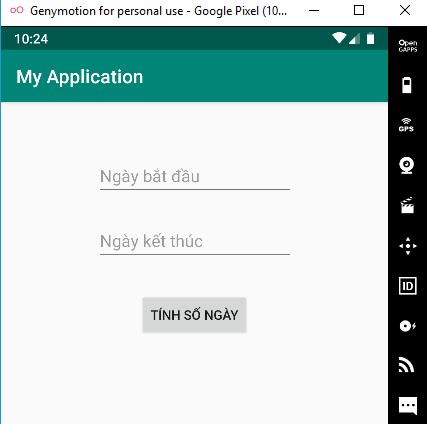
**Bài thực hành số 5**

**Bài 5.1.** Xây dựng màn hình cho phép nhập ngày bắt đầu và ngày kết thúc. Với các yêu cầu:

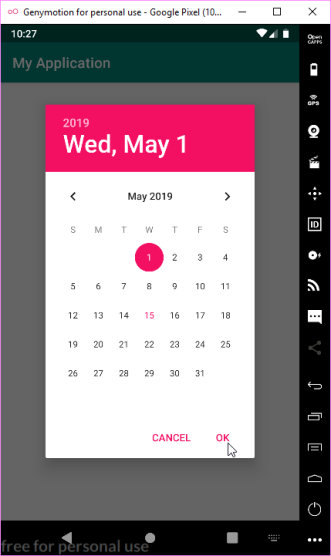
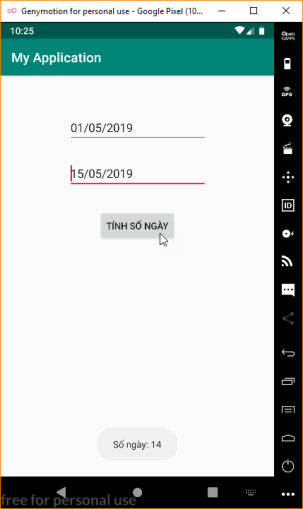
a) Thiết kế layout như:



b) Chức năng:

1) Khi người dùng nhấn vào các EditText ngày bắt đầu, ngày kết thúc thì hiển thị hộp thoại DatePickerDialog.

2) Khi nhấn vào nút “TÍNH SỐ NGÀY”, hiển thị thông báo Số ngày = Ngày kết thúc – Ngày bắt đầu.

Hướng dẫn:

**public class** DemoDateTime **extends** AppCompatActivity {  
  
 EditText **edtNgayBD**, **edtNgayKT**;  
 Button **btnTinhNgay**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_demo\_date\_time***);

*// Ánh xạ*  
 **edtNgayBD** = findViewById(R.id.***editTextNgayBD***);  
 **edtNgayKT** = findViewById(R.id.***editTextNgayKT***);  
 **btnTinhNgay** = findViewById(R.id.***buttonTinhNgay***);  
  
  
 *// Bị duplicate code, sinh viên tự cấu trúc lại!* **edtNgayBD**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 chonNgayBatDau();  
 }  
 });  
  
 **edtNgayKT**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 chonNgayKetThuc();  
 }  
 });  
  
 **btnTinhNgay**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **SimpleDateFormat** sdf = **new** SimpleDateFormat(**"dd/MM/yyyy"**);  
 **try** {  
 Date d1 = sdf.parse(**edtNgayBD**.getText().toString());  
 Date d2 = sdf.parse(**edtNgayKT**.getText().toString());  
  
 **long** miliGiay = d2.getTime() - d1.getTime();  
  
 **long** soNgay = miliGiay / (1000 \* 60 \* 60 \* 24);  
  
 Toast.*makeText*(DemoDateTime.**this**,  
 **"Số ngày: "** + String.*valueOf*(soNgay),  
 Toast.***LENGTH\_SHORT***).show();  
  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
  
 }  
 });  
 }  
  
 **void** chonNgayBatDau() {  
 **final** Calendar calendar = Calendar.*getInstance*();  
 **int** ngay = calendar.get(Calendar.***DATE***);  
 **int** thang = calendar.get(Calendar.***MONTH***);  
 **int** nam = calendar.get(Calendar.***YEAR***);  
  
 DatePickerDialog dp = **new** DatePickerDialog(**this**,  
 **new** DatePickerDialog.OnDateSetListener() {  
 @Override  
 **public void** onDateSet(DatePicker view, **int** year, **int** month, **int** dayOfMonth) {  
 calendar.set(year, month, dayOfMonth);  
  
 SimpleDateFormat sdf =  
 **new** SimpleDateFormat(**"dd/MM/yyyy"**);  
  
 **edtNgayBD**.setText(sdf.format(calendar.getTime()));  
 }  
 }, nam, thang, ngay);  
  
 dp.show();  
 }  
  
 **void** chonNgayKetThuc() {  
 **final** Calendar calendar = Calendar.*getInstance*();  
 **int** ngay = calendar.get(Calendar.***DATE***);  
 **int** thang = calendar.get(Calendar.***MONTH***);  
 **int** nam = calendar.get(Calendar.***YEAR***);  
  
 DatePickerDialog dp = **new** DatePickerDialog(**this**,  
 **new** DatePickerDialog.OnDateSetListener() {  
 @Override  
 **public void** onDateSet(DatePicker view, **int** year, **int** month, **int** dayOfMonth) {  
 calendar.set(year, month, dayOfMonth);  
  
 SimpleDateFormat sdf =  
 **new** SimpleDateFormat(**"dd/MM/yyyy"**);  
  
 **edtNgayKT**.setText(sdf.format(calendar.getTime()));  
 }  
 }, nam, thang, ngay);  
  
 dp.show();  
 }  
}

**Bài 5.2.** Tạo ứng dụng sổ liên lạc với các yêu cầu sau:

a) Thiết kế layout gồm:

* Các view: Họ tên, số điện thoại (10 chữ số), địa chỉ Email.
* Nút lệnh: Xóa trống, Lưu, Cập nhật, Tìm kiếm.
* Một ListView để hiển thị danh sách liên lạc. Mỗi dòng hiển thị theo định dạng Họ tên - số điện thoại (không cần hiển thị địa chỉ Email).

b) Chức năng:

* Nút lệnh Xóa trống có chức năng xóa dữ liệu hiện có trên các view nhập dữ liệu và chuyển tiêu điểm đến view Họ tên.
* Nút lệnh Lưu có chức năng kiểm tra thông tin đã được nhập đầy đủ và đúng quy định trên các view chưa, rồi tiến hành lưu dữ liệu và hiển thị lên ListView.
* Với Nút cập nhật, khi chọn một mục trên ListView thì thông tin mục được chọn sẽ hiển thị lên các view thành phần, tiến hành sửa nội dung, rồi khi click vào nút Cập nhật thì nội dùng đã được chỉnh sửa sẽ cập nhật lại trong danh sách lưu trữ và hiển thị lên Listview.
* Nút tìm kiếm: tìm kiếm thông tin liên lạc trong danh sách theo họ tên và hiển thị kết quả tìm kiếm lên listview.

c) Hướng dẫn:

* Tạo layout gồm các thành phần giao diện như mô tả.
* Tạo một class mô tả thông tin đối tượng cần lưu.
* Tạo một đối tượng tập hợp (danh sách) để lưu thông tin mỗi lần thêm một liên lạc
* Tạo một adapter để chuyển tiếp dữ liệu từ danh sách lên ListView.
* Đưa dữ liệu lên ListView thông qua phương thức setAdapter.