

온라인 다마고치 서비스 기획서 (초안)

버전: v0.4 | 기준일: 2026-02-13

0. 한 페이지 요약

목표: 웹/모바일 브라우저에서 즐기는 온라인 다마고치형 육성 서비스. 계정 기반으로 크리처(가칭)를 육성하고, 배틀/합체/진화를 통해 성장시키는 경험 제공.

핵심 루프: 돌보기(먹이/치료/취침) -> 상태 유지/성장 -> 배틀/합체 -> 진화/수집 -> 반복.

MVP 우선순위: 시간/상태 계산의 일관성(오프라인 포함), Active/Backup 전환 정책, 배틀 운영/조작 방지 기초.

1. 용어 및 명칭 제안

현재 “몬스터”라는 임시 명칭은 의미 전달이 좋지만, 서비스 톤(돌봄/육성)과 전투/합체 요소를 함께 담기 위해 아래 대안을 추천한다.

- 크리처(Creature):** 가장 범용적이며 육성/전투/합체 모두 어울림. 개발/문서 용어로도 간결.
- 펫(Pet):** 돌봄 톤은 강하지만 배틀/진화와 결합 시 느낌이 약할 수 있음.
- 에그(Egg) / 유닛(Unit):** 수집형/시스템 톤. 감성보다는 기능 중심 서비스에 적합.

본 문서에서는 “크리처”를 기본 용어로 사용한다.

2. 서비스 개요

서비스는 회원 로그인 기반이며, 계정 단위로 최대 3 마리까지 크리처를 보유한다. 동시에 육성 가능한 크리처는 1 마리(Active)이며 나머지는 Backup으로 보관한다(추후 다중 Active 확장 가능).

크리처는 시간 경과에 따라 나이가 증가하고, 돌봄 행동과 이벤트(질병/부상, 배틀 결과)에 따라 상태가 변화한다. 일정 조건에 따라 성장/진화하며, 두 크리처를 합체해 새로운 크리처를 생성할 수 있다.

3. 크리처 상태 모델

크리처는 아래 상태 지표를 가진다. 각 지표의 단위/범위는 MVP에서 단순화하고, 이후 밸런싱 가능하도록 서버에서 파라미터화한다.

상태	의미	권장 범위(예)	비고
나이	시간 경과에 따른 성장 단계/진화 조건	0~무제한	취침 중에도 증가

건강	체력/면역/부상 영향의 기반 수치	0~100	0 유지 시 컨디션 하락 유발
배고픔	먹이 필요도	0~100	0 유지 시 컨디션 하락 유발
컨디션	전반적 상태(행동 성공/배틀 성능/사망 위험)	0~100	건강/배고픔 0 유지 시 하락
승률	배틀 기록 기반 지표	0~100 또는 승-패	랜덤배틀 기준 반영 권장

상태 변화는 “이벤트 기반(Event Sourcing 유사)” 또는 “누적 계산(Lazy Evaluation)” 중 한 방식으로 일관되게 설계한다. MVP에서는 누적 계산이 운영/성능 측면에서 단순할 수 있다.

4. 시간/낮-밤/취침 규칙

핵심은 “서버 기준 시간”으로 일관된 규칙을 제공하는 것이다. 사용자별 로컬 시간은 UI 표시용으로 두고, 게임 규칙은 서버 시간(예: Asia/Seoul 고정)으로 통일하는 방식을 추천한다.

4.1 낮-밤

예시 규칙(조정 가능):

- 낮: 07:00~22:59 (KST), 밤: 23:00~06:59 (KST)
- 밤 시간대에는 취침 상태가 권장되며, 밤에 장시간 미취침 시 컨디션 하락 또는 질병 확률 증가 같은 페널티 부여 가능

4.2 취침

- 취침 시작 시점부터 “나이”만 증가하고, 건강/배고픔/컨디션은 변하지 않는다(요구사항 반영).
- 취침 중 “잠깐 깨움”을 허용하되, 연속 깨움 악용 방지를 위해 쿨다운(예: 10 분) 또는 1 회 행동 제한을 둔다.
- 밤 시간대에 Active 크리처를 취침시키지 않으면(또는 일정 시간 이상 깨어있으면) 컨디션 감소/질병 확률 증가를 적용할지 정책 확정 필요.

4.3 오프라인(미접속) 처리

서비스가 온라인이므로 “사용자가 접속하지 않는 동안에도 시간은 흐른다”는 전제를 추천한다. 구현 옵션:

- 옵션 A - 주기 스케줄링(tick): 서버가 분 단위로 상태를 갱신
- 옵션 B - 요청 시 누적 계산(lazy): 마지막 갱신 시각과 현재 시각 차이로 상태를 계산 후 저장

MVP 권장: 옵션 B. 대량 유저에서도 스케줄러 부하를 줄이고, 데이터 정합성 디버깅이 쉬운 편이다.

5. 상태 하락 및 사망 규칙

5.1 건강/배고픔 0 유지 시 컨디션 하락

요구사항: 건강과 배고픔이 0 인 상태로 일정 시간 이상 유지되면 컨디션 수치가 감소한다.

정책 초안(예시):

- 건강==0 또는 배고픔==0 상태가 연속 T 분 이상이면 컨디션 -X/분
- 건강==0 AND 배고픔==0 이면 하락 가속(예: -2X/분)
- 컨디션이 낮을수록 질병/부상 발생 확률 증가 또는 배틀 성능 페널티

5.2 질병/부상

질병/부상은 상태 이상(디버프)으로 모델링한다. 치료하지 않고 방치하면 사망 위험이 증가한다(요구 사항 반영).

MVP 모델 추천:

- 상태 이상 타입: 질병(감염), 부상(외상), 피로(수면 부족)
- 각 상태 이상은 severity(경도/중등도/중증), 시작 시각, 마지막 악화 시각을 가진다
- 치료 행동(약/치료키트)으로 제거 또는 severity 감소

5.3 사망

사망은 서비스 이탈을 유발할 수 있으므로 정책을 명확히 한다. 선택지:

- 영구 사망(하드코어): 수집/육성 긴장감은 높지만 이탈 위험 증가
- 부활(소프트): 일정 재화 또는 긴 쿨다운으로 부활 가능
- 리셋: 사망 시 같은 종의 초기 단계로 회귀(소유 슬롯 유지)

MVP 권장: “리셋 또는 제한적 부활”. 운영 안정화 후 하드코어 모드 도입도 가능.

6. 성장/진화/합체

6.1 성장/진화

- 나이와 상태(컨디션, 승률, 질병 여부 등)에 따라 성장 단계가 변한다.
- 진화 조건은 서버 설정값으로 관리하여 밸런싱을 쉽게 한다(예: age>=A AND condition>=B AND winrate>=C).

6.2 합체

요구사항: 크리처끼리 합체해서 새로운 크리처가 될 수 있다.

정책 초안(예시):

- 합체는 두 크리처를 소비(삭제)하고 신규 크리처를 생성

- 합체 결과는 조합표(레시피) 또는 확률형(가중치) 중 선택
- 악용 방지를 위해 합체 가능 조건(나이, 컨디션 최소치, 쿨다운)을 둔다

7. 배틀

요구사항: 랜덤배틀, 방 생성 배틀 모두 제공. 채팅은 제공하지 않음.

7.1 배틀 타입

- 랜덤 배틀: 매칭 시스템을 통해 유사 전력/티어로 매칭. 승률 지표 반영의 주 대상.
- 방 배틀(친구 대전): 초대/코드 기반 입장. 승률 반영은 별도 지표로 분리하거나 가중치를 낮춰 조작을 줄인다.

7.2 전투 방식(MVP 추천)

- 자동 전투(서버 시뮬레이션)로 시작: 실시간 입력/동기화 부담이 적고, 공정성 검증이 쉬움
- 전투 결과는 “결정적”이어야 함(같은 입력/시드에서 같은 결과) - 재현 가능 로그가 운영에 유리

7.3 승률 반영

- 승률은 전적(승/패/무) 또는 최근 N 전 기반 가중 평균 중 선택
- 연속 패배 보호(매칭 완화) 또는 컨디션 낮을 때 패배 페널티 완화 같은 UX 정책 고려

8. 계정/육성 슬롯/Active-Backup

요구사항: 로그인 필수. 계정당 최대 3 마리 보유. 동시에 1 마리만 육성(Active), 나머지는 Backup.

악용 방지 관점에서 아래 정책을 권장한다:

- Active <-> Backup 전환 쿨다운(예: 30 분) 또는 일일 전환 횟수 제한
- 위험 상태(건강 0, 컨디션 급감)를 회피하기 위한 전환 남용 방지
- Backup 은 상태 변동이 없거나(완전 동결) 최소한으로만 진행되도록 정책화

9. 클라이언트/아트 방향

디자인: 도트 아트 스타일. 크리쳐 스프라이트(진화 단계별), 애니메이션(Idling/취침/배틀) 최소 세트 정의.

웹 클라이언트: React 기반 SPA 를 추천. 상태 표시, 행동 버튼, 알림(밤 취침 권장, 질병 발생)을 포함.

실시간성: 배틀 매칭/진행은 WebSocket, 단순 알림은 SSE 또는 폴링으로 단계적 도입 가능.

10. 비기능 요구사항(운영/대규모 트래픽)

- 정합성: 상태 계산은 서버가 단일 소스 오브 트루스. 클라이언트는 표시/요청만 수행.
- 확장성: 배틀/매칭/상태 계산을 모듈로 분리. 트래픽 증가 시 독립 스케일링 가능하도록 설계.

- 캐시: 자주 조회되는 크리처 상태는 Redis 캐시를 사용하되, 소스는 RDB(또는 이벤트 로그)로 유지.
- 관측성: 행동/상태 변경/배틀 결과를 이벤트 로그로 남겨 CS 및 치트 대응.
- 보안: JWT 기반 인증, 요청 서명/레이트리밋, 배틀 조작 방지(클라 결과 제출 금지).

11. 데이터 모델(초안)

RDB 기준 엔티티 초안(예시). 실제 구현에서는 정규화/성능을 고려해 조정한다.

엔티티	주요 필드	설명
User	id, email, password_hash, created_at	회원
Creature	id, user_id, name, species, stage, is_active, created_at	크리처 기본 정보
CreatureState	creature_id, age, health, hunger, condition, winrate, updated_at	상태 스냅샷(마지막 갱신 시각 중요)
StatusEffect	id, creature_id, type, severity, started_at, updated_at	질병/부상/피로 등 상태 이상
Battle	id, type(random/room), started_at, ended_at, seed, result	배틀 세션
BattleParticipant	battle_id, creature_id, user_id, snapshot_json, outcome	참여자/전투 당시 스냅샷
ActionLog	id, user_id, creature_id, action_type, payload_json, created_at	행동/상태 변경 감사 로그

12. API 설계 초안(REST 기준)

인증은 JWT 를 가정한다. 배틀 진행/매칭은 WebSocket 또는 별도 매칭 API 로 확장 가능.

Method	Endpoint	설명
POST	/auth/signup	회원가입
POST	/auth/login	로그인(JWT 발급)
GET	/creatures	내 크리처 목록(Active/Backup 포함)
POST	/creatures	크리처 생성(슬롯 한도 체크)
POST	/creatures/{id}/activate	Active 전환(콜다운/정책 체크)
POST	/creatures/{id}/actions/feed	먹이 주기
POST	/creatures/{id}/actions/sleep	취침 시작/종료 또는 토글
POST	/creatures/{id}/actions/treat	치료(질병/부상 제거/완화)

POST	/battles/random/queue	랜덤 배틀 큐 참가
POST	/battles/rooms	방 생성
POST	/battles/rooms/{code}/join	방 참가
GET	/battles/{id}	배틀 결과 조회

13. MVP 범위 및 로드맵

13.1 MVP(1 차 출시)

- 회원가입/로그인, 크리처 생성/조회, Active/Backup 전환
- 상태 계산(오프라인 포함) + 기본 돌봄(먹이/취침/치료)
- 질병/부상 최소 1~2 종
- 배틀: 랜덤 배틀 자동 전투 + 결과 기록(승률 반영)
- 합체/진화: 최소 1 단계(또는 합체만 먼저)

13.2 2 차(확장)

- 방 배틀 정식 지원 + 승률 반영 정책 세분화
- 다중 Active 육성(동시 2~3 마리) 옵션
- 아이템/상점/퀘스트/일일 보상
- 랭킹/시즌, 전적 통계

14. 참고 링크(공식 문서)

- Spring Boot: <https://spring.io/projects/spring-boot>
- Kotlin: <https://kotlinlang.org/docs/home.html>
- Spring Security: <https://spring.io/projects/spring-security>
- Hibernate/JPA: <https://hibernate.org/orm/documentation/>
- PostgreSQL: <https://www.postgresql.org/docs/>
- Redis: <https://redis.io/docs/>
- React: <https://react.dev/learn>
- WebSocket (RFC 6455): <https://www.rfc-editor.org/rfc/rfc6455>