Android storage

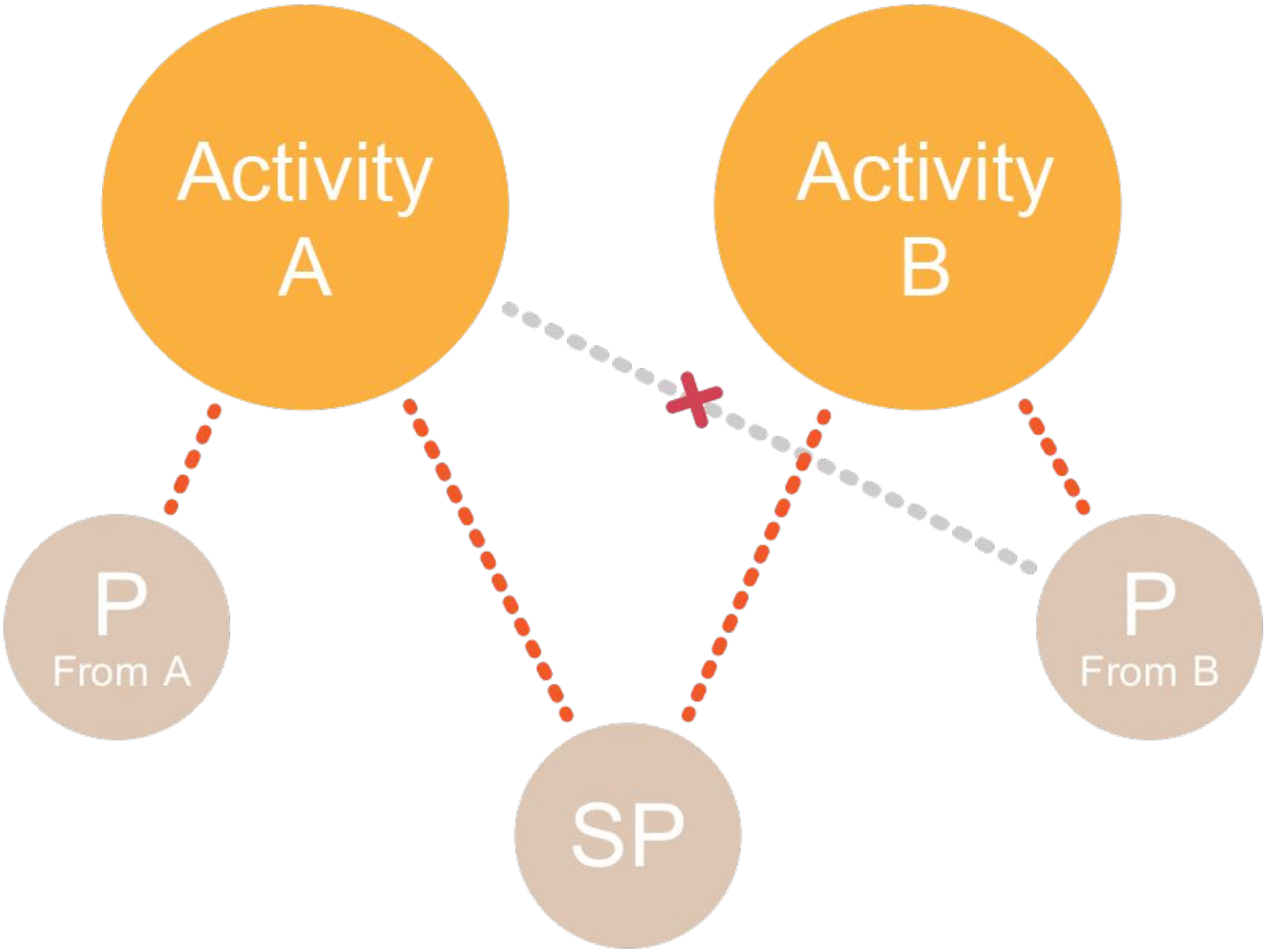
Agenda

- Preference and SharedPreference
- SQLite
- ROOM

Preference

SP = Shared Preferences

P = Preferences



Using preference

Shared preference

```
SharedPreferences sp = getSharedPreferences(name, mode);
```

Preference

```
SharedPreferences sp = getPreferences(mode);
```

All mode described here: https://www.tutorialspoint.com/android/android_shared_preferences.htm

Available modes

Sr.No	Mode & description
1	**MODE_APPEND** This will append the new preferences with the already existing preferences
2	**MODE_ENABLE_WRITE_AHEAD_LOGGING** Database open flag. When it is set , it would enable write ahead logging by default
3	**MODE_MULTI_PROCESS** This method will check for modification of preferences even if the sharedpreference instance has already been loaded
4	**MODE_PRIVATE** By setting this mode, the file can only be accessed using calling application
5	**MODE_WORLD_READABLE** This mode allow other application to read the preferences
6	**MODE_WORLD_WRITEABLE** This mode allow other application to write the preferences

From https://www.tutorialspoint.com/android/android_shared_preferences.htm

Support types

- Boolean
- Float
- Integer
- Long
- String
- Array of string

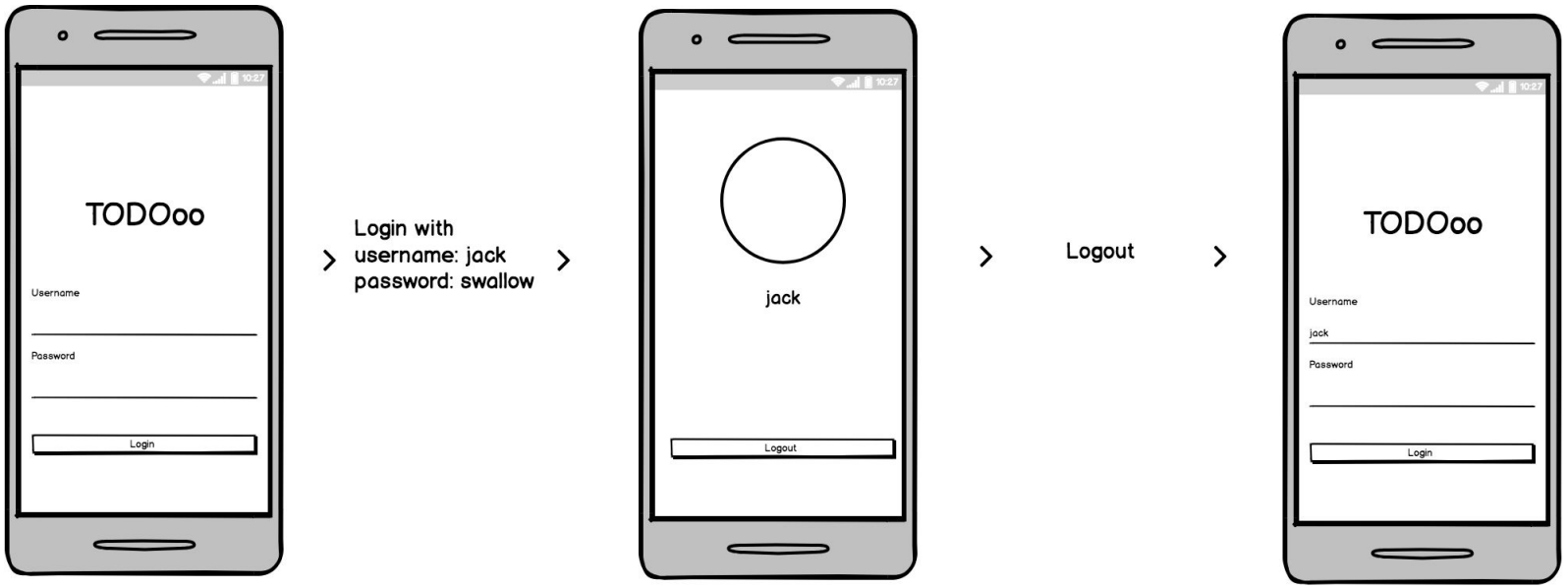
Insert into preference

```
SharedPreferences pref = getSharedPreferences("user", Context.MODE_PRIVATE);
SharedPreferences.Editor editor = pref.edit();
editor.putString("username", usernameInput.getText().toString());
editor.apply();
```

Reading from preference

```
SharedPreferences pref = getSharedPreferences("user", Context.MODE_PRIVATE);
String lastUser = pref.getString("username", "");
```

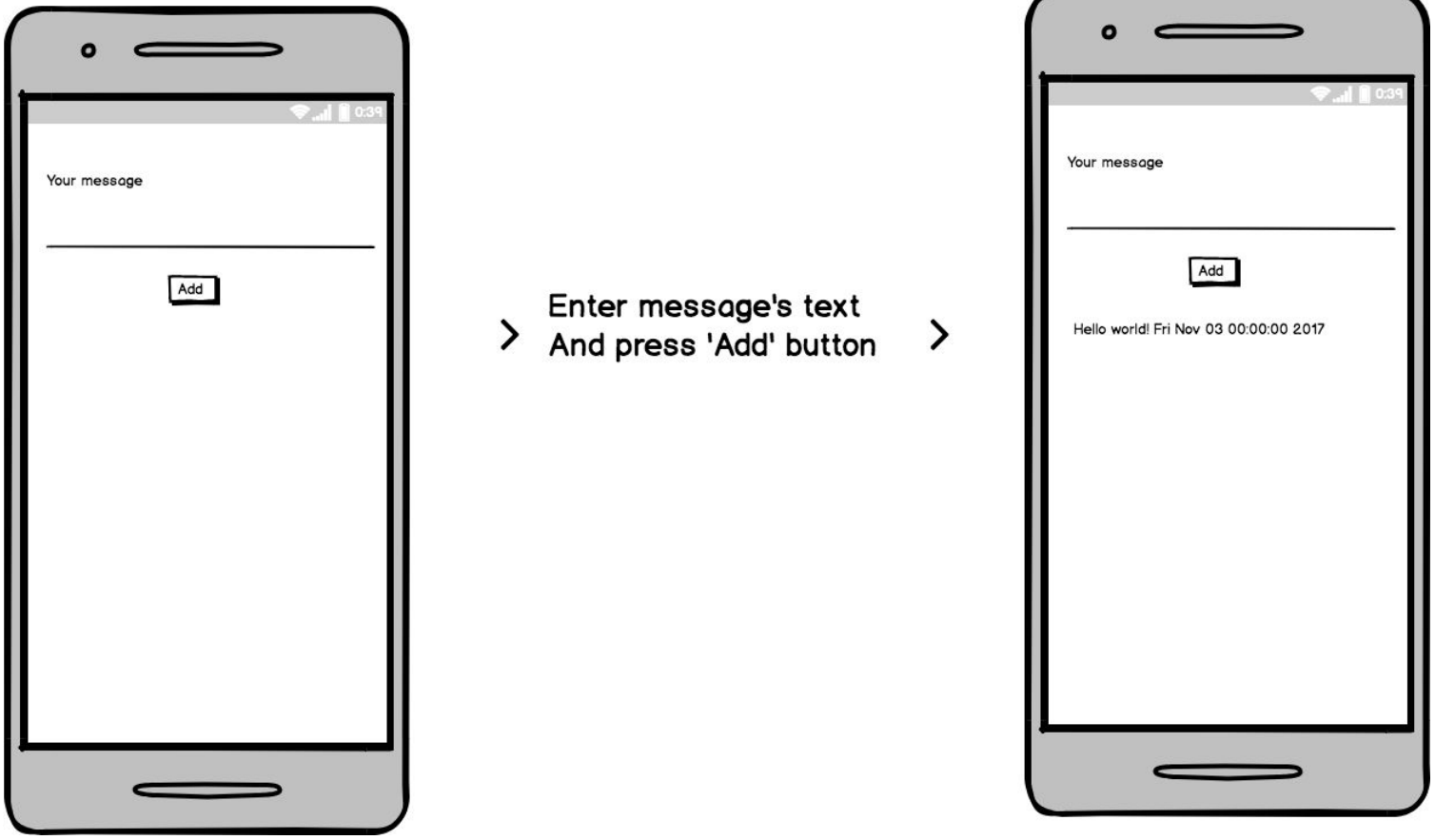
Scenario: remember the last logged in user name



Last logged in username should filled in automatically

SQLite

Scenario


Your message
Add
Enter message's text
And press 'Add' button
Your message
Add
Hello world! Fri Nov 03 00:00:00 2017

Setup

Create class extended from ‘SQLiteOpenHelper’

```
public class MessagesDB extends SQLiteOpenHelper {

   private static final String DB_NAME = "SQLITE_DEMO";
   private static final int DB_VERSION = 1;
   private static final String TABLE_MESSAGE = "Messages";
   private static final String COL_TEXT = "message";
   private static final String COL_TIME = "time";


   public MessagesDB(Context context) {
       super(context, DB_NAME, null, DB_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase sqLiteDatabase) {
       sqLiteDatabase.execSQL(String.format("CREATE TABLE %s (_id INTEGER PRIMARY KEY AUTOINCREMENT, %s TEXT, %s TEXT);",
TABLE_MESSAGE, COL_TEXT, COL_TIME));

       sqLiteDatabase.execSQL(String.format("INSERT INTO %s ('%s', '%s') VALUES ('%s', '%s');", TABLE_MESSAGE, COL_TEXT, COL_TIME,
"Hello world", new Date().toString()));
   }

   @Override
   public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

   }
}
```

Setup

Create entity class ‘Message’

```
public class Message {

   private String text;
   private String time;

   public String toString(){
       return String.format("%s %s", this.text, this.time);
   }

    // Getters and Setters...
}
```

Querying

Add method for perform ‘SELECT’ operation

```
public List<Message> findAll(){
       Cursor cursor = getWritableDatabase().rawQuery(String.format("SELECT * FROM %s;",
TABLE_MESSAGE), null);
       cursor.moveToFirst();

       List<Message> messages = new ArrayList<>();

       Message message;
       while (!cursor.isAfterLast()) {
           message = new Message();
           message.setText(cursor.getString(cursor.getColumnIndex(COL_TEXT)));
           message.setTime(cursor.getString(cursor.getColumnIndex(COL_TIME)));

           messages.add(message);

           cursor.moveToNext();
       }

       return messages;
   }
```

Insert

Add method to perform ‘INSERT’ operation

```
public void insert(Message message) {
       getWritableDatabase().execSQL(String.format("INSERT INTO %s ('%s', '%s')
VALUES ('%s', '%s');", TABLE_MESSAGE, COL_TEXT, COL_TIME, message.getText(),
message.getTime()));
   }
```

SQLite Cheat sheet

http://www.sqlitetutorial.net/sqlite-cheat-sheet/

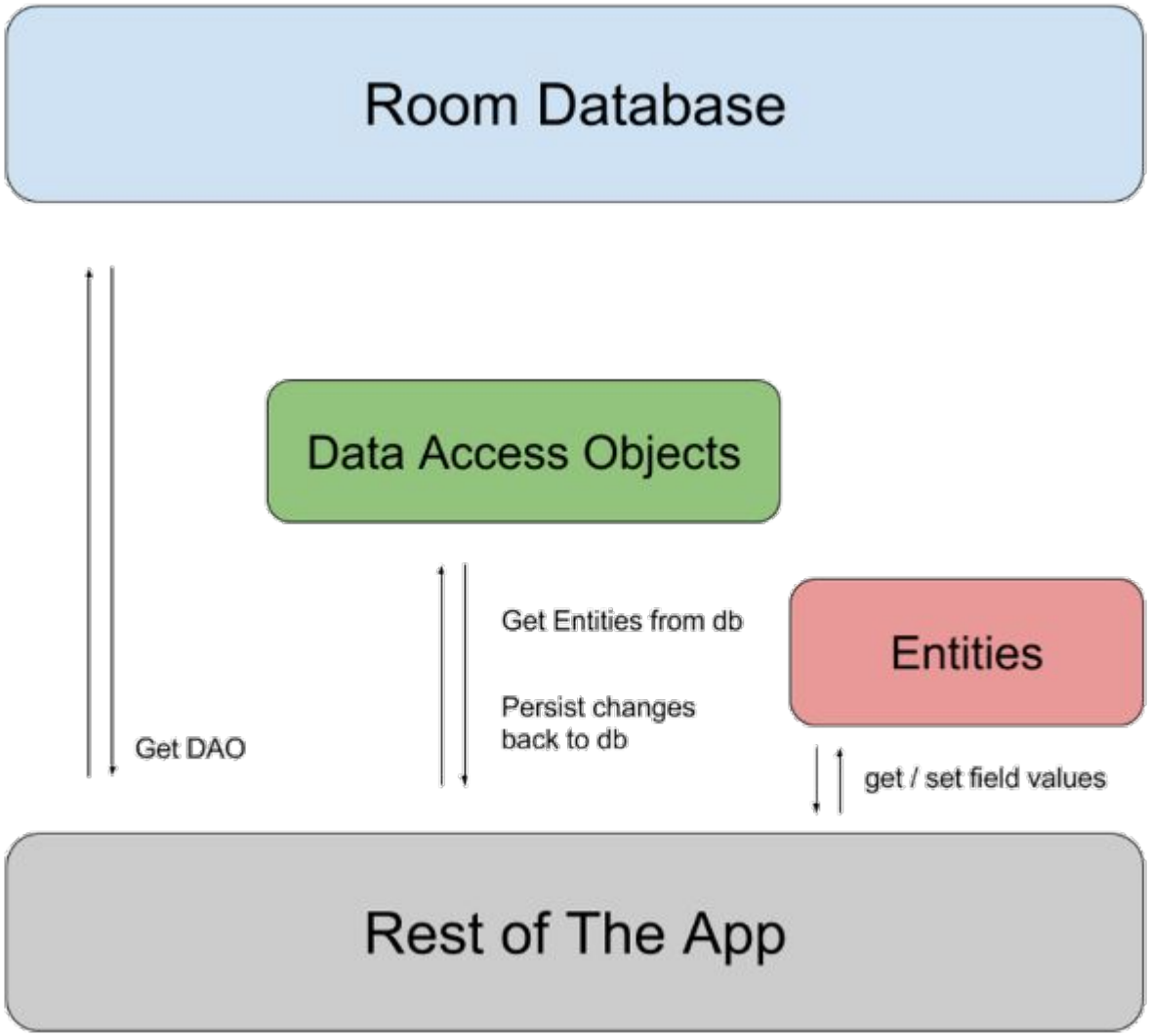
ORMLite

http://ormlite.com/sqlite_java_android_orm.shtml

https://realm.io/

ROOM

https://developer.android.com/topic/libraries/architecture/room.html?



https://developer.android.com/topic/libraries/architecture/room.html

Add repository to project's build.gradle

```
allprojects {
    repositories {
        jcenter()
        maven { url 'https://maven.google.com' }
    }
}
```

Add dependencies to app’s build.gradle

```
dependencies {
    // … Other dependencies
   implementation "android.arch.persistence.room:runtime:1.0.0-rc1"
   annotationProcessor "android.arch.persistence.room:compiler:1.0.0-rc1"
}
```

Room’s core components

- Database
- Entity
- DAO

Create database class

```
//UserInfoDatabase.java
@Database(entities = {UserInfo.class}, version = 1)
public abstract class UserInfoDatabase  extends RoomDatabase{
    public abstract UserInfoDAO userInfoRoomDAO();
}
```

Create Entity class

```java
//UserInfo.java
@Entity
public class UserInfo {

   @PrimaryKey(autoGenerate = true)
   private int id;

   @ColumnInfo(name = "name")
   private String name;

   @ColumnInfo(name = "age")
   private String age;
}
```

Create DAO interface

```
//UserInfoDAO.java
@Dao
public interface UserInfoRoomDao {
    @Query("SELECT * FROM UserInfo")
    List<UserInfo> getAll();


    @Insert
    void insert(UserInfo userInfo);
}
```

Building ROOM's Database

```
//MainActivity.java
UserInfoDatabase userInfoDB = Room.databaseBuilder(getApplicationContext(),
        UserInfoDatabase.class, "DEMOINFO")
        .fallbackToDestructiveMigration()
        .build();
```

Insert

```
final UserInfo user = new UserInfo();
user.setName(nameEditText.getText().toString());
user.setAge(ageEditText.getText().toString());

new AsyncTask<Void, Void, UserInfo>() {
    @Override
    protected UserInfo doInBackground(Void... params) {
        userInfoDB.userInfoDAO().insert(user);
        return user;
    }

    @Override
    protected void onPostExecute( UserInfo userInfo) {
        Intent intent = new Intent(UserInfoFormActivity.this, UserListActivity.class);
        startActivity(intent);
    }
}.execute();
```

Select

```
new AsyncTask<Void, Void, List<UserInfo>>() {
    @Override
    protected List<UserInfo> doInBackground(Void... params) {

        return userInfoDB.userInfoDAO().getAll();
    }

    @Override
    protected void onPostExecute(List<UserInfo> users) {
        //Process data...
    }
}.execute();
```