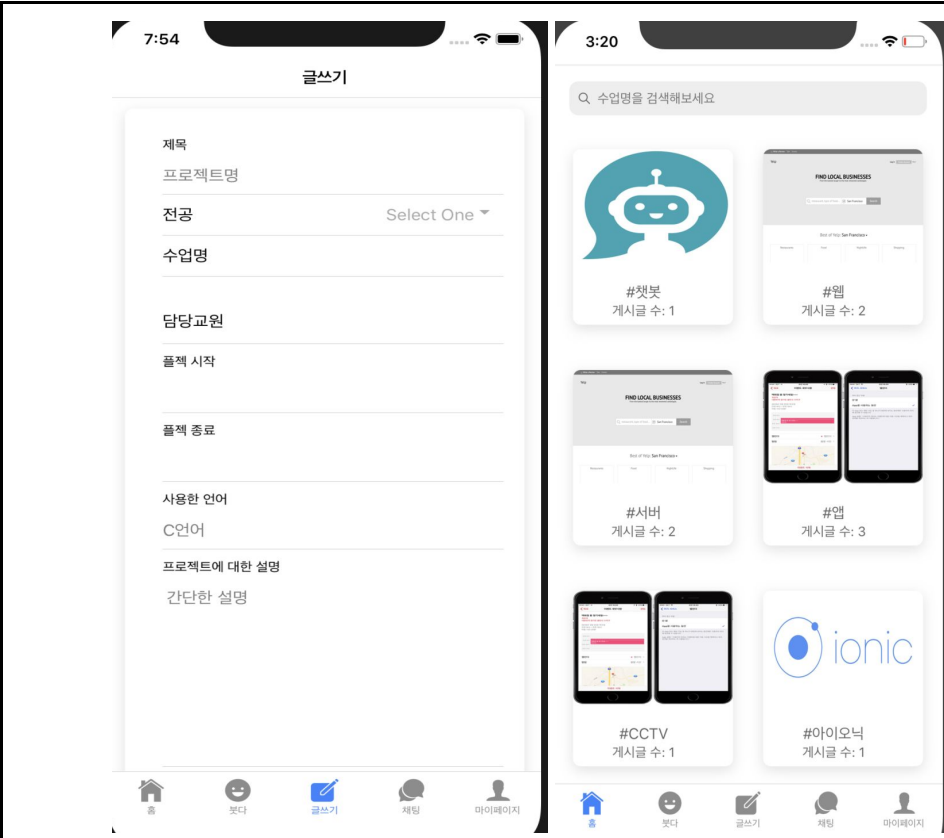


공개 소프트웨어 프로젝트 최종보고서

과제 수행원 현황						
수행 학기	📅2019년 3월~6월					
프로젝트명	소프트웨어학과 학생들을 위한 프로젝트 서포팅 앱 '봣다'					
팀명	Myway (1조)					
	학과	학번	성명	성별	연락처	E-mail
팀장	컴퓨터공학과	2017112113	유소영	여	010-8914-5587	yooso020599@naver.com
팀원	컴퓨터공학과	2017112131	이민주	여	010-4127-2357	napbest@naver.com
	컴퓨터공학과	2017112120	정유나	여	010-6600-9014	yoona607@naver.com
	컴퓨터공학과	2017112142	하유진	여	010-9533-2887	hoj2887@naver.com
지도교수	교과목명	공개SW프로젝트				
	소속	📍컴퓨터공학전공				
	성명	손윤식교수님				

# Key Words	아이오닉	Firebase	챗봇	채팅	
1.개발동기/ 목적/필요성 및 개발 목표	<p>소프트웨어 학과 학생들은 고학년이 될수록 자유주제를 가지고 프로젝트를 진행하는 경우가 많습니다. 이 때 가장 중요한 부분 중 하나인 주제 선정에서 많은 시간을 소비해 구현에 적절한 시간을 투자하지 못하는 경우가 많이 발생합니다. 또, 주제 선정 후에도 실제 구현 과정에서 많은 어려움을 겪습니다. 이러한 고민에서 아이디어를 얻어 저희는 다음과 같은 앱을 구현하기로 했습니다. 이전에 프로젝트를 진행했던 학생들이 게시글을 작성해 정보를 제공하면 프로젝트를 진행할 학생들이 주제 선정에 도움을 받고, 채팅을 통해 세부적인 정보들을 개발자에게서 직접 들을 수 있으며, 챗봇을 통해 게시글의 내용뿐만 아니라 추가적인 정보들을 생성하고 (좋아요 수 계산을 통해 게시글 인기순위 생성) 제공하는 것이 저희의 목표입니다.</p> <p><u>즉, 소프트웨어 학과 학생들이 프로젝트를 진행할 때 프로젝트 주제 선정과 구현</u></p>				

	<p>과정에서 겪는 어려움을 해소할 수 있도록 도와주는 것이 저희의 목표입니다.</p>
<p>2.최종 결과물 소개</p>	<p>저희 App의 주요 기능은 크게 게시물, 채팅, 챗봇 3가지로 나눌 수 있습니다.</p>
	<p>1. 게시물</p> <p>먼저 게시물은 이전에 프로젝트를 진행했던 개발자가 직접 자신이 개발한 프로젝트에 대한 정보를 글로 작성하는 것입니다. 작성된 게시물은 메인 페이지에서 태그별로 분류하여 확인할 수 있습니다. 태그별로 분류함으로써 <u>관심있는 기술이나 언어, 플랫폼 등을 한 번에 모아볼 수 있습니다.</u> 또한 검색을 통해 과목별로 어떤 게시글이 있는지 확인할 수 있습니다. 게시글을 보여주는 페이지에서 좋아요나 스크랩 버튼을 누르게 되면, 마이페이지에서 내가 좋아요, 스크랩한 게시글의 목록을 확인할 수 있습니다.</p> 
	<p>[그림 1]. TAB3(게시글 작성) 및 TAB1(메인 페이지)</p>
	<p>먼저 TAB3 글쓰기 페이지에서 개발자가 본인의 프로젝트에 대해 입력하는 정보는 다음과 같습니다.</p> <p>게시글 제목, 작성자의 전공 및 프로젝트를 수행한 수업명, 담당교원, 프로젝트 시작 및 종료 날짜입니다. 또한 프로젝트에 관련된 상세한 정보도 입력할 수 있도록 하였습니다. 우선 이 앱 사용자를 소프트웨어 학과로 한정하였기 때문에 프로젝트 개발과정에서 사용한 언어 , 프로젝트에 대한 개요, 실행 방법, 실행환경, 프로젝트를 올린 깃허브 주소,</p>

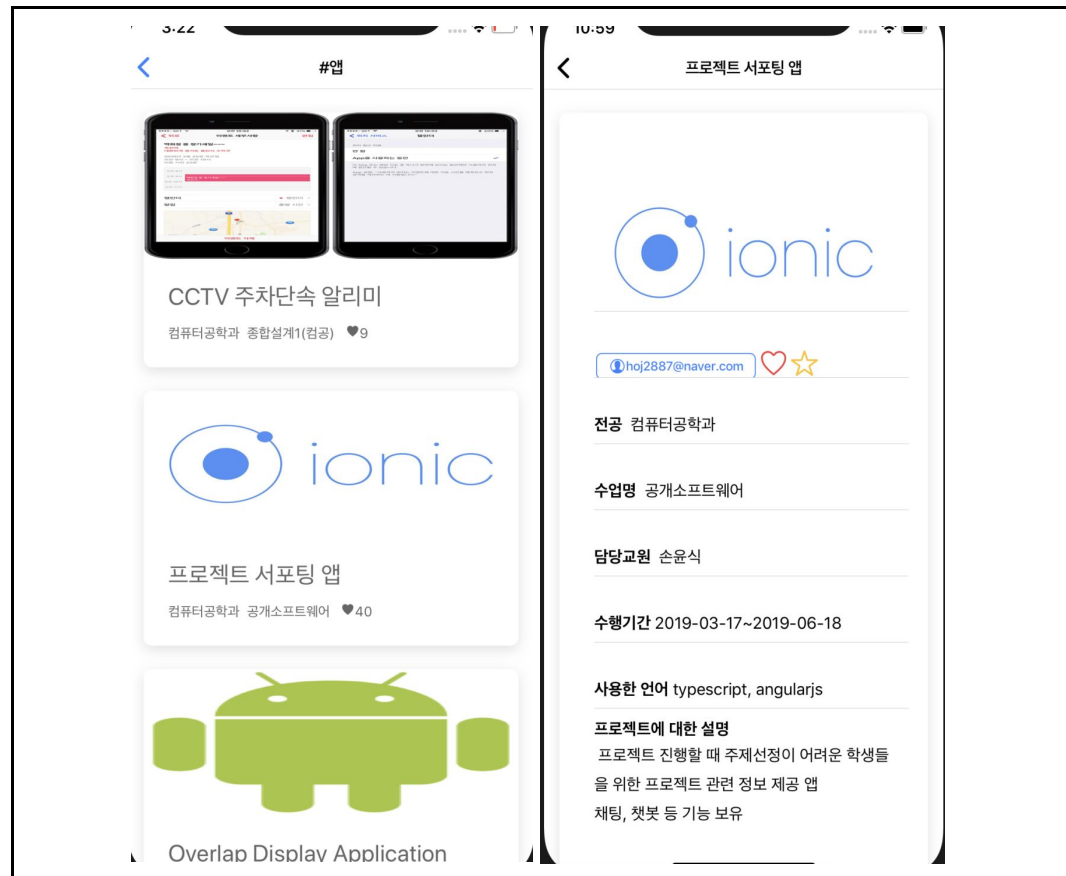
프로젝트 결과물을 보여주는 이미지가 있습니다.

또한 저희의 게시글을 보여주는 것 중 핵심 부분이라고 할 수 있는 해시태그를 입력하는 칸도 있습니다.

해시태그는 이 프로젝트를 잘 나타낼 수 있는 기술 및 키워드들을 입력하는 것입니다.

이렇게 작성한 게시글의 data는 firebase에 json 형식으로 저장이 됩니다.

이후 작성된 게시글이 태그별로 분류된 것을 TAB1에서 확인할 수 있습니다.



왼쪽부터 [그림 2, 3]. TAB1에서 특정 태그 및 게시글을 선택한 경우

다음은 TAB1에 태그별로 분류된 것 중 관심있는 태그를 선택하면, 그 태그와 관련된 게시글들의 목록이 보여집니다. 게시글에 업로드한 이미지와, 제목 및 이 게시글이 받은 좋아요 수를 보여줍니다.

특정 태그에 해당하는 게시글 목록들이 보여지는 페이지(왼쪽)에서 특정 게시글을 선택하면 관련 정보들이 보여집니다. 또한 게시글 작성자를 선택하고 채팅을 원하면 채팅 페이지로 넘어가 프로젝트 개발자와 1:1 대화가 가능하도록 하였습니다. 또한 좋아요/스크랩 등을 통해 관심있는 게시글을 저장하여 마이페이지에서 이를 확인할 수 있습니다.



[그림 4]. 좋아요/스크랩 누른 후

userInfo

- |- userId
- |- like
- |- scrap
- |- liketotal
- |- scraptotal

[그림 4-1]. 좋아요/스크랩 DB 저장 구조

좋아요와 스크랩은 [그림 4-1]과 같은 트리 구조로 데이터베이스에 저장됩니다.

각 사용자의 좋아요와 스크랩은 userInfo에 저장되며 전체 좋아요 수(= liketotal) 스크랩 수(=scraptotal)로 각각 like와 scrap에 저장되는 인덱스를 정해주어 그 인덱스에 좋아요/스크랩 한 게시글의 인덱스를 저장합니다.

```
|- like
  |- 0: 1
  |- 1: 3
  |- 2: 7
|- liketotal: 3
```

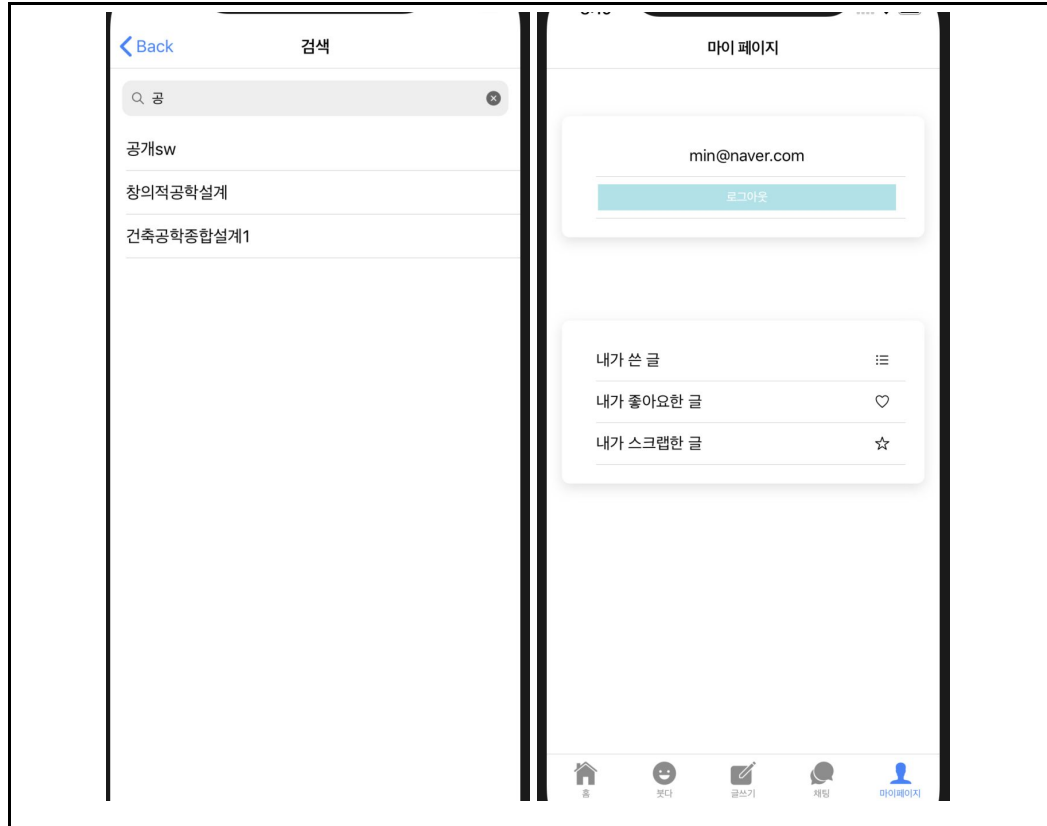
[그림 4-2]. 좋아요 저장구조 (7번 글 좋아요 했을 경우)

```
|- like
  |- 0: 7
  |- 1: 3
|- liketotal: 2
```

[그림 4-3]. 좋아요 저장구조 (1번 글 좋아요 취소)

예를 들어, [그림 4-2]를 보면 좋아요한 게시글은 게시글 1번, 3번, 7번 총 3개임을 알 수 있고 이 그림에서 게시글 1번 좋아요 취소를 했을 경우 [그림 4-3]이 됨을 알 수있는데 만약 1번 게시글이 저장돼있던 0번 인덱스를 삭제해버린다면 인덱스 1, 2 가 남고 liketotal은 2가 되므로 새로 다른 게시글의 좋아요를 누르면 인덱스 2 자리에 새로운 게시글 인덱스가 저장되므로 데이터 손실이 일어나게 됩니다. 이를 방지하기 위해 좋아요 취소한 게시글이 있는 인덱스 값에 제일 마지막 인덱스의 값을 저장한 뒤 마지막 인덱스를 삭제하고

liketotal의 값을 하나 줄여주어 데이터 손실을 막았습니다.
스크랩도 좋아요와 같은 방식으로 추가와 삭제를 하였습니다.



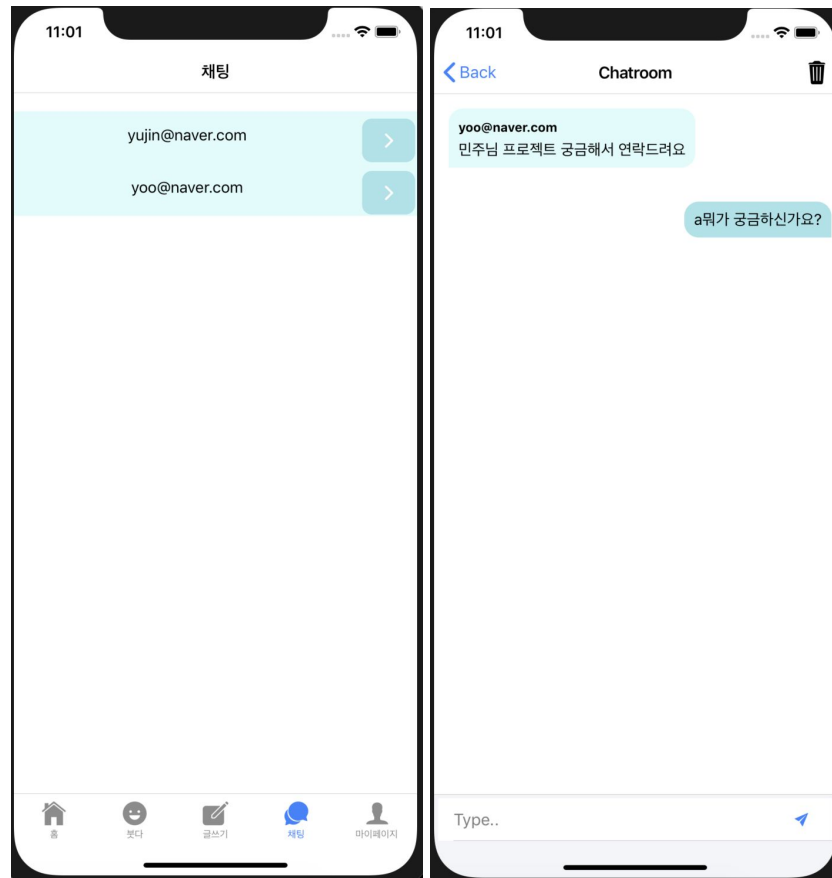
[그림 5]. TAB1(검색창) 및 TAB5(마이페이지)

다음은 TAB1 메인페이지에서 검색을 하는 화면입니다. 같은 글자가 하나라도 있으면 해당하는 수업들이 나열되므로 검색하기 편합니다. 수업명을 검색하면 그에 맞는 게시글로 이동합니다.

마이페이지에서는 나의 계정(이메일)으로 로그인/로그아웃이 가능하고 내가 쓴 글, 좋아요/스크랩한 글 을 볼 수 있습니다. 해당 게시글로 이동이 가능합니다.

2. 채팅

관심있는 게시글들의 정보를 모두 본 이후, 프로젝트에 대해 더 자세한 정보를 얻고 싶을 경우, 게시글 작성자와의 채팅을 통해 그 궁금증을 해결 할 수 있는 기능입니다.



[그림 6]. TAB4(채팅 목록 및 내용)

```
chat_list
|- index
  |- Timestamp
  |- uid1
  |- uid2
  |- num
```

[그림 7]. 채팅 목록 data 구조

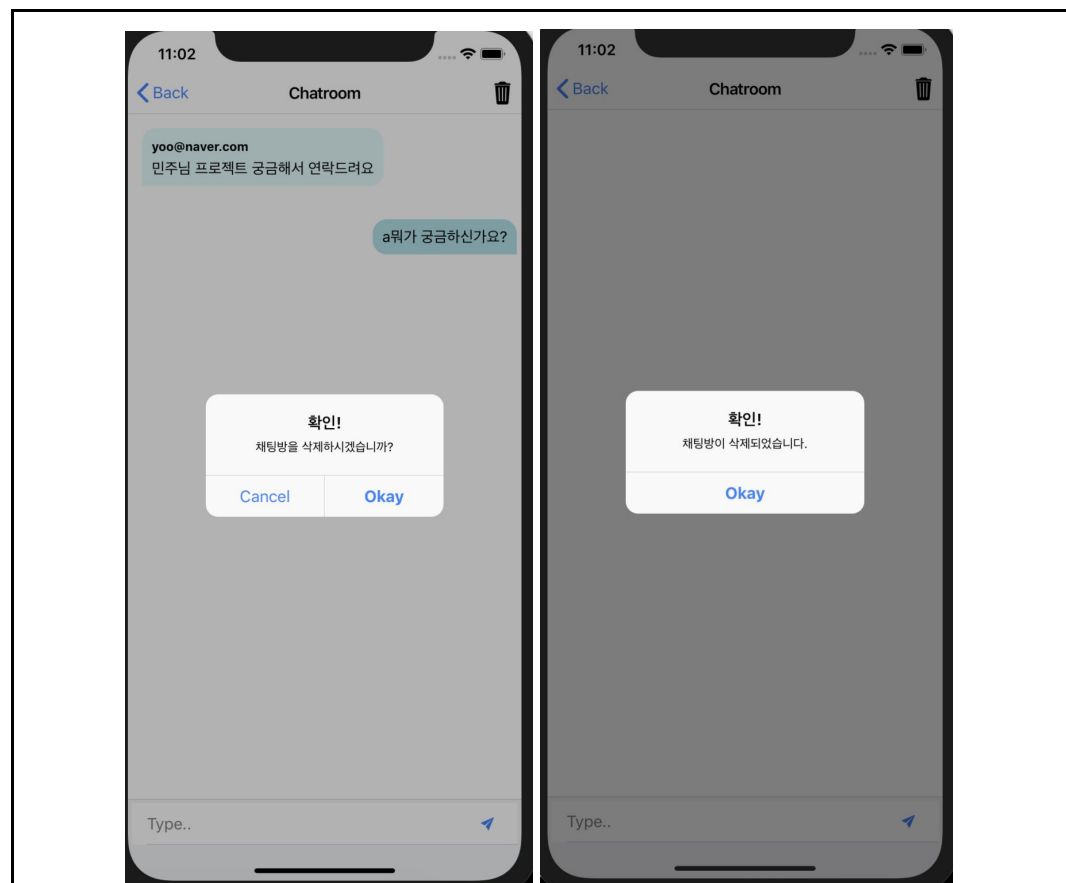
```
chat_message
|- index
  |- Timestamp
  |- me
  |- you
  |- num
```

[그림 8]. 채팅 내용 data 구조

우선 채팅은 실시간으로 메시지를 주고받는 변화가 보여야 하기때문에 실시간으로, 저장된 데이터 내용이 바뀔 때마다 데이터를 가져올 수 있는 firebase의 realtime 데이터베이스를 사용했습니다. 현재 로그인한 사용자가 지금까지 채팅한 상대 사용자들의 목록이 보이고, 그 중 한 목록을 선택하면 그 상대와 주고받은 내용이 나오도록 했습니다. 먼저 사용자가 채팅한 상대방들을 저장하기 위해 채팅 목록 데이터를 [그림 7]와 같이 저장했습니다. 새로운 채팅방이 개설 될 때, 채팅방에 참가하는 두 사용자를 uid1, uid2로

저장하고, 해당 채팅방이 개설된 시각을 timestamp로 저장했습니다. timestamp를 저장함으로써 firebase에서 데이터를 가져올 때 timestamp로 정렬을 해서 개설된 채팅방 순으로 보입니다.

2명의 사용자가 주고받은 채팅 내용은 [그림 8]과 같은 구조로 저장했습니다. 메시지를 보낸 사용자의 아이디를 “me”에, 상대방의 아이디를 “you”에 저장하고, 메시지가 전송된 시각을 timestamp로 저장했습니다. firebase에서 데이터를 가져올 때, 선택한 채팅 목록의 uid1과 uid2를 me와 you랑 매칭을 시켜서 해당 사용자들이 주고받은 메시지인지 판단했습니다. 또한 timestamp로 정렬을 해서 데이터를 가져오기 때문에 2명이 주고받은 내용을 순서대로 출력해 채팅 내용의 순서를 보존할 수 있습니다.



[그림 9]. TAB4(채팅 목록 및 내용 삭제)

다음은 채팅 목록 및 내용을 삭제하는 경우입니다. 더 이상 채팅을 하고 싶지 않거나 채팅방을 유지하고 싶지 않은 경우, 우측 상단에 있는 휴지통 버튼을 눌러 채팅방 및 채팅 내용을 삭제할 수 있습니다. 삭제할 시 firebase에 저장되어있던 데이터들도 삭제됩니다.

3. 챗봇

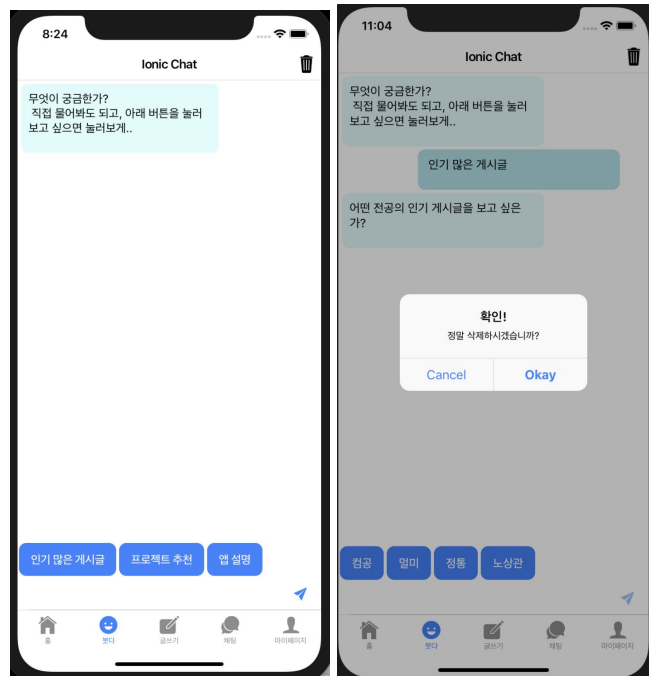
사용자가 추가로 궁금해하는 정보를 제공해 주는 봇다, 즉 봇다와 같은 말투를 가진

챗봇입니다. 챗봇에서 가장 중요한 대화흐름을 구상하기에 앞서 페르소나, 즉 이 앱의 사용자는 누구 일지, 이 사용자가 앱을 사용하는 이유를 가정하였습니다.

[페르소나]

	직업	앱 사용 이유
사용자 1	동국대 학생	내 전공의 인기 많은 게시글을 보고 내 프로젝트의 주제선정에 도움을 받아야겠다.
사용자 2		프로젝트 주제를 뭐로 할지 전혀 감이 안 잡힌다. 내 전공의 사람들, 혹은 나와 같은 수업을 들은 사람들은 어떤 주제로 프로젝트를 진행했을까?
사용자 3		내 전공이나 수업의 게시글을 확인 하고 싶은데 내가 참고할 만한 게시글이 존재할까? 게시글이 아예 없는건 아닐까? 몇개가 있을까?
사용자 4		학생들이 어떤 주제로 프로젝트를 많이 진행할까? 요즘 핫한 주제는 무엇일까?

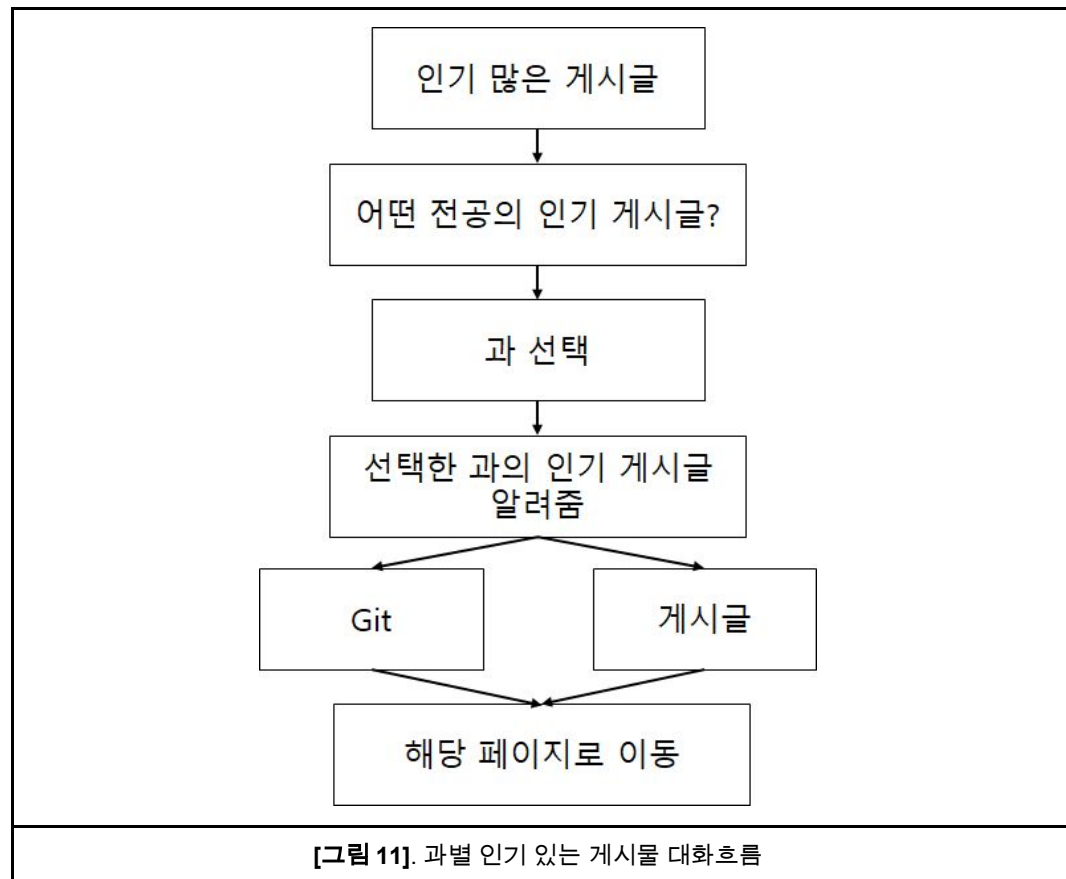
위의 페르소나로 크게 2가지 대화 흐름인 과별 가장 인기 있는 게시물 대화흐름, 전공별 프로그램 대화흐름, 그리고 그 외로 짧은 일문일답 대화흐름으로 구성하였습니다.



[그림10] TAB2(챗봇)

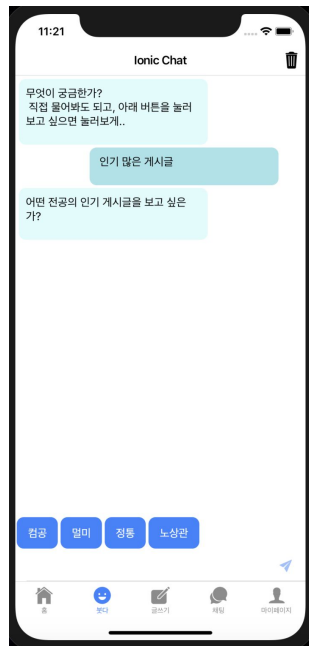
다음은 처음의 챗봇 붓다 화면입니다. 버튼 형식의 답을 설정해 놓아 챗봇이 사용자에게 어떤 질문에 대한 답을 제공하는지 알 수 있게 하였고, 버튼을 누르거나 직접 메시지를 작성하여 질문을 할 수 있도록 하였습니다. 그리고 오른쪽 위에 휴지통 아이콘을 달아

경고 문구를 나타내어 의사 확인 후 내용 삭제가 가능하도록 하였습니다.

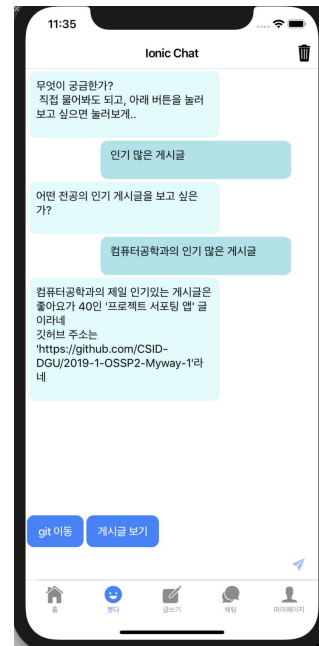


먼저 첫번째, [그림 8]의 붓다 첫 화면에서의 '인기 많은 게시물'의 버튼을 눌렀을 경우의 과별 인기 있는 게시물 대화흐름 입니다. 어떤 전공을 선택할것인지에 대한 붓다의 물음[그림10]에 사용자가 컴퓨터공학과, 멀티미디어공학과, 정보통신공학과 중 궁금한 과를 선택하거나 전체 게시글을 선택 한다면, 과를 선택했을 경우 선택한 과의 게시글들 중 좋아요 수가 가장 많은 게시글을 알려주고, 반대의 경우 전체 게시글들 중 좋아요 수가 가장 많은 게시글을 알려주며 [그림 13, 16, 17, 18] 깃 주소가 있다면 깃을 확인 할수 있게 [그림 14], 그리고 게시글을 확인할 수 있게 [그림 13] 대답을 합니다. 만약, 선택한 과의 모든 게시물의 좋아요 수가 0개이거나 게시물이 없다면 인기 있는 게시글이 무엇인지 판단할 수 없어 게시글을 알려주지 않습니다.

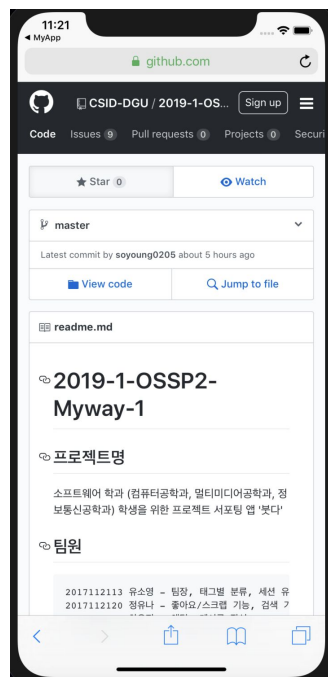
위의 대화흐름에 대한 앱 화면은 아래와 같고, 과에 대한 사용자의 선택에 대해 붓다가 정확히 대답 해준다는 것을 확인할 수 있습니다.



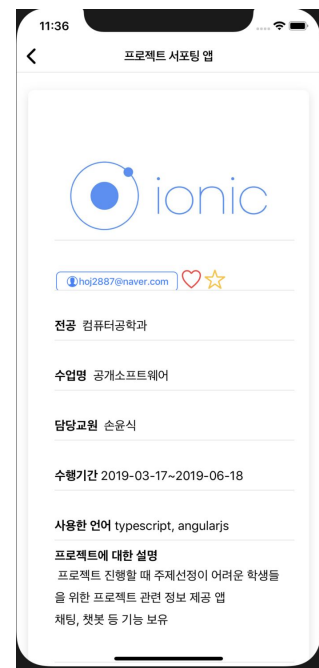
[그림 12]. 과를 선택하는 대화 단계



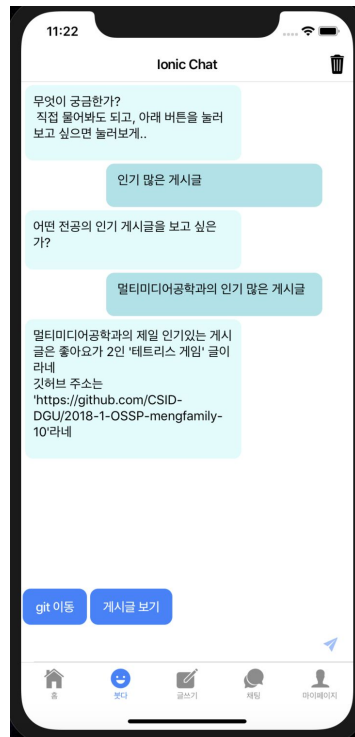
[그림 13]. 컴퓨터공학과 인기 게시글 대답하는 단계



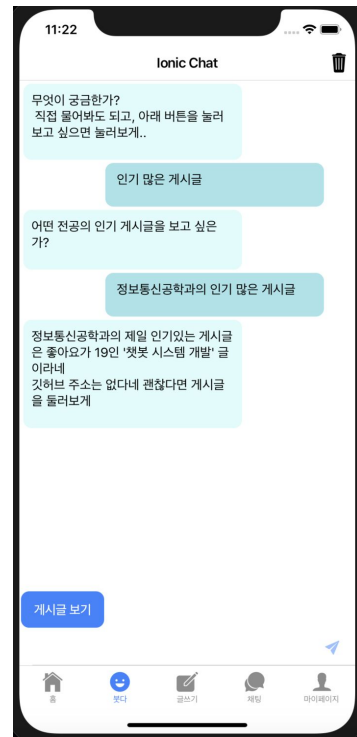
[그림 14]. 그림 11에서 'git 이동' 버튼을 누른 경우



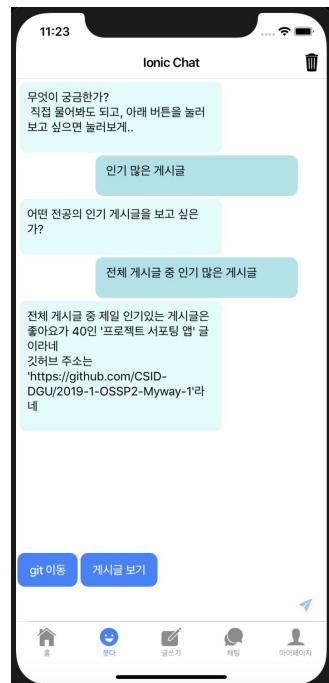
[그림 15]. 그림 11에서 '게시글 보기' 버튼을 누른 경우



[그림 16]. 멀티미디어공학과 인기 게시글을 답하는 단계



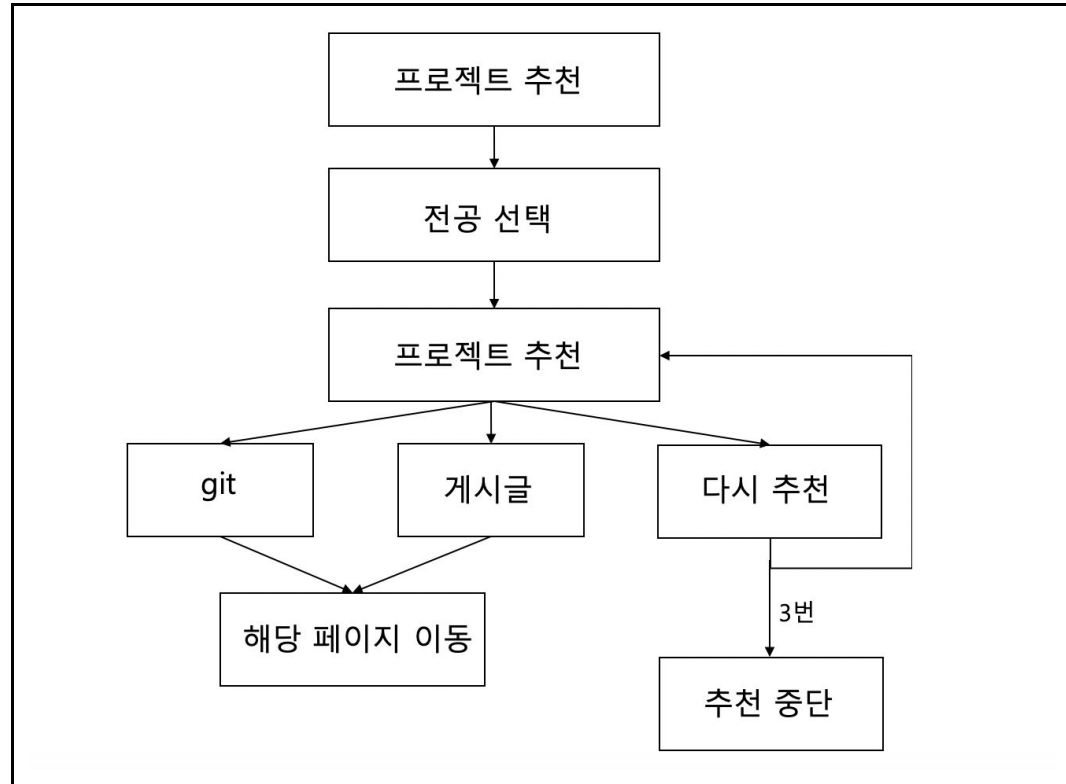
[그림 17]. 정보통신공학과 인기 게시글을 답하는 단계



[그림 18]. 과를 선택하지 않은 단계

그림 16, 17, 18에서 git 이동, 게시물 보기 버튼을 누른 경우 실행 화면은 그림 14, 15과 같습니다.

다음으로 두번째, 전공 별 프로젝트 추천 대화흐름입니다.

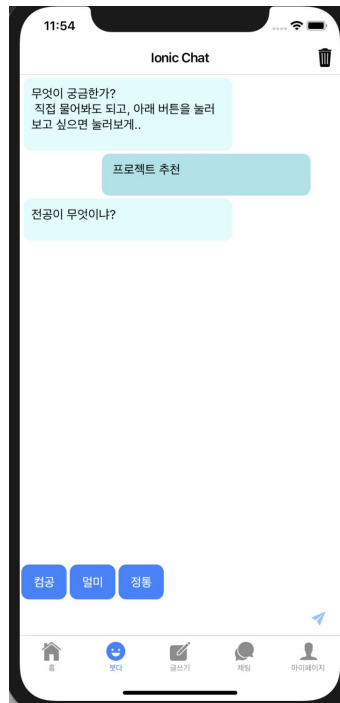


[그림 19]. 전공 별 프로젝트 추천

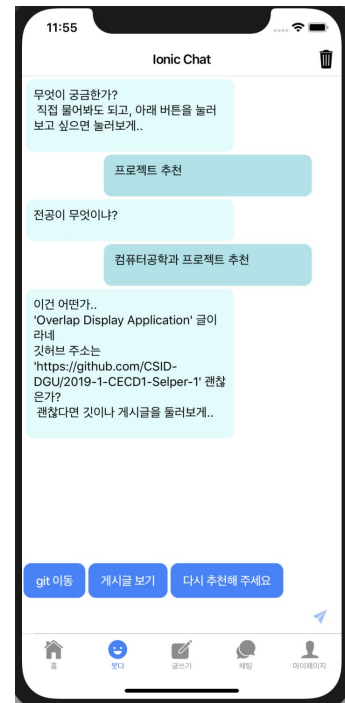
[그림 10]의 붓다 첫 화면에서 프로젝트 추천 버튼을 누르면 시작되는 대화흐름입니다. 어떤 전공을 선택할 것인지에 대한 붓다의 물음에 [그림 20] 사용자가 원하는 과를 선택하면 각 과에 대한 게시글을 추천해주며[그림 21, 22, 23] 해당 게시글의 깃 주소가 있다면 깃을 확인할 수 있게, 게시글을 볼 수 있게 [그림 14, 15] 대답해주며 붓다의 추천이 마음에 들지 않을 경우 '다시 추천' 버튼을 눌러 다시 추천 받을 수 있도록 하였습니다. 하지만 이 버튼을 연속 3번 누르게 되면 붓다는 자신의 추천을 계속 거부하는 사용자가 꽤심하여 추천을 중단합니다.[그림 22]

초기에 이 대화흐름을 기획할 때 전공 선택의 대화 단계가 아닌 사용자의 좋아요, 스크랩한 게시물의 해시태그 확인을 통해 관심사를 파악한 후, 그에 맞는 게시글을 추천해주는 대화 단계를 구상하였습니다. 하지만, 이를 위해서는 사용자의 아이디를 Dialogflow의 함수로 넘겨 주어야 하는데 방법이 없어 아쉽지만 선택한 전공의 게시글들 중에 랜덤으로 추천해주는 방식을 적용하였습니다.

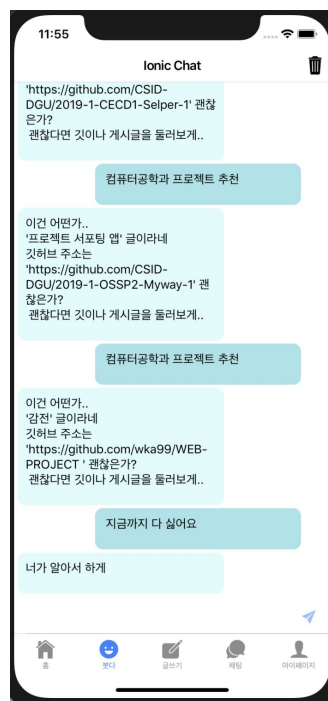
위의 대화흐름에 대한 앱 화면은 아래와 같고, 과에 대한 사용자의 선택에 대해 붓다가 정확히 대답 해준다는 것을 확인할 수 있습니다.



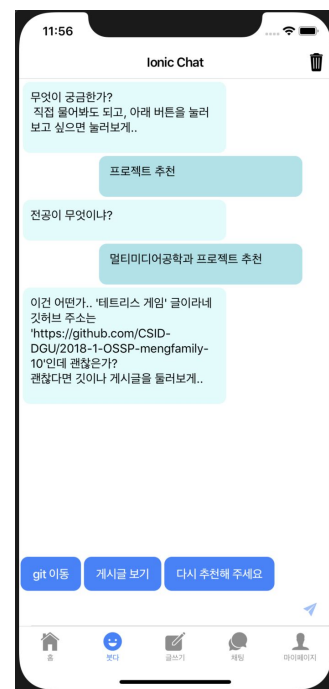
[그림 20]. 전공 선택하는 대화흐름



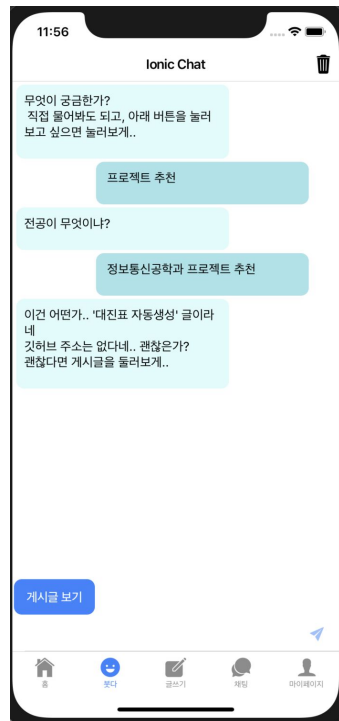
[그림 21]. 컴퓨터공학과를 선택한 단계



[그림 22]. '다시 추천'버튼을 3연속 누른 단계

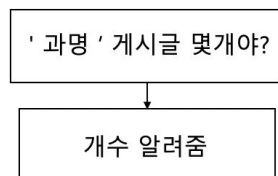


[그림 23]. 멀티미디어공학과를 선택한 단계

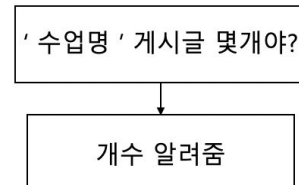


[그림 24]. 정보통신공학과를 선택한 단계

마지막으로 짧은 일문일답 대화흐름인 과별 게시글 개수, 수업별 게시글 개수, 요즘 핫한 태그, 수업별 프로젝트 추천 대화흐름입니다.



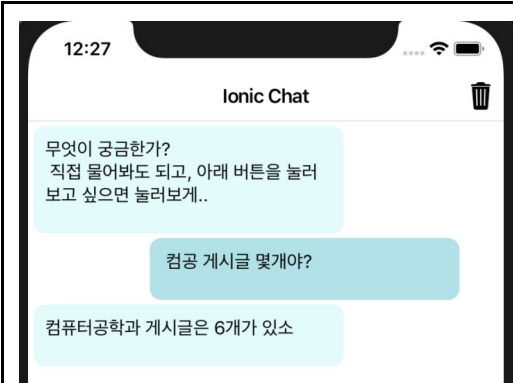
[그림 25]. 과별 게시글 개수 대화흐름



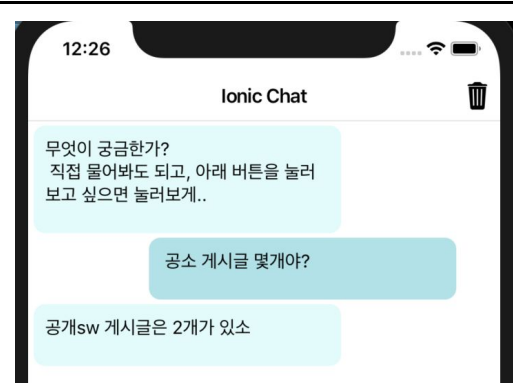
[그림 26]. 수업별 게시글 개수 대화흐름

위의 그림은 과별 게시글 개수와 수업별 게시글 개수 흐름입니다.

사용자가 궁금한 과명 또는 수업명에 대한 게시글이 몇개인지 질문한다면 봇다가 게시글 개수를 알려줍니다. [그림 27, 28]

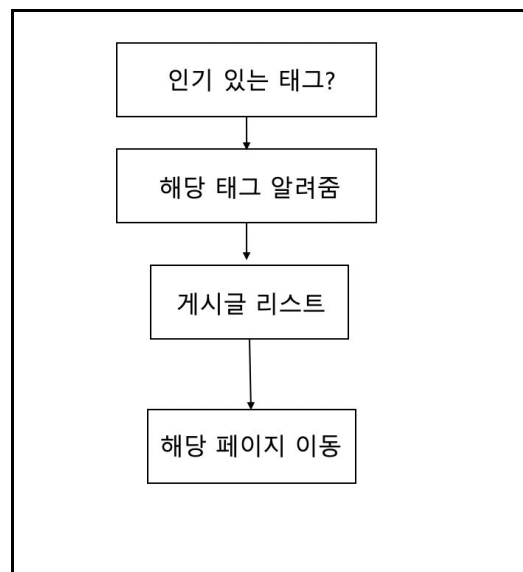


[그림 27]. 과별 게시글 개수 대화 단계

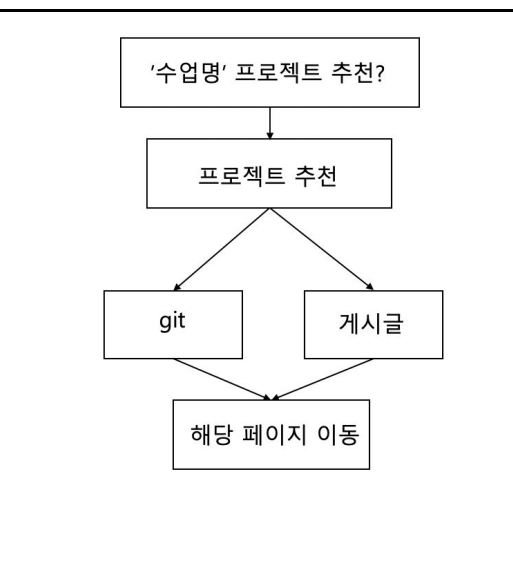


[그림 28]. 수업별 게시글 개수 대화 단계

위 그림에서 예시로 든 과, 수업명 이외의 다른 과와 수업명에 대해서도 같은 단계로 정확한 답을 줍니다.



[그림 29]. 요즘 핫한 태그 대화흐름

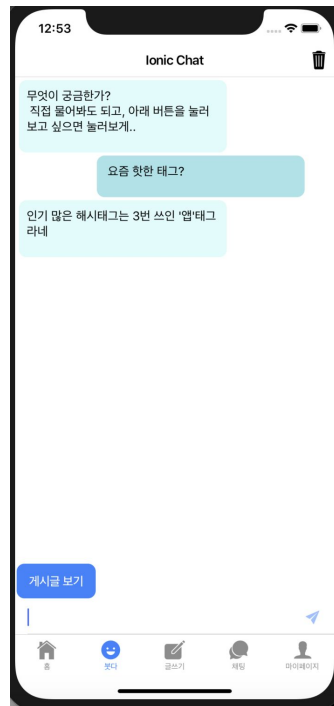


[그림 30]. 수업별 프로젝트 추천

위 그림은 요즘 핫한 태그, 수업별 프로젝트 추천 대화흐름입니다.

요즘 핫한 태그의 대화흐름[그림 31]은 사용자가 인기 있는 태그가 무엇인지 질문하면 붓다는 해당 태그를 알려주며 사용자가 원할 경우, 해당 태그의 게시글 목록을 보여주는 화면[그림 32]으로 넘어 갈 수 있도록 대답을 해줍니다. 요즘 어떤 주제로 프로젝트를 많이 진행하는지 알 수 있도록 하였습니다.

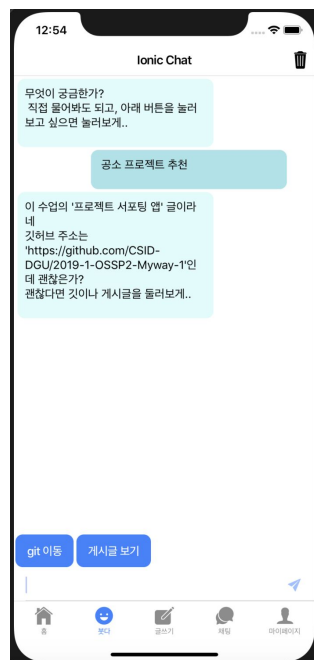
수업별 프로젝트 추천 대화흐름은 사용자가 궁금해 하는 수업의 프로젝트 게시글을 추천해 주며 해당 프로젝트의 깃과 게시글을 확인할 수 있도록 대답합니다.[그림 33, 34] 이를 통해 그 수업에 해당하는 게시글을 쉽게 확인할 수 있도록 하였습니다.



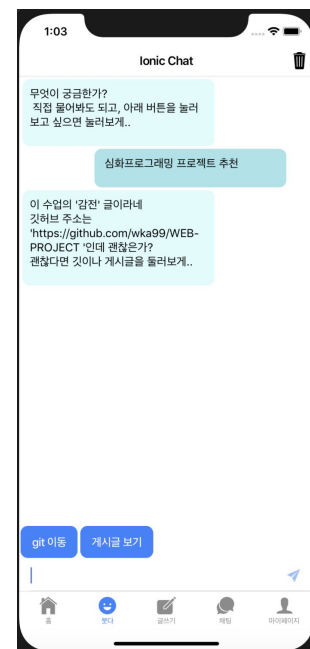
[그림 31]. 핫한 태그 대화 단계



[그림 32]. 그림 31의 게시글 보기를 눌렀을 경우



[그림 33]. 공개sw 프로젝트 추천 대화 단계

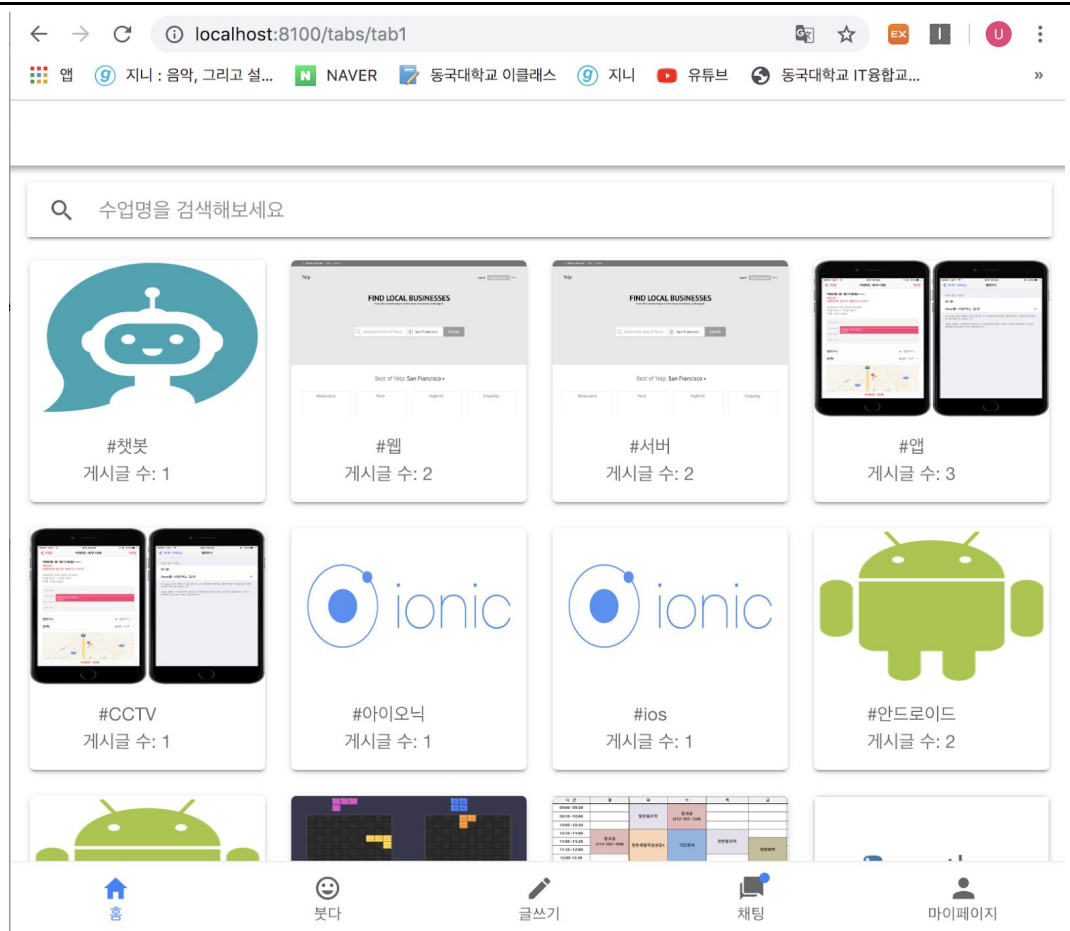


[그림 34]. 심화프로그래밍 프로젝트 추천 대화 단계

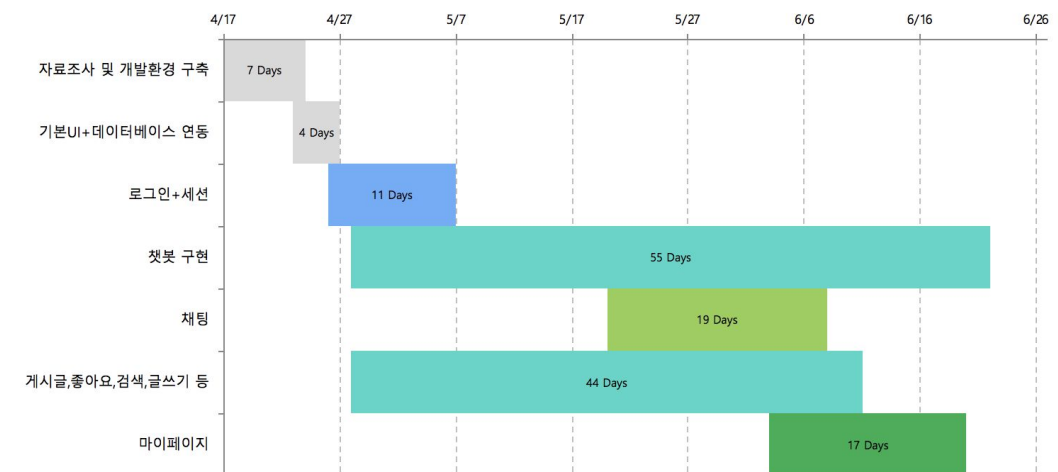
위의 예시 뿐만 아니라 다른 수업들에 대해서도 정확한 대답을 줄 수 있도록 하였습니다.

++

하이브리드 앱이기 때문에 위의 과정들을 웹에서도 실행 가능합니다.

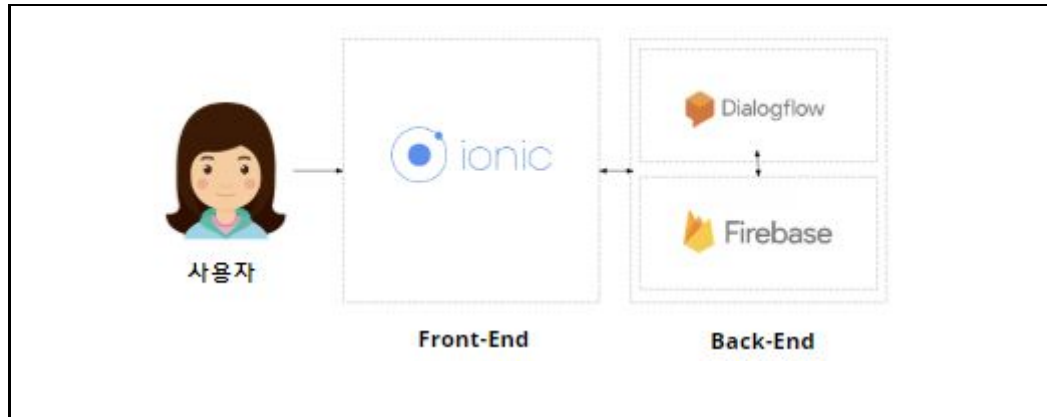


3.프로젝트 추진 내용



프로젝트 진행과정입니다. 저희는 중간에 주제를 한번 바꿨기 때문에 7주차에 새롭게 자료조사와 개발환경 구축을 시작했습니다. 팀원 전원이 빠르게 아이오닉 개발환경을 구축하고 기본 UI와 데이터베이스 연동을 했습니다. 그 이후로는 각자 역할을 맡아서 로그인세션부분, 챗봇, 그리고 tab1, tab3인 글쓰기-글보기로 나눠 구현을 시작했습니다. 하다가 막히는 부분은 서로 역할을 바꿔가며 문제를 해결했고 UI와 마이페이지같은 경우 모든 팀원이 함께 참여하여 앱의 완성도를 높였습니다.

1. 프로젝트 구성도



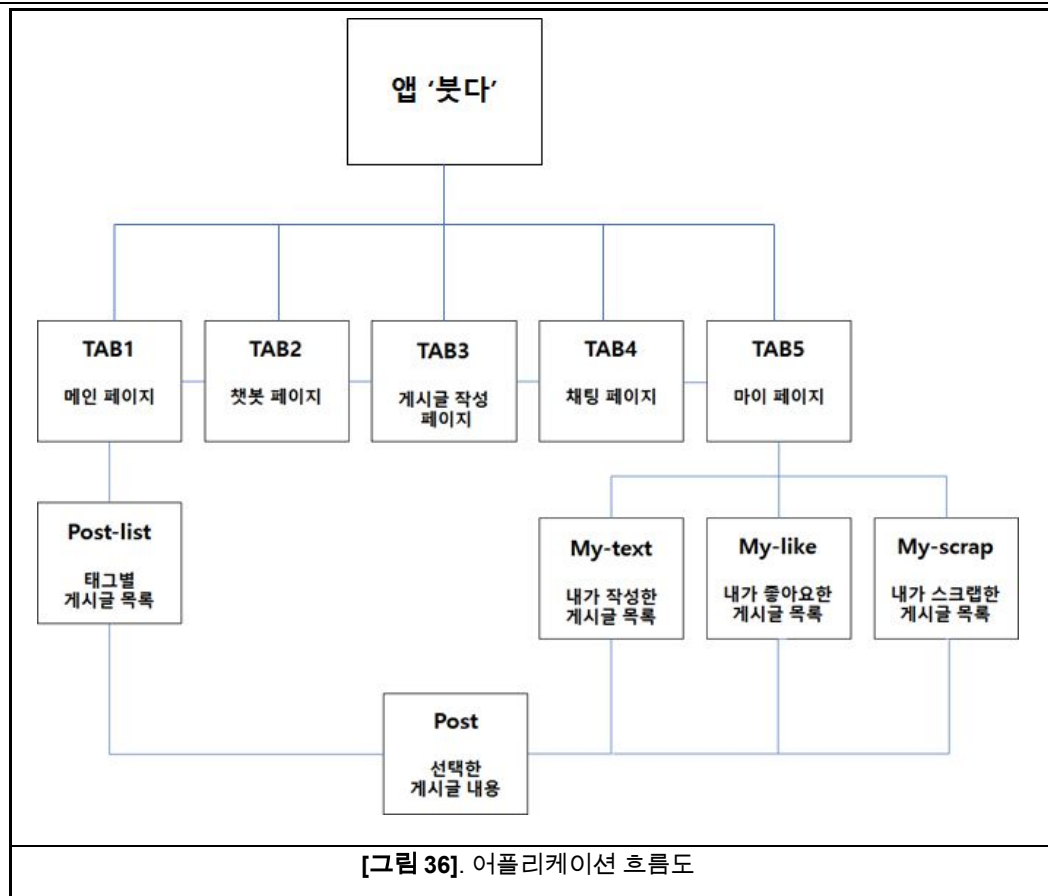
[그림 35]. 프로젝트 구성도

우선 저희는 웹과 앱에서 이용이 가능하도록 하이브리드 앱 개발 플랫폼인 아이오닉을 사용해 앱을 개발했습니다. 사용한 백엔드는 챗봇을 위한 DialogFlow와 데이터베이스인 firebase입니다.

데이터베이스로 firebase를 선택한 이유는 사용자 인증 시스템을 지원하고 서버가 필요없는 앱 개발 플랫폼이기 때문입니다. 또한 실시간 데이터베이스가 이용 가능해서 쉽게 데이터를 읽고 쓸 수 있어서 게시글 작성 및 채팅에 적합한 데이터라고 생각했습니다.

챗봇 프레임워크로 DialogFlow를 선택한 이유는 한국어가 지원되는 프레임워크이기 때문입니다. DialogFlow는 대화에 있어 각각의 단계를 뜻하는 intent를 설정해 대화 흐름을 구성하고, 사용자의 질문에 관한 답에 필요한 데이터는 firebase에서 직접 읽어와 답할 수 있도록 했습니다.

2. 어플리케이션 흐름도



저희의 ‘붓다’ 앱은 다음과 같이 크게 5가지의 탭으로 구성되어 있습니다. TAB1은 메인 페이지로 게시글을 해시 태그로 분류하여 보여주는 페이지이고, 그 중 한 태그를 선택할 경우 해당 태그에 알맞는 게시글 목록들이 보여지는 Post-list 페이지로 넘어갑니다. 또한 그 중 하나의 게시글을 선택할 경우 해당 게시글 내용이 나오는 Post 페이지로 넘어갑니다.

TAB2는 학생들의 질문에 답을 해주는 붓다가 있는 챗봇 페이지입니다. TAB3는 게시글을 작성하는 페이지이고, TAB4는 채팅을 위한 페이지, 그리고 마지막으로 TAB5는 마이페이지입니다. TAB5에서는 내가 쓴 글/좋아ю한 글/스크랩한 글 버튼을 선택함에 따라 각각 My-text/My-like/My-scrap 페이지로 연결됩니다.

3. Future Work

이 프로젝트를 더 발전시킬 수 있는 방향은 다음과 같다고 생각했습니다.

먼저 현재 ‘붓다’ 앱은 사용 대상이 소프트웨어 학과 학생들로 한정되어있습니다. 따라서 게시글 작성 시 사용한 언어, 프로젝트 실행 환경 등 데이터 내용이 한정되어있습니다. 이를 프로젝트가 있는 모든 학과에 적용해 사용할 수 있도록 학과별 특성에 따라 게시글 작성시 저장해야 될 데이터 내용들을 수정해야합니다.

다음으로 현재 챗봇에는 대화 흐름이 6개밖에 되지 않습니다. 기존에 계획했던 대화 흐름 중 하나인 ‘사용자가 좋아요 한 글을 분석해 관련 태그들을 기반으로 프로젝트 주제

	<p>추천해주기' 같은 흐름은 DialogFlow에서 매개변수를 받아서 그에 따라 다른 대화를 유도해 나가는 것이 지원되지 않기 때문에 구현하지 못했습니다. 따라서 DialogFlow가 아닌 다른 챗봇 플랫폼들을 검색해 본 후 다양한 플로우를 짤 수 있도록 지원해주는 플랫폼을 선택해서 챗봇의 다양한 대화 흐름을 구성할 수 있을 것입니다.</p>
4.기대효과	 <p>기존의 캡스톤 디자인, 종합설계 프로젝트 결과물들은 위와 같은 책자의 형식으로 되어있습니다. 이런 책자를 보면서 아이디어를 얻기도 했지만 휴대성이 떨어진다는 단점이 있었습니다. 또, 책자로 보았을 때 잘 이해되지 않는 내용들과 더 궁금한 내용에 대한 대답도 얻을 수 없었습니다. 저희가 개발한 봇다는 책자를 제작하는데 드는 비용을 절감할 수 있고, 모바일 소프트웨어인 앱을 이용하기 때문에 휴대성의 향상 또한 기대할 수 있습니다. 또, 봇다라는 하나의 네트워크 안에서 동국대 공대생들은 프로젝트에 대한 물음과 대답을 채팅기능을 통해 편하게 얻을 수 있습니다. 이러한 학생들 간의 활발한 소통을 통해, 단순히 한 팀의 프로젝트가 시작하고 끝나는 것이 아니라 더욱 좋은 프로젝트 결과물이 나올 것을 기대합니다.</p>
5.구현과정에서의 문제점 및 해결과정	<p>저희가 직면했던 문제는 크게 3가지로 나눌 수 있습니다.</p> <ol style="list-style-type: none"> 1. typescript에서 firebase에 접근해서 데이터 가져오기 <p>먼저 저희는 새로운 언어인 typescript와 새로운 데이터베이스인 firebase를 사용했기때문에 typescript에서 firebase에 접근해 데이터를 가져오는 과정에서 많은 어려움을 겪었습니다. 특히 valueChange()함수를 사용해서 원하는 데이터를 가져와서 typescript 코드안에서 사용하고 싶을 때, 데이터를 가져오는 과정과 이를 사용해서 typescript에서 사용하는 과정이 작성했던 코드 순서대로 실행되지 않아서 오류를 찾는데</p>

굉장히 오래 걸렸습니다.

2. 이미지 업로드

다음으로 저희가 직면한 문제는 게시글 작성시 이미지를 업로드하는 과정입니다. 게시글을 작성할 때 이미지를 함께 업로드 할 경우, 이미지가 firebase storage에 저장되는데, 이 이미지를 게시글 페이지에서 보여주기 위해서는 downloadURL이 필요했습니다. 이 downloadURL이 평소에 익숙한 배열이나 스트링의 형태가 아닌 프로미스라는 새로운 객체의 형태로 반환되었기 때문에 객체 안의 데이터를 찾아내는 과정이 어려웠습니다.

3. 챗봇에서 대화를 주고 받는 방식 선택

챗봇과 사용자가 대화를 주고 받는 방식도 직접 대화로 주고받는 방식, 챗봇에서 몇개의 선택지를 준 후 사용자가 선택해서 대화를 주고받는 방식 등 여러가지가 있어서 하나의 방식을 선택하는 데까지 많은 시간이 걸렸습니다. 버튼형식으로 사용자의 대화를 유도해내는 방식을 선택하고도 ionic에서 DialogFlow에 버튼 형식의 대답이 가능한 UI를 제공해주지 않았기 때문에 대화 흐름을 원하는 방향으로 유도해나가는 문제에 직면했습니다.

다음은 저희가 직면했던 문제들을 해결해나가는 과정입니다.

1. typescript에서 firebase에 접근해서 데이터 가져오기

저희는 이 문제를 valueChange()함수와 snapshot을 이용해 해결했습니다. 이 방법들은 firebase에 저장되어있는 데이터들 중 내가 원하는 데이터를 가리키는 reference 방식을 이용해 데이터에 접근한 후, 그 데이터 값이 변경될 때마다 가져오는 방법입니다.

또한 작성한 코드 순서대로 실행하게 하기 위해서는 valueChange()및 snapshot을 이용해 데이터를 가져오면 그 함수 안에서 가져온 데이터를 모두 처리하도록 코드를 작성했습니다.

2. 이미지 업로드

프로미스 형태의 downloadURL을 이행상태로 만들어 처리 결과 값을 반환 받도록 하여 해결했습니다.

3. 챗봇에서 대화를 주고 받는 방식 선택

처음에 인기많은 게시글/프로젝트 주제 추천/앱설명 버튼을 사용해 사용자의 첫 대화 흐름을 유도했습니다. 이후 사용자의 대답에 따라 챗봇이 답을 준 후, 챗봇의 대답에 따라 다른 버튼이 나타나도록 직접 구현하여 이후 사용자의 다음 행동을 특정 경우로 한정시켰습니다. 이를 통해 저희가 구현해 놓은 흐름대로 사용자와 챗봇이 대화를 이어

	나갈 수 있도록 했습니다.
6.참고문헌	<ol style="list-style-type: none"> 1. 아이오닉 Docs : https://ionicframework.com/docs/api 2. firebase Docs : https://firebase.google.com/docs/guides?hl=ko 3. Dialogflow Docs: https://dialogflow.com/docs