

발표 대본 (script for presentation, 2021320018 Taehee Jeong)

## Slide 5

On the **Client-side**, the main function offers two processes. In **Process 1**, users put in their Student ID, which is sent to the server to retrieve available classrooms. After selecting a classroom, the system submits the attendance. In **Process 2**, users can directly check the attendance list.

On the **Server-side**, these APIs are implemented. The **GET /attendance** API retrieves attendance records. The **POST /verify-attendance** API verifies attendance by checking for a valid MAC address. The **POST /clear-attendance** API clears existing attendance data. Additionally, **POST /register-mac** registers new MAC addresses for validation, while **GET /valid-macs** fetches all valid MAC addresses. Lastly, **POST /clear-mac-addresses** removes all stored MAC addresses.

(Central)

/1/ When students attempt attendance, /2/ program will capture network packets, sending them to the server for attendance verification, /3/ and validate them by matching a valid MAC address. /4/ The server then sends the result back to the client, completing the process securely via HTTPS.

## Slide 6

And, this is our Demo Video (23초 + 3~4초 소요 예정)

Slide 5 script 참고해서 영상 돌아가는 동안 실시간 부연설명

## Slide 7

There are two parts for further improvement.

First is the **accuracy of location checks**. Currently, packets can be captured even when the student is near, but not inside, the classroom. To address this, we propose using multiple access points (APs) and analyzing signal strength to pinpoint precise locations.

Second, ensuring **persistent presence** throughout the class. A student may attend initially but leave afterward. To solve this, packets will be captured multiple times: the first verification is immediate, followed by two additional checks at 30-minute intervals. For not interpreting students' study, this should be done in background. These results are sent during the next attendance check.

(시간 너무 남으면 아래 것도 포함시키기 )

Some additional improvement such as distinguishing users into students and professor is needed. Currently, roles are selected by users, so we can change it by User Authentication.

**Slide 8** (시간 보고 남으면 역할 소개, 아니면 바로 감사인사)

This our team absent catcher's project. Thank you for listening.

Is there any question?