

===== 1번 슬라이드 =====

안녕하세요 1조의 이재현, 홍혁기, 발표를 맡게된 이윤후 입니다.

저희 조는 "AWS를 활용한 풋살 구장 예약 및 매칭 서비스" 를 만들어보고자 하였으며,
웹과 모바일환경을 지원하도록 하였습니다.

발표 시작하겠습니다.

===== 2번 슬라이드 =====

저희가 준비한 목록입니다.

서비스 사용 방법, 시연 영상으로 간단한 소개 이후
시스템 구성 및 기능에 대해 설명하도록 하겠습니다.

===== 3번 슬라이드 =====

먼저 프로젝트 개요 입니다.

앞에 보이는 두 사진은 서로 다른 사이트에서 예약과 매칭할 팀을 구하는 과정입니다
해당 과정에서 저희 조가 파악한 불편함은 총 3가지 입니다.

매칭 시스템 부재로 예약 이후 따로 매칭 커뮤니티에서 상대를 구해야 하는 점

매칭 이후에 구장 이용료 정산에 대해 논의해야 하는 점

상대에게 직접 팀의 정보를 물어봐야 하는 점이 있었습니다.

===== 4번 슬라이드 =====

이러한 불편함을 개선하기 위해

이용자에게 편의를 제공하는 쿼리 메뉴 바

상대팀에 대한 구체적인 정보 제공

상대 확정 시 이용료 정산 자동화

위치 정보 제공 동의 시 가까운 구장 정보 제공

매칭 관련 알림 전송을 지원 하도록 설계하였습니다.

===== 5번 슬라이드 =====

다음은 서비스에 적용된 매칭 시스템 입니다.

이용자가 구장을 예약할 때 구장을 전부 예약 할 지, 부분으로 예약할 지 선택지를 제공합니다.

부분 대여로 예약 시, 다른 팀이 도전 신청을 진행하게 되면 해당 팀에 대한 정보가 우리 팀에게 알림으로 전송되고

또한, 우리팀이 매칭을 수락하게 되면 양 팀에게 매칭이 확정되었다는 알림이 전송되도록 하였습니다.

===== 6번 슬라이드 =====

서비스 사용 방법에 대해 설명드리도록 하겠습니다.

보고계신 페이지는 구장을 선택 했을 때 보여지는 페이지 인데요.

날짜, 시간 선택에 따라서 해당 구장의 날씨 정보를 제공하고 있으며

직관적으로 예약 가능한 시간을 파악 할 수 있도록 색상을 구분하였습니다.

매칭 대기 중인 시간을 선택할 경우 상대팀에 대한 구체적인 정보를 제공합니다.

===== 7번 슬라이드 =====

처음 구장을 예약하는 팀은 구장 이용료 만큼의 포인트를 지불하게 되지만,

부분 대여로 예약 했을 경우, 도전 신청을 진행한 팀에 대한 정보를 토대로 상대 팀을 선택할 권한이 주어지고

===== 8번 슬라이드 =====

원하는 상대를 선택하면 해당 팀에게서 구장 이용료의 절반 만큼 포인트를 되돌려 받도록 설정하였습니다.

상대 팀 선택을 하게 되면

===== 9번 슬라이드 =====

양 팀 모두에게 웹, 모바일 환경에서 볼 수 있는 알림을 전송하도록 하였습니다.

===== 10번 슬라이드 =====

다음 으로 시연영상을 보시겠습니다.

===== 11번 슬라이드 (시연 영상) =====

먼저 웹에서 예약하는 과정을 보여드리도록 하겠습니다.

웹에서 로그인 시, 카카오, 네이버, 구글 소셜로그인을 지원하고 있습니다.

(영상 실행 - 카카오 로그인 이후 메인에서 내 위치 확인 보일 때 멈춤)

이용자가 위치 정보 제공을 동의한 경우에 근처 구장 정보를 제공하도록 하였습니다.

(영상 실행 - 꿀벌Fs 시 멈춤)

원하시는 날짜를 선택하게 되면 해당 날짜에 예약가능한 시간을 색상별로 표현하였습니다.

시간에 따라 최대 5일이내의 날씨 정보를 제공받으실 수 있으며

매칭 대기중인 시간을 선택할 경우, 상대팀에 대한 정보 확인이 가능합니다. (말 끝나면 영상 시작)

지도보기를 통해 해당 구장의 위치와 근처 구장들에 대한 정보도 확인 가능합니다.

이후 예약 페이지에서 구장을 전부 예약 할 지, 부분으로 예약할 지 선택할 수 있습니다.

(게시판 가면 말해야됨)

기존의 커뮤니티 기능 또한 제공하고있어 예약 정보를 담은 글을 작성하고

글을 확인하는 다른 이용자들이 쉽게 예약할 수 있도록 설계하였습니다.

(글 작성 이후 메인에서 알림 창 띄운 후 멈춤)

다음으로 모바일에서 도전 신청하는 과정을 보여드리도록 하겠습니다.

모바일 환경 또한 소셜로그인 및 위치 기반 서비스를 제공하고 있습니다.

(영상 실행 - 모바일 커뮤니티 진입 후 멈춤)

게시글에서 상대 팀 정보를 파악하고 도전 신청도 가능하지만,

시연 과정에서는 구장에 들어가서 직접 도전 신청하는 과정으로 진행하겠습니다.

(영상 실행 - 모바일 메인에서 검색버튼 이후 지도화면에서 멈춤)

구장 검색을 진행하면, 최근에 검색한 목록들을 저장하고

검색어 입력 시 구장 리스트를 제공합니다.

검색 결과에서는 가장 가까운 구장을 위주로 나타나게 하였습니다.

(영상 실행 - 모바일에서 구장에 진입 후 멈춤)

웹과 마찬가지로 날짜, 시간 선택에 따라서 정보를 제공 받으실 수 있습니다.

앞에 예약된 구장에 도전 신청을 진행하여 알림을 보내도록 하겠습니다.

(영상 실행 - 모바일에서 도전 신청 후 멈춤)

도전 신청 시, 부분 대여를 진행했던 이용자에게 알림을 전송하며

매칭 상대를 수락할 경우, 양 팀인원 모두에게 매칭이 확정되었음을 알려주도록 하였습니다.

(영상 실행 - 모바일에서 알림 카운트 1 올라가는거 보이면 멈춤)

모바일에서 알림 확인시 앱이 실행되도록 설정 하였으며, 자동로그인 되도록 하였습니다.

또한 나의 예약에서 매칭이 완료됨을 확인할 수 있습니다.

-- 영상 끝 --

===== 12번 슬라이드 =====

다음은 저희 프로젝트의 시스템 구성 및 기능에 관한 설명입니다.

저희는 Python 의 Selenium, BeautifulSoup 라이브러리를 활용하여 크롤링 한 데이터를 pandas로 가공해 약 1600개의 구장 데이터를 구축하였습니다.

그리고 스프링부트의 RestTemplate을 사용하여 OpenWeatherAPI에 날씨 데이터를 가져오고 Scheduler annotation을 이용해 이를 주기적으로 갱신하도록 하였습니다.

===== 13번 슬라이드 =====

다음은 알림 기능에 관한 설명입니다.

우선 웹의 경우, Spring에서 지원하는 구독/발행 방식의 stomp프로토콜을 사용하여 이용자가 로그인하는 순간부터 알림을 받아야 하는 이벤트를 구독하고 있도록 하였습니다. 이용자가 누구인지 알 수 있게 하기 위해 interceptor를 구현하여 웹소켓 세션에서 http세션을 접근할 수 있도록 하였고, 모바일에서도 웹으로 알림을 보낼 수 있게끔 기존의 send to 어노테이션을 사용하는 것에서 spring에 simple message template으로 메시지 전송 방법을 변경하여 구현하였습니다.

모바일의 경우, 구글의 FireBase Cloud Messaging을 사용하여

앱에서 로그인 시 회원 DB에 FireBase token을 저장하고, 서버 내에 알림을 보내야 하는 이벤트가 발생 시

해당 토큰에 알림을 보냄으로써 앱 화면 상단에 알림이 노출될 수 있도록 하였습니다.

하지만 위의 방법으로는 앱이 백그라운드 상태일 경우에만 알림이 올리기 때문에,

앱이 포그라운드 상태일 경우에도 알림을 주기위해 앱 내의 broadcast receiver가 요청을 받았을 때

사용자가 현재 앱을 사용중인지 판별하여 알을 구성하도록 하였습니다.

===== 14번 슬라이드 =====

저희 서비스는 회원의 위치 정보 제공 동의를 받아, 가까운 구장들을 보여주고 있습니다.

모바일의 경우, 위치권한 사용을 동의 받은 후에 내장된 GPS를 통해 사용자의 현 위치를 가져오도록 하였으나

웹의 경우, google의 Geolocation API 의 사용조건이 HTTPS와 같은 secure context 였기에 변환 작업이 필요했습니다.

그래서 nginx를 사용하여 80 포트 변환

route 53을 사용한 dns 설정

Certificate Manager 를 활용한 SSL 인증

load balancer 를 사용하여 443 포트 변환 과정을 통해 HTTPS 프로토콜로 변환하였습니다.

===== 15번 슬라이드 =====

저희 프로젝트에 사용된 기술입니다.

웹의 경우,

프론트엔드는 Springboot와 Thymeleaf template engine을 사용하였으며

백엔드도 마찬가지로 Springboot를 사용하였습니다.

모바일의 경우,

Retrofit2 라이브러리를 활용하여 Springboot를 REST API로 사용하였습니다.

===== 16번 슬라이드 =====

다음으로는 팀 역할입니다.

발표자인 저는 백엔드, 안드로이드, AWS, 디비를 맡아서 진행하였고,

조원 이재현은 프론트엔드와 백엔드,

조원 홍혁기는 백엔드와 디비,

그리고 마지막으로 취업으로 인해 이 자리에 없는 조원 정성근이 프론트 엔드를 맡았습니

다.

===== 17번 슬라이드 =====

이상으로 질문 받는 시간을 가지며 발표를 마치도록 하겠습니다.

=====