Activity: SplashActivity

This is an Activity that let our application open smoothly. Users often see a white screen before entering the application without splash screen. (Splash screen will jump out automatically and immediately when users click to open the application icon. Developers can put their basic information they want to show to users in this screen.)



Implement:

We simply used a new activity called splashActivity (Layout: activity\_splash).

Set an integer called SPLASH\_DISPLAY\_LENGTH = 2000, which means the application will automatically jump to the main system after 2 seconds wait.

We imported Intent class and bundle class to control jumping between this activity and the next activity.

Here is the main code:

**private final int SPLASH\_DISPLAY\_LENGHT** = 2000; *// enter the system after 2 seconds*@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_splash***);  
  
 **new** android.os.Handler().postDelayed(**new** Runnable() {  
 **public void** run() {  
 Intent mainIntent = **new** Intent(SplashActivity.**this**,  
 fblogin.**class**);  
 SplashActivity.**this**.startActivity(mainIntent);  
 SplashActivity.**this**.finish();  
 }  
  
 }, **SPLASH\_DISPLAY\_LENGHT**);}

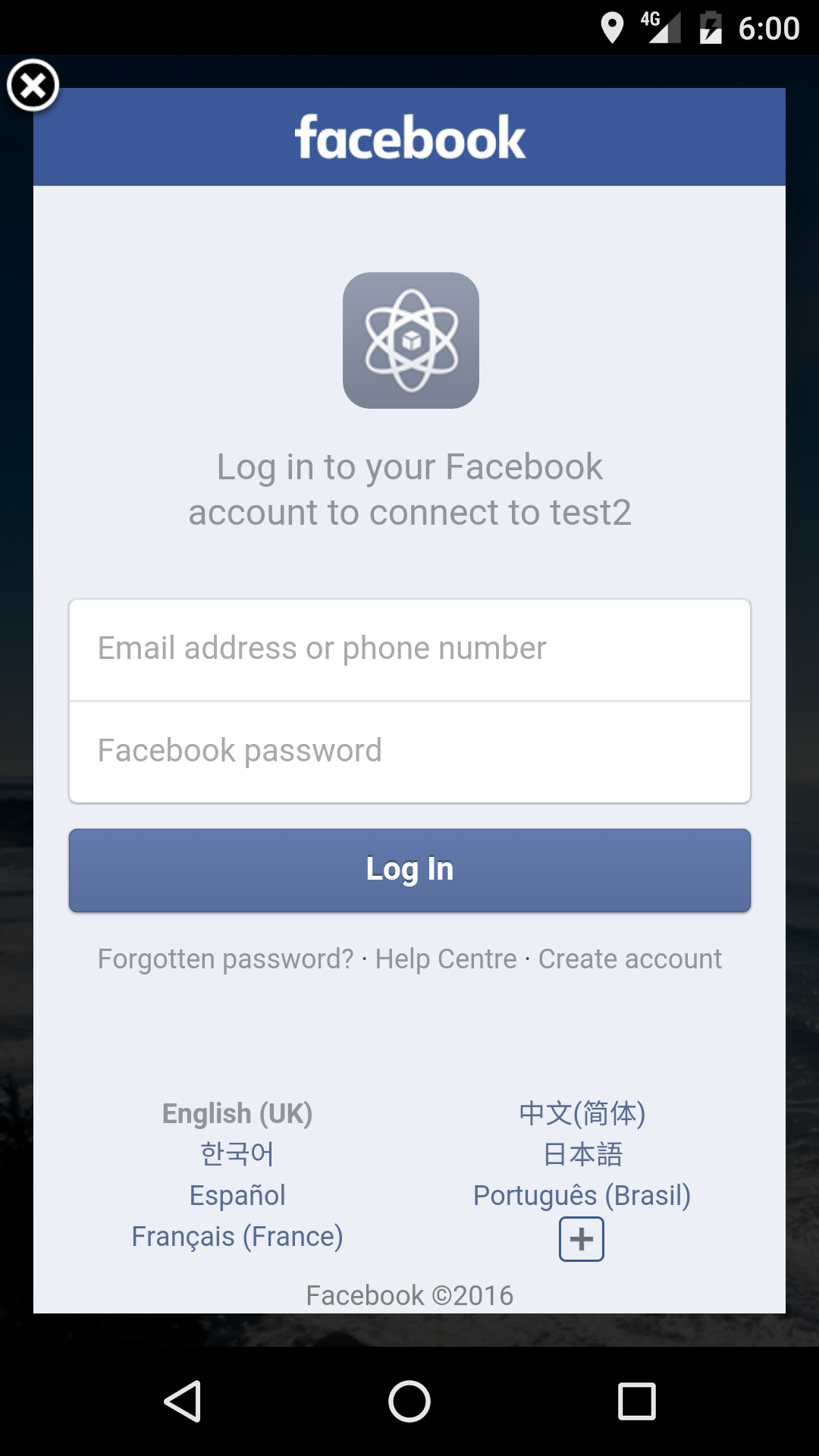
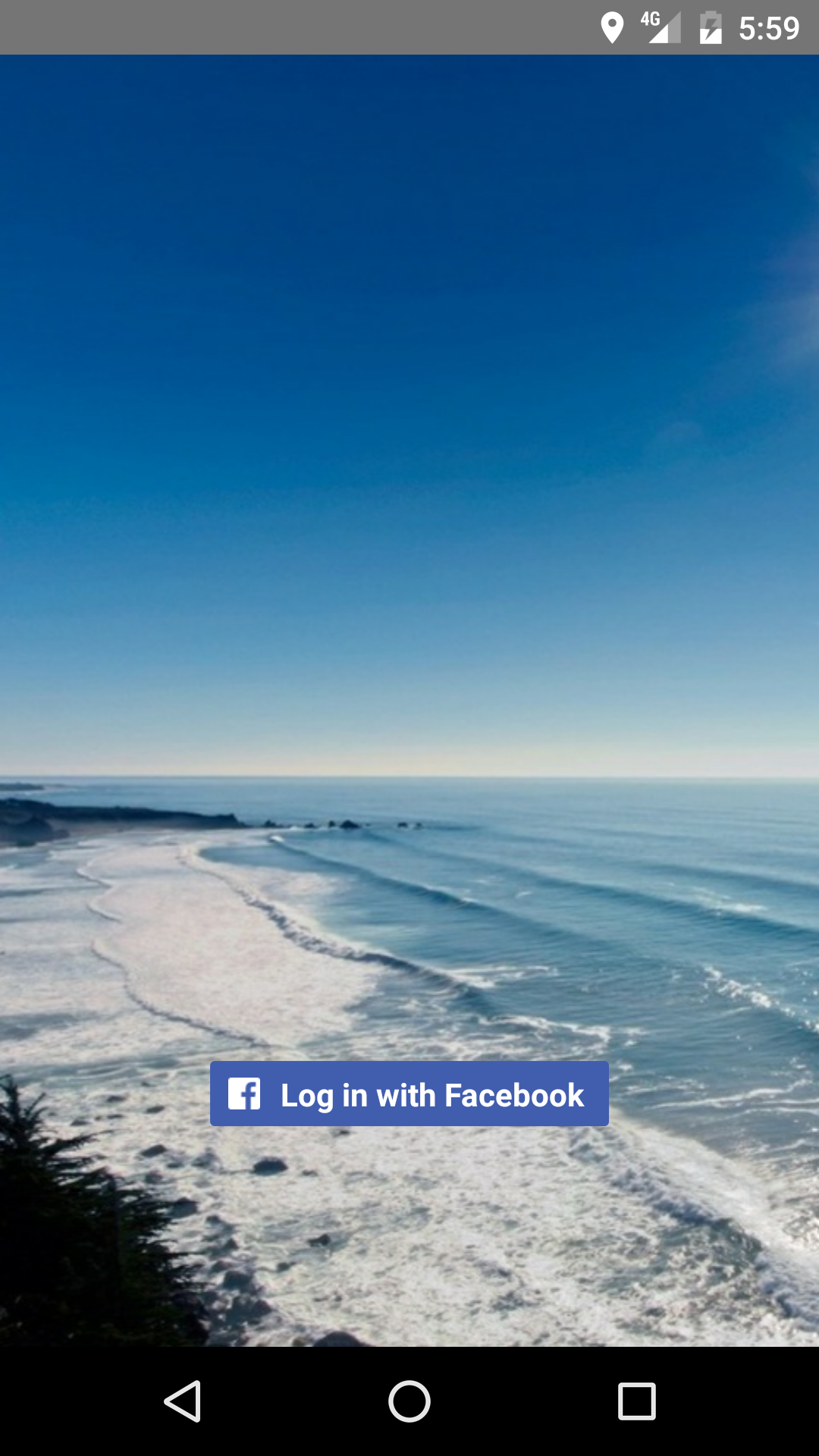
By the way, we settled this activity launch firstly in Manifest

<**activity  
 android:name=".SplashActivity"  
 android:theme="@style/AppSplash"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
</**activity**>

Activity: fblogin

In this activity, we let users to login with their Facebook accounts. All users with Facebook accounts can login our system. And they can share their notes after logging in. The activity will jump to Facebook login page when clicking the Facebook login button.

Furthermore, if the system checked the users are already logged in, users will directly enter the main page instead of this page.



Implement: In this activity, we mainly used 3 methods

1. onCreate()

This method is mainly control the log in button. When this button is clicked, the application will show a page that let users to login in with account names and passwords. And judge what will happen if it success or not. (CallbackManage is provided by Facebook SDK)

**callbackManager** = CallbackManager.Factory.*create*();  
  
**loginButton**.registerCallback(**callbackManager**, **new** FacebookCallback<LoginResult>() {  
 @Override  
 **public void** onSuccess(LoginResult loginResult) {  
 ForwardtoMainactivity();  
 }

@Override  
 **public void** onCancel() {  
 *// App code* }

@Override  
 **public void** onError(FacebookException exception) {  
 *// App code* }  
});

1. ForwardtoActivity()

This method is called when logging successfully and let program jump to Main activity. (We used the Intent class to reuse the parameter).

**public void** ForwardtoMainactivity() {  
 Intent intent = **new** Intent(fblogin.**this**, MainActivity.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** | Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK*** | Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(intent);  
 finish();}

1. isLogIn()

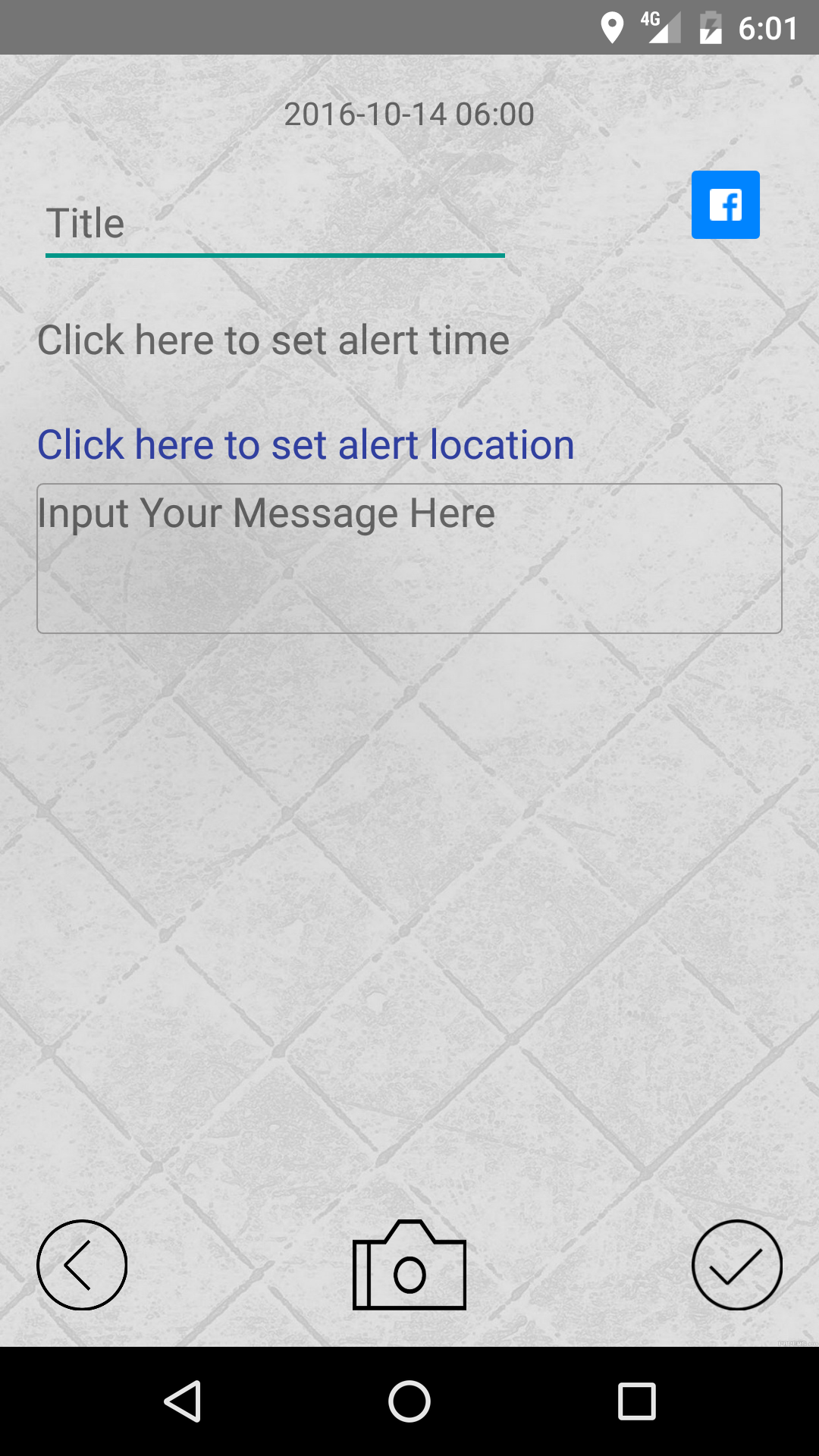
If users have logged in, the program will directly jump to main activity. We used the accessToken class to access the logging information.

**public boolean** isLoggedIn() {  
 AccessToken accessToken = AccessToken.*getCurrentAccessToken*();  
 **return** accessToken != **null**;  
}

Activity: EditActivity

When clicking the add new button on the main activity, users will jump to this activity.

Here is the Edit Layout:



In this Layout, several buttons, textviews, imageview and editviews are settled. And they all have different functions which are important to the whole system.

1. Last edited time textview



On the top of the layout, the textview is settled to show the last edited time of the note. It will also be shown on the main activity.

Implement: We used the Calendar class to get the system time and SimpleDateFormat class to format the string, then used Intent (Flag) class to show in the textview.

Intent intent = getIntent();  
**realEditingStatusFlag** = intent.getStringExtra(**editingStatusFlag**);

**if**(**addingNewNote**.equals(**realEditingStatusFlag**)){  
 *//Use calendar to get the current system time* SimpleDateFormat f = **new** SimpleDateFormat(**"yyyy-MM-dd hh:mm"**);  
 **timeOfCreation** = Calendar.*getInstance*();  
 **time** = f.format(**timeOfCreation**.getTime());  
 *//Change the text in the textView to the current system time* **timeTextView**.setText(**time**);

Every time the note is edited, the time will change and save automatically.

1. Title editview

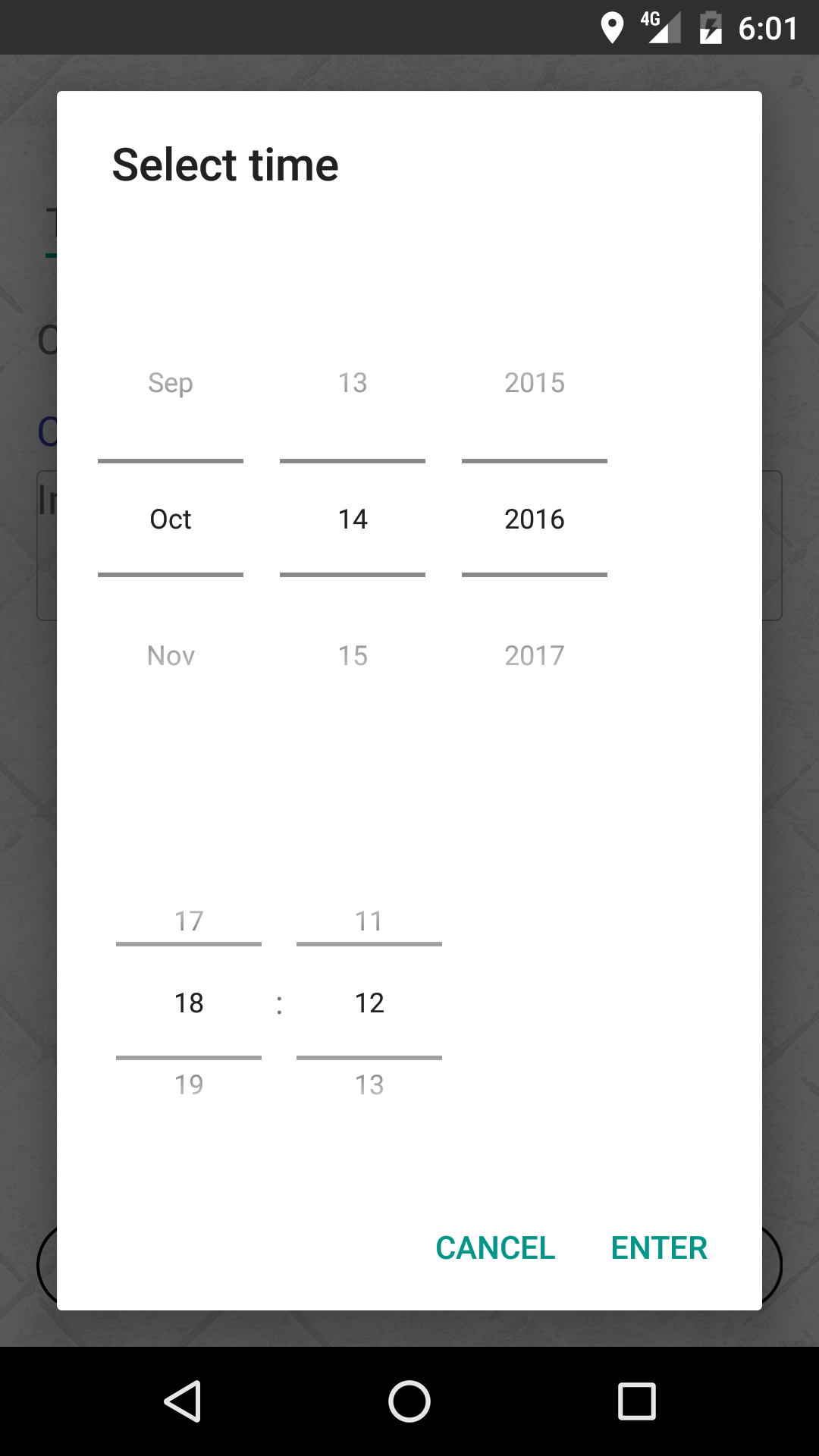


This is the editview to enter the title of the note and will be shown on the main screen.

1. Set Alert time



This is a textview to show and set the alert time. It will jump out a dialog to let people choose what time should be alerted, then save or cancel it. Then the settled time will be shown on the textview and main activity.



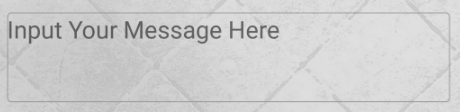
Implement: We used the onTouch() method via View class and AlertDialog class to make the dialog then used the Calendar get time and date from system and set time and date in timepicker and datepicker.

**public boolean** onTouch(View v, MotionEvent event) {  
 **if** (event.getAction() == MotionEvent.***ACTION\_DOWN***) {  
  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 View view = View.*inflate*(**this**, R.layout.***date\_time\_dialog***, **null**);  
 **final** DatePicker datePicker = (DatePicker) view.findViewById(R.id.***date\_picker***);  
 **final** TimePicker timePicker = (android.widget.TimePicker) view.findViewById(R.id.***time\_picker***);  
 builder.setView(view);  
  
 Calendar cal = Calendar.*getInstance*();  
 cal.setTimeInMillis(System.*currentTimeMillis*());  
 datePicker.init(cal.get(Calendar.***YEAR***), cal.get(Calendar.***MONTH***), cal.get(Calendar.***DAY\_OF\_MONTH***), **null**);  
  
 timePicker.setIs24HourView(**true**);  
 **if** (Build.VERSION.***SDK\_INT*** >= 23 ){  
 timePicker.setHour(cal.get(Calendar.***HOUR\_OF\_DAY***));  
 timePicker.setMinute(Calendar.***MINUTE***);  
 }  
 **else**{  
 timePicker.setCurrentHour(cal.get(Calendar.***HOUR\_OF\_DAY***));  
 timePicker.setCurrentMinute(Calendar.***MINUTE***);  
 }  
  
  
 **if** (v.getId() == R.id.***et\_set\_time***) {  
 **final int** inType = **etSetTime**.getInputType();  
 **etSetTime**.setInputType(InputType.***TYPE\_NULL***);  
 **etSetTime**.onTouchEvent(event);  
 **etSetTime**.setInputType(inType);  
 *//etSetTime.setSelection(etSetTime.getText().length());* builder.setTitle(**"Select time"**);  
 builder.setPositiveButton(**"ENTER"**, **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
 StringBuffer sb = **new** StringBuffer();  
 sb.append(String.*format*(**"%d-%02d-%02d"**,  
 datePicker.getYear(),  
 datePicker.getMonth() + 1,  
 datePicker.getDayOfMonth()));  
 sb.append(**" "**);  
 **if** (Build.VERSION.***SDK\_INT*** >= 23 ){  
 sb.append(timePicker.getHour())  
 .append(**":"**).append(timePicker.getMinute());  
 }  
 **else**{  
 sb.append(timePicker.getCurrentHour())  
 .append(**":"**).append(timePicker.getCurrentMinute());  
 }  
 String a = sb.toString();  
  
 **alertTime** = a;  
 **etSetTime**.setText(a);  
  
  
 dialog.cancel();  
 }  
 }).setNegativeButton(**"CANCEL"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.cancel();*//nothing happens* }  
 });  
  
 }  
  
  
 Dialog dialog = builder.create();  
 dialog.show();  
 }  
  
 **return true**;  
}

1. Set location textView



1. Message Editview



This editview is used to input the message, no words limited and return button is settled to change lines.

Implement: the method it used is same with the title editview.

1. Photo button

When clicking the photo button, two secondary buttons will come out. And we can choose take photo and choose photos from gallery. Then the photo will be shown in the imageview.

Implement:

1. Show(View view)

This method is used to show the secondary buttons which used the dialogclass.

**public void** show(View view){  
 **dialog** = **new** Dialog(**this**,R.style.***ActionSheetDialogStyle***);  
 *//填充对话框的布局* **inflate** = LayoutInflater.*from*(**this**).inflate(R.layout.***dialog\_layout***, **null**);  
 *//初始化控件* **choosePhoto** = (TextView) **inflate**.findViewById(R.id.***choosePhoto***);  
 **takePhoto** = (TextView) **inflate**.findViewById(R.id.***takePhoto***);  
 **choosePhoto**.setOnClickListener(**this**);  
 **takePhoto**.setOnClickListener(**this**);  
 *//将布局设置给Dialog* **dialog**.setContentView(**inflate**);  
 *//获取当前Activity所在的窗体* Window dialogWindow = **dialog**.getWindow();  
 *//设置Dialog从窗体底部弹出* dialogWindow.setGravity( Gravity.***BOTTOM***);  
 *//获得窗体的属性* WindowManager.LayoutParams lp = dialogWindow.getAttributes();  
 lp.**y** = 20;*//设置Dialog距离底部的距离  
// 将属性设置给窗体* dialogWindow.setAttributes(lp);  
 **dialog**.show();*//显示对话框* }

1. Take photo button(onTakePhotoClick)

Used the intent class to open the camera and transfer the photo to a uri to save and load.

**public void** onTakePhotoClick() {  
 *// create Intent to take a picture and return control to the calling application* Intent intent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  
 String photoDir = **photoFileName** + **time** + **".jpg"**;  
 intent.putExtra(MediaStore.***EXTRA\_OUTPUT***, getFileUri(photoDir)); *// set file name  
  
 // Start the image capture intent to take photo* startActivityForResult(intent, ***CAPTURE\_IMAGE\_ACTIVITY\_REQUEST\_CODE***);  
}

1. Choose from gallery(onLoadPhotoClick)

Used the intent class to get the uri of the photo file from system gallery then load and save it.

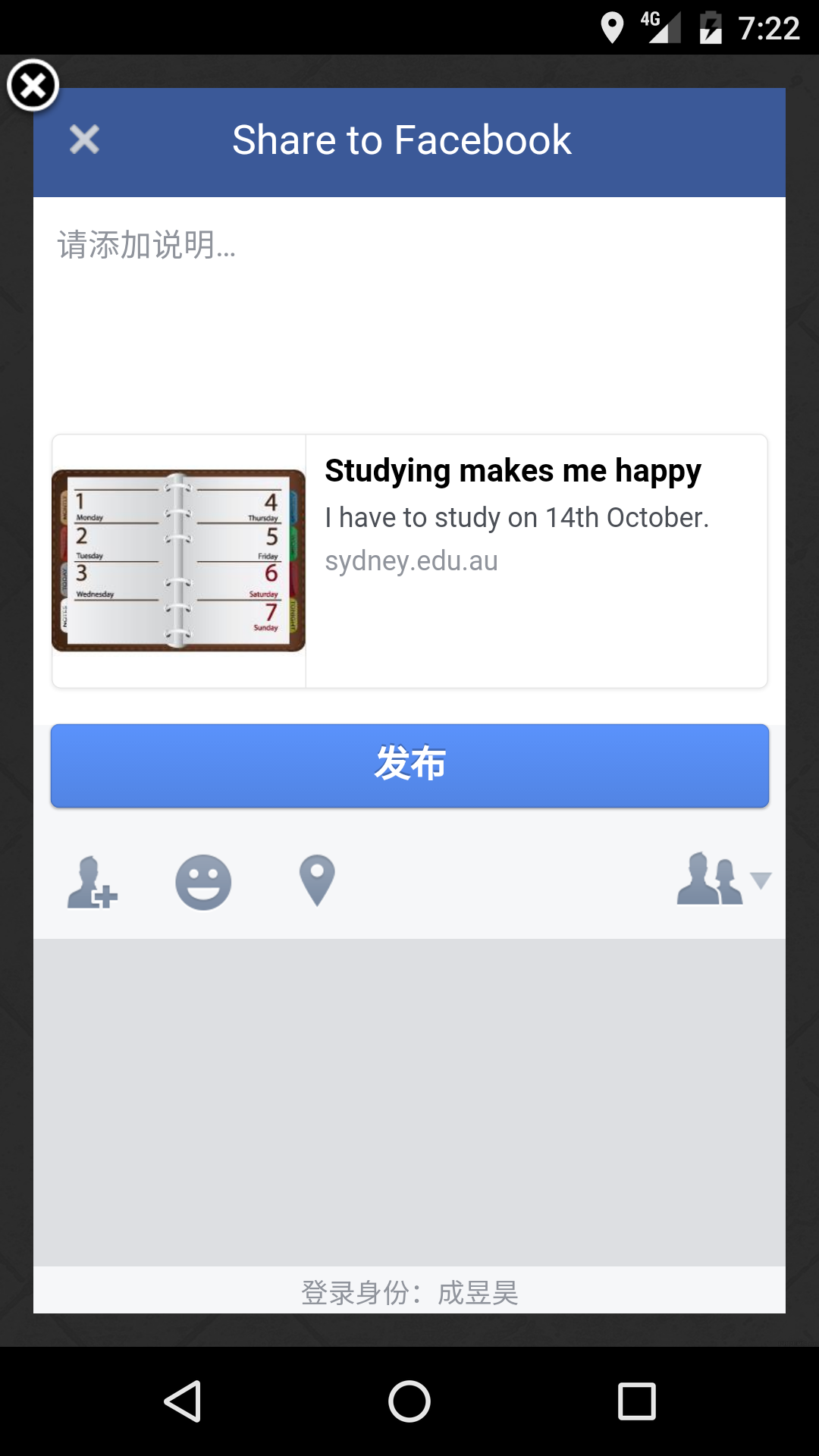
**public void** onLoadPhotoClick() {  
 *// Create intent for picking a photo from the gallery* Intent intent = **new** Intent(Intent.***ACTION\_PICK***, MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***);  
 *// Bring up gallery to select a photo* startActivityForResult(intent, ***PICK\_PHOTO\_CODE***);  
}

1. Finish button

When clicking this button, the note will save to the database and show on the main screen.

Implement:Intent

1. Share button

When clicking this button, the basic information will be share to Facebook (title, time and message)

Implement:

**fbShareButton** = (ShareButton) findViewById(R.id.***button\_share***);

String fbShareTitle = **titleEditText**.getText().toString();  
String fbShareDescription = **messageEditText**.getText().toString();  
**if**(fbShareTitle.isEmpty()){  
 fbShareTitle = **"No title"**;  
}  
**if**(fbShareDescription.isEmpty()){  
 fbShareDescription = **"No message"**;  
}  
Uri myUri = Uri.*parse*(**"http://www.free-icons-download.net/images/notebook-icon-44654.png"**);  
  
ShareLinkContent content = **new** ShareLinkContent.Builder()  
 .setContentTitle(fbShareTitle)  
 .setContentDescription(fbShareDescription)  
 .setImageUrl(myUri)  
 .setContentUrl(Uri.*parse*(**"http://sydney.edu.au/"**))  
 .build();  
  
**fbShareButton**.setShareContent(content);

Finish screen shoot:

