

Hyperledger Fabric Parking System

2019.06.10

임베디드 소프트웨어

3조

201320878 박용채

201720725 김훈민

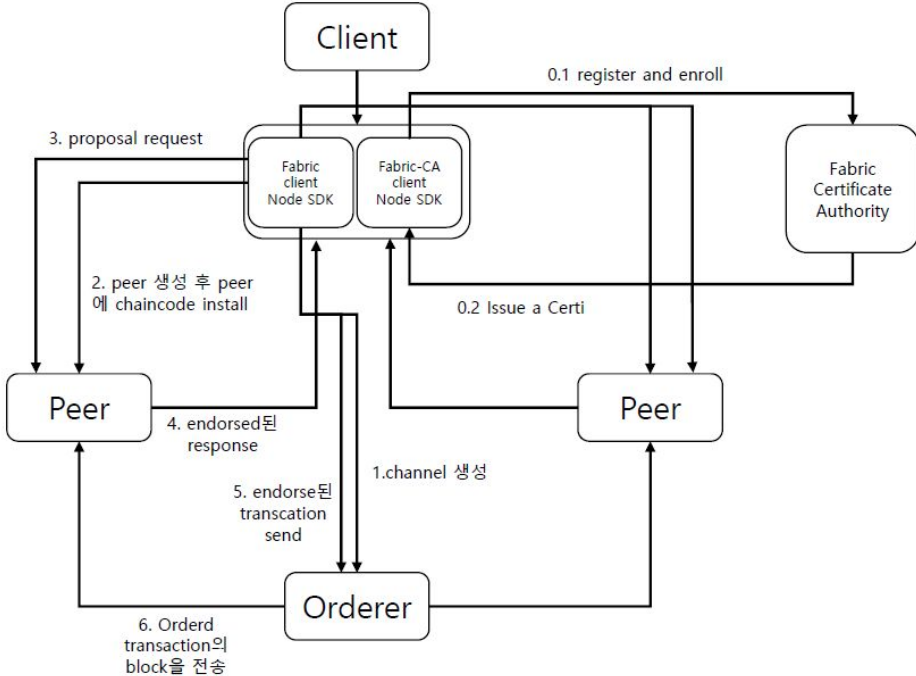
201720746 김용민

201720747 양준모

목차

프로젝트 요약	3
프로젝트 개요	4
동기 및 기대 효과	4
1.1 사회적 측면	4
1.2 기술적 측면	5
기술 사용 현황 (유사 프로젝트)	8
2.1 스마트시티 챌린지	8
2.2 Smarking	8
2.3 젠틀마스	8
아이템 개발 동기	9
3.1 PEST 분석을 통해 문제점 도출	9
3.1.2 Political issue	9
3.1.3 Economical issue	9
3.1.4 Social issue	10
3.1.5 Technical issue	10
3.2 PEST 문제점 해결방안	10
아이템 목적	11
4.1 3C 분석	11
4.1.1 3C 분석 - Customer	11
4.1.2 3C 분석 - Competitor	14
4.1.3 3C 분석 - Company	14
사업화 전략	15
New Business Model	16
프로젝트 설계 및 구현	17
7.1 시스템 구조	17
7.2 Use Case Diagram	18
7.3 Functional Requirement	18
7.4 프로젝트 구현	19
구현 결과	22
역할 분담	24
개발 일정	25
참고 문헌	25
발표 / 데모 영상	26

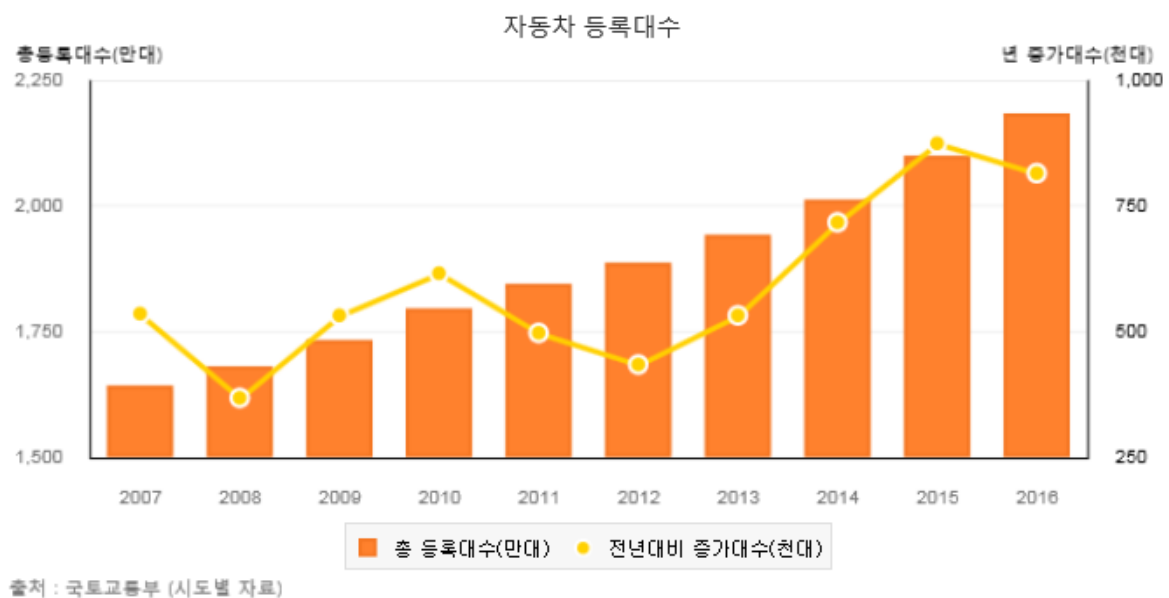
0. 프로젝트 요약

작품제목	Hyperledger Fabric Parking System
제작자	<ul style="list-style-type: none"> ◆ 지도교수 : 김기형 ◆ 조 번호 : 3조 ◆ 팀 원 : 박용채, 김훈민, 김용민, 양준모
제작목표	주차난 해결과 주차를 편하게 하기 위함을 위한 어플리케이션(또는 웹)
작품개요	<p>등록된 자동차의 수가 늘어남에 따라 주차난이 심각해졌다. 주차하기 위한 자리를 찾기가 쉽지 않아 도심에서 주차를 하기란 좀처럼 어려운 일이다. 이를 해결하기 위한 어플리케이션(또는 웹)을 개발하려고 한다.</p> <p>시스템 구성도는 아래와 같다.</p>  <pre> graph TD Client[Client] -- "0.1 register and enroll" --> FabricCA[Fabric-CA client Node SDK] FabricCA -- "0.2 Issue a Certi" --> FabricAuth[Fabric Certificate Authority] FabricAuth -- "0.2 Issue a Certi" --> FabricCA Client -- "1.channel 생성" --> Peer1[Peer] Client -- "1.channel 생성" --> Peer2[Peer] Peer1 -- "2. peer 생성 후 peer 에 chaincode install" --> FabricClient[Fabric client Node SDK] Peer2 -- "2. peer 생성 후 peer 에 chaincode install" --> FabricClient FabricClient -- "3. proposal request" --> Peer1 Peer1 -- "4. endorsed된 response" --> FabricClient FabricClient -- "5. endorse된 transaction send" --> Orderer[Orderer] Orderer -- "6. Orderd transaction의 block을 전송" --> Peer1 Orderer -- "6. Orderd transaction의 block을 전송" --> Peer2 </pre> <p><시스템 구성도></p>

1. 프로젝트 개요

1. 동기 및 기대 효과

[그림1]은 국내에 등록된 자동차의 수를 연도별로 조사한 그래프 이다. 한 눈에 봐도 꾸준히 증가하고 있음을 알 수 있다. 이에 따라 주차 공간 확보의 필요성이 강조되고 있는데, 특히 장애인 구역 불법주차, 주차 공간 센서 고장, 주차 시간 조작, 무인 주차 시스템 오류 등 많은 문제점이 대두되고 있는 상황이다. 본 프로젝트에서는 이러한 문제점을 해결하기 위해, Hyperledger Fabric을 활용한 주차 관리 시스템을 구현해 보고자 한다.



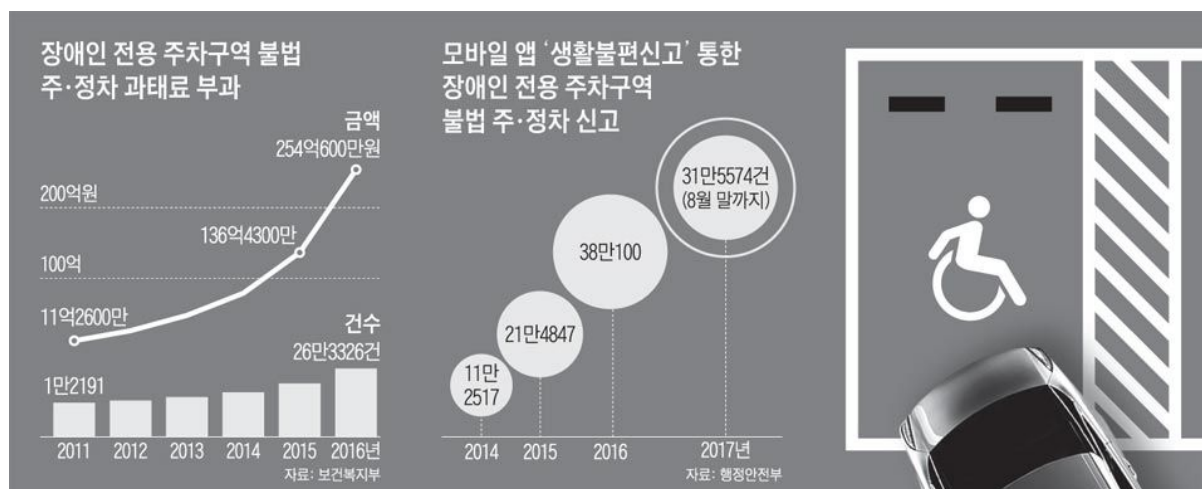
[그림1] 자동차 등록대 수 추이

1.1 사회적 측면

매년 장애인 전용 주차 구역의 민원신고 수가 급증하고 있다. 2018년 보건복지위원회 국정감사에서 발표된 내용을 보면, 2012년 3만 9천 334건에서 2017년 33만 359건 으로 6년동안 29만 1천 25건, 총 7.4배가 증가한 것으로 확인됐다. 또한 [그림2] 에서 볼 수 있듯이, 장애인 전용 주차 구역 위반으로 인한 과태료 부과액도 2012년 19억원에서 2017년 236억 3,900만원으로 66년간 11배가 증가했다. 실제로 장애인 전용 주차구역에 비장애인이 주차를 하면 최대 200만원의 과태료를

물어야 한다. 그러나 현실은 단속 자체도 쉽지 않으며, 특별한 페널티 없이 넘어가는 경우도 허다하다. 이는 현재 주차 관리 시스템의 한계로 볼 수 있다.

이번 프로젝트에서 개발한 Hyperledger Fabric 주차 관리 시스템은 주차 차량의 정보를 블록체인 기술을 활용하여 보관한다. 이를 활용하여, 비 장애인이 장애인 전용 구역에 주차를 한다면 과태료를 바로 지불하도록 하거나, 관리자에게 알람이 가도록 하는 등, 다양한 방법을 이용하여 위의 문제점을 해결 할 수 있을 것이다.



[그림2] 장애인 전용 주차구역 불법 주/정차 과태료 부과 현황

1.2 기술적 측면

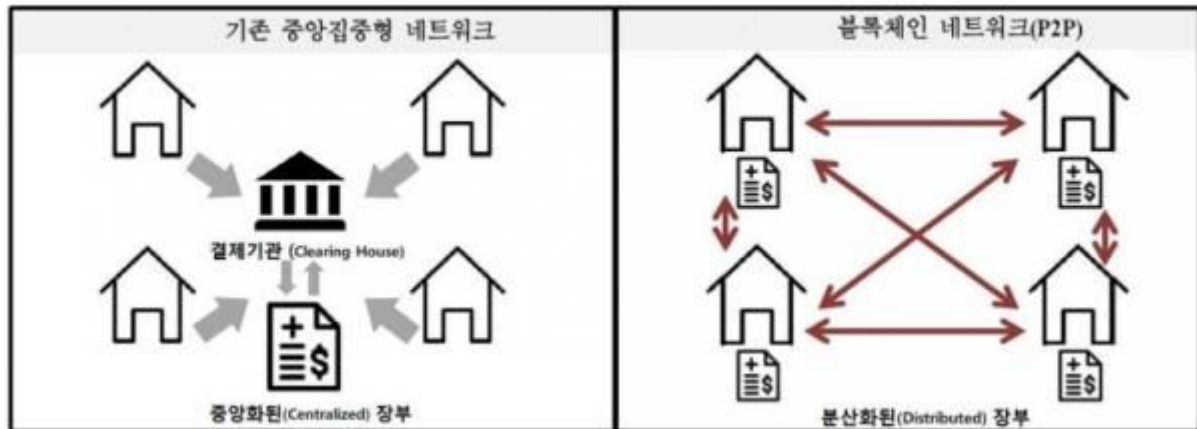
전 세계적으로 암호화폐 열풍이 불기 시작하면서, 블록체인 기술 또한 화제의 기술로 급부상하였다. [그림3]은 현재 블록체인이 사용되고 있는 분야들을 나타낸 그래프이다.



[그림3] 블록체인 적용 분야 및 활용 기업

블록체인 기술의 핵심 중 하나가 분산 장부 기술이다. [그림4]에서 볼 수 있듯이, 이 기술은 중앙화된 장부를 결제기관이 관리하는 기존의 중앙 집중형 네트워크와는 달리, 다수의 참가자가 동기화된 장부를 공동으로 관리하는 기술이다. 블록체인 기술을 활용하여, 일정 시간동안 발생한 거래 내역을 모아 Block 단위로 기록 및 검증하고 이를 기존 Block에 Chain 하여, 인위적인 기록의 변경이나 가감을 사실상 불가능하게 한다. 이것이 분산 장부 기술, 즉 블록체인 기술의 보안성과 안정성을 크게 높인다. 장부를 분산하고 데이터를 동기화하여, 과반수 이상의 Chain된 컴퓨터를 해킹하지 않는 이상 위조가 불가능하기 때문이다.

〈블록체인 네트워크 개념도〉



< 자료 : Santander, 국제금융센터 >

[그림4] 중앙집중형 네트워크와 블록체인 네트워크

현재 대부분의 주차 관리 시스템은 중앙집중형 네트워크를 이용한다. 실제로 중앙집중형 네트워크의 보안성 허점을 이용하여, 주차관리원들이 전산을 조작, 요금 처리를 취소한 후 요금을 가로채는 일이 일어나기도 했다. 블록체인 네트워크를 이용하면, 이러한 문제를 해결 할 수 있으며, 무결성과 투명성 또한 보장할 수 있다는 장점이 있다. 또한 4차 산업 혁명 시대 경제의 핵심 인프라 라고도 불리는 블록체인 기술을 응용하여 블록체인 주차 관리 시스템을 개발한다면, 기존 주차 관리 시스템에 대한 단점을 보완할 뿐만 아니라 블록체인 응용 기술 발전에 이바지할 수 있을 것이다.

1.3 경제적 측면

국토교통부에선 2018년, 651억원을 들여 전국 지방 자치 단체를 지원해, 공영 주차장과 무인 주차시설 등, 주차 관련 시설을 확충한다고 밝혔다. 이 액수는 2017년 451억원에 비해 144% 증액된 것으로, 국토부에서는 전체 사업비의 50%를 도심과 상가 밀집지 등 주차난이 심각한 지역의 주차난 해소를 위해 사용한다고 밝히기도 하였다. 이는 정부에서도 주차 공간 부족 문제를 적극적으로 해결하려 한다는 것을 알 수 있는 부분이다.

이와 같이, 단순히 주차 관련 시설 확충 만으로 주차 공간 부족 문제를 해결하는것은 막대한 비용을 필요로 한다. 본 프로젝트에서는 이러한 기존 주차 관리 시스템의 낮은 경제성을, 블록체인을 활용한 주차 관리 시스템을 개발 함으로써 개선해보았다.

2. 기술 사용 현황 (유사 프로젝트)

2.1 스마트시티 챌린지

2019년 5월 3일 스마트시티 솔루션 기업 데일리블록체인은 국토교통부에서 추진하는 ‘스마트시티 챌린지’ 사업에 참여한다고 밝혔으며, 이 스마트시티 챌린지 사업은 지자체와 민간기업이 협력해 교통과 환경 등의 도시문제를 해결하고 새로운 비즈니스 모델을 만들 수 있도록 지원하는 사업이다. 현재 ‘스마트시티 챌린지’ 사업에는 대전, 인천, 광주, 수원, 부천, 창원 등 여러 도시가 참여하고 있다.

‘스마트시티 챌린지’ 사업은 블록체인 기술을 이용해 도심의 주차 문제를 해소하고자 블록체인 기반 서비스 포털을 구축하여, 주차장 정보 공유, 주차 대행 서비스 등 주차 서비스를 통합 관리해 나가고자 하는 사업이다. 지금까지는 클라우드 기반으로 ‘스마트시티 챌린지’ 사업이 운영되었으나, 현재는 블록체인 기반으로 운영되고 있다.

2.2 Smarking

현재 미국 보스턴과 샌프란시스코 지역에서 서비스 중에 있는 Smarking은 사용자들의 주차장 이용 패턴과 주차 요금, 날씨, 비행 일정, 대규모 행사 개최 등 주차장 이용에 영향을 끼칠 수 있는 요소들을 실시간으로 분석하여 빅데이터를 구축한 후 이를 기반으로 주차장 운영업주는 이를 토대로 현재 남아 있는 주차 공간은 얼마나 되는지, 주차장 이용 현황, 주차장과 관련된 향후 전망치를 실시간으로 파악할 수 있게 된다. 데이터는 그래프 형태로 제공되어 한 눈에 파악할 수 있으며, 웹사이트나 앱으로 주차장 이용자와 정보를 공유할 수 있다.

Smarking을 통해 주차장 업주들은 주차장을 관리하는 주차요원들 같이 불필요한 인원을 감축할 수 있으며, 사용자들과 주차장 정보를 공유함으로써 주차공간이 남아 있거나 주차장 이용이 어려울 것으로 예상되는 날을 사용자들에게 미리 알려줄 수 있다.

2.3 젠틀마스

젠틀마스는 블록체인 플랫폼 전문기업으로 2018년 9월 14일 대한주차산업협회와 블록체인 기반 스마트3.0 주차 플랫폼 구축을 위한 업무 협약을 맺었다. 협약은 블록체인을 통한 주차장 플랫폼 구축이 목적이며 이를 통해 주차산업의 발전과 전국 주차장에서 국내최초로 젠트리온 암호화폐로 주차비 결제를 할 수 있는 계기를 마련할 수 있게 된 것이다. 젠틀마스는 분산원장기술 및 암호화기술을 기반으로 암호화폐 인프라 및 시스템 구축과 함께 블록체인 플랫폼 개발 인프라를 공유하기로 했다. 이 사업이 잘 활성화 된다면 기존 신용카드나 체크카드, 현금으로 결제하던 주차비를 암호화폐로 결제할 수 있는 시대가 열리게 될 것이다.

3. 아이템 개발 동기

3.1 PEST 분석을 통해 문제점 도출

3.1.2 Political issue

현재 정부에서도 블록체인 기술을 주목하고 있으며 이를 이용해 주차난 문제 해결에 힘쓰고 있다. 그에 따라 국토교통부에서 추진하는 ‘스마트시티 챌린지’ 사업 내용 중에 블록체인 기술을 이용한 주차장 자동 관리 시스템도 포함되었다. 지금은 적은 수의 도시만 ‘스마트시티 챌린지’ 사업에 참여하고 있지만 앞으로 많은 도시들이 사업에 참여하게 될 것이다. 그렇기 때문에 Hyperledger Fabric Parking System이 완성도 높게 만들어 진다면 정부의 ‘스마트시티 챌린지’ 사업에 기술로 채택될 수 있을 것이다. 문제는 Hyperledger Fabric Parking System을 정부가 채택한다면 주차장을 누가 운영하느냐에 따라 부가가치세 과세대상이 될 수도, 되지 않을 수도 있다는 것이다. 지방자치단체가 직접 또는 위탁 운영하는 경우, 즉 지방자치단체의 명의로 계산으로 하는 경우 주차요금은 부가가치세를 면제되겠지만 지방자치단체가 민간인에게 주차장부지를 임대하여 민간인으로 하여금 주차장업을 영위하게 하는 경우엔 주차료가 부가가치세 과세대상이 될 것이다. 그렇기 때문에 Hyperledger Fabric Parking System은 두 경우를 구분하여 주차 요금을 다르게 징수해야 할 것이다.

3.1.3 Economical issue

2007년 자동차 등록대 수는 1600만대 가량이며 2016년 자동차 등록대 수는 2150만대 가량이다. 자동차 등록수는 지금도 늘어나고 있으며 자동차 등록수가 급격하게 늘어남에 따라서 주차난 문제가 심각해졌다. 현재 존재하는 무인 주차 시스템은 즉각적인 대처가 어려울 뿐만 아니라 CCTV를 관리하는 인력 등 다른 인력도 필요한 상황이다. 뿐만 아니라 무인 주차 시스템의 차량 번호 자동 인식 시스템은 허점이 많아 차량 번호가 제대로 인식되지 않는 경우도 허다하다. 즉 진정한

의미의 무인 주차 시스템이 아직 구축되지 않은 것이며 현존하는 무인 주차 시스템으로는 주차난을 해결하기 힘들며 인력을 운용하는데 있어서 막대한 비용이 들기 때문에 시스템이 유용하지 않은것이 사실이다. Hyperledger Fabric Parking System은 이를 해결해야 할 것이다.

3.1.4 Social issue

매년 장애인 전용 주차 구역의 민원신고 수가 급증하고 있다. 장애인 전용 주차구역에 비장애인이 주차를 하여 민원신고가 들어오는 것이다. 장애인 전용 주차 구역 뿐만이 아니라 임산부 전용 주차 구역도 제대로 관리되고 있지 않다. 그러나 단속 자체도 워낙 쉽지 않아 신고 없이는 불법 주차 차량을 단속하기 힘들며 단속에 적발되었더라도 특별한 처벌 없이 넘어가는 경우가 대부분이다. 이는 현재 주차 관리 시스템의 한계로 볼 수 있다.

3.1.5 Technical issue

현재 대부분의 주차 관리 시스템은 중앙집중형 네트워크를 이용한다. 중앙 집중형 네트워크의 보안성 허점을 이용하여, 주차관리원들이 전산을 조작, 요금 처리를 취소한 후 요금을 가로채는 일이 일어나기도 했다. 이는 현재 주차 관리 시스템의 중앙 집중형 네트워크의 한계라고 볼 수 있다.

3.2 PEST 문제점 해결방안

Political issue 측면에서의 주차 요금을 다르게 징수해야하는 문제는 Hyperledger Fabric Parking System의 과금 시스템을 2가지로 나누면 해결 할 수 있다. Hyperledger Fabric Parking System은 주차장 관리인이 시스템을 신청하면 주차장 관리인이 월마다 요금을 내고 월 정액제로 시스템을 제공하는 방식을 채택하였는데, Hyperledger Fabric Parking System을 신청한 주차장이 부가가치세가 면제되는 주차장인지 면제되지 않는 주차장인지에 따라 다른 과금 시스템을 사용하면 해결할 수 있다. 단순히 요금중 일부를 세금으로 징수하면 되기 때문에 시스템 변경도 매우 쉽다.

Economical issue 측면에서의 인력을 감축하여 비용을 감소시켜야하는 문제는 Hyperledger Fabric Parking System 을 도입하면 자연스럽게 해결할 수 있을 것으로 보인다. Hyperledger Fabric Parking System 은 transaction을 모두 기록하며 기록이 되어야만 주차를 할 수 있기 때문에 차량 번호가 제대로 인식되지 않는 문제를 해결 할

수 있으며, 그렇기 때문에 CCTV를 관리하는 인력, 불법 주차된 차량이 있는지 확인해야 하는 인원 등 인원 감축을 할 수 있다. 그에 따라 자연히 비용도 감축 될 것이다.

Social issue 측면에서의 장애인 전용 주차 구역을 민간인이 불법으로 사용하는 문제 또한 해결 할 수 있다. Hyperledger Fabric Parking System에 차량을 등록하고 transaction을 기록해야만 Hyperledger Fabric Parking System을 사용할 수 있는데, 차량을 등록할때 장애인은 장애인 증명서, 장애인 등록증, 장애인 확인서 등을 등록하도록 하여 장애인임을 인증받은 차량만 장애인 주차 구역에 주차할 수 있게 하면 된다. 만약 비장애인이 장애인 주차 구역에 주차할 시 transaction이 아예 일어나지 않게 하여 주차를 할 수 없도록 만들 것이다. 이를 통해 문제를 해결 할 수 있다.

Technical issue 측면에서의 문제는 Hyperledger Fabric Parking System이 blockchain 기술을 바탕으로 하는 system이기 때문에 해결 할 수 있다. blockchain 기술은 분산화된 장부를 이용하기 때문에 admin의 권한으로라도 장부를 조작하여 요금 처리를 취소해 요금을 가로챌 수 없다. 뿐만 아니라 모든 피어들은 transaction의 기록을 볼 수 있기 때문에 장부는 투명하게 관리 된다. 무엇보다도 Hyperledger Fabric Parking System 은 무인 주차 관리 시스템이라서 주차 관리원들이 존재하지 않기 때문에 이러한 문제는 일어나지 않을 것이다.

4. 아이템 목적

4.1 3C 분석

4.1.1 3C 분석 - Customer

소비자 트렌드 분석 (트렌드 코리아 2019)

P	lay the Concept	: 컨셉을 연출하라
I	nvite to the ‘Cell Market’	: 세포마켓
G	oing New-tro	: 요즘 옛날, 뉴트로
G	reen Survival	: 필환경시대
Y	ou Are My Proxy Emotion	: 감정대리인, 내 마음을 부탁해
D	ata Intelligence	: 데이터 인텔리전스
R	ebirth of Space	: 공간의 재탄생, 카멜레온
E	merging ‘Millennial Family’	: 밀레니얼 가족

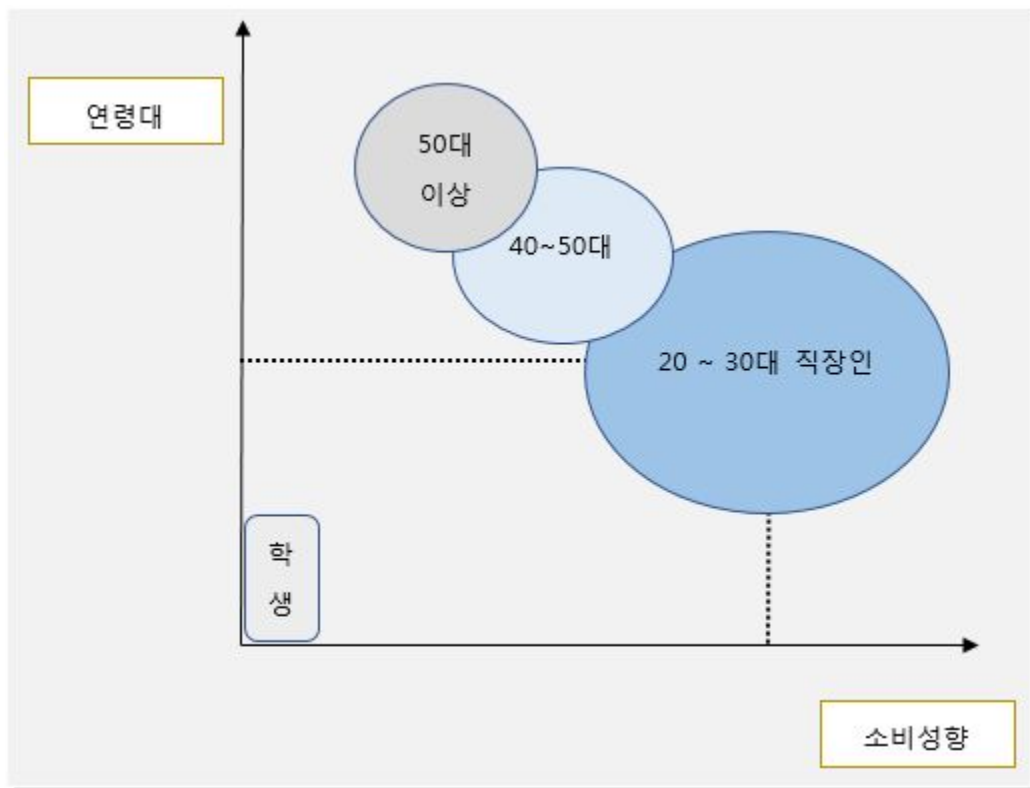
A s Being Myself : 그곳만이 내 세상, 나나랜드
M anners Maketh the Consumer : 매너 소비자

Cell market – 세포 마켓은 전문성을 바탕으로 자기 만족을 추구하면서 유급 생산 또는 유통을 개별적으로 수행하는 개인 마켓을 의미하는데 우리는 우리의 주차 시스템을 활용하여 어플리케이션 개발 등을 통하여 주차 활동을 우리 자사의 개인 마켓으로 사용할 수 있을 것이다.

Data Intelligence – 데이터 지능의 시대로 오면서 데이터는 정보로, 정보는 지식으로, 지식은 지혜로 한 단계씩 업그레이드된다고 할 수 있다. 우리의 자동 주차 시스템은 데이터 지능화 시대에 적절하게 사용될 수 있는 시스템이라고 할 수 있다.

Manner Maketh the Consumer – “No show”와 같은 매너의 결여로 인한 피해비용이 매우 크다. 우리의 시스템은 이러한 매너 없는 행위에 대해 대비책을 마련하여 사용자들과 이것에 관련된 사람들에게 피해가 생기지 않도록 해야 한다.

[목표 고객]



○ 시장 확대 가능성

<자동 주차 시스템은 특정 사용자를 겨냥한 상품이 아니므로 나이 별로 구분하였다.>

학생 : 학생들은 보통 자동차를 운전할 수 없기 때문에 소비성향이 거의 없다고 봐도 된다.

20 ~ 30대 : 주 타겟이라고 할 수 있다. 자동차를 운전하는 사람들이 매우 많으며 출/퇴근 및 각종 활동에 주차 활동이 일어난다.

40대 이상 : 나이가 먹을수록 자동차를 사용할 일이 줄어든다. 또한 자동시스템인 자동 주차 시스템이 나이 드신 일반 사용자 분들께 생소할 수 있다.

4.1.2 3C 분석 - Competitor

경쟁사 분석

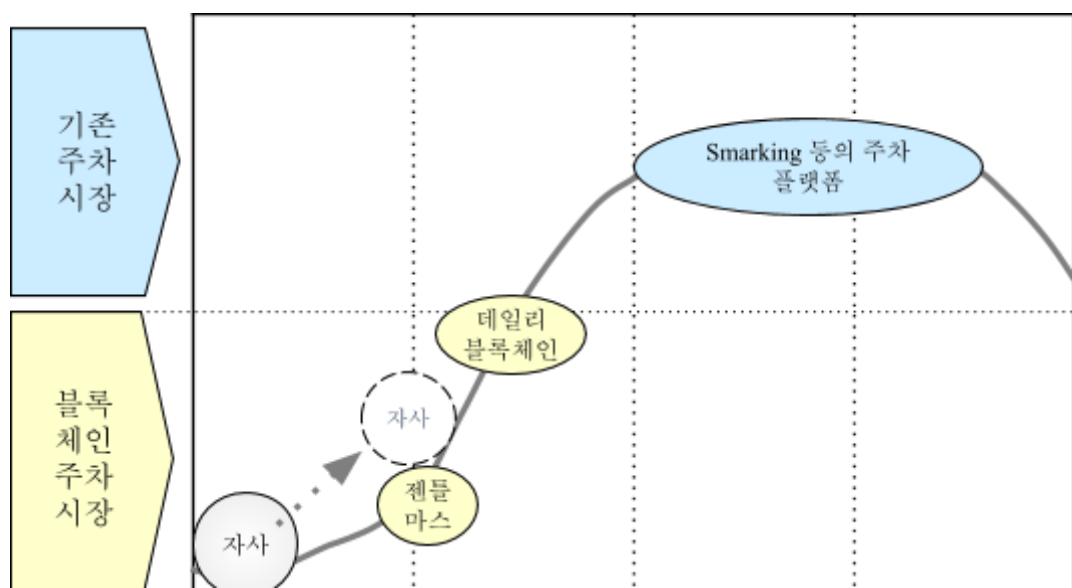
데일리 블록체인 – ‘스마트시티 챌린지’ 사업에 참여하고 있는 데일리블록체인은 블록체인 기반으로 주차장 관련 서비스를 제공하고 있으며 블록체인 주차 관련에서 가장 큰 경쟁력을 지닌 것으로 보인다. 데일리블록체인이 참여하고 있는 ‘스마트시티 챌린지’는 아직 선정된 지역에서만 시행되는 것으로 보인다.

젠틀 마스 – (사)대한주차산업협회와 블록체인 기반 스마트 3.0 주차 플랫폼 구축을 위한 ALL IN ONE 업무 협약(MOU)을 체결하였다. 젠트리온 암호 화폐로 주차비 결제를 할 수 있는 계기를 마련하였다. 하지만 계기를 마련했으나 아직 구체적인 실행은 보이지 않는다. 또한 자사와 비교했을 때 젠트리온이라는 암호 화폐를 사용함으로써 암호화폐를 사용하지 않고 하이퍼레저 패브릭을 사용하는 자사와 차별화된다.

Smarking INC – Smarking INC는 미국에서 Smarking 서비스를 하고 있으며 주차장에 관련된 정보를 이용자에게 제공한다. 현재 주차 플랫폼에서 큰 시장을 소유하고 있다. 하지만 우리나라에서는 서비스되지 않고 있으며 블록체인 기술을 사용하고 있지는 않는다. 만약 해외로 진출하게 될 경우 경쟁하게 될 회사로 예상된다.

4.1.3 3C 분석 - Company

산업 사이클 분석



- 시사점

- 젠틀 마스는 최근 주차 관련 협약을 맺음으로써 암호화폐 블록체인 주차 시장에 발을 들였다
- 데일리 블록체인은 스마트 시티 챌린지에 참여하면서 블록체인 주차 시장의 가장 큰 영향력을 지니고 있다.
- 자사는 적절한 홍보 및 프로젝트 사업을 통하여 인지도를 높이고 사용자들을 늘림으로써 블록 체인 주차 시장의 영향력을 늘려야 한다.

- 고객 제공 가치 (사용자 가치)

- 혼잡한 주차 환경의 축소
- 주차를 함으로써 소요되는 시간 절약
- 블록체인을 활용함으로써 확실한 결제 및 익명성을 보장

5. 사업화 전략

5.1 SWOT 분석을 통한 핵심 과제 도출

Strength - 강점	Weakness - 약점
<ul style="list-style-type: none">- 블록체인 기반 기술- 분산성/보안성- 비용적 우위	<ul style="list-style-type: none">- 블록체인 활용 주차 관리 시스템 홍보 및 인지도 상향 필요
Opportunities - 기회	Threats - 위협
<ul style="list-style-type: none">- 암호화폐 유행으로 인한 화제성- 2025년 전 세계 총생산의 10%가 블록체인 기술로 저장될 것 (세계경제포럼, WEF 보고서)	<ul style="list-style-type: none">- 블록체인 기술의 과대포장 가능성- Hyperledger와 이더리움 기술의 미숙함으로 인한 버그 가능성

도출된 전략 목표

- 기회 활용 전략 (SO 전략)
-> 블록체인 기술을 활용하여 서비스 비용을 줄이고, 보안성을 올린다.
- 위협 극복 전략 (ST 전략)
-> 보안성 강화 위주로 활용한다. 버그가 발견되는 경우 빠르게 피드백 한다.
- 약점 보완 전략 (WO 전략)
-> 암호화폐의 화제성을 이용하여 홍보한다.
- 역량 강화 전략 (WT 전략)
-> 블록체인 기술에 전문성을 가지면서도 대중적으로 인지도 있는 사업 파트너를 구한다.

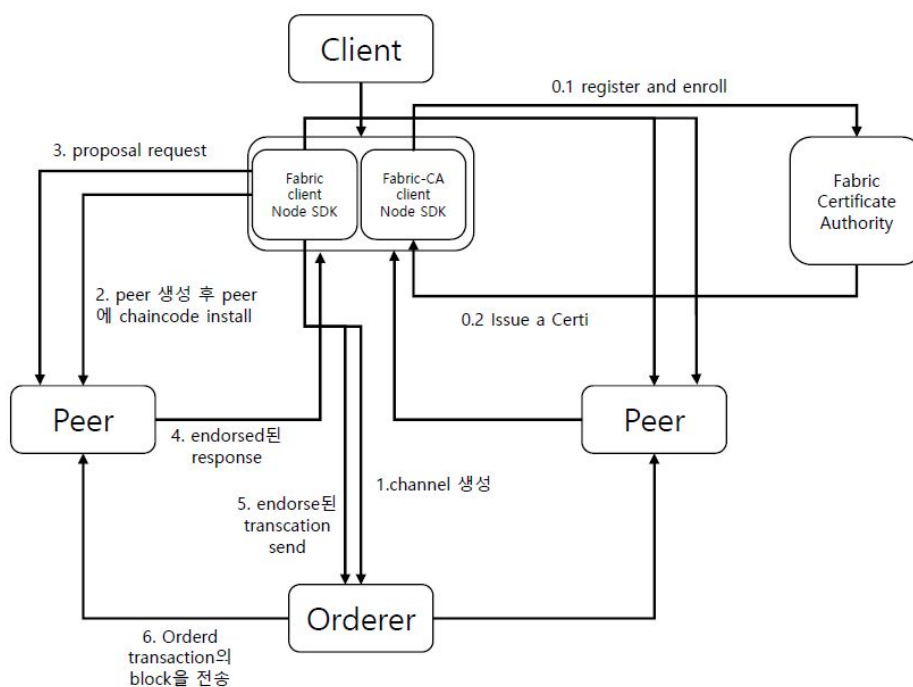
6. New Business Model

1.문제 자동차 등록수가 꾸준히 증가함에 따라 주차 공간의 부족 기존 주차 관리 시스템의 허점을 이용한 비리 발생	4.솔루션 블록 체인 플랫폼인 Hyperledger Fabric을 활용한 주차 관리 시스템 8.핵심구조 Admin, Client Parking Spot Parking Payment Parking reservation	3.가치제안 운전자의 정보를 등록 가능 분산성으로 인한 보안성 증대 Hyperledger Fabric을 활용한 경제성 증가	5.경쟁우위 블록 체인 기술을 활용해 주차 효율성과 보안성 증대 9.채널 유튜브 페이스북	2.고객군 주차 공간 부족으로 불편함을 느낀 자동차 이용자들
--	---	--	---	---

7.비용구조 서버 유지 비용 관리 시스템 설치 비용 마케팅 비용(유튜브와 페이스북을 주로 사용하기 때문에 비용이 적게 든다.)	6.수익원 정액제 개념의 시스템 가입/사용비
--	--

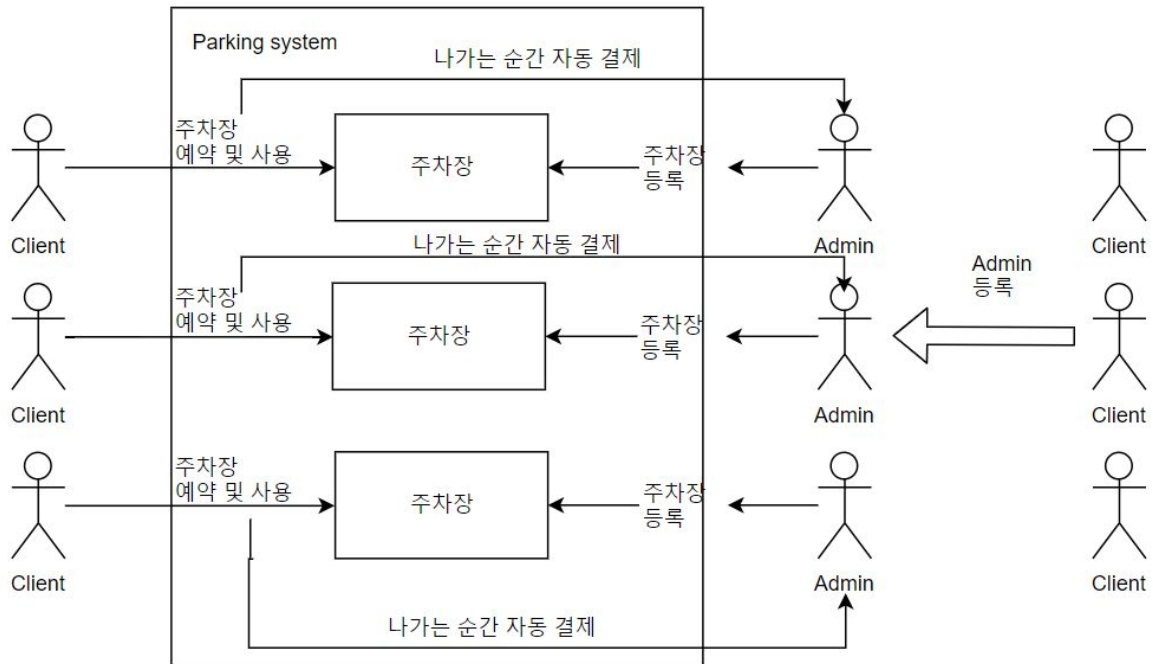
7. 프로젝트 설계 및 구현

7.1 시스템 구조



node.js를 통한 register로 admin과 user 간 MSP 형성, 채널 형성. 블록체인을 통한 장부는 모두가 다 갖고있다. 사용자가 transaction을 만들면 orderer는 endorsing peer를 통해 smart contract를 확인하고 Container에 저장한다. Fabric_client를 통해 채널을 생성하여 peer가 channel에 join하게 한다. 그 이후에 peer에 chaincode를 설치하고 application의 최신 state를 위해서 chaincode에 query를 보내 최신화 한다. 다양한 query를 보낼 수 있고 chaincode에 있는 함수를 통해 monitoring events를 주고 받을 수 있다.

7.2 Use Case Diagram



7.3 Functional Requirement

- Admin, client 등록
 - 사용자는 자신이 원하는 조건에 따라 admin과 user로 node.js를 통해 등록 할 수 있다.
- Parking spot 등록
 - Admin은 search parkingspot을 통해 경도와 위도를 기반으로 원하는 곳에 주차장을 등록 할 수 있다.
- Parking

- 사용자가 admin이 등록한 주차장에 원하고자 하는 시간에 사용을 할 수 있다.
- Payment
 - 사용자가 원하는 만큼 사용하고 나가면 admin이 설정해둔 WON/Min 만큼의 cost가 자동으로 결제 된다.
- Parking reservation
 - user가 원하는 parkingspot이 있다면 예약도 가능하다 만약 예약이 이미 되어 있다면 RESERVATION으로 표기된다.

7.4 프로젝트 구현

- ubuntu 16.04v
- Hyperledger fabric 1.4v
- node 8.16.0v
- npm 5.6.0v
- <https://github.com/suhasranganath/Hyperledger-Fabric-Tuna-App>
오픈소스를 기반으로 수정하여 웹 서비스 작성
- docker_compose.yml

```
services:
  ca.example.com:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca.example.com
      - FABRIC_CA_SERVER_CA_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem
      - FABRIC_CA_SERVER_CA_KEYFILE=/etc/hyperledger/fabric-ca-server-config/4239aa0dcd76daeeb8ba0cda701851d14504d31aad1b2dddbac6a57365e497c_sk
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ./crypto-config/peerOrganizations/org1.example.com/ca:/etc/hyperledger/fabric-ca-server-config
    container_name: ca.example.com
    networks:
      - basic
```

```

peer0.org1.example.com:
  container_name: peer0.org1.example.com
  image: hyperledger/fabric-peer
  environment:
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - CORE_PEER_ID=peer0.org1.example.com
    - CORE_LOGGING_PEER=debug
    - CORE_CHAINCODE_LOGGING_LEVEL=DEBUG
    - CORE_PEER_LOCALMSPID=Org1MSP
    - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/peer/
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_basic
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric
  command: peer node start
  ports:
    - 7051:7051
    - 7053:7053
  volumes:
    - /var/run:/host/var/run/
    - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/hyperledger/msp/peer
    - ./crypto-config/peerOrganizations/org1.example.com/users:/etc/hyperledger/msp/users
    - ./config:/etc/hyperledger/configtx
  depends_on:
    - orderer.example.com
    - couchdb
  networks:
    - basic

```

```

cli:
  container_name: cli
  image: hyperledger/fabric-tools
  tty: true
  environment:
    - GOPATH=/opt/gopath
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - CORE_LOGGING_LEVEL=DEBUG
    - CORE_PEER_ID=cli
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    - CORE_PEER_LOCALMSPID=Org1MSP
    - CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
    - CORE_CHAINCODE_KEEPALIVE=10
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
  command: /bin/bash
  volumes:
    - /var/run:/host/var/run/
    - ../../chaincode:/opt/gopath/src/github.com/
    - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
  networks:
    - basic

```

org1과 peer0 구현 주차장을 담당하는 org와 peer0을 통해 노드간 채널 형성

- basic-network/start.sh

```

# Create the channel
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp"
peer0.org1.example.com peer channel create -o orderer.example.com:7050 -c mychannel -f /etc/hyperledger/configtx/channel.tx
# Join peer0.org1.example.com to the channel.
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example.com/msp"
peer0.org1.example.com peer channel join -b mychannel.block

```

Org1MSP를 통해 채널을 생성하고 peer0을 가입시킨다.

- chaincode

```
// Define the Smart Contract structure
type SmartContract struct {
    ParkingspotService    parkingservice.ParkingspotService
    ParkingTimeService    parkingservice.ParkingTimeService
    UserService           parkingservice.UserService
    ParkingCommonService  parkingservice.ParkingCommonService
    GeoHashService        parkingservice.GeoHashService
}
```

이와 같은 함수들을 통해 장소 서비스, 시간 서비스, 유저 서비스, 지도 서비스를 지원

- registerAdmin.js, registUser.js

```
20 var fabric_client = new Fabric_Client();
21 var fabric_ca_client = null;
22 var admin_user = null;
23 var member_user = null;
24 var store_path = path.join(os.homedir(), '.hfc-key-store');
25 console.log(' Store path:'+store_path);
```

key-store를 통해 신원 확인

```
44 fabric_ca_client = new Fabric_CA_Client('http://localhost:7054', null, '', crypto_suite);
```

Fabric_CA_Client

```
ca.example.com:
  image: hyperledger/fabric-ca
  environment:
    - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
    - FABRIC_CA_SERVER_CA_NAME=ca.example.com
    - FABRIC_CA_SERVER_CA_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem
    - FABRIC_CA_SERVER_CA_KEYFILE=/etc/hyperledger/fabric-ca-server-config/4239aa0dcd76daeeb8ba0cda701851d14504d31aad1b2dddbac6a57365e497c_sk
  ports:
    - "7054:7054"
  command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
  volumes:
    - ./crypto-config/peerOrganizations/org1.example.com/ca:/etc/hyperledger/fabric-ca-server-config
  container_name: ca.example.com
  networks:
    - basic
```

docker-compose.yml의 ca와 연결된 포트를 통해 가져오는 것 확인

```
49 if (user_from_store && user_from_store.isEnrolled()) {
50     console.log('Successfully loaded admin from persistence');
51     admin_user = user_from_store;
52 } else {
53     throw new Error('Failed to get admin.... run registerAdmin.js');
54 }
55
56 // at this point we should have the admin user
57 // first need to register the user with the CA server
58 return fabric_ca_client.register({enrollmentID: 'user1', affiliation: 'org1.department1'}, admin_user);
59 }).then((secret) => {
60     // next we need to enroll the user with CA server
61     console.log('Successfully registered user1 - secret:'+ secret);
```

registerAdmin 실행 후 registerUser를 실행하면 등록이 완료됨.

```

1  {
2    "name": "parking-app",
3    "version": "1.0.0",
4    "description": "Hyperledger Fabric MOOC Sample Application",
5    "main": "server.js",
6    "scripts": {
7      "start": "PORT=8001 node server.js",
8      "start-dev": "PORT=8001 node server.js & browser-sync start --port 18000 --no-open --proxy 'localhost:8001' --serveStatic './' --files './'",
9    },

```

package.json에 start-dev를 통해 8001번 포트, 18000번 포트를 통해 서버가 실행되도록 설정해 둬 npm run start-dev를 통해 서버 실행 가능

8. 구현 결과

Key	ID	Name
1	PARK	
2	KIM	
3	YANG	
4	HOON	
5	YEON	

Buttons: Admin parkingspots, Admin parkingtimes

Save

Update user: -

Id:

Name:

Get all users

localhost:18000, 8001 포트로도 접속이 가능하다. init chaincode로 3명을 넣어 두었고 2명을 새로 추가해 주었다. 하지만 user1로 등록했기 때문에 Id가 1인 PARK owner로만 수정이 가능하다.

Hyperledger Fabric Application

35.221.86.176:8001/#/parkingspots/user/1

Prototype Admin users Admin parkingspots Search parkingspot About

ID	Name	Cost per min	Owner
1	PARK's parkingspot	400 WON	PARK
2	AJOU	300 WON	PARK
3	WONCHUN	500 WON	PARK
4	PALDAL	200 WON	PARK
5	YULGOK	300 WON	PARK
xxx	xxx	10 WON	PARK

Update parkingspot: -

Id:

Name:

Location.x:

Location.y:

Cost per min:

Save

User parkingspots

Admin parkingspot이다. 여기서는 Admin이 주차장 등록을 할 수 있다. 해당 주차장 id와 가격, 위치를 지정할 수 있다.

Hyperledger Fabric Application

35.221.86.176:8001/#/parkingtimes/user/1

Prototype Admin users Admin parkingspots Search parkingspot About

ID	Name	Cost per min	Owner
1	PARK's parkingspot	400 WON	PARK
2	AJOU	300 WON	PARK
3	WONCHUN	500 WON	PARK
4	PALDAL	200 WON	PARK
5	YULGOK	300 WON	PARK
xxx	xxx	10 WON	PARK

Update parkingtime: -

Id:

ParkingStart:

ParkingEnd:

CostPerMinute:

ParkingspotId:

new open time new parking new reservation

Get all parkingspots

ID	Type	Start	End	WON/min	Cost	Parkingspot	ParkingspotId
1	RESERVATION	09.06.2019 02:20:33 +0900	09.06.2019 03:20:33 +0900	3	0	WONCHUN	2
2	PARKING	09.06.2019 02:19:02 +0900	09.06.2019 03:19:02 +0900	12	0	AJOU	2

해당 주차장에 주차가 되어있는지 혹은 예약이 되어있는지도 확인이 가능하다.

The screenshot shows a web application titled "Hyperledger Fabric Application" with a navigation bar containing "Prototype", "Admin users", "Admin parkingspots", "Search parkingspot", and "About". The main interface is divided into two sections: a search form on the left and an update form on the right, with a map on the far right.

Search Form:

- X: 37.28115563160916
- Y: 127.04542314986614
- Zoom: 14
- Start: 2019-06-09T02:57:24.659+09:00
- End: 2019-06-09T03:57:24.660+09:00
- Search button

Update parkingtime:-

- id: Ex: 11
- ParkingStart: Ex: 2017-12-11T14:09:38.7378781Z
- ParkingEnd: Ex: 2017-12-11T14:09:38.7378781Z
- CostPerMinute: Ex: 10
- Parkingspot.id: Ex: 2
- Parkingspot.name: Ex: 2
- Create new parking slot button
- Create new parking reservation button

Map: A map showing a location in Seoul, Korea, with a blue pin indicating the selected parking spot.

Table:

ID	Type	Start	End	WON/min	Cost	Parkingspot
1	RESERVATION	09.06.2019 02:20:33 +0900	09.06.2019 03:20:33 +0900	3	0	WONCHUN (PARK)
2	PARKING	09.06.2019 02:19:02 +0900	09.06.2019 03:19:02 +0900	12	0	AJOU (PARK)
3	RESERVATION	09.06.2019 02:20:33 +0900	09.06.2019 03:20:33 +0900	3	0	WONCHUN (PARK)
4	RESERVATION	09.06.2019 02:20:33 +0900	09.06.2019 03:20:33 +0900	3	0	PALDAL (PARK)

search parkingspot에서는 위도와 경도를 입력하면 인근 주차장을 찾아준다. 사용이 가능하면 FREE라고 표시되고 주차중이거나 예약이 되었다면 위의 화면과 같이 이미 사용중인 것을 알 수 있다. update parkingtime을 통해 create new parking spot은 주차를 하는, create new parking reservation은 예약을 하는 함수가 실행된다.

9. 역할 분담

이름	역할
박용채	개발(tuna_app 기반 웹, hyperledger, node.js, server.js 등), 데모 촬영 및 편집
김훈민	보고서(기술 사용 현황, PEST 분석) 및 PPT 작성, 발표영상 촬영 및 편집 유튜브 업로드
김용민	보고서(동기 및 기대효과, 사업화 전략 SWOT 분석 등) 작성 및 PPT 작성
양준모	개발 환경 초기 세팅 및 보고서(3c 분석, 개발 일정 등) 작성

10. 개발 일정

개발일정표

개발구분	세부항목	4	5				6	
		4w	1w	2w	3w	4w	1w	2w
분석 설계	프로젝트 아이디어 구상							
	프로젝트 프로포절							
개발	hyperledger 서버 개발							
	차량 시스템 개발							
	자동 결제 시스템 개발							
	차량 시스템과 결제 시스템 융합							
	ui 개발							
	주차 시스템 서버 연동							
테스트	서버 테스트							
	툴 테스트							
	통합 테스트							
마무리	ppt 및 보고서 작성							
	발표							

11. 참고 문헌

- The Blockchain as a Software Connector -2016 13th Working IEEE/IFIP Conference on Software Architecture -
- Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab -Kevin Delmolino^{1(B)}, Mitchell Arnett¹, Ahmed Kosba¹, Andrew Miller^{1,2}, and Elaine Shi² ¹ Department of Computer Science, University of Maryland, College Park, College Park, USA del@terpmail.umd.edu, marnett@umd.edu, {akosba, amiller}@cs.umd.edu ² Initiative for Cryptocurrencies and

Contracts (IC3), Department of Computer Science, Cornell University,
Ithaca, USA -

- Sven Mitt Blockchain Application - Case Study on Hy-perledger Fabric
Master's Thesis -UNIVERSITY OF TARTU Institute of Computer
Science Software Engineering Curriculum-

12. 발표 / 데모 영상

데모 영상 : <https://youtu.be/PXsJj81vIZ0>

발표 part 1 영상 : <https://youtu.be/4FbdwOz2LYE>

발표 part 2 영상 : <https://youtu.be/FHspRSAkS2w>