

# 블록체인을 활용한 카페 이용시간 관리 시스템

박재훈\*\*, 강예준\*, 김원웅\*, 서화정\*\*†

\*\*한성대학교 IT융합공학부 (대학원생)

\*한성대학교 IT융합공학부 (학부생)

\*\*† 한성대학교 IT융합공학부 (교수)

## Implementation of Management System of Customer Spending Time in Cafe with Blockchain

Jae-Hoon Park\*\*, Yea-Jun Kang\*, Won-Woong Kim\*, Hwa-Jeong Seo\*\*†

\*\*Division of IT convergence engineering, Hansung University  
(Graduate student)

\*Division of IT convergence engineering, Hansung University  
(Undergraduate student)

\*† Division of IT Convergence Engineering, Hansung University  
(Professor)

### 요약

COVID-19가 발생한 이래로 보건복지부로부터 여러 방역수칙이 제정되었으며, 최근 확진자의 계속 증가함에 따라 2.5단계가 시행되어 카페 매장 내에서의 취식은 불가하였다. 하지만 COVID-19가 다시금 누그러짐에 따라 카페 매장 내 취식이 1시간에 한해 허용되었으나, 이 수칙을 사람들이 잘 지키지 않을뿐더러 마땅히 잡아낼 수단도 존재하지 않는다. 본 논문에서는 블록체인을 활용한 카페 출입 관리 시스템을 통해 이러한 수칙 위반자들에 대한 검출을 가능케 한다. 또한 향후에는 이 시스템에 백엔드 및 프론트엔드 구성을 통해 하나의 애플리케이션으로서의 개발도 해볼 것이다.

## I. 서론

최근 COVID-19가 누그러짐에 따라 수도권 지역은 사회적 거리두기 2.5단계임에도 카페 매장 내 취식이 가능하게 되었다. 그에 따라 카페 내 취식 가능 시간은 21시까지, 매장을 1시간만 이용할 수도록 권고되고 있다.

식당·카페(무인카페 포함)	<ul style="list-style-type: none"><li>▶ 2인 이상이 커피음료류, 디저트류만을 주문했을 경우 매장 내 머무르는 시간을 1시간으로 제한(강력 권고)</li><li>▶ 식당·카페 모두 21시~익일 05시까지 포장·배달만 허용</li><li>▶ 테이블 또는 좌석 한 칸을 띄워 매장 좌석의 50%만 활용하되, 이를 준수하기가 어려운 경우 ①[고정폭 빈 칸] 테이블 간 1m 거리 간·칸막이/가림막 설치 중 한 가지 준수(시설 면적 50㎡ 이상)</li><li>▶ 뷔페의 경우 공용접점시수저 등 사용 전후 손소독제 또는 비닐장갑 사용, 음식 담기 위한 대가사 이용자 간 간격 유지</li></ul>
----------------	---

(그림 1) 카페 이용제한 권고[1]

하지만 대부분의 사람들이 이 1시간 이용제한을 잘 지키지 않고 있으며, 이것에 대한 확인 조차 명확히 되지 않고 있다. 이러한 상황에 대해 명확한 해결 방안이 요구된다. 본 논문에서

는 블록체인을 활용하여 이러한 문제에 대해 1시간 이용제한 수칙을 잘 지킬 수 있도록 확인 및 통제가 가능한 시스템을 구현하였다.

## II. 관련 연구

### 2.1 블록체인

블록체인이란 기존의 거래 시스템 방식처럼 중앙 서버에서 장부를 관리하는 방식이 아닌, 암호화된 전자 장부를 거래에 참여한 모든 사람의 컴퓨터에 분산화 시켜 보안성을 높이는 방식이다[2]. 즉, 장부를 은행이 독점적으로 관리하는 방식에서 벗어나, 동일한 장부를 모든 사람이 가지고 있는 탈중앙화 형태의 방식이다. 거래에 참여한 모든 사람이 가지고 있는 장부를 계속해서 비교하기 때문에, 한두 명의 장부

를 조작하더라도 이는 조작된 장부라는 것을 바로 알 수 있다.

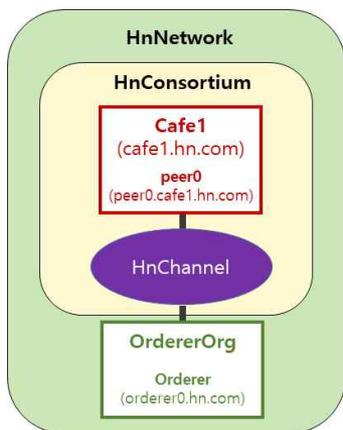
## 2.2 하이퍼레저 패브릭

하이퍼레저 패브릭은 리눅스 재단이 주도하는 오픈소스 블록체인 프로젝트이며 Private Blockchain의 형태를 띤다[3]. Private Blockchain이란 신뢰할 수 있는 기관인 CA(Certificate Authority)에 의해 허가를 받은 사용자만 네트워크에 참여할 수 있는 형태를 말한다. 이러한 과정에 의해 네트워크에 참여하는 노드들은 자연스럽게 신뢰를 가진 노드로 판별되기 때문에 복잡한 합의 알고리즘을 필요로 하지 않게 된다.

하이퍼레저 패브릭 내의 네트워크에 있는 참여자들은 체인코드(Chaincode)를 실행하여 원장에서 기존의 내용을 읽어오거나 새로운 내용을 업데이트 하게 된다[4].

## III. 카페 이용 제한 시스템

본 시스템은 하이퍼레저 패브릭 2.3.0을 통해 구현되었으며, 체인코드에는 typescript를, 데이터베이스는 CouchDB를 사용하였다. 네트워크는 기본적으로 1개의 오더러와 1개의 조직으로 구성되어 있으며, 각 조직에는 1개의 피어가 구성되어 있다. 하나의 카페가 조직을 구성하며, 피어는 이용객의 수에 따라 조절될 수 있다.



(그림 2) 네트워크 구성

네트워크는 쉘 스크립트를 이용하여 실행시키게 되며, 실행 시 그림 2와 같은 형태의 네트워크가 생성 및 실행된다.

본 시스템에서는 카페 이용 고객의 출입 정

보를 기록하기 위해 IO 클래스를 생성하였다. IO 클래스의 필드는 표 1과 같다.

필드	형식	설명
key	string	키값
cafe	string	카페명
phone	string	핸드폰 번호
in_ms	number	입장시간
out_ms	number	퇴장시간
agreeToOfferInfo	boolean	정보제공동의여부

(표 1) IO 클래스 필드

체인코드는 입장할 때 이용되는 gettingIn, 퇴장할 때 이용되는 gettingOut, 1시간 이상 머무르는 고객을 확인하는 checkDisobeyed 메소드로 구성된다.

```

async gettingIn(context: Context, cafe: string, phone: string, agreeToOfferInfo: boolean) {
  if (!agreeToOfferInfo) {
    console.log('not available without agreeing to offer the information');
    return;
  }
  try {
    const io = {
      key: this.generateKey(),
      cafe: cafe,
      phone: phone,
      in_ms: new Date().getTime(),
      out_ms: -1,
      agreeToOfferInfo: agreeToOfferInfo
    };
    await context.stub.putState(io.key, stateValue(io));
    console.log('new IO generated');
  } catch(err) {
    console.log(err);
  }
}
  
```

(그림 3) gettingIn 메소드

gettingIn 메소드는 카페명과 핸드폰 번호를 입력받아 새로운 IO 객체를 생성하여 DB에 추가하는 메소드이다. 객체를 생성할 때 in\_ms 값에 현재의 밀리세컨드 값을 입력함으로써 퇴장시 검증에 이용한다. 또한 out\_ms 값은 -1으로 설정함으로써 향후 out\_ms가 -1인 데이터는 아직 나가지 않은 손님으로 구분할 수 있게끔 한다. 만약 고객이 정보제공동의를 하지 않았을 경우에는 입장이 거부된다.

```

async gettingOut(context: Context, phone: string) {
  try {
    const res = context.stub.getQueryResult(JSON.stringify({
      selector: {
        phone: phone
      },
      limit: 1
    }));
    if (!res) throw 'could not find the given phone number';
    for await (const {key, value} of res) {
      const io = toItem(value);
      io.out_ms = new Date().getTime();

      if (io.out_ms - io.in_ms > 60 * 60 * 1000) {
        console.log('the customer '${io.phone}' stayed in the cafe '${io.cafe}' over 1 hour');
      }
      await context.stub.putState(io.key, stateValue(io));
      return;
    }
  } catch(err) {
    console.log(err);
  }
}

```

(그림 4) gettingOut 메소드

gettingOut 메소드는 사용자의 핸드폰 번호를 입력받아 그 사용자의 퇴장 처리를 진행하는 메소드이다. 메소드가 실행되면 DB에서 핸드폰 번호를 조회하여, out\_ms에 현재의 밀리초값을 입력한 뒤 저장하게 된다. 만약 입장 시간과 퇴장 시간이 1시간 이상 차이날 경우, 알람을 띄우게 된다.

```

async checkDisobeyed(context: Context) {
  const ts = new Date().getTime();
  const res = context.stub.getQueryResult(JSON.stringify({
    selector: {
      $and: [
        {
          out_ms: { $lt: 0 },
        }, {
          in_ms: { $lt: ts - (60 * 60 * 1000) }
        }
      ]
    }
  }));
  const disobeyed: IO[] = [];
  for await (const {key, value} of res) {
    disobeyed.push(toItem(value));
  }
  console.log(`${disobeyed.length} customer(s) existing`);
  return disobeyed;
}

```

(그림 5) checkDisobeyed 메소드

checkDisobeyed 메소드는 DB로부터 out\_ms 값이 0 미만인, 즉 아직 퇴장하지 않은 손님 및 in\_ms 값이 현재의 밀리초값보다 1시간 이상 차이나는, 즉 입당한지 1시간이 초과된 손님에 해당하는 쿼리를 통해 방역수칙을 지키지 않은 손님의 목록을 얻어낼 수 있는 메소드이다. 이 메소드를 통해 어떠한 손님이 1시간을 넘겼는지, 그 수는 몇명인지 알 수 있다.

## IV. 결론 및 향후 연구

본 논문에서는 블록체인을 활용하여 카페에 1시간 이상 머무르는 고객을 검출할 수 있는 시스템을 구현하였다. 이 시스템을 통해 COVID-19에 대한 좀 더 강경한 대응이 가능할 것으로 기대된다.

현재 시스템에서는 도커를 통한 네트워크 및 체인코드 구동만이 구현되어 있는데, 향후에 이 네트워크를 애플리케이션으로서 활용할 수 있도록 백엔드 및 프론트엔드 또한 구현할 예정이다.

## V. Acknowledgment

이 성과는 부분적으로 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2020R1F1A1048478) 그리고 이 성과는 부분적으로 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천 기술 연구).

## [참고문헌]

- [1] “사회적 거리 두기 조정안 내일(1.18일)부터 시행”, 보건복지부 [Internet]. Available: <https://bit.ly/39IjK6l>
- [2] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”
- [3] Androulaki, Elli, et al. “Hyperledger fabric: a distributed operating system for permissioned blockchains.” Proceedings of the thirteenth EuroSys conference. 2018.
- [4] Beckert, Bernhard, et al. “Formal specification and verification of hyperledger fabric chaincode.” 3rd Symposium on Distributed Ledger Technology (SDLT-2018) co-located with ICFEM. 2018.