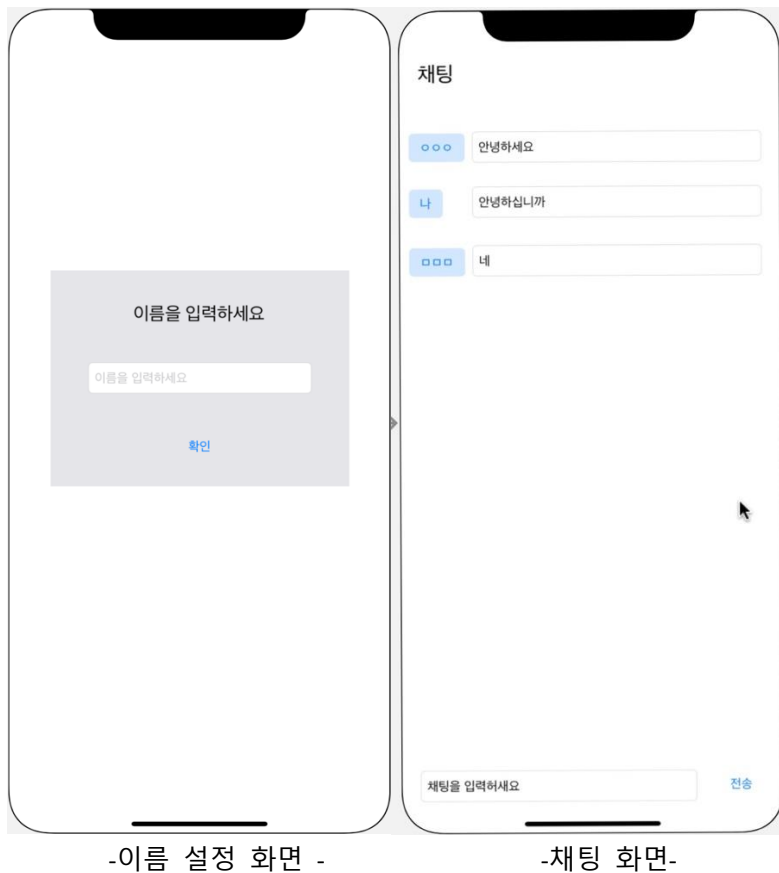


1. 앱 개요 및 구조

개요: 다른 기기와 메시지를 공유할 수 있는 어플리케이션

구조:



2. 앱 기능

이름 설정 화면: 이 화면에서 이름을 입력한 후 확인버튼을 누르면 preferences에 이름을 저장한다.

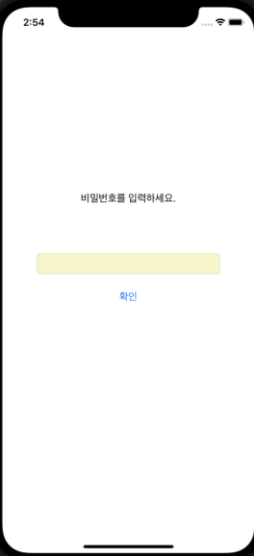
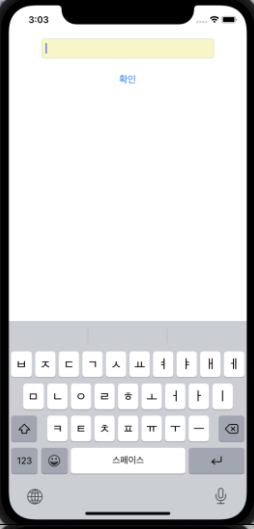

채팅 화면: 하단 텍스트필드에 메시지를 입력하고 전송 버튼을 눌러 메시지를 공유할 수 있다

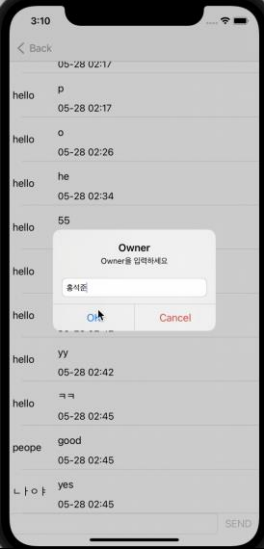


3. 적용 핵심 기술

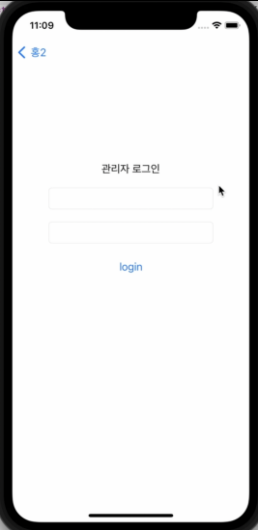

이름 설정 화면: Alert기능을 이용하여 UITextField의 내용을 preferences에 저장한다.

채팅 화면: UITextField의 내용을 UIButton이 눌릴 때 DB에 저장하고 DB와 Group 그리고 View의 상호 작용으로 ViewController를 갱신한다.

4. 구현 내용

<p>1-1</p>		<p>AuthViewController</p> <p>핵심기능</p> <p>FireBase 의 Authentication 기능을 수정</p> <p>ID:123456/pw:123456 을 미리 firebase 에 생성한 뒤</p> <p>signIn 과정에서 PW 를 '123456'으로 고정시켜 비밀번호인증을 구현</p> <p><code>Auth.auth().signIn(withEmail: email, password: "123456")</code></p>
<p>1-2</p>		<p>AuthViewController</p> <p>상세기능 (키보드)</p> <p>2. appdelegation 안에 UINavigationController extension 하여 키보드 hide 함수를 구현한 후 해당 UINavigationController 에서 호출하여 외부 터치시 키보드 hide 를 적용한다</p>
<p>1-3</p>		<p>AuthViewController</p> <p>상세기능 (비밀번호 입력)</p> <p>확인 버튼 누르기</p> <p>1.틀린 비밀번호일 경우 반응하지 않는다</p> <p>2.올바른 비밀번호일 경우</p> <p>2-1, 확인 버튼이 사라진다 (전이 딜레이 사이에 여러번 터치하여 여러개의 controller 가 스택에 쌓이는 것을 방지)</p> <p>2.2, pods 를 이용한 progress 모양이 등장한다</p> <p>2-3 이후 다음 controller 로 전이된다</p>

2		<p>Owner 핵심기능</p> <ol style="list-style-type: none"> 1. 최초 접속시 Owner 를 입력받고 해당 textfield 값을 preference 에 저장한다 2. Owner 값을 입력받게 되면 notification 발생. 그후 (chat)viewController 에서 observer 를 이용해 재접속 없이 owner 값이 입력되는 즉시 navigation.tilt 이 갱신, 이와 동시에 system 채팅이 발생하여 owner 가 입장하였다는 메시지 발생 3. SYSTEM 메시지는 owner 값이 SYSTEM 인지 검사를 진행하기 때문에 사용자가 오너값을 SYSTEM 으로 입력한다면 소문자 system 으로 자동변경된다.
3-1		<p>(chat)viewController 핵심기능</p> <ol style="list-style-type: none"> 1. 좌측 상단에 navigationController.popViewController 를 이용해 비밀번호 창으로 돌아가는 로그아웃 버튼을 생성 2. 하단 textField 의 값을 입력하고 SEND 버튼을 누르면 firebase/firestore 에 메시지가 저장되고 notification 을 주고받아 모든 기기에서 tableView 에 갱신된다 3. tableView Delegation 에서 현재 기기의 owner 와 tableView cell 의 owner 가 같으면 자신이 보낸 메시지이므로 text Alignment 우측으로 변경하고 owner 를 표시하지 않는다 4. database 로부터 오는 notification 을 핸들링하는 함수에 tableView reloadData 후 scrollToRow 함수를 이용하여 메시지를 보내게되면 자동으로 맨아래로 스크롤한다
3-2		<p>(chat)viewController 상세기능 (키보드)</p> <ol style="list-style-type: none"> 1. 키보드 나타남, 사라짐 시 view 화면 올리기 내리기 <pre>if let keyboardFrame: NSValue = notification.userInfo?[UIResponder.keyboardFrameEndUserInfoKey] as? NSValue { let keyboardRectangle = keyboardFrame.cgRectValue let keyboardHeight = keyboardRectangle.height 키보드의 높이를 알아낸 뒤 UIView.animate(withDuration: 1) { self.view.window?.frame.origin.y -= keyboardHeight } } </pre> <p>현재 뷰의 origin.y 에서 키보드 높이 만큼 감소/증가 시킨다 이외의 keyboardbool: bool 변수를 이용해 키보드 appear 함수가 두 번 호출되는 것을 방지한다</p> 2. 외부 터치시 키보드 숨기기 1-2 와 동일하여 생략합니다.

4-1		<p>AdminAuthUIViewController</p> <p>핵심기능</p> <p>Firebase 인증기능을 이용한 로그인,</p> <ol style="list-style-type: none"> 1.초기 비밀번호 입력과 다른 로그인을 진행하기 위해 비밀번호를 담당하는 사용자 계정은 인증으로서 기능을 못하도록 구현 2.현재 관리자계정에 로그인되어 있다면 로그인을 거치지 않고 관리자화면으로 바로 전이
4-2		<p>AdminViewController</p> <p>핵심기능</p> <ol style="list-style-type: none"> 1. 채팅 전체 삭제하기 전이마다 데이터를 전송하고 해당 viewController 는 전송받은 데이터를 이용해 채팅을 모두 삭제하고 갱신한다. 2. 시스템 메시지 발송하기 Owner 값을 SYSTEM 으로 하여 메시지를 업로드 후 갱신하면 (chat)Viewcontroller 에 tableView Delegation 에서 cell 을 시스템 형식으로 구현하여 준다