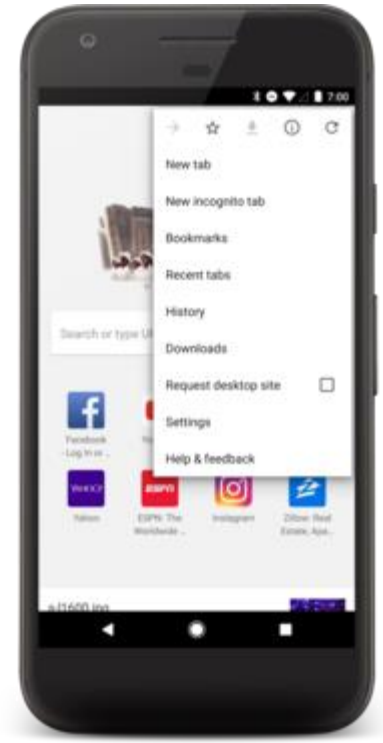
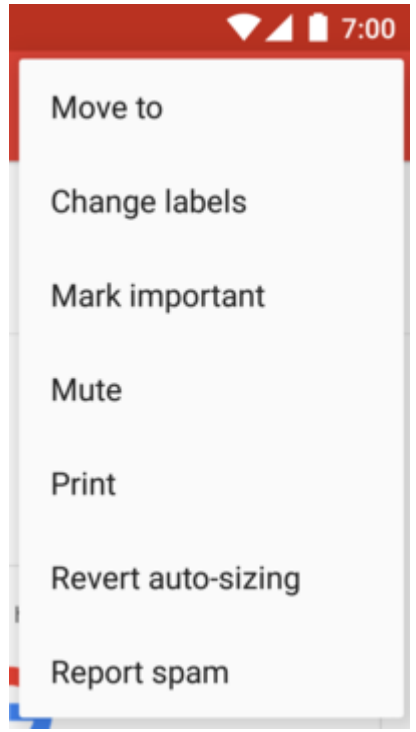
MENUS



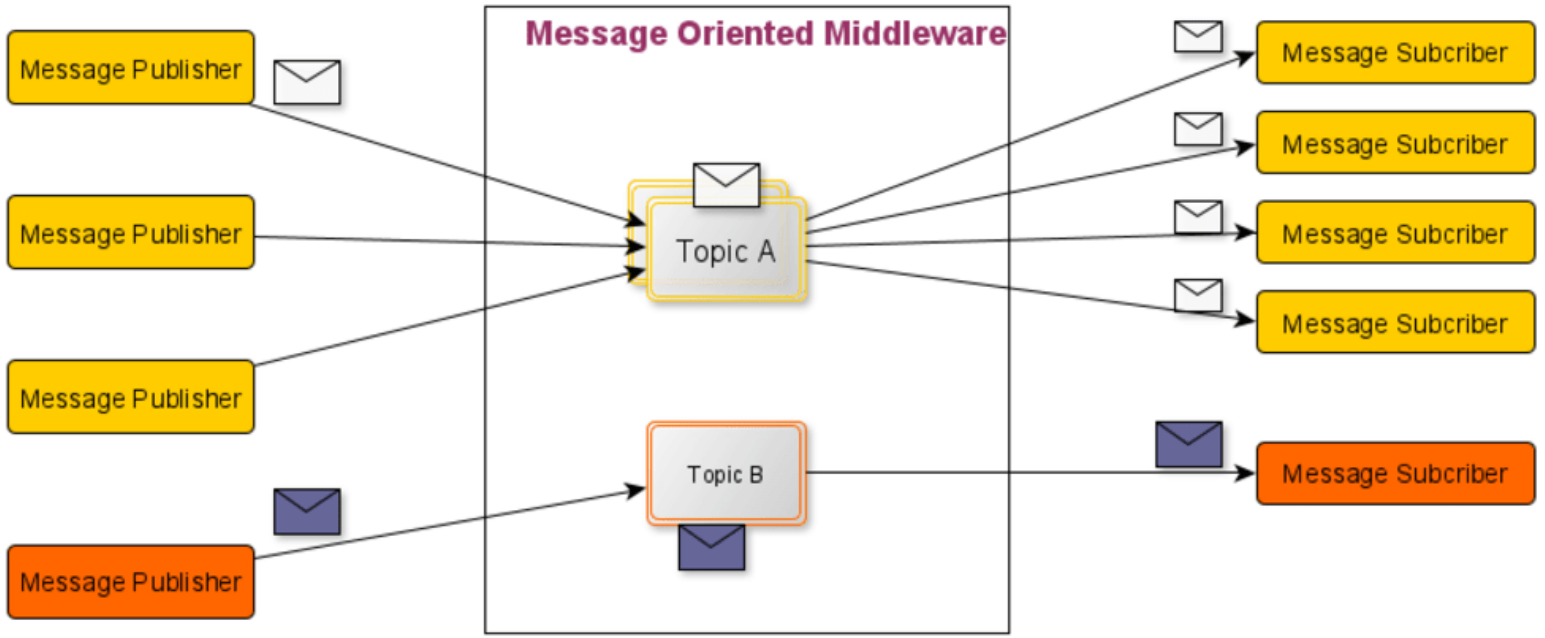
Options menu
in the Browser

Floating context menu (left)
Contextual action bar (right)



A popup menu in the Gmail app, anchored to the overflow button at the top-right

PUBLISHER SUBSCRIBER



BROADCAST RECEIVER

```
private IntentFilter intentFilter;
private BroadcastReceiver receiver;


onCreate() {
  …
  intentFilter = new IntentFilter("android.intent.action.AIRPLANE_MODE");

  receiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.d("AirplaneMode", "Service state changed");
    }
  };
}

onStart() {
   registerReceiver(receiver, intentFilter);
}

onPause() {
   unregisterReceiver(receiver);
}
```

ROOM PERSISTENCE LIBRARY

Room persistence library provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite

Three major components in Room:
•The database class that holds the database and serves as the main access point for the underlying connection to your app's persisted data.
•Data entities that represent tables in your app's database.
•Data access objects (DAOs) that provide methods that your app can use to query, update, insert, and delete data in the database.

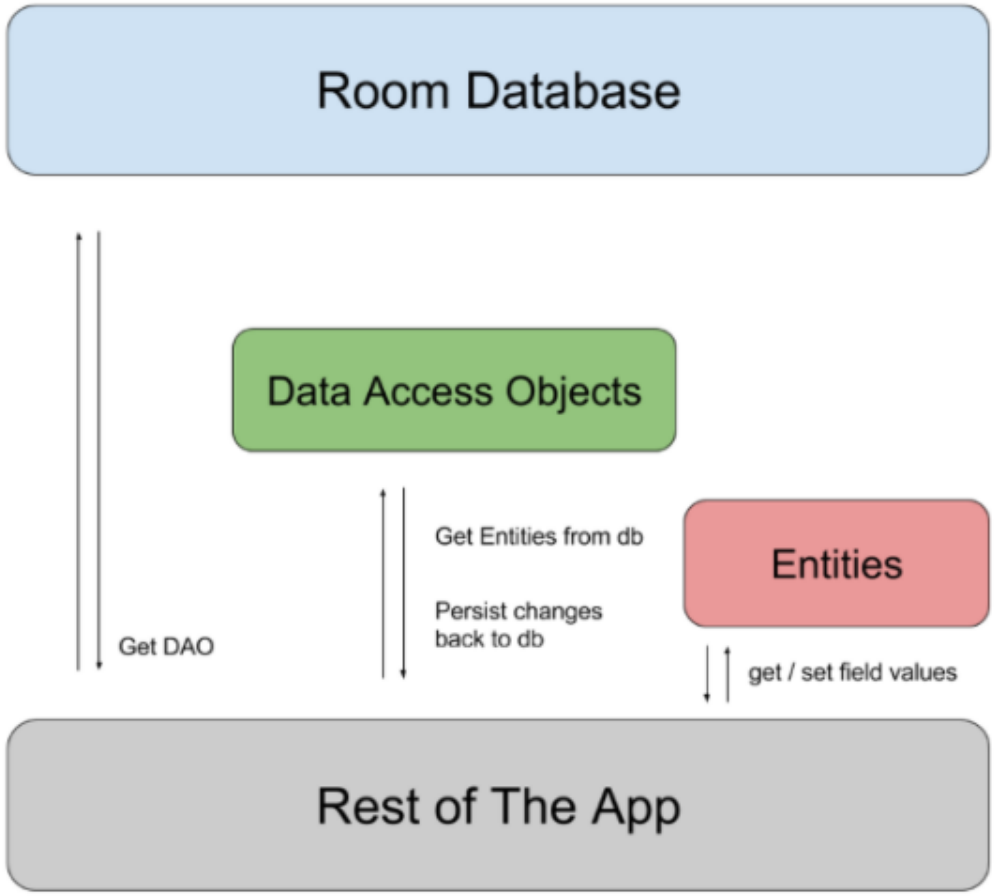


Figure 1. Diagram of Room library architecture.

ROOM PERSISTENCE LIBRARY

```
@Entity
public class User {
    @PrimaryKey
    public int uid;

    @ColumnInfo(name = "first_name")
    public String firstName;

    @ColumnInfo(name = "last_name")
    public String lastName;
}
```

```
@Database(entities = {User.class}, version
= 1)
public abstract class AppDatabase
extends RoomDatabase {
    public abstract UserDao userDao();
}
```

```
@Dao
public interface UserDao {
    @Query("SELECT * FROM user")
    List<User> getAll();

    @Query("SELECT * FROM user WHERE uid IN (:userIds)")
    List<User> loadAllByIds(int[] userIds);

    @Query("SELECT * FROM user WHERE first_name LIKE :first AND " +
           "last_name LIKE :last LIMIT 1")
    User findByName(String first, String last);

    @Insert
    void insertAll(User... users);

    @Delete
    void delete(User user);
}
```

ROOM PERSISTENCE LIBRARY

```
AppDatabase db =
Room.databaseBuilder(getApplicationContext(),
        AppDatabase.class, "database-name").build();
```

```
UserDao userDao = db.userDao();
List<User> users = userDao.getAll();
```