

일정 관리 프로그램

(Schedule Management Program)

-Memory Planner-

이영미

2019.12

◦목차

○ 개요

1. 문서 개요

- 1) 문서 목적
- 2) 개발 범위

2. 시스템 개요

- 1) 시스템 목적
- 2) 시스템 범위
- 3) 시스템 조작

○ .상세 개발 내용

1. 소프트웨어 상세 요구사항

- 1) 요구사항 정의서
- 2) 설계 사양서
- 3) 테이블 구조

2. 주요 코드

3. 구현 화면

문서 개요서

프로젝트 명	일정관리 프로그램 개발(Memory Planner)
깃허브 주소	https://github.com/ymi316/kime

1. 문서 목적

- 본 문서는 '일정 관리 프로그램(Schedule Management Program)' 개발을 위한 요구사항을 명세하고 있다.

본 문서는 기본 사양서를 바탕으로 클라이언트의 요구사항을 명확하게 도출하여 향후 개발 과정에서 이를 반영하는데 그 목적이 있다. 따라서 본 문서는 클라이언트의 정확한 요구사항을 수집하고 이를 분석하여 명세한다.

2. 개발 범위

- 이 소프트웨어는 "일정 관리 프로그램(Schedule Management Program) - Memory Planner" 이다.
- 일정 관리 프로그램(Memory Planner)' 프로그램은 웹 및 모바일에서 사용이 가능한 프로그램으로 웹에서는 사용자가 마우스를 이용하여 조작을 하고, 모바일은 터치를 통해 사용 가능하다.
- '일정 관리 프로그램' 은 회원의 회원 가입 후 일정의 추가, 삭제, 수정이 가능하다
- 월 별, 주 별, 일 별로 일정을 관리 할 수 있다.

시스템 개요서

프로젝트명

일정관리 프로그램 개발(Memory Planner)

1. 시스템 목적

1) 일정관리

간편하고 손쉽게 일정을 관리 하는 프로그램

2) 남녀노소 누구나 쉽게 사용

요구되는 제약이 없어서 간단한 회원가입 후 간결한 인터페이스로 간편하게 일정을 관리 할 수 있다.

3) 월 별 관리 및 일 별, 주 별 일정관리

월별 일정을 관리할 수 있을 뿐만 아니라 일별 및 주별 일정을 관리할 수 있어

일정을 더 자세하게 관리 할 수 있다.

2. 시스템 범위

‘일정 관리 프로그램’은 웹(Web)에서의 제작을 우선으로 하고 장소에 따라

사용자가 원하는 기기로 언제 어디서나 프로그램을 이용할 수 있게 제작 할

것이다.

3. 시스템 조작

1) 마우스 및 키보드 조작으로 일정 관리 및 추가 수정 삭제

2) 월 별 및 일 별, 주 별 수정

상세 개발 내용

프로젝트명 일정관리 프로그램 개발(Memory Planner)

요구사항 정의서

No	기능적 요구 사항	관리자	회원
1	관리자 관리		
1.1	관리자 등록	관리자 등록	
1.2	관리자 수정	관리자 수정	
1.3	관리자 삭제	관리자 삭제	
2	회원 관리		
2.1	회원 권한 부여	회원 권한 부여	
2.2	회원 권한 제거	회원 권한 제거	
2.3	회원 권한 수정	회원 권한 수정	
2.4	전체 회원 보기	전체 회원 보기	
2.5	회원 삭제	회원 삭제	
3	개인 회원 관리		
3.1	개인 정보 동의		개인 정보 동의
3.2	회원 등록		회원 등록
3.3	회원 수정		회원 수정
3.4	회원 탈퇴		회원 탈퇴
3.5	아이디 검색		아이디 검색
3.6	비밀번호 검색		비밀번호 검색
3.7	회원 정보 보기		회원 정보 보기
3.8	회원 로그인		회원 로그인
4	일정 관리		
4.1	회원 일정 확인	회원 일정 확인	
4.2	회원 일정 삭제	회원 일정 삭제	
5	개인 일정 관리		
5.1	일정 확인		일정 확인
5.2	일정 등록		일정 등록
5.3	일정 삭제		일정 삭제
5.4	일정 수정		일정 수정
5.5	일 단위 스케줄		일 단위 스케줄
5.6	주 단위 스케줄		주 단위 스케줄
5.7	월 단위 스케줄		월 단위 스케줄

6	사이트 관리		
6.1	사이트 소개 등록	사이트 소개 등록	
6.2	사이트 관리자 소개	사이트 관리자 소개	
6.3	프로그램 소개	프로그램 소개	
7	FAQ 관리		
7.1	FAQ 등록	FAQ 등록	
7.2	FAQ 수정	FAQ 수정	
7.3	FAQ 삭제	FAQ 삭제	

상세 개발 내용

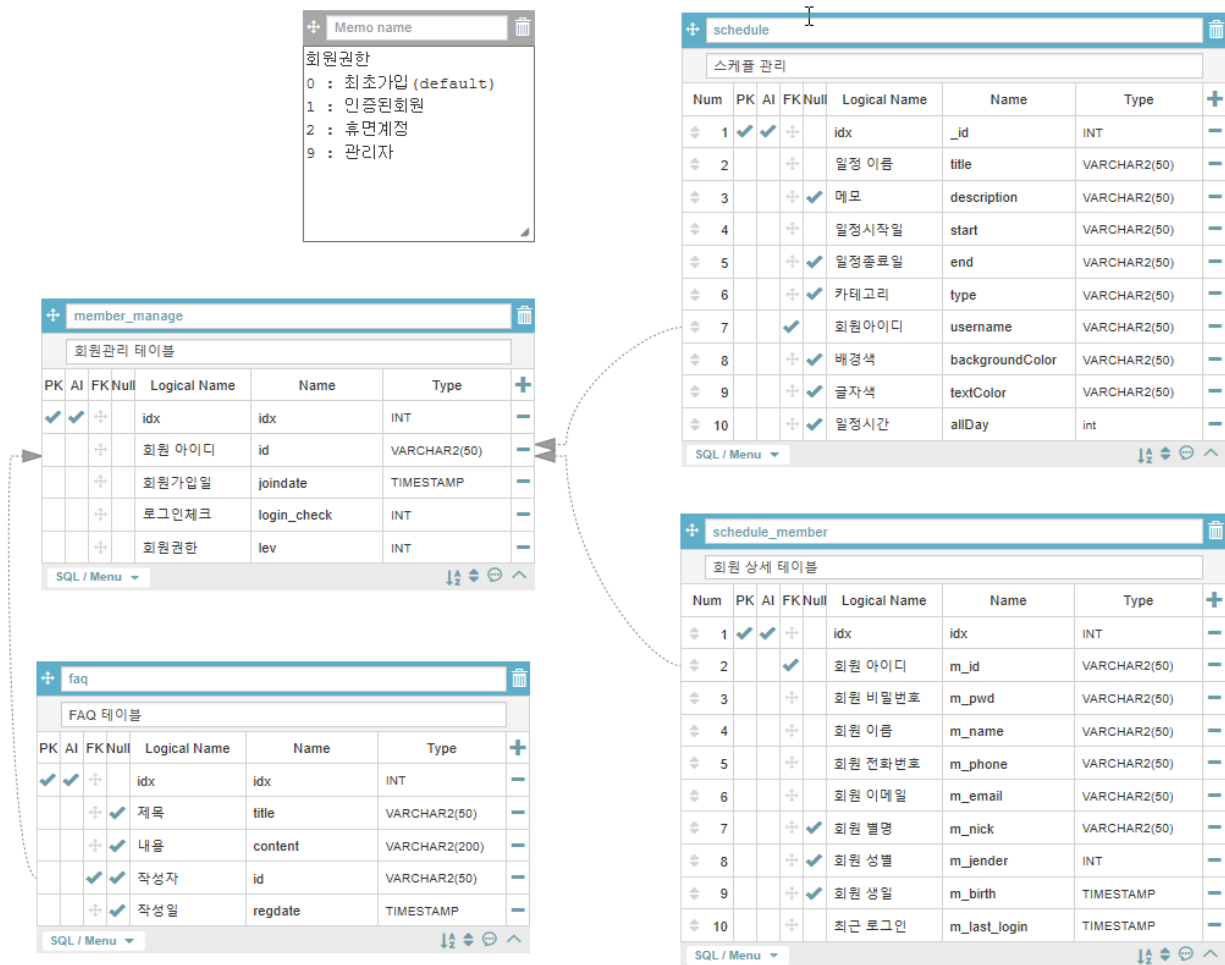
프로젝트명

일정관리 프로그램 개발(Memory Planner)

ERD 링크(password : x63qsa)

<http://aquerytool.com:80/aquerymain/index/?rurl=4b75dbf1-ce54-4dcc-a8ef-ce41b16dddfd>

설계 사양서



테이블 구조

테이블 이름		member_manage						
테이블 설명		회원관리 테이블						
PRIMARY KEY		idx						
FOREIGN KEY								
N O	P K	A I	F K	NU LL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y			idx	INT	idx	
2					id	VARCHAR2 (50)	회원 아이디	
3					joindate	TIMESTAMP	회원가입일	
4					login_check	INT	로그인체크	
5					lev	INT	회원권한	

테이블 이름		schedule_member						
테이블 설명		회원 상세 테이블						
PRIMARY KEY		idx						
FOREIGN KEY		m_id						
N O	P K	A I	F K	NU LL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y			idx	INT	idx	
2			Y		m_id	VARCHAR2 (50)	회원 아이디	member_manage
3					m_pwd	VARCHAR2 (50)	회원 비밀번호	
4					m_name	VARCHAR2 (50)	회원 이름	
5					m_phone	VARCHAR2 (50)	회원 전화번호	
6					m_email	VARCHAR2 (50)	회원 이메일	
7				Y	m_nick	VARCHAR2 (50)	회원 별명	
8				Y	m_jender	INT	회원 성별	
9				Y	m_birth	TIMESTAMP	회원 생일	
10					m_last_login	TIMESTAMP	최근 로그인	

테이블 이름		schedule						
테이블 설명		스케줄 관리						
PRIMARY KEY		_id						
FOREIGN KEY		username						
N O	P K	A I	F K	NU LL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y			_id	INT	idx	
2					title	VARCHAR2 (50)	일정 이름	
3				Y	description	VARCHAR2 (50)	메모	
4					start	VARCHAR2 (50)	일정시작일	
5				Y	end	VARCHAR2 (50)	일정종료일	
6				Y	type	VARCHAR2 (50)	카테고리	

7			Y		username	VARCHAR2 (50)	회원아이디	member_manage
8				Y	backgroundColor	VARCHAR2 (50)	배경색	
9				Y	textColor	VARCHAR2 (50)	글자색	
10				Y	allDay	int	일정시간	

테이블 이름					faq			
테이블 설명					FAQ 테이블			
PRIMARY KEY					idx			
FOREIGN KEY					id			
N O	P K	A I	F K	NU LL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y			idx	INT	idx	
2				Y	title	VARCHAR2 (50)	제목	
3				Y	content	VARCHAR2 (200)	내용	
4			Y	Y	id	VARCHAR2 (50)	작성자	member_manage
5				Y	regdate	TIMESTAMP	작성일	

상세 개발 내용

프로젝트명 일정관리 프로그램 개발(Memory Planner)

개발 서비스 환경

	구분	개발 환경		서비스 환경
SW	OS	Windows 10		Windows 10
	웹브라우저	Crome / Explorer / Edge		
	서버	Pivortal-tc-Server v4.0.1		
	개발언어	클라이언트	JavaScript	
			HTML5	
			CSS	
			jsp	
			jQuery	
	Framework	bootstrap v4.3.1		클라이언트
		SpringFramework 4.3.0		
		Mybatis 3.3.0		
		jQuery		
		FullCalendar		
		SummerNote		
	DBMS	Mybatis 3.3.0		Mybatis 3.3.0
Tools	Spring Tool Suite 3.9.5			
	Dbeaver 6.1.2			
	GitHub(github.com/Kim-dohee/synergy.git)			
	Maven			

상세 개발 내용

프로젝트 명

일정관리 프로그램 개발(Memory Planner)

주요 코드

로그인 컨트롤러

```
@RequestMapping(value = "/m")
public String login(HttpServletRequest request, Model model) {
    Cookie[] cookies = request.getCookies();

    if (cookies != null && cookies.length > 0) { // 쿠키를 읽어 아이디 저장확인
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals("m_id")) {
                model.addAttribute("m_id", cookie.getValue());
                break;
            }
        }
    }
    String referer = request.getHeader("referer");//이전페이지 확인
    try {
        referer = referer.replace("http://localhost:8080/project/", "");
    } catch (Exception e) {};
    request.getSession().setAttribute("referer", referer);
    logger.info("login referer : "+referer);
    return "member/login";
}

@RequestMapping(value = "/m/loginOk", method = RequestMethod.POST)
public String loginOkPost(@ModelAttribute MemberVO memberVO, @RequestParam(required = false) String remember,
    HttpServletRequest request, HttpServletResponse response) {
    // 서비스를 호출해서 로그인 확인
    MemberVO vo = memberService.loginOk(memberVO);
    if (vo == null) {
        return "redirect:/m";
    } else {
        request.getSession().setAttribute("vo", vo); // 세션에 저장
        if (remember != null && remember.equals("save")) {
            Cookie cookie = new Cookie("m_id", vo.getM_id()); //아이디저장이면 쿠키를 저장
            cookie.setMaxAge(60 * 60 * 24 * 7);
            response.addCookie(cookie);
        } else {
            // 쿠키 삭제
            Cookie cookie = new Cookie("m_id", "");
            cookie.setMaxAge(0);
            response.addCookie(cookie);
        }
    }
    //return "redirect:/";
    if(request.getSession().getAttribute("referer")==null) return "redirect:/";
    return "redirect:/"+request.getSession().getAttribute("referer");
}
```

회원가입 폼 체크

```
function validateEmail(email) {
    var re = /^[^\w-]+(?:\.[^\w-]+)*@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$/i;
    return re.test(email);
}
function validatePhone(m_phone){
    var re = /^^\d{2,3}-\d{3,4}-\d{4}$/;
    return re.test(m_phone);
}
function emailCheck(){
    var value= $('#m_email').val();
    if(value.length>5 && !validateEmail(value)){
        $('#emailCheck').css('color','red').html("이메일 형식으로 입력해주세요.");
    }
    else if(value.length>5){
        $.ajax({
            type:"POST",
            url:"${pageContext.request.contextPath }/m/checkSignupEmail",
            data:{
                "m_email":value
            },
            dataType:'json',
            success:function(data){
                if(data==1){
                    $('#emailCheck').css('color','red').html("사용 불가능한 이메일입니다.");
                }else{
                    $('#emailCheck').css('color','green').html("사용가능한 이메일입니다.");
                }
            }
        });
    }
    else{
        $('#emailCheck').html('');
    }
}
```

회원가입 컨트롤러

// 이메일 중복 체크

```
@RequestMapping(value = "/m/checkSignupEmail", method = RequestMethod.POST)
public @ResponseBody int AjaxView(@RequestParam("m_email") String m_email) {
    HashMap<String, String> map = new HashMap<String, String>();
    map.put("m_email", m_email);
    int check = (memberService.selectByEmail(map) == null ? 0 : 1);
    return check;
}
```

// 이메일 중복 체크

```
@RequestMapping(value = "/m/checkSignupID", method = RequestMethod.POST)
public @ResponseBody int AjaxID(@RequestParam("m_id") String m_id) {
    HashMap<String, String> map = new HashMap<String, String>();
    map.put("m_id", m_id);
    int idcheck = (memberService.selectByUserid(map) == null ? 0 : 1);
    return idcheck;
}
```

게시판 글쓰기 스크립트

```
$(function() {
    $('#summernote').summernote(
        {
            lang : 'ko-KR', // default: 'en-US'
            height : 300, // set editor height
            minHeight : null, // set minimum height of editor
            maxHeight : null, // set maximum height of editor
            fontNames : [ '맑은고딕', 'Arial', 'Arial Black',
                          'Comic Sans MS', 'Courier New', ],
            fontNamesIgnoreCheck : [ '맑은고딕' ],
            focus : true,
            callbacks : {
                onImageUpload : function(files, editor, welEditable) {
                    for (var i = files.length - 1; i >= 0; i--) {
                        sendFile(files[i], this);
                    }
                }
            }
        }
    );
});

function sendFile(file, el) {
    var form_data = new FormData();
    form_data.append('file', file);
    $.ajax({
        data: form_data,
        type: "POST",
        url: 'profileImage',
        cache: false,
        contentType: false,
        enctype: 'multipart/form-data',
        processData: false,
        success: function(img_name) {
            $(el).summernote('editor.insertImage', img_name);
        }
    });
}
```

게시판 컨트롤러

```
@RequestMapping(value = "/faq/faqView")
public String faqView(Model model, @RequestParam(required=false) int idx) {
    FaqVO vo = faqService.selectByIdx(idx);
    model.addAttribute("vo", vo);
    if(vo==null||idx<=0) return "/";
    return "manage/faqView";
}
```

```

// Post 방식 파일 업로드
@RequestMapping(value = "/faq/profileImage", produces = "text/plain;charset=utf-8")
@ResponseBody
public String multiUploadFormPOST(MultipartHttpServletRequest request) throws Exception {
    Logger.info("multiUploadFormPOST");
    // 서버의 실제 경로
    @SuppressWarnings("deprecation")
    String realPath = request.getRealPath(uploadPath);
    Logger.info("서버경로 : " + realPath);
    // 파일받기
    List<MultipartFile> files = request.getFiles("file");
    Logger.info("files : " + files);
    if (files == null || files.size() == 0) {
        return "redirect:/faq/faqEdit";
    }
    String savedName = "";
    for (MultipartFile file : files) {
        if (file != null && file.getSize() > 0) {
            savedName = uploadFile(request, realPath, file.getOriginalFilename(), file.getBytes());
        }
    }
    return savedName;
}
// 업로드된 파일을 저장하는 함수
private String uploadFile(MultipartHttpServletRequest request, String realPath, String originalName, byte[] fileData) throws IOException {
    UUID uid = UUID.randomUUID();
    String savedName = uid.toString() + "_" + originalName;
    File target = new File(realPath, savedName);
    // org.springframework.util 패키지의 FileCopyUtils는 파일 데이터를 파일로 처리하거나, 복사하는 등의 기능이
    // 있다.
    FileCopyUtils.copy(fileData, target);
    String contextPath = request.getContextPath();
    return contextPath + "/upload/" + savedName; // 이미지가 저장된 URL을 리턴하면 된다.
}

```

캘린더 컨트롤러

```

// responseBody로 디비에 있는 데이터 json파일로 파싱할 수 있도록 함
@RequestMapping(value = "/saveJson2")
@ResponseBody
public List<CalendarVO> responseBodyTest(HttpServletRequest request) {
    // List<CalendarVO> result = calendarService.selectByUserid(vo.getUsername());
    MemberVO memberVO = (MemberVO)request.getSession().getAttribute("vo");
    Logger.info("memberVO saveJson2 log : " + memberVO);
    List<CalendarVO> result = calendarService.selectByUserid(memberVO.getM_id());
    Logger.info("memberVO saveJson2 result : " + result);
    return result;
}

@RequestMapping(value = "/cal/insert", method = RequestMethod.POST)
public @ResponseBody boolean insert(@ModelAttribute CalendarVO vo, HttpServletRequest request) {
    Logger.info("Calendar insert : "+vo);
    MemberVO memberVO = (MemberVO) request.getSession().getAttribute("vo");
    vo.setUsername(memberVO.getM_id());
    Logger.info("CalendarVO : " + vo);
    calendarService.insert(vo);
    return true;
}

@RequestMapping(value = "/cal/insert", method = RequestMethod.POST)
public @ResponseBody boolean insert(@ModelAttribute CalendarVO vo, HttpServletRequest request) {
    Logger.info("Calendar insert : "+vo);
    MemberVO memberVO = (MemberVO) request.getSession().getAttribute("vo");
    vo.setUsername(memberVO.getM_id());
    Logger.info("CalendarVO : " + vo);
    calendarService.insert(vo);
    return true;
}

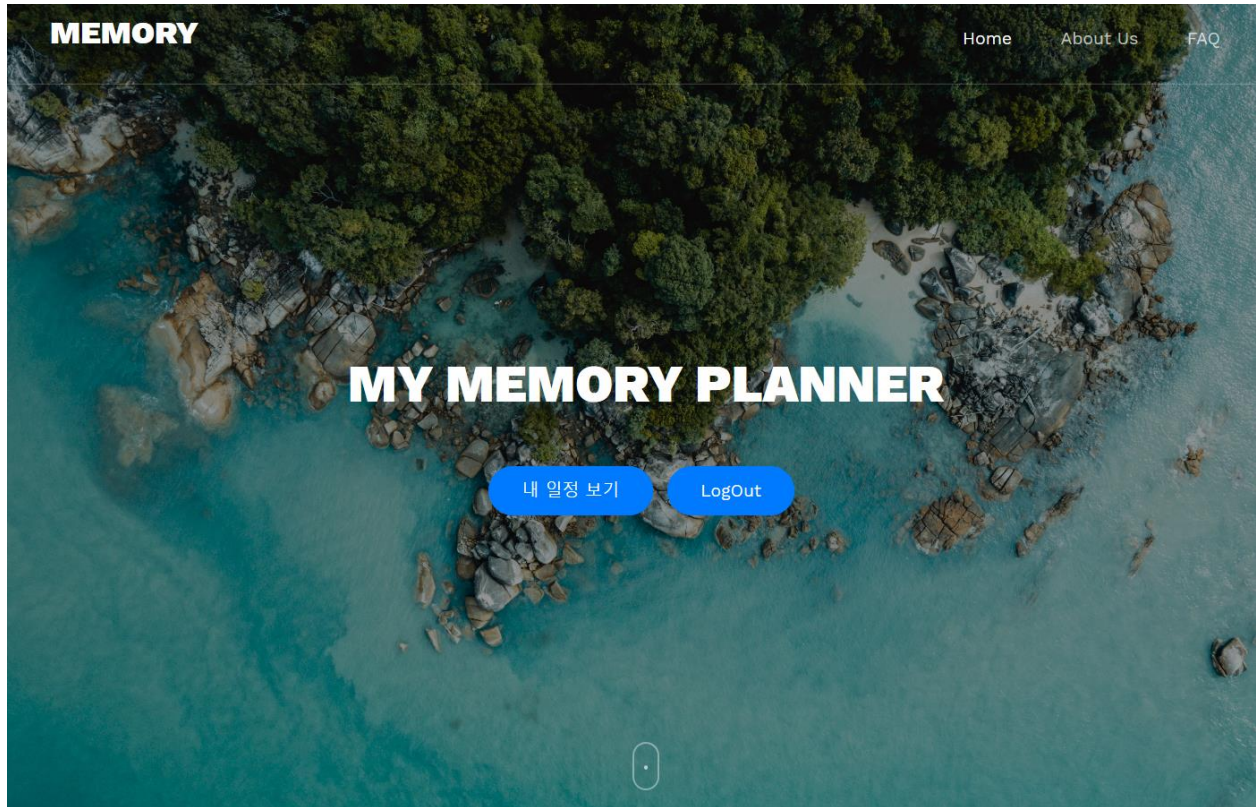
```

상세 개발 내용

프로젝트명

일정관리 프로그램 개발(Memory Planner)

구현화면



About us



Schedule Management Program

Schedule Management Program by LeeYM

- ✓ 회원 가입 및 로그인
- ✓ 일정 입력하기
- ✓ 일정 수정하기

Get In Touch

LOGIN

Email address

Password

☐ 아이디 저장

[회원가입](#)

[아이디 찾기](#)

[비밀번호 찾기](#)

LOGIN



회원가입

admin1234

사용가능한 아이디입니다.

●●●●●●

●●●●●●

admin1234@co.kr

사용가능한 이메일입니다.

사용자

별명

2019/12/05



010-0000-1234

돌아가기

가입하기

<div> <div>오늘</div> <div><<</div> <div>>></div> <div>< 2019년 12월 ></div> <div> <div>월</div> <div>주</div> <div>일</div> <div>일정목록</div> </div> </div>						
일	월	화	수	목	금	토
1	2	3 00:00 - 18:30 테스트 일정	4	5	6	7
8	9	10 짧은 약속	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

새로운 일정

하루종일

☐

일정명

시작

2019-12-11 15:10

끝

2019-12-11 16:10

구분

약속

색상

연두색

설명

취소

저장

오늘 << >>

< 2019년 12월 8 - 14일 >

월 주 일 일정목록

	일 12.08.	월 12.09.	화 12.10.	수 12.11.	목 12.12.	금 12.13.	토 12.14.
총일				짧은 약속			
00:00							
01:00							
02:00							
03:00							
04:00							
05:00							
06:00							
07:00							
08:00							
09:00							
10:00							
11:00							

오늘 << >>

< 2019년 12월 10일 >

월 주 일 일정목록

	화요일
총일	
06:00	
07:00	
08:00	
09:00	
10:00	
11:00	
12:00	
13:00	
14:00	
15:00	
16:00	
17:00	

오늘 << >>

< 2019년 12월 8 - 14일 >

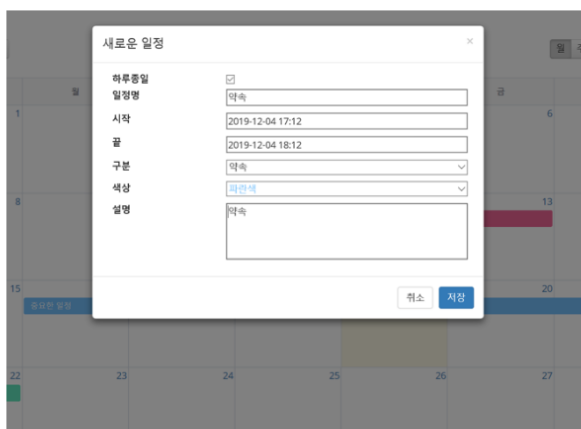
월 주 일 일정목록

수요일	2019년 12월 11일
종일 ● 짧은 막속	
목요일	2019년 12월 12일
종일 ● 짧은 막속	
금요일	2019년 12월 13일
종일 ● 짧은 막속	

FAQ - Memory Planner 사용법

글 작성하기

어떻게 이용하나요?



- 우선 일정을 입력할 날짜를 선택합니다.
- 팝업창에 세부일정을 작성합니다.
- 버튼을 누르면 일정이 입력됩니다.

글 수정

FAQ

어떻게 이용하나요?

내용

    Helvetica            

- 우선 일정을 입력할 날짜를 선택합니다.
- 팝업창에 세부일정을 작성합니다.
- 버튼을 누르면 일정이 입력됩니다.

수정

돌아가기