

블록체인을 활용한 카페 이용시간 관리 시스템 구현

박재훈¹ · 권혁동² · 서화정^{3*}

Implementation of Management of Café Staying Time

Jae-hoon Park¹ · Hyeok-dong Kwon² · Hwa-jeong Seo^{3*}

¹Graduate Student, Department of IT Engineering, Hansung University, Seoul, 02876 Korea

²Graduate Student, Department of Information Computer Engineering, Hansung University, Seoul, 02876 Korea

^{3*}Associate Professor, Department of IT Engineering, Hansung University, Seoul, 02876 Korea

요 약

COVID-19 바이러스가 발생한 이래로 방역수칙에 관한 다양한 단계가 제정되었다. 1~3단계 중 2.5단계의 경우 카페 매장 내 취식은 불가하며 오직 테이크 아웃만이 허용되었으나, 최근 정부에서 조치사항을 변경하여 매장 내에서의 취식이 가능하게 되었다. 하지만 취식 가능 시간은 1시간으로 권고하였으나, 대형 매장을 제외하고서는 잘 이루어지지 않으며 그마저도 수기 위주로 관리가 되고 있다. 본 논문에서는 이에 대해 사용할 수 있는 시스템을 IBM의 프라이빗 블록체인 프레임워크인 하이퍼레저 패브릭을 통해 구현하였다. 이 시스템을 통해 향후 COVID-19에 대한 더 확실한 방역을 기대한다. 또한, 향후 이 시스템을 백엔드로 하여 애플리케이션으로의 개발 또한 진행해볼 것이다.

ABSTRACT

Since COVID19 virus has been occurred, variety steps of the quarantine guidelines were enacted. In step 2.5, eating in a cafe was prohibited and only takeout was permitted, but the government has changed several rules. So now people can have a dessert or coffee in the cafe. Though the government advised available staying time as 1 hour, it is not obeyed often except superstore, and even the management depends on handwriting. In this paper, we implemented a blockchain system to use for this case. This system is implemented with using hyperledger fabric, the blockchain framework which is made by IBM. For a test, 1 organization is in the system and chaincode is installed to the organization to run the system. We expect the certain effect to the quarantine via this system. Moreover, we will develop an application by using this system as a backend.

키워드 : 블록체인, 카페, 하이퍼레저 패브릭, 코로나19

Keywords : Blockchain, Cafe, Hyperledger fabric, COVID19

Received 24 April 2021, Revised 1 May 2021, Accepted 18 June 2021

* Corresponding Author Hwa-jeong Seo(E-mail:hwaJeong84@gmail.com, Tel:+82-2-760-8033)

Associate Professor, Department of IT Engineering, Hansung University, Seoul, 02876 Korea

Open Access <http://doi.org/10.6109/jkiice.2021.25.7.946>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

코로나바이러스감염증-19(이하 코로나19)가 시작된 후로 사람들 사이의 접촉이 감염의 주요 원인이 된다는 것이 널리 알려진 이후 정부에서는 ‘사회적 거리두기’라는 안을 마련하고 그것에 대해 표 1과 같이 여러 단계를 정하였다. 하지만 이런 거리두기 조치에도 불구하고 그림 1과 같이 2020년 12월 일일 확진자 수가 1000명을 넘게 됨에 따라 사회적 거리두기 단계는 2.5단계로 거의 고정되어 있는 상태이다. 그에 따라 일반 식당은 매장 이용이 21시까지 가능하고, 카페는 오직 테이크아웃만 가능하게 되었다. 또한 거리두기 지침이 더욱 강화되어 수도권에서는 5인 이상의 인원수에 대한 집합금지 명령도 내려졌다.

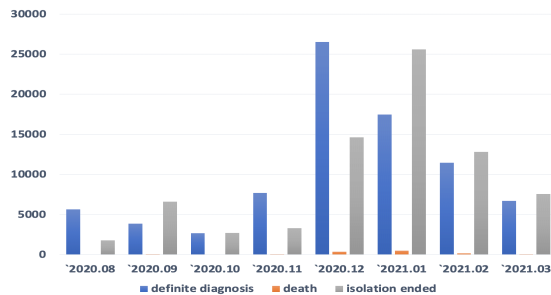


Fig. 1 Monthly COVID19 confirmed[1]

Table. 1 Main quarantine in facilities[2]

	step 1	step 1.5	step 2	step 2.5	step 3
	Quarantine in living	Spreading locally		Spreading national	
Priority managed facility	Must follow core quarantine rules (e.g. limit the number of people)	Reinforce limiting the number of people, prohibit dangerous activity	Prohibit gathering in 5 types entertaining facilities Close the facility after 21'o clock, apply one strike out when disobeyed	prohibit gathering in a public rations officer, karaoke, indoor standing concert hall	Limit operating not only gathering prohibited facility except essential facility
Normal managed facility	Operating normally, must follow 3 quarantine rules	Limit number of people as the facility's feature	Reinforce limiting the number of people, prohibit dangerous activity	Close the facility after 21'o clock, apply one strike out when disobeyed	
etc.	Operating normally		must put a mask	Limit number of people	
National facility	Limit number of people as 50%	20% in facilities like tracking, horse riding, 50% the others	30% in available facilities	30% in available facilities	Stop operating indoor, outdoor facility
Welfare facility (including day care center)	Operating under a rigorous quarantine * Consider the aspect of infection spreading, degree of risk, situation of quarantine and stop several facilities and only provide essential services like emergency day care service				Advice to stop Maintain essential

그러한 노력을 통해 12월 당시 급증하던 확진자 수는 1월부터 서서히 감소하여 2월에는 일일 확진자 수가 12월 당시의 반 이상 떨어지게 되었다. 하지만 여전히 매일 수백명 가량의 확진자가 생겨나고 있기에, 정부에서는 거리두기 2.5단계 및 5인 이상 집합금지는 여전히 유지하고 있으나 카페의 매장 취식은 허용하게끔 표 2와 같이 거리두기 내용을 수정하였다.

Table. 2 Advice to limit cafe available time[3]

Restaurant, Cafe (including unmanned)	When 2 or people more ordererd just type of drinks or desserts, limit maximum available store time to 1 hour (advice strongly)
	Only takeout and delivery are allowed to restaurant and cafe in 21 ~ 5'o clock
	Use 50% of seats by increasing space between tables, and when if hard to obey, set distance of the tables as 1m or install partitions (area over 50m ²)
	In buffet, use sanitizer before using shared tongs or put vinyl gloves, and maintain customer's gap to wait for having foods

정부에서는 카페 매장 내 취식은 1시간으로 강력히 권고하고 있으나, 이는 잘 지켜지지 않으며 통제에 대한 확인조차 제대로 되지 않고 있다. 본 논문에서는 블록체인을 활용하여 카페 매장 이용 수칙 준수에 관한 시스템을 구현하였다.

II. 관련 연구 동향

2.1. 블록체인 (Blockchain)

블록체인은 분산원장을 구현하기 위해 도입된 방식이다. 분산원장이란 기존의 시스템과 같이 중앙의 서버에 사람들이 접속하는 방식이 아닌, 중앙 없이 각 노드들이 데이터를 가진 상태로 운용하는 방식을 말한다. 이러한 방식의 Peer-to-Peer 네트워크는 ‘비잔틴 장군 문제’라는, 각 노드들이 데이터를 각자 갖고 있다고 할 때, 데이터의 위/변조를 어떻게 확인하는지에 관한 문제를 안고 있었다[4]. 블록체인은 이 문제를 데이터를 체인 형태로 엮는 것으로 해결하였다. 사토시 나카모토는 2008년 암호화폐 ‘비트코인’의 구현을 위해 블록체인을 개발하였다[5]. 비트코인은 이 블록체인을 이용하여 중앙 서버 없이 거래 참여자들이 전체 거래 장부를 공유하게끔 만들어졌다.

블록체인에서는 데이터가 블록 형태로 구성되어 있으며, 각 블록들은 자신의 해시와 이전, 이후 블록의 해시, 그리고 거래에 필요한 기타 정보들로 구성되어 있다. 체인을 통해 잘못된 블록이 포함되어 있는지 확인할 수 있다. 또한 새로운 블록의 생성을 위해서는 네트워크 참여자들의 합의를 얻어야 하며, 이에 사용되는 것을 ‘합의 프로토콜’이라고 한다. 비트코인에서는 새로운 블록의 해시의 일부를 컴퓨팅 파워를 이용한 연산을 통해 맞추는 ‘작업증명’ 방식을 이용한다. 이 과정이 마치 광산에서 블록을 캐내는 것과 같다고 여겨져 ‘채굴’이라고 부른다. 작업증명 방식은 확실하지만 컴퓨팅 파워를 지나치게 많이 소비한다는 점 때문에 그에 대한 다른 프로토콜들 또한 등장하였다. 대표적으로 지분증명, 지분위임증명 등이 있다.

프라이빗 블록체인에서는 조금 다른 종류의 합의 프로토콜이 쓰인다. 프라이빗 블록체인은 기본적으로 네트워크 참여자를 제한하여 받기 때문에, 퍼블릭 블록체인에 비해 네트워크 참여자들에 대한 신뢰가 조금 더 있게 된다. 그렇기에 일부 장애를 허용한 ‘비잔틴 장애 허용(PBFT)’ 등의 프로토콜이 이용된다.

블록체인은 상기한 내용과 마찬가지로 중앙 서버가 따로 존재하지 않는 분산원장이어서, 블록체인 시스템을 적용할 때에는 중앙 시스템을 구축할 필요가 없으므로 그만큼의 비용이 절약된다. 그렇기에 여러 점포가 고객들의 출입 정보를 공유 및 관리하는 카페 시스템에 적

용하여, 중앙 서버를 따로 구축할 필요 없이 각 카페에 손쉽게 적용할 수 있다. 중앙 서버를 따로 구축해야 할 경우, 설치 비용이 들면서 그 서버의 보안 관리를 직접 해주어야 한다. 하지만 블록체인을 이용하면 이에 대한 문제를 해결할 수 있다. 본 논문에서는 이러한 블록체인의 장점을 활용하여 설치 비용은 줄이면서 데이터 무결성 등의 보안성은 확보한 시스템을 구현하였다.

2.2. 하이퍼레저 패브릭 (Hyperledger Fabric)

2015년 12월, 리눅스 재단에서는 하이퍼레저 프로젝트를 공개하였다. 이 프로젝트는 프라이빗 블록체인 플랫폼으로, 분산원장과 스마트 컨트랙트를 이용하여 기업 비즈니스 구현에 초점을 맞추었다. 하이퍼레저 프로젝트에는 여러 회사가 참여하여 서비스 중인 프레임워크들이 있으며, 본 논문에서는 그 중 IBM에서 개발 및 서비스 중인 하이퍼레저 패브릭을 구현에 이용하였다.

하이퍼레저 패브릭은 프라이빗 블록체인 프레임워크이며, 기업 비즈니스 구현에 적합하다[6]. 프라이빗 블록체인이기 때문에 CA(Certificate Authority)에 의해 허가를 받은 사용자만이 네트워크에 참여할 수 있다. 하이퍼레저 패브릭은 1.x 버전에서는 Kafka, 2.x 버전에서는 Raft라는 합의 알고리즘을 이용한다[7]. Raft는 Kafka에 비해 트랜잭션 쿼리 능력은 떨어지나, Kafka에 비해 학습 곡선이 훨씬 완만하고 구현이 간단하며 invoke 트랜잭션 시 성공률과 처리량이 더 우수하다[8]. Kafka와 Raft는 모두 CFT(Crash Fault Tolerance) 방식이며, 이 방식은 BFT에 비해 과정이 훨씬 간단하여 데이터를 더 빠르게 처리할 수 있다. BFT 방식에 비해 보안성이 떨어진다고 여겨질 수도 있으나, 모든 네트워크 참여자들이 CA에 허가를 받은 만큼 기본적으로 신뢰를 기반으로 하고 있다. 그렇기에 더욱 복잡한 합의 알고리즘을 필요로 하지 않는다.

비즈니스 로직 구현을 위해 하이퍼레저 패브릭은 ‘체인코드’를 사용한다. 이는 이더리움 등에서 쓰이는 스마트 컨트랙트와 비슷한 개념이다. 체인코드에는 Go, 자바, node.js 세가지 언어를 통해 작성될 수 있다. 체인코드를 이용하여 원장에 있는 내용을 읽어오거나 새로운 내용을 추가 혹은 업데이트 하는 등의 작업을 할 수 있다[9].

2.3. 도커 (Docker)

도커는 컨테이너를 이용한 실행 환경 추상화 시스템이다[10]. 이미지로부터 컨테이너를 생성할 수 있으며, 이렇게 생성된 컨테이너들을 도커를 통해 관리할 수 있다.

하이퍼레저 패브릭의 구성 요소들은 이러한 도커를 이용하여 실행된다. 도커는 서비스를 정의하여 컨테이너를 생성할 수 있는데, 하이퍼레저 패브릭에서의 조직, DB 등이 하나의 컨테이너로써 동작한다.

III. 카페 이용 제한 시스템

본 논문에서 구현한 시스템은 하이퍼레저 패브릭 2.2.2 LTS 버전을 통해 테스트 되었다. 체인코드는 typescript를 사용하였으며 데이터베이스는 NoSQL 데이터베이스인 CouchDB를 이용하였다. 본 논문에서의 네트워크는 HnNetwork라는 네트워크 내에 피어가 1개 있는 Cafe1 조직이 컨소시엄을 구성하고 있으며, 이것은 HnChannel이라는 채널 내에 구성되어 있다. HnChannel은 OrdererOrg라는 오더러 조직을 가진다. 이 오더러에서 네트워크 내에서 쓰이는 블록체인의 블록 생성을 담당하게 된다.

도커 설정을 통해 이와 같은 네트워크의 구성을 정의하였다. 오더러와 Cafe1 조직의 정의는 표 3과 같다.

Table. 3 Docker setting

Organization	orderer	cafe1
Name	OrdererOrg	Cafe1
Peer	-	peer0
Host	orderer.hn.com	peer0.cafe1.hn.com
Port	7050	7051
CA Port	9054	8054
DB Port	-	5984

조직은 기본적으로 오더러 조직 1개와 Café 조직 1개 (Cafe1)로 구성되며, 스크립트를 통해 조직을 추가할 수 있다. cafe1 조직은 피어 1개를 가지고 있다. 또한 각 조직들이 CA를 통해 인증을 진행하며, 오더러는 9054 포트에, cafe1 조직은 8054 포트에 CA를 구동시킨다. 오더러는 DB를 갖고 있지 않기에 cafe1 조직의 피어만이 DB를 가지고 있다. 이 DB는 5984 포트에서 작동한다. 조직의 구성은 그림 2와 같다.

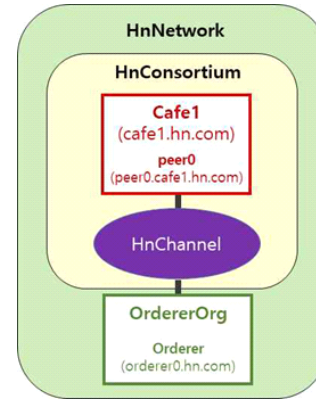


Fig. 2 HnNetwork

네트워크는 network.sh 파일을 통해 시작된다. 구동이 시작되면 먼저 도커를 통해 cafe1 조직의 인증서를 생성하고, 생성된 인증서에 따라 유저와 피어와 어드민을 등록한 뒤 네트워크 프로필을 생성하고 오더러 조직의 인증서를 생성하고 어드민을 등록한다. 그 후 채널을 생성하여 cafe1 조직을 등록시킨 뒤 체인코드를 설치하게 된다.

Table. 4 IO class field

Field	Format	Description
key	string	key value
cafe	string	café name
phone	string	phone number
in_ms	number	enterance time
out_ms	number	exit time
agreeToOfferInfo	boolean	agreed to offer information

설치할 체인코드에서 이용한 모델인 IO 클래스는 표 4와 같다. 키는 'C-{핸드폰번호}-{타임스탬프}' 형식으로 설정된다. in_ms는 입장하는 순간의 타임스탬프로 설정되며, out_ms는 퇴장하는 순간의 타임스탬프로 설정된다. agreeToOfferInfo가 false일 경우에는 카페 입장이 거부된다. 체인코드의 주요 기능을 하는 함수는 gettingIn, gettingOut, checkDisobeyed가 있다.

gettingIn 함수는 표 5와 같다. 입장 시 정보제공동의 여부를 물어봐 만약 false일 경우에는 입장을 제한한다. 입장 시간인 in_ms는 입장하는 순간의 타임스탬프를 저장하며, out_ms는 -1로 기록한다. gettingOut 함수는 표 6과 같다. gettingOut 함수에서는 입력받은 핸드폰 번호

며, out_ms 값은 -1이 되었다. out_ms가 -1인 데이터는 아직 퇴장을 하지 않은 고객이라는 뜻이 된다. 그림 7은 첫번째 데이터에 대해 gettingOut 함수를 사용한 결과이다. out_ms 값이 퇴장시간으로 설정되었다.

```
[
  {
    "agreeToOfferInfo": true,
    "cafe": "starbucks",
    "in_ms": 1618798881,
    "key": "C-12345678-1618798881",
    "out_ms": -1,
    "phone": "01012345678"
  },
  {
    "agreeToOfferInfo": true,
    "cafe": "ediya",
    "in_ms": 1618798882,
    "key": "C-34345656-1618798881",
    "out_ms": -1,
    "phone": "01034345656"
  }
]
```

Fig. 5 Result of the function getIOs

```
[
  {
    "agreeToOfferInfo": true,
    "cafe": "starbucks",
    "in_ms": 1618798881,
    "key": "C-12345678-1618798881",
    "out_ms": -1,
    "phone": "01012345678"
  },
  {
    "agreeToOfferInfo": true,
    "cafe": "ediya",
    "in_ms": 1618798882,
    "key": "C-34345656-1618798881",
    "out_ms": -1,
    "phone": "01034345656"
  },
  {
    "agreeToOfferInfo": "true",
    "cafe": "starbucks",
    "in_ms": 1618800463,
    "key": "C-44445555-1618800463",
    "out_ms": -1,
    "phone": "01044445555"
  }
]
```

Fig. 6 Result of the function gettingIn

```
[
  {
    "agreeToOfferInfo": true,
    "cafe": "starbucks",
    "in_ms": 1618798881,
    "key": "C-12345678-1618798881",
    "out_ms": 1618800557,
    "phone": "01012345678"
  },
  {
    "agreeToOfferInfo": true,
    "cafe": "ediya",
    "in_ms": 1618798882,
    "key": "C-34345656-1618798881",
    "out_ms": -1,
    "phone": "01034345656"
  },
  {
    "agreeToOfferInfo": "true",
    "cafe": "starbucks",
    "in_ms": 1618800463,
    "key": "C-44445555-1618800463",
    "out_ms": -1,
    "phone": "01044445555"
  }
]
```

Fig. 7 Result of the function gettingOut

```
[
  {
    "agreeToOfferInfo": "true",
    "cafe": "starbucks",
    "in_ms": 161880046,
    "key": "C-44445555-1618800463",
    "out_ms": -1,
    "phone": "01044445555"
  }
]
```

Fig. 8 Result of the function checkDisobeyed

그림 8은 checkDisobeyed 함수를 사용한 결과이다. 테스트를 위해 임의로 마지막 데이터의 입장 시간을 조정하였으며, 이렇게 조정된 데이터는 out_ms 값이 -1로 아직 퇴장을 하지 않은 상태에 in_ms 값이 현재의 밀리세컨드와 1시간이 차이가 나기에 checkDisobeyed 함수에 탐지가 되어 출력되게 되었다. 이 함수를 통해 입장한지 1시간이 넘는 고객을 검출할 수 있으며, 향후 이렇게 검출된 손님을 블랙리스트에 추가하는 등의 작업 또한 진행해볼 예정이다. 또한 카페 입장 시 QR 체크인 등의 행동을 할 때 이 시스템이 자동으로 연동되어 gettingIn 함수가 실행되고, 나갈 때 QR 체크아웃과 같은 방식을 통해 gettingOut 함수를 실행시킨다면 관리에 대해 완전한 자동화가 가능할 것이다.

IV. 성능 평가

현재 카페에서는 1시간 이용에 대한 체크를 아예 하지 않고 고객들에게 말로써 고지를 해주거나, 아니면 수기를 이용하여 확인을 하고 있다. 고객에게 말을 통해 고지하는 방식은 고객들에게 실효성이 크게 없으며, 실제로 잘 지켜지지도 않고 있다. 프랜차이즈와 같은 대형 매장이 아닌 경우 대부분 이렇게 고객에게 말로써 퇴장 시간을 고지하는 경우가 다수인데, 이런 매장 특성상 고객의 입/퇴장을 관리하기도 버거운 경우가 대부분이다. 더욱이 이러한 상황에서 추가적인 관리를 위해 인력을 따로 배치하는 것은 어려울 것이다. 수기로 작성하는 방식은 고객이 카페에 입장할 때마다 언제 퇴장해야 하는지를 메모 등으로 작성하여 고객에게 전해주는 방식이다. 그렇기에 고객이 입장할 때 작성해주는 인력과 입장해있는 고객들에 대한 확인을 위한 인력이 필요하고, 수기로 작성하였기에 위/변조 또한 가능하다. 하지만 본

논문에서의 블록체인을 활용한 방식으로는, 고객이 입장 및 퇴장할 때의 시간이 자동으로 기록되기에 이에 대해 추가적인 인력이 필요하지 않으며, 1시간 이용 수칙을 어긴 고객에게는 패널티를 부과하는 등의 추가적인 제재 또한 가능하기에 보다 확실한 규제가 가능하다. 이에 대한 이점은 표 8과 같다.

Table. 8 Compare blockchain to others

	Blockchain	Talk	Handwrite
Checkable	O	X	O
Not need manpower	O	X	X
Preventable falsify	O	X	X

Table. 9 API of the backend

	On chaincode
getting-in	gettingIn
getting-out	gettingOut
io	getIO
ios	getIOs
check-disobeyed	checkDisobeyed

전체적인 작동 방식에 대한 테스트는 Angular 11 프론트엔드를 이용하여 진행하였다. 먼저 블록체인 네트워크 상에서 작성된 체인코드를 node.js의 Express 프레임워크를 통해 하나의 백엔드로써 구동하게끔 만들었으며, 이에 대한 API는 표 9와 같다.

프론트엔드는 백엔드에서 정의된 API에 따라 해당 함수들을 실행시킨다. 이를 통해 입장 및 퇴장을 기록할 수 있으며, 전체 출입 기록 및 현재 1시간 이상 카페에 머물러 있는 사람들에 대한 검사가 가능하다.

getting in

cafe

phone

☒ agree to offer info

Fig. 9 Input form

localhost:4200 내용:

submitted successfully

```
{
  "cafe": "twosome",
  "phone": "01011112222",
  "agreeToOfferInfo": true
}
```

phone number to exit

Fig. 10 Output form and result of input form

get all customers

key	cafe	phone	in_ms	out_ms	agreeToOfferInfo
C-12345678-1623307428	starbucks	01012345678	162330742	-1	true
C-34345656-1623307428	ediya	01034345656	1623307429	-1	true

Fig. 11 All records of the customers

disobeyed customers

key	cafe	phone	in_ms	out_ms	agreeToOfferInfo
C-12345678-1623307428	starbucks	01012345678	162330742	-1	true

Fig. 12 Disobeyed customers

그림 9는 고객의 입장을 기록할 수 있는 form이다. 본 테스트에서는 간단한 진행을 위해 이렇게 form을 만들어 하였으며, 실제 서비스가 적용된다면 QR코드 기록과 같은 단순하고 간단한 행동을 통해 이 과정을 대체할 수 있을 것이다. 그림 10은 고객의 입장이 확인될 경우 서버에서 입장이 확인되었다고 출력되는 모습이다. 그림 11은 이 프론트엔드에서 확인한 전체 출입 기록이며, 본 테스트에서는 1시간 넘는 고객에 대한 확인을 위해 데이터 하나의 입장 시간을 임의로 조작하였다. 이에 대한 결과 그림 12에서와 같이 1시간 이용수칙을 어긴 고객에 대한 확인이 가능해졌다. 이러한 방식을 통해 전체 시스템의 타 어플리케이션과의 연계 및 작동을 확인하였다.

이와 같이 블록체인을 활용한 시스템에서는 다양한 이점을 얻을 수 있다. 언제든지 확인이 가능하며, 수기로 작성했을 경우의 고객들에 대한 감시도 필요하지 않으며 고객들이 자신의 카페 입장 시간을 속이는 것도 검출 가능하다.

고객의 정보를 다루는 차원에서도 블록체인은 다른 방식들에 비해 굉장한 이점을 가진다. 카페 입장 시 명부를 수기로 작성하는 것은 매우 큰 위험성을 가진다. 수기 명부에는 매장마다 조금씩 다르나 주로 거주지역, 핸드폰 번호, 입장시간 등이 적힌다. 그런데 이렇게 작성한 정보에 대한 비밀이 보장되지 않고, 다른 사람들도 얼마든지 명부를 통해 타인의 핸드폰 번호나 거주지와

같은 정보들을 알 수 있다. 실제로 과거 이러한 취약점을 이용해 한 여성이 명부를 작성하는 것을 보고 명부를 토대로 그 여성에게 연락을 한 사건이 발생한 적도 있었다. 입장 시 QR코드를 통해 인증하는 방법 또한 절대적으로 안전한 것은 아니다. QR 인증 정보의 경우 점주들에 의해 관리가 되며 역학조사 시 참고 자료가 될 수 있다. 하지만 이것 역시 따로 시스템이 있는 것이 아닌 점주가 관리하는 것이며 그렇기에 해킹 등으로 데이터가 위/변조될 가능성 또한 존재한다. 블록체인을 이용한 시스템은 충분한 인증을 통해 이러한 문제점들을 원천적으로 차단한다. 자신의 정보를 블록체인 네트워크를 통해 관리함에 따라 해킹의 위험성으로부터도 방어할 수 있으며, 그렇기에 타인으로부터 정보가 노출 및 이용될 가능성이 현저히 낮아지게 된다. 또한 블록체인 네트워크를 통해 정보가 관리되기에 역학조사 시에도 위/변조되지 않은 온전한 정보를 제공할 수 있다.

V. 결 론

본 논문에서는 블록체인을 활용하여 카페 체류가 1시간이 넘는 고객을 찾아낼 수 있는 시스템을 구현하였다. 대형 카페의 경우 1시간 이용 수칙을 지키려는 모습을 보이고 있으나 입장 및 퇴장 시간을 수기로 관리하는 정도에 그치고, 규모가 작은 카페의 경우 관리가 거의 되지 않는 모습을 보여 준다. 이 시스템을 통하여 코로나19에 대한 더욱 강경한 대응을 기대할 수 있다. 또한, 추후 이 시스템을 웹사이트 등의 애플리케이션으로까지의 확장 또한 구현할 예정이다.

ACKNOWLEDGEMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00264, Research on Blockchain Security Technology for IoT Services)

References

- [1] CoronaBoard [Internet]. Available: <https://coronaboard.kr/>.
- [2] COVID-19. Summary of Social distance plan [Internet]. Available: <http://ncov.mohw.go.kr/socdisBoardView.do?brdId=6&brdGubun=1>.
- [3] Ministry of Health and Welfare. Social distancing adjustment plan will be implemented from tomorrow [Internet]. Available: https://www.mohw.go.kr/react/al/sal0301vw.jsp?PAR_MENU_ID=04&MENU_ID=0403&page=1&CONT_SEQ=363130.
- [4] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382-401, July. 1982.
- [5] N. Satoshi. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review [Internet]. Available: <https://bitcoin.org/bitcoin.pdf>.
- [6] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, New York: NY, pp. 1-15, 2018.
- [7] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172-181, Jan. 2020.
- [8] B. Beckert, M. Herda, M. Kirsten, and J. Schiff. "Formal specification and verification of Hyperledger fabric chaincode," in *3rd Symposium on Distributed Ledger Technology (SDLT-2018) co-located with ICFEM*, pp. 44-48, 2018.
- [9] H. Yusuf and I. Surjandari, "Comparison of Performance Between Kafka and Raft as Ordering Service Nodes Implementation in Hyperledger Fabric," *International Journal of Advanced Science and Technology*, vol. 29, no. 7, pp. 3549-3554, May. 2020.
- [10] B. B. Rad, H. J. Bhatti, and M. Ahmadi, "An introduction to docker and analysis of its performance," *International Journal of Computer Science and Network Security*, vol. 17, no. 3, pp. 228-235, Mar. 2017.



박재훈(Jae-hoon Park)

2020년 2월: 한성대학교 정보시스템공학과 학사
2020년 3월-현재: 한성대학교 IT융합공학과 석사과정
※관심분야: 블록체인, 네트워크 보안



권혁동(Hyeok-dong Kwon)

2018년 2월: 한성대학교 정보시스템공학과 학사
2020년 2월: 한성대학교 IT융합공학과 석사과정
2020년 3월-현재: 한성대학교 정보컴퓨터공학과 박사과정
※관심분야: 블록체인, 암호구현



서화정(Hwa-jeong Seo)

2010년 2월: 부산대학교 컴퓨터공학과 학사 졸업
2012년 2월: 부산대학교 컴퓨터공학과 석사 졸업
2012년 3월-2016년 1월: 부산대학교 컴퓨터공학과 박사 졸업
2016년 1월-2017년 3월: 싱가포르 과학기술청
2017년 4월-현재: 한성대학교 IT융합공학부 조교수
※관심분야: 암호구현