

# 블록체인 기반 출입통제 시스템

2019. 10. 29

중부대학교 정보보호학과

지도교수 : 양환석 교수님



2 조

조영선  
김민정  
김예진  
이정한  
조수홍

- 조원 편성
- 주제 선정
- 구 상 도
- 추진 경과
- 개발 환경 및 개발 내용
- 개발 시스템 운영
- 결론 및 기대효과

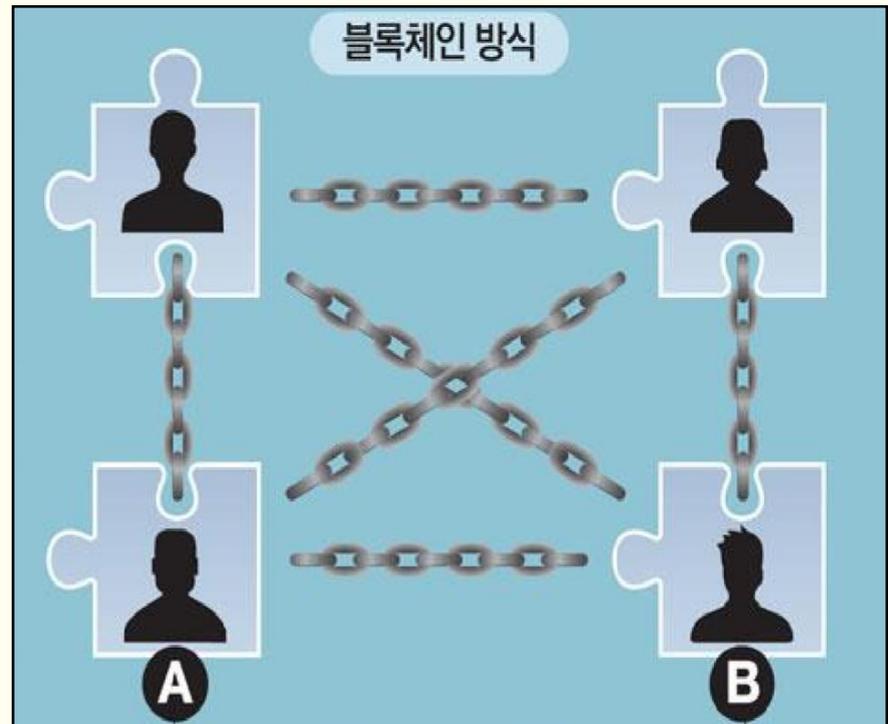
이름	역할
조영선(조장)	얼굴인식기술 개발, 작품 총괄
김민정	블록체인 개발, ppt 및 보고서 작성
김예진	블록체인 개발, GUI 개발
이정한	블록체인 개발, GUI 개발
조수홍	블록체인 개발, ppt 및 보고서 작성

# 주제 선정(1/2)

## 블록체인이란?

'블록'으로 불리는 데이터들이 연결된 체인 형태의 분산 원장 기술

- ◆ 중앙집중기관 없이 시스템 참가자들이 **공동**으로 거래 정보를 기록 · 보관 · 검증
  - ▷ 거래 정보의 신뢰성 확보
  - ▷ 데이터 위 · 변조 방지
  - ▷ 탈중앙화, 보안성, 투명성을 보장



# 주제 선정(2/2)

## 주제 선정

[구축사례] **정부청사 출입통제 얼굴인식 시스템**

얼굴인식 솔루션 도입해 출입보안 강화

[컴퓨터  
설비  
수행  
을 초  
이  
가  
사

**운전면허증, 스마트폰에 담긴다.. 개인정보는 블록체인  
기술로 보호**

도예리기자 | 2019-10-02 14:40:48

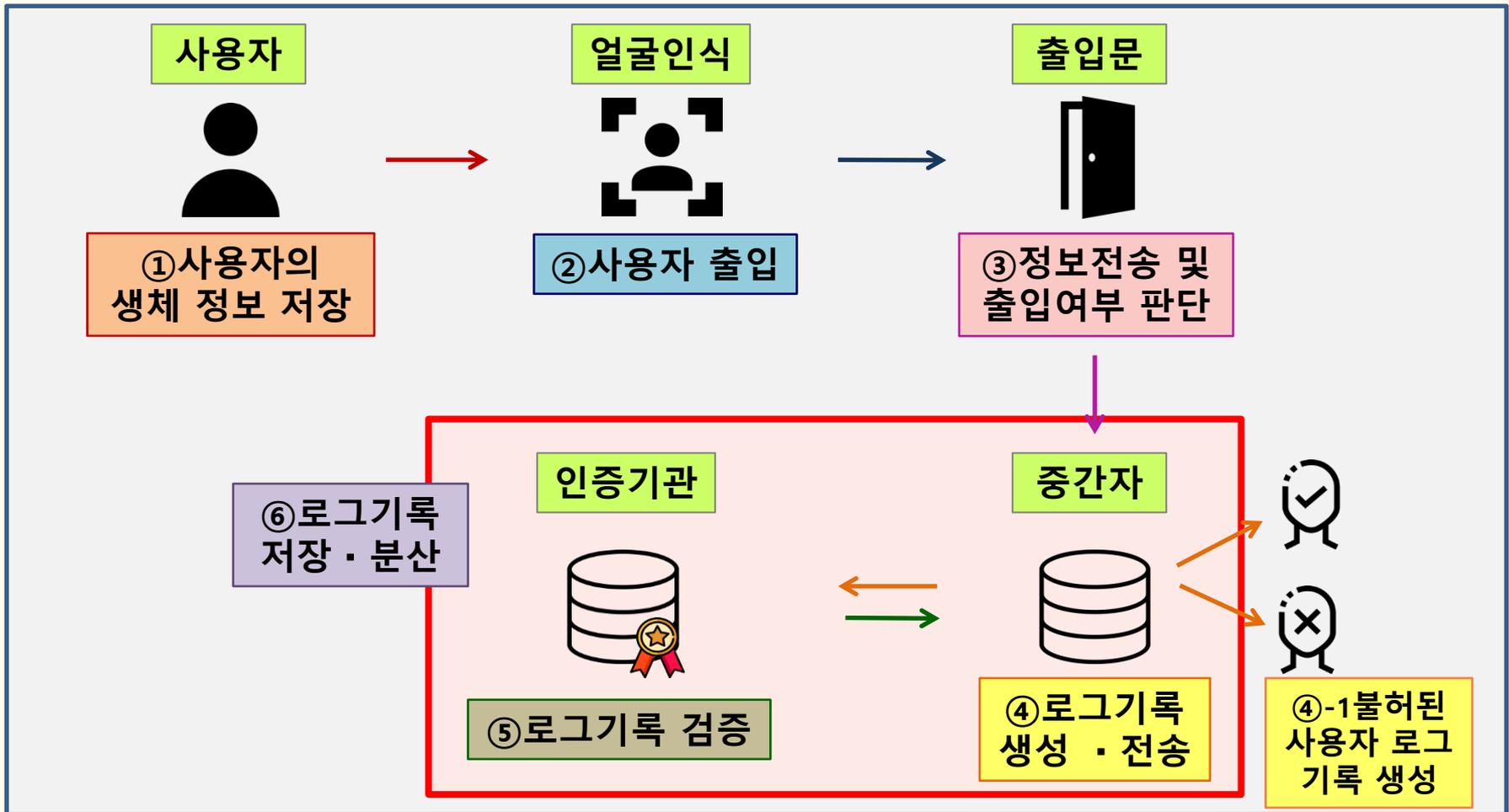


스마트폰에 운전면허증이 담긴다. 운전면허증 상 개인정보는 블록체인 기술로 보호된다.

2일 경찰청은 서울 중구 롯데호텔에서 '모바일 운전면허 확인 서비스'를 추진하기 위해 SK텔레콤, KT, LG유플러스 등 통신 3사와

**출입기록의 데이터 위·변조에 대한 위험성을 줄이기 위해  
블록체인 기술을 사용하여 출입통제 시스템 구현**

## 동작 원리



# 추진 경과

수행 업무	추진기간 (2019년)								
	3월	4월	5월	6월	7월	8월	9월	10월	
자료 조사 및 연구	■								
얼굴인식 기술개발		■							
블록체인 개발		■							
GUI 개발					■				
시스템 검증 및 보완						■			

## 개발 환경

### Tools

- 얼굴인식 : Dlib , opencv
- 블록체인 : hashlib, socket

### OS

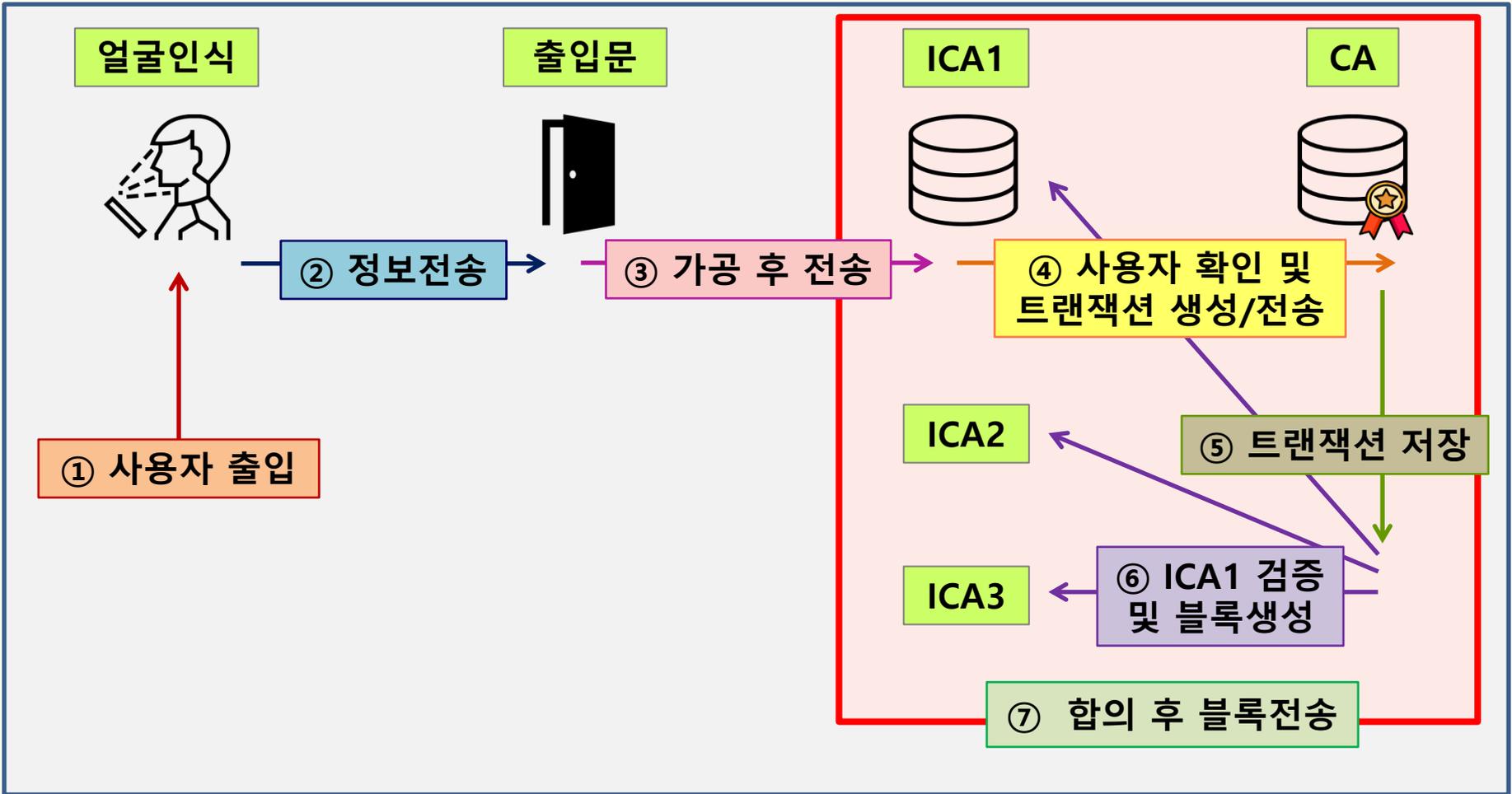
Windows 10

### Development Language

Python 3.6

## 블록체인 활용체계

Certificate Authority ⇨ 중앙인증기관  
Intermediate Certificate Authority ⇨ 중간 인증기관



# 개발 환경 및 개발 내용(3/8)

## 얼굴인식 생체정보 관리(1/2)

```
video_capture = cv2.VideoCapture(1)

path_dir = r'C:\Users\lee\Desktop\training'
images_name = os.listdir(path_dir)

names = ""
del known_face_encodings = []
del known_face_names = []

for image in images_name:
    print((image.split(".")[0]))
    name_image = face_recognition.load_image_file(path_dir + "\\\\" + image)
    name_face_encoding = face_recognition.face_encodings(name_image)[0]
    known_face_encodings.append(name_face_encoding)
    known_face_names.append((image.split(".")[0]))
```

생체정보 저장  
- 캠 활성화

생체정보 저장  
- 인식과정

① 캠 활성화 ② training 파일의 저장된 사용자 정보를 가공 후 저장  
(사용자 생체정보 생성 · 저장)

## 얼굴인식 생체정보 관리(2/2)

```
if process_this_frame:

    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
    face_names = []

    for face_encoding in face_encodings:
        face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
        min_value = min(face_distances)
        name = "Unknown"
        now = datetime.now()

        if min_value < 0.435:
            index = np.argmin(face_distances)
            name = known_face_names[index]
            value = "True"
            print("등록된 사용자입니다. [True]")
        else :
            value = "False"
            print("등록되지 않은 사용자입니다. [False]")
        day = (name,value,
            '%s-%s-%s-%s:%s:%s' % (now.year, now.month, now.day, now.hour, now.minute, now.second))
        result_value=[day]
```

생체정보 관리 · 전송

**현재 비디오 프레임 속 얼굴과  
저장된 생체정보가 일치하는지 확인 후 결과 전송**

## 트랜잭션 생성 및 전송

```
elif(data.get('type') == 'face'):
    name = data.get('data')
    n1 = name.replace("[('", "'")
    n2 = n1.replace("'", "")

    if(n2 in names):
        passfail = data.get('passfail')
```

사용자 확인 과정

```
if data:
    if(data.get('type') == 'transaction'):
        block.input_transaction(data)
        tranNsign = copy.deepcopy(data)
        tranNsign.update({'type': 'tranNsign', 'cp_name':self.cp_name, 'cp_sign':self.cp_sign})
        global ui
        ui.setText("\n"+str(tranNsign))
        self.__debug_client("서명 검증을 위한 서명 전송.")

        self.__send(ca_conn, tranNsign)

    elif(data.get('type') == 'tranNsign'):
        data.update({'type': 'transaction'})
        block.input_transaction(data)
```

트랜잭션 생성 · 전송

사용자 확인 절차 및 트랜잭션 생성 · 전송

# 개발 환경 및 개발 내용(6/8)

## 블록 생성 · 전송 및 합의(1/2)

```
class Blockchain(object):
    def __init__(self):
        self.chain = []
        self.current_transactions = []

        self.vote = []
        self.last_block = {}
        self.new_block(previous_hash = 1)
```

블록 생성

```
def new_block(self, previous_block, nonce=None, difficulty=None, mark=None):
```

```
def new_transaction(self, number, user, checkpoint, time, pass_fail):
    self.current_transactions.append({
        'type': 'transaction',
        'number': len(self.current_transactions)+1,
        'user': user,
        'checkpoint': checkpoint,
        'time': time,
        'pass_fail': pass_fail,
    })
```

트랜잭션 생성

```
return self.last_block['index'] + 1
```

트랜잭션 생성과 블록생성

## 블록 생성 · 전송 및 합의(2/2)

```
def verify_sign(self, vk_sign, vk_name):  
    self.vk_sign = vk_sign[:48]  
    global ui
```

검증 과정

```
with open('verify_signfile.pickle', mode='rb') as file:  
    data = pickle.load(file)
```

```
if data:  
    if(data.get('type') == 'tranNsign'):  
        sign_PF = self.verify_sign(data.get('cp_sign'), data.get('cp_name'))  
        if sign_PF:  
            del data['cp_name']  
            del data['cp_sign']  
            ui.setText(str(data))  
            self.__send_all(s, data)  
    elif(data.get('type') == 'block'):  
        self.__send_all(s, {'type':'vote', 'data':sign_PF}, 1)  
    elif(data.get('type') == 'r_vote'):  
        if (data.get('data')==0):  
            count_vote_F +=1
```

생성된 블록에 대한 합의과정

- ① 신뢰할 수 있는 정보(트랜잭션) 인지에 대한 검증
- ② ICA간 합의 진행
- ③ 블록전송

## GUI

```
def retranslateUi(self, Main_F):
    _translate = QtCore.QCoreApplication.translate
    Main_F.setWindowTitle(_translate("Main_F", "GUI Test"))
    Main_F.setToolTip(_translate("Main_F", "<html><head></body><p>hi</p><p><br/></p></body></html>"))
    Main_F.setWhatsThis(_translate("Main_F", "<html><head></body><p>h</p></body></html>"))
    self.label.setText(_translate("Main_F", "<html><head></body><p align=\\\"center\\\"><span style=\\\" col
self.label_2.setText(_translate("Main_F", "<html><head></body><p align=\\\"center\\\"><span style=\\\" f
self.label_3.setText(_translate("Main_F", "<html><head></body><p align=\\\"center\\\"><span style=\\\" f
self.pushButton.setText(_translate("Main_F", "추가"))
self.pushButton_2.setText(_translate("Main_F", "삭제"))
self.label_4.setText(_translate("Main_F", "사용자 이름"))
self.label_5.setText(_translate("Main_F", "전화번호"))
self.label_6.setText(_translate("Main_F", "이메일주소"))
self.label_7.setText(_translate("Main_F", "해당부서"))
self.label_8.setText(_translate("Main_F", "마지막 출입시간"))
self.pushButton_3.setText(_translate("Main_F", "1"))
self.pushButton_4.setText(_translate("Main_F", ">"))
self.pushButton_5.setText(_translate("Main_F", "<"))
self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), _translate("Main_F", "사용자 관리"))
self.plainTextEdit.setPlainText(_translate("Main_F", "CA"))
self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_3), _translate("Main_F", "로그 기록"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Main_F = QtWidgets.QDialog()
    ui = Ui_Main_F()
    ui.setupUi(Main_F)
    Main_F.show()
```

GUI

시스템의 전반적인 그래픽 사용자 인터페이스

# 개발 시스템 운영 (1/6)

## 시스템 운영(1/6) - 통신관리



**CA-ICA-출입문 연결**    **-출입문-얼굴인식 서버끼리 연결**

# 개발 시스템 운영 (2/6)

## 시스템 운영(2/6) - 사용자관리

사용자 관리 | 로그 기록 | 실시간 블록

등록된 사용자 수 : 5

사용자  
 추가 · 삭제 버튼

메인 화면

추가    삭제

사용자 삭제완료

사용자 관리 | 로그 기록 | 실시간 블록

<input type="checkbox"/>	사용자 이름	전화번호	해당부서
<input type="checkbox"/>	김민정	010-1111-1111	개발
<input type="checkbox"/>	김예진	010-2222-2222	개발
<input type="checkbox"/>	조영선	010-5555-5555	개발
<input type="checkbox"/>	조수홍	010-6666-6666	개발

### 사용자 추가 사용자 추가화면

사용자 이름  
 김민정

---

이메일 주소  
 joognbu1@naver.com

---

전화번호  
 010-1111-1111

---

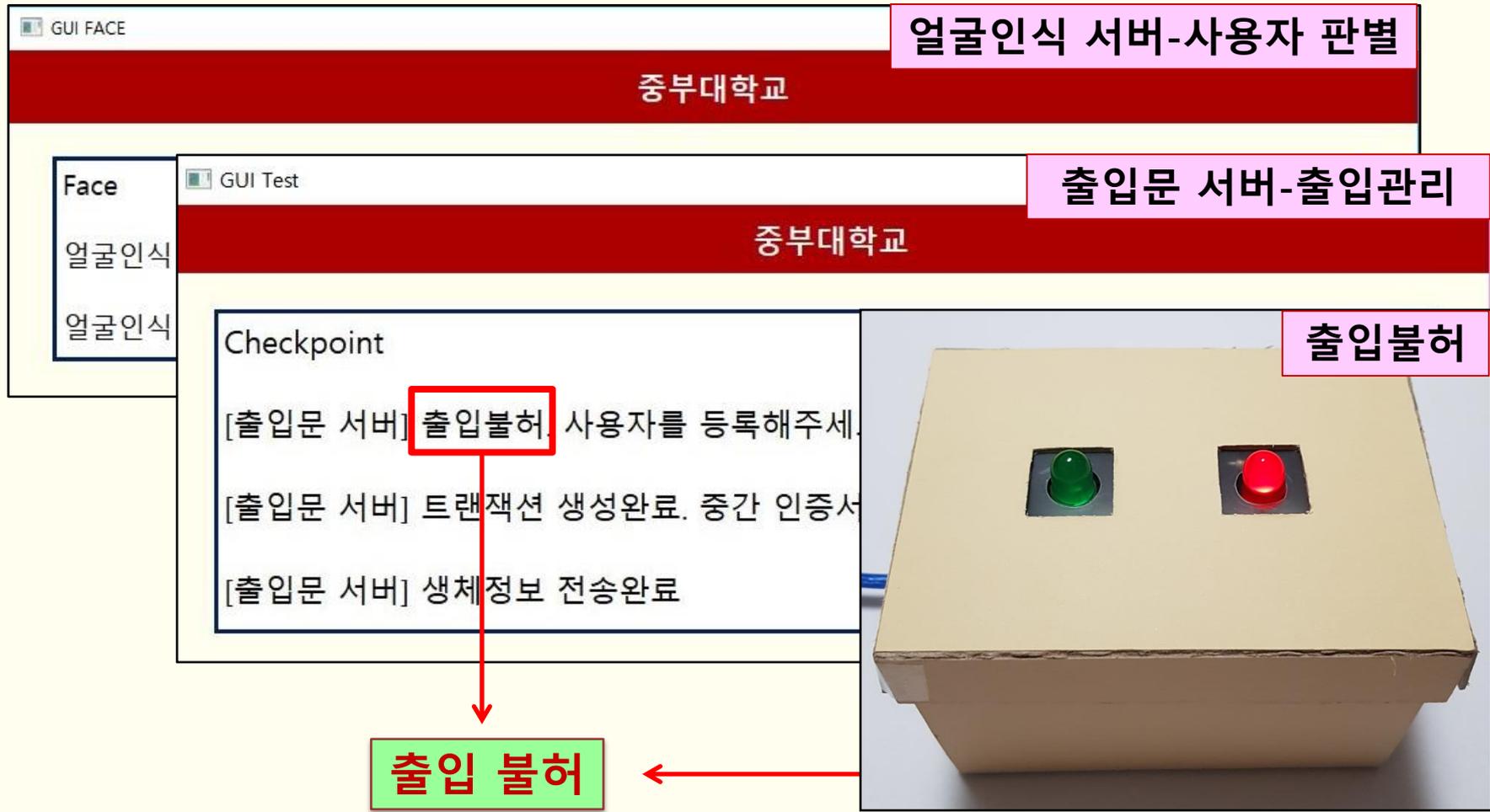
해당부서  
 개발

---

사용자 정보

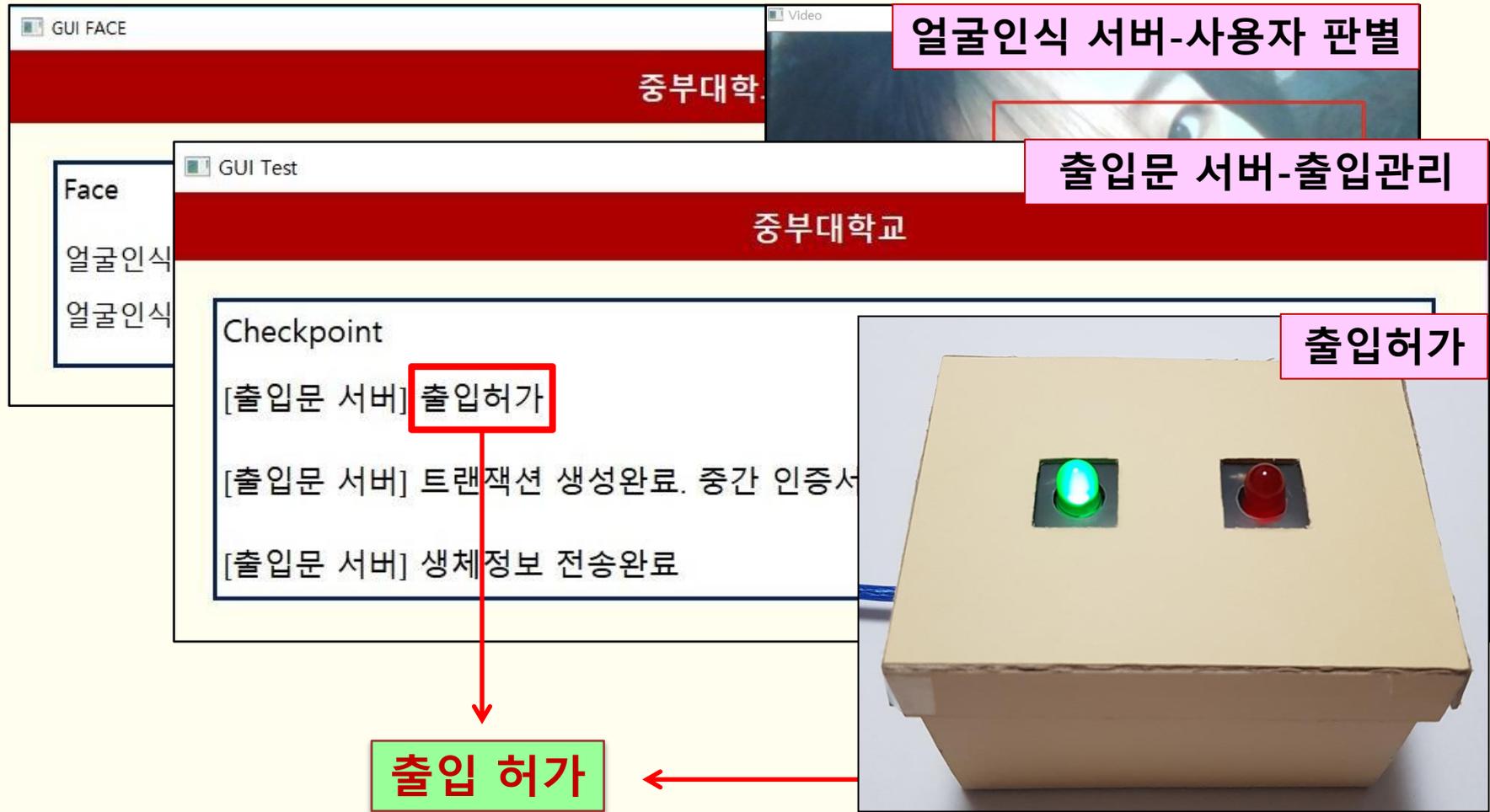
# 개발 시스템 운영 (3/6)

## 시스템 운영(3/6) - 출입관리



# 개발 시스템 운영 (4/6)

## 시스템 운영(4/6) - 출입관리



# 개발 시스템 운영 (5/6)

## 시스템 운영(5/6) - 출입관리

사용자 관리		로그 기록	실시간 블록	출입기록 관리-마지막 출입시간 변경		
등록된 사용자 수 : 4				<a href="#">추가</a>	<a href="#">삭제</a>	
<input type="checkbox"/>	사용자 이름	전화번호	해당부서	이메일주소	마지막 출입시간	
<input type="checkbox"/>	김민정	010-1111-1111	개발	joongbu1@naver.com	2019년 10월 09일	
<input type="checkbox"/>	김예진	010-2222-2222	개발	joongbu2@naver.com	2019년 10월 09일	
<input type="checkbox"/>	조영선	010-5555-5555	개발	joongbu4@naver.com	2019년 10월 11일	
<input type="checkbox"/>	조수홍	010-6666-6666	개발	joongbu5@naver.com	2019년 10월 09일	
<input type="checkbox"/>	<b>조영선</b>	010-5555-5555	개발	joongbu4@naver.com	<b>2019년 10월 09일</b>	
<input type="checkbox"/>	<b>조영선</b>	010-5555-5555	개발	joongbu4@naver.com	<b>2019년 10월 11일</b>	
< 1 >						

**출입한 사용자의 정보에 따라 출입시간 변경**

# 개발 시스템 운영 (6/6)

## 시스템 운영(6/6) - 블록관리

GUI Test

중간 인증기관 서버  
-트랜잭션 · 블록 전송

중부대학교

ICA1

메인화면-생성된 블록 확인

[첫번째]

[첫번째]

[첫번째]

사용자 관리

로그 기록

실시간 블록

```

블록번호 : 4
발생시간 : 2019.10.07.16:52
이전해쉬 : f665a62063fa892e4ad3236ea3c3419f6f09f9ddfa6a38991727c49bf87cc045
=====4번 블록의 트랜잭션 시작=====

트랜잭션번호 : 1
발생시간 : None
출입자 : unknown
체크포인트 : checkpoint1
개폐여부 : False

=====4번 블록의 트랜잭션 끝=====
                    
```

실시간 블록 화면에서 생성된 블록 확인

# 결론 및 기대 효과

## ○ 결 론

- 자체 기술력으로 얼굴인식 기술을 구현하고, 이를 기반으로 출입통제 및 블록체인을 개발하는데 성공
- 조원들에게 적절히 임무를 분담하여 필요 기술을 직접 구현하고 팀워크로 연구하는 조직체제를 가동, 기술역량을 배가

## ○ 기대효과

- 블록체인 기반 출입통제 시스템을 완성함으로써 데이터 위·변조를 방지하고 얼굴인식 기술 등을 응용, 출입통제 체제의 보안성과 편의성을 향상
- 인증된 사용자만을 출입통제함으로써 비인증 외부인의 출입을 차단, 외적 위험요소를 1차적으로 제거. 끝.

Q & A

감사합니다