

컴퓨터공학 프로젝트 최종보고서

팀 명	우리 이쁘조♡				
작 품 명	CEP TALK				
개발기간	2016년 11월 3일 ~ 2016년 12월 8일				
지도교수	컴퓨터공학과			정 진우 교수님	
구 분	학년	학 번	성 명	휴대전화	E-mail
책임자(팀장)	3	20140350	김 한설	01062396318	gkstjf22@naver.com
팀원1	3	20140998	이 정은	01099371194	wjddms6769@naver.com
팀원2	3	20141280	진 연주	01066507951	aurora32s@naver.com
팀원3	3	20141337	최 은선	01049968345	chldmstjs96@naver.com
<p>본인은 컴퓨터공학 프로젝트 심화과제 최종 보고서를 첨부와 같이 제출합니다.</p> <p style="text-align: right;">2016년 12 월 13 일</p> <p style="text-align: right;">팀 장 김 한설 (서명)</p> <p style="text-align: center;">컴퓨터공학심화프로그램</p>					

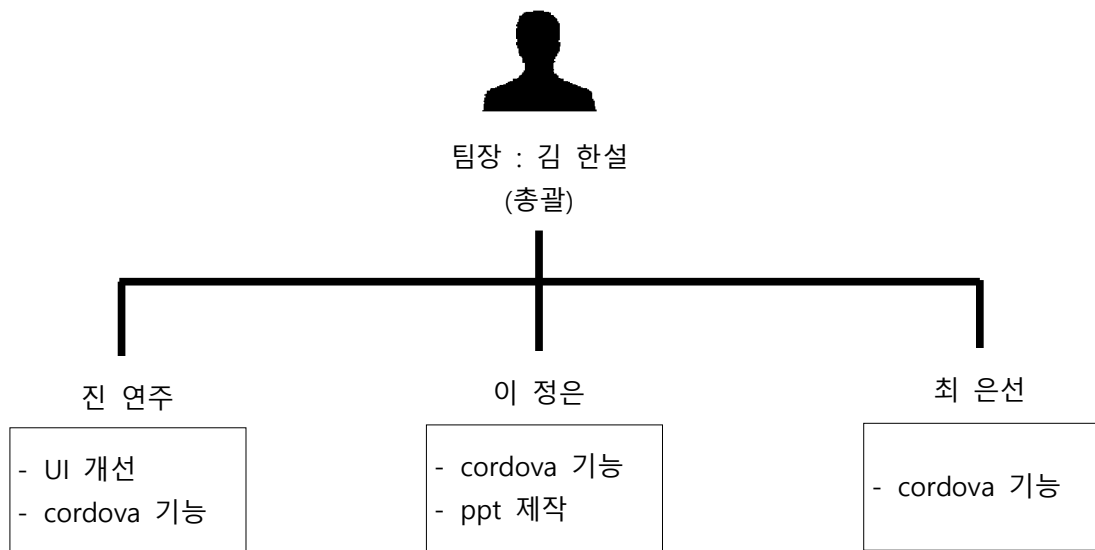
CEP TALK

1. 개 요

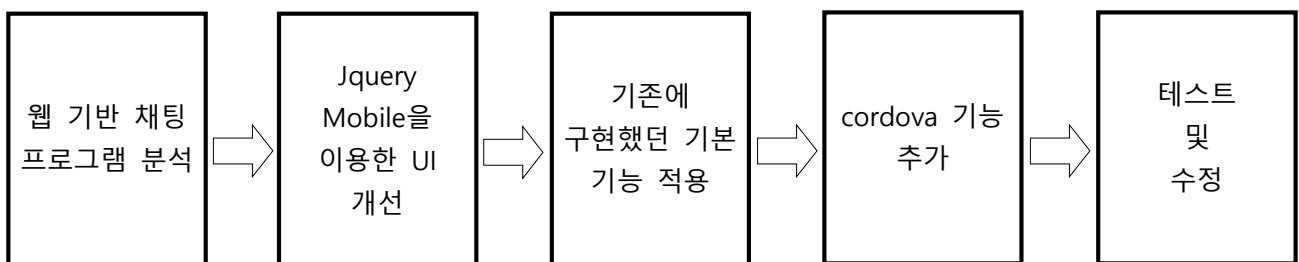
기존의 웹으로 구현한 채팅 프로그램을 모바일 환경에 최적화 되도록 UI / UX를 개선하고 Cordova에서 지원하는 플러그인을 이용하여 여러 기능을 제공하는 하이브리드 어플리케이션 개발

2. 작품개발 방법

2-1. 역할 분담



2-2. 개발 절차



1. 웹으로 구현했던 채팅프로그램 분석 및 기능 정리
2. 분석한 내용을 토대로 JQuery Mobile을 이용해 모바일 환경에 최적화되도록 UI 개선
3. 개선된 UI에 기존에 구현했던 기본 기능들을 적용
4. Cordova plugin을 이용하여 기능 추가
5. 테스트 및 수정
6. 코드 정리

3. 개발내용

3-1. 설계구성요소 및 설계제한 요소

설계 구성요소						설계 제한요소						
목표 설정	합성	분석	구현/ 제작	시험/ 평가	결과 도출	성능	규격/ 표준	경제 성	미학	신뢰성	안정성/ 내구성	환경
○		○	○	○	○	○	○			○	○	

-**목표 설정** : 기존의 웹 기반으로 구현된 채팅 프로그램을 모바일 환경에 최적화 되도록 UI / UX를 개선하고 Cordova 플러그인을 이용한 여러 기능들을 제공하여 사용자가 안드로이드 폰으로 다른 사용자와 여러 정보와 대화를 주고받을 수 있는 하이브리드 어플리케이션 개발한다.

-**분석** : 기존에 존재하는 채팅 어플리케이션들이 사용자들에게 지원해주는 여러 기능들과 use case들을 분석하여 사용자에게 필요하다고 생각된 기능들을 use case 별로 정리 후 개발할 어플리케이션의 기능 및 구조를 확정한다.

-**구현/제작** : 개발 도구 간편한 코드 수정과 개선을 위해 notepad ++을 활용한다.

-**시험/평가** : 어플리케이션의 기능과 성능을 평가하기 위하여 기능을 추가할 때마다 어플리케이션을 다운로드하여 다른 기능에 영향을 끼치지 않는지 확인하고 여러 사용자를 서버에 접속시켜서 기능들을 실행한다.

-**결과 도출** : 최종 결과물은 node.js 기반 채팅 회원 및 관리 서버 시스템, 하이브리드 앱 기반의 안드로이드 폰 용 클라이언트 어플리케이션 등이다.

-**성능** : 작품의 성능 요소와 목표로 하는 성능은 다음과 같다.

- (1) 동시 접속자 수 : 20명
- (2) 메모리 사용량 : 4G
- (3) 실행 속도 : 채팅 프로그램 특성상 전송된 파일이나 데이터는 바로 바로 업데이트 될 수 있을 만큼

-**규격/표준** : 본 프로그램은 브라우저 종류에 관계없이 모든 웹 환경에서 동작한다. 스마트폰 환경에서는 안드로이드 플랫폼에서 동작 가능하다.

-**신뢰성** : 로그인 시 사용자가 입력한 내용과 서버의 저장된 내용이 항상 일치해야만 채팅창으로 들어갈 수 있다. 사용자의 입력이 잘못된 경우나 고의적인 오류 입력의 경우에도 적절한 오류를 alert창으로 띄어줌으로써 무결성 및 신뢰성이 보장되도록 한다. 또한 사용자가 입장/퇴장 시에 적절한 처리를 하면서 사용자들에게 정확한 정보들을 제공한다.

-**안정성/내구성** : 로그인 시 사용자의 입력이 잘 못된 경우나 고의적인 오류 입력의 경우 적절한 오류 처리를 통해서 어플리케이션 실행이 중단되지 않는다. 데이터베이스 변동 시에 오류가 발생하지 않도록 클라이언트와 서버 간의 정보 교환을 일치시킨다.

3-2. 기능 리스트

■ Improvements

① 로그인 및 회원가입

- 웹 채팅프로그램

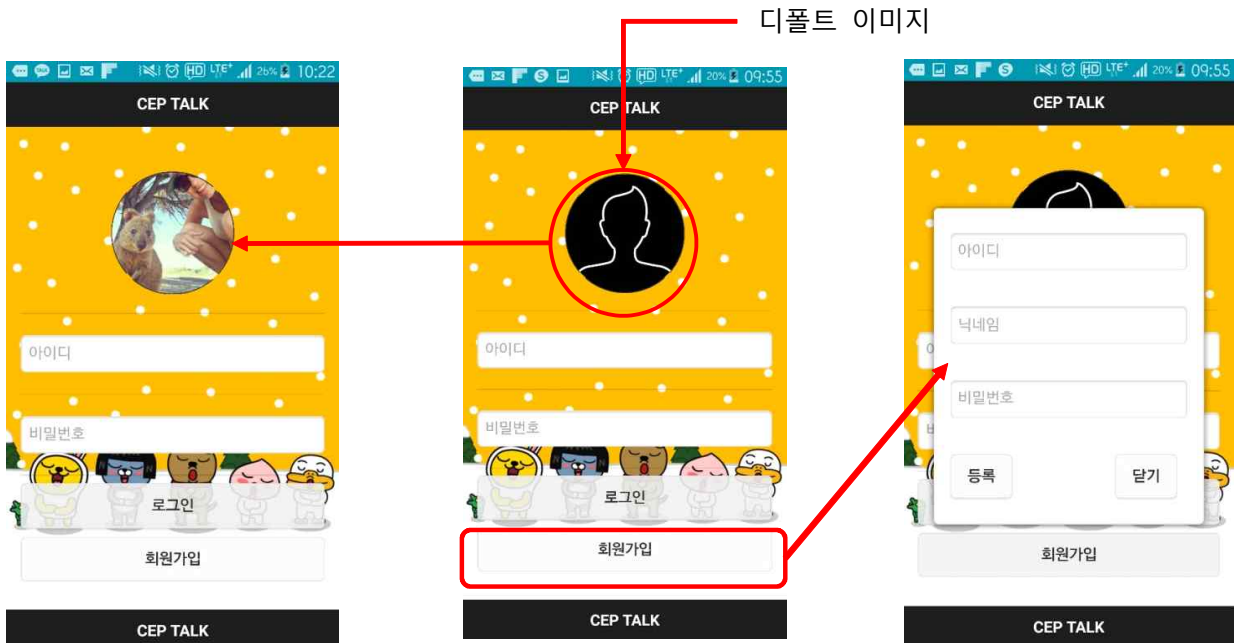
[좌 - 로그인화면 / 우 - 회원가입화면]

The image shows two web browser windows side-by-side. The left window is the login screen, titled 'CEP TALK', with input fields for '아이디' (ID) and '비밀번호' (Password), and buttons for '로그인' (Login) and '회원가입' (Sign Up). The '회원가입' button is highlighted with a red box. The right window is the registration screen, also titled 'CEP TALK', with input fields for '아이디' (ID), '닉네임' (Nickname), and '비밀번호' (Password), a search bar with the text '찾아보기... 선택한 파일이 없습니다.' (Search... No files selected), and a '등록' (Register) button. A red arrow points from the '회원가입' button in the left window to the registration screen in the right window.

기존 웹 채팅프로그램에서는 회원가입 버튼을 클릭 시 새로운 창으로 이동해서 아이디, 닉네임, 비밀번호, 프로필 이미지를 입력받았다.

- 모바일 채팅프로그램

[좌- 프로필사진 설정 화면 / 가운데- 기본 화면 / 우- 회원가입 팝업창을 활성화 시킨 화면]

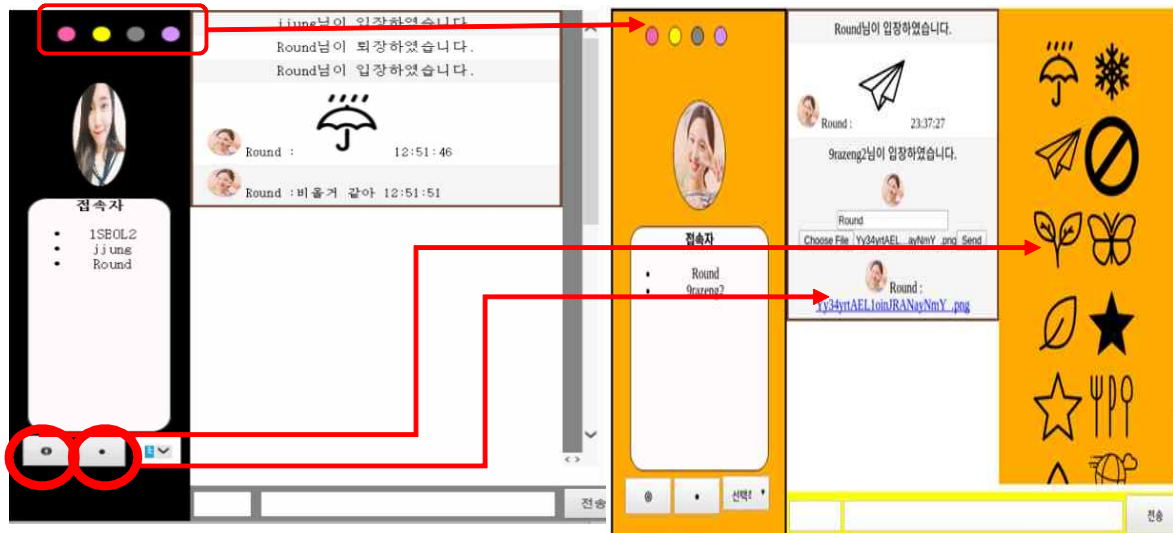


모바일 채팅프로그램에서는 어플리케이션을 실행하면 로딩화면을 4초간 보여주고 로그인페이지로 이동. 로그인 페이지에서 회원가입을 누를시 새 창으로 넘어가던 부분을 팝업으로 표시하도록 변경하였다. 또한 회원가입에서 아이디, 닉네임, 비밀번호만을 입력받고 프로필 사진을 변경하고 싶다면 로그인페이지에서 프로필 사진을 변경할 수 있도록 하였다. (변경하지 않는다면 디폴트 이미지를 사용한다.)

② 채팅방 UI

-웹 채팅프로그램

[좌- 채팅UI의 기본화면 / 우- 이모티콘 패널 활성화 및 파일전송 화면]



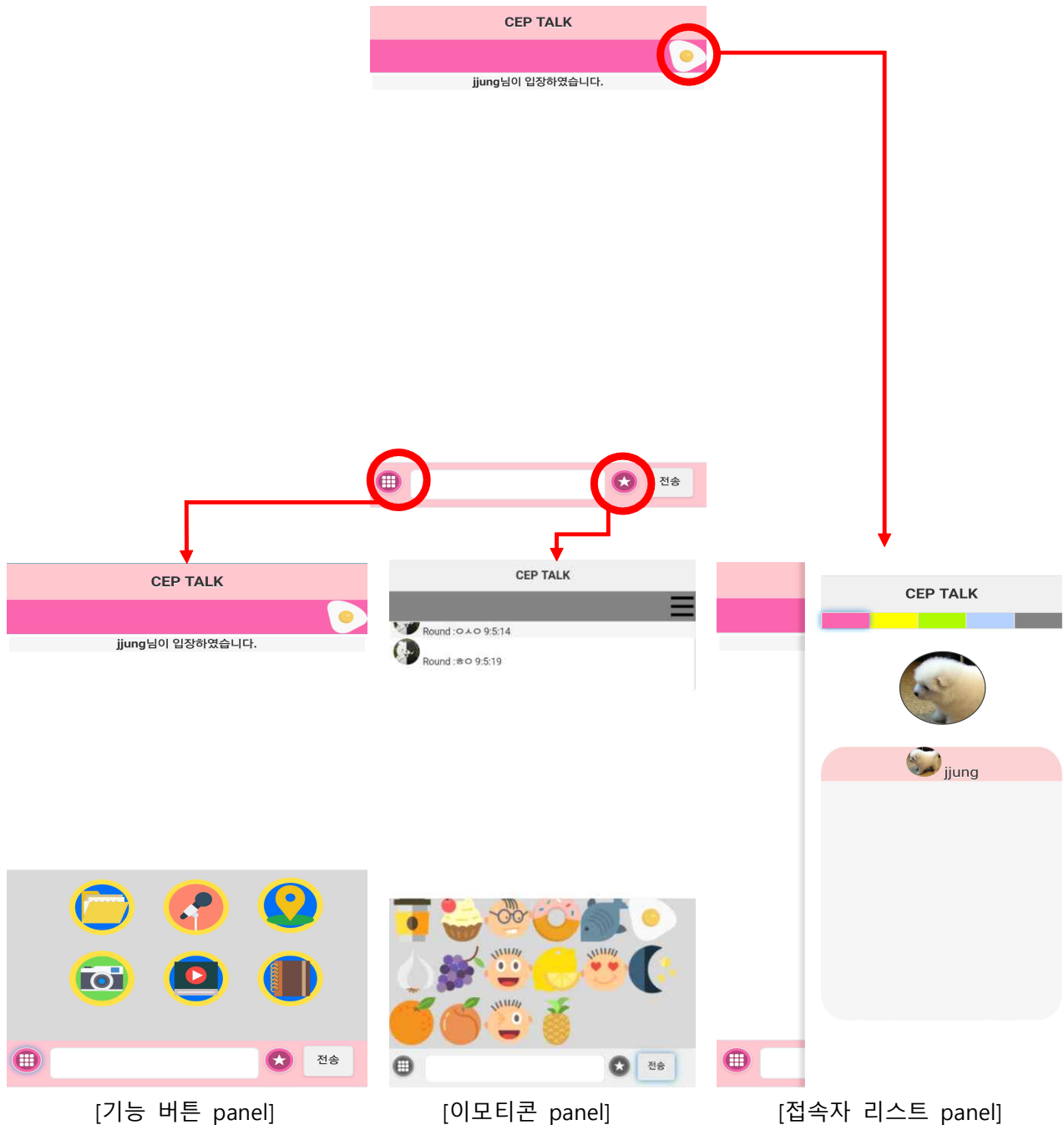
웹 채팅프로그램에서는 기본화면에서 테마 변경과 접속자리스트 등을 확인할 수 있고 아래 버튼들 중 ☺ 버튼을 클릭시 오른쪽에서 이모티콘을 선택할 수 있는 패널 창이 활성화되며 ● 버튼을 클릭 시 파일 전송을 할 수 있도록 하였다.

-모바일 채팅프로그램

[테마 별 채팅창 UI]



[기본 채팅 화면]





-jQuery mobile을 사용하여 모바일 환경에 최적화

모바일 채팅프로그램에서는 기본화면에서는 채팅방의 대화창만 볼 수 있도록 하였다. 좌측 하단 버튼 클릭 시 추가기능들을 실행할 수 있는 버튼들이 있는 창이 활성화되며 각 버튼들은 해당 이미지에 맞는 기능들을 실행시킨다. 우측 상단 버튼을 클릭하면 테마 색 변경, 프로필 사진 변경, 접속자리스트를 볼 수 있는 패널 창이 활성화되며 테마 색 변경 시 패널 창을 활성화 시키는 버튼의 이미지 또한 변경된다. 우측 하단 버튼을 클릭 시 이모티콘을 사용할 수 있는 창이 활성화된다.

■ Challenges

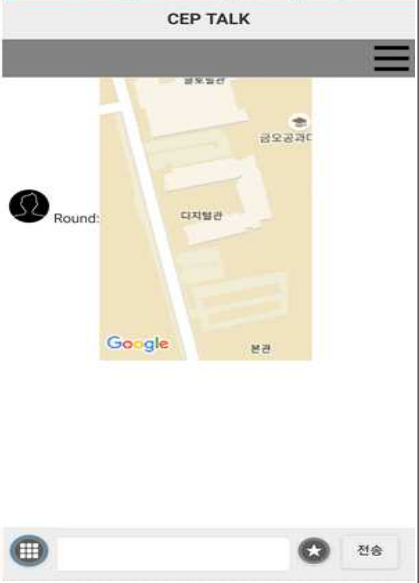
기능 수 : 16개

<p>application icon</p>	<p>[어플리케이션의 기본 아이콘을 변경]</p> <p>-어플리케이션 이미지 변경 /platforms/android/res/mipmap 안에 있는 icon이미지를 자신이 원하는 이미지로 변경</p> <p>-어플리케이션 이름 (1) config.xml 에 있는 <code><name>"Name"</name></code> Name에 자신이 원하는 어플리케이션 이름을 적어준다.</p> <p>(2) /platforms/android/res/values에 있는 string 파일 안에서 <code><string name="app_name">"Name"</string></code> Name에 자신이 원하는 어플리케이션 이름을 적어준다.</p> 
<p>splash screen</p>	<p>[어플리케이션 실행 시 로딩화면을 띄워줌]</p> <p>(1) plugin 추가 cordova plugin add cordova-plugin-splashscreen</p> <p>(2) config.xml <code><preference name="SplashScreen" value="splash" /></code> <code><preference name="SplashMaintainAspectRatio" value="true" /></code> <code><preference name="SplashScreenDelay" value="3500" /></code> - splash screen을 유지할 시간을 3500으로 설정 - 너무 작게 설정하면 splash screen과 채팅 화면 간에 빈 화면이 나오는 term이 생김.</p> <p>(3) navigator.splashscreen.hide();</p>
<p>battery- status</p>	<p>[배터리가 일정량 이하로 떨어지면 배터리 손실을 최소화하기 위해 사진촬영, 동영상 촬영 등 기능사용을 제한함.]</p> <p>(1) plugin 추가 cordova-plugin-battery-status</p>

battery-status	<pre> (2) window.addEventListener('batterystatus',function(status){ var now=5; var fileBtn = document.getElementById("file"); if(status.level<now){ if(status.level==now-1){ navigator.notification.alert("Battery Level Low! you can't use it! try charge your device!",alertDismissed , '알림','닫기'); function alertDismissed(){ } fileBtn.className = "ui-state-disabled"; } else{ fileBtn.className = "ui-state-enabled"; } },false); </pre>  <p>- 기능 확인을 편하게 하기 위해 변수 now를 이용하여 원하는 배터리 퍼센트를 설정할 수 있게 함. 지정한 값보다 작은 숫자가 되면 최초 한번만 알림을 띄워주도록 함. 충전을 통해 정해진 값보다 커 질 시 다시 기능을 사용할 수 있음. 버튼 비활성화 활성화는 각각 "ui-state-disabled", "ui-state-enabled" 사용.</p>
vibration & sensor	<p>[채팅방에 유저가 입장/퇴장 시 진동으로 알림] [디바이스 화면을 바닥으로 향하도록 뒤집을시 진동 비활성화]</p> <p>(1) plugin 추가 cordova plugin add cordova-plugin-vibration cordova plugin add cordova-plugin-device-motion</p> <pre> (2) function vibrate(){ navigator.accelerometer.getCurrentAcceleration(onSuccess); -가속도계(x,y,z축을 기준으로 변경 내용 감지) 기능 function onSuccess(acceleration) { if(acceleration.z > -9.6) //디바이스 화면을 뒤집을시 navigator.vibrate(1000);//1초동안 진동 } } </pre> <pre> (3) socket.on('adduser',function(nickname){ ... vibrate(); ... }); socket.on('subuser',function(nickname){ ... vibrate(); ... }); </pre> <p>- 가속도계의 z축을 이용함. 지구의 중력가속도에 의해서 디바이스를 뒤집어 놓았을</p>

	<p>시 -9.6 안팎으로 z축이 변하므로 acceleration.z를 -9.6보다 클 때만 진동이 1초간 울리게 설정함. 입장(adduser), 퇴장(subuser)시에 vibrate가 실행되도록 함.</p>
<p>Volumekey로 textsize 조절</p>	<p>[개인의 시력에 따라 채팅방의 폰트크기를 조절하여 눈을 보호하기 위함.] [대화 도중 한 화면에 많은 내용을 담아서 screen shot을 원할 때 폰트 크기를 작게 해서 내용을 많이 보이게 하면 유용하게 쓰일 수 있음.]</p> <p>(1) volumeupbutton과 volumedownbutton의 이벤트리스너 추가.</p> <pre>function onDeviceReady() { document.addEventListener('volumeupbutton',textSizeUp); document.addEventListener('volumedownbutton',textSizeDown); }</pre> <p>(2) function textSizeUp() {</p> <pre> var up = document.getElementById("messages"); var currentSize = up.style.fontSize; var num = parseFloat(currentSize); var unit = currentSize.slice(-2); if(num<14){ num*=1.1; var changedSize = num + unit; up.style.fontSize=changedSize; } else{ alert("maximum FontSize");} }</pre> <p>function textSizeDown() { //textSizeUp()과 방법 동일</p> <pre> if(num>8){ num/=1.1; } else{ alert("minimum FontSize"); } }</pre> <p>- 채팅창의 폰트사이즈를 받아오게 되면 단위까지 받아와짐. 이를 parseFloat를 이용하여 사이즈의 숫자만 받아오고, slice(-2)를 이용하여 폰트사이즈에서 뒤의 두 글자, 즉 단위인 'pt'도 따로 받아옴. 받아온 숫자의 크기를 증가, 감소 시켜 조절하고 단위를 붙여 사이즈를 변경해주는 방식임.</p> <p>- 일정 크기보다 크거나 작을 시에는 alert 창을 띄우고, 크기 변경은 없음.</p>
<p>Geolocation & Google map</p>	<p>[Geolocation으로 현재 자신의 위치를 위도와 경도로 받아와서 구글 맵 상에서 확인하고 확인한 자신의 위치를 채팅방에 이미지 형식으로 업로드 하여 다른 사용자에게 자신의 위치를 알려주는 기능]</p>

<p>Geolocation & Google map</p>	<p>(1) plugin 추가 cordova plugin add cordova-plugin-geolocation</p> <p>(2) API키 추가 cordova plugin add https://github.com/phonegap-googlemaps-plugin/cordova-plugin-googlemaps --variable API_KEY_FOR_ANDROID=" key " --variable API_KEY_FOR_IOS=" key "</p> <p>- key 부분에 google map에서 받아온 API 키를 추가해준다 -</p> <p>(3) 구현 - 구글맵 나타내기 <pre><script src="http://maps.google.com/maps/api/js" type="text/javascript"> </script> map = plugin.google.maps.Map.getMap(div); map.addListener(plugin.google.maps.event.MAP_READY,onMapReady);</pre> </p> <p>★ 자신의 위치 찾아가기</p> <pre>navigator.geolocation.getCurrentPosition(); map.animateCamera({ target:{lat:position.coords.latitude,lng:position.coords.longitude}, // 현재 자신의 위치 (위도, 경도) animation: plugin.google.maps.Animation.BOUNCE</pre> <p>- 기본 화면에서 자신의 위치까지 애니메이션 효과를 주며 이동하게 함.</p> <p>★ 채팅방에 이미지로 업로드하기</p> <pre>\$('#sendMap').on('click',function() { map.toDataURL(function(imageData) { uploadImage(imageData); }); });</pre> <p>- 구글맵 이미지를 screen shot해서 채팅창에 업로드하기 위해 screenshot 플러그인을 사용하려 하였으나 구글 맵의 transparent 성질 때문에 구글 맵을 제외한 부분만이 캡처 되는 등 난항을 겪었습니다. map option에서 back-ground에 색을 지정해보기도 하고 교수님께서 조언해주셨던 z-index로 위치를 조절하는 등 많은 방법을 시도하였으나 생각대로 되지 않았습니다. 며칠 동안 검색을 이용해 방법을 찾던 중 구글 맵 자체에서 toDataURL이라는 함수를 지원하는 것을 알게 되어 성공할 수 있었습니다.</p>
---	---

<p>Geolocation & Google map</p>	<p>★ toDataURL : 현재 화면의 띄어진 지도 이미지를 URL로 반환해줌.</p> <p>-반환된 URL을 사용하여 서버로 이미지 전송</p> <pre>function uploadImage{ ... var ft=new FileTransfer(); ft.upload(imageURL,encodeURI(</pre>  <p>[구글맵 화면을 채팅창에 업로드]</p>
<p>weather information</p>	<ul style="list-style-type: none"> - 사용자의 위치를 plugin geolocation을 사용하여 위치정보를 확인 후 사용자가 위치한 곳의 날씨의 정보를 json형식으로 불러옴. - openweathermap사이트의 URL에 lat와 lon에 위도와 경도정보를 입력하고 appid에 사이트에서 부여받은 API key값을 입력하면 사용자의 날씨 정보를 json형식으로 받아옴. - ajax를 사용하여 json형식으로 파일 불러오기가 성공을 하면 날씨의 정보에 따라 미리 지정해 두었던 메시지(문구)를 출력시키고 나의 현재 위치의 날씨 이미지를 출력해 주는 기능 <p>(1)plugin 추가 cordova-plugin-geolocation</p> <p>(2)function getWeather(){ navigator.geolocation.getCurrentPosition(onSuccess,onError, {enableHighAccuracy: true}); function onSuccess(position) { var latitude = position.coords.latitude; var longitude = position.coords.longitude; var weather = 'http://api.openweathermap.org/data/2.5/forecast?lat='+latitude+'&lon='+longitude+'&appid='+OpenWeatherAppKey; \$.ajax({ dataType: "jsonp", url: weather, success: function(data){ var weather=data.list[0].weather[0].icon; if(weather == "01d" weather == "01n") \$("#text").text("오늘은 데이트가기 좋은 날씨~~!"); </p>


weather information	<pre> else \$("#text").text("눈앞이 안보이다~~~까가까가"); var url = "http://openweathermap.org/img/w/" + weather + ".png"; \$("#temp").attr("src", url); } }); </pre> <p>- openweathermap 사이트에서 받아온 json에서 날씨에 대한 정보를 가져옴. if문을 사용하여 날씨에 따라 문구를 띄워줌.</p>
녹음	<p>-plugin meda capture을 사용하여 10초미만의 1개의 파일만을 선택하거나 녹음할 수 있게 옵션을 지정</p> <p>-녹음이 끝난 후 파일을 서버로 upload -서버로부터 전송된 녹음 파일을 append를 사용하여 채팅창에 띄워줌</p> <p>(1)plugin 추가 cordova-plugin-mediga-capture</p> <p>(2) 클릭 이벤트로 녹음 함수를 실행 document.getElementById("mic").addEventListener("click",audioCapture);</p> <p>(3) function audioCapture() { var options = { limit: 1, duration: 10 }; navigator.device.capture.captureAudio(onSuccess, onError, options); function onSuccess(mediaFiles) { for (i = 0, len = mediaFiles.length; i < len; i += 1) { path = mediaFiles[i].fullPath; } uploadAudio(path); } }</p> <p>- 오디오 파일의 URL을 오디오 파일을 서버로 업로드 하는 함수인 uploadAudio()로 전달</p> <p>(4)audio태그를 사용하여 채팅창에 파일 올리기. var profileURL="http://202.31.200.143/"+data.image; fileURL="http://202.31.200.143/"+data.file; \$('#messages').append(' <audio src="'+fileURL+'"' preload="auto" controls></audio>');;</p>

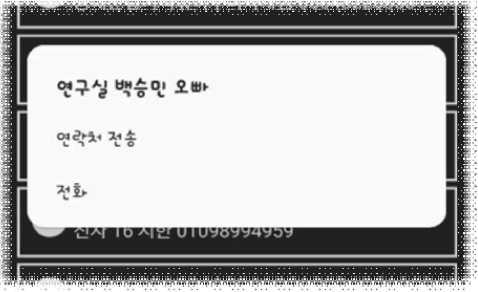
<p>back button 처리</p>	<p>- 로그인 후 채팅창에 들어가서 back button을 누르면 이전 화면인 로그인 화면으로 돌아가게 되지만 접속정보는 그대로 유지되게 됨. (disconnect 이벤트가 발생하지 않음.) 그 상태에서 동일 아이디로 로그인하거나, 다른 아이디로 로그인 하게 되면 그 전의 채팅내용은 그대로 남아있게 되고 접속 정보는 여전히 잘못되어있음. 이 점을 고치기 위해 back button을 눌렀을 때의 disconnect 이벤트 처리를 해줌.</p> <p>(1) plugin cordova-plugin-dialogs</p> <p>(2) function onDeviceReady() { document.addEventListener("backbutton", onBackKeyDown); }</p> <p>(3) function onBackKeyDown() { navigator.notification.confirm('앱을 종료하시겠습니까?', onConfirm, '알림','종료,취소'); navigator.notification.beep(1); function onConfirm(buttonIndex) { if(buttonIndex == 1){ navigator.app.exitApp(); } if(buttonIndex == 2) { } } }</p> <p>- back 키가 눌리면 dialog창과 함께 소리가 발생. 종료를 누르면 어플리케이션이 종료되고 정상적으로 disconnect 됨. 취소를 누르면 채팅창에 머무르게 됨.</p>
<p>동영상 촬영</p>	<p>- 카메라 플러그인에는 사진촬영만 되기 때문에 미디어 capture 플러그인을 사용함. - 기능 창에서 비디오모양 버튼을 누르면 동영상 촬영이 바로 가능함. 촬영이 끝나고 저장을 누르면 서버로 전송. - 서버로부터 전송된 비디오를 채팅창에 띄워줌. - 옵션을 통해 동영상 최대 촬영 시간을 정할 수 있음.</p> <p>(1) plugin cordova-plugin-media-capture</p> <p>(2)onDeviceReady() { document.getElementById("video").addEventListener("click",videoCapture); }</p> <p>(3) function videoCapture{ var list = document.getElementById('messages'); navigator.device.capture.captureVideo(onSuccess, onError, { limit: 1,</p>

동영상 촬영	<pre> destinationType: Camera.DestinationType.FILE_URL,duration: 10}); function onSuccess(mediaFiles) { uploadVideo(mediaFiles[0].fullPath); } function onError(error) {} } </pre> <p>- 비디오 파일의 URL을 비디오 파일을 서버로 업로드 하는 함수인 uploadVideo()로 전달</p> <p>(4) video태그를 사용하여 채팅창에 올리기. 재생 조절가능. var profileURL="http://202.31.200.143/"+data.image; fileURL="http://202.31.200.143/"+data.file; \$('#messages').append(' <video width="320" height="176" controls> <source src ='"+fileURL+" type="video/mp4" > </video> ') </p>
동영상 앨범	<p>- mediaType Name을 VIDEO로 설정하여 앨범을 불러올 때 동영상 파일만 불러옴. sourceType Name을 SAVEDPHOTOALBUM로 설정하여 디바이스의 앨범에서 가져오도록 함.</p> <p>- 선택된 비디오를 서버로 전송</p> <p>- 서버로부터 전송된 비디오를 채팅창에 띄어줌.</p> <p>(1) plugin cordova-plugin-camera</p> <p>(2)function onDeviceReady() { document.getElementById("GetVideo").addEventListener("click",cameraGetVideo); }</p> <p>(3) function cameraGetVideo() { navigator.camera.getPicture(onSuccess,onFail, { quality: 100, destinationType: Camera.DestinationType.FILE_URL, sourceType: Camera.PictureSourceType.SAVEDPHOTOALBUM, mediaType: Camera.MediaType.VIDEO //video만 }); function onSuccess(imageData){ uploadVideo(imageData); } function onFail(message){} }</p> <p>(4) 동영상 촬영과 같은 방식으로 채팅창에 띄어줌.</p>

사진 촬영	<p>[사용자가 사진을 촬영하여 그 결과를 채팅창에 업로드 해주는 기능]</p> <p>(1) plugin cordova-plugin-camera</p> <p>(2)function onDeviceReady() { document.getElementById("camera").addEventListener("click",cameraTakePicture); }</p> <p>(3) function cameraTakePicture(){ //alert("hello"); navigator.camera.getPicture(onSuccess, onFail, { quality: 100, destinationType: Camera.DestinationType.DATA_URI, targetWidth:200, targetHeight:200 }); function onSuccess(imageURL) { uploadImage(imageURL); } function onFail(message) { //alert('Failed because: ' + message); } }</p> <p>- 결과 이미지의 URL을 이미지를 서버로 업로드 해주는 uploadImage()함수로 전달</p> <p>(4)img 태그를 사용하여 채팅창에 올리기.</p> <p>var profileURL="http://202.31.200.143/"+data.image; fileURL="http://202.31.200.143/"+data.file; \$('#messages').append(' ' +data.nickname+ ':'+'> ');</p>
사진 앨범	<p>[사진 앨범에서 채팅창에 전송하고 싶은 사진을 선택하여 다른 사용자들에게 전송하는 기능]</p> <p>(1) plugin cordova-plugin-camera</p> <p>(2)function onDeviceReady() { document.getElementById("GetPhoto").addEventListener("click",cameraGetPicture); }</p>

사진 앨범	<pre> (3)function cameraGetPicture(){ navigator.camera.getPicture(onSuccess,onFail, { quality: 100, destinationType: Camera.DestinationType.DATA_URI, sourceType: Camera.PictureSourceType.PHOTOLIBRARY, mediaType: Camera.MediaType.PICTURE, encodingType: Camera.EncodingType.JPEG, targetWidth:200, targetHeight:200 }); //alert('click'); function onSuccess(imageURL){ uploadImage(imageURL); } function onFail(message){ //alert("Failed because : " + message) } } </pre> <p>- 결과 이미지의 URL을 이미지를 서버로 업로드 해주는 uploadImage()함수로 전달</p> <p>(4) 사진 촬영과 같은 방식으로 채팅창에 띄어줌.</p>
프로필 사진 변경	<p>- 앞에서 설명했던 것처럼 로그인을 할 때 자신의 프로필 사진을 지정하거나 채팅 도중 자신의 프로필 사진을 변경하고 싶을 경우 자신의 프로필 사진을 클릭하면 앨범에서 사진을 선택하는 창이 뜬다.</p> <p>- 원하는 사진을 클릭한 후 확인 버튼을 누르면 자신의 프로필 사진이 변경 되는 기 능</p> <p>(1) plugin cordova-plugin-camera cordova-plugin-crop</p> <pre> (2) function changeLoginProfile(){ var change1 = document.getElementById("profile"); var options = { ... }; navigator.camera.getPicture(onSuccess,onFail,options); function onSuccess(imageURI){ plugins.crop(function success(data){ var change = document.getElementById("my_profile"); change.src = data; change1.src=data; }, function fail(){ </pre>

	<pre> }, imageURI,{quality:100}); image=imageURI; uploadProfile(); } function onFail(message){ //alert("Failed because : " + message) } } </pre> <ul style="list-style-type: none"> - crop plugin을 사용하여 사진을 자신이 원하는 크기만큼 잘라서 사용 가능 - 사진 선택이 완료되면 Image URL을 서버의 데이터베이스에서 프로필 사진을 업로드 해주는 uploadProfile() 로 전달한다.
연락처	<ul style="list-style-type: none"> - 사용자의 디바이스에서 연락처 정보를 가져와 listview 형식으로 사용자에게 보여줌 - 사용자는 원하는 연락처를 클릭하게 되면 연락처를 다른 사용자에게 전송할 것인지 아니면 직접 전화를 할 것인지 선택하는 dialog창이 뜨게 된다. <p>(1) plugin cordova-plugin-contactsPhoneNumbers cordova-plugin-dialog-list</p> <p>(2) 디바이스에서 연락처 정보를 가져와 listview 형식으로 띄어주기.</p> <pre> function onContacts(){ navigator.contactsPhoneNumbers.list(onSuccess,onError); function onSuccess(contacts) { var list=document.getElementById("contacts"); \$(list).empty(); for(var i = 0; i < contacts.length; i++) { \$(list).append('<li id="address'+i+'">'+ contacts[i].displayName+" "+ contacts[i].phoneNumbers[0].number+''); } }; } </pre> <ul style="list-style-type: none"> - listview에 이름과 연락처 정보만을 띄어준다. 

<p>연락처</p>	<p>(3) listview에서 원하는 이름을 클릭하면 dialog로 연락처를 다른 사용자에게 전송할 것인지 직접 전화를 할 것인지 선택하는 창이 뜨게 된다.</p> <ul style="list-style-type: none"> - 선택한 연락처에 대한 정보를 for문을 이용하여 찾음. <pre>for(var i = 0; i < contacts.length; i++) { var id="address"+i; if(addressId == id){ contact=contacts[i];</pre> - dialog 창을 띄어줌. <pre>cordova.plugins.dialogList(contactsSuccess,contactsError,contacts[i].displayName,["연락처 전송","전화"]);</pre> <pre>function contactsSuccess(index){ -연락처 전송을 선택하였다면 연락처를 append if(index == 0){ \$('#messages').append(" - "); } -전화를 선택하였다면 전화 어플리케이션을 실행 else{ window.open("tel:"+contact.phoneNumbers[0].number); } }</pre> 
<p>파일 전송</p>	<p>[위에서 보았던 여러 파일과 데이터들을 서버로 전송하는 기능]</p> <p>(1) plugin cordova-plugin-file-transfer</p> <p>(2) 파일들에 대한 함수가 비슷하므로 한 번에 설명하겠습니다.</p> <pre>function uploadFile(imageURL){ var options=new FileUploadOptions(); options.fileKey="file"; - 서버에서 file의 변수명을 결정 -> 서버에서 req.files.file 로 사용 options.fileName=imageURL.substr(imageURL.lastIndexOf('/')+1); - 다른 파일들과 구별할 파일의 이름 설정</pre>

파일 전송	<pre> options.mimeType="video/mp4"; - 파일의 mimeType을 설정 - 오디오 : "audio/m4a" - 비디오 : "video/mp4" - 이미지 : "image/jpeg" - 카카*톡에서 직접 파일들을 전송하여 확장자 확인 options.params={nickname:nickname}; - 서버에 nickname도 함께 전달 (프로필 사진을 받아오기 위해서) - 서버에서는 req.body.nickname으로 사용 function success(message){}; function fail(error){}; var ft=new FileTransfer(); ft.upload(imageURL,encodeURIComponent('http://202.31.200.143/uploadVideo'),success,fail,options); - (데이터 URL, 서버 URI, 옵션)을 순서대로 넣어줌 (3) 이미지, 동영상, 오디오 전송 시 서버 부분 app.post('/URL',function(req,res){ var filename=path.basename(req.files.file.path); - 파일의 이름만 받아온다. io.sockets.emit(...,{file:filename, nickname: req.body.nickname, image: ...}); - 다른 사용자들에게 파일 전송 res.status(204).end(); } - 파일을 서버에 업로드 하는 부분은 app.use(express.static(__dirname+'/public')); app.use(multipart({uploadDir:__dirname+'/public'})); 에 의해서 자동으로 public 폴더 안에 저장된다. (4) 프로필 사진 업로드 app.post('/uploadProfile',function(req,res){ var filename=path.basename(req.files.file.path); connect.query('UPDATE REGIST SET image=? WHERE nickname=?',[filename,req.body.nickname]); - query문을 사용하여 데이터베이스를 업데이트 해준다. res.status(204).end(); }); </pre>
-------	--

4. 추진 일정

(수행 기간 : 2016년 11월 3일 ~ 2016년 12월 8일)						
과제내용	구분	1주차	2주차	3주차	4주차	5주차
분석 및 설계		○				
UI 개발		○	○			
UI에 기본 기능 적용			○	○		
추가 기능 삽입				○	○	○
테스트 및 수정			○	○	○	○

5. 결과 분석 및 검토

초기 목표	최종 결과물
사진촬영	사진촬영
동영상촬영	동영상촬영
녹음	녹음
파일전송	파일전송
지도	지도
연락처	연락처
	날씨정보
	back button
	진동/가속도센서
	배터리상태
	볼륨키로 폰트사이즈 조절
	로딩화면
	어플리케이션 아이콘 변경

- 초기에 계획했던 모든 기능 구현 완료

- 기존 계획된 기능들을 구현하면서 추가적으로 필요한 기능들을 구현

- plugin을 이용해 디바이스의 기능들을 사용할 수 있는 점과 하나의 plugin을 어떻게 이용할 지에 대해서 생각하는 과정이 흥미로웠다. 또한 같은 plugin을 다른 use case로 구현한 조들을 보며 더 많이 배울 수 있었다.