클래스 다이어그램

B511135 이범학, B711166 장혜리

# 장혜리

### LoginActivity 클래스

|  |
| --- |
| * private SignInButton btn\_login\_google * private LoginButton btn\_login\_facebook * private static final int RC\_SIGN\_IN * private FirebaseAuth mFirebaseAuth * private GoogleSignInClient mGoogleSignInClient * private CallbackManager mCallbackManager * String TAG |
| * protected void onCreate(Bundle savedInstanceState) * public void onStart() * public void onActivityResult(int requestCode, int resultCode, Intent data) * private void firebaseAuthWithGoogle(GoogleSignInAccount acct) * private void handleFacebookAccessToken(AccessToken token) * private void myStartActivity(Class c) * public void checkUser(FirebaseUser user) * private void setLoginButton() |

### ProfileImage 클래스

|  |
| --- |
| * private StorageReference mStorageRef * File cacheDir * File cacheFile * String userNum * String TAG |
| * public ProfileImage(File cacheDir, String userNum) * public void showProfileImage(ImageView iv) * public void uploadProfileImage(Uri profileImageUri) * public boolean downloadProfileImage(String userNum) * public void makeCacheProfileImage(File localFile) * public void copyFile(File src, File dst) * public void setProfileImageCache(ImageView iv) |

### FriendListActivity 클래스

|  |
| --- |
| * Button btn\_my\_friend\_list * final Map<String, Object> emptyObject * private boolean isFreindsFragment * final String TAG |
| * protected void onCreate(Bundle savedInstanceState) * public void switchFragment(Fragment fr) |

### FriendRequetsFragment 클래스

|  |
| --- |
| * FriendListAdapter adapter * ListView lv\_friend\_req * final String TAG |
| * public FriendRequetsFragment() * public void onCreate(Bundle savedInstanceState) * public View onCreateView(LayoutInflater inflater, ViewGroup container,   Bundle savedInstanceState) |

### FriendsFragment 클래스

|  |
| --- |
| * FriendListAdapter adapter * ListView lv\_friends * final String TAG |
| * public FriendsFragment() * public void onCreate(Bundle savedInstanceState) * public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) |

### FriendList VO 클래스

|  |
| --- |
| * private String userNum * private String name * private String stateMessage * private String mbti |
| * public FriendVO(String userNum, String name, String stateMessage, String mbti) * public String getUserNum() * public String getName() * public String getStateMessage() * public String getMbti() |

### FriendList Adapter 클래스

|  |
| --- |
| * private Context context * private ArrayList<FriendVO> Data |
| * public FriendListAdapter(Context context, int textViewResourceId, ArrayList<FriendVO> list) * public Object getItem(int position) * public long getItemId(int position) * public int getCount() * public View getView(int position, View convertView, ViewGroup parent) |

### ViewHolder 클래스

|  |
| --- |
| * ImageView iv\_profileImage * TextView tv\_nickname * TextView tv\_state\_message * Button btn\_accept |
|  |

### FriendList 클래스

|  |
| --- |
| * FirebaseUser user * FirebaseFirestore db * String UserNum * public ArrayList<FriendVO> Friends * public ArrayList<FriendVO> friendRequets * public ArrayList<FriendVO> sentFriendRequests |
| * void sendFriendRequest(User user) * void acceptFriendRequest(User user) * *void rejectFriendRequest(User user) : 미구현* * *void deleteFriend(User user) : 미구현* |

# 이범학

### JoinActivity 클래스

|  |
| --- |
| * String nickname * String gender * String age * String mbti * String address * String stateMessage * String userImage * FirebaseAuth mfirebaseAuth * FirebaseUser currentUser |
| * void profileUpdate() |

### myProfileActivity 클래스

|  |
| --- |
| * String nickname * String gender * String age * String mbti * String address * String stateMessage * String userImage * FirebaseAuth mfirebaseAuth * FirebaseUser currentUser * FirebaseFirestore db |
| * void updateProfile (User user) |

### MainActivity 클래스

|  |
| --- |
| * FirebaseAuth mfirebaseAuth * FirebaseUser currentUser |
| * void init( ) * void logout(FirebaseAuth mfirebaseAuth) |

### MemberInfo 클래스

|  |
| --- |
| * String userNum * String nickname * String userImage * String gender * String age * String mbti * String address * String stateMessage |
| * public void InitProfile (User user) * 각 멤버 변수별 getter, setter |

### Matching 클래스

|  |
| --- |
| * double latitude * double longitude |
| * void matchUser(User user) * void showCurrentLocation (Location location) * void addPictures (Location location) * void onRequestPermissionsResult |

### Searching 클래스

|  |
| --- |
| * String gender * String age * String mbti |
| * void searchingAlgorithm( ) * MemberInfo memberinfo * void SearchingPerson() |

# 미구현

### ChatList 클래스

|  |
| --- |
| * String userNum :회원 일련번호 * int[] chatNum : 채팅방 일련번호 |
| * public void createChatList(ChatList chatList) : 채팅 리스트를 생성한다. * public void getChatList(ChatList chatList) : 특정 유저의 채팅 리스트를 가져온다. * public void addChat(ChatList chatList) : 채팅 리스트에 채팅방을 추가한다 * public void deleteChat(ChatList chatList) : 채팅 리스트에서 채팅방을 삭제한다. |

## 

### Chat 클래스

|  |
| --- |
| * int chatNum : 채팅방 일련번호 * String chatName : 채팅방 이름 * String chatImage : 채팅방 대표 이미지 * String otherProfileImage : 상대 유저의 프로필 이미지 * String otherNickname : 상대 유저의 닉네임 * short chatParticipantNum : 채팅방 참여 인원 * Message[] messges : 해당 채팅방의 메세지들 * String lastSentTime : 마지막 메세지 발송 시각 |
| * public void createChat (Chat chat) : 채팅방을 생성한다. * public void getChatInfo(Chat chat) : 채팅방 정보를 가져온다. * public void getMessageList(Chat chat): 메세지 리스트를 가져온다. * public void updateMessgaeList(Chat chat) : 메세지 리스트를 전달한다. |

### 

### Message 클래스

|  |
| --- |
| * unsigned int chatNum : 채팅방 일련번호 * unsigned int senderNum : 발신자의 회원일련번호 * unsigned int messageNum : 메세지 일련번호 * String text :내용 * String sentTime : 발송 시각 |
| * public void sendMessage(Message message) : Message를 전달한다. * public void recieveMessage(Message message) : Message를 가져온다. |

### Board 클래스

|  |
| --- |
| * unsigned int boardNum : 게시판 일련번호 * String boardTitle : 게시판 제목 |
| * public void getList(Board board) : 게시글 리스트 확인 |

### Article 클래스

|  |
| --- |
| * unsigned int boardNum : 게시판 일련번호 * String writerId: 작성자 ID * String articleTitle : 게시글 제목 * String articleContents : 게시글 내용 * String articleCreationDate : 게시글 작성 날짜 * unsigned int viewCount : 조회 수 * unsigned int replyCount : 댓글 수 * unsigned int likeCount : 좋아요 수 |
| * public void addArticle(Article article) : 게시글 추가 * public void getArticle(Article article) : 게시글 확인 * public void updateArticle(Article article) : 게시글 수정 * public void delArticle(Article article) : 게시글 삭제 |

### Reply 클래스

|  |
| --- |
| * unsigned int replyNum : 댓글 일련번호 * unsigned int articleNum : 게시글 일련번호 * String replyWriterId : 댓글 작성자 ID * String replyContents : 댓글 내용 * String replyCreationDate : 댓글 작성 날짜 |
| * public void addReply(Reply reply) : 댓글 추가 * public void getReply(Reply reply) : 댓글 확인 * public void updateReply(Reply reply) : 댓글 수정 * public void delReply(Reply reply) : 댓글 삭제 |

# ……………..

### mainActivity

* 로그인 한 유저가 어플을 처음 실행했을 때 보이는 화면
* 로그인 이후 접근 가능
* menu 구현

### chatListActivity

* 채팅 리스트를 보여주는 화면

### profileActivity

* 타 유저의 프로필을 보여주는 화면
* 수정버튼 없음

### chatActivity

* 하나의 채팅방을 보여주는 화면
* 채팅방의 Chat 클래스 정보를 가져온다.

### matchingActivity

* 유저 매칭을 위한 화면
* 원하는 매칭상대를 위한 옵션설정
* 매칭 결과를 한명씩 제안한다
* 마음에 안드는 결과는 스킵 가능

### searchActivity

* 검색 조건설정

### boardActivity

* Board 클래스에 저장된 게시글들을 나열하여 보여준다.

### articleActivity

* Article 클래스의 사진이나 글 정보를 보여준다.
* 게시글의 Reply클래스 정보를 보여준다.