

투표 기반 커뮤니티 어플리케이션 - Do표

최승빈*, 김보성*, 한동근*, 송성민*, 신병철**, 배성민***, 박세진****

Vote-based community application - Do Vote

Seung-Bin Choi*, Bo-Sung Kim*, Dong-Geun Han*, Seong-Min Song*, Byung-Cheol Shin**,
Seong-Min Bae***, and Se-Jin Park****

요 약

보통 커뮤니티라 함은 특정 관심사를 기준으로 사용자들이 유대감을 가지고 서로 정보를 공유하거나 커뮤니케이션을 하는 것을 커뮤니티라 한다. 커뮤니티의 기본 개념에 투표 기능을 합쳐 일상 속에서 주변 사람들에게 말하기 애매했던 고민들을 투표로 부담 없이 물어보고, 투표를 통해 다른 사람들의 의견을 수치화해 자신이 물어보고 싶었던 부분에 대해 수치화된 타인의 의견을 확인할 수 있는 커뮤니티를 만들고자 한 캡스톤 디자인 프로젝트에 관해 설명하고자 한다.

Abstract

In general, a community is a community where users share information or communicate with each other in a bond based on a specific interest. By combining the basic concept of the community with the voting function, we would like to explain the Capstone design project to create a community where you can freely ask people around you through voting and check the opinions of others by quantifying the opinions of others through voting.

Key words

Application, Android, iOS, React Native, Spring Boot, Vote-based Community

1. 서 론

구상한 프로젝트의 아이디어는 기본적으로 일상 생활에서 생겨나는 고민들 중 지인들에게 말하기 곤란한 고민 등을 익명으로 올리고, 다른 사람들의 의견을 투표를 통해 받을 수 있으며 댓글을 통해 의견을 남겨준 사용자가 자신의 의견에 대한 추가적인 설명이나 다른 사용자와의 커뮤니케이션이 가

능한 서비스를 제공하는 어플리케이션의 개발을 하고자 한다.

일반적인 커뮤니티는 특정 관심사를 기준으로 사람들이 해당 관심사를 중심으로 하는 커뮤니케이션을 수행하고, 게시글과 댓글을 통해 상호작용한다. 스마트폰 등의 보급으로 인터넷에 접근하기 쉬워지자 전체 연령대에서 커뮤니티를 사용하는 사람 수가 늘어났는데 이에 대한 통계자료[1]에서는 10대부

* 계명대학교 컴퓨터공학과, csb8226@naver.com, ,kimbosung22@naver.com, ehdrms001030@naver.com, song47166@naver.com,

** 계명대학교 수학과, sb5811c98@gmail.com,

*** 계명대학교 통계학과, smin9e@naver.com,

****계명대학교 컴퓨터공학과(교신저자), baksejin@kmu.ac.kr

터 40대까지 표본 집단 900명을 선발해 조사한 결과, 조사대상 전체 대상으로는 커뮤니티 이용률이 71.4%로 과반수를 넘어서고, 특히 20대의 경우 전체 평균보다 높은 74.8%의 이용률을 보인다. 커뮤니티 활성화를 바탕으로 투표와 커뮤니티의 개념을 합쳐 주변 사람들에게 말하기 힘들었던 고민이나 다른 사람에게 물어보고 싶은 주제에 대해서 물어볼 수 있는 투표 기반 커뮤니티 서비스를 개발하면 넓은 사용자층을 얻을 수 있을 것으로 기대하며 해당 프로젝트를 생각했다.

보강 자료로 커뮤니티 사용자의 목적에 대한 자료[2]에서 커뮤니티 사용자를 대상으로 설문조사를 진행한 결과 정보지향 인원의 비중이 전체 60%, 정보를 얻으며 친목까지 형성하는 관계지향 인원이 전체 20% 정도를 차지하는 것을 알 수 있다. 그리고 커뮤니티에서 얻을 수 있는 정보에 대한 신뢰도 조사에선 긍정 응답 비중이 85%이기에 커뮤니티 서비스가 기대하는 방향으로 발전 가능할 것이다.

II. 본 론

2.1 투표와 커뮤니티의 결합

투표와 커뮤니티를 결합하는 것의 가장 큰 목적은 ‘의견의 수치화’이다. 일반적으로 커뮤니티에선 의견을 서로 교환 할 때 게시글에 대해서 다른 사용자가 댓글을 달아 게시글 작성자에게 자신이 해당 게시글에 대해 어떤 생각을 하는지 알려주는 것이 일반적이다. 이는 의견을 구하려고 게시글을 작성한 경우 게시글로 올려둔 의제에 대해 의견의 비중이 수치적으로 얼마가 되는지 한 눈에 알아보는 불편하다. 이러한 문제점을 해결하고 의견의 수치화를 투표를 통해 수행하며 자신이 달아둔 의견에 대한 추가 설명을 댓글로 수행하며 댓글 탭 안에서 사용자들 간 의견 교환이 일어나면서 논리적으로 대화하는 과정을 거치며 생산적으로 돌아갈 수 있는 커뮤니티가 될 것으로 기대한다.

추가 자료[3]에서는 집단의 지능이 항상 개인보다 좋은 결정을 내리지 않는 않지만, 상호작용과 협업 방식에 따라 그 효과가 크게 증대될 수도 있다는

부분에서 명확하게 자신의 의견을 투표로 주장하고 주장에 대한 추가 설명을 댓글로 수행하는 커뮤니티면 타인의 의견을 모아서 결정하고자 하는 사람들에게 훨씬 효과적일 것으로 예상된다.

2.2 유사 어플리케이션과 비교

유사 어플리케이션들도 존재하지만, 그들과는 차이점이 존재한다. 표 1에서 차이점을 정리한다.

표 1. 유사 어플리케이션과 비교

Table 1. Compare to similar applications

서비스명	요약	차이점
뉴밍	관심분야 뉴스기사 제공, 부가기능으로 투표기능 존재	투표는 단순 부가기능으로만 존재
더폴	뉴스기사 기반 운영자 주관 투표	개별 유저가 투표 생성 불가
모두를 위한 설문조사	유저 제작 투표	카테고리 없음, 기본적으로 비공개 투표

표와 같이 유사 어플리케이션은 존재하더라도 유저가 생성하는 투표가 메인이면서 카테고리 기능까지 제공하고, 기본적으로 투표를 생성하면 다른 사용자들이 공개적으로 접근할 수 있는 어플리케이션은 아직까지 없는 것으로 파악된다.

2.3 시스템 설계 및 구현 기술 소개

어플리케이션 개발을 위해 사용된 기술 스택은 프론트엔드는 React Native를, 백엔드는 Spring Boot를 채택해 각 부분 구현을 수행했다. 데이터베이스는 AWS RDS에서 Maria DB를 사용하며 구축은 자료[4]를 참고했다. 시스템 설계도는 그림 1과 같다.

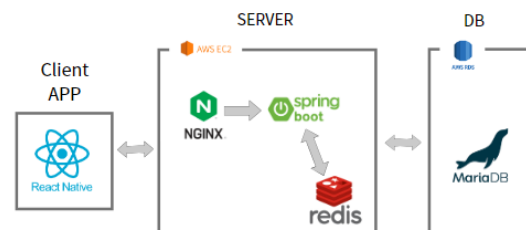


그림 1. 어플리케이션 시스템 설계도
Fig. 1. Application System Design Diagram

프론트엔드는 React Native Expo를 사용해 작성한 코드를 실시간 빌드하며 실제 구현할 때 어플리케이션이 어떻게 디스플레이에 표현되고 구현한 기능들이 작동하는지, 그리고 혹시나 구현상 문제가 있어 오류가 발생하면 그 부분에 오류 창을 띄워주는 등 개발과정 전반을 실시간으로 확인한다. 그리고 웹소켓(WebSocket)[5]이라는 실시간 양방향 통신 기술을 사용해 사용자 간 쪽지를 주고받을 수 있으며, 수신에 대한 알림을 받을 수 있도록 구현한다. 그리고 카-값 저장소인 레디스(Remote Dictionary Service)를 적용해 이를 활용한 휴대폰 인증 기능[6]을 탑재해 사용자가 계정을 생성할 때 본인 확인 절차를 수행하도록 만들고 구글 클라우드 플랫폼을 이용해 게시물과 댓글에 이미지를 업로드 할 수 있도록 구현한다. 그리고 구조적으로 무중단 배포를 수행할 수 있도록 백엔드 구조를 구현한다. AWS를 택할 때는 클라우드 서비스 성능 평가에 관한 논문[7]을 따랐다.

2.4 시스템 구현

2.3에서 언급한 시스템 설계도를 기반으로 하여 어플리케이션의 기능으로는 회원가입, 로그인, 마이페이지, 메인 탭, 카테고리 탭, 게시물, 쪽지로 크게 분류할 수 있다.

구현하는 어플리케이션의 UI 구성 및 흐름도는 아래와 같다.

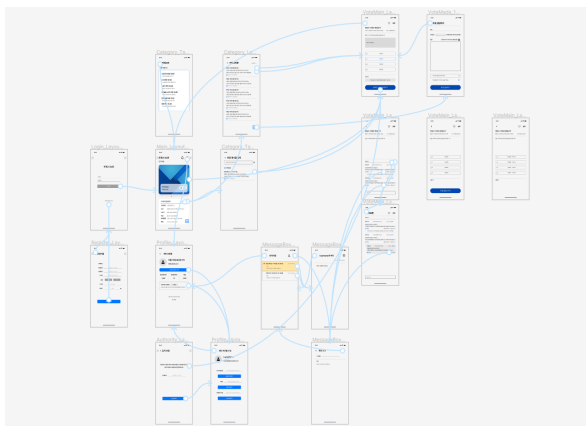


그림 2. UI 구성 및 어플리케이션 작동 흐름도

Fig. 2. UI Layout and Application flowchart.

2.4.1 회원가입

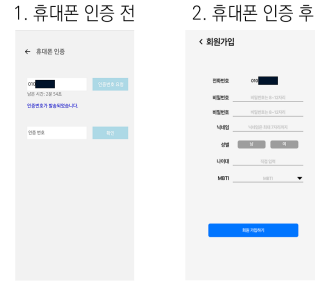


그림 3. 회원가입 페이지 구성 및 기능

Fig. 3. Registration Page Layout and Functionality

회원 가입 기능은 로그인 페이지의 아이디 및 패스워드 입력, 로그인 버튼의 하단에 ‘처음 사용하시나요?’라는 멘트 아래 텍스트 버튼을 구현하여 하이퍼링크와 같이 회원가입 페이지로 이동시켜 회원가입을 수행토록 한다. 회원가입 시 휴대폰 인증을 수행하여 본인 확인 절차를 거치며 다중 계정 생성을 방지하고 아이디와 패스워드, 닉네임, 성별, 나이대 등 필요한 기본 정보들을 획득한다.

2.4.2 로그인

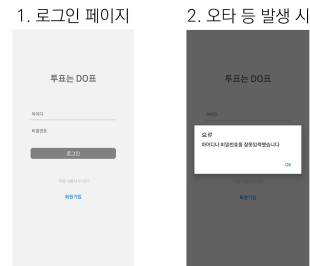


그림 4. 로그인 페이지 구성 및 기능

Fig. 4. Login Page Layout and Functionality

로그인 페이지는 어플리케이션 로그인이 되지 않은 상태라면 기능을 사용하지 못하므로 메인 페이지에서 로그인을 하고 나서 어플리케이션의 모든 서비스를 이용할 수 있도록 한다. 해당 페이지에는 처음 사용하는 사람들을 위해 회원가입 페이지로 연결되는 버튼을 하단부에 배치한다.

2.4.3 마이페이지

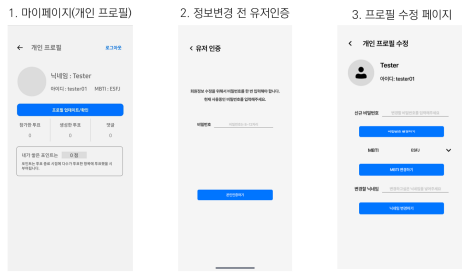
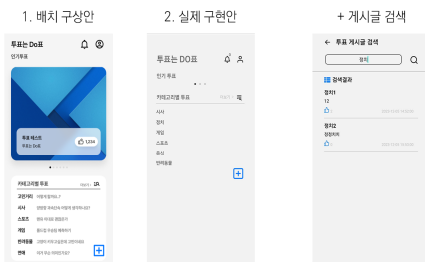


그림 5. 마이페이지(유저 프로필 페이지) 구성 및 기능
Fig. 5. User Profile Page Layout and Functionality

마이페이지는 메인 페이지 우측 상단에 마이페이지로 이동할 수 있는 아이콘을 배치하여 마이페이지로 이동할 수 있도록 하고, 마이페이지로 들어가면 사용자가 앱 내에서 한 활동 (작성한 게시글 수, 작성한 댓글 수 등..)을 수치로 볼 수 있고, 게시글에 대한 투표를 한 후 만약 투표한 항목이 투표 종료 시점에 다수의 의견이면 점수를 부여하는 대중성 포인트를 마이페이지에서 확인할 수 있도록 한다. 그리고 마이페이지에서는 회원정보 수정 페이지로 넘어갈 수 있게 하며 회원정보 수정 페이지로 갈 땐 본인이 아닌 타인이 해당 페이지에 들어갈 수 없도록 패스워드를 입력하도록 한다.

2.4.4 메인 페이지



* 실 구현한 이미지는 테더베이스에 게시물 데이터가 없는 상태

그림 6. 메인 및 게시글 검색 페이지 구성
Fig. 6. Main and Post-Search Page Layout

메인 페이지에서는 추천을 많이 받은 게시글을 메인 페이지의 중앙부에 띄워주는 인기투표 영역을 배치하고, 하단에 각 카테고리 별 최신 투표를 볼 수 있는 카테고리 별 투표 영역을 배치한다.

그리고 메인 페이지 카테고리별 투표 우측편에 존재하는 돋보기 버튼은 투표 게시글을 검색할 수

있게 해 주는 게시글 검색 페이지로 넘어가는 버튼으로, 게시글의 제목 또는 내용을 토대로 검색한다.

2.4.5 카테고리 탭

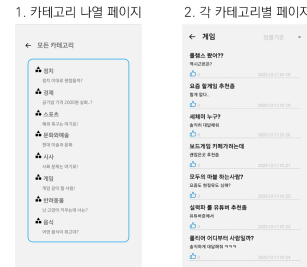


그림 7. 카테고리 페이지들(나열, 카테고리별) 구성
Fig. 7. Category Pages (Listed, By Category) Layout

카테고리 탭에서는 해당 어플리케이션에서 제공할 다양한 카테고리 종류를 보여주고, 각 카테고리별 게시글을 분리해서 보여줄 수 있는 역할을 한다.

2.4.6 게시글

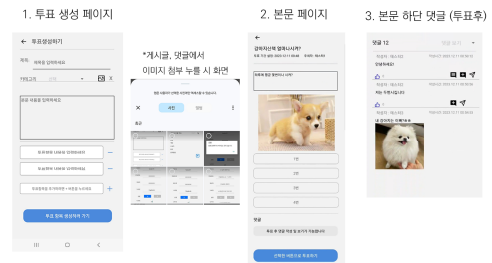


그림 8. 게시글 페이지들(생성, 투표, 댓글) 구성 및 기능
Fig. 8. Post Pages (Creation, Vote, Comment) Layout and Functionality

게시글 페이지는 게시글 생성과 투표 수행을 위한 본문페이지를 포함한다. 페이지의 상단부에는 게시글의 내용과 투표 항목(투표 항목은 최소 2개에서 최대 4개) 영역으로, 투표 항목에 대해서는 하나만 선택할 수 있도록 한다. 투표를 완료하고 나면 댓글 작성과 확인이 활성화되고, 다른 사용자들이 작성한 댓글을 확인하고 답글을 달 수 있다. 그리고 게시글 페이지 우측 상단부에는 좋아요 버튼이 존재하고, 해당 버튼을 누르게 되면 인기 게시글로서 추천하는 기능을 한다. 일정 기간(ex. 일주일)동안 많은 좋아요를 받게 되면 해당 게시글은 메인 페이지의 인기투표 탭에 띄워지게 한다.

게시글 작성의 경우 카테고리를 선택할 수 있으며 게시글의 본문을 작성하고, 이미지를 추가[8]할 수 있으며 투표 항목의 개수 조절과 항목별 내용을 작성할 수 있도록 한다.

2.4.7 쪽지

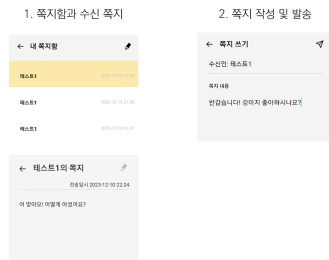


그림 9. 쪽지 관련 페이지들 구성 및 기능
Fig. 9. The Layout and functionality of Message-Related pages

쪽지는 메인 페이지의 중 모양 아이콘을 누르면 자신에게 온 쪽지를 확인할 수 있으며, 쪽지의 발송은 게시글 내 댓글에서 쪽지 보내기를 누르면 쪽지를 보낼 수 있고, 받은 쪽지에 대한 답장은 쪽지 페이지에서 우측 상단의 말풍선 아이콘을 누르면 답장을 보낼 수 있도록 구성한다.

III. 결 론

본 논문에 작성된 개발 프로젝트는 서론에서 말한 내용과 같이 일상생활에서 생겨나는 고민들 중 지인들에게 말하기 곤란한 고민이나 지인이 아닌 다른 사람들의 의견을 묻고 싶을 때 사용할 수 있는 의견을 구하는 커뮤니티로서 다양한 사람들의 의견을 얻을 수 있고, 그 의견을 수치화 해 주는 독특한 커뮤니티로서 작용하면 유용하게 사용될 수 있을 것 같다고 생각해 해당 프로젝트를 기획했다. 프론트엔드는 React를 사용하므로 모바일 환경 (Android, iOS) 뿐만 아니라 웹 환경에도 후에 대응하여 서비스를 사용할 수 있는 장치 종류를 늘릴 수 있다. 현재 구현된 기능은 투표와 커뮤니티를 혼합한 어플리케이션에 기본적으로 필요한 기능을 모아둔 것이고, 추가 기능들이 더 구현된다면 발전할 수 있는 여지가 충분히 존재한다고 생각한다.

참 고 문 헌

- [1] Lee Ye Ji, "More than 70% of MZ generation use online communities, popularity of everyday, economic, financial, and game topics", <https://www.casenews.co.kr/news/articleView.html?idxno=4272>, 사례뉴스, 2021. [asssed: Apr. 26, 2024]
- [2] Hee-Jung Cho, Seok Jin Lew, Si-Hong Jeon, and Hun-A Lee, "Study of online community user behavior based on using objective type : compared of media use and emotional factors", OUGHTOPIA, Vol. 30, No. 1, pp. 167-199, 2015.
- [3] J. Salminen, "Collective Intelligence in Humans: A Literature Review," arXiv:1204.3401 [cs.CY], Apr. 2012. [Online]. DOI: 10.48550/arXiv.1204.3401
- [4] AWS, "Amazon RDS for MariaDB", https://docs.aws.amazon.com/ko_kr/AmazonRDS/latest/UserGuide/CHAP_MariaDB.html, Amazon Relational Database Service User Guide, [asssed: Apr, 28, 2024]
- [5] F-Lab, "How Web Sockets Work and Use in Real-Time Applications", https://f-lab.kr/insight/understanding-websocket-and-it-s-application?gad_source=1, F-Lab short insight, 2024. [asssed: Apr, 28, 2024]
- [6] develop study memo log =], "[Spring boot] Implement text SMS authentication (2)", <https://diddl.tistory.com/122>, [asssed: Apr, 28, 2024]
- [7] B. E. Zant and M. Gagnaire, "Performance evaluation of Cloud Service Providers," 2015 International Conference on Information and Communication Technology Research (ICTRC), Abu Dhabi, United Arab Emirates, 2015, pp. 302-305, DOI: 10.1109/ICTRC.2015.7156482.
- [8] Seo-Yeon Choo, "Upload Image Files from Springboot to GCS Total Cleanup (multipart/form-data)(Google Cloud Storage)(GCP)", <https://choo.oopy.io/35bffd94-7a41-4cfa-812c-b8aaf148604a>, in Choo's head, 2023 [asssed: Apr, 28, 2024]