

## 주간 연구보고서

작성자	김한호	작성일자	2020.04.12.
연구기간	2020.04.06 ~ 2020.04.12		
1. 주간 연구 요약			
<div>hyperledger.org 예제 공부</div> <div><div><div>• fabcar</div><div>createCar: 차 정보를 만들어낸다.</div><div>queryAllCars: 모든 차의 정보를 가져온다.</div><div>quaryCarProperties: 특정차의 정보를 가져온다.</div><div>updateCarColor: 차 색깔을 업데이트한다.</div><div>updateCarOwner: 차 소유자를 업데이트한다.</div></div><div><div>• commercial paper</div><div>issue: 기업어음 발행</div><div>buy: 기업어음 구입</div><div>redeem: 기업 어음 상환</div></div></div>			
2. 주간 연구 상세			
<div>VirtualBox에 설치한 우분투에 hyperleger fabric 안에 fabric-samples 안에 fabcar commercial paper를 교대로 실행했습니다. fabcar 폴더 밑에 startFabric.sh javascript로 설치 스크립트를 실행했습니다. docker를 설정하고 채널을 만들고 체인코드를 생성하고 후에 javascript폴더로 옮겨가서 npm install을 하고 node enrollAdmin node registerUser를 통해서 등록했습니다. node invoke를 통해서 트랜잭션을 확인하고 node query를 통해서 모든 차의 정보를 확인했습니다. 후에 basic-netwok의 ./stop.sh를 사용하여 fabcar 예제를 삭제했습니다.</div> <div>다시 basic-network폴더로 이용해서 start.sh 스크립트를 사용해서 도커(docker)에서 peer, orderer, couchdb, ca 네개의 네트워크를 생성합니다. 새로운 명령어 창을 열어서 commercial-paper/organization/magnetocorp/configuration/cli/로 이동합니다. monitordocker.sh 통해서 net_basic의 모든 컨테이너 모니터링을 실행합니다. commercial-paper/organization/magnetocorp/configuration/cli로 이동해서 docker exec cliMagnetocorp peer chaincode install -n</div>			

papercontract -v 0 -p /opt/gopath/src/github.com/contract -l node 체인 코드를 설치합니다. 다음으로 피어설치한 체인코드를 인스턴스화 합니다. docker exec cliMagnetCorp peer chaincode instantiate -n papercontract -v 0 -l node -c '{"Args":["org.papernet.commercialpaper:instantiate"]}' -C mychannel -P"AND ('Org1MSP.member')"

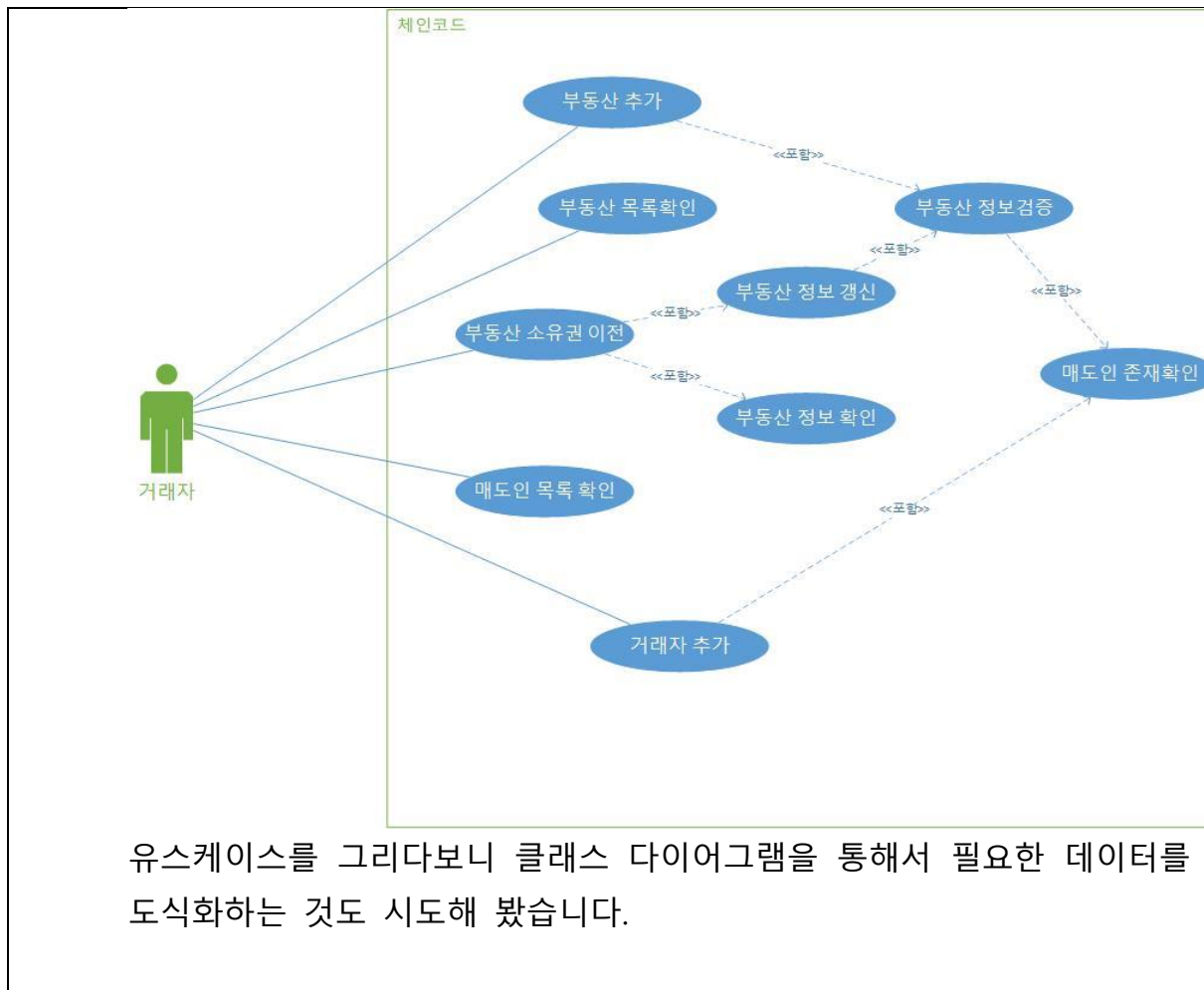
commcial-paper/organization/magnetocorp/application/에서 npm install node addToWallet.js를 합니다. node issue.js로 기업어음을 발행합니다.

commercial-paper/organization/digiback/configuration/cli/로 이동합니다. docker-compose -f docker-compose.yml up -d cliDigiBack를 통해서 cliDigiBank 컨테이너를 실행합니다. commercial-paper/organization/digibank/application 에서 npm install을 합니다. 그 대로 node addToWallet.js로 인증서 폴더에서 인증서를 옮깁니다. node buy.js로 기업 어음을 구입합니다. node redeem.js로 어음을 상환합니다.

### 3. 예제 수정을 통한 부동산 관련 블록체인 개발

처음에 연구 주제 방향을 잡고서 fabcar와 commercial-paper를 추천하 셴을까 고민했습니다. 하나는 자동차 매물을 통해서 소유자를 바꾸는 것 이고, 기업 어음과 관련된 예제는 상환이 될 때까지 구입되어 있어서 은 행이 가지고 있는 상태가 존재하는 예제였습니다. 두 가지 예제와 부동 산을 생각해봤을 때 한가지 예제 만으로는 부족하다는 생각이 들었습니 다. 처음에 예제를 실행하고 이해를 하면서 fabcar에 commercial-paper 를 병합해서 구현을 하는 방향으로 생각을 했습니다. 실행시키면서 값을 넣어보고 있는 값을 변형시키는 것은 간단한 일이었지만 막상 두 예제 를 병합하면서 쉽지 않았습니다.

생각을 너무 쉽게 변형을 하려던 것이 아닌가 생각이 들어서 설계부터 해야 하는 것이 아닌가 방향을 전환하기로 했습니다. fabric-samples에 있는 basic-network부터 자세히 공부를 하며 구현을 해보려 합니다. 처 음에는 부동산에 관한 유스케이스부터 그려보기로 했습니다.



### Participant

Id: String  
Name: String  
Token: int  
Muti Estate: int

### Real Estate

Id: String  
Type: String  
OwnerList: [["owner id",  
"equity percent"]]  
Transaction Amount: int  
TimeStamp: time.Time  
IsTrading: boolean  
Name: String  
Size: String

### Trading Ledger

Ledgerkey: String  
Idx: int  
SellerList: [["seller id",  
"equity payment"]]  
BuyerList: [["buyer id",  
"equity payment"]]  
Init Payment: int  
Mid Payment: int  
Last Payment: int  
IsNextStep: boolean  
State: String(Ready|  
contract|implement)

이번에 설계를 하면서 생긴 의문점은 기존에 설계를 통해서 체인코드를 구현하는 것은 의미 있는 일인지 생각을 해보게 되었습니다. 기존에 관계형 데이터베이스 형식으로 액티비티 다이어그램까지 상세설계를 하는 것이 바람직한 일인지 고민하게 되었습니다. 리스트 형태로 키-벨류 사용자 리스트를 통해서 구현을 하게 될 때에 장단점도 있을지 고민을 하고 있지만 그 부분에 관해서는 관계형 데이터베이스만 생각을 제한하는 것 같아서 아직 결정을 하지 못했습니다.

#### 4. 연구 진행시 생긴 의문점 혹은 아이디어

현실 부동산 거래에서는 공동명의로 같은 소유자와 판매하는 사람이 한 사람이 아닌 경우도 있는데 복잡한 상황을 모사하는 것이 의미가 있을지 생각을 해보게 되었습니다. 만약 복잡한 소유관계도 넣어서 구현을 하게 된다면 부동산 증권화와 같은 거래 관계도 생각해 보는 것도 괜찮은 방향일 수도 있겠다는 생각이 들었습니다. 어느 수준의 부동산 데이터를 가져올지도 생각을 해보고 있습니다. 등기부 등본 상에 있는 모든 부동산 표제부와 이력을 넣어서 구현을 하는 것이 좋을지 단순한 거래

프로그램에서 블록체인의 장점을 강조하는 것이 나을지 생각해보고 있습니다.