

2023 전기 졸업과제 착수보고서

클라우드 기반 크레딧 거래 플랫폼

클라우드 나인

2023

목차

[1. 과제 배경 및 목표 2](#_Toc135318230)

[1-1 개발 배경 및 필요성 2](#_Toc135318231)

[1-2 개발 목표 및 방향 3](#_Toc135318232)

[2. 요구사항 분석 4](#_Toc135318233)

[2-1 구현 내용 4](#_Toc135318234)

[2-2 현실적 제약사항 및 대책 4](#_Toc135318235)

[3. 시스템 설계 5](#_Toc135318238)

[3-1 전체 시스템의 문맥도 5](#_Toc135318239)

[3-2 서비스별 개요도와 Usecase 6](#_Toc135318242)

[3-3 배포 파이프라인 15](#_Toc135318245)

[4. 개발 환경 및 기술 스택 16](#_Toc135318246)

[5. 개발 일정 및 역할 분배 17](#_Toc135318247)

[5-1 개발 일정 17](#_Toc135318248)

[5-2 역할 분배 18](#_Toc135318249)

# 과제 배경 및 목표

## 1-1 개발 배경 및 필요성

2012년 이후 스마트폰 보급률 확산, 무선 통신 인프라와 전자 상거래의 발전으로 모바일 결제 수단이 세계적으로 발전하는 추세이다. 한국 역시 2015년 이후 모바일 인증서 규제 완화를 통해 모바일 결제 서비스가 급속히 성장하고 있고 2022년도 기준 전체 결제의 59%가 모바일 결제로 이루어지고 있다. 그러나 우리나라에서 행해지는 전자 결제 방식들은 대금의 수취인, 수령인 및 거래 대상, 금액의 정보만을 가지고 금전과 재화 및 서비스의 교환이 이루어진다.

우리는 이 결제라는 행위에서 단순히 금전이 오가는 것만이 아니라 부가 이득을 창출할 수 있도록 확장성이 유연한 독자적인 결제 플랫폼을 구축하고자 한다. 예를 들면, 결제 정보로부터 어떤 소비자층이 주로 어떤 재화, 서비스를 구매하는지, 어느 정도의 가격대에서 자주 결제가 이루어지는지 등의 데이터를 수집하는 알고리즘에서부터, 데이터를 바탕으로 사용자 맞춤형 정보를 제공하거나, 사용자 결제 데이터를 바탕으로 사용자의 결제내역, 신용카드 연체 여부, 각종 요금 납부 상황, 모바일 결제 내역, 재테크 상품 가입 현황을 제공하는 등 부가적인 기능을 포함한 개별 시스템들의 탈 부착 즉, 사용자의 입맛대로 커스터마이징 하기 편한 플랫폼을 구축한다.

시스템의 확장성 확보를 용이하게 하기 위해 마이크로 서비스의 형태로 구현하되 클라우드 컴퓨팅을 활용한다. 마이크로 서비스란 소프트웨어 아키텍처 패턴 중 하나로, 복잡한 애플리케이션을 작은 독립적인 서비스 단위로 분할하여 구성하는 것이다. 각각의 마이크로 서비스는 자체적으로 독립적인 업무를 수행하며, 필요에 따라 서로 통신하여 전체적인 시스템을 완성하는 아키텍처 패턴이다. 이 독립적인 서비스들을 하나의 모듈로 간주하고 본 시스템의 핵심적인 모듈에 부차적으로 다른 모듈들을 덧붙여 기능을 추가하는 방식으로 커스터마이징을 하듯이 시스템을 구성할 수 있게끔 만들어 확장성을 확보할 수 있게 된다.

클라우드 컴퓨팅(Cloud Computing)의 반대되는 개념은 온프레미스(On-premises) 방식인데, 본 마이크로 서비스 형태의 서비스를 구현하면서 온프레미스 방식보다 클라우드 컴퓨팅 방식을 도입하는 것에 유리한 점이 있다. 온프레미스 방식은 컴퓨터 시스템이나 소프트웨어를 사용자 자신이 보유하고 운영하는 방식을 말한다. 즉, 사용자는 해당 시스템이나 소프트웨어를 직접 구매 및 설치하고, 서버를 직접 관리하고 유지보수를 수행해야 한다. 그러므로 인프라를 확장하기 위해서는 서버를 추가로 구매하고 설치해야 하므로 확장성이 부족하고 더 나아가 보안 측면에서도 사용자가 직접 보안에 대한 책임을 지고 대비해야 하므로 노하우가 부족하다면 보안 취약점이 발생할 수 있다.

또한 유지보수에서도 인력이 많이 들 수 있는 등 부가적인 단점이 따른다. 그에 반해 클라우드 컴퓨팅은 인터넷을 통해 컴퓨팅 자원을 제공하는 기술을 말한다. 즉, 사용자가 인터넷을 통해 필요한 컴퓨팅 자원을 빌려서 사용하는 것이다. 이를 통해 사용자가 직접 인프라나 소프트웨어를 보유하지 않아도 되며, 필요한 만큼 컴퓨팅 자원을 확장하거나 축소할 수 있다. 이를 통해 사용자는 비용을 절감하면서도 필요에 따라 유연하게 자원을 사용할 수 있다. 또한 보안 측면에서도 클라우드 제공업체가 보안에 대한 책임을 지고 있으므로 보안성이 보장된다. 데이터 백업과 같은 보안 기능을 제공하므로 데이터 손실을 예방할 수 있다.

즉 온프레미스 방식을 택하는 것보다 클라우드 컴퓨팅을 도입하면 자원 확장 및 축소, 보안에 대한 부담, 지리적 제약으로부터의 자유도에 있어서 장점을 가지며, 무엇보다도 중요한 점은 자원 확장 및 축소에 용이하다는 점이 마이크로 서비스의 작은 독립적인 서비스들을 모듈처럼 탈 부착하여 하나의 큰 시스템을 이룬다는 특징과 궁합이 잘 맞아 클라우드 컴퓨팅을 도입하여 구현한다.

본 플랫폼을 개발함으로써 단순히 결제만 하는 것이 아니라, 결제라는 행위에 따른 다른 부가적인 이득을 창출할 가능성을 열어 두고, 확보한 확장성을 바탕으로 다양한 서비스 방식을 실현한다. 이런 특장점을 들어 개발 필요성을 제고하는 바이다.

## 1-2 개발 목표 및 방향

본 졸업과제는 클라우드 기반 크레딧 거래 플랫폼을 개발하는 것을 목표로 한다. 거래 방식의 확장성을 위해 인증 서비스, 크레딧 서비스를 중심으로 한 마이크로서비스를 적용한다. 각각의 서비스는 컨테이너 환경에서 동작하며, 운영 및 확장을 위해 컨테이너 오케스트레이션 기술인 쿠버네티스를 사용하여 배포한다. 서비스의 보안을 위해 비밀번호의 저장은 Bcrypt, 클라이언트의 로그인은 RSA 비대칭 키 암호화 방식, 클라이언트와 서버 간 통신에는 JWT를 사용한다.

# 2. 요구사항 분석

## 2-1 구현 내용

본 과제의 목표인 ‘클라우드 환경의 크레딧 거래 플랫폼’을 구현하기 위해 가게의 키오스크 QR 결제를 예시 서비스로 하여 시스템을 구현하되, 마이크로서비스 아키텍처를 적용하여 다른 거래 방식을 적용할 수 있는 확장성을 부여한다. 그리고 모든 서비스는 자유로운 scale-out을 위해 무상태로 구현되어야 한다. 사용자는 웹서비스를 이용하여 크레딧을 충전하며, 모바일 앱을 통해 QR을 생성하여 인증을 통해 크레딧 거래가 가능하다. 또한 키오스크는 웹 앱으로 구현하고 QR 리더기를 통해 QR 정보를 획득할 수 있어야 한다. 시스템 전체에 대한 요구사항을 정리하면 아래와 같다.

* 플랫폼은 마이크로서비스 형태로 클라우드 환경에 구현되어 서비스의 확장이 가능해야 한다.
* 각 서비스는 무상태로 운영되어 자유로운 scale-out이 가능해야 한다.
* 사용자는 웹을 통해 크레딧의 충전이 가능하며, 모바일 앱을 통해 QR 인증 및 결제가 가능해야 한다.
* 키오스크는 웹 앱 형태로 구현하고 QR 리더기를 통해 QR 정보를 획득 및 전송할 수 있어야 한다.

## 2-2 현실적 제약사항 및 대책

### 상용 클라우드 서비스 요금

프로젝트는 상용 클라우드 서비스 AWS에 컨테이너 형태로 배포를 진행한다. 클러스터는 AWS EC2 t3.small 이상의 리소스를 가진 인스턴스 3개로 구성되며, Jenkins 빌드 서버를 포함하여 총 4개의 EC2 인스턴스가 필요하다. 그 외 다른 AWS 서비스인 RDS, S3, ELB, Route53을 사용하여, 한 달에 최소 5만원의 요금이 발생할 것으로 예상된다. 이에 AWS에 클러스터를 구성하는 것은 최대한 늦추고, 개발용 온프레미스 환경 쿠버네티스 클러스터를 구성하여 개발 및 테스트를 진행할 것이다. 자동화된 CI/CD 파이프라인이 잘 구축된다면 클라우드 환경 클러스터로의 전환도 쉽게 할 수 있을 것으로 예상된다.

### 실제 결제의 처리

프로젝트 중 크레딧 충전 시 웹페이지에서 결제가 이루어져야 하는 부분이 있다. 그러나 졸업 과제에서 실제 결제를 진행하는 것은 문제가 될 수 있으므로 전자지급결제대행(PG) 연동 서비스에서 지원하는 테스트 결제를 사용하여 결제를 진행한다.

# 3. 시스템 설계

## 3-1 전체 시스템의 문맥도

도표, 텍스트, 원, 라인이(가) 표시된 사진

자동 생성된 설명

그림 1 전체 시스템 문맥도

전체 시스템은 3가지의 클라이언트와 4가지의 서비스로 구성되어 있다.

### 클라이언트

1. 웹 사이트  
   웹 사이트는 사용자의 회원가입과 크레딧 충전, 키오스크 서비스 이용자의 가게 등록 및 상품 추가를 담당한다.
2. 사용자 앱  
   사용자 앱은 QR을 생성하여 결제 인증을 담당한다.
3. 키오스크 앱  
   키오스크 앱은 상품의 판매 및 결제를 담당한다.

### 서비스

1. 인증 서비스  
   크레딧 거래 플랫폼의 핵심 서비스로 플랫폼 이용자의 식별과 서비스 간 인증번호의 생성과 확인을 담당한다.
2. 크레딧 서비스  
   플랫폼 이용자의 크레딧을 관리하는 서비스로 계좌의 생성, 크레딧 충전 및 거래를 담당한다.
3. 사용자 서비스  
   일반 사용자를 생성 및 관리하며 크레딧의 충전을 요청하거나 결제를 위한 QR 정보를 생성하는 등 사용자와 다른 서비스 간 중간자 역할을 한다.
4. 가게 서비스  
   상품을 판매하고자 하는 가게를 등록 및 관리하며 키오스크를 통한 결제 처리를 담당한다.

## 3-2 서비스별 개요도와 Usecase

### 컨텍스트 다이어그램

#### 인증 서비스

텍스트, 도표, 평면도, 라인이(가) 표시된 사진

자동 생성된 설명

그림 2 인증 서비스 문맥도

1. 개체식별번호 발급
   1. 외부 서비스가 인증번호 확인 응답을 받을 엔드 포인트를 전달하며 개체식별번호 발급을 요청한다.
   2. 인증 서비스에서 개체식별번호를 생성하고 등록한다.
   3. 외부 서비스로 개체식별번호를 응답한다.
2. 인증번호 생성

2.1 외부 서비스가 인증 요청자의 개체의 식별번호와 함께 인증번호 생성을 요청한다.

2.2 인증 서비스에서 인증번호를 생성한다.

2.3 인증번호와 요청자의 개체식별번호를 저장한다.

2.4 인증번호를 응답한다.

1. 인증번호 확인

3.1 인증번호를 획득한 외부 서비스에서 획득한 인증번호와 인식자의 개체식별번호를 가지고 인증 서비스에 인증을 요청한다.

3.2 인증 서비스에서 인증번호를 바탕으로 인증번호 생성자의 개체식별번호를 탐색한다.

3.3 인증번호 생성자와 인증번호 인식자에게 저장된 인증번호 확인 응답 엔드 포인트로 생성자의 개체식별번호와 인식자의 개체식별번호를 응답한다.

#### 크레딧 서비스

텍스트, 도표, 직사각형, 평면도이(가) 표시된 사진

자동 생성된 설명

그림 3 크레딧 서비스 문맥도

1. 계좌 생성
   1. 개체식별번호를 받아서 계좌를 생성한다.
   2. 계좌 생성이 완료되었음을 응답한다.
2. 크레딧 충전

2.1 외부 서비스에서 개체식별번호와 충전 정보를 획득한다.

2.2 결제 시스템을 통해 결제한다.

2.3 이용자의 크레딧 양을 업데이트한다.

2.4 충전 완료를 응답한다.

3. 송금

3.1 다른 서비스에서 송금자의 개체식별번호와 수령자의 개체식별번호, 그리고 송금 정보를 획득한다.

3.2 송금자의 계좌에서 수령자의 계좌로 크레딧을 이동시킨다.

#### 사용자 서비스

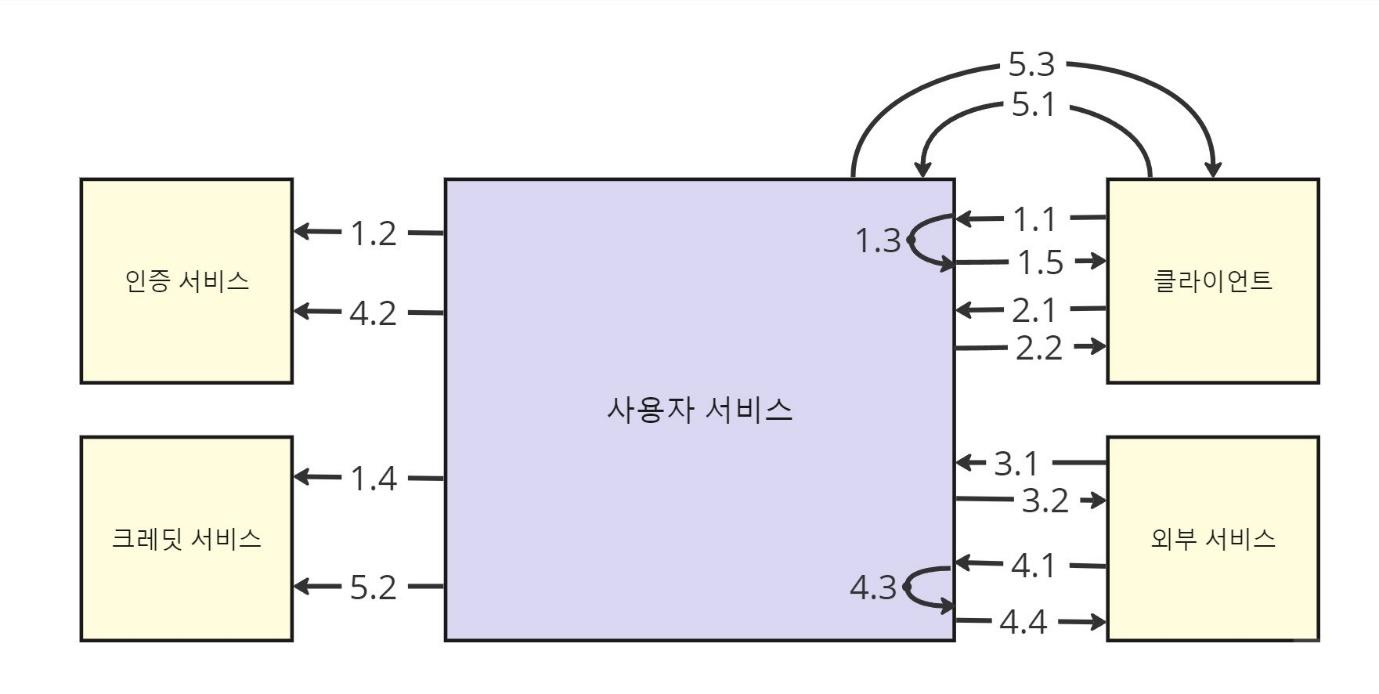


그림 4 사용자 서비스 문맥도

1. 사용자 회원가입

1.1 사용자 정보, 아이디, 비밀번호를 클라이언트로부터 받는다.

1.2 인증 서비스로부터 개체식별번호를 발급받는다.

1.3 사용자 정보, 아이디, 비밀번호, 개체식별번호를 저장한다.

1.4 개체식별번호를 크레딧 서비스로 전달하여 계좌를 생성한다.

1.5 클라이언트에게 회원가입 완료를 응답한다.

1. 사용자 로그인

2.1 클라이언트로부터 아이디와 비밀번호를 받는다.

2.2 인증 토큰을 응답한다.

1. 사용자 개체식별번호 획득

3.1 다른 서비스로부터 사용자의 인증 토큰을 받는다.

3.2 해당 사용자의 개체식별번호를 응답한다.

1. QR 생성 요청

4.1 클라이언트로부터 사용자 인증 토큰을 받는다.

4.2 인증 서비스로 사용자 개체식별번호를 전달하면서 인증번호 생성을 요청한다.

4.3 인증번호와 함께 QR 정보를 생성하고 기록한다.

4.4 클라이언트로 QR 정보를 응답한다.

1. 크레딧 충전 요청

5.1 클라이언트에서 사용자 인증 토큰과 충전 정보를 받는다.

5.2 크레딧 서비스에 개체식별번호와 충전 정보를 전달한다.

5.3 클라이언트로 크레딧 충전 완료를 응답한다.

#### 가게 서비스

텍스트, 도표, 스크린샷, 평면도이(가) 표시된 사진

자동 생성된 설명

그림 5 가게 서비스 문맥도

1. 가게 등록
   1. 사용자 인증 토큰과 가게 이름을 클라이언트로부터 받는다.
   2. 사용자 인증 토큰을 사용자 서비스에 전달하여 사용자의 개체식별번호를 획득한다.
   3. 인증 서비스에 개체식별번호 발급을 요청하여 가게의 개체식별번호를 발급받는다.
   4. 사용자의 개체식별번호와 가게의 개체식별번호를 함께 저장한다.
   5. 클라이언트에 가게 등록 완료를 응답한다.
2. 가게 로그인
   1. 사용자의 인증 토큰과 가게 번호를 클라이언트로부터 받는다.
   2. 사용자 인증 토큰을 사용자 서비스로 전달하여 사용자의 개체식별번호를 획득한다.
   3. 사용자의 개체식별번호와 가게 번호를 검증한다.
   4. 가게 인증 토큰을 발급하여 클라이언트에 응답한다.
3. 상품 추가
   1. 클라이언트로부터 사용자 인증 토큰과 가게 번호, 상품 정보를 받는다.
   2. 사용자 서비스를 통해 사용자의 개체식별번호를 획득한다.
   3. 사용자의 개체식별번호와 가게 번호를 검증하고 상품을 업데이트한다.
   4. 클라이언트에 상품 추가 완료를 응답한다.
4. 상품 판매
   1. 클라이언트로부터 QR 정보와 가게 인증 토큰, 상품선택정보를 받는다.
   2. 가게 개체식별번호와 인증번호를 인증 서비스로 넘겨 인증번호 생성자(구매자)의 개체식별번호를 획득한다.
   3. 크레딧 서비스에 가게 소유자의 개체식별번호와 구매자의 개체식별번호, 그리고 송금 정보를 넘긴다.
   4. 클라이언트에게 결제 완료를 응답한다.

### UseCase

#### 1. 인증 서비스

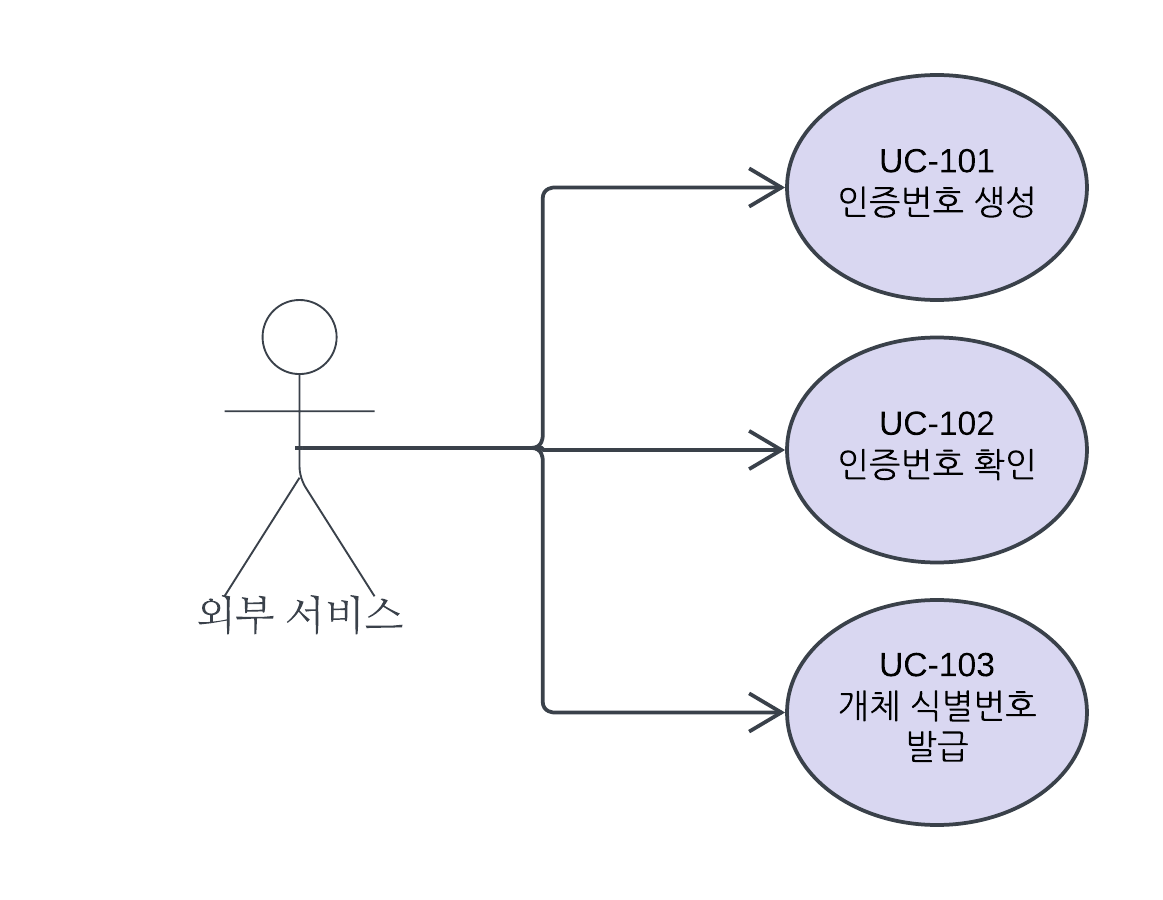


그림 6 인증 서비스 Usecase

* UC-101 인증번호 생성

외부 서비스에서 요청자의 개체식별번호와 함께 인증번호 생성을 요청하면, 유효기간이 짧은 인증번호를 생성하여 응답한다.

* UC-102 인증번호 확인

외부 서비스에서 인식한 인증번호, 인식자의 개체식별번호와 함께 인증번호 확인을 요청하면, 인증번호 생성자의 개체식별번호를 조회하여 서비스에 저장된 생성자와 인식자 각각의 인증번호 확인 응답 엔드 포인트로 생성자와 인식자의 개체식별번호를 전달한다.

* UC-103 개체식별번호 발급

외부 서비스에서 인증번호 확인이 발생했을 때 응답을 받을 엔드 포인트와 함께 개체식별번호 발급을 요청한다. 인증 서비스는 시스템에서 유일한 식별번호를 생성하고, 이를 전달받은 엔드 포인트와 함께 저장한다. 그리고 요청한 외부 서비스로 개체식별번호를 응답한다.

#### 2. 크레딧 서비스

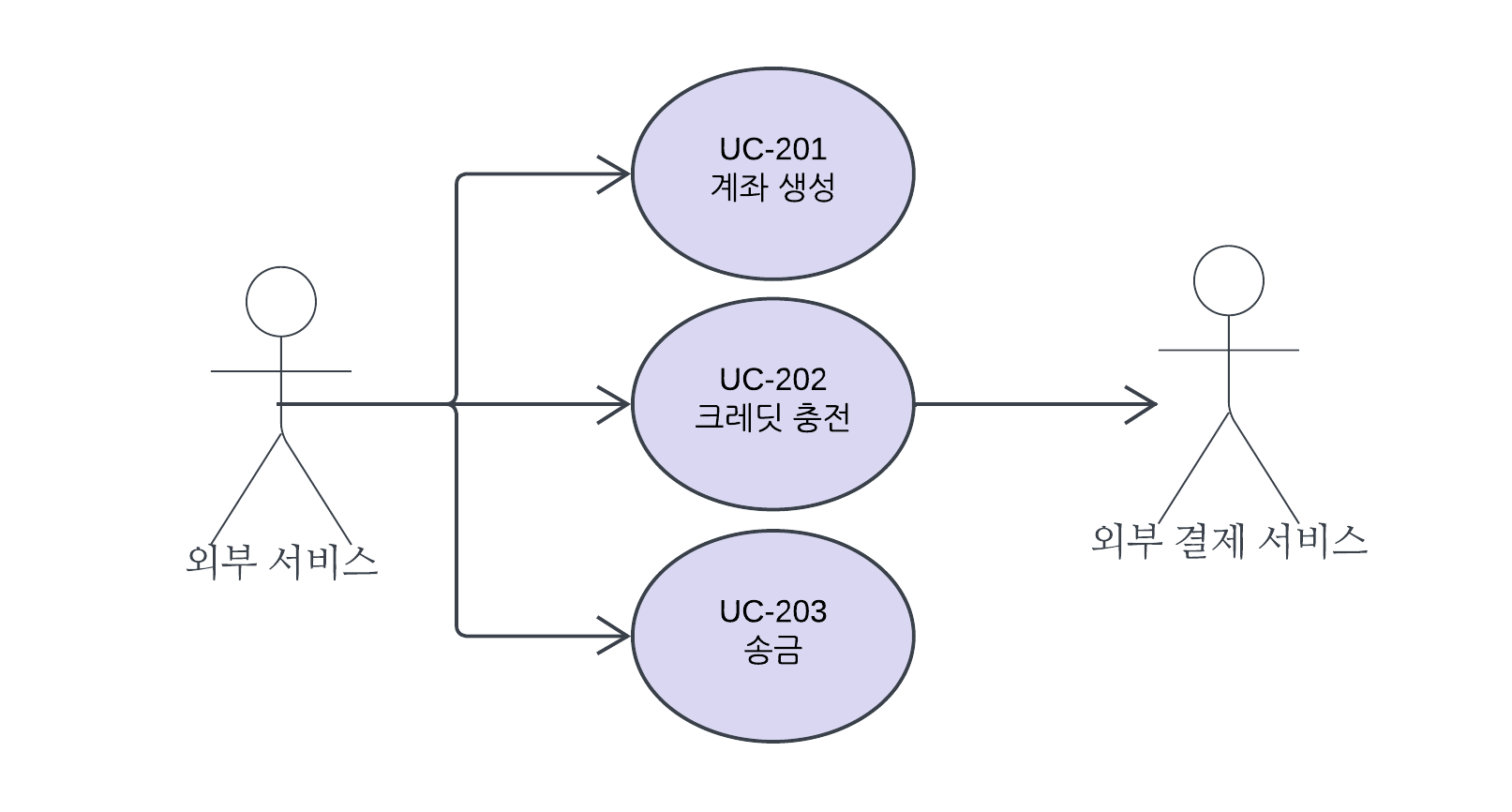


그림 7 크레딧 서비스 Usecase

* UC-201 계좌 생성

외부 서비스로부터 개체식별번호를 전달받아서 해당 식별번호의 크레딧 계좌를 생성한다.

* UC-202 크레딧 충전

외부 서비스로부터 개체식별번호와 충전 금액, 결제 정보를 받아서 외부 결제 서비스를 통해 결제를 진행한 후, 해당 식별번호의 계좌에 충전 금액만큼 크레딧을 충전한다.

* UC-203 송금

외부 서비스로부터 송금자와 수금자의 개체식별번호, 송금 정보를 받아서 송금자의 계좌에서 수금자의 계좌로 크레딧을 이동한다.

#### 3. 사용자 서비스

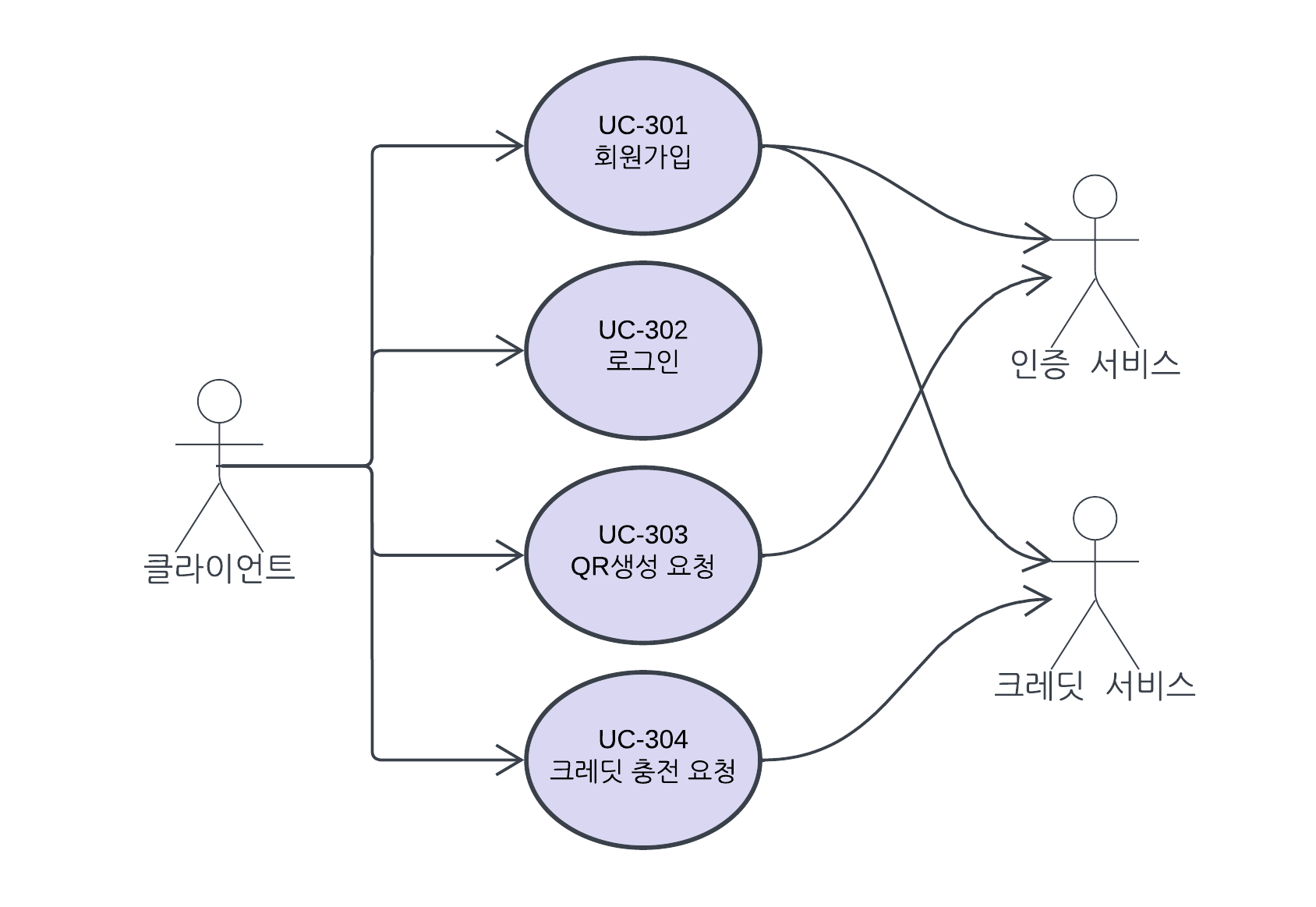


그림 8 사용자 서비스 Usecase

* UC-301 회원가입

클라이언트로부터 회원 정보를 입력 받아서 인증 서비스, 크레딧 서비스를 통해 개체식별번호와 계좌를 발급받고 회원 정보를 저장한다.

* UC-302 로그인

클라이언트로부터 아이디와 비밀번호를 입력 받아서 유효기간이 짧은 인증 토큰을 발급한다.

* UC-303 QR 생성 요청

클라이언트로부터 인증 토큰을 받아서 인증 서비스를 통해 인증번호를 생성한 후, QR 정보를 만들어 클라이언트에 응답한다.

* UC-304 크레딧 충전 요청

클라이언트로부터 인증 토큰과 충전 정보를 받아서 크레딧 서비스를 통해 크레딧을 충전한다.

#### 4. 가게 서비스

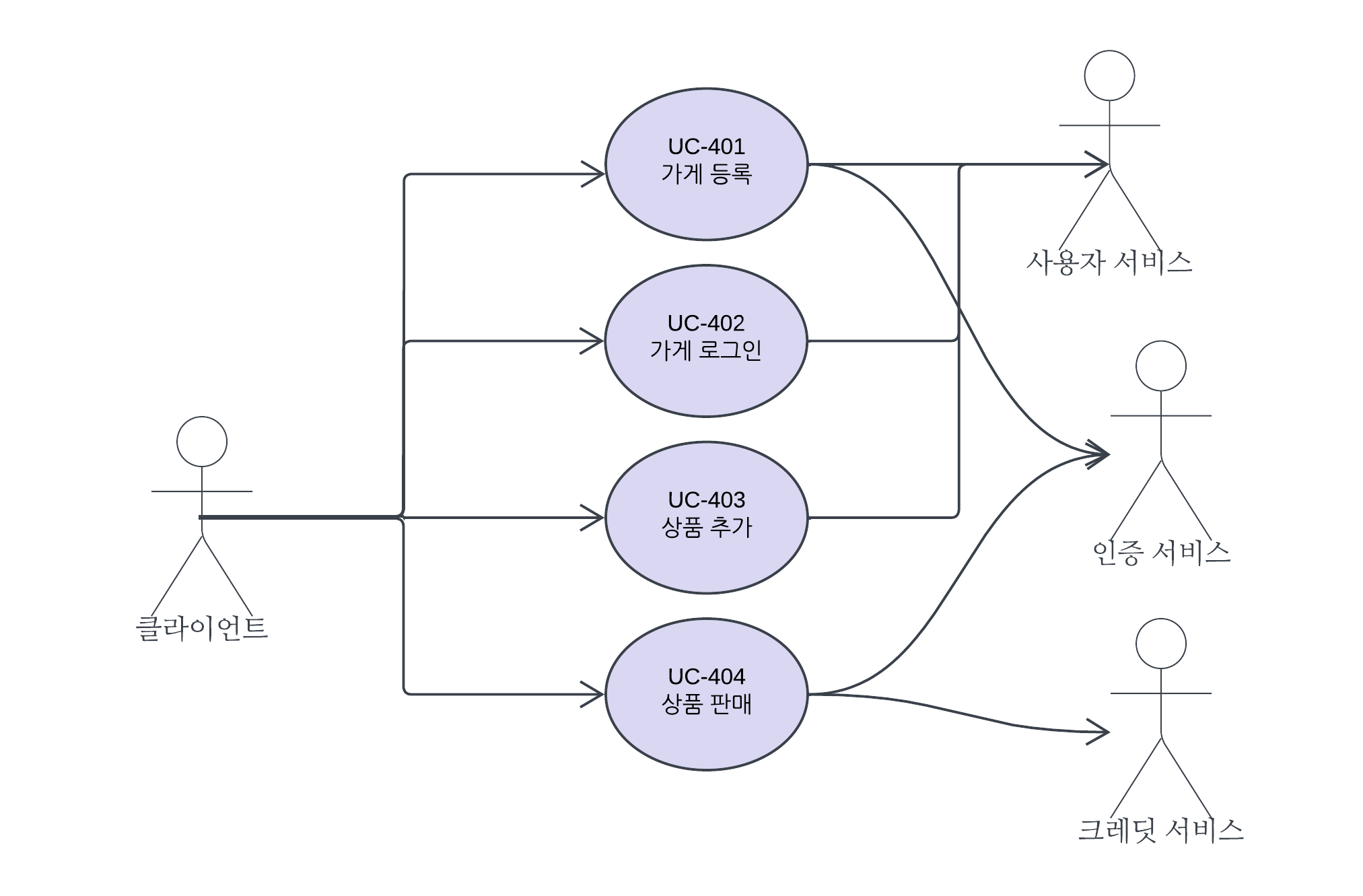


그림 9 가게 서비스 Usecase

* UC-401 가게 등록

클라이언트로부터 사용자 인증 토큰과 가게 생성 정보를 받아서 사용자 서비스를 통해 사용자의 개체식별번호를 획득한다. 그 후, 인증 서비스를 통해 가게의 개체식별번호를 생성하고 가게 생성 정보를 바탕으로 가게를 생성하고 등록한다.

* UC-402 가게 로그인

클라이언트로부터 사용자 인증 토큰과 가게 번호를 받아서 사용자 서비스를 통해 사용자의 개체식별번호를 획득한다. 이를 가게 번호와 함께 검증하고 유효기간이 비교적 긴 가게 인증 토큰을 생성하여 응답한다.

* UC-403 상품 추가

클라이언트로부터 사용자 인증 토큰과 가게 번호, 상품 추가 정보를 받아서 사용자 서비스를 통해 사용자 개체식별번호를 획득한 뒤, 이를 가게 번호와 함께 검증하고 상품 추가 정보를 바탕으로 상품을 추가한다.

* UC-404 상품 판매

클라이언트로부터 가게 인증 토큰, 구매 정보, QR 정보를 받아서 QR 정보에 포함된 인증번호를 인증 서비스에 전달하여 인증번호 생성자(구매자)의 개체식별번호를 획득한다. 이후, 가게 소유주와 구매자의 개체식별번호, 구매 정보를 크레딧 서비스에 전달하여 송금을 실행한다.

## 3-3 배포 파이프라인

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

그림 10 배포 파이프라인

개발은 Spring Boot와 Android로 진행하며 각각의 클라이언트와 서비스는 Jenkins가 SCM Poll하고 있는 서로 다른 git repository에 Push 한다. 그 후 특정 branch에 Commit이 발생하면 Jenkins는 SCM Poll을 통해 변경을 감지하고, 해당 코드를 테스트하고 빌드를 진행하게 된다.

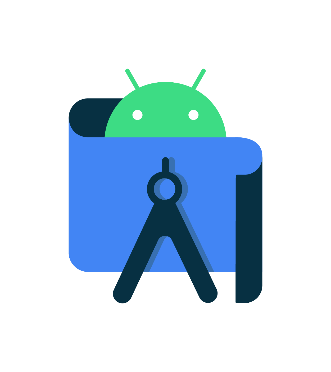
Jenkins에서 성공적으로 빌드가 완료되었다면, 안드로이드는 Fastlane을 통해 Play Store에 배포되고 서비스는 도커 이미지로 만들어져 Docker Hub에 저장된다. 그리고 클라우드상에 존재하는 cluster에서 Rolling Update를 진행한다. 사용자는 Play store에서 안드로이드 앱을 다운로드 받을 수 있고, 웹페이지와 앱을 통해 서비스를 이용할 수 있다.

# 4. 개발 환경 및 기술 스택

* Spring Boot

Spring Boot는 자바기반의 오픈 소스 프레임워크로, 스프링 기반 웹 애플리케이션 및 마이크로서비스 개발에 사용된다.

장점으로는 자동 구성을 통해 개발자가 설정 작업을 최소화하고 비즈니스 로직에 집중할 수 있다. 또한 내장형 서버를 제공하여 별도의 웹 서버 설정 없이도 애플리케이션 실행이 가능하다.

* Android Studio

Android Studio는 모바일 애플리케이션을 위한 통합 개발 환경이며 안드로이드 앱 개발에 필요한 다양한 도구, 에뮬레이터, 디버깅 기능 등을 제공한다.

장점으로는 풍부한 기능과 지원을 통해 안드로이드 앱 개발 과정을 효율적으로 수행할 수 있다. 또한 자동 완성, 코드 분석, 디자인 툴, 배포 기능 등 개발 생산성을 높여준다. 무엇보다 구글에서 공식적으로 안드로이드 개발도구로서, 안정성과 신뢰성이 매우 높다.

* Kubernetes

원, 상징, 로고, 바퀴이(가) 표시된 사진

자동 생성된 설명kubernetes는 컨테이너 오케스트레이션 플랫폼으로, 복잡한 어플리케이션 배포 및 관리를 간편하게 도와주며 분산 시스템을 위한 자동화된 컨테이너 배포, 확정, 복구, 로드 밸런싱 등을 제공한다. 장점으로는 자원 효율성을 높이고, 가용성과 확정을 보장한다. 또한 강력한 스케줄링 및 자가 치유 기능을 통해 안정적인 시스템 운영을 가능하게 해준다.

# 5. 개발 일정 및 역할 분배

## 5-1 개발 일정

Table 1 개발 일정

| 6월 | | | | | 7월 | | | | | 8월 | | | | | 9월 | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1주 | 2주 | 3주 | 4주 | 5주 | 1주 | 2주 | 3주 | 4주 | 5주 | 1주 | 2주 | 3주 | 4주 | 5주 | 1주 | 2주 | 3주 | 4주 | 5주 |
| 기자재 구입 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | API  endpoint 명세 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 개발 환경 구축 및  개별 서비스 구현 | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 서비스 간 통신  구현 | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | 중간보고서 작성 | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | 서비스 통합 | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | 배포 및 테스트 | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  | 시스템  최적화 | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | 보약 취약점 파악 및 보완 | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 최종 보고서 작성 및 발표,심사 준비 | | | | |

## 5-2 역할 분배

Table 2 역할 분배

|  |  |
| --- | --- |
| 이름 | 역할 |
| 송재원 | * CI & CD 환경 구축 및 클라우드 환경 관리 * 쿠버네티스 클러스터 구축 * 시스템 최적화와 테스트 |
| 김민승 | * 안드로이드 어플리케이션 개발 * 키오스크 클라이언트 개발 * 크레딧 충전 웹페이지 구현 |
| 김민종 | * 마이크로 서비스 환경 구축 및 개별 서비스 구현 * 서비스 단위 테스트와 검증 * 서비스 디스커버리, 게이트웨이 개발 |
| 공통 | * 시스템 보안성 추가 * 데이터베이스 설계 및 관리 * 기타 문서화 작업 |