



피로회복

헌혈증 기부 서비스

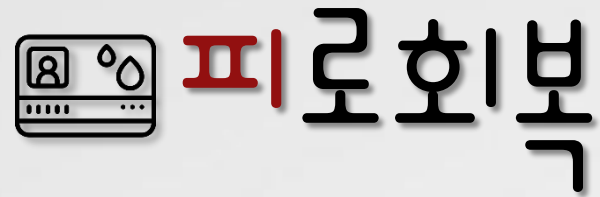


CONTENTS

- 1 프로젝트 기획
- 2 프로젝트 계획
- 3 프로젝트 진행
- 4 프로젝트 시연
- 5 회고

프로젝트 기획 프로젝트 계획 프로젝트 진행 시연 회고

- 프로젝트 주제



: 헌혈증 전산화를 통한 헌혈증 기부 시스템 구축

- 프로젝트 주제 선정 이유



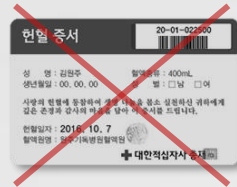
출처: 대한적십자사

- 프로젝트 주제 선정 이유



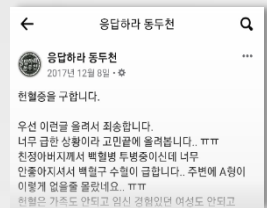
1. 헌혈증 보관 문제

: 헌혈증은 종이로 되어있어서 분실 위험이 높음



2. 헌혈증 재발급 불가

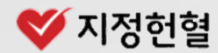
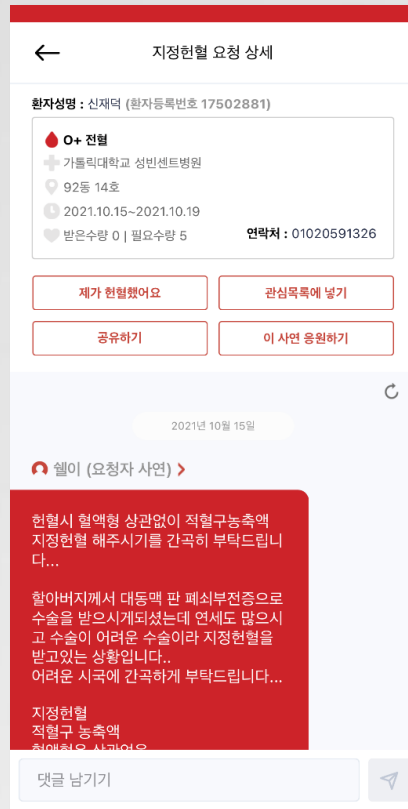
: 헌혈증은 유가증권의 성격을 지니고 있어, 분실하거나 훼손되어도 재발급 불가



3. 헌혈 커뮤니티 부족

: 다양한 사이트에서 헌혈증 기부 요청글과 지정헌혈 요청글이 올라옴

■ 벤치마킹

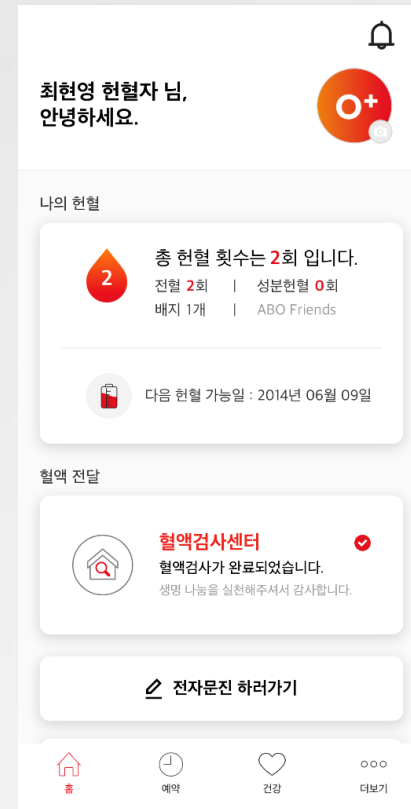


지정헌혈

:유자들 간에
지정헌혈을 요청하고 신청하는
커뮤니티 어플리케이션



**신청하지 않은
유저에게도
환자의 개인정보 노출**



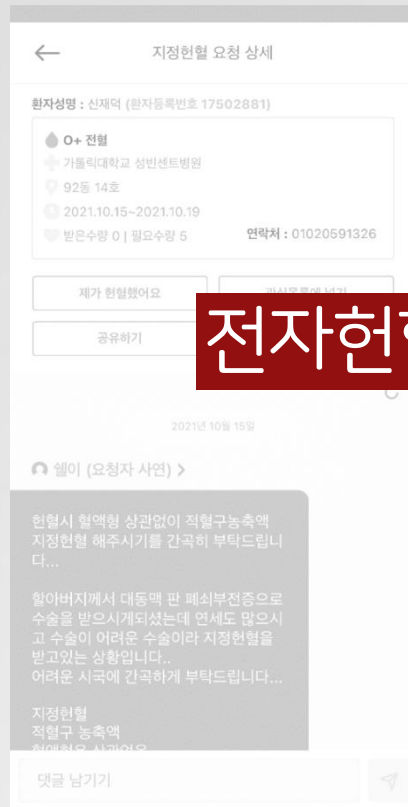
:나의 헌혈 정보 및 기록을 확인하고
헌혈의 집 예약을 할 수 있는
어플리케이션



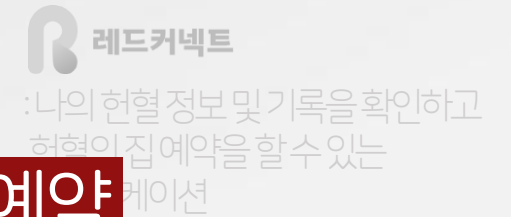
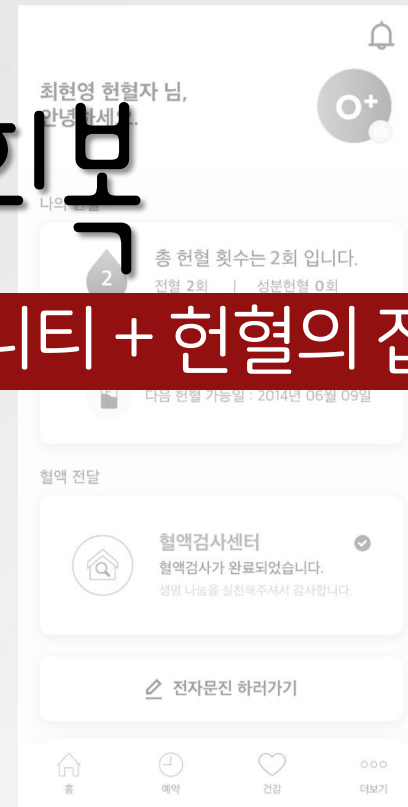
헌혈 커뮤니티 부재

프로젝트 기획 프로젝트 계획 프로젝트 진행 시연 회고

■ 벤치마킹



전자헌혈증 지갑 + 헌혈 커뮤니티 + 헌혈의 집 예약



1. 헌혈증 전산화

BIMS란?

: 모든 혈액관리업무를 인터넷을 통하여 관리할 수 있는 **혈액정보관리시스템**
(Blood Information Management System ; BIMS)

BISS란?

: BIMS를 구성하는 시스템 중 하나로 헌혈자 관리, 혈액예약/주문, 검사결과입력/조회, 혈액출고관리, 혈액입출고 등을 관리할 수 있는 **혈액정보공유시스템**
(Blood Information Sharing System ; BISS)

1. 헌혈증 전산화

★ 나의메뉴 추가

검사결과

 **최현영님!** 사랑의 헌혈에 참여해 주셔서 감사합니다. 앞으로도 지속적인 관심과 참여 부탁드립니다.

헌혈횟수 총 2회

· 전혈 : 2회 · 혈장 : 0회 · 혈소판 : 0회 · 혈소판혈장 : 0회 · 기타 : 0회

혈액검사결과

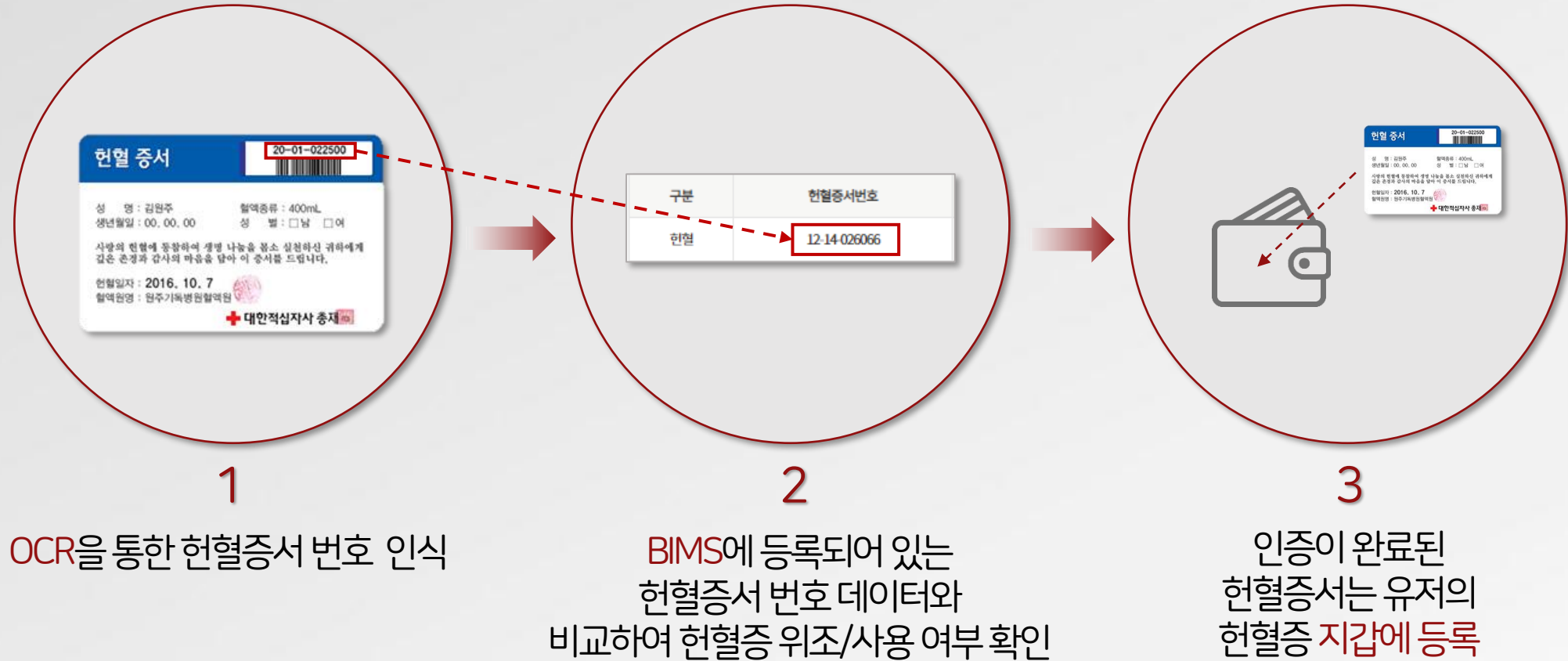
구분	헌혈증서번호	혈액형	헌혈하신날짜	헌혈종류	헌혈횟수
헌혈	I2-14-026066	O(+)	2014-04-14	전혈	2/2

검사항목	참고치	검사결과
B형간염 (HBV)	바이러스항원 (HBsAg)	음성

BIMS를 활용하는
대한적십자사 혈액본부 사이트에서
출력한 **헌혈 검사 결과지**

➡ BIM의 데이터 유추 가능

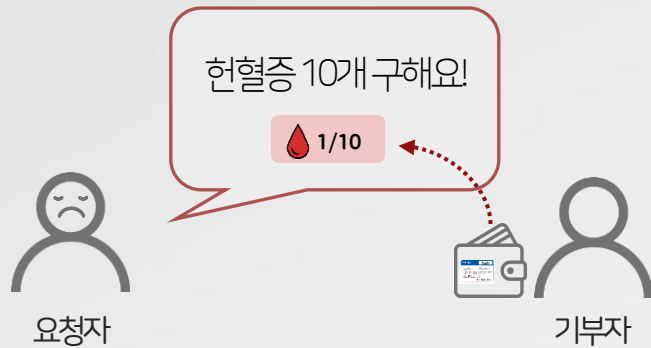
1. 헌혈증 전산화



2. 헌혈증 기부 및 지정헌혈 신청

헌혈증 기부

:유저들 간에 헌혈증을 요청하고, 전산화 된 헌혈증을 기부



지정 헌혈

:유저들 간에 지정 헌혈을 요청하고, 지정 헌혈 신청 및 인증

지정 헌혈이란? 헌혈을 하는 사람이 수혈자를 지정하여 헌혈하는 것

1. 지정헌혈 신청 후 수혈자 **정보 열람**

2. 신청자는 요청 기간 내에 헌혈증으로 지정 헌혈 **인증**

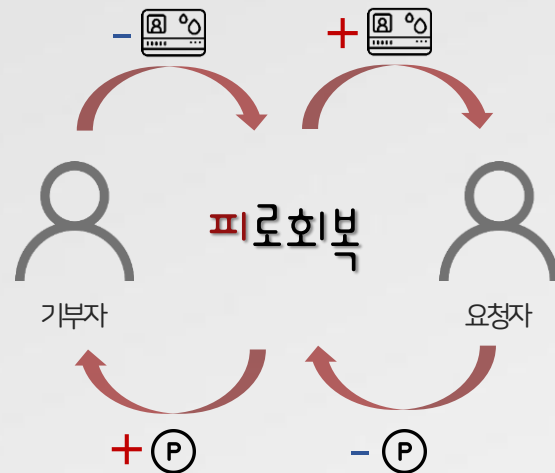
*기간 내에 3번 이상 인증하지 않을 시 페널티 부여

*진행 중인 지정 헌혈을 완료하기 전까지 중복 신청 불가능

3. 포인트제도 및 랭킹

헌혈증 기부, 지정헌혈과 프로모션 참여를 통한

포인트 제도 운영



커뮤니티 활성화를 위한

랭킹 서비스 운영



■ 수익 구조

===== 9월 프로모션 =====

2021.09.30

 나청일

안녕하세요. '피로회복' 담당자 나청일 입니다!
9월 프로모션 안내드립니다! 다음 달에도 많은 참여 부탁드립니다.



*이미지 클릭 시 링크로 이동합니다. 접속 시 5p 적립됩니다 :)



- 서비스 관련 법안

혈액관리법(1999. 2. 8, 전부개정)

제3조 (혈액매매행위등의금지)

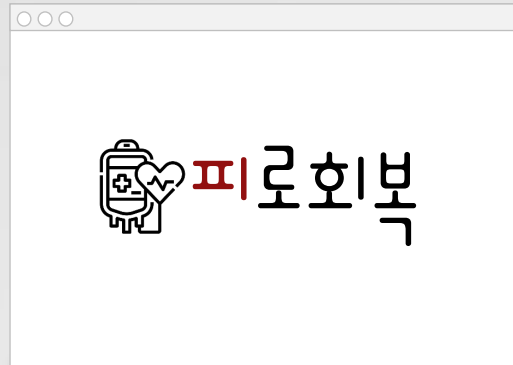
①누구든지 금전·재산상의 이익 기타 대가적 급부를 받거나 받기로 하고 자신의 혈액(제14조에 따른 헌혈증서를 포함한다)을 제공하거나 이를 약속하여서는 아니된다.

②누구든지 금전·재산상의 이익 기타 대가적 급부를 주거나 주기로 하고 타인의 혈액(제14조에 따른 헌혈증서를 포함한다)을 제공받거나 이를 약속하여서는 아니된다.

③누구든지 제1항 및 제2항의 규정에 위반되는 행위를 교사·방조 또는 알선하여서는 아니된다.

④누구든지 제1항 및 제2항의 규정에 위반되는 행위가 있음을 안 때에는 그 행위와 관련되는 혈액을 채혈하거나 수혈하여서는 아니된다.

- 기대효과



1. 헌혈증 관리 편리
2. 헌혈증 기부 편리
3. 헌혈증 사용률 증가
4. 헌혈 증진효과

■ 팀원 소개



한정헌멘토님



강혜수

스크럼 마스터



김민정

프론트 팀장



나청일

인프라 팀장

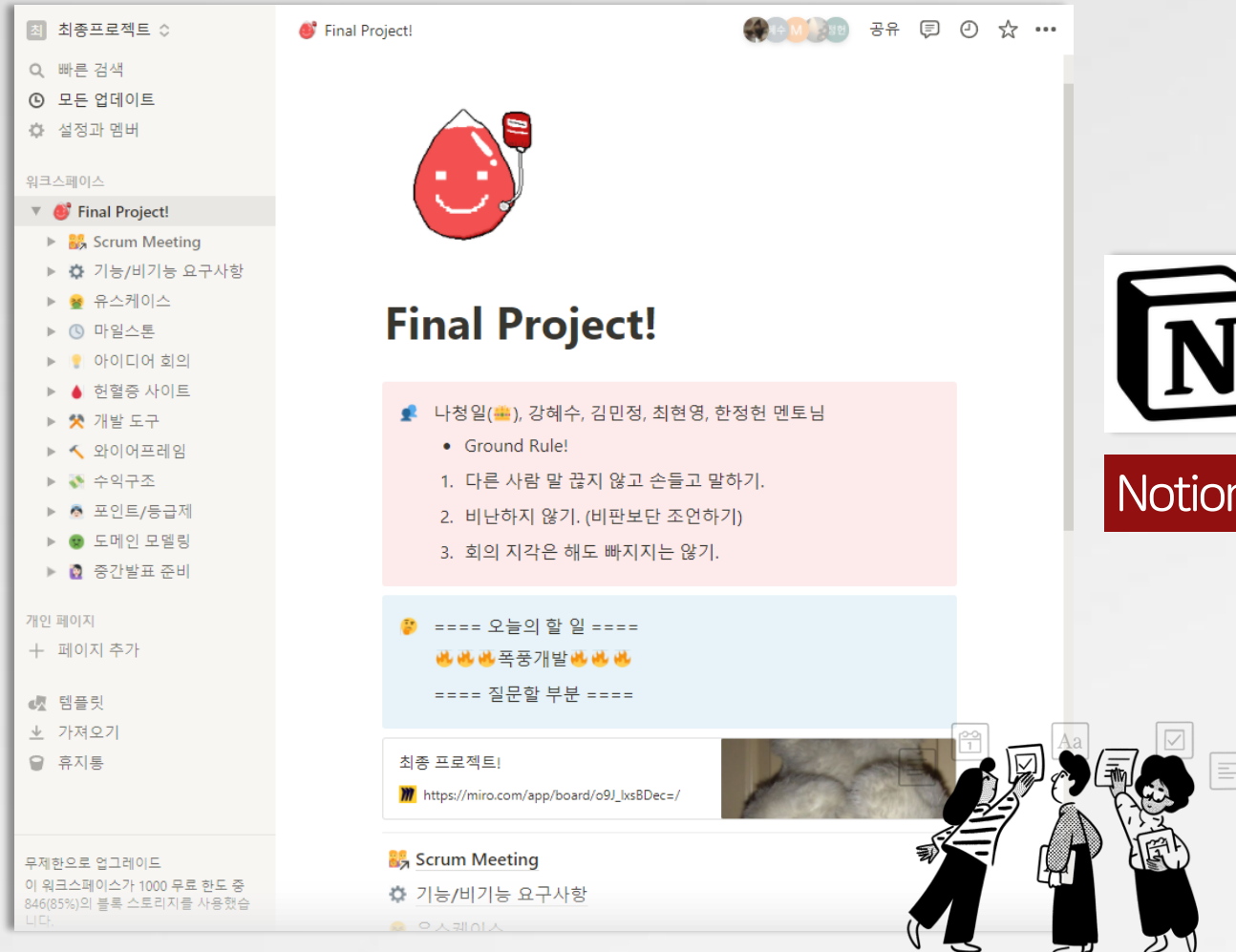
PM



최현영

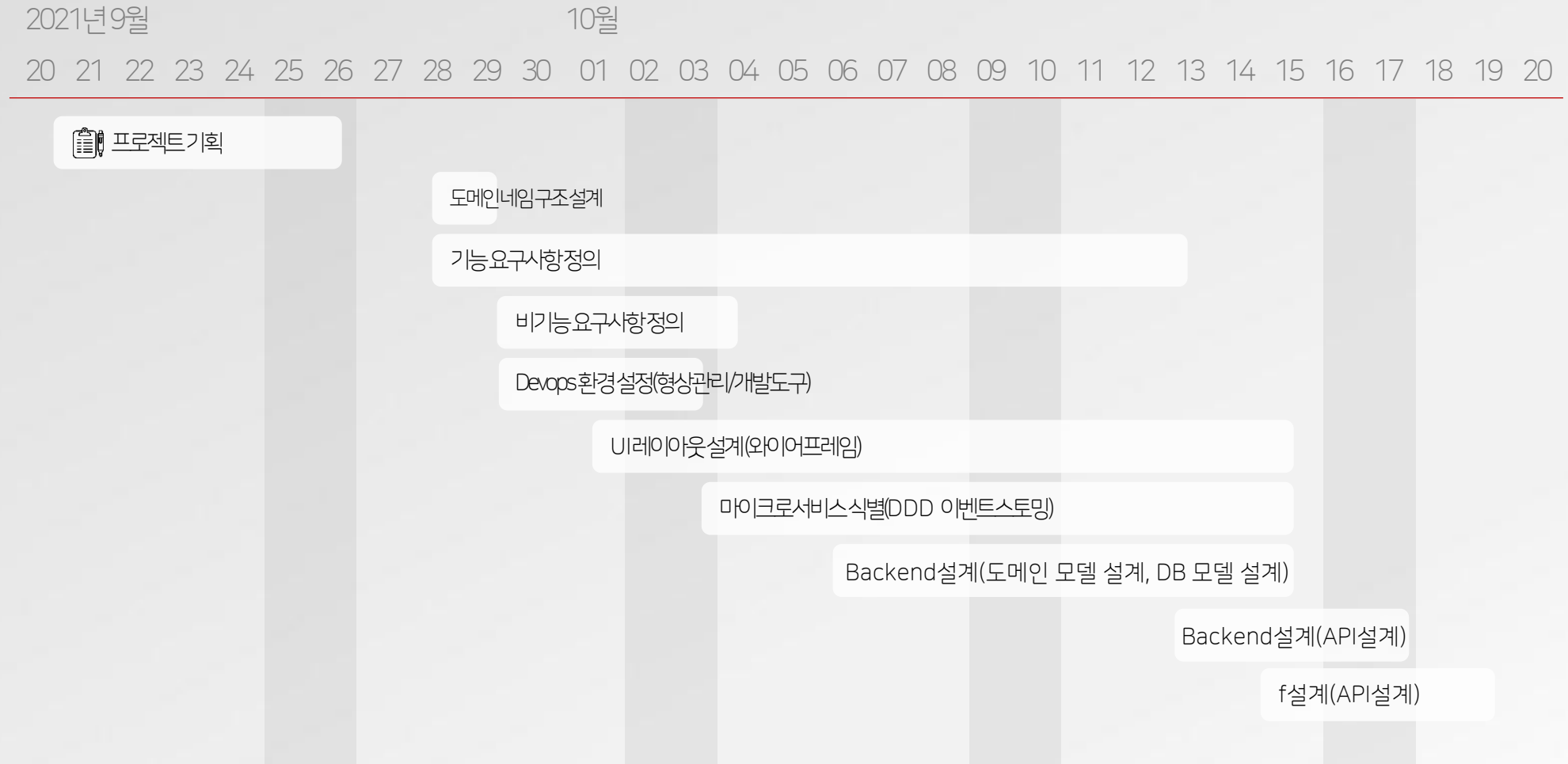
백엔드 팀장

■ 일정관리 및 의사소통



Notion을 통한 일정관리 및 의사소통

■ 프로젝트 설계 일정



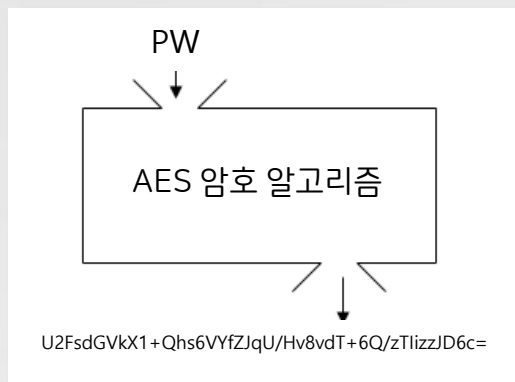
■ 기능 요구사항 정의서

👤 기부					
헌혈증 기부					
Aa 이름	≡ 설명	≡ 비고	≡ 소요 시간(일)	● 중요도	+
<u>헌혈증 기부 요청 정보 보기</u>	모든 사용자는 모든 헌혈증 기부 요청 정보 보기가 가능하다.	헌혈증 기부 요청의 순서는 최신순으로 하되, VIP 등급의 요청은 항상 최신 요청 취급을 받는다.	3	상	
<u>헌혈증 기부 요청</u>	로그인이 된 상태의 사용자만 헌혈증 기부 요청이 가능하다.	<ul style="list-style-type: none"> - 헌혈증 기부 요청의 양식은 제목, 요청 사연(사진첨부가 가능)으로 이루어진다. - 요청 작성시 필요한 헌혈증 개수를 선택 가능하다. (요청 하개 당 최대 헌혈증 10개까 	1	중	

■ 비기능 요구사항 정의서

보안

1. 사용자 정보를 주고 받을 때는
사용자 정보가 외부에 노출되지 않는다.
(post 전송, 비밀번호 암호화)



2. 보안과 관련된 정보(쿼리문, api 주소)는
사용자에게 노출하지 않는다.

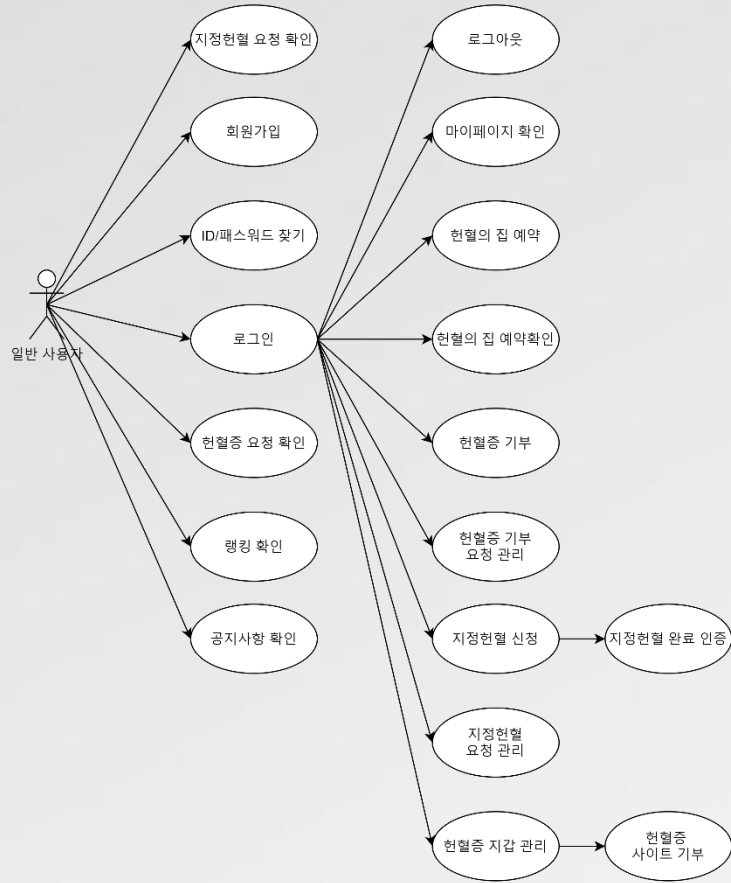
가용성

1. 이 시스템은 365일 24시간 가동된다.
2. 시스템에 부하가 걸려도 서비스를 유지한다.(마이크로서비스로 설계한다.)

통신 및 성능

1. 응답 시간은 3초를 넘기지 않는다.
2. 각 서비스는 rest api로 제공한다.

■ 유스케이스 모델 정의

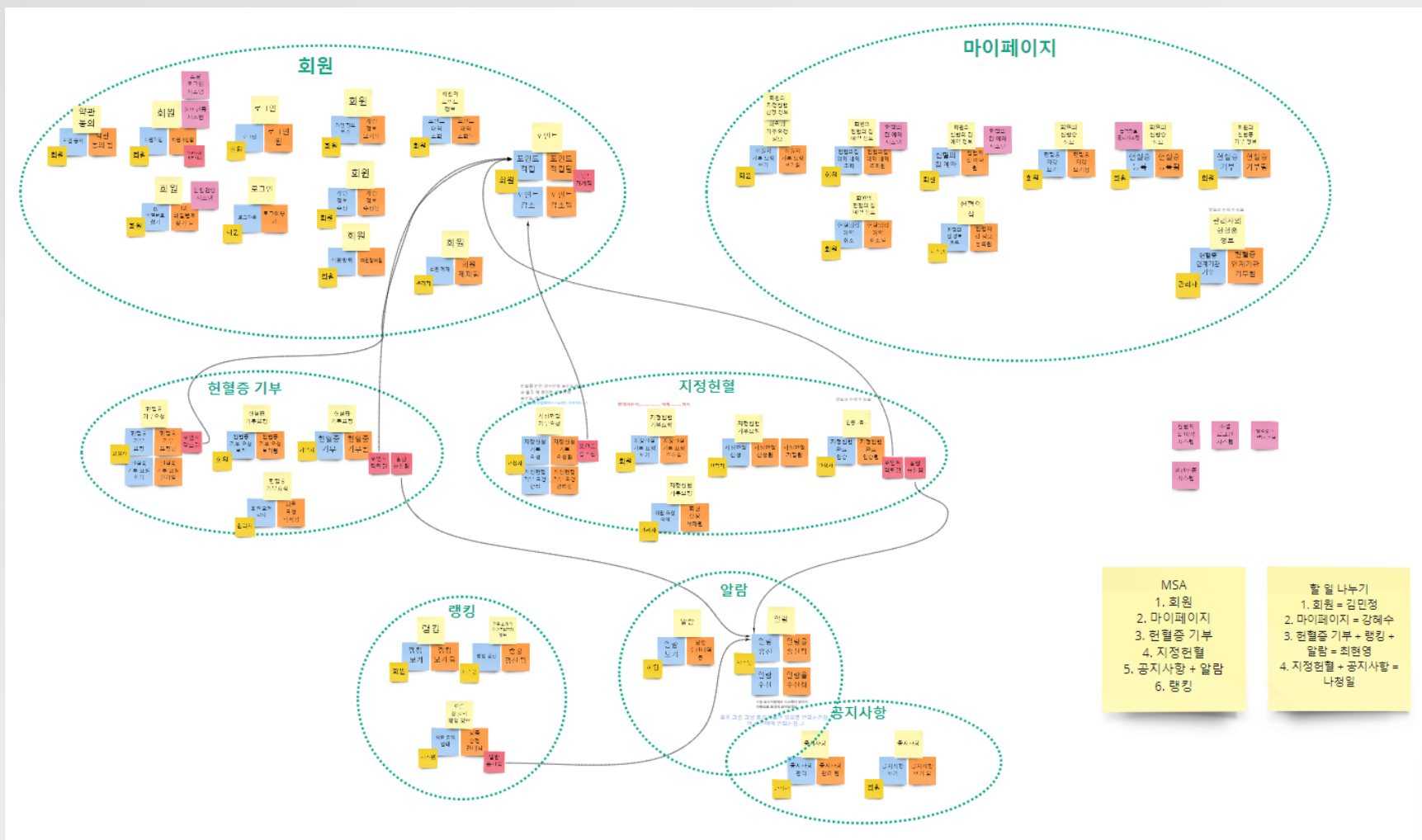


<유스케이스 다이어그램>

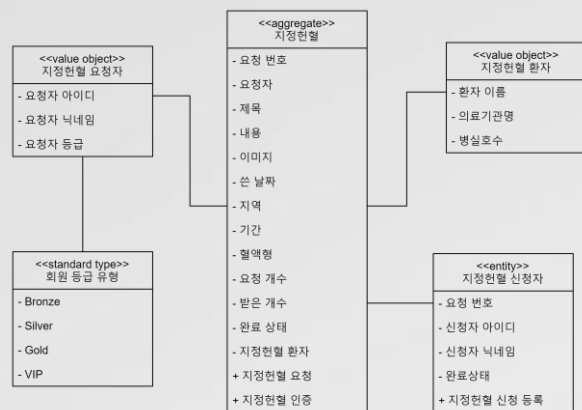
마이페이지		
헌혈증 기부		
헌혈증 기부 요청 확인		
Aa 제목	≡ 내용	≡ 비고
<u>유스케이스명</u>	사용자의 헌혈증 요청 확인	
<u>액터명</u>	사용자	
<u>개요</u>	사용자는 헌혈증 요청을 확인 할 수 있다.	
<u>사전조건</u>		
<u>사후조건</u>		
<u>기본흐름</u>	1. 사용자가 헌혈증 기부 요청을 확인을 시작한다. 2. 시스템은 헌혈증 기부 요청 리스트를 보여준다. 3. 사용자는 헌혈증 기부 요청 리스트를 볼 수 있다. 4. 사용자는 키워드를 검색하거나 기부 요청 필터링을 설정한다. 5. 시스템은 헌혈증기부DB에서 해당 키워드와 필터링 조건을 검색 후 결과를 보여준다. 6. 사용자는 필터링 된 헌혈증 기부 요청 리스트를 볼 수 있다.	-필터링 1. 요청의 상태 : 진행중/완료

<유스케이스 명세서>

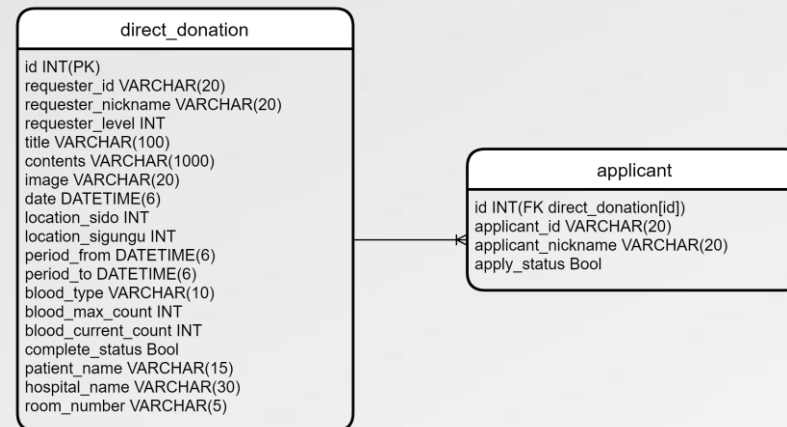
- 이벤트스토밍 워크숍을 통한 마이크로서비스 도출



■ BackEnd 설계



<도메인 모델링>

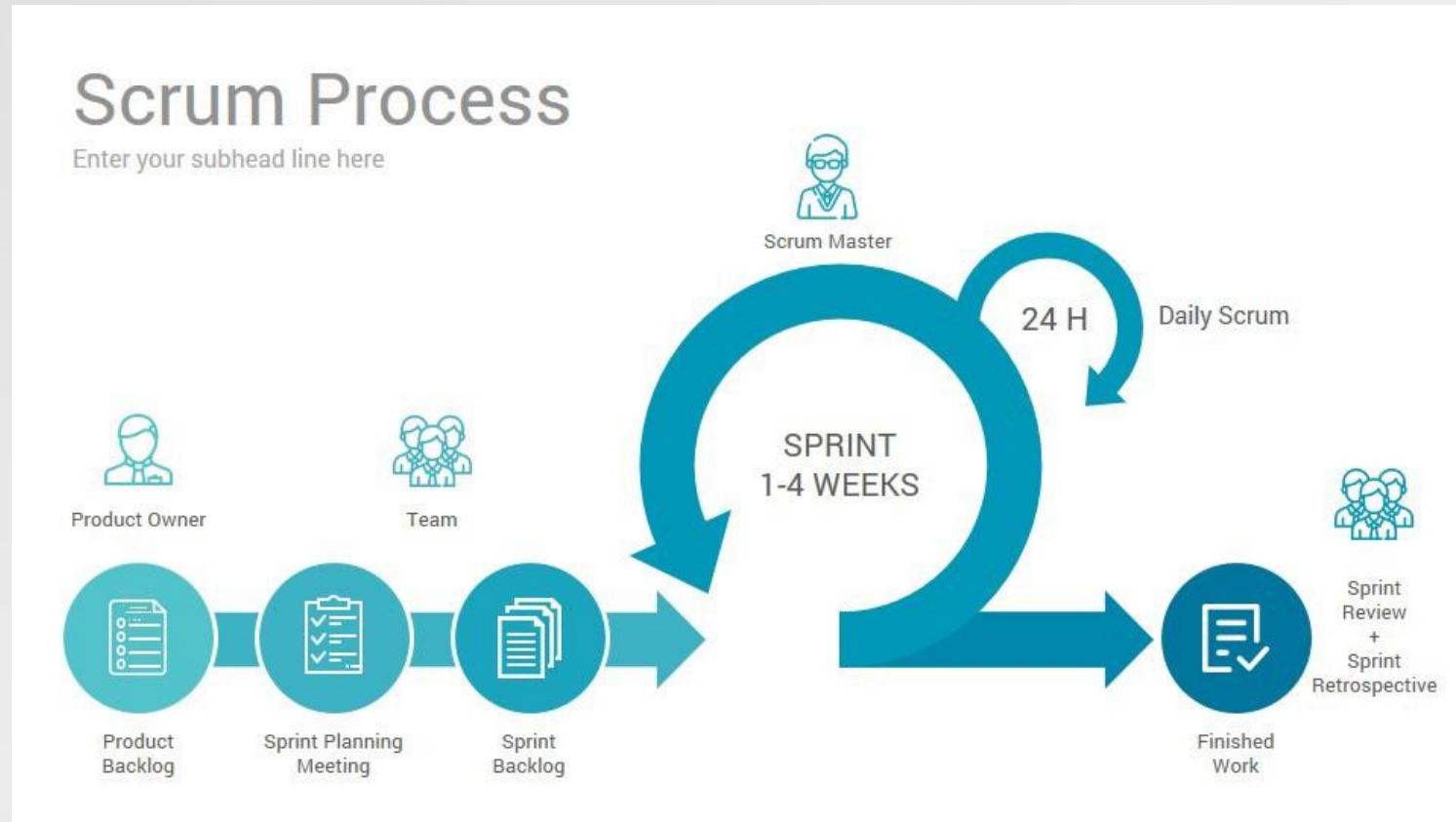


<데이터 모델링>

API 설계서							
서비스명	CardRequest						
API명	헌혈증 기부 요청 수정						
URI	/requests/requestItem/{id}						
Method	PUT						
Request							
Parameter	L1	L2	L3	L4	Type	필수여부	설명
index	id				Number	O	수정할 요청의 고유번호
통신시간	trans_time				String	O	통신발행시간
응답코드	result_code				String	O	정상통신시 OK
비고	desc				String	X	
DATA	data				Object	O	BODY부

<API설계서>

- 개발 방법론



■ 개발 방법론

1. 목표 설정

: 어떤 기능의 어디까지 구현하겠다는 명확한 목표

2. 구현

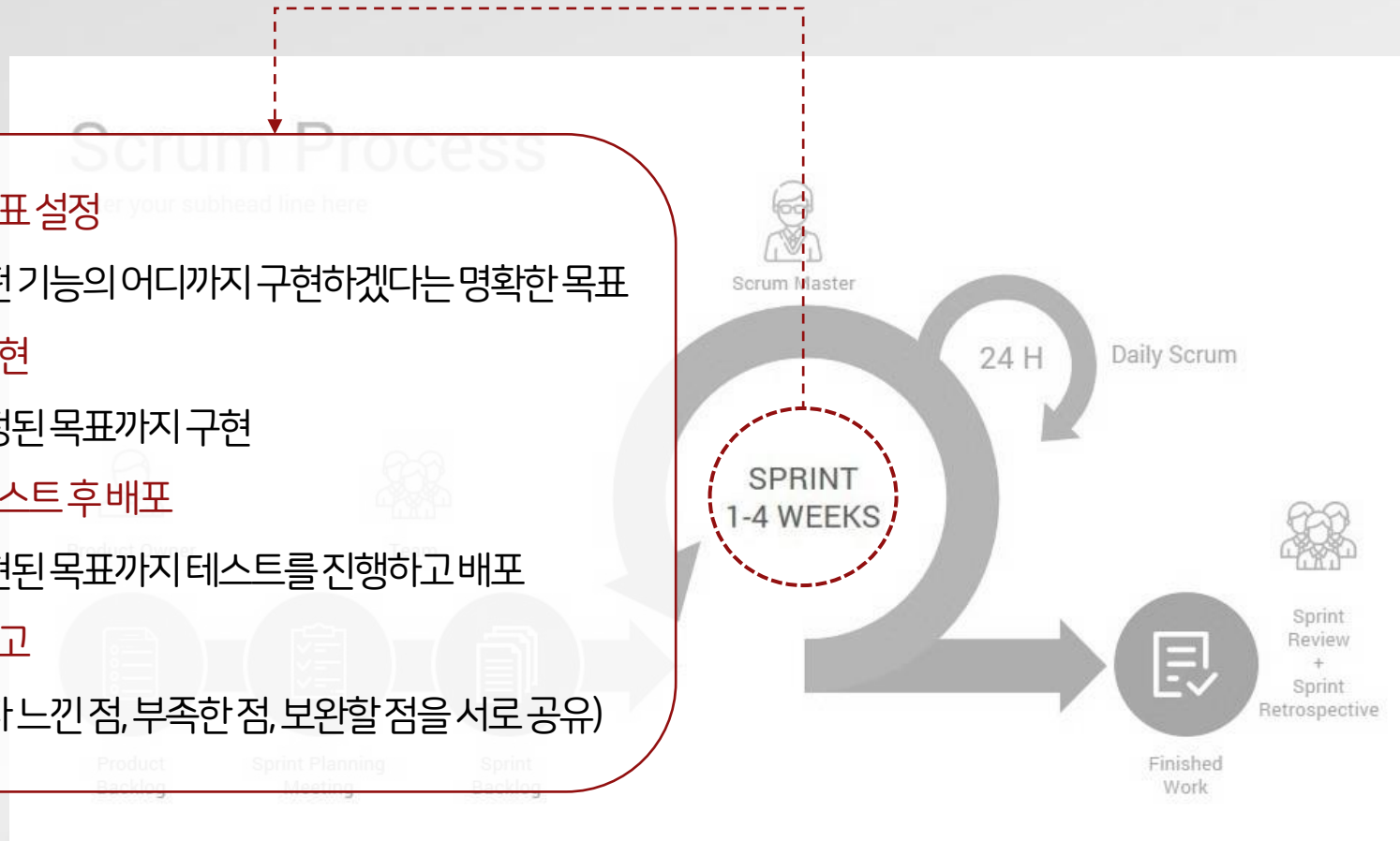
: 설정된 목표까지 구현

3. 테스트 후 배포

: 구현된 목표까지 테스트를 진행하고 배포

4. 회고

: 각자 느낀 점, 부족한 점, 보완할 점을 서로 공유)

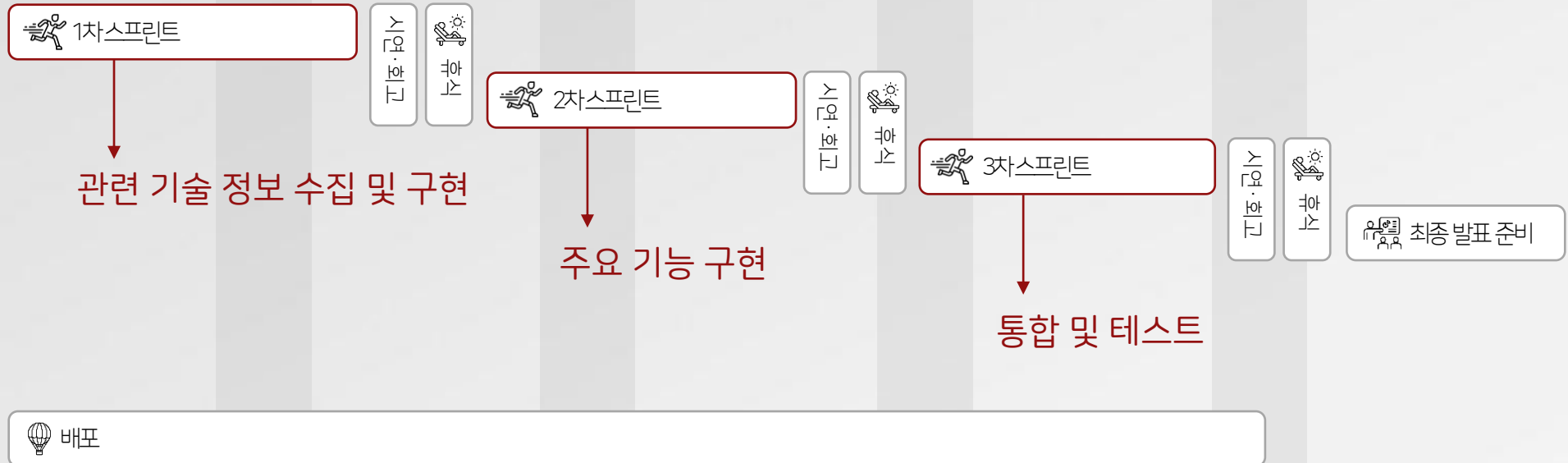


- 개발 진행 일정

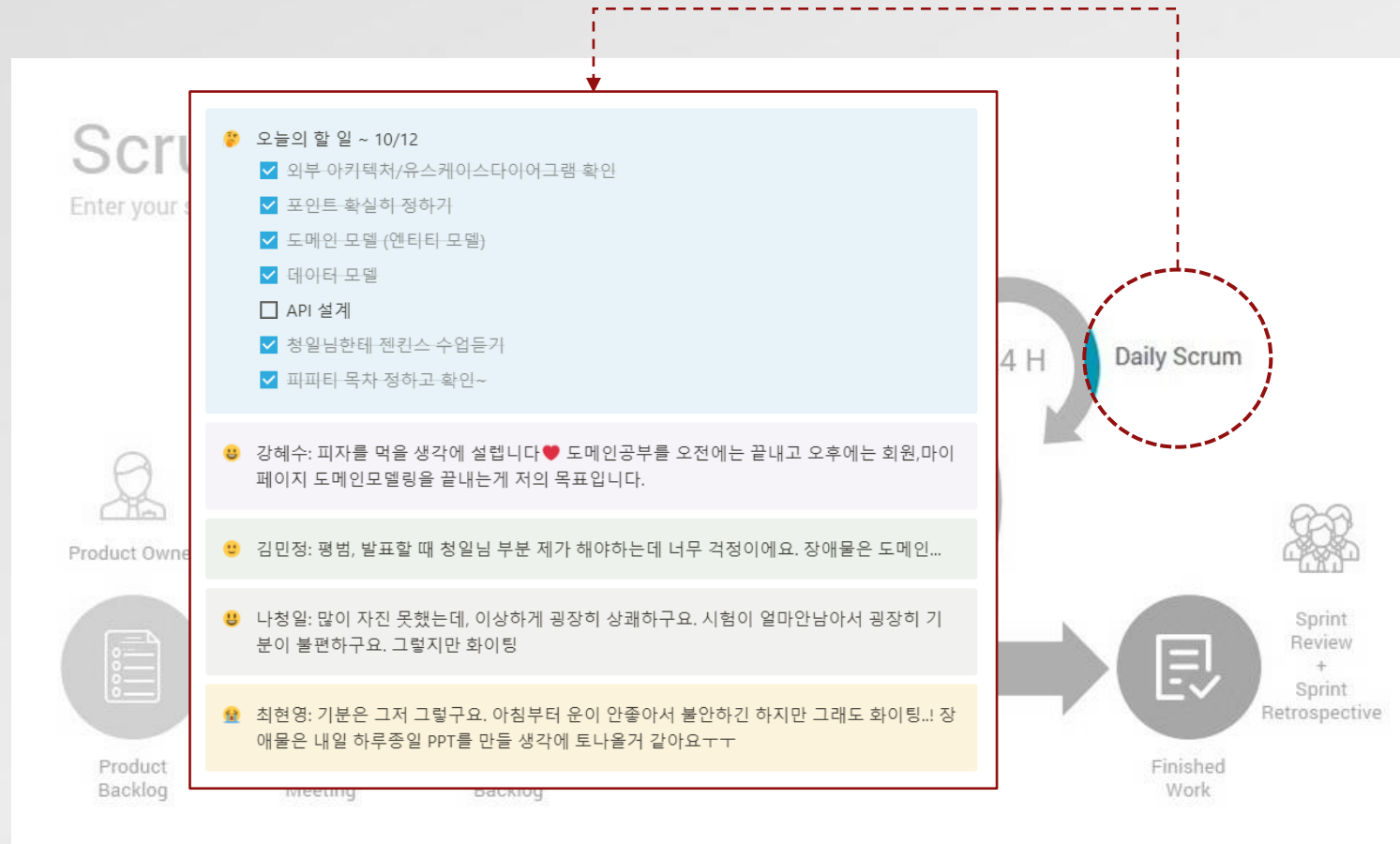
2021년 10월

11월

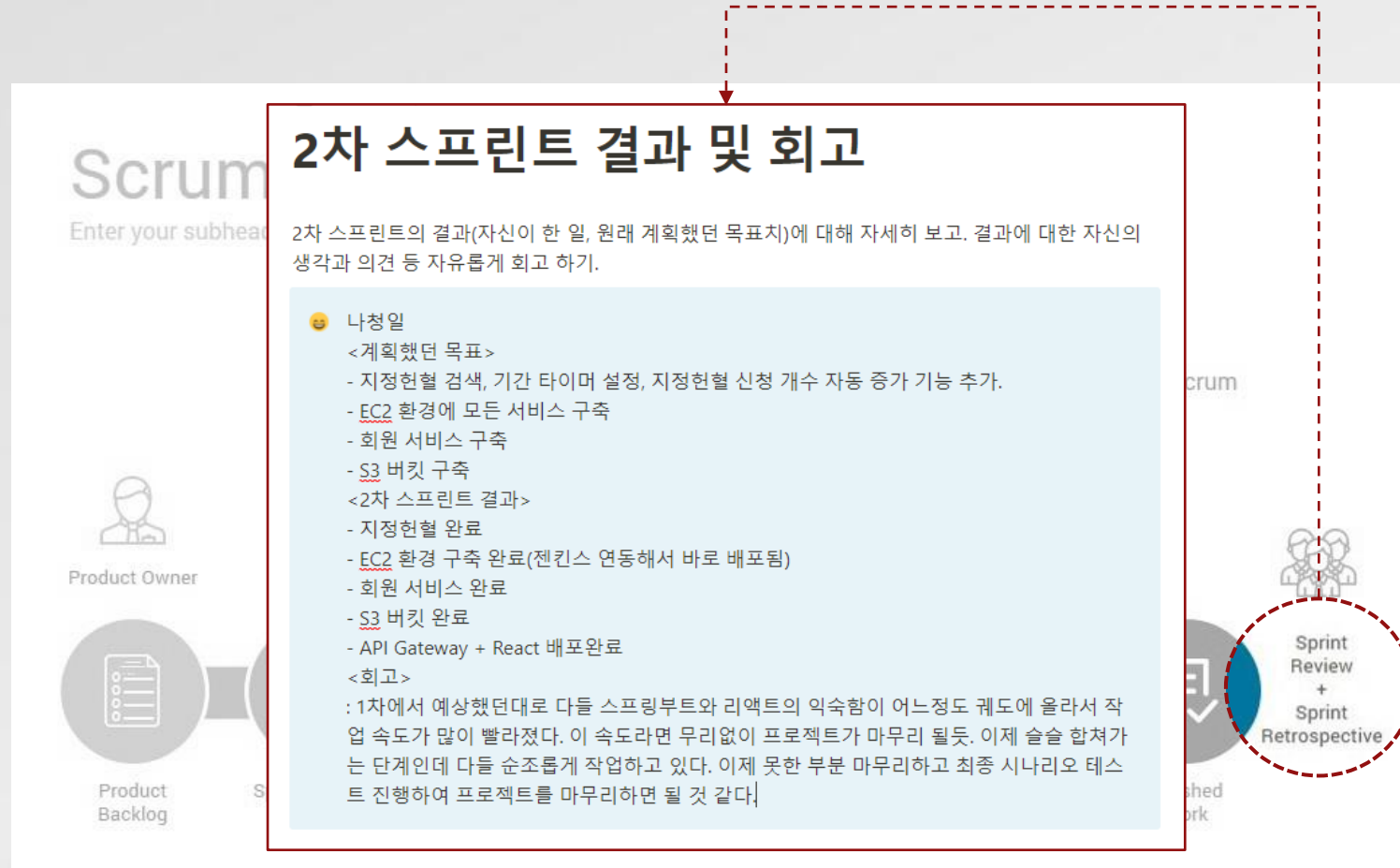
18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20



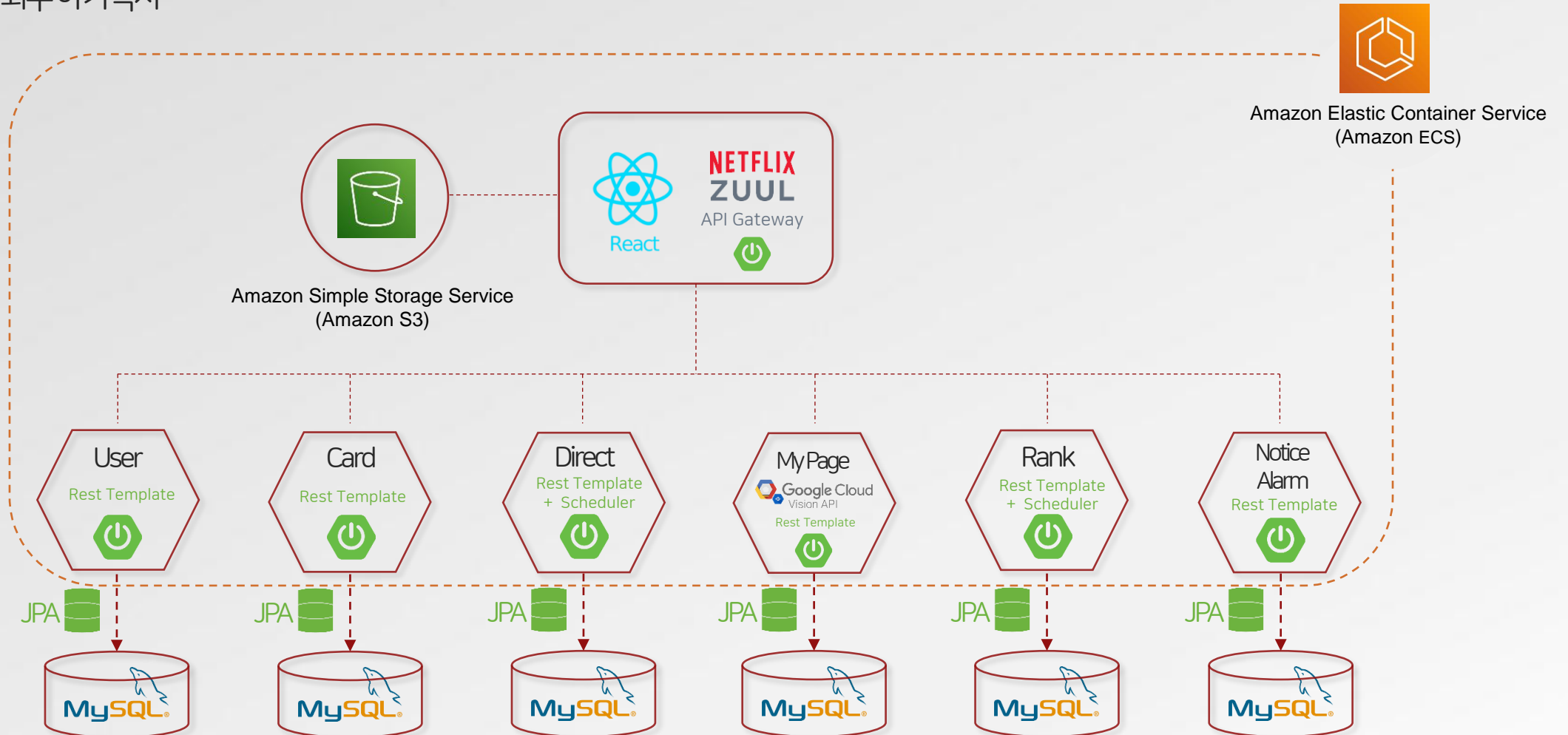
■ 개발방법론



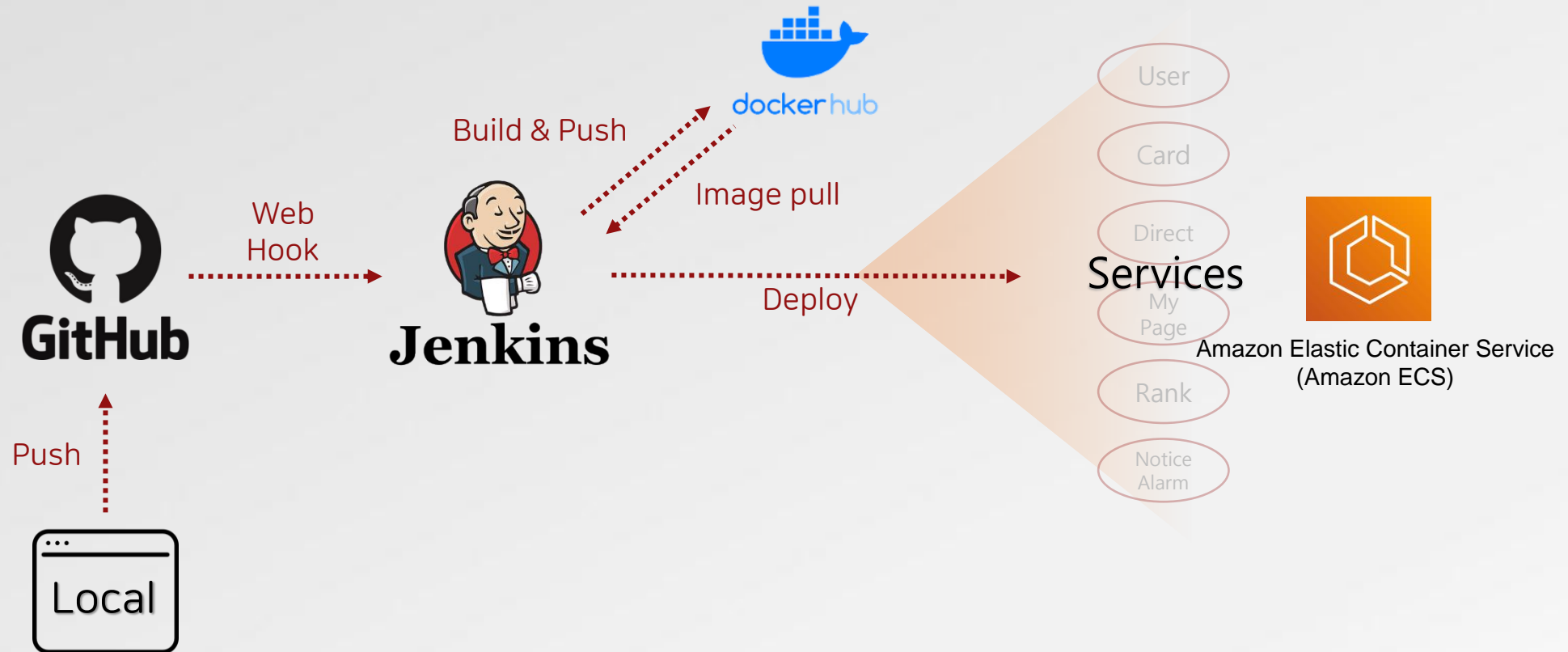
■ 개발방법론



■ 외부아키텍처



- DevOps 환경 구축 (CI/CD)



■ API 테스트

The screenshot displays the Postman web interface. On the left, the 'My Workspace' sidebar shows a collection of API requests under the '지정현혈' (Designated Blood) folder. The main panel shows a POST request to 'BloodRecovery-LB-1423483073.us-east-2.elb.amazonaws.com:8000/direct/directedItem'. The request body is in JSON format, containing fields like 'requesterUserId', 'title', 'contents', 'image', 'locationSido', 'locationSigungu', 'periodFrom', 'periodTo', 'bloodType', and 'bloodMaxCount'. The response is also in JSON format, showing the same fields with values. The status bar at the bottom indicates a successful 200 OK response with a 309 ms latency and 858 B of data.

Home Workspaces Reports Explore

Search Postman

My Workspace

Collections

APIs

Environments

Mock Servers

Monitors

History

지정현혈 / 지정현혈 등록

POST BloodRecovery-LB-1423483073.us-east-2.elb.amazonaws.com:8000/direct/directedItem

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

```
1 {
2   "requesterUserId": "아이디10",
3   "title": "제목이지롱~222",
4   "contents": "내용이지롱~~222",
5   "image": "https://bloodrecovery.s3.us-east-2.amazonaws.com/direct/
6     97883a2f-dafa-4dc5-9875-1a4a450cd45b.jpg",
7   "locationSido": "Busan",
8   "locationSigungu": "TestGu",
9   "periodFrom": "2021-11-08T16:00:00Z",
10  "periodTo": "2021-11-30T00:00:00Z",
11  "bloodType": "B",
12  "bloodMaxCount": 1,
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
8 "image": "https://bloodrecovery.s3.us-east-2.amazonaws.com/direct/
9   97883a2f-dafa-4dc5-9875-1a4a450cd45b.jpg",
10 "date": "2021-11-16T15:05:14.405028",
11 "locationSido": "Busan",
12 "locationSigungu": "TestGu",
   "periodFrom": "2021-11-08T16:00:00",
```

200 OK 309 ms 858 B Save Response

Find and Replace Console

Bootcamp Runner Trash



➡ API 테스트 활용

■ DevOps 환경 구축

The screenshot displays the AWS Management Console interface for an Amazon ECS cluster named 'BloodRecovery'. The left sidebar shows navigation options for Amazon ECS, including Clusters, Account Settings, Amazon EKS, Amazon ECR, and AWS Marketplace. The main content area shows the cluster details, including its ARN, state (ACTIVE), and a list of tasks. The tasks are listed in a table with columns for service name, status, service type, task definition, and platform version.

클러스터 : BloodRecovery

클러스터에 있는 리소스의 현황을 상세히 볼 수 있습니다.

클러스터 ARN: arn:aws:ecs:us-east-2:709745608741:cluster/BloodRecovery

상태: ACTIVE

등록된 컨테이너 인스턴스: 7

대기 중인 작업 개수: Fargate 0개, EC2 0개, External 0개

실행 중인 작업 개수: Fargate 0개, EC2 7개, External 0개

활성 서비스 개수: Fargate 0개, EC2 7개, External 0개

드레이닝 서비스 개수: Fargate 0개, EC2 0개, External 0개

서비스 작업 **ECS 인스턴스** 측정치 예약된 작업 Tags 용량 공급자

생성 업데이트 삭제 작업 ▼

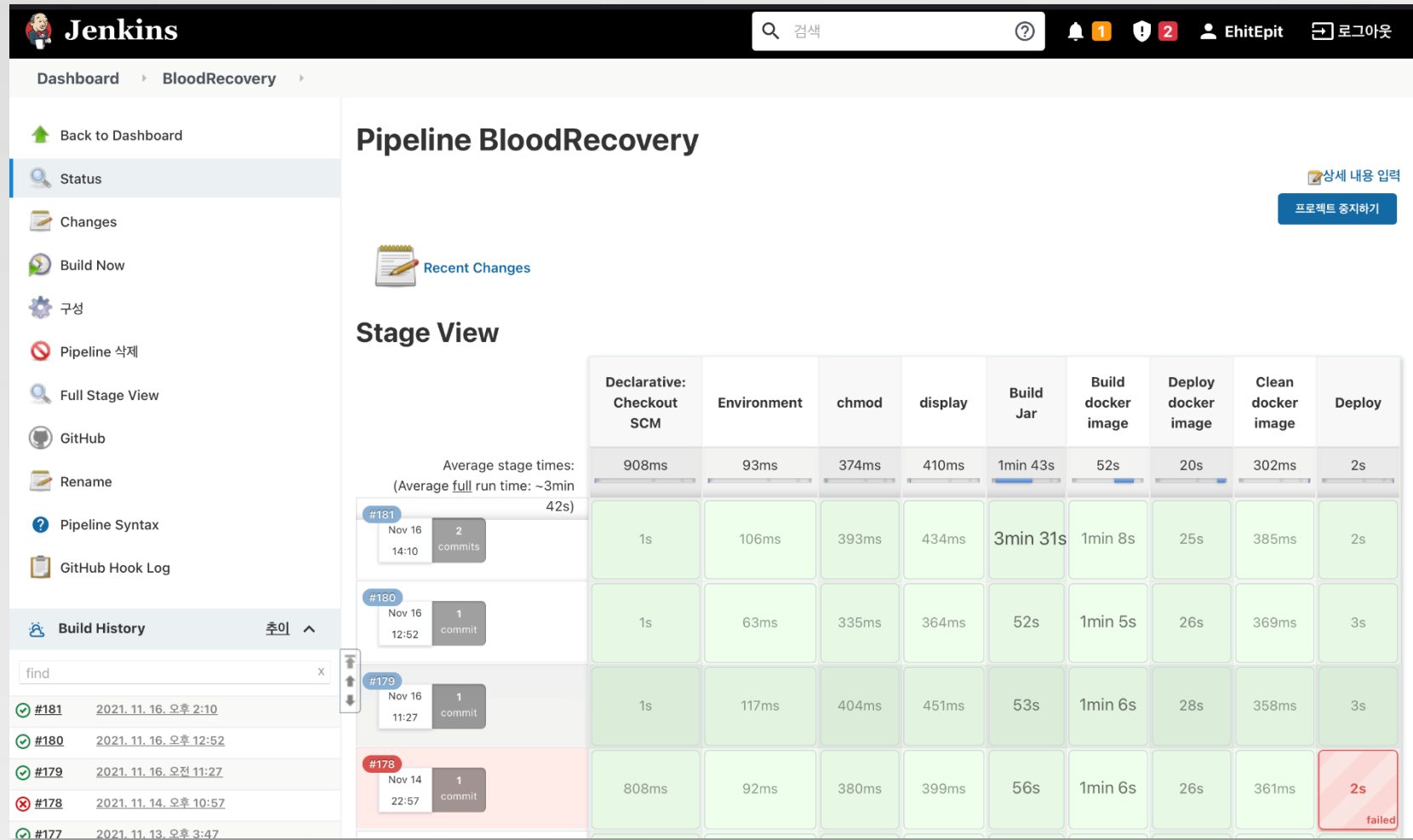
11월 16, 2021 3:09:07 오후 (0분 전에 최종 업데이트함)

서비스 이름	상태	서비스 유형	작업 정의	원하는 작업	실행 중인 작업	시작 유형	플랫폼 버전
Notice-SVC	ACTIVE	REPLICA	NoticeAlarm:2	1	1	EC2	--
React-SVC	ACTIVE	REPLICA	React:2	1	1	EC2	--
Card-SVC	ACTIVE	REPLICA	Card:2	1	1	EC2	--
Rank-SVC	ACTIVE	REPLICA	Rank:2	1	1	EC2	--
User-SVC	ACTIVE	REPLICA	User:2	1	1	EC2	--
Mypage-SVC	ACTIVE	REPLICA	Mypage:2	1	1	EC2	--
Direct-SVC	ACTIVE	REPLICA	Direct:11	1	1	EC2	--



➡ ECS를 통한 배포

■ DevOps 환경 구축



The screenshot shows the Jenkins web interface for a pipeline named 'BloodRecovery'. The 'Stage View' is active, displaying a table of stage execution times for recent builds. The left sidebar contains navigation links like 'Back to Dashboard', 'Status', 'Changes', 'Build Now', '구성', 'Pipeline 삭제', 'Full Stage View', 'GitHub', 'Rename', 'Pipeline Syntax', and 'GitHub Hook Log'. The bottom left shows a 'Build History' table with columns for build number, time, and status.

Declarative: Checkout SCM	Environment	chmod	display	Build Jar	Build docker image	Deploy docker image	Clean docker image	Deploy
Average stage times: 908ms	93ms	374ms	410ms	1min 43s	52s	20s	302ms	2s
(Average full run time: ~3min 42s)								
#181 Nov 16 14:10 2 commits	1s	106ms	393ms	434ms	3min 31s	1min 8s	25s	385ms
#180 Nov 16 12:52 1 commit	1s	63ms	335ms	364ms	52s	1min 5s	26s	369ms
#179 Nov 16 11:27 1 commit	1s	117ms	404ms	451ms	53s	1min 6s	28s	358ms
#178 Nov 14 22:57 1 commit	808ms	92ms	380ms	399ms	56s	1min 6s	26s	361ms

Build History Table:

Build Number	Time	Status
#181	2021. 11. 16. 오후 2:10	Success
#180	2021. 11. 16. 오후 12:52	Success
#179	2021. 11. 16. 오전 11:27	Success
#178	2021. 11. 14. 오후 10:57	Failed
#177	2021. 11. 13. 오후 3:47	Success



Jenkins

➔ 파이프라인을 통한
CI / CD

- DevOps 환경 구축 (형상관리)

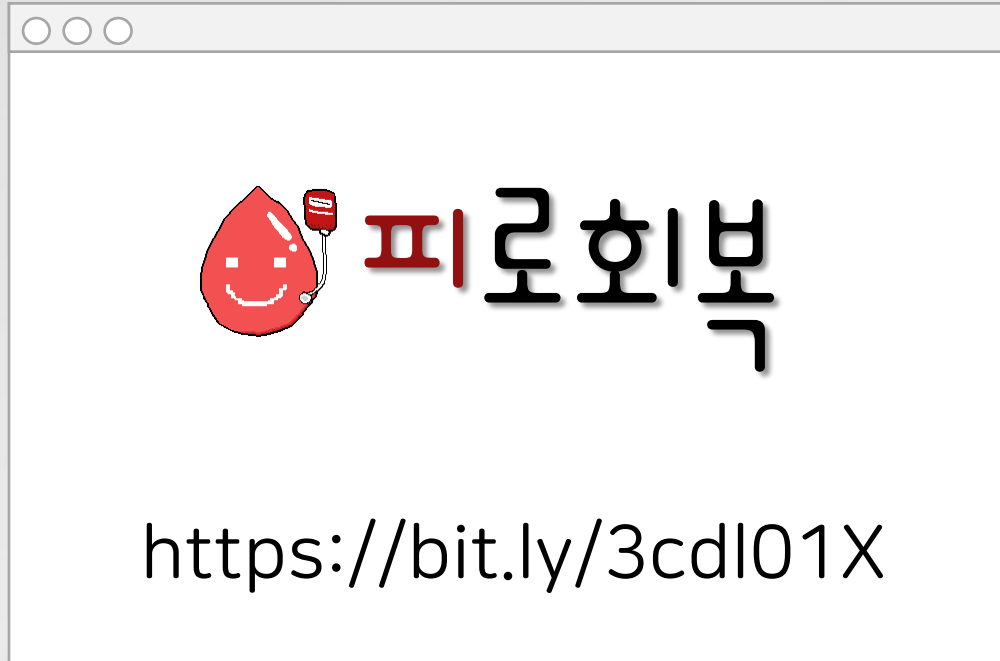
The screenshot shows the GitHub interface for the repository 'EhitEpit / BloodRecovery'. The 'Commits' tab is selected, displaying a commit history graph and a list of recent commits. The graph shows a series of commits branching off from the 'master' branch and merging back. The commit list includes details such as the commit message, the user who made the commit, and the time of the commit.

Commit Message	User	Time
origin React&ApiGateway 현혈의 집, 메시지 css 추가	E	14:06
aws 설정 변		12:34
Merge pull request #121 from 1718390/React&ApiGateway	강	Saturday 17:13
css	1	Saturday 17:11
Merge pull request #119 from santos019/React&ApiGateway	MK	Saturday 14:46
2021-11-13 회원가입수정	S	Saturday 14:46
Merge branch 'ReactApiGatewayDeploy' into React&ApiGateway	MK	Saturday 14:17
Merge pull request #114 from gobongbong/React&ApiGateway	H	Saturday 09:53
Merge branch 'ReactApiGatewayDeploy' into React&ApiGateway		Saturday 09:53
마이페이지,현혈의집 연결 오류 수정	E	Saturday 01:33
Merge pull request #116 from santos019/React&ApiGateway	MK	Saturday 14:06
2021-11-13 회원가입수정	S	Saturday 14:06
Merge pull request #115 from 1718390/React&ApiGateway	강	Saturday 10:32
뒤로가기버튼	1	Saturday 10:02
Merge branch 'EhitEpit:React&ApiGateway' into React&ApiGateway	강	Saturday 09:39
Merge pull request #113 from santos019/React&ApiGateway	MK	Saturday 00:50
2021-11-13 지정현혈 인증 api수정하면 다시할것	S	Saturday 00:49
Merge pull request #112 from gobongbong/React&ApiGateway	H	Friday 23:49
나의 게시물 조회		Friday 22:20
뒤로가기버튼	1	Saturday 09:38
Merge pull request #1 from EhitEpit/React&ApiGateway	강	Friday 20:19
2021-11-13 지정현혈 인증 api수정하면 다시할것	S	Saturday 00:46
나의 게시물 조회	H	Friday 22:02
Merge branch 'React&ApiGateway' into ReactApiGatewayDeploy	E	Saturday 01:18
Merge branch 'React&ApiGateway' into React&ApiGateway	강	Friday 20:19
포인트내역	1	Friday 20:12
Merge pull request #110 from gobongbong/React&ApiGateway	H	Friday 19:59
Merge branch 'React&ApiGateway' into React&ApiGateway		Friday 19:59



소스코드 관리 · 협업 편리

- 사이트 시연



PC로 접속해주세요 :)

게스트 ID: guest1 / guest2 / guest3

게스트 PW: 12341 / 12342 / 12343

■ 회고

김민정

“마이크로 서비스의 설계와 클라우드 응용에 대해 배울 수 있는 프로젝트였습니다.
직접 마이크로 서비스를 설계함으로써 MSA의 확장성, 유연한 배포 등을 직접 체감해 볼 수 있었습니다.
또한 팀원들 간의 협업을 통해 협동심을 기르고, 의사소통하는 방법에 대해 배울 수 있었습니다.”

나청일

“React와 Spring boot로 웹 서비스의 기초를 다지면서 Docker와 AWS를 통한 데브옵스 환경 구축을 배울 수 있는 프로젝트가 되었습니다.
또한 이들을 적절하게 활용하여 MSA의 구조를 익히고 적용해 보면서 클라우드 서비스에 대한 사고의 폭이 넓어졌습니다.
유쾌하신 멘토님, 우수한 팀원들과 같이 프로젝트를 진행하고 마무리하면서, 협업할 때 이전보다 더 나은 모습과 아직 부족한 부분을 볼 수 있었습니다.
프로젝트가 잘 마무리되었다는 것에 성취감을 느낄 수 있는 시간이 되었습니다.”

강혜수

“직접 API를 설계하고 구현 및 적용해 봄으로써 Jpa 내부 동작 방식을 이해하고 기본기를 다질 수 있었습니다.
또한 멘토님의 가르침에 따라 실무와 비슷하게 마이크로 서비스를 설계함으로써 MSA의 장점인 서비스 단위로 독립적인 개발, 빌드, 테스트, 배포를
직접 배울 수 있었습니다. 백과 프론트 둘 다 경험해 보고 나니까 어느 한 쪽이 중요한 것이 아닌 양쪽 모두가 중요하다는 것을 알게 되었습니다.
설계와 팀원 간 협업이 얼마나 중요한지 깨닫게 해주는 프로젝트였습니다.”

최현영

“이번 프로젝트를 통해서 클라우드 네이티브 애플리케이션 개발자 양성 과정의 전반적인 내용을 다시 되돌아볼 수 있어 유익한 경험이었습니다.
JPA에 대한 이해력을 높일 수 있었고 익숙하지 않았던 React 기술과 MSA 경험을 쌓을 수 있었습니다.
무엇보다 협업의 필요성에 대해 배우게 되었습니다. 좋은 결과물이 나오기 위해서는 혼자만의 역량뿐만 아니라 팀원 간의 원활한 의사소통과 팀의 분
위기가 중요하다는 것을 깨닫게 되었습니다.”

Q&A



피로회복

THANK YOU :)

