

# **HW2:**

## **Contact Info Application**

Martin Mudenda Bbela  
2582912

## Intro

This application will be able to switch between two activities. On opening it the user should be prompted to add contact information. Will switch to the 2<sup>nd</sup> activity to add their details. At this point the user can confirm the information which will bring them back to the first activity. This time the activity should have all the information selected in the previous page.

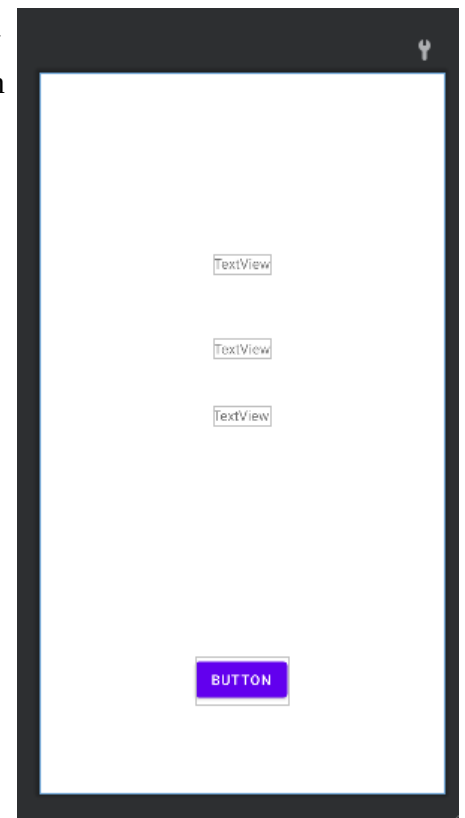
## Method:

### Main Activity

The main activity will have a layout containing three text boxes. They will be empty initially. These will be wired to update on returning from the 2<sup>nd</sup> screen with information housed in an intent object. Note that for elegance, a Constants class is created that will only house strings that can be used to get and set extras in the Intent. The button is wired to start the next activity.

```
btnCreateContact = findViewById(R.id.buttonCreateContact);
tvName           = findViewById(R.id.textViewName);
tvPhone          = findViewById(R.id.textViewPhone);
tvEmail          = findViewById(R.id.textViewEmail);

mGetContent = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {
    @Override
    public void onActivityResult(ActivityResult result) {
        Toast.makeText(getApplicationContext(), "returned", Toast.LENGTH_LONG).show();
        tvName.setText(result.getData().getStringExtra(ContactConstants.CONTACT_NAME));
        tvPhone.setText(result.getData().getStringExtra(ContactConstants.CONTACT_PHONE));
        tvEmail.setText(result.getData().getStringExtra(ContactConstants.CONTACT_EMAIL));
    }
});
```



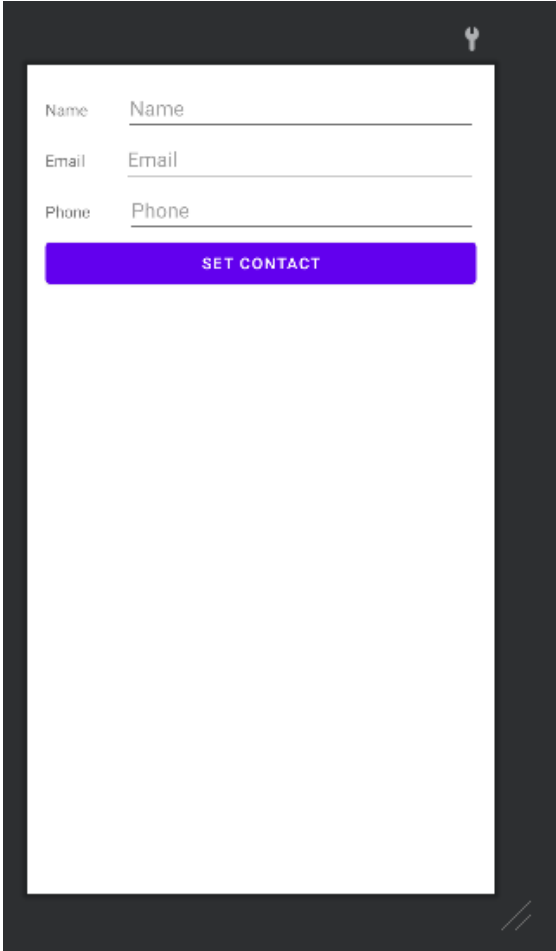
```
btnCreateContact.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent startCreateContentIntent = new Intent(getApplicationContext(), CreateContactActivity.class);
        mGetContent.launch(startCreateContentIntent);
    }
});
```

## Activity 2

Activity will have 3 Edit Text Boxes corresponding to the contact information they user will have to put in. It also contains a button that will close this activity and return to the first activity with the text as written in the boxes.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_create_contact);
    btnFinish = findViewById(R.id.buttonFinish);
    etPersonName = findViewById(R.id.editTextTextPersonName);
    etEmailAddress = findViewById(R.id.editTextTextEmailAddress);
    etPhone = findViewById(R.id.editTextTextPhone);

    btnFinish.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent returnContact = new Intent();
            returnContact.putExtra(ContactConstants.CONTACT_NAME, etPersonName.getText().toString());
            returnContact.putExtra(ContactConstants.CONTACT_EMAIL, etEmailAddress.getText().toString());
            returnContact.putExtra(ContactConstants.CONTACT_PHONE, etPhone.getText().toString());
            setResult(RESULT_OK, returnContact);
            finish();
        }
    });
}
```



The screenshot shows a mobile application interface for creating a contact. It features three input fields labeled "Name", "Email", and "Phone", each with a corresponding text entry box. Below these fields is a red button labeled "SET CONTACT". The interface is set against a light blue background with a subtle grid pattern. A dark blue header bar at the top contains a white back arrow icon, and a dark blue footer bar at the bottom contains a white double-slash icon.

# Result:

The application is able to prompt the user to fill in the Edit text boxes and return the user to the main activity with the Contact information

