Owlet

MiniGolf

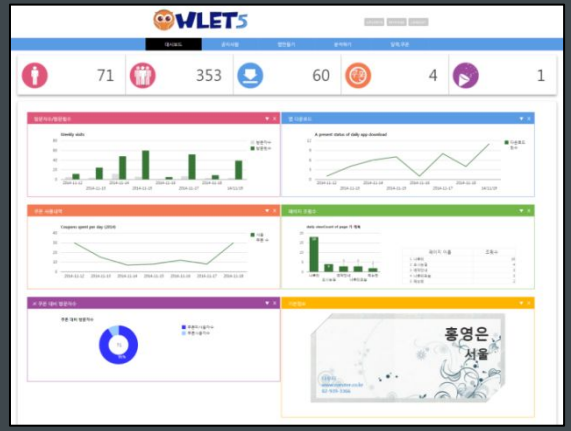
Products

Owlet

Ichatt

Noriter

BetterBrain



개요

웹앱 기반의 정보제공을 목적하는 앱을 누구나가 쉽게 저작할 수 있는 저작 도구 서비스

웹 기반의 관리자 Editor에서 사용자가 글 쓰듯이 화면을 제작하고 모바일 또는 웹 방식으로 제장 방식을 택하면 각각 모바일 그리고 웹 형태로 앱이 도출되는 시스템

목적

- 손쉽게 앱을 제작하여 마케팅에 활용
- 앱 사용 내역 분석하여 홍보현황 파악
- 다양한 컨텐츠 활용 기능을 통해 홍보 방향 확장

MiniGolf

Products

Owlet

Ichatt

Noriter

BetterBrain

개발환경

형상관리	SVN
개발 언어	Javascript, Java, HTML
개발 도구	Eclipse Kepler, Aptana
Database	MySQL
이슈 관리 도구	Redmine

역할 및 이용기술

총 참여인원 5명
클라이언트 담당, 서버 통신 담당
대시보드, 사진 업로더, 지도 업로더, 통계 페이지, 공지사항 개발

Products

MiniGolf

Owlet

Ichatt

Noriter

BetterBrain

기술상세

Full calendar API
- 달력 뷰 및 일정관리 기능을 제공하며 사용자정의(customize)가 용이한 오픈 소스 API 간단하여 쿠폰과 일정에 관한 기능에 활용.

Google Map API
- 핀 꽂기 및 위치 찾기 등의 기능을 개발해야 했는데 대부분이 지원되어 위치 서비스에 활용

CKEditor
- 웹 오픈 소스로 편집기를 제작할 수 있음. 상당히 많은 고급 기능이 제공되어 채택. 사용자 정의하여 사용.

HTML & JSP
- Client의 View로 사용. HTML5/CSS3/JS로 semantic 구조로 작성.

Spring Framework 3 with MVC
- 웹개발 서버 프레임워크로 사용. IOC, AOP 개념으로 가독성 높고, 유지보수 용이한 서버 개발

Google Chart, Rapheal
- 사용자 통계를 보여주기 위한 차트 API로 채택하여 사용. D3는 제외됨.

Ddslick, drag/drop.js, Modernizer, bootstrap
- 클라이언트 단에서 높은 사용자 경험을 위해 사용. 드래그, 플랫 디자인 등 제작에 적합하다 판단.

MiniGolf

Products

Owlet

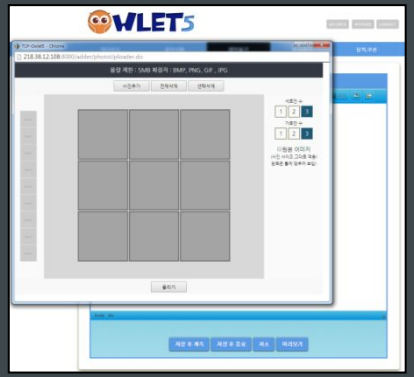
Ichatt

Noriter

BetterBrain

스크린샷

주 화면, 통계 정보를 제공.



글 작성 화면, 동시에 사진 업로드 화면.

상세 통계 화면

모바일 작성 화면

MiniGolf

Products

Owlet

Ichatt

Noriter

BetterBrain

고찰

사용자가 컴퓨터에 익숙하지 않음이 전제되었기에 이를 고려한 UX 설계가 핵심 고민

사용자가 기능을 사용하는데 문제가 있어서는 안 되었음. 이에 간단한 템플릿 구조를 기반으로 원하는 내용을 작성할 수 있도록 요소 배치 및 이미지 및 직관적 아이콘과 설명 텍스트를 정하는데 주 시간 소요.

또한 편의를 제공하기 위한 코드를 작성하는데 시간 소요. 드래그를 사용한 설정, 상황에 따른 적절한 도움말, 문제가 될 경우 원인의 상세하면서도 쉬운 설명과 해결을 유도하는 방향 제시를 진행. 난잡하지 않도록 고려.

사진 업로더의 경우, 자유롭게 격자형으로 배치할 영역과 칸수를 지정하고 보여질 이미지를 즉시 알려주도록 설계.

필요한 통계 데이터를 고심하여 축적하고 이를 요약해 보여주는 대시보드 페이지와 상세 조회가 가능한 통계 페이지를 제공

Web 개발 이해도 향상 및 Redmine을 통한 이슈 관리의 효용성 및 이용능력 획득

Agile 개발 방법론 진행.

ICHATT

개요

캐릭터 기반의 소통형 메신저.

모바일에서 아바타를 중심으로 한, 상호 통신을 진행하는 캐릭터 기반의 채팅 서비스. 부가적으로 아바타 꾸미기, 카페 관리, 아이템 구매 등 아바타와 관계된 기능 탑재. 캐릭터를 이용하여 텍스트나 일반 이모티콘으로 전달하기 힘든 부분 커버.

개발 중심

- 앱의 완성도 높임
- 부가 기능(미개발 기능) 추가
- 최적화 작업, 버그 수정

MiniGolf

Owlet

Products

Ichatt

Noriter

BetterBrain

개발환경

형상관리	Git
개발 언어	Java 8
개발 도구	Eclipse Mars, Android Studio
Database	MySQL, MariaDB(AWS)
이슈 관리 도구	Redmine
플랫폼	Android, Linux Server(AWS)
협업 도구	Slack

역할 및 이용기술

총 참여인원 4명
각자가 동시에 진행하는 타 업무에 맞추어 이슈를 정하고 분담하여, Android 버전의 앱과 해당 서버의 고도화 작업 진행. 특정 외주 개발사가 만든 코드를 받아온 것으로, 이어 받아 최적화하고 추가 개발 불필요한 기능 제거, 인앱 결재 추가, 통신 구조 변경/삭제, 미세 버그 수정, 데이터베이스 쿼리 수정 등

MiniGolf

Owlet

Products

Ichatt

Noriter

BetterBrain

기술상세

Spring Framework 4
- 웹서버 프레임워크로 채택, 적용

Spring JPA
- Hibernate, Mybatis 혼용된 것을 교체하는 데 사용. QueryDSL과 같이 이용

Spring Mockito
- 테스트 프레임워크로 사용. 예상 가능한 입력 시나리오와 결과 시나리오 테스트

Gradle
- 라이브러리 의존 테스트 및 자동 태스크 수행

AWS
- Lambda, EC2 등 서버 운용을 위해 사용

Message queue services
- Mqtt로 ActiveMQ를 이용하여 Push 알림에 사용

Google protobuf
- 안드로이드 앱과 서버간 통신에 사용. Json 대신 byte단위로 통신량을 줄이고자 함

Logstash
- 서버 로그를 수집하기 위하여 사용. AWS S3 에 저장.

Google Developer API
- 앱 정보 조회 및 인앱 아이템 관리를 위해 사용, 인앱 결제 시스템 개발

서버 구성(초기)

MiniGolf

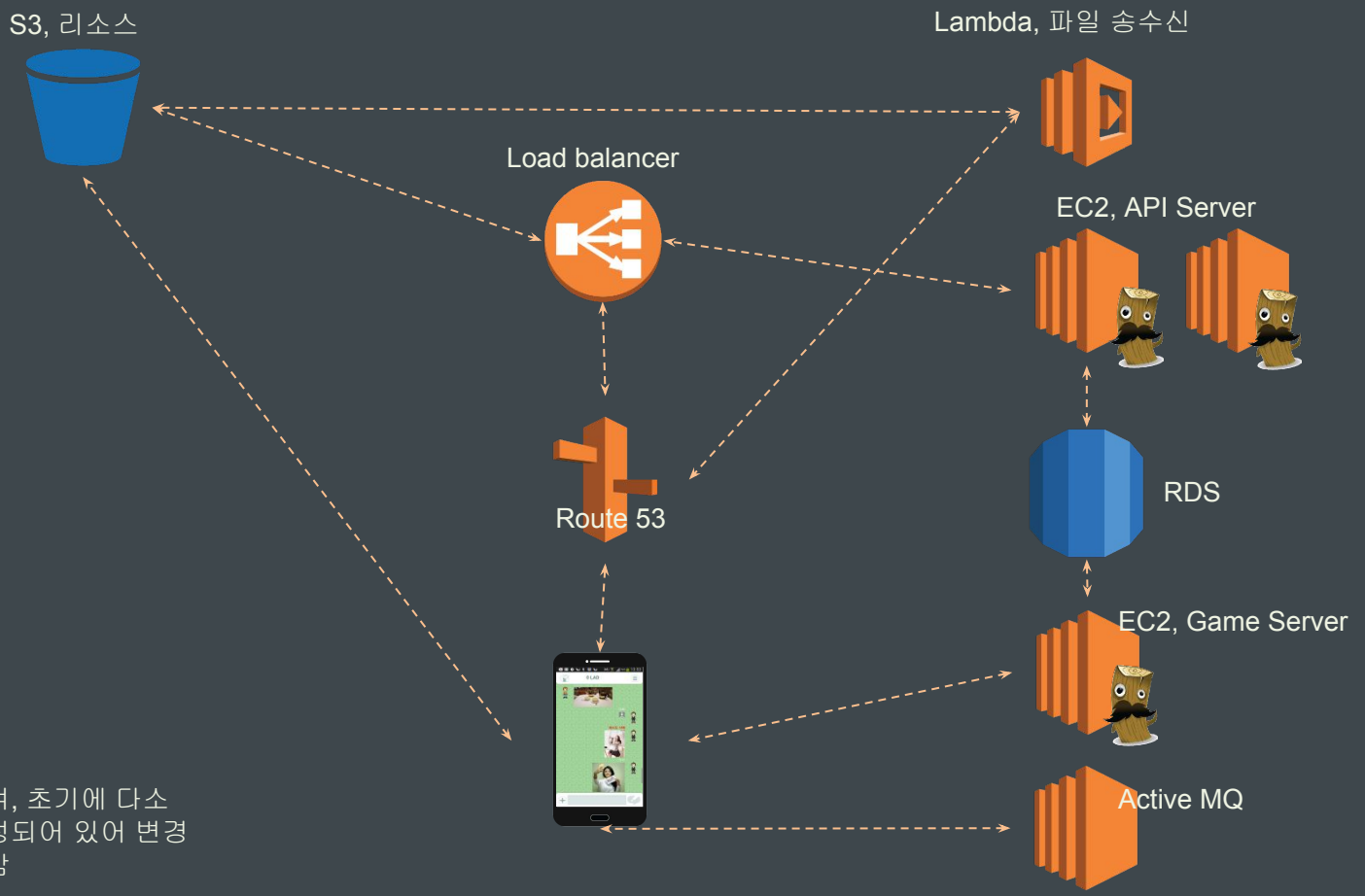
Owlet

Products

Ichatt

Noriter

BetterBrain



최종이 아니며, 초기에 다소 복잡하게 구성되어 있어 변경 작업에 들어감

작업 상세(팀 포함)

MiniGolf

Owlet

Products

Ichatt

Noriter

BetterBrain

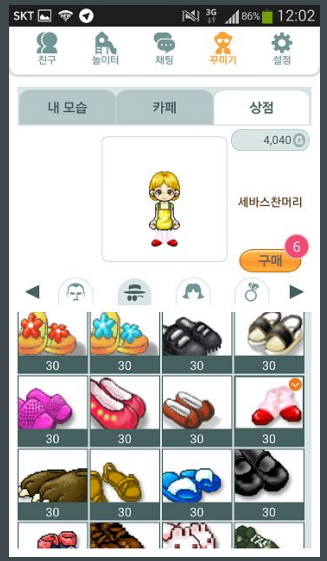
- Game Server든 Api Server든 여기저기서 DB에 접근하는 구조에서 API 서버 단일 접근으로 변경
- AWS SNS 서비스를 이용하며 JMS 자체 포맷을 이용하여 통신하고 있던 부분 제거
- 한 개 API Server 내에서 복수의 ORM 프레임워크를 사용하던 것을 Spring JPA, QueryDSL로 통합
- Protobuf의 큰 오브젝트로 모든 통신을 하여 데이터 소모가 상당한 부분을 간소화
- 불필요한 데이터 통신 제거
- 통신 프로토콜 간소화를 통한 Active MQ에서 Qos2를 Qos1로 변경
- Logstash를 이용, S3에 서버 로그 저장
- Google InApp Purchase 기능 추가
- 게시판 기능 추가
- AWS 리전을 Tokyo에서 Seoul로 이전
- Seoul Region에서 Lambda가 지원되지 않아, Lambda 기능 대체
- SNS 서비스 사용 부분 일절 제거
- 앱 자체의 UX 관련 이슈 수정
- ‘더보기’를 통한 기능 확장의 틀 마련
- OAuth 2.0 을 통한 Google 로그인 기능 추가
- 테이블 구조 수정
- SQL 스크립트 관리 시작
- 푸시 구조 개선
- 기타 자잘한 코드 개선
- 데드 코드 제거.
- Spring mockito 테스트 프레임워크 적용.
- 쿠폰 시스템 추가.
- Version 1.0 앱 런칭

스크린샷

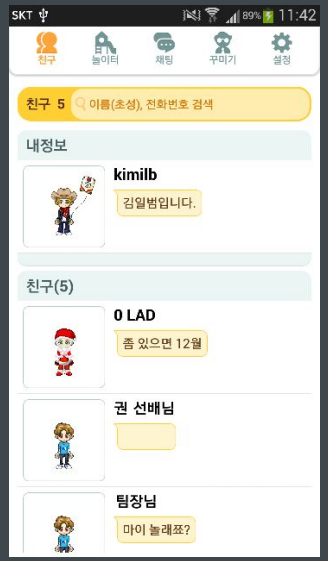
MiniGolf
Owlet
Products
Ichatt
Noriter
BetterBrain

캐릭터 채탱 화면
(액션 취하기 가능)

텍스트 채팅 화면



꾸미기/상점 화면



친구 화면

설정 화면

MiniGolf

Owlet

Products

Ichatt

Noriter

BetterBrain

고찰

코드의 가독성과 잘 정의된 명명 규칙 및 성실한 준수가 큰 도움이 됨을 확인.
개인의 잘못된 코드로 인해 다른 사람들이 고생하는 것을 경험하여 개인의 책임 중요성 크게 확립.
구조적 문제로 인하여 수정할 때 사이드 이펙트의 영향을 확인, 철저한 코드 이력 관리 필요성 파악.

초기 Rabbit MQ를 통한 Push를 진행하고자 하였으나 QOS 1 까지만 지원되어 문제에 봉착.
이에 통신 구조와 메시지 내용을 대폭 수정했는데 그 규모가 재개발 수준이기에 초기 설계 중요성 확인.

중복된 코드와 데드 코드의 존재로 시간을 많이 빼앗겨 주석과 깨끗한 코드 작성기술에 대해 고찰함.

Redmine과 slack 및 git을 통한 협업의 효율과 코드 관리의 효과 경험.

코드 명명법 및 작성 전반에 대한 규칙 제정으로 가독성 향상.
타인의 코드 리뷰 및 분석에 드는 시간 감소.

처음 접하는 기술에 대해 주 1회 1~2시간 스터디 및 회의을 갖고 학습 커버리지 낮춤.