

Project:

Chatty Application

จัดทำโดย

นายจิระเดช ปัญจพรศิวนกุล 5804062663035

ชลภััส มุกสกุล 5804062660184

Section: CSB

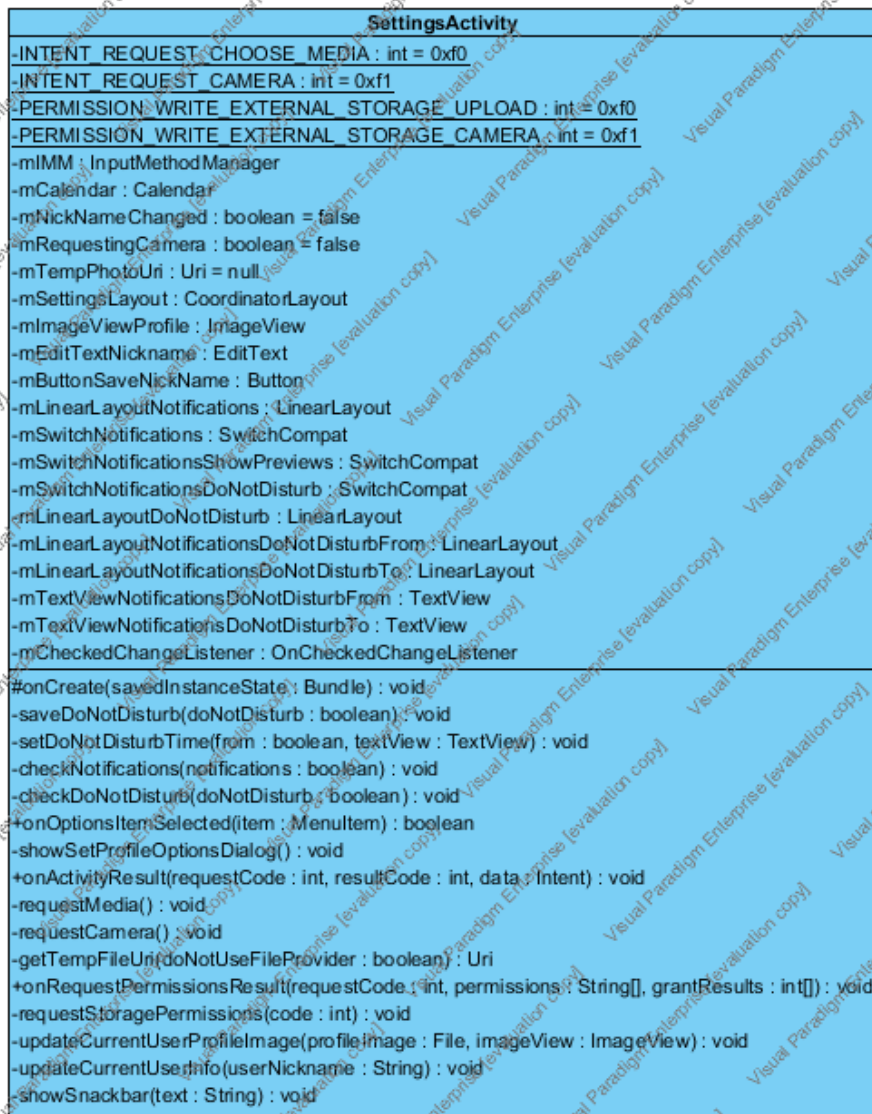
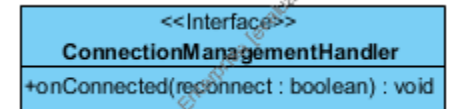
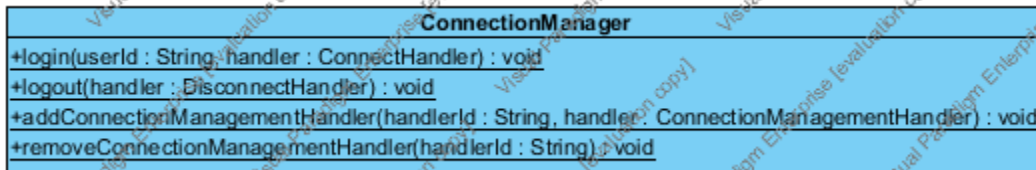
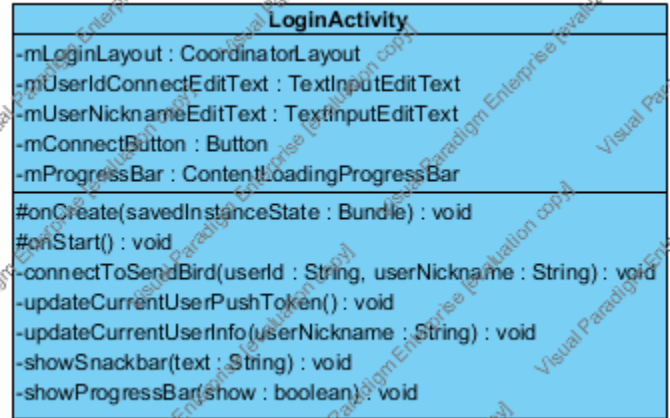
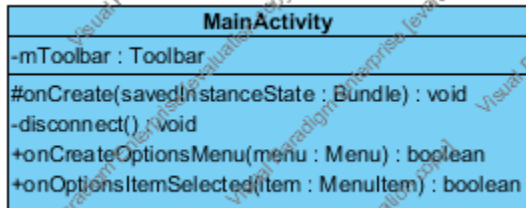
เสนอ

ดร.กอบเกียรติ สระอุบล

Advance Programing 040613424

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

## Diagram (Architecture)



GroupChannelActivity
-mOnBackPressedListener : onBackPressedListener
#onCreate(savedInstanceState : Bundle) : void
+setOnBackPressedListener(listener : onBackPressedListener) : void
+onBackPressed() : void
+onOptionsItemSelected(item : MenuItem) : boolean
~setActionBarTitle(title : String) : void

GroupChannelListFragment
+EXTRA_GROUP_CHANNEL_URL : String = "GROUP_CHANNEL_URL"
-INTENT_REQUEST_NEW_GROUP_CHANNEL : int = 302
-CHANNEL_LIST_LIMIT : int = 15
-CONNECTION_HANDLER_ID : String = "CONNECTION_HANDLER_GROUP_CHANNEL_LIST"
-CHANNEL_HANDLER_ID : String = "CHANNEL_HANDLER_GROUP_CHANNEL_LIST"
-mRecyclerView : RecyclerView
-mLayoutManager : LinearLayoutManager
-mCreateChannelFab : FloatingActionButton
-mChannelListQuery : GroupChannelListQuery
-mSwipeRefresh : SwipeRefreshLayout
-mChannelListAdapter : GroupChannelListAdapter
+newInstance() : GroupChannelListFragment
*onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View
+onResume() : void
+onPause() : void
+onActivityResult(requestCode : int, resultCode : int, data : Intent) : void
-setUpRecyclerView() : void
-setUpChannelListAdapter() : void
-showChannelOptionsDialog(channel : GroupChannel) : void
-setChannelPushPreferences(channel : GroupChannel, on : boolean) : void
~enterGroupChannel(channel : GroupChannel) : void
~enterGroupChannel(channelUrl : String) : void
-refresh() : void
-refreshChannelList(numChannels : int) : void
-loadNextChannelList() : void
-leaveChannel(channel : GroupChannel) : void

GroupChannelListAdapter
-mChannelList : List<GroupChannel>
-mContext : Context
-mSimpleTargetIndexMap : ConcurrentHashMap<SimpleTarget<Bitmap>, Integer>
-mSimpleTargetGroupChannelMap : ConcurrentHashMap<SimpleTarget<Bitmap>, GroupChannel>
-mChannelImageNumMap : ConcurrentHashMap<String, Integer>
-mChannelImageViewMap : ConcurrentHashMap<String, ImageView>
-mChannelBitmapMap : ConcurrentHashMap<String, SparseArray<Bitmap>>
-mItemClickListener : OnItemClickListener
-mItemLongClickListener : OnItemLongClickListener
~GroupChannelListAdapter(context : Context)
~clearMap() : void
+load() : void
+save() : void
+onCreateViewHolder(parent : ViewGroup, viewType : int) : ViewHolder
+onBindViewHolder(holder : ViewHolder, position : int) : void
+getItemCount() : int
~setGroupChannelList(channelList : List<GroupChannel>) : void
~addLast(channel : GroupChannel) : void
~updateOrInsert(channel : BaseChannel) : void
~setOnItemClickListener(listener : OnItemClickListener) : void
~setOnItemLongClickListener(listener : OnItemLongClickListener) : void

CreateGroupChannelActivity
+EXTRA_NEW_CHANNEL_URL : String = "EXTRA_NEW_CHANNEL_URL"
~STATE_SELECT_USERS : int = 0
~STATE_SELECT_DISTINCT : int = 1
-mNextButton : Button
-mCreateButton : Button
-mSelectedIds : List<String>
-mIsDistinct : boolean = true
-mCurrentState : int
-mToolbar : Toolbar
#onCreate(savedInstanceState : Bundle) : void
+onOptionsItemSelected(item : MenuItem) : boolean
~setState(state : int) : void
+onUserSelected(selected : boolean, userId : String) : void
+onDistinctSelected(distinct : boolean) : void
-createGroupChannel(userIds : List<String>, distinct : boolean) : void

GroupChatFragment
-LOG_TAG : String = GroupChatFragment.class.getSimpleName() -CHANNEL_LIST_LIMIT : int = 30 -CONNECTION_HANDLER_ID : String = "CONNECTION_HANDLER_GROUP_CHAT" -CHANNEL_HANDLER_ID : String = "CHANNEL_HANDLER_GROUP_CHANNEL_CHAT" -STATE_NORMAL : int = 0 -STATE_EDIT : int = 1 -STATE_CHANNEL_URL : String = "STATE_CHANNEL_URL" -INTENT_REQUEST_CHOOSE_MEDIA : int = 301 -PERMISSION_WRITE_EXTERNAL_STORAGE : int = 13 -EXTRA_CHANNEL_URL : String = "EXTRA_CHANNEL_URL" -mIMM : InputMethodManager -mFileProgressHandlerMap : HashMap<SendMessageWithProgressHandler, FileMessage> -mRootLayout : RelativeLayout -mRecyclerView : RecyclerView -mLayoutManager : LinearLayoutManager -mMessageEditText : EditText -mMessageSendButton : Button -mUploadFileButton : ImageButton -mCurrentEventLayout : View -mCurrentEventText : TextView -mChannel : GroupChannel -mChannelUrl : String -mPrevMessageListQuery : PreviousMessageListQuery -mIsTyping : boolean -mCurrentState : int = com.example.asus.chatty.sample.groupchannel.GroupChatFragment.STATE_NORMAL -mEditingMessage : BaseMessage = null -mChatAdapter : GroupChatAdapter
+newInstance(channelUrl : String) : GroupChatFragment +onCreate(savedInstanceState : Bundle) : void +onCreateView(inflater : LayoutInflater, container : ViewGroup, savedInstanceState : Bundle) : View -refresh() : void +onResume() : void +onPause() : void +onDestroy() : void +onSaveInstanceState(outState : Bundle) : void +onCreateOptionsMenu(menu : Menu, inflater : MenuInflater) : void +onOptionsItemSelected(item : MenuItem) : boolean +onActivityResult(requestCode : int, resultCode : int, data : Intent) : void -setUpRecyclerView() : void -setUpChatListAdapter() : void -showMessageOptionsDialog(message : BaseMessage, position : int) : void -setState(state : int, editingMessage : BaseMessage, position : int) : void +onAttach(context : Context) : void -retryFailedMessage(message : BaseMessage) : void -displayTyping(typingUsers : List<Member>) : void -requestMedia() : void -requestStoragePermissions() : void -onFileMessageClicked(message : FileMessage) : void -showDownloadConfirmDialog(message : FileMessage) : void -updateActionBarTitle() : void -sendUserMessageWithUrl(text : String, url : String) : void -sendUserMessage(text : String) : void -setTypingStatus(typing : boolean) : void -sendFileWithThumbnail(uri : Uri) : void -editMessage(message : BaseMessage, editedMessage : String) : void -deleteMessage(message : BaseMessage) : void



Class diagram แบบเต็ม

<https://goo.gl/jDmKrS>

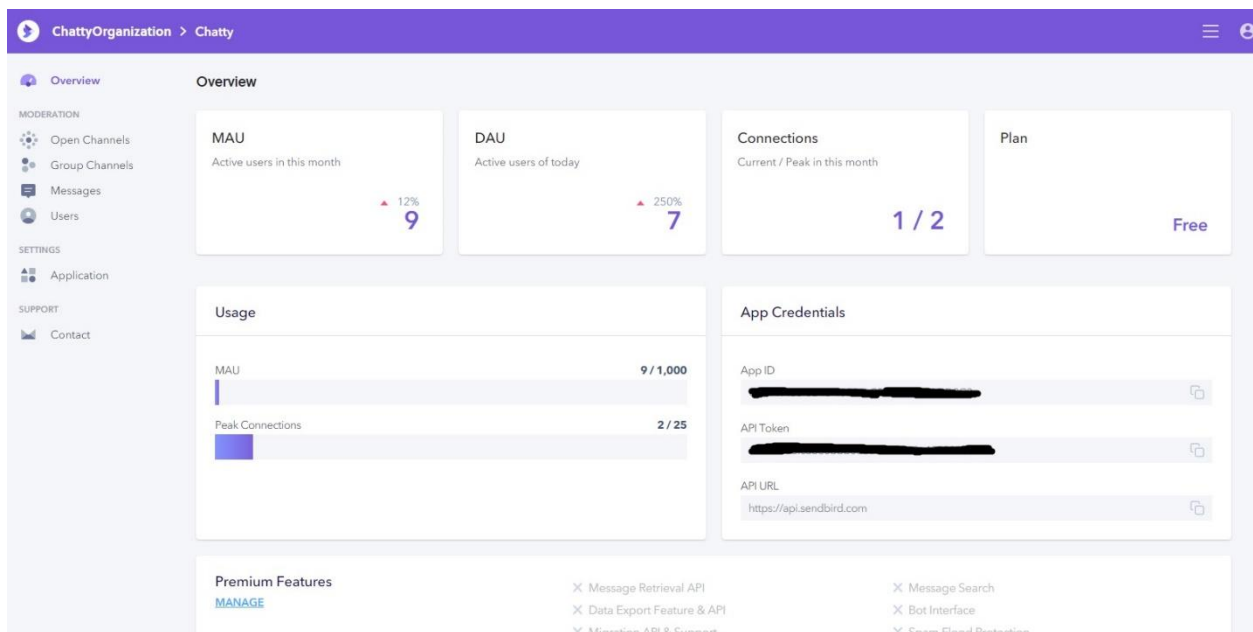


## หลักการทำงาน

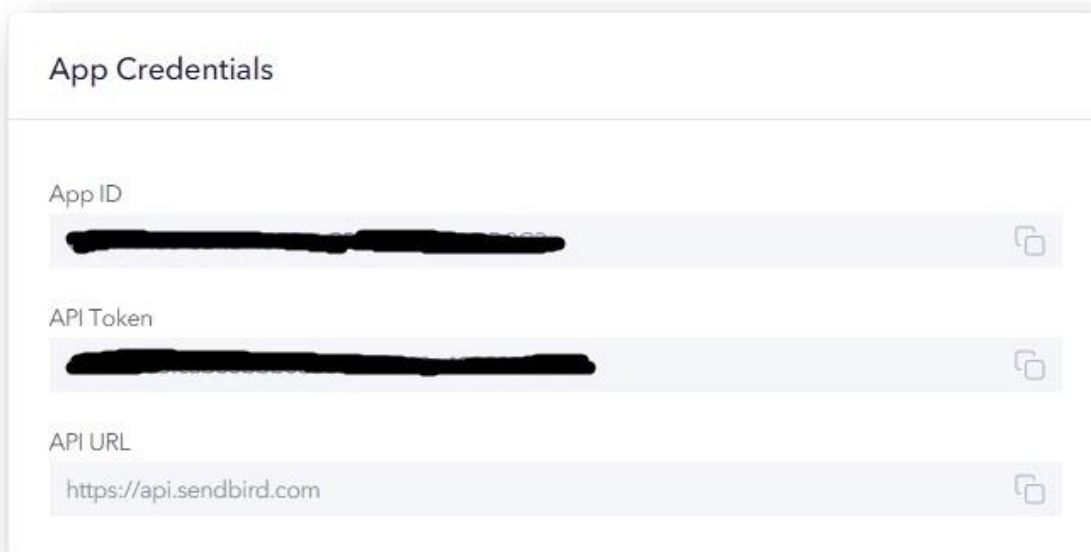
Chatty Application นี้ใช้ API ที่มีชื่อว่า Sendbird ซึ่งเป็น API ที่สร้างขึ้นมาเพื่ออำนวยความสะดวกในการทำ application จำพวก chat โดยเฉพาะ โดย Sendbird จะเป็นตัวกลางในการติดต่อกับ back-end ให้เราซึ่ง back-end ที่ Sendbird ติดต่อให้หนึ่งในนั้นก็คือ firebase ของ android studio เอง ซึ่งการใช้ API นี้จะทำให้เราสามารถสร้าง Chat Application ได้ง่ายๆโดยไม่ต้องเขียนโค้ดในส่วนของ back-end เอง

ในส่วนของการจัดการกับ back-end นั้น เราก็สามารถจัดการผ่าน dashboard ของ Sendbird ได้เลย

(ตัวอย่าง dashboard)



เมื่อเริ่มต้น Application จะรัน class BaseApp ก่อน เพราะ class นี้ extends มาจาก class Application ซึ่งด้วยคุณสมบัติของ class Application จะทำให้ class ที่สืบทอดมาถูกเรียกใช้ก่อน class หรือ activity อื่น ๆ แม้แต่ตัว Launcher activity เองโดยจุดประสงค์ของการ extends Application นี้คือเพื่อการผูกเชื่อม App ของเรากับ server และ dashboard ที่จะเป็นของ App ของเราเท่านั้น (Unique) โดย key ก็คือ App ID ที่สามารถเข้าไปก๊อปปี้ได้มาจาก dashboard ของ App ของเรา



```
BaseApp.java
BaseApp onCreate()
1 package com.example.asus.chatty.sample.main;
2
3
4 import ...
5
6
7
8
9 public class BaseApp extends Application {
10
11     private static final String APP_ID = "XXXXXXXXXXXX";
12
13     @Override
14     public void onCreate() {
15         super.onCreate();
16         PreferenceUtils.init(getApplicationContext());
17
18         // Our App id from Sendbird dashboard
19         SendBird.init(APP_ID, getApplicationContext());
20     }
21 }
```

หลังจากรัน BaseApp ไปแล้วก็จะเริ่ม Launcher Activity ซึ่งใน App ของเราจะเป็น LoginActivity จะเป็นหน้าสำหรับการ login โดยในตอนนี้จะเป็นการ login แบบไม่มี authentication คนที่จะเข้ามาใช้ App จะสามารถเข้าใช้ได้เลยถึงแม้จะไม่ได้สมัครเพราะเมื่อผู้ใช้กรอกไอดีเข้าไปหากไอดีนั้นไม่มีอยู่ในระบบ App จะทำการเพิ่มลงในฐานข้อมูลของ Sendbird โดยอัตโนมัติ และผู้ใช้อีกจะต้องกำหนด nickname ของตัวเองเพื่อใช้เป็นชื่อที่แสดงใน App

เมื่อผู้ใช้กด connect จะทำการเรียกใช้ class ConnectionManager เพื่อตั้งค่าตัวแปร SharedPreferences เช่น nickname ของผู้ใช้ และทำการ login เข้าฐานข้อมูลของ Sendbird ผ่าน ConnectionHandler ของ Sendbird และ Intent ไปยังหน้า MainActivity

```
// userid , usernickname
private void connectToSendBird(final String userId, final String userNickname) {
    // Show the loading indicator
    showProgressBar(true);
    mConnectButton.setEnabled(false);

    ConnectionManager.login(userId, (user, e) -> {
        // success login Method
        showProgressBar(false);

        if (e != null) {
            // Error!
            Toast.makeText(
                context, LoginActivity.this, text: "" + e.getCode() + ": " + e.getMessage(),
                Toast.LENGTH_SHORT
            ).show();

            // show error in snackbar
            showSnackbar(text: "Login to SendBird failed");
            mConnectButton.setEnabled(true);
            PreferenceUtils.setConnected(false);
            return;
        }

        PreferenceUtils.setNickname(user.getNickname());
        PreferenceUtils.setProfileUrl(user.getProfileUrl());
        PreferenceUtils.setConnected(true);

        // Update the user's nickname
        updateCurrentUserInfo(userNickname);
        updateCurrentUserPushToken();

        // intent to main activity
        Intent intent = new Intent(context, LoginActivity.this, MainActivity.class);
        startActivity(intent);
        finish();
    });
}
```

```
public static void login(String userId, final SendBird.ConnectHandler handler) {
    SendBird.connect(userId, (user, e) -> {
        if (handler != null) {
            handler.onConnected(user, e);
        }
    });
}
```



ในหน้า MainActivity จะเป็นหน้า App ที่เป็นทางผ่านไปยังส่วนต่าง ๆ ของ App โดยส่วนสำคัญของหน้าจะมีปุ่ม disconnect สำหรับการตัดการเชื่อมต่อจากระบบ ปุ่มนี้สำหรับหน้า setting และที่สำคัญที่สุดคือปุ่มสำหรับเข้าสู่รายชื่อห้องแชท หน้านี้จะมี onClicklistener ไว้ดักจับการกดปุ่ม เพื่อเข้าสู่หน้าต่าง ๆ

```
// สำหรับ group channel
findViewById(R.id.linear_layout_group_channels).setOnClickListener((v) -> {
    Intent intent = new Intent( packageContext: MainActivity.this, GroupChannelActivity.class);
    startActivity(intent);
});

// สำหรับ disconnect
findViewById(R.id.button_disconnect).setOnClickListener((v) -> {
    //disconnect
    disconnect();
});
}
```

เมื่อกดปุ่ม disconnect ระบบจะไปเรียกใช้ logout handler ของ Sendbird เพื่อออกจากฐานข้อมูลแล้ว intent เข้าสู่หน้า LoginActivity

```
//disconnect
private void disconnect() {
    SendBird.unregisterPushTokenAllForCurrentUser((e) -> {
        if (e != null) {
            // Error!
            e.printStackTrace();
        }
    });
    ConnectionManager.logout(() -> {
        PreferenceUtils.setConnected(false);
        Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
        startActivity(intent);
        finish();
    });
}
```

เมื่อผู้ใช้เข้าสู่หน้ารายชื่อห้องแชทจะเข้าสู่ `GroupChannelActivity` เมื่อเข้าหน้านี้ระบบจะเรียกใช้ `GroupChannelListFragment` เพื่อเรียกรายชื่อห้องที่เคยสร้างเอาไว้ออกมาโดยห้องแต่ละห้องก็คือ 1 fragment แต่ละ fragment จะมีตัวดักจับ `LongClickListener` เพื่อเปิด Dialog ให้ผู้ใช้เลือกว่าจะออกจาก chat หรือไม่

เมื่อกดเข้าห้องแชท method `onCreate()` จะทำงานเพื่อโหลดข้อความจาก server ผ่านทาง handler ของ `Sendbird` โดย `Sendbird` จะเก็บห้องแชทแต่ละห้องเป็น URL ที่ผูกติดกับ `UserID` และข้อความจะเก็บอยู่ในรูป `String Array`

เมื่อออกจาก App ด้วยปุ่มย้อนกลับ method `onDestroy()` จะทำงานแล้วเซฟข้อความลงใน server โดยผูก `UserID` เข้ากับ URL ของ `String Array`

เมื่อออกจาก App ด้วยปุ่ม Home method `onPause()` จะทำงานแล้วทำการปิด `ConnectionHandler` เพื่อตัดการเชื่อมต่อกับห้องแชทชั่วคราวป้องกันการป้อนข้อความลง server โดยไม่ได้ตั้งใจ

เมื่อกลับเข้ามาใน App method `onResume()` จะทำงานแล้วโหลดข้อความกลับมา นอกจากนั้นใน method นี้ยังมีการสร้าง channel handler ที่เป็นตัวจัดการการส่งข้อความในห้องแชทซึ่งมีการ override method ที่จำเป็น

ในห้องแชทเราสามารถอัปโหลดรูปภาพหรือวิดีโอได้โดยการกดปุ่ม + เมื่อกดแล้วจะเรียก method `requestMedia` เพื่อเข้าสู่การเลือกรูปภาพหรือวิดีโอ

ในหน้านี้ยังมี Floating Action Button เมื่อกดเข้าไปจะเป็น Activity ที่ใช้สร้างห้องแชทเมื่อกดเข้าไปจะแสดง user ทั้งหมดที่อยู่ในฐานข้อมูล สามารถเลือกคนเดียวเพื่อแชท 1-1 หรือเลือกหลายคนเพื่อแชทกรุ๊ปได้โดยใช้ method ของ `Sendbird` เป็นตัวช่วย

```

@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_group_channel);

    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_group_channel);
    setSupportActionBar(toolbar);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setHomeAsUpIndicator(R.drawable.arrow_left);
    }

    if (savedInstanceState == null) {
        // Load list of Group Channels
        Fragment fragment = GroupChannelListFragment.newInstance();

        FragmentManager manager = getSupportFragmentManager();
        manager.popBackStack();

        manager.beginTransaction()
            .replace(R.id.container_group_channel, fragment)
            .commit();
    }

    String channelId = getIntent().getStringExtra("name:groupChannelUrl");
    if (channelId != null) {
        // If started from notification
        Fragment fragment = GroupChatFragment.newInstance(channelId);
        FragmentManager manager = getSupportFragmentManager();
        manager.beginTransaction()
            .replace(R.id.container_group_channel, fragment)
            .addToBackStack(null)
            .commit();
    }
}

```

```

@Override
public void onResume() {
    Log.d("tag: \"LIFECYCLE\", \"msg: \"GroupChannelListFragment onResume()\"");

    ConnectionManager.addConnectionManagementHandler(CONNECTION_HANDLER_ID, new ConnectionManager() {
        @Override
        public void onConnected(boolean reconnect) { refresh(); }
    });

    SendBird.addChannelHandler(CHANNEL_HANDLER_ID, new SendBird.ChannelHandler() {
        @Override
        public void onMessageReceived(BaseChannel baseChannel, BaseMessage baseMessage) {
        }

        @Override
        public void onChannelChanged(BaseChannel channel) {
            mChannelListAdapter.clearMap();
            mChannelListAdapter.updateOrInsert(channel);
        }

        @Override
        public void onTypingStatusUpdated(GroupChannel channel) {
            mChannelListAdapter.notifyDataSetChanged();
        }
    });

    super.onResume();
}

```

```

private void createGroupChannel(List<String> userIds, boolean distinct) {
    GroupChannel.createChannelWithUserIds(userIds, distinct, new GroupChannel.GroupChannelCreateHandler() {
        @Override
        public void onResult(GroupChannel groupChannel, SendBirdException e) {
            if (e != null) {
                // Error!
                return;
            }

            Intent intent = new Intent();
            intent.putExtra(EXTRA_NEW_CHANNEL_URL, groupChannel.getUrl());
            setResult(RESULT_OK, intent);
            finish();
        }
    });
}

```

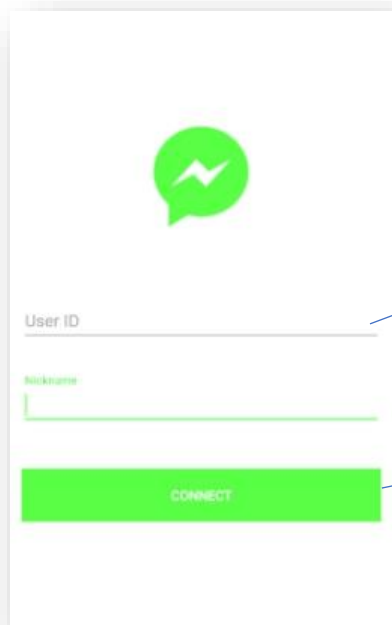
เมื่อผู้ใช้กดปุ่มเฟืองจะเข้าสู่ Setting Activity เป็นหน้าที่ใช้สำหรับตั้งค่าต่าง ๆ ของผู้ใช้ สามารถเปลี่ยน nickname ได้โดยการกด edit จะเรียก method updateCurrentUserInfo เพื่อ update nickname และสามารถเปลี่ยนรูปโปรไฟล์ได้โดยสามารถเลือกได้ว่าจะใช้ใน gallery หรือ camera หากเลือก gallery จะเรียกใช้ method requestMedia แต่หากเรียกใช้ camera จะเรียกใช้ method requestCamera เพื่อเข้าใช้กล้องถ่ายรูปแบบ Intent

```
private void updateCurrentUserInfo(final String userNickname) {  
    final String profileUrl = PreferenceUtils.getProfileUrl();  
    SendBird.updateCurrentUserInfo(userNickname, profileUrl, (e) -> {  
        if (e != null) {  
            // Error!  
            Toast.makeText(context: SettingsActivity.this, text: "" + e.getCode() + ":" + e.get  
            // Show update failed snackbar  
            showSnackbar(text: "Update user info failed");  
            return;  
        }  
        PreferenceUtils.setNickname(userNickname);  
    });  
}
```

```
private void showSetProfileOptionsDialog() {  
    String[] options = new String[] { "Upload a photo", "Take a photo" };  
    AlertDialog.Builder builder = new AlertDialog.Builder(context: SettingsActivity.this);  
    builder.setTitle("Set profile image")  
        .setItems(options, new DialogInterface.OnClickListener() {  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                if (which == 0) {  
                    requestMedia();  
                } else if (which == 1) {  
                    requestCamera();  
                }  
            }  
        })  
        .create().show();  
}
```

```
private void requestCamera() {  
    if (ContextCompat.checkSelfPermission(context: this, Manifest.permission.WRITE_EXTERNAL_STORAGE  
        != PackageManager.PERMISSION_GRANTED) {  
        requestStoragePermissions(PERMISSION_WRITE_EXTERNAL_STORAGE_CAMERA);  
    } else {  
        mRequestingCamera = true;  
        mTempPhotoUri = getTempFileUri(doNotUseFileProvider: false);  
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
        intent.putExtra(MediaStore.EXTRA_OUTPUT, mTempPhotoUri);  
        List<ResolveInfo> resInfoList = getPackageManager().queryIntentActivities(intent, Package  
        for (ResolveInfo resolveInfo : resInfoList) {  
            String packageName = resolveInfo.activityInfo.packageName;  
            grantUriPermission(packageName, mTempPhotoUri, modeFlags: Intent.FLAG_GRANT_WRITE_URI  
        }  
        startActivityForResult(intent, INTENT_REQUEST_CAMERA);  
        SendBird.setAutoBackgroundDetection(false);  
    }  
}
```

## User Interface

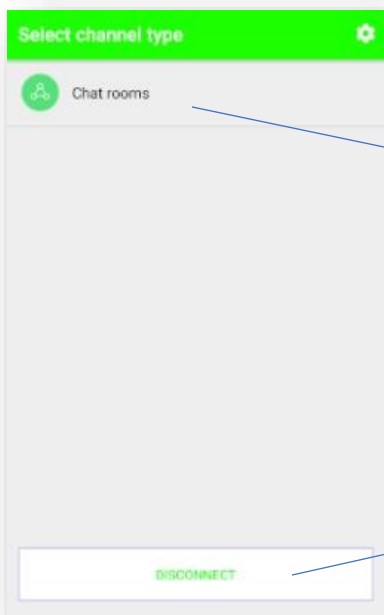


Log - In ใส่ User และ Pass

หากไม่มี Account ระบบจะทำการ Register ให้

ปุ่ม Connect สำหรับ Log - In และ Register

### หน้าแรกของ App

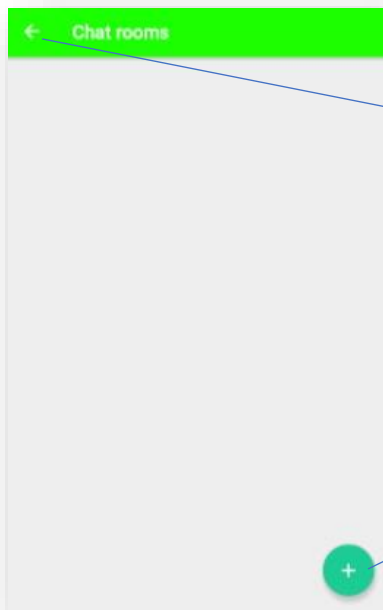


หน้าแก้ไข Profile

สร้างห้องแชต

ออกจากระบบ (Sign Out)

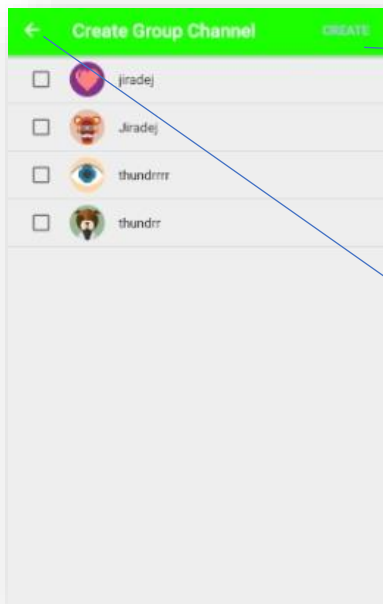
### หน้าหลักของ App



ปุ่มย้อนกลับ

ค้นหารายชื่อเพื่อนที่ต้องการคุย

รายชื่อประวัติห้องแชต  
(ยังไม่เริ่มคุย)



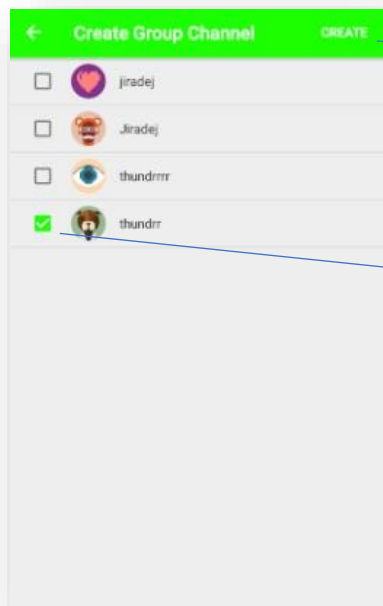
คลิกเพื่อสร้างห้องแชต

รายชื่อเพื่อน(ติกเพื่อนก่อนสร้างห้อง)

ปุ่มย้อนกลับ

รายชื่อเพื่อน

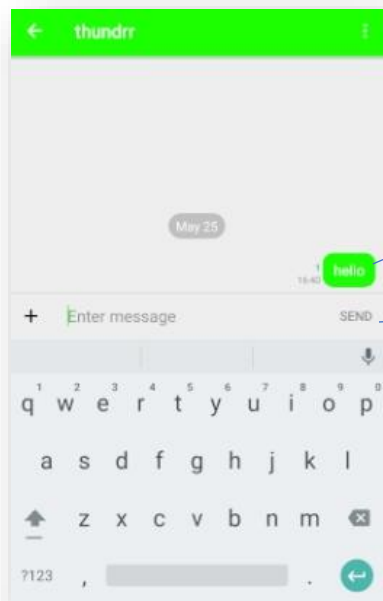




สร้างห้องแชต

เลือกเพื่อนเพื่อคุย

รายชื่อเพื่อน

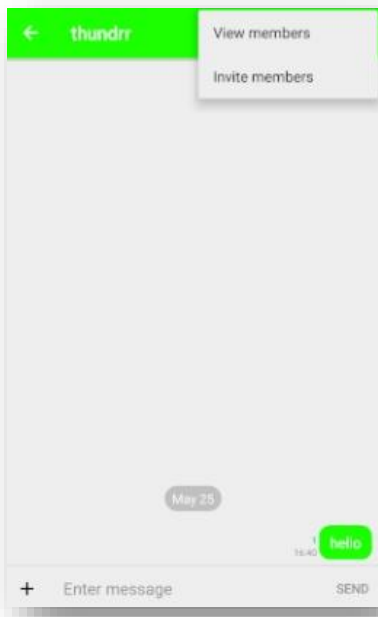


ข้อความที่ส่งไป

ปุ่มข้อความส่งข้อความ

คีย์บอร์ด

ห้องแชต



ห้องแชต

ดูสมาชิกในห้องแชต และปุ่มเชิญเพิ่มเพื่อน



จำนวนคนที่อ่าน

วันที่ที่ส่ง

เวลาที่ส่ง



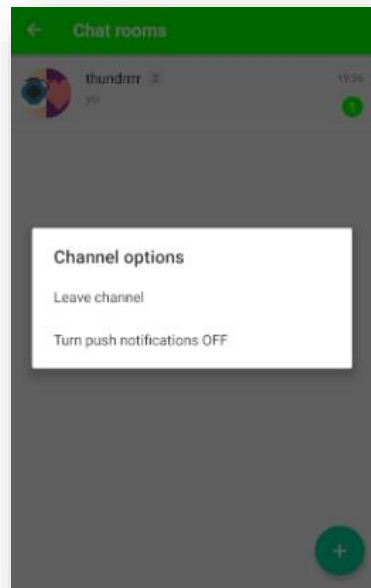
รายชื่อประวัติห้องแชต  
(เมื่อกดย้อนกลับห้องแชต)

ชื่อห้องแชต(ชื่อเพื่อน)



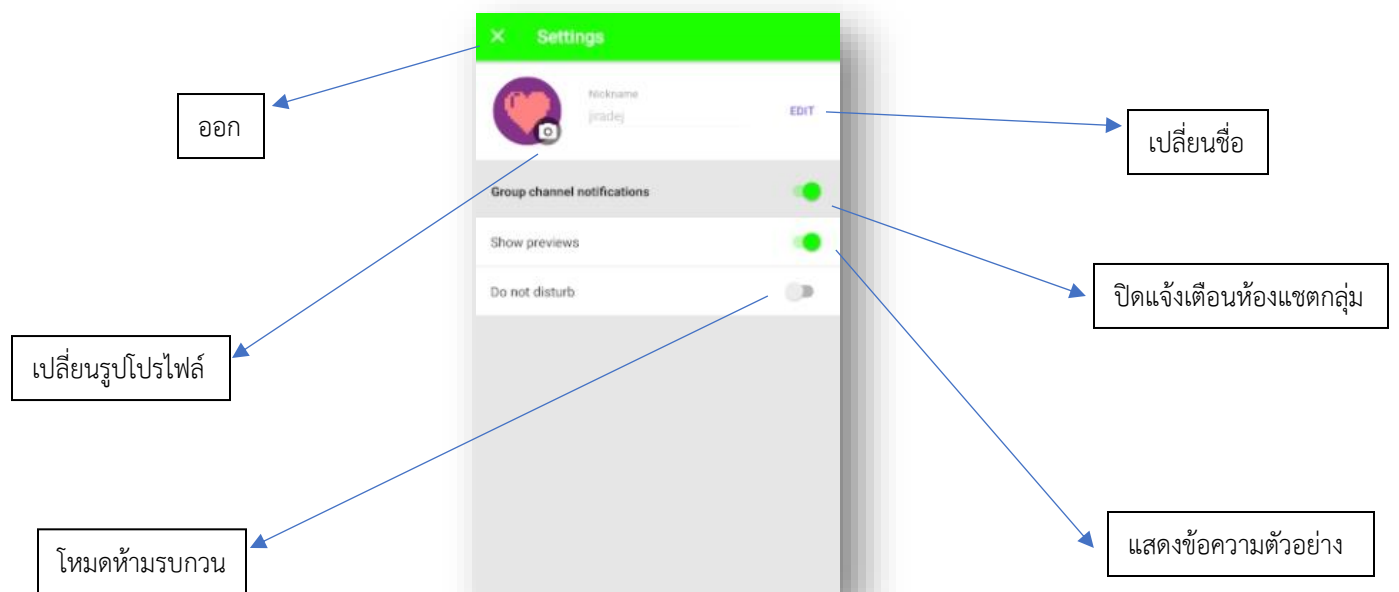
รายชื่อประวัติห้องแชต

แจ้งเตือนเมื่อมีคนส่ง  
ข้อความมา



เมื่อกดแชตค้างไว้จะปรากฏ 2 ตัวเลือก  
ออกจากห้องแชต และ ปิดการแจ้งเตือน

รายชื่อประวัติห้องแชต



หน้า Account Setting





