

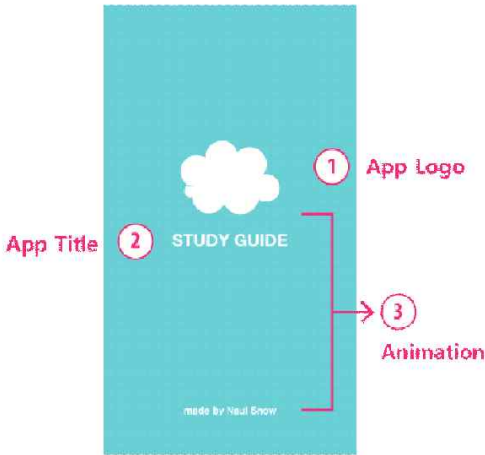
# Android 프로젝트 기술서

## 프로젝트 개요

프로젝트 명	STUDY GUIDE	
개발기간	2018-03-07 ~ 2018-04-20 1달 반	
개발인원	1인 개발	
담당역할	기획, 디자인, 로고 제작, 레이아웃 설계, 기능 개발	
개발환경	OS	Windows 7, Mac
	Development Tool	Android Studio, Genymotion, Adobe Illustrator, Adobe xd
	Language	Java, HTML, CSS, PHP
	DB	Dothome 웹호스팅 (MySQL)

## 프로젝트 소개

### 인트로 화면 (IntroActivity)

주요화면	설명
	<ol style="list-style-type: none"><li>1. 로고는 Adobe Illustrator의 펜툴을 사용하여 자체제작 되었습니다.</li><li>2. 앱 이름</li><li>3. Animation을 사용하여 res에 따로 anim폴더를 만들어 로고, 이름용 애니메이션 xml파일을 만들었습니다. Timer 과 TimerTask를 이용해 특정 시간이 지난뒤 LoginActivity를 intent로 호출하였습니다.</li></ol>


## 로그인 화면 (LoginActivity, FirstSettingActivity)

주요화면	설명
<p>1 SNS Login</p> <p>2 세부사항 설정</p> <p>3 회원가입 완료</p>	<p><b>1. SNS로그인 구현</b></p> <p>-ISessionCallback을 상속받은 SessionCallback 객체로 카카오톡 로그인 되었는지를 확인합니다. 로그인 정보가 저장되어있으면 자동 로그인이 되어 MainActivity로 이동합니다.</p> <p>● KakaoAPI 의 LoginButton에 자체제작된 이모티콘을 연결해 앱의 디자인 / 통일성부분을 강화했습니다.</p> <p>● 카카오톡 로그인인을 시도하면 requestMe 함수가 불려지고 성공하면 UserProfile에서 유저의 id정보와 thumgbnailImagePath를 가져옵니다.</p> <p>● 이때 해당 유저의 id를 dothome 웹호스팅의 Database로 Volley를 사용하여 전송해 user table에 해당유저가 가입한적이 있는지를 확인합니다. 가입정보가 있는 경우 MainActivity로 바로 이동합니다.</p> <p><b>2. 가입정보가 없는 경우 세부사항을 입력하여 Database에 유저의 정보를 저장합니다.</b> 이때 NickName은 입력된 내용을 Database의 user table과 비교해 중복검사를 합니다.</p> <p><b>3. 회원가입 완료되며 Database에 id, SNS종류, profileImage, NickName, 나이 등이 저장됩니다.</b></p>

## 메인 화면 (MainActivity, Fragments)

주요화면	설명
<p>1 Home</p> <p>2 Feed</p> <p>3 Study</p> <p>4 Report</p> <p>5 Setting</p> <p>2 Feed Fragment</p> <p>3 Study Fragment</p>	<p><b>0. activity_main.xml에 ViewPager과 BottomNavigationView를 함께 사용하였으며 내부에 Fragment와 FragmentPagerAdapter를 상속받은 FragAdapter를 이용해 하단 아이콘이 눌리면 pager.setCurrentItem()을 통해 페이지 변경을 하였고 옆으로 슬라이드 된 경우에는 BottomNavigationView의 checked된 item을 변경했습니다.</b></p> <p><b>1. 홈페이지(F1Home Fragment)로 이동</b></p> <p><b>2. 피드(F2Feed Fragment)로 이동</b></p> <p><b>3. 공부화면(F3Study Fragment)으로 이동</b></p> <p><b>4. 보고서화면(F4Report Fragment)으로 이동</b></p> <p><b>5. 설정화면(F5Settings Fragment)으로 이동</b></p>

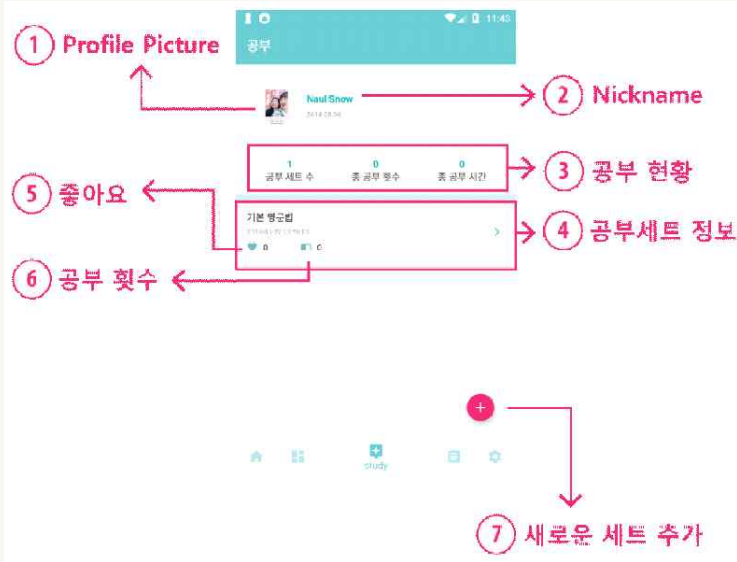
## 피드 Fragment 화면 (F1Home Fragment)

주요화면	설명
 <p>1 피드 2 좋아요/댓글수 3 새피드 작성 4 피드 상세 5 배너광고 6 댓글 작성 7 댓글 화면</p>	<ol style="list-style-type: none"> <li>1. RecyclerView 와 RecyclerView.Adapter를 상속받은 FeedFragAdapter를 이용한 모습입니다. DB에 저장된 feed 테이블을 불러와서 나타냅니다. 이때 UTC시간을 LocalTime으로 변환하여 나타냅니다.</li> <li>2. FeedLike테이블에서 feedID가 해당 feedID와 동일한 record의 합으로 총 좋아요수를 구하고, feedID가 해당 아이디이면 userID가 사용자 아이디인 record가 있으면 ToggleButton인 하트가 checked됩니다. 댓글도 comment테이블에서 feedID를 이용해 개수를 구합니다.</li> <li>3. FAB를 이용해 AddFeedActivity로 이동합니다.</li> <li>4. FeedDetailActivity이며 SwipeBackActivity, SwipeBackActivityHelper로 애니메이션 효과를 주었습니다.</li> <li>5. Admob의 AdView를 활용한 배너광고를 넣었습니다.</li> <li>6. 댓글작성시 comment테이블에 댓글정보가 insert 됩니다.</li> <li>7. FeedCommentActivity입니다. 댓글의 댓글을 볼 수 있습니다.</li> </ol>


## 피드 추가 화면 (AddFeedActivity)

주요화면	설명
 <p>1 글 작성 영역 2 카메라/갤러리 3 Permission</p>	<ol style="list-style-type: none"> <li>1. EditText에 글작성이 가능합니다. 위에는 ImageView가 있는데 사진 추가시에 나타납니다.</li> <li>2. 카메라/갤러리 중 선택해서 startActivityResult()를 사용해 사진을 Uri혹은 Bitmap을 Uri로 변환한 내용을 imgPath에 저장해 Volley의 multipartRequest.addfile()을 통해 전송합니다. PHP에서 파일 이름이 UTC시간으로 변경됩니다.</li> <li>3. SDK버전이 M이상인 경우 외부 저장소 읽기쓰기 Permission과 카메라 Permission 여부를 체크하여 허용이 안된 상태인 경우 requestPermissions를 통해 Dialog를 띄워 Permission을 받습니다</li> </ol>


## 공부 Fragment 화면 (F3Study Fragment)

주요화면	설명
 <p>1 Profile Picture</p> <p>2 Nickname</p> <p>3 공부 현황</p> <p>4 공부세트 정보</p> <p>5 좋아요</p> <p>6 공부 횟수</p> <p>7 새로운 세트 추가</p>	<ol style="list-style-type: none"> <li>1. Glide를 이용해 SNS로그인시 받아온 ProfileImage를 불러옵니다.</li> <li>2. 첫 로그인시 생성한 Nickname입니다.</li> <li>3. studySet에서 내가 가진 세트들의 개수와 studyQuestion 테이블에서 내 userID에 해당되는 레코드의 합, 공부시간의 합계를 계산하여 표시합니다.</li> <li>4. RecyclerView를 이용해 DB에 저장된 모든 나의 studySet과 foreignKey로 가지고 있는 sgSetID를 이용해 sgSet의 정보들을 불러와 나열합니다. 클릭시 AddSetActivity로 이동합니다. (SwipeBackActivity사용)</li> <li>5. sgSetLike에서 해당 sgSet의 record 개수와 내 userID도 그 안에 있는지 확인하여 checked를 설정합니다.</li> <li>6. studySet과 나의 userID를 둘다 가진 record를 studyQuestion 테이블에서 가져와 각 레코드의 timeLength의 합계를 냅니다.</li> <li>7. FAB를 이용해 AddSetActivity로 이동합니다.</li> </ol>

## 공부 세트 추가 화면 (AddSetActivity)

주요화면	설명
 <p>1 세트 기본 정보</p> <p>2 세부 설정</p> <p>3 문제</p> <p>4 답</p> <p>5 문제 설명</p> <p>6 FAB 맨위로 가기</p> <p>7 문제 추가</p>	<ol style="list-style-type: none"> <li>1. 세트의 기본적인 정보입니다.</li> <li>2. AlertDialog.Builder를 사용하고 LayoutInflater로 dialog_add_set.xml을 불러와 Dialog를 띄웁니다. (나만보기/전체보기)</li> <li>3. RecyclerView와 ArrayList로 제작되어 있습니다.</li> <li>4. 각 Question객체안의 ArrayList로 개수를 자유롭게 놓으며 Checkbox의 checked여부를 통해 답이냐를 정합니다.</li> <li>5. 문제에 대한 설명입니다. EditText</li> <li>6. FAB를 클릭하면 맨위로 올라갑니다.</li> <li>7. OnClickListener로 한번 눌리면 기본 문제 타입이 밑으로 하나 추가되고 LongClickListener를 사용하여 누르면 기본 문제 타입이 선택된 타입으로 변경되며 해당타입의 문제가 추가됩니다.</li> </ol>

## 공부 세트 메인 화면 (StudySetMainActivity)

주요화면	설명
 <p>1 세트정보 ← 기본 영문법 기본 문법법을 공부하고 복습하기 위한 세트입니다.</p> <p>2 좋아요 ← 0 61</p> <p>3 공부 현황 20 문제 수 91 공부 횟수 4 공부 시간</p> <p>4 문제관리/수정 ← 문제 공부</p> <p>5 공부/시험 문제 결과 보고서</p> <p>6 시험결과/그래프</p> <p>7 배너광고</p>	<ol style="list-style-type: none"> <li>1. Intent를 통해 전달된 studySetID를 통해 세부 정보들과 SgSet, Question, Answer, StudyQuestion의 정보들을 DB에서 받아옵니다.</li> <li>2. 앞과 같은 방법으로 좋아요와 공부횟수 정보를 불러옵니다.</li> <li>3. 해당 studySetID를 가지고있는 question레코드 개수와 그 안의 timeLength합을 구해 사용합니다.</li> <li>4. SwipeBackActivityHelper를 이용해 서재로 이동합니다. (문제관리/수정 작업)</li> <li>5. SwipeBackActivityHelper를 이용해 공부로 이동합니다. (공부/시험)</li> <li>6. SwipeBackActivityHelper를 이용해 서재로 이동합니다. (공부/시험)</li> <li>7. Admob의 AdView를 배치하고 AdRequest를 받아와 광고를 담니다.</li> </ol>

## 서재 화면 : 모든 문제 보기 / 문제 수정 (StudyStorageActivity, UpdateSetActivity)

주요화면	설명
 <p>1 세트정보 ← 기본 영문법 기본 문법법을 공부하고 복습하기 위한 세트입니다.</p> <p>2 세트정보 ← 문제 1 기본형 다음 중 De동사의 사용법으로 옳지 않은 것은? <input type="checkbox"/> I am a student. <input checked="" type="checkbox"/> You is a great teacher. <input type="checkbox"/> She is a famous doctor. <input type="checkbox"/> They are so cute! You은 한명도 여러명도 될 수 있지만 De동사는 are만 사용합니다.</p> <p>3 수정페이지 문제 2 기본형 다음중 형용사 국가명을 잘못 기술한 것은? <input type="checkbox"/> Poland <input type="checkbox"/> Korea <input type="checkbox"/> France <input checked="" type="checkbox"/> China</p>	<ol style="list-style-type: none"> <li>1. StudySetMainActivity에서 가져온 기본 정보입니다.</li> <li>2. RecyclerView와 ArrayList를 사용해 가진 모든 문제들을 볼 수 있습니다. 현재는 사진 추가까지 가능합니다.</li> <li>3. 문제 추가/삭제, 답 추가/삭제 또는 문제/답 수정이 일어난 부분만 추가/삭제시의 OnClickListener, 수정시의 OnCheckedChangeListener와 TextWatcher를 이용해 정보를 모아 서버에 전송하여 DB에 insert, update, delete를 합니다.</li> </ol>

## 공부 화면 : 공부 하기 / 시험 보기 (StudyingPageActivity, TestingPageActivity)

주요화면	설명
<p>1 공부 인트로</p> <p>2 공부중 화면</p> <p>3 시험 인트로</p> <p>4 시험중 화면</p> <p>5 시험 현황</p>	<p>1. 공부 버튼을 누르면 StudyStudyActivity로 오며 공부/테스트를 선택하여 이동합니다.</p> <p>2. question들이 랜덤하게 등장하며 답또한 랜덤하게 분포되고 각각의 결과들과 걸린시간에 대한 정보가 서버에 전송되어 studyQuestion에 저장됩니다.</p> <p>3. 오른쪽 설정 이모티콘을 클릭해 시험 시간과 문제 개수 설정이 가능하며 이때 설정된 값은 sharedPreferences에 저장되어 다음번에도 같은값으로 나옵니다.</p> <p>4. Timer와 TimerTask를 사용해 타이머 기능을 구현했고 시간 종료시 자동 제출/제점됩니다. 제출전까지 답선택 정보가 answer에 저장되며 수정이 가능합니다.</p> <p>5. 풀이 현황 바를 클릭하면 안푼 문제들로 바로 이동이 가능하며 문제 현황 화면이 자동으로 사라집니다.</p>

## 보고서 화면 : 시험 결과 / 결과 그래프 (StudyReportActivity)


주요화면	설명
<p>1 그래프</p> <p>2 시험결과 리스트</p> <p>3 배너광고</p>	<p>1. test 테이블에서 이 studySetID와 사용자의 userID에 해당되는 모든 시험결과를 불러옵니다. HelloChart Library의 LineChartView를 사용하여 결과하나하나를 LineChartData에 담아 나타냅니다.</p> <p>2. 시험결과를 Recyclerview를 사용해 리스트로 나타내고, 오른쪽에는 백분위로 나타낸 점수와 그 점수에 해당되는 등급을 보여줍니다.</p> <p>3. Admob을 활용한 AdView를 달아 광고를 담니다.</p>



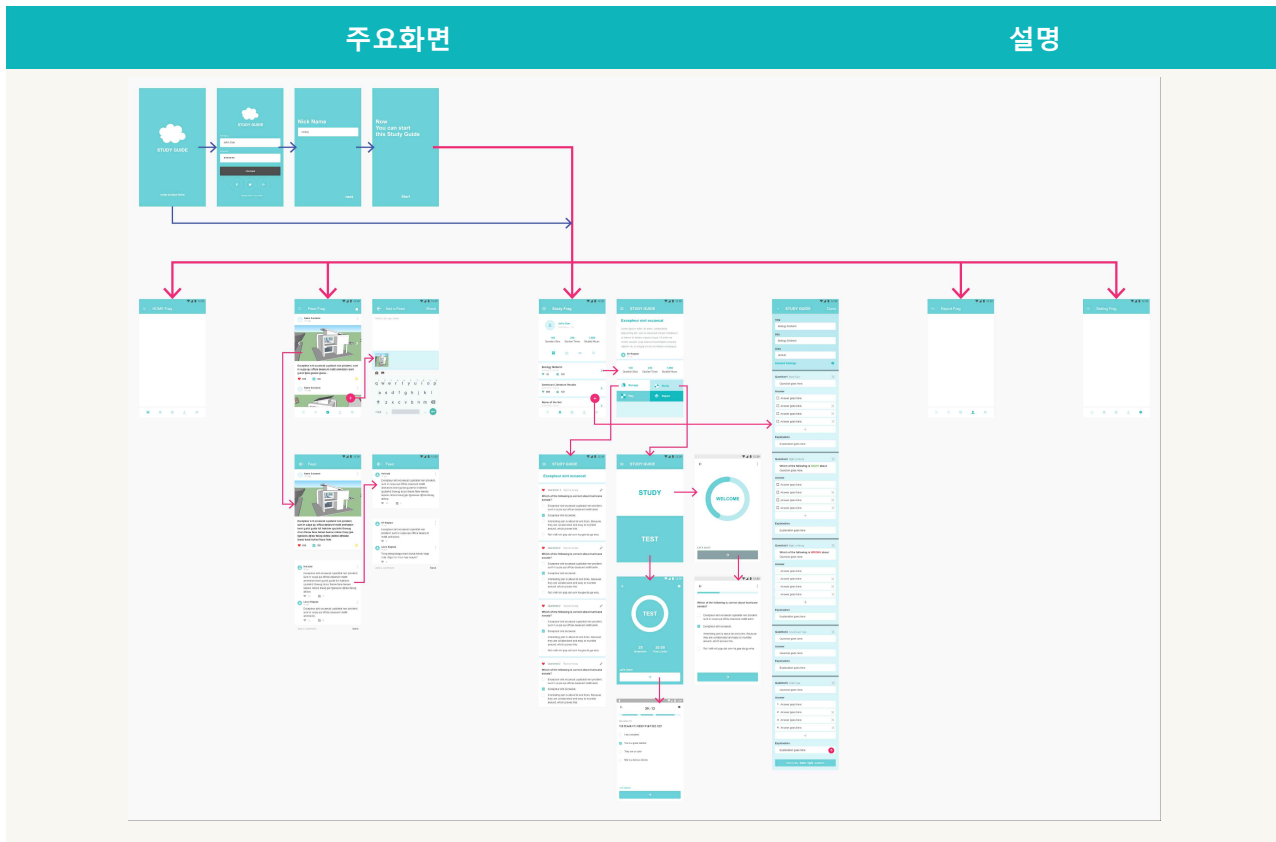
## 다국어 리소스 추가 : 영어(기본언어), 한국어

주요화면	설명
<pre> &lt;string name="app_name"&gt;STUDY GUIDE&lt;/string&gt; &lt;string name="intro_madeby" translatable="false"&gt;made by Nau  &lt;string name="coming_soon"&gt;Coming soon..&lt;/string&gt;  &lt;string name="login_id"&gt;ID&lt;/string&gt; &lt;string name="login_psw"&gt;Password&lt;/string&gt; &lt;string name="login_continue"&gt;CONTINUE&lt;/string&gt; &lt;string name="login_lostpsw"&gt;Forgot your password?&lt;/string&gt;  &lt;string name="login_fail01"&gt;An ERROR has occurred.&lt;/string&gt; &lt;string name="login_fail02"&gt;This user doesn't exist.&lt;/string&gt; &lt;string name="login_fail03"&gt;Failed to connect to the server.  &lt;string name="dialog_positive"&gt;Okay&lt;/string&gt; &lt;string name="dialog_negative"&gt;Cancel&lt;/string&gt;  &lt;string name="test_submit"&gt;Are you &lt;string name="app_name"&gt;STUDY GUIDE&lt;/string&gt; &lt;string name="coming_soon"&gt;업데이트 준비중..&lt;/string&gt; &lt;string name="login_id"&gt;아이디&lt;/string&gt; &lt;string name="login_psw"&gt;비밀번호&lt;/string&gt; &lt;string name="login_continue"&gt;계속&lt;/string&gt; &lt;string name="login_lostpsw"&gt;비밀번호를 잊어버렸나요?&lt;/string&gt; &lt;string name="login_fail01"&gt;알 수 없는 에러가 발생했습니다..&lt;/string&gt; &lt;string name="login_fail02"&gt;해당 유저는 존재하지 않습니다.&lt;/string&gt; &lt;string name="login_fail03"&gt;서버 연결에 실패했습니다.&lt;/string&gt; &lt;string name="dialog_positive"&gt;확인&lt;/string&gt; &lt;string name="dialog_negative"&gt;취소&lt;/string&gt; &lt;string name="test_submit"&gt;정말 제출하시겠습니까?\n제출 이후 &lt;string name="first_info01"&gt;느낌&lt;/string&gt; &lt;string name="first_info02"&gt;미&lt;/string&gt; &lt;string name="first_info03"&gt;관심사&lt;/string&gt; &lt;string name="first_info04"&gt;소셜&lt;/string&gt; &lt;string name="f1_home"&gt;홈&lt;/string&gt; &lt;string name="f2_feed"&gt;피드&lt;/string&gt; &lt;string name="f3_study"&gt;공부&lt;/string&gt; &lt;string name="f4_report"&gt;보고서&lt;/string&gt; &lt;string name="f5_setting"&gt;설정&lt;/string&gt;  &lt;string name="menu_edit"&gt;수정&lt;/string&gt; &lt;string name="menu_delete"&gt;삭제&lt;/string&gt; </pre>	<ol style="list-style-type: none"> <li>로케일 디렉토리와 문자열 파일을 생성해 다국어를 지원합니다.</li> <li>values의 strings.xml파일을 이용해 자동 언어 변환을 가능하게 하였습니다.</li> </ol>

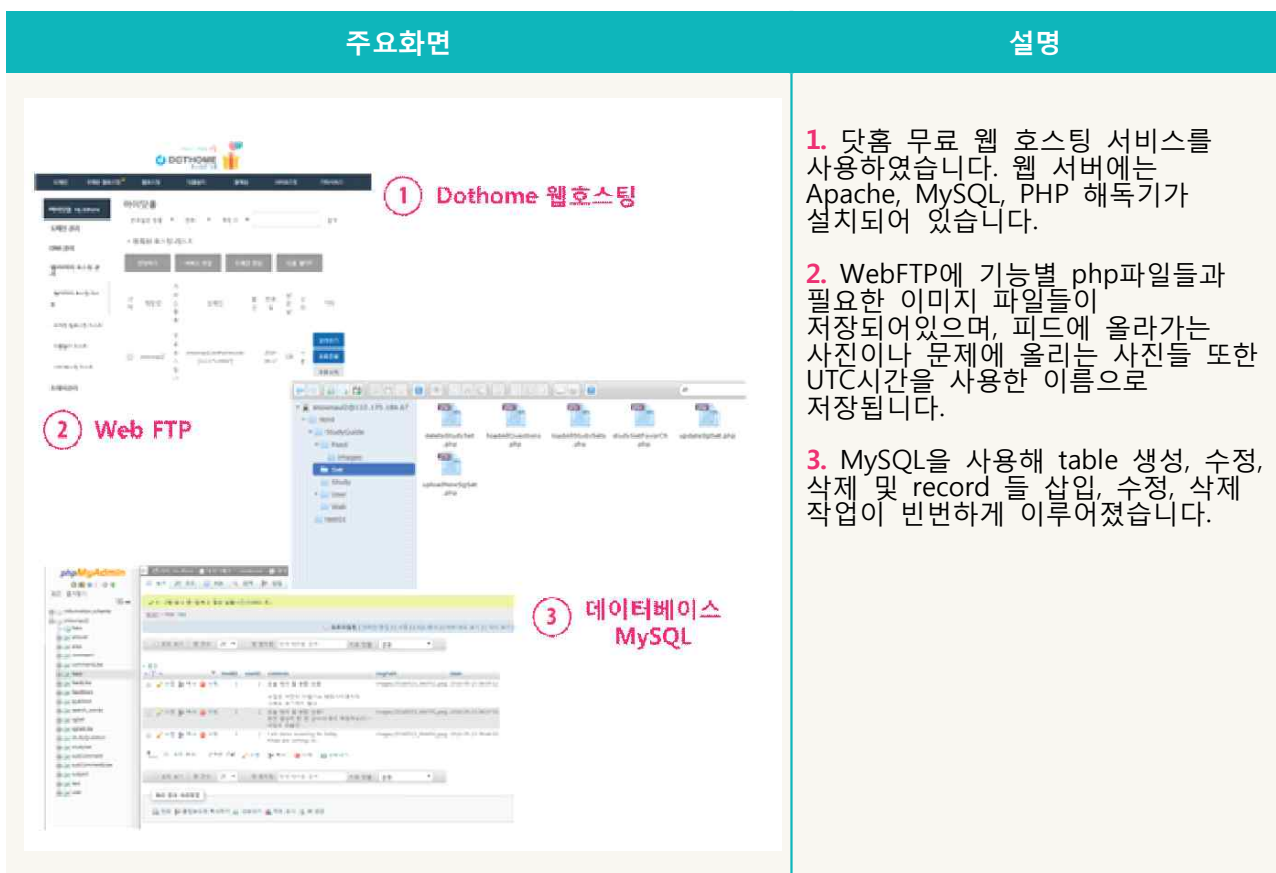
## [기획] 앱 기획서

주요화면	설명
	<ol style="list-style-type: none"> <li>작성된 앱 기획서의 모습입니다.</li> <li>애플리케이션 기획 배경(의도), 애플리케이션 주요기능, 주요 사용대상(타겟), 애플리케이션 구현방법 및 보충설명에 해당되는 내용이 기입되어있습니다.</li> </ol>

## [기획] 와이어프레임 (Adobe XD, Adobe Illustrator)

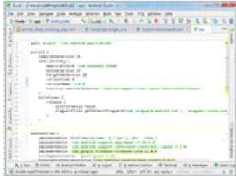





## Dothome 웹호스팅 서버 사용

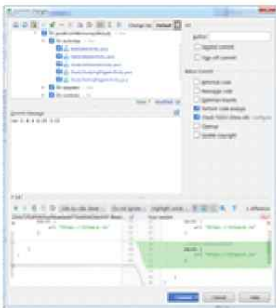
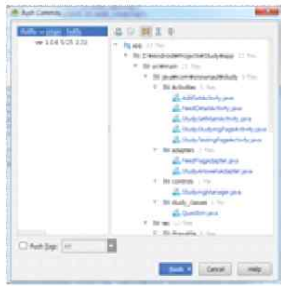
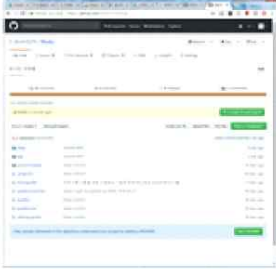
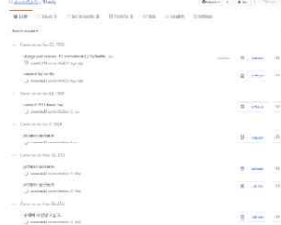




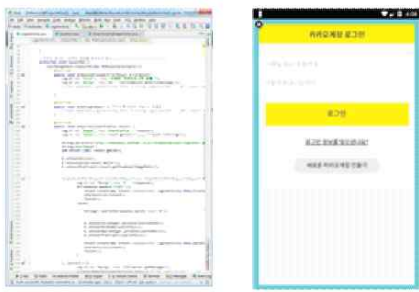

## 구글 플레이스토어 업로드

주요화면	설명
<div> <div>① 앱 버전 수정</div>  </div> <div> <div>② apk파일</div>  </div> <div> <div>③ 새버전 출시</div>  </div> <div> <div>④ 버전출시 완료</div>  </div> <div> <div>⑤ 구글 플레이스토어의 모습</div>  </div>	<ol style="list-style-type: none"> <li>1. build.gradle의 versionCode와 versionName을 수정합니다.</li> <li>2. generate Signed APK를 통해 apk파일을 생성합니다.</li> <li>3. Google Play Console에서 새 버전을 출시합니다.</li> <li>4. 새버전 출시가 완료된 모습입니다.</li> <li>5. 구글 플레이 스토어에 나타나는 모습입니다. 영어(기본언어), 한국어 두 버전으로 나와있습니다.</li> </ol>





## Github 사용

주요화면	설명
<div> <div>① commit</div>  </div> <div> <div>② push</div>  </div> <div> <div>③ github</div>  </div> <div> <div>④ commit history</div>  </div>	<ol style="list-style-type: none"> <li>1. 변경사항을 add / commit 하는 모습입니다.</li> <li>2. commit된 내용을 github에 push 하는 모습입니다.</li> <li>3. github에 현재까지 push된 내용이 올라가있는 모습입니다.</li> <li>4. 지금까지 commit한 commit history의 모습입니다.</li> </ol>

## 카카오 API (로그인), Admob API (광고)

주요화면	설명
<p>① 카카오 로그인 기능</p>  <p>② Admob 광고 기능</p> 	<p>1. 카카오 API를 사용해 SNS로그인 기능을 구현하였습니다. SessionCallback객체로 자동 로그인을 하고 LoginButton을 이용해 카카오로그인 화면으로 이동하고 Requestme()함수를 통해 결과를 받아옵니다. 받은 결과 UserProfile에서 id, thumbnailImagePath를 가져옵니다.</p> <p>2. AdmobAPI의 광고 기능을 사용했습니다. layout에는 AdView를 놓고 java단에서 AdRequest를 통해 광고를 load해 사용했습니다.</p>

## 외부 라이브러리 사용

주요화면	설명
<p>① Glide</p>  <p>② Volley</p>  <p>③ Jelly Refresh Layout</p>  <p>④ 아래로 스와이프</p>  <p>⑤ 로딩중</p>  <p>⑥ Android Swipe Layout</p>  <p>⑦ 옆으로 스와이프</p>  <p>⑧ Circle Image View</p> 	<p>1. Glide - 이미지 작업을 용이하게 합니다.</p> <p>2. Volley - 서버 전송을 용이하게 합니다.</p> <p>3. Jelly Refresh Layout - 아래로 스와이프 하는 모션으로 귀여운 애니메이션과 함께 데이터를 다시 받아옵니다.</p> <p>4. 아래로 스와이프 되었을 때 보이는 모습입니다.</p> <p>5. 로딩중 메시지와 함께 데이터를 받아옵니다.</p> <p>6. Android Swipe Layout - 옆으로 스와이프 하는 모션으로 전 페이지로 이동하게 합니다.</p> <p>7. 스와이프하여 전 화면으로 가는 모습입니다.</p> <p>8. Circle Image View - 이미지를 동그랗게 사용하게 합니다.</p>

# Android Messenger 프로젝트 기술서

## 프로젝트 개요

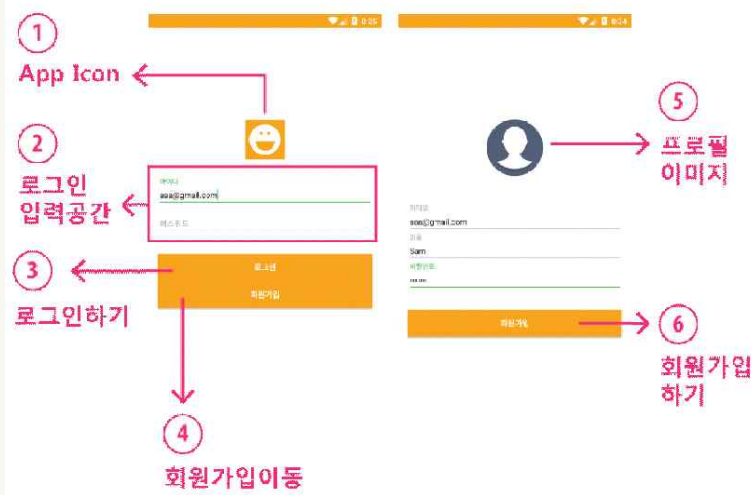
프로젝트 명	Chatting Mon		
개발기간	2018-05-23 ~ 2018-06-22 1달		
개발인원	1인 개발		
담당역할	개발, 레이아웃 설계, 디자인, Firebase작업		
개발환경	OS	Windows 7	
	Development Tool	Android Studio, Genymotion	
	Language	Java	
	DB	Firebase	

## 프로젝트 소개

### Firebase Remote Config (원격 제어)

주요화면	설명
 <p>1 아이콘 클릭</p> <p>2 서버점검중인 경우</p> <p>3 서버점검중이 아닌 경우</p>	<ol style="list-style-type: none"><li>아이콘을 클릭하면 ChattingMon앱의 SplashActivity가 실행됩니다. FirebaseRemoteConfig에 서버에 설정된 값들을 FirebaseRemoteConfigSettings를 사용하여 fetch하여 저장합니다.</li><li>저장된 값들중 서버 점검중여부가 true인 경우 AlertDialog와 함께 앱이 fetch해서 받아온 색상으로 배경화면을 전환한뒤 종료됩니다.</li><li>서버점검중이 아닌 경우 Intent를 통해 LoginActivity로 이동뒤 SplashActivity는 finish됩니다.</li></ol>

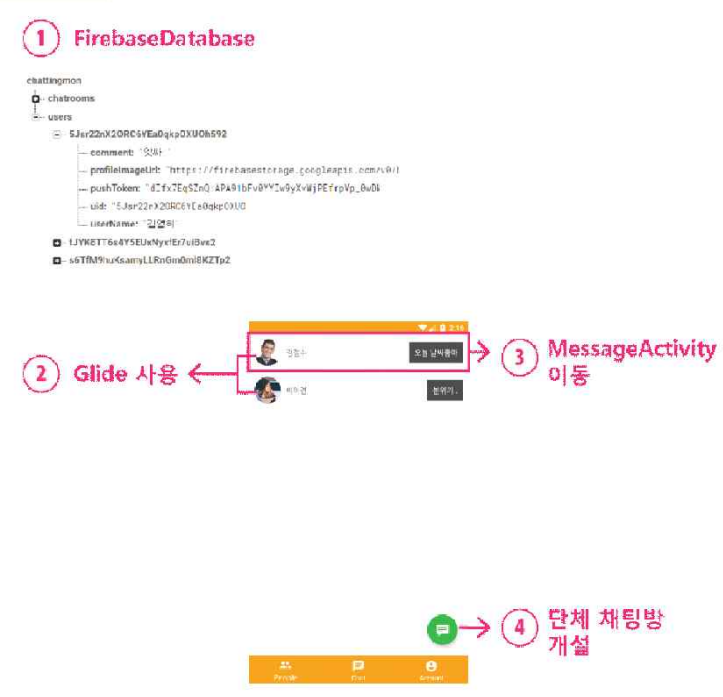
## 로그인 / 회원가입 화면 (LoginActivity / SignUpActivity)

주요화면	설명
 <p>1 App Icon</p> <p>2 로그인 입력공간</p> <p>3 로그인하기</p> <p>4 회원가입이동</p> <p>5 프로필 이미지</p> <p>6 회원가입하기</p>	<p>0. 앱실행시 Activity가 start될 때 AuthStateListener로 FirebaseAuth에 getCurrentUser를 통해 로그인되어있는지 확인합니다. 로그인된 경우 MainActivity로 이동됩니다.</p> <p>1. 앱 아이콘입니다.</p> <p>2. android.support.design의 TextInputLayout을 통해 애니메이션을 겸비한 레이아웃을 구현했습니다.</p> <p>3. FirebaseAuth의 signInWithEmailAndPassword를 사용하며 성공시 MainActivity로 이동합니다.</p> <p>4. 회원가입 페이지로 이동합니다.</p> <p>5. 프로필 이미지를 startActivityForResult로 불러옵니다.</p> <p>6. FirebaseAuth의 createUserWithEmailAndPassword로 회원가입하며 FirebaseStorage와 FirebaseDatabase에 정보를 저장합니다.</p>

## 메인화면 (MainActivity)

주요화면	설명
 <p>1 BottomNavigationView</p> <p>2 pushToken</p> <pre> chattingmon ├── chatrooms └── users     ├── 5Jsr22nX2ORC6YEa0qkpOXU0h592     │   ├── comment: "944!"     │   ├── profileImageUrl: "https://firebasestorage.googleapis.com/v0/..."     │   ├── pushToken: "dIfx7EgSZr0:APA91bFv0YIe9yXvWjPEfrpVp_0eDI"     │   ├── uid: "5Jsr22nX2ORC6YEa0qkpOXU0"     │   └── userName: "김민준"     ├── fJK8TT6e4Y5EUxNyzfEr7ulBvs2     └── s6TfM9huKsamyLLRnGm0ml8KZTp2                     </pre>	<p>1. BottomNavigationView를 이용해 구현하였습니다. 아이템 선택시 OnNavigationItemSelectedListener이 실행되며 FragmentManager의 beginTransaction, replace를 통해 화면이 전환됩니다.</p> <p>2. FirebaseAuth에서 uid를 받고 FirebaseInstanceId에서 Token을 받아 Map에 넣어 해당 user의 FirebaseDatabase pushToken값을 업데이트합니다.</p>

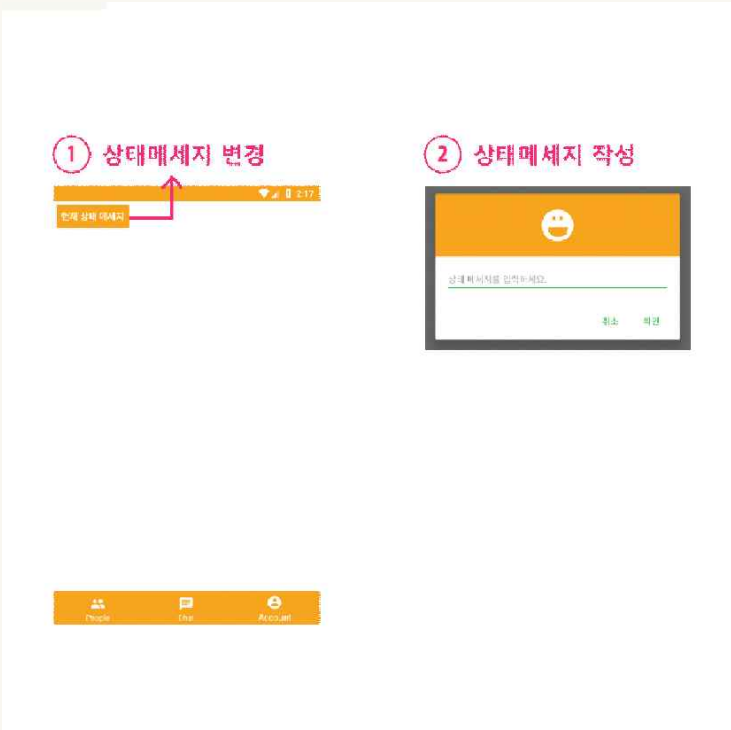
## 친구목록 / 단체 채팅방 개설 (People Fragment)

주요화면	설명
 <p>1. <b>FirebaseDatabase</b></p> <pre> chatrooms ├── chatrooms └── users     ├── 5Jsr22nX20RCsVEaDqkpOXU0b592     │   ├── comment: '333'     │   ├── profileImageUrl: 'https://firebasestorage.googleapis.com/v1/...'     │   ├── pushToken: '42f42EgS2n0APa91bFu0Y92w9yXw8jPEfrpVp_0u0k'     │   ├── uid: '5Jsr22nX20RCsVEaDqkpOXU0'     │   └── username: '김민희'     ├── LJYK6T76x4Y5EJxNyxE7uI0w2     └── u6T0M9huKsamyLLRnGdm18KZTp2         </pre> <p>2. <b>Glide 사용</b> ←</p> <p>3. <b>MessageActivity 이동</b></p> <p>4. <b>단체 채팅방 개설</b></p>	<ol style="list-style-type: none"> <li>1. FirebaseAuth에서 나의 Uid를 가져오고 FirebaseDatabase에서 users의 정보를 ValueEventListener로 가져옵니다. dataSnapshot의 children들을 하나씩 받아서 UserModel class로 받아서 내 uid가 아닌 경우 userModels ArrayList에 추가합니다.</li> <li>2. Binding시 Glide를 이용해 profileImage를 circleCrop하여 불러오며 userName과 comment를 나타냅니다.</li> <li>3. 사람을 누르면 그 사람의 uid를 intent에 넣어 MessageActivity를 실행합니다. ActivityOptions를 사용해 애니메이션 효과도 주었습니다.</li> <li>4. FAB를 누르면 단체 채팅방에 초대할 사람을 고르는 SelectFriendActivity로 이동합니다.</li> </ol>

## 채팅방 리스트 (ChatFragment)

주요화면	설명
 <p>1. <b>채팅방 리스트</b> ←</p> <p>2. <b>채팅방 정보</b></p> <pre> chatrooms ├── chatrooms │   ├── LH2G2B13pAZhg_uadP │   ├── LHGP1m0Pb8xvR457u0 │   └── LH975YK8Pm7Y8n0a │       ├── comment: │       │   ├── 5Jsr22nX20RCsVEaDqkpOXU0b592: true │       │   ├── LJYK6T76x4Y5EJxNyxE7uI0w2: false │       │   └── u6T0M9huKsamyLLRnGdm18KZTp2: true │       └── users │           ├── 5Jsr22nX20RCsVEaDqkpOXU0b592 │           ├── LJYK6T76x4Y5EJxNyxE7uI0w2 │           └── u6T0M9huKsamyLLRnGdm18KZTp2 └── users     </pre>	<ol style="list-style-type: none"> <li>1. FirebaseDatabase에서 나의 uid가 들어있는 chatroom들을 불러옵니다. dataSnapshot으로 불러진 데이터를 List chatModels에 넣으며 다른 List인 keys에 각채팅방의 key를 넣어 관리합니다.</li> <li>2. 각 채팅방의 유저들과 TreeMap의 Collections.reverseOrder를 사용해 마지막 메시지를 FirebaseDatabase에서 받아옵니다. 클릭시 유저수를 파악하여 2명인 경우 일반채팅방 MessageActivity로 3명이상인 경우 GroupMessageActivity로 이동합니다. SimpleDateFormat을 사용해 시간을 표현합니다.</li> </ol>

## 내 계정관리 / 상태메세지 변경 (AccountFragment)

주요화면	설명
	<ol style="list-style-type: none"> <li>1. 버튼 클릭시 상태메세지를 수정할 수 있는 AlertDialog가 띄워집니다.</li> <li>2. 새로 적힌 내용을 FirebaseAuth의 uid를 받고 HashMap을 생성하여 "comment"와 함께 put하고 이 내용을 가지고 FirebaseDatabase에서 user의 child를 업데이트합니다.</li> </ol>

## 채팅방 (MessageActivity/GroupMessageActivity)

주요화면	설명
	<ol style="list-style-type: none"> <li>1. 해당 채팅방의 chatRoomUid를 받아와 FirebaseDatabase에서 해당 chatRoomUid에 해당되는 comments들을 받아옵니다. 각 comments들의 user uid를 확인해 내가 보낸메세지 / 상대방이 보낸 메시지를 따로 bind합니다.</li> <li>2. ChatRoom의 유저들을 받아와 몇 명인지 센뒤 comments안의 readUsers숫자와 비교해 읽지 않은 사람의 수를 구합니다.</li> <li>3. GCM과 Notification을 사용해 상대의 pushToken으로 알림을 전송합니다.</li> <li>4. 체크된 user의 userModel을 받아 2명이상이 체크된 경우 GroupMessageActivity를 생성합니다.</li> </ol>



# Android Game 프로젝트 기술서

## 프로젝트 개요

프로젝트 명	Spin Dragon	
개발기간	2018-04-23 ~ 2018-05-04 2주	
개발인원	1인 개발	
담당역할	개발, 레이아웃 설계	
개발환경	OS	Windows 7
	Development Tool	Android Studio, Genymotion, Adobe Illustrator
	Language	Java
	DB	SharedPreferences

## 프로젝트 소개

### 인트로 화면

주요화면	설명
 <p>The image shows the intro screen of the game 'Spin Dragon'. It features a dark background with a large, stylized 'kile soft' logo at the top. Below it, the game title 'SPIN SHOOTING MONSTER' is displayed. At the bottom, there are two buttons: 'Game Start' and 'EXIT'. Red arrows point to these elements with numbered annotations: 1. Intro animation (pointing to the logo), 2. App Title (pointing to the game title), 3. Game Start (pointing to the 'Game Start' button), and 4. Game End (pointing to the 'EXIT' button).</p>	<p><b>0.</b> FullScreen모드로 설정해놔으며 screenOrientation을 landscape으로 설정했습니다.</p> <p><b>1.</b> Animation, Timer, TimerTask를 이용해 애니메이션 효과를 주었습니다. 이후 MainActivity로 이동합니다.</p> <p><b>2.</b> App Title입니다. MediaPlayer과 Soundpool을 이용하여 R의 raw폴더에 저장된 배경음악이 재생됩니다.</p> <p><b>3.</b> 버튼을 누르면 SoundPool의 버튼클릭 효과음이 나오며 Intent를 통해 GameActivity로 이동합니다.</p> <p><b>4.</b> SoundPool의 버튼클릭 효과음과 함께 게임이 종료됩니다.</p>

## 게임화면

주요화면	설명
	<p>0. SurfaceView를 상속받은 GameView를 사용했으며 그 외 인터페이스는 layout xml을 이용해 구현했습니다. inner class인 GameThread로 객체 만들기, 움직이기, 충돌체크작업, 그려내기를 반복했습니다. 또한 Canvas를 사용해 화면을 변경했습니다.</p> <p>1. 충돌체크로 내 미사일의 좌표와 적군의 좌표가 겹치면 적군은 isDead상태가 되며 점수를 얻습니다.</p> <p>2. 충돌체크에서 나와 적군의 좌표가 겹치면 생명을 하나씩 잃고 다 잃는 경우 GameOverActivity로 이동합니다.</p> <p>3~7 다음페이지에서 설명</p> <p>8. TouchEvent로 touchDown, UP, Move된 좌표를 이용해 내 dy, dx를 결정해 이동합니다.</p> <p>9. 현재까지의 최고 점수가 나타납니다.</p> <p>10. SoundPool의 폭탄 효과음과 모든 Enemy들의 isDead가 true가 됩니다.</p>

## Dialogs

주요화면	설명
	<p>1. GameView의 pauseGame메소드로 gameThread의 pauseThread를 호출하여 isWait을 true로 바꿔 Thread를 wait시킵니다. 재생을 누르면 resumeThread가 호출되어 gameThread의 isWait이 false가 되고 gameThread.notify()가 실행됩니다.</p> <p>2. 애니메이션 효과와 SoundPool효과음과 함께 나타나며 gameThread또한 Pause됩니다.</p> <p>3. 애니메이션 효과와 SoundPool효과음과 함께 나타나며 gameThread또한 Pause됩니다.</p> <p>4. 애니메이션 효과와 SoundPool효과음과 함께 나타나며 gameThread또한 Pause됩니다. 아이콘들은 selector로 되어있으며 OnCheckedChangeListener를 통해 배경음악/소리/진동여부가 변경됩니다.</p>

# Website 프로젝트 기술서

## 프로젝트 개요

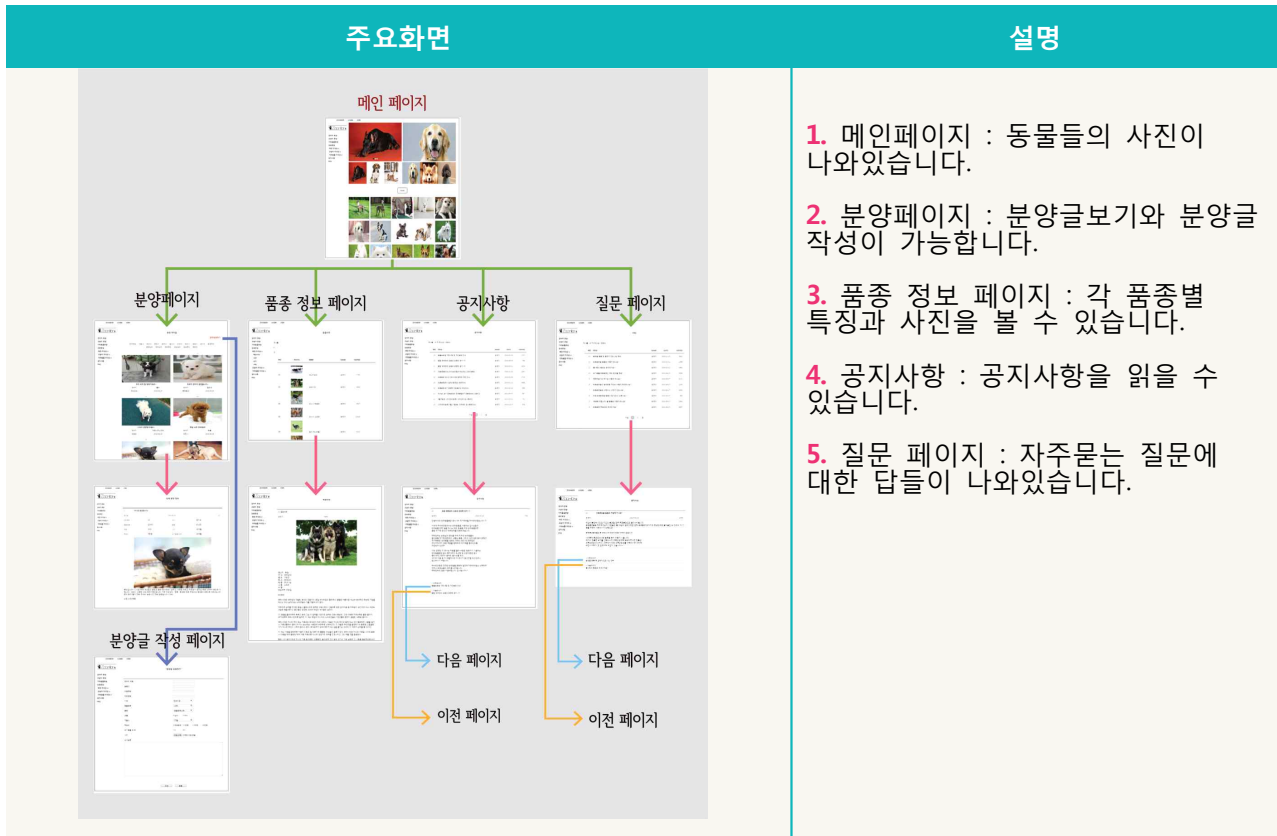
프로젝트 명	ZOONIROR (유기동물 / 가정분양 사이트)	
개발기간	2018-06-11 ~ 2018-06-29 3주	
개발인원	4인 개발 (팀장)	
담당역할	<ul style="list-style-type: none"> <li>- 팀장 : 게시판 설계, 공통 CSS제작, 레이아웃 제작, 공지사항 FAQ 페이지 제작</li> <li>- 팀원 : 분양 게시판 및 분양 글 등록 페이지, 상세페이지 제작</li> <li>- 팀원 : 로고 제작 지식인 페이지 백과사전 리스트 제작</li> <li>- 팀원 : 메인페이지 제작</li> </ul>	
개발환경	OS	Windows 7
	Development Tool	SublimeText3, Adobe Illustrator
	Language	HTML, CSS, JS, JQuery, PHP
	DB	Dothome 웹호스팅, MySQL

## 프로젝트 소개

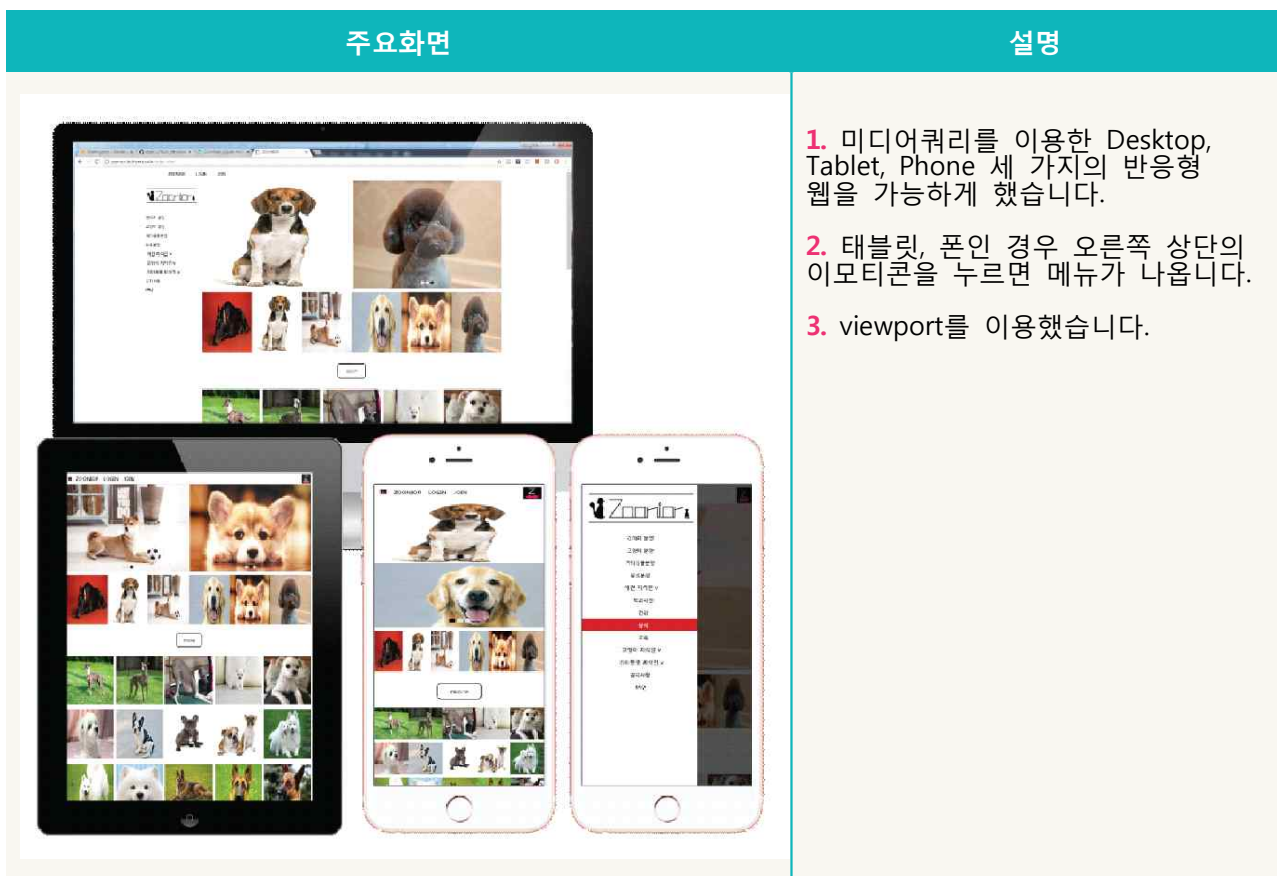
### 화면 설계서

주요화면	설명
	<p>1. 화면을 설계한 모습입니다.</p> <p>2. 미디어 쿼리 사용으로 Web, Tablet, Phone 세 종류의 반응형 웹사이트를 구현했습니다.</p> <p>Web 922px~</p> <p>Tablet 661~991</p> <p>Phone ~660</p>

## 주요화면 와이어프레임



## 반응형 웹 구현



## 홈페이지

주요화면	설명
 <p>1 Header</p> <p>2 Navigation</p> <p>3 Content</p> <p>4 Footer</p>	<p>1. a 태그로 홈페이지로 다시 돌아올 수 있습니다. tablet/phone버전이 되는 경우 오른쪽 상단에 메뉴가 열리는 img가 나타납니다.</p> <p>2. ul 과 li태그로 이루어져있으며 각각의 li 안에있는 a 태그의 href를 통해 페이지가 전환됩니다. tablet/phone의 상황에서는 Navigation 부분이 감춰지고 header상단 메뉴 아이콘이 나타나 index.js에서 메뉴가 눌리면 Navigation부분이 나타나고 사라지는 기능을 JQuery와 fadeIn, fadeOut을 사용해 정의했습니다.</p> <p>3. 페이지별 내용이 들어가는 곳이며 iframe을 사용하여 contents의 내용을 바꿨습니다.</p>

## 동물 분양 페이지

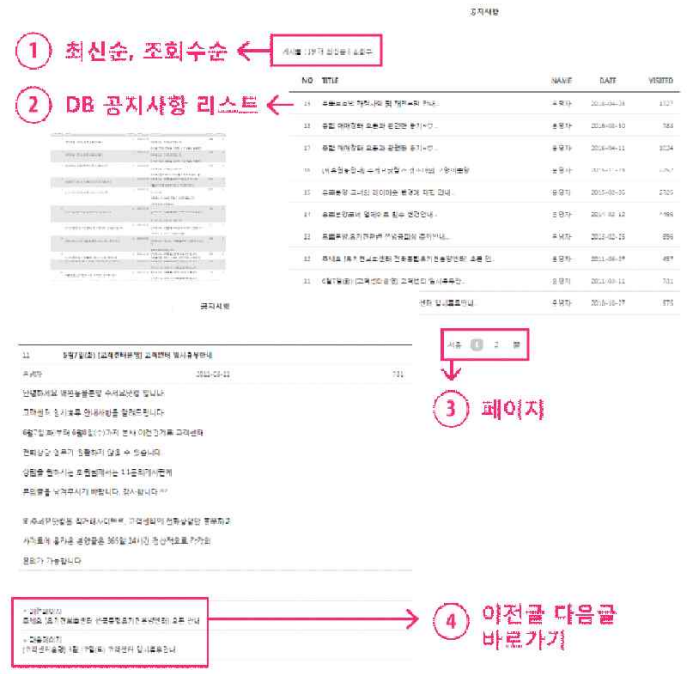
주요화면	설명
 <p>1 새글 작성하기</p> <p>2 지역별 골라보기</p> <p>3 분양 리스트</p> <p>4 자세히 보기</p> <p>5 페이지 사용</p> <p>6 Database</p>	<p>1. a 태그를 이용해 새글작성 페이지(zoo_write.php)로 이동합니다.</p> <p>2. 버튼이 눌리면 a 태그의 local='지역이름'을 다시 이 페이지가 GET방식으로 받아 해당 local 의 record를 php와 MySQL 구문을 이용해 Database에서 최신순으로 받아와 페이지를 다시 생성합니다.</p> <p>3. 받아온 record들을 fetch_assoc한뒤 while문을 돌려서 한줄씩 div와 span으로 생성합니다.</p> <p>4. 클릭시 자세한 페이지(view.php)로 이동합니다.</p> <p>5. 한페이지당 10개의 게시글이 나오며 페이지 변경시 GET방식으로 page 번호를 받아와 새로 생성합니다.</p> <p>6. dothome웹 호스팅 서버의 MySQL을 이용해 table을 생성하였습니다. no(Primary Key, AI), title, id(글작성자), local(지역), kind(동물종류), kinds(품종), rgsex(성별), pic(사진) 등으로 구성되어 있습니다.</p>



## 분양글 작성 페이지

주요화면	설명
 <p>1. 분양글 등록 페이지</p> <p>2. 분양글 DB저장</p>	<p>1. form을 사용해 submit시 zoo_write_update.php에 데이터를 post메소드로 전송했습니다. phone사이즈시 input들이 label아래올수 있도록 div와 span으로 레이아웃을 구성했습니다. text, password, tel, select, radio, file, textarea 와같은 다양한 input, compound button 등을 이용해 form을 구성했습니다.</p> <p>2. zoo_write_update.php에서는 zoonior.php를 library로 작성하여 require_once를 사용해 추가했고 post방식으로 데이터를 받아 Database에 insert문으로추가했습니다. 추가 한뒤 추가한 record의 no를 받아와 get방식으로 no를 함께 전송해 상세페이지를 location.replace를 이용해 오픈했습니다.</p>

## 공지사항 / FAQ페이지

주요화면	설명
 <p>1. 최신순, 조회순순</p> <p>2. DB 공지사항 리스트</p> <p>3. 페이지</p> <p>4. 이전글 다음글 바로가기</p>	<p>1. 최신순을 클릭하면 record를 order by noticeNO desc로 받아오고, 조회수 순으로 하면 order by visited desc로 받아옵니다.</p> <p>2. Database에서 record들을 받아와 div와 span으로 리스트가 만들어칩니다. 한페이지에 10개의 게시물로 이루어져있습니다. 페이지가 변경되면 page를 GET방식으로 같은 페이지에 보내서 재 생성합니다.</p> <p>3. 리스트를 클릭하면 해당 게시물의 no에 해당되는 번호가 GET방식으로 상세 페이지(detail.php)에 보내집니다. 이 no에 해당되는 record를 Database에서 받아와 내용을 채웠습니다.</p> <p>4. 또한 이전글 다음글에 해당되는 정보를 Database에서 받아와 누르면 no에 해당 번호가 전달된 detail.php가 열리게 됩니다.</p>



# Objective C 프로젝트 기술서

## 프로젝트 개요

프로젝트 명	도서관리프로그램	
개발기간	2018-07-02 ~ 2018-07-04 3일	
개발인원	1인개발	
담당역할	개발, 레이아웃 설계	
개발환경	OS	Mac
	Development Tool	XCode
	Language	Objective C
	DB	

## 프로젝트 소개

### 도서관리 프로그램 화면

주요화면	설명
<p>① 전체보기</p> <p>② 입력공간</p> <p>③ 버튼들</p> <p>④ 리스트 공간</p> <p>⑤ 도서 등록</p> <p>⑥ 등록 메시지</p> <p>⑦ 도서 삭제</p> <p>⑧ 도서 검색</p>	<ol style="list-style-type: none"> <li>1. Main.StoryBoard를 이용한 레이아웃 설계를 하고 ViewController의 IBAction 메소드들과 연결 했습니다.</li> <li>2. UILabel과 UITextField를 사용했습니다.</li> <li>3. ViewController에 메소드를 만들어연결했습니다.</li> <li>4. UITextView resultTextView이며 text를 변경해 체크리스트나 결과들을 표시했습니다.</li> <li>5. 도서 등록시 Book 객체 생성뒤 myBook BookManager에 추가합니다.</li> <li>6. 추가 완료 메시지를 resultTextView에 나타냅니다.</li> <li>7. BookManager의 findBook메소드를 사용해 해당책이 있는 경우 removeBook메소드로 삭제합니다.</li> <li>8. BookManager의 findBook메소드를 사용해 그 결과를 나타냅니다.</li> </ol>

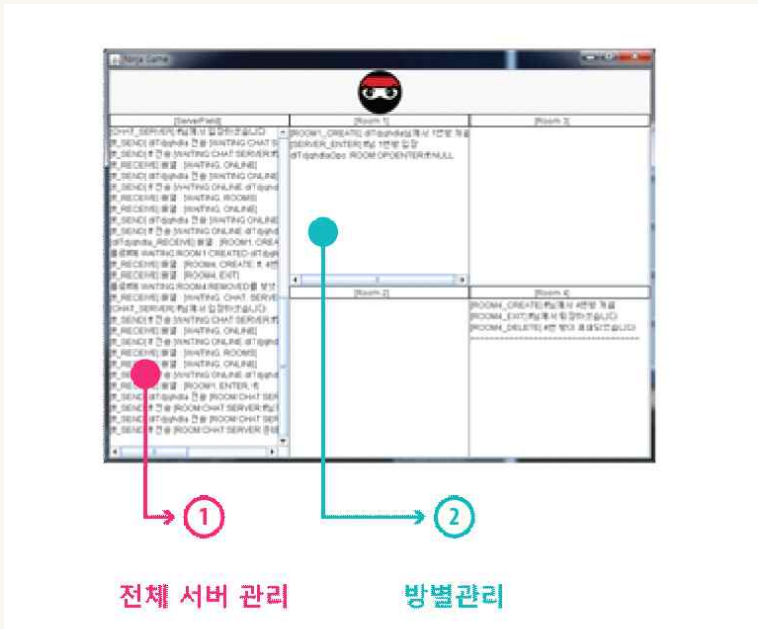
# Java 프로젝트 기술서

## 프로젝트 개요

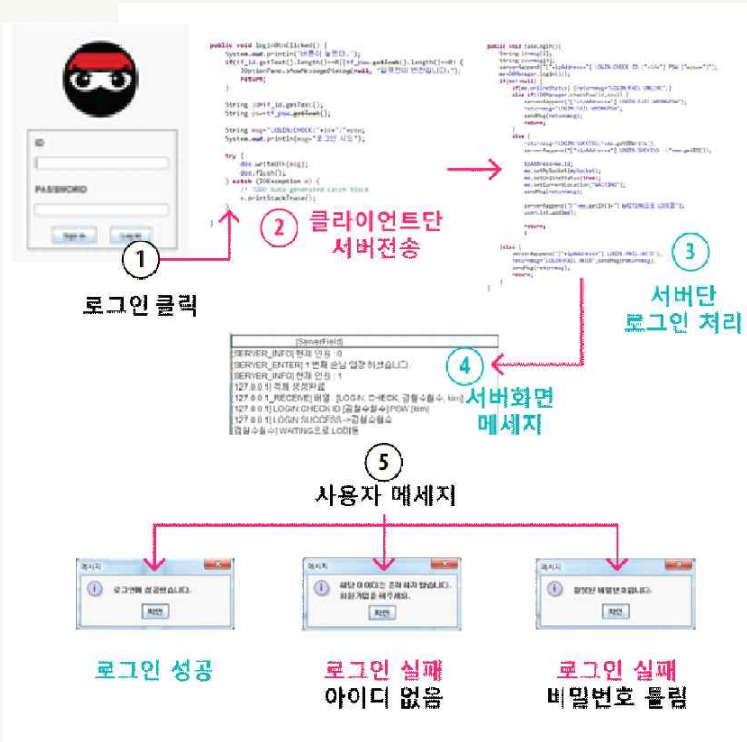
프로젝트 명	2인용 닌자게임	
개발기간	2018-01-31 ~ 2018-02-14 2주	
개발인원	1인개발	
담당역할	기획, 디자인, 캐릭터 제작, 레이아웃 설계, 서버단/클라이언트단 인터페이스 구현, 기능 개발	
개발환경	OS	Windows 7
	Development Tool	Eclipse, Adobe Illustration
	Language	Java
	DB	txt

## 프로젝트 소개


### 서버화면

주요화면	설명
 <p>① 전체 서버 관리</p> <p>② 방별관리</p>	<ol style="list-style-type: none"><li>1. 전체서버 관리<ul style="list-style-type: none"><li>● MainServerThread객체가 접속한 socket들을 받아 각각의 client들을 따로 관장하는 UserThread를 생성해 socket을 넘기고 UserThread는 onlineUserList에 추가됩니다. MainServerThread는 새로운 유저를 받기 위해 대기합니다.</li><li>● UserThread에서 client의 데이터를 받아 처리합니다. 로그인 / 회원가입 / 로비내용들기 기록됩니다.</li></ul></li><li>2. 게임 방별관리<ul style="list-style-type: none"><li>● 각각의 GameRoom에 해당되는 채팅정보나 게임 진행상황들을 socket으로 각각받아 기록합니다.</li></ul></li></ol>

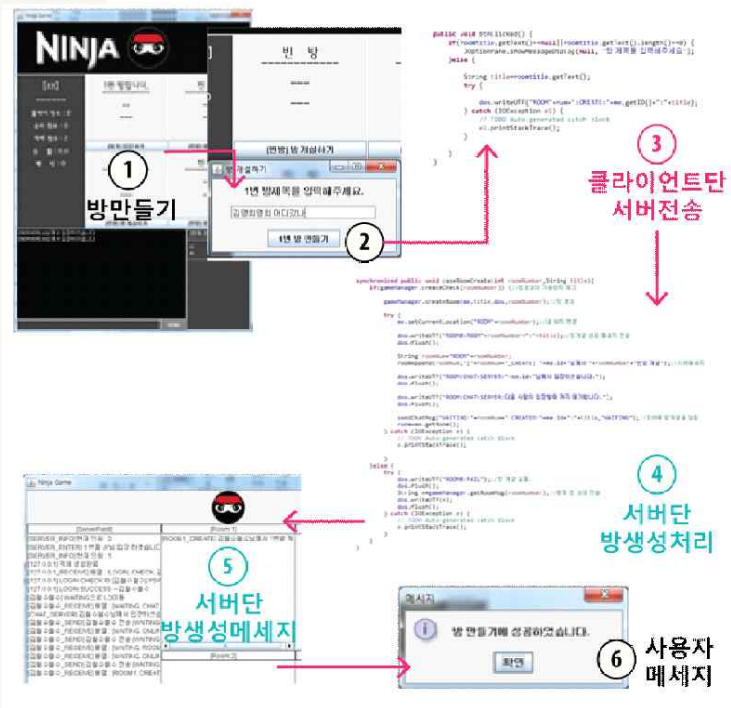
## 로그인 과정

주요화면	설명
 <p>1 로그인 클릭</p> <p>2 클라이언트 서버전송</p> <p>3 서버단 로그인 처리</p> <p>4 서버화면 메시지</p> <p>5 사용자 메시지</p> <p>로그인 성공</p> <p>로그인 실패 아이디 없음</p> <p>로그인 실패 비밀번호 틀림</p>	<ol style="list-style-type: none"> <li>로그인 버튼 클릭하여 ActionListener 익명객체의 actionPerformed메소드가 불리거나 KeyListener의 KeyEvent KeyCode가 Enter인 경우 loginBtnClicked()메소드가 호출됩니다.</li> <li>DataOutputStream을 통해 mysoket으로 writeUTF메소드를 사용해 로그인 정보를 전송합니다.</li> <li>DataInputStream으로 데이터를 받아 유저들 정보를 관장하는 DBManager의 login메소드로 로그인이 가능한지 확인합니다. 로그인 결과가 socket으로 client에게 다시 전송됩니다.</li> <li>로그인확인한 것, 로그인 결과, 현재 인원수 등의 정보들이 server에 작성됩니다.</li> <li>사DataInputStream을 통해 받은 결과에 따라 JOptionPane의 showMessageDialog를 통해 결과를 띄운뒤 로그인 성공시 로비(WaitingPanel) 패널로 ContentPane을 remove한뒤 WaitingPanel을 추가합니다.</li> </ol>

## 회원가입 과정

주요화면	설명
 <p>회원가입</p> <p>회원가입 성공</p> <p>회원가입 실패</p>	<ol style="list-style-type: none"> <li>signUp버튼 클릭시 ActionListener에 의해 SignupFrame이 생성됩니다.</li> <li>SignUp 버튼 클릭시 ActionListener, Enter키를 누른경우 KeyListener 의 메소드를 통해 회원가입 정보가 서버에 dos를 통해 전송됩니다.</li> <li>dis로 받은 정보를 DBManager.isAvailable함수로 통해 아이디 생성 여부를확인해 가능하면 생성한뒤 결과를 다시 client에게 전송합니다.</li> <li>전송된 데이터를 dis로 받아 회원가입 성공여부 메시지를 MatDialog로 띄웁니다.</li> </ol>

## 방개설 과정

주요화면	설명
 <p>1. 방만들기 2. 방제목 입력 후 방만들기 버튼 클릭 3. 클라이언트단 서버전송 4. 서버단 방생성처리 5. 서버단 방생성메세지 6. 사용자메세지</p>	<ol style="list-style-type: none"> <li>1. 방개설 버튼이 클릭되면 JFrame을 생성합니다.</li> <li>2. 방제목 입력 후 방만들기 버튼이 클릭되면 ActionListener의 actionPerformed메소드가 호출됩니다.</li> <li>3. dos로 server에 방제목을 전송합니다.</li> <li>4. 서버에서 방생성을 한 뒤 user의 location을 해당 방번호로 바꾼뒤 현재 location이 waiting인 client들에게 모두 해당 방의 상태변경 메세지를 전송합니다.</li> <li>5. 방생성 개설퍼부와 방입장 메세지 등의 정보가 서버에 올라갑니다.</li> <li>6. 방생성 완료 메세지를 dis를 통해 받으면 MatDialog를 띄운뒤 GameRoomPanel을 새로 생성합니다.</li> </ol>

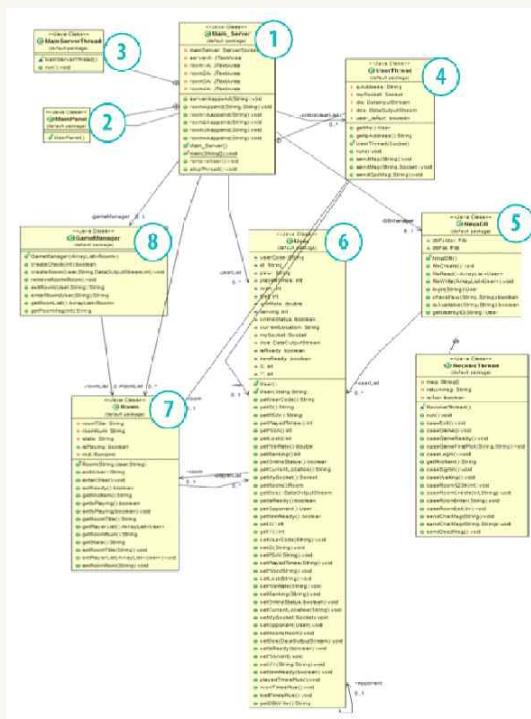
## 게임 진행 과정

주요화면	설명
 <p>1. 방입장 2. 게임시작 3. 자리선택 처리 4. 자리선택 처리 5. 서버메세지 6. 상대위치 처리 7. 게임진행 화면</p>	<ol style="list-style-type: none"> <li>1. 방입장 상대가 입장 할때까지 기다림 두 유저 모두 READY 상태가되면 게임이 시작됨 1:1 채팅 가능</li> <li>2. 게임시작 첫시작시 숨을 장소 선택</li> <li>3. 자리선택 처리 사용자가 선택한 자리를 화면에 표시, 서버 전송</li> <li>4. 두 유저의 자리 선택 좌표 서로에게 전송</li> <li>5. 관리자 페이지에 방1의 게임진행 과정 알림</li> <li>6. 서버에서 받은 상대 좌표를 이용 상대 위치 처리</li> </ol>

## 게임 진행 화면

주요화면	설명
 <p>① 게임 시작</p>  <p>② 게임 진행중</p>  <p>③ 승리 화면</p>  <p>④ 패배 화면</p>	<p>1. READY상태가 되면 해당 정보가 server에 전송되며 상대방에게도 전송됩니다. 둘다 READY가 된 상태에서 서버가 게임시작 메시지를 전송해줍니다.</p> <p>2. TimerThread와 sleep으로 타이머기능을 구현했으며 턴제, 이동이 가능합니다. 타이머게임이 진행중인 모습입니다. DrawingThread와 paintComponent, Graphics를 이용해 화면을 repaint했습니다.</p> <p>3. 공격시 opponent의 hp가 0이하로 떨어지면 server에 dos로 알린뒤 승리화면을 띄우며 게임이 종료됩니다.</p> <p>4. dis로 패배 메시지를 받으면 패배 화면을 띄우며 게임이 종료됩니다.</p>

## [SERVER] 클래스 다이어그램

주요화면	설명
	<p>1. MainServer - 서버 프레임 구동</p> <p>2. MainPanel - 서버 페이지 인터페이스 구형</p> <p>3. MainServerThread - 각각의 멤버들의 연결을 관장하는 오픈형 서버소켓</p> <p>4. UserThread - 멤버별로 각각 돌아가는 소켓Thread</p> <p>5. NinjaDB - 회원명단 관리, 데이터 저장 및 불러오기</p> <p>6. User - 유저 class 각 유저의 정보저장</p> <p>7. Room - 방 class 각 방의 정보 저장 (한방당 유저 두명)</p> <p>8. GameManager - 게임방의 개설과 철거를 담당</p>



# 프로젝트 기술서 1

## 프로젝트 개요

프로젝트 명	주민등록번호/카드번호 로직	
개발기간	2018-11-29 1일	
개발인원	1인 개발	
담당역할	레이아웃 설계, 개발	
개발환경	OS	Windows 7
	Development Tool	Visual Studio
	Language	C
	DB	

## 프로젝트 소개

### CMD 결과창

주요화면	설명
	<ol style="list-style-type: none"> <li>숫자만 16개 입력한 경우</li> <li>주민등록 번호 로직으로 성공한 경우 (특수 문자 '-' 포함)</li> <li>주민등록번호 로직 인증 성공한 경우 (특수문자 미포함)</li> <li>공백, '-' 이외의 문자를 입력 한 경우 = 인증 실패</li> <li>공백 포함 카드번호를 입력 한 경우 = 카드번호 인증 성공</li> </ol>



## 허용 외 문자 입력 확인

주요화면	설명
<pre> while (1) {     char arr[30];     int num = 0;     int c[30];     int error = 0; //영어나 특문입력시 오류      printf("숫자를 입력해주세요\n");     printf("공백, (-)가능 그외 전부 무효\n");      for (int i = 0; i &lt; 30; i++)     {         scanf("%c", &amp;arr[i]); //스캔받는다.          if (arr[i] == 10) //엔터 치면 끝낸다!             break;          if ((arr[i] &gt;= '0') &amp;&amp; (arr[i] &lt;= '9')) //숫자인지 확인         {             c[num] = (int)arr[i] - 48;             num++;         }          else if (!(arr[i] == '-'    arr[i] == ' ')) //그외문자 확인         {             error = 1;         }     } } </pre>	<p>사용자에게 번호를 입력받아 char들을 비교해 허용되지 않은 문자들을 걸러냅니다.</p>

## 주민등록번호 로직 검사

주요화면	설명
<pre> int A = c[0] * 2,     B = c[1] * 3,     C = c[2] * 4,     D = c[3] * 5,     E = c[4] * 6,     F = c[5] * 7,     G = c[6] * 8,     H = c[7] * 9,     I = c[8] * 2,     J = c[9] * 3,     K = c[10] * 4,     L = c[11] * 5;  int M = (11 - ((A + B + C + D + E + F + G + H + I + J + K + L) % 11)) % 10;  if (M == c[12]) //주민번호 유효성확인 {     printf("주민번호가 맞습니다. =====\n\n");     break; }  else //그외 {     no();     break; } </pre>	<p>주민등록번호 로직을 통해 유효성을 검사합니다.</p>