

# ST LAB 9

**SAHIL BONDRE: U18CO021**

**Question:** Extend your lab assignment 8 by inserting a new table which stores messages related to the persons available in your contact list. This is to save any important information mentioned during the recent or an earlier phone call. Your new table should be named as “Message” with message\_id as the primary key and contact\_no (or any id which you are using as your primary key for the contact table) as your foreign key here. You will also store date and time stamp to identify the recent message stored for a particular person. A list of operations which are required to provide to the user are as follows.

- (a) Display all messages related to a person.
- (b) Display recent message of a person.
- (c) Add a new message.
- (d) Delete an existing message.
- (e) Update an existing message.

**Submission:** A single (doc/pdf) file which will contain your source code and the screenshots of generated outputs. Name of your file should be assignment-number\_yourid.

## Directory Structure

```
.
├── AppDatabase.java
├── models
│   ├── Contact.java
│   ├── ContactWithMessages.java
│   └── Message.java
├── services
│   ├── ContactDao.java
│   └── MessageDao.java
├── util
│   └── DateConverter.java
└── views
    ├── AddActivity.java
    ├── MainActivity.java
    └── ViewContactActivity.java
```

## Code

### Models

#### Contact.java

```
package com.example.contactsmessage1ab9.models;

import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

import java.io.Serializable;

@Entity
public class Contact implements Serializable {
    @PrimaryKey
    public long uid;
    @ColumnInfo(name = "name")
    public String name;
    @ColumnInfo(name = "email")
    public String email;
    @ColumnInfo(name = "mobile")
    public String mobile;
    @ColumnInfo(name = "address")
```

```

        public String address;

        public Contact(long uid, String name, String email, String mobile,
String address) {
            this.uid = uid;
            this.name = name;
            this.email = email;
            this.mobile = mobile;
            this.address = address;
        }
    }
}

```

## Message.java

```

package com.example.contactsmessageLab9.models;

import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;
import androidx.room.TypeConverters;

import com.example.contactsmessageLab9.util.DateConverter;

import java.util.Date;

@Entity
@TypeConverters(DateConverter.class)
public class Message {
    @PrimaryKey
    public long messageId;
    @ColumnInfo(name = "user_id")
    public long userId;
    @ColumnInfo(name = "message")
    public String message;
    @ColumnInfo(name = "date")
    public Date date;

    public Message(long messageId, long userId, String message, Date
date) {
        this.messageId = messageId;
        this.userId = userId;
        this.message = message;
        this.date = date;
    }
}

```

```
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }
}
```

### ContactWithMessages.java

```
package com.example.contactsmessageLab9.models;

import androidx.room.Embedded;
import androidx.room.Relation;

import java.util.List;

public class ContactWithMessages {
    @Embedded
    public Contact contact;
    @Relation(
        parentColumn = "uid",
        entityColumn = "user_id"
    )
    public List<Message> messages;
}
```

## Services

### ContactDao.java

```
package com.example.contactsmessageapp9.services;

import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.Query;
import androidx.room.Transaction;
import androidx.room.Update;

import com.example.contactsmessageapp9.models.Contact;
import com.example.contactsmessageapp9.models.ContactWithMessages;

import java.util.List;

@Dao
public interface ContactDao {
    @Query("SELECT * FROM contact")
    List<Contact> getAll();

    @Query("SELECT * FROM contact WHERE uid IN (:contactIds)")
    List<Contact> loadAllByIds(int[] contactIds);

    @Query("SELECT * FROM contact WHERE name LIKE :name LIMIT 1")
    Contact findByName(String name);

    @Insert
    void insertAll(Contact... contacts);

    @Delete
    void delete(Contact contact);

    @Update
    void update(Contact contact);

    @Transaction
    @Query("SELECT * FROM contact")
    List<ContactWithMessages> contactsWithMessages();

    @Transaction
    @Query("SELECT * FROM contact WHERE uid = :uid")
    ContactWithMessages contactsWithMessagesOfId(long uid);
}
```

## MessageDao.java

```
package com.example.contactsmessagelab9.services;

import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.OnConflictStrategy;
import androidx.room.Query;
import androidx.room.Update;

import com.example.contactsmessagelab9.models.Message;

import java.util.List;

@Dao
public interface MessageDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void insertMessages(Message... messages);

    @Update
    void updateMessages(Message... messages);

    @Delete
    void deleteMessage(Message messages);

    @Query("SELECT * FROM Message")
    List<Message> getAll();
}
```

## Activities

### AddActivity.java

```
package com.example.contactsmessageLab9.views;

import android.os.Bundle;
import android.telephony.PhoneNumberUtils;
import android.widget.Button;
import android.widget.EditText;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.room.Room;

import com.example.contactsmessageLab9.AppDatabase;
import com.example.contactsmessageLab9.R;
import com.example.contactsmessageLab9.models.Contact;
import com.example.contactsmessageLab9.services.ContactDao;

public class AddActivity extends AppCompatActivity {

    private EditText emailEditText;
    private EditText nameEditText;
    private EditText phoneEditText;
    private EditText addressEditText;
    private Button saveButton;
    private String name;
    private String phone;
    private String address;
    private String email;
    private AppDatabase db;
    private ContactDao contactDao;

    private boolean isValidEmail(CharSequence target) {
        if (target == null)
            return false;

        return
            android.util.Patterns.EMAIL_ADDRESS.matcher(target).matches();
    }

    private void showErrorDialog(String message) {
        new AlertDialog.Builder(this)
```

```

        .setIcon(android.R.drawable.stat_notify_error)
        .setTitle("Error")
        .setMessage(message)
        .setNegativeButton("OK", (dialogInterface, i) -> {
            //set what would happen when positive button is
clicked
        })
        .show();
    }

    private void showAddDialogue() {
        new AlertDialog.Builder(this)
            .setIcon(android.R.drawable.ic_menu_save)
            .setTitle("Save")
            .setMessage("Are you sure you want to add new contact?")
            .setPositiveButton("Yes", (dialogInterface, i) -> {
                new Thread(() -> {
                    contactDao.insertAll(new
Contact(System.currentTimeMillis(), name, email, phone, address));
                    runOnUiThread(() -> {
                        setResult(1);
                        finish();
                    });
                }).start();
                //set what would happen when positive button is
clicked
            }).setNegativeButton("No", (dialogInterface, i) -> {

            }).show();
    }

    private void initialiseUI() {
        emailEditText = findViewById(R.id.email_edit_text);
        nameEditText = findViewById(R.id.name_edit_text);
        phoneEditText = findViewById(R.id.phone_edit_text);
        addressEditText = findViewById(R.id.address_edit_text);
        saveButton = findViewById(R.id.save_button);

        saveButton.setOnClickListener(v -> {
            name = nameEditText.getText().toString();
            email = emailEditText.getText().toString();
            phone = phoneEditText.getText().toString();
            address = addressEditText.getText().toString();

            if (name.length() == 0) {
                showErrorDialog("Name cannot be empty");
            }
        });
    }

```



```

        return;
    }

    if (email.length() != 0 && !isValidEmail(email)) {
        showErrorDialog("Invalid email");
        return;
    }

    if (!PhoneNumberUtils.isGlobalPhoneNumber(phone)) {
        showErrorDialog("Invalid Phone");
        return;
    }

    showAddDialogue();
    });
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add);
    db = Room.databaseBuilder(this, AppDatabase.class,
"contacts").build();
    contactDao = db.contactDao();
    initialiseUI();
}
}

```

## MainActivity.java

```

package com.example.contactsmessageLab9.views;

import android.content.Intent;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.room.Room;

```

```

import com.example.contactsmessageLab9.AppDatabase;
import com.example.contactsmessageLab9.R;
import com.example.contactsmessageLab9.models.Contact;
import com.example.contactsmessageLab9.services.ContactDao;
import com.example.contactsmessageLab9.services.MessageDao;
import
com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.List;
import java.util.Vector;

public class MainActivity extends AppCompatActivity {

    private List<String> contactNames;
    private List<Contact> contacts;
    private ListView contactListView;
    private Button searchButton;
    private EditText searchEditText;
    private FloatingActionButton fab;
    private ArrayAdapter<String> arrayAdapter;
    private AppDatabase db;
    private ContactDao contactDao;
    private MessageDao messageDao;

    private void updateContactList() {
        arrayAdapter = new ArrayAdapter<>(this,
R.layout.activity_listview, R.id.contact_text_view, contactNames);
        contactListView.setAdapter(arrayAdapter);
    }

    private void initialiseUI() {
        contactListView = findViewById(R.id.contacts_list_view);
        searchButton = findViewById(R.id.search_button);
        searchEditText = findViewById(R.id.search_edit_text);
        fab = findViewById(R.id.fab);

        contactListView.setOnItemClickListener((parent, view, position,
id) -> {
            Intent intent = new Intent(this, ViewContactActivity.class);
            intent.putExtra("contact", contacts.get(position));
            startActivityForResult(intent, 2);
        });

        fab.setOnClickListener(v -> {
            Intent intent = new Intent(this, AddActivity.class);

```

```

        startActivityForResult(intent, 1);
    });

    searchButton.setOnClickListener(v -> {
        new Thread(() -> {
            Contact contact =
contactDao.findByName(searchEditText.getText().toString());
            runOnUiThread(() -> {

                if (contact != null) {
                    Intent intent = new Intent(this,
ViewContactActivity.class);
                    intent.putExtra("contact", contact);
                    startActivityForResult(intent, 2);
                } else {
                    new AlertDialog.Builder(this)

.setIcon(android.R.drawable.stat_sys_warning)
                    .setTitle("404")
                    .setMessage("Contact not found!")
                    .setNegativeButton("OK",

(dialogInterface, i) -> {

                        }).show();
                    }
                });
            }).start();
        });
    }

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    db = Room.databaseBuilder(this, AppDatabase.class,
"contacts").build();
    contactDao = db.contactDao();
    messageDao = db.messageDao();
    initialiseUI();
    new Thread(this::initialiseContacts).start();
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

```

```

        new Thread(this::initialiseContacts).start();
    }

    private void initialiseContacts() {
        contacts = contactDao.getAll();
        contactNames = new Vector<>(contacts.size());
        for (Contact contact : contacts) {
            contactNames.add(contact.name);
        }
        runOnUiThread(this::updateContactList);
    }
}

```

## ViewContactActivity.java

```

package com.example.contactsmessagelab9.views;

import android.os.Bundle;
import android.telephony.PhoneNumberUtils;
import android.text.InputType;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.room.Room;

import com.example.contactsmessagelab9.AppDatabase;
import com.example.contactsmessagelab9.R;
import com.example.contactsmessagelab9.models.Contact;
import com.example.contactsmessagelab9.models.ContactWithMessages;
import com.example.contactsmessagelab9.models.Message;
import com.example.contactsmessagelab9.services.ContactDao;
import com.example.contactsmessagelab9.services.MessageDao;
import
com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.Collections;
import java.util.Date;
import java.util.List;
import java.util.Vector;

```

```

public class ViewContactActivity extends AppCompatActivity {
    private EditText emailEditText;
    private EditText nameEditText;
    private EditText phoneEditText;
    private EditText addressEditText;
    private ListView messageListView;
    private Button saveButton;
    private Button deleteButton;
    private String name;
    private String phone;
    private String address;
    private String email;
    private AppDatabase db;
    private ContactDao contactDao;
    private MessageDao messageDao;
    private Contact contact;
    private ContactWithMessages contactWithMessages;
    private List<String> messageNames;
    private List<Message> messages;
    private ArrayAdapter<String> arrayAdapter;
    private FloatingActionButton fab;

    private boolean isValidEmail(CharSequence target) {
        if (target == null)
            return false;

        return
android.util.Patterns.EMAIL_ADDRESS.matcher(target).matches();
    }

    private void showErrorDialog(String message) {
        new AlertDialog.Builder(this)
            .setIcon(android.R.drawable.stat_notify_error)
            .setTitle("Error")
            .setMessage(message)
            .setNegativeButton("OK", (dialogInterface, i) -> {
                //set what would happen when positive button is
clicked
            })
            .show();
    }

    private void showAddDialogue() {
        new AlertDialog.Builder(this)

```

```

        .setIcon(android.R.drawable.ic_menu_save)
        .setTitle("Save")
        .setMessage("Are you sure you want to save contact?")
        .setPositiveButton("Yes", (dialogInterface, i) -> {
            new Thread(() -> {
                contactDao.update(new Contact(contact.uid, name,
email, phone, address));
                runOnUiThread(() -> {
                    setResult(1);
                    finish();
                });
            }).start();
            //set what would happen when positive button is
clicked
        }).setNegativeButton("No", (dialogInterface, i) -> {

        }).show();
    }

    private void initialiseUI() {
        emailEditText = findViewById(R.id.email_edit_text);
        nameEditText = findViewById(R.id.name_edit_text);
        phoneEditText = findViewById(R.id.phone_edit_text);
        addressEditText = findViewById(R.id.address_edit_text);
        saveButton = findViewById(R.id.save_button);
        deleteButton = findViewById(R.id.delete_button);
        fab = findViewById(R.id.fab);

        messageListView = findViewById(R.id.messages_list_view);

        emailEditText.setText(contact.email);
        nameEditText.setText(contact.name);
        phoneEditText.setText(contact.mobile);
        addressEditText.setText(contact.address);

        fab.setOnClickListener(v -> {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("New Message for " + contact.name);

            final EditText input = new EditText(this);
            input.setInputType(InputType.TYPE_CLASS_TEXT);
            builder.setView(input);

            builder.setPositiveButton("OK", (dialog, which) -> new
Thread(() -> {
                if (input.getText().length() != 0) {

```

```

        messageDao.insertMessages(new
Message(System.currentTimeMillis(), contact.uid,
input.getText().toString(), new Date()));
        initialiseMessages();
    }
    }).start());
    builder.setNegativeButton("Cancel", (dialog, which) ->
dialog.cancel());
    builder.show();
});

messageListView.setOnItemClickListener((parent, view, position,
id) -> {
    Message message = messages.get(position);

    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(message.date.toString());

    final EditText input = new EditText(this);
    input.setInputType(InputType.TYPE_CLASS_TEXT);
    input.setText(message.message);
    builder.setView(input);

    builder.setPositiveButton("OK", (dialog, which) -> new
Thread(() -> {
        if (input.getText().length() != 0) {
            messageDao.updateMessages(new
Message(message.messageId, contact.uid, input.getText().toString(), new
Date()));
            initialiseMessages();
        }
    }).start());
    builder.setNegativeButton("Cancel", (dialog, which) ->
dialog.cancel());
    builder.setNeutralButton("Delete", (dialog, which) -> new
Thread(() -> {
        messageDao.deleteMessage(message);
        initialiseMessages();
    }).start());
    builder.show();
});

saveButton.setOnClickListener(v -> {
    name = nameEditText.getText().toString();
    email = emailEditText.getText().toString();
    phone = phoneEditText.getText().toString();

```

```

        address = addressEditText.getText().toString();

        if (name.length() == 0) {
            showErrorDialog("Name cannot be empty");
            return;
        }

        if (email.length() != 0 && !isValidEmail(email)) {
            showErrorDialog("Invalid email");
            return;
        }

        if (!PhoneNumberUtils.isGlobalPhoneNumber(phone)) {
            showErrorDialog("Invalid Phone");
            return;
        }

        showAddDialogue();
    });

    deleteButton.setOnClickListener(v -> showDeleteDialogue());
}

private void showDeleteDialogue() {
    new AlertDialog.Builder(this)
        .setIcon(android.R.drawable.ic_delete)
        .setTitle("Delete")
        .setMessage("Are you sure you want to delete this
contact?")
        .setPositiveButton("Yes", (dialogInterface, i) -> {
            new Thread(() -> {
                contactDao.delete(contact);
                runOnUiThread(() -> {
                    setResult(1);
                    finish();
                });
            }).start();
            //set what would happen when positive button is
clicked
        }).setNegativeButton("No", (dialogInterface, i) -> {

        }).show();
}

@Override
protected void onCreate(Bundle savedInstanceState) {

```



```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_contact);
        contact = (Contact) getIntent().getSerializableExtra("contact");

        db = Room.databaseBuilder(this, AppDatabase.class,
"contacts").build();
        contactDao = db.contactDao();
        messageDao = db.messageDao();
        initialiseUI();
        new Thread(this::initialiseMessages).start();
    }

    private void updateMessageList() {
        arrayAdapter = new ArrayAdapter<>(this,
R.layout.activity_listview, R.id.contact_text_view, messageNames);
        messageListView.setAdapter(arrayAdapter);
    }

    private void initialiseMessages() {
        contactWithMessages =
contactDao.contactsWithMessagesOfId(contact.uid);
        messages = contactWithMessages.messages;
        Collections.reverse(messages);
        messageNames = new Vector<>(messages.size());

        for (Message message : messages) {
            messageNames.add(message.message);
        }
        runOnUiThread(this::updateMessageList);
    }
}

```

## DateConverter.java

```

package com.example.contactsmessagelab9.util;

import androidx.room.TypeConverter;

import java.util.Date;

public class DateConverter {

    @TypeConverter
    public static Date toDate(Long dateLong) {
        return dateLong == null ? null : new Date(dateLong);
    }
}

```

```

    }

    @TypeConverter
    public static Long fromDate(Date date) {
        return date == null ? null : date.getTime();
    }
}

```

## AppDatabase.java

```

package com.example.contactsmessageLab9;

import androidx.room.Database;
import androidx.room.RoomDatabase;

import com.example.contactsmessageLab9.models.Contact;
import com.example.contactsmessageLab9.models.Message;
import com.example.contactsmessageLab9.services.ContactDao;
import com.example.contactsmessageLab9.services.MessageDao;

@Database(entities = {Contact.class, Message.class}, version = 1,
exportSchema = false)
public abstract class AppDatabase extends RoomDatabase {
    public abstract ContactDao contactDao();

    public abstract MessageDao messageDao();
}

```

## Layouts

### activity\_add.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingHorizontal="6dp"
    android:paddingVertical="6dp"
    tools:context=".AddActivity">

```

```
<EditText
    android:id="@+id/name_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/email_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/phone_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Phone Number"
    android:inputType="phone" />

<EditText
    android:id="@+id/address_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Address"
    android:inputType="textPostalAddress" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/save_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:text="Save" />

</RelativeLayout>

</LinearLayout>
```

## activity\_listview.java

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/contact_text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="4dp"
    android:padding="4dp"
    android:textSize="20dp">

</TextView>
```

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:paddingHorizontal="5dp"
    android:paddingVertical="5dp"
    tools:context=".views.MainActivity">

    <LinearLayout
        android:id="@+id/search_bar"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <EditText
            android:id="@+id/search_edit_text"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="4dp"
            android:layout_weight="1" />

        <Button
            android:id="@+id/search_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
```

```

        android:layout_margin="4dp"
        android:text="Search" />

</LinearLayout>

<ListView
    android:id="@+id/contacts_list_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_below="@+id/search_bar" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:contentDescription="Add"
    android:src="@android:drawable/ic_input_add" />

</RelativeLayout>

```

### activity\_view\_contact.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingHorizontal="6dp"
    android:paddingVertical="6dp"
    tools:context=".views.AddActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <EditText
            android:id="@+id/name_edit_text"
            android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="textPersonName" />

<EditText
    android:id="@+id/email_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/phone_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Phone Number"
    android:inputType="phone" />

<EditText
    android:id="@+id/address_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Address"
    android:inputType="textPostalAddress" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/delete_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="16dp"
        android:layout_toLeftOf="@+id/save_button"
        android:text="Delete" />

    <Button
        android:id="@+id/save_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:text="Save" />
</RelativeLayout>

```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Messages"
    android:textSize="24dp"
    android:textStyle="bold" />

<ListView
    android:id="@+id/messages_list_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent" />
</LinearLayout>
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_margin="16dp"
    android:contentDescription="Add"
    android:src="@android:drawable/ic_input_add" />

</RelativeLayout>
```

10:12



## ContactsMessageLab9

Sahil

sahilbondre@gmail.com

1234567890

Pune

DELETE

SAVE



Save

Are you sure you want to save contact?

NO

YES

Yo Yo





10:12



## ContactsMessageLab9

Sahil

sahilbondre@gmail.com

1234567890

Pune

DELETE

SAVE

 Delete

Are you sure you want to delete this contact?

NO

YES

Yo Yo



10:12



## ContactsMessageLab9

Sahil

sahilbondre@gmail.com

1234567890

Pune

DELETE

SAVE

New Message for Sahil

CANCEL

OK

Yo Yo



10:12



## ContactsMessageLab9

Sahil

sahilbondre@gmail.com

1234567890

Pune

DELETE

SAVE

Fri Apr 16 19:56:49 GMT+05:30  
2021

New

DELETE

CANCEL

OK

Yo Yo



10:11



## ContactsMessageLab9

Sahil

sahilbondre@gmail.com

1234567890

Pune

DELETE

SAVE

## Messages

New

Wassup!

Hello World!

Hi

Yo Yo





10:11



## ContactsMessageLab9

SEARCH

Sahil

John Doe

