion to images
- Distortion is occurred by a Camera Parameter
- Intrinsic Parameter : Focal Length, Principal Point, Skew coefficient
- Extrinsic Parameter : Rotation, Translation vectors
- In this part, we can get Camera Parameter with `cv2.ChessBoardCorners`

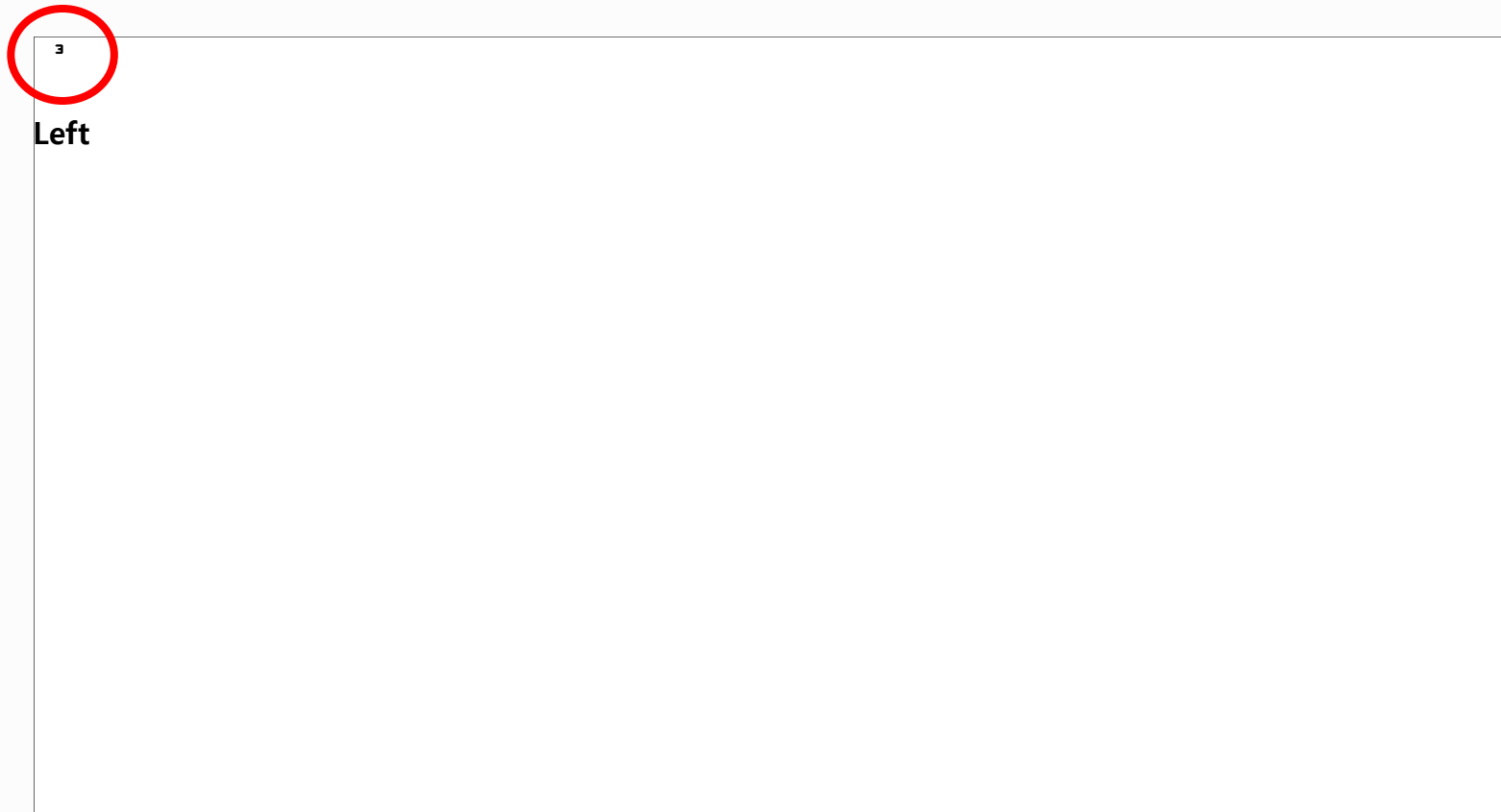
Proposed System : Eye-Tracking

2. Collect User Data

- Collect User's Gaze Data
- A character 'E' appears on the screen.
- User checks the E's position and direction, and presses the corresponding direction with the keyboard.
- In this case, 'E' appears 9 fixed places and 5 random places, and it can be changed.

Proposed System : Eye-Tracking

2. Collect User Data



Proposed System : Eye-Tracking

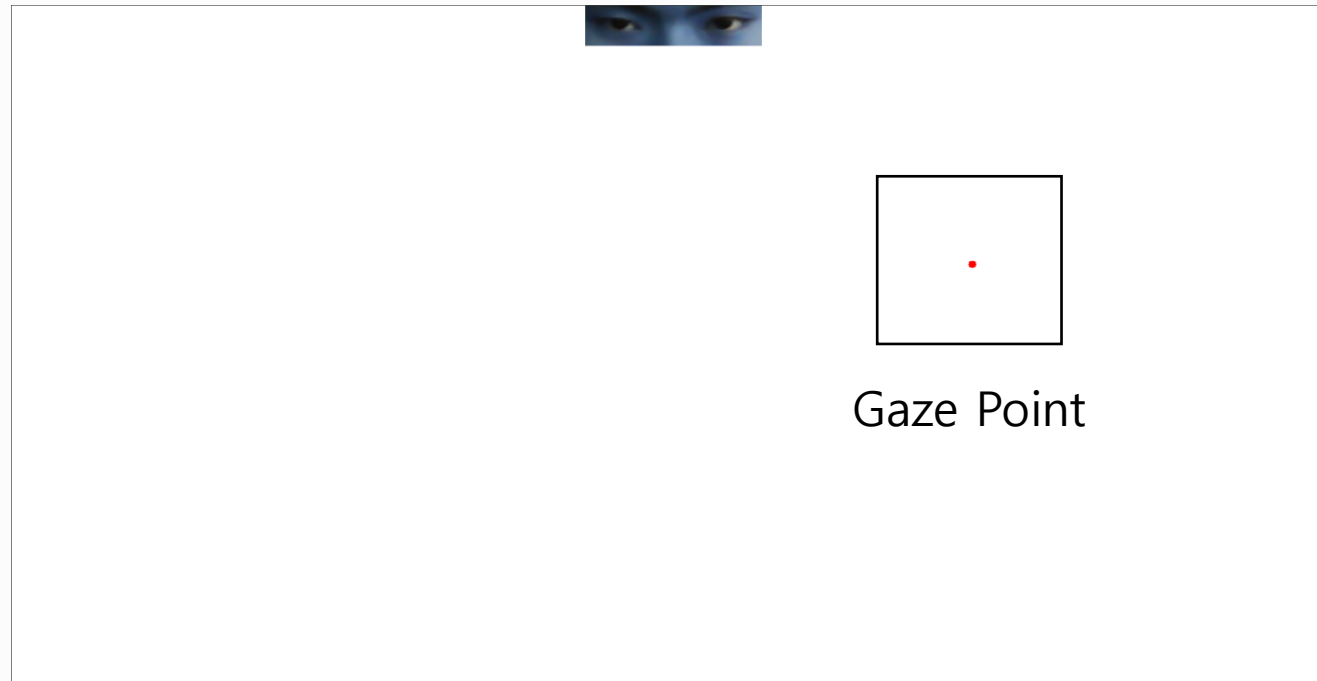
3. Fine-Tuning

```
0000> , Validation: 17.37
0100> Train: 17.49, Validation: 15.89
0200> Train: 16.14, Validation: 14.49
0300> Train: 14.86, Validation: 13.21
0400> Train: 13.68, Validation: 12.02
0500> Train: 12.58, Validation: 10.92
0600> Train: 11.54, Validation: 9.87
0700> Train: 10.56, Validation: 8.88
0800> Train: 9.63, Validation: 8.01
0900> Train: 8.74, Validation: 7.19
1000> Train: 7.92, Validation: 6.42
```

- Train with previously collected data.
- Using pre-trained Gaze MLP model

Proposed System : Eye-Tracking

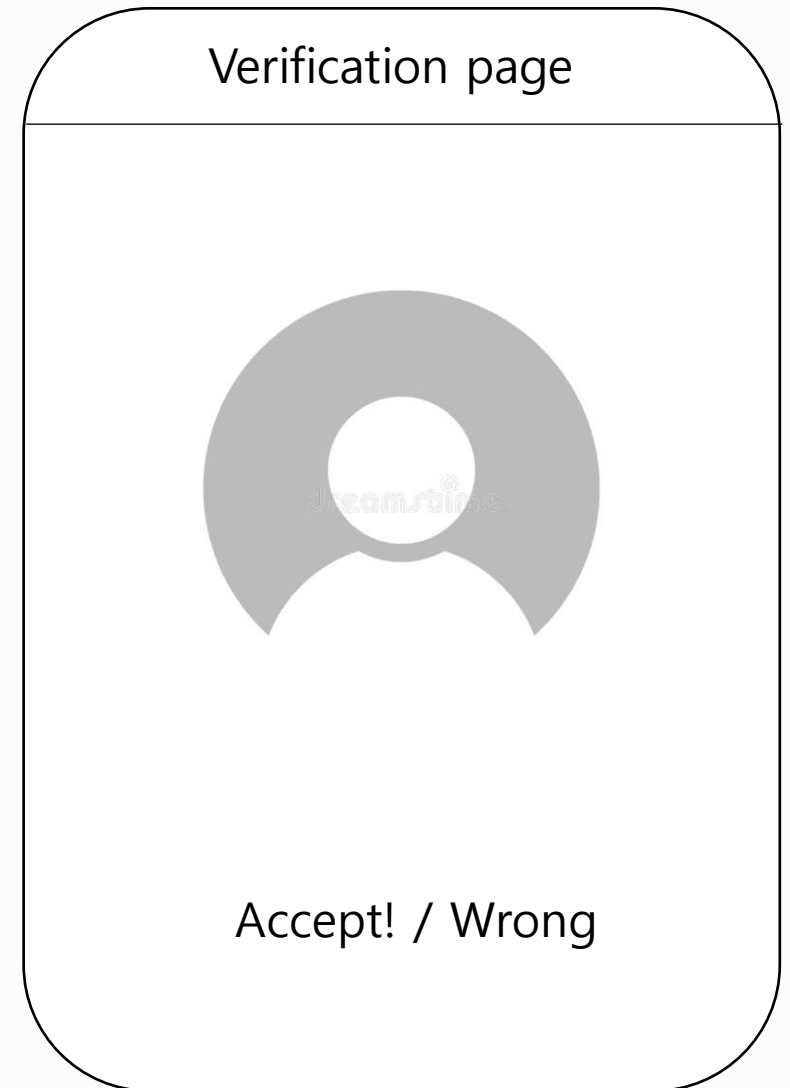
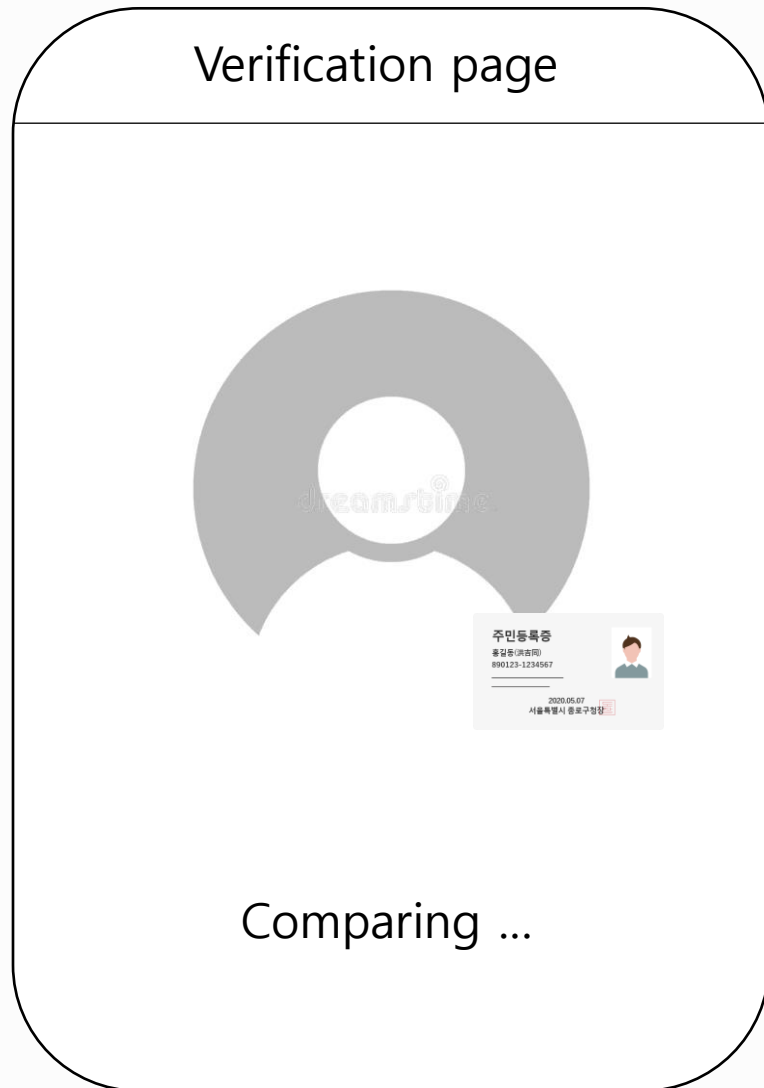
4. Display



Plans

- Implement Screen Sharing using WebRTC, Implement Questionnaire Board, Implement test result Board
- DB interworking
- Server deployment
- Implemented function to extract photo and name from ID card
- Add Alarm Function
- Find an anti-spoofing network suitable for our program.
- Poor performance in CPU environments -> Using GPU server
- Take a long time and Many files need to be saved -> Using DB and Deleting Calibration process
- Add Alarm Function
- Improving performance on webcams

Target Output : Verify Page



Target Output : Test Page


Client Screen

1. A is Apple

☐ True ☐ False




2. B is Banana




☐ True ☐ False



Professor : Hi lim
Lim : Hi professor
...
...

Supervisor Screen





Alert