

11-Migration

Dans `list_item.xml`, ajouter l'attribut :

```
android:clickable="true"
```

Et suppression du bouton edit :

```
<ImageView
    android:id="@+id/ivRowEdit"
    android:layout_width="20dp"
    android:layout_height="20dp"
    android:layout_margin="10dp"
    android:backgroundTint="#9E9E9E"
    android:src="@drawable/ic_edit"
/>
```

Petits réglages pour améliorer l'UI :

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

    <ImageView
        android:id="@+id/ivRowDelete"
        android:layout_width="20dp"
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        android:backgroundTint="#9E9E9E"
        android:src="@drawable/ic_delete" />

</LinearLayout>
```

(ajout de match_parent pour les height)

Commenter le premier listener dans BookListAdapter

Puis ajouter un listener sur le clique :

```
itemView.setOnClickListener {
    val intent = Intent(context, EditBookActivity::class.java)
```

```

        intent.putExtra("id", bookList[pos].id)
        intent.putExtra("author", bookList[pos].author)
        intent.putExtra("book", bookList[pos].book)
        (context as Activity).startActivityForResult(
            intent,
            MainActivity.UPDATE_BOOK_REQUEST_CODE
        )
    }
}

```

Ajout d'un nouveau champ à la classe Book

Ajout d'un champ description dans Book :

```

class Book(
    @PrimaryKey
    val id: String,
    @ColumnInfo(name = "author")
    val author: String,
    val book: String,
    val description: String
) {}

```

Ajout d'un text field à activity_new_book :

```

<EditText
    android:id="@+id/etDescription"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:ems="15"
    android:hint="@string/description"
    android:maxLength="30"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etBookName"
/>

```

Ajout du nouveau champ dans NewBookActivity :

```

val authorName = etAuthorName.text.toString()
val bookName = etBookName.text.toString()
val description = etDescription.text.toString()
resultIntent.putExtra(NEW_AUTHOR, authorName)
resultIntent.putExtra(NEW_BOOK, bookName)

```

```
resultIntent.putExtra(NEW_DESCRIPTION, description)
setResult(Activity.RESULT_OK, resultIntent)
```

Déclaration du tag :

```
const val NEW_DESCRIPTION: String = "new_description"
```

Ajout de la description dans la fonction insertNewBook de MainActivity :

```
private fun insertNewBook(data: Intent?) {
    val id = UUID.randomUUID().toString()
    val authorName: String? =
data?.getStringExtra(NewBookActivity.NEW_AUTHOR)
    val bookName : String? = data?.getStringExtra(NewBookActivity.NEW_BOOK)
    val description : String? =
data?.getStringExtra(NewBookActivity.NEW_DESCRIPTION)
    val book = Book(id, authorName!!, bookName!!, description!!)

    bookViewModel.insert(book)

    Toast.makeText(applicationContext, R.string.saved,
Toast.LENGTH_LONG).show()
}
```

Même chose pour le updateBook :

```
private fun updateBook(data: Intent?) {
    val id : String ? = data?.getStringExtra(EditBookActivity.ID)
    val author: String? =
data?.getStringExtra(EditBookActivity.UPDATED_AUTHOR)
    val book : String ? =
data?.getStringExtra(EditBookActivity.UPDATED_BOOK)
    val description: String? =
data?.getStringExtra(EditBookActivity.UPDATED_DESCRIPTION)
    val updated_book = Book(id!!, author!!, book!!, description!!)
    bookViewModel.update(updated_book)
}
```

Dans BookListAdapter , modification du listener pour ajouter la description :

```
itemView.setOnClickListener {
    val intent = Intent(context, EditBookActivity::class.java)
    intent.putExtra("id", bookList[pos].id)
    intent.putExtra("author", bookList[pos].author)
```

```

        intent.putExtra("book", bookList[pos].book)
        intent.putExtra("description", bookList[pos].description)
        (context as Activity).startActivityForResult(
            intent,
            MainActivity.UPDATE_BOOK_REQUEST_CODE
        )
    }
}

```

Dans le EditBookActivity:

```

val bundle: Bundle? = intent.extras
bundle?.let {
    id = bundle.getString("id")
    val author: String? = bundle.getString("author")
    val book: String? = bundle.getString("book")
    val description: String? = bundle.getString("description")

    etAuthorName.setText(author)
    etBookName.setText(book)
    etDescription.setText(description)
}

bAdd.setOnClickListener {
    val updatedAuthor = etAuthorName.text.toString()
    val updatedBook = etBookName.text.toString()
    val updatedDescription = etDescription.text.toString()
    val resultIntent = Intent()
    resultIntent.putExtra(ID, id)
    resultIntent.putExtra(UPDATED_AUTHOR, updatedAuthor)
    resultIntent.putExtra(UPDATED_BOOK, updatedBook)
    resultIntent.putExtra(UPDATED_DESCRIPTION, updatedDescription)
    setResult(Activity.RESULT_OK, resultIntent)
    finish()
}

```

Ajout du tag :

```

const val UPDATED_DESCRIPTION: String = "update_description"

```

Migrations

Dans BookRoomDatabase:

```

// @Database(entities = [Book::class], version = 1, exportSchema = false)
@Database(entities = [Book::class], version = 2, exportSchema = false)

```

Ajout d'une migration :

```
val MIGRATION_1_2: Migration = object : Migration(1, 2) {
    override fun migrate(database: SupportSQLiteDatabase) {
        database.execSQL(
            "ALTER TABLE book " +
            "ADD COLUMN description TEXT Default 'Add Description' " +
            "NOT NULL"
        )
    }
}
```

Appliquer la migration :

```
fun getDatabase(context: Context): BookRoomDatabase? {
    if (bookRoomInstance == null) {
        synchronized(BookRoomDatabase::class.java) {
            if (bookRoomInstance == null) run {
                bookRoomInstance = Room.databaseBuilder<BookRoomDatabase>(
                    context.applicationContext,
                    BookRoomDatabase::class.java,
                    "book_room_db.sqlite"
                )
                    .addMigrations(MIGRATION_1_2) // AJOUT DE CETTE LIGNE
                    .build()
            }
        }
    }
    return bookRoomInstance!!
}
```