Ex. No. 4	**Implementing CRUD Operations Using SQLite in Android**
Date of Exercise	28 - 08 - 2024

Aim

The aim of this experiment is to design and implement a mobile application for a movie streaming service. The application allows users to select movies from a ListView and add them to a watchlist. The system includes features for editing, deleting, and inserting movies, as well as viewing and managing a dynamic watchlist.

Program

MainActivity.kt

```
import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import android.widget.*
import android.widget.PopupMenu

class MainActivity : AppCompatActivity() {
    private lateinit var movieListView: ListView
    private val movieList = mutableListOf(
        Movie(R.drawable.movie1, "Avatar", 100.00),
        Movie(R.drawable.movie2, "Avengers", 200.00),
        Movie(R.drawable.movie3, "Titanic", 500.00),
        Movie(R.drawable.movie4, "Star Wars", 100.00),
        Movie(R.drawable.movie5, "The Lion King", 300.00)
    )
    private val watchlist = mutableMapOf<Movie, Int>()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        movieListView = findViewById(R.id.movie_list_view)

        val adapter = object : ArrayAdapter<Movie>(this, R.layout.list_item_movie, movieList)
{
            override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
                val view = convertView ?:
LayoutInflater.from(context).inflate(R.layout.list_item_movie, parent, false)
                val movie = getItem(position)!!

view.findViewById<ImageView>(R.id.movie_image).setImageResource(movie.imageResId)
                view.findViewById<TextView>(R.id.movie_name).text = movie.name
                view.findViewById<TextView>(R.id.movie_price).text = "Price: $${movie.price}"

                // Set up the popup menu on item click
                view.setOnClickListener {
                    showPopupMenu(view, movie, position)
                }

                return view
            }
        }
        movieListView.adapter = adapter
        findViewById<Button>(R.id.view_watchlist_button).setOnClickListener {
```

```
        val intent = Intent(this, WatchlistActivity::class.java)
        intent.putExtra("watchlist", HashMap(watchlist))
        startActivity(intent)
      }

    }

    // Method to show the popup menu
    private fun showPopupMenu(view: View, movie: Movie, position: Int) {
      val popup = PopupMenu(this, view)
      popup.inflate(R.menu.movie_popup_menu)
      popup.setOnMenuItemClickListener { menuItem ->
        when (menuItem.itemId) {
          R.id.edit -> {
            showEditDialog(movie)
            true
          }
          R.id.delete -> {
            movieList.remove(movie)
            (movieListView.adapter as ArrayAdapter<*>).notifyDataSetChanged()
            Toast.makeText(this, "${movie.name} deleted",
  Toast.LENGTH_SHORT).show()
            true
          }
          R.id.insert -> {
            showInsertDialog(position)
            true
          }
          R.id.add_to_watchlist -> {
            val count = watchlist.getOrDefault(movie, 0) + 1
            watchlist[movie] = count
```

```
                    Toast.makeText(this, "${movie.name} added to watchlist",
Toast.LENGTH_SHORT).show()
                    true
                }
                else -> false
            }
        }
        popup.show()
    }

    // Method to show the Edit dialog
    private fun showEditDialog(movie: Movie) {
        val dialogView = LayoutInflater.from(this).inflate(R.layout.dialog_edit_movie, null)
        val dialog = AlertDialog.Builder(this)
            .setTitle("Edit Movie")
            .setView(dialogView)
            .setPositiveButton("Save") { _, _ ->
                val newName =
dialogView.findViewById<EditText>(R.id.edit_movie_name).text.toString()
                val newPrice =
dialogView.findViewById<EditText>(R.id.edit_movie_price).text.toString().toDouble()
                movie.name = newName
                movie.price = newPrice
                (movieListView.adapter as ArrayAdapter<*>).notifyDataSetChanged()
                Toast.makeText(this, "${movie.name} updated", Toast.LENGTH_SHORT).show()
            }
            .setNegativeButton("Cancel", null)
            .create()
        dialog.show()
    }
    // Method to show the Insert dialog
    private fun showInsertDialog(position: Int) {
        val dialogView = LayoutInflater.from(this).inflate(R.layout.dialog_insert_movie, null)
        val dialog = AlertDialog.Builder(this)
            .setTitle("Insert Movie")
            .setView(dialogView)
            .setPositiveButton("Add") { _, _ ->
                val name =
dialogView.findViewById<EditText>(R.id.insert_movie_name).text.toString()
                val price =
dialogView.findViewById<EditText>(R.id.insert_movie_price).text.toString().toDouble()
                val newMovie = Movie(R.drawable.default_image, name, price)
                movieList.add(position, newMovie)
                (movieListView.adapter as ArrayAdapter<*>).notifyDataSetChanged()
                Toast.makeText(this, "$name added", Toast.LENGTH_SHORT).show()
            }
            .setNegativeButton("Cancel", null)
            .create()
        dialog.show()
    }
}
```

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ListView
        android:id="@+id/movie_list_view"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />

    <Button
        android:id="@+id/view_watchlist_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="View Watchlist" />
</LinearLayout>
```

WatchlistActivity.kt

```
import android.os.Bundle
import android.widget.ListView
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class WatchlistActivity : AppCompatActivity() {
    private lateinit var listView: ListView
    private lateinit var totalPriceTextView: TextView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_watchlist)
        listView = findViewById(R.id.watchlist_list_view)
        totalPriceTextView = findViewById(R.id.total_price)

        val watchlist = intent.getSerializableExtra("watchlist") as HashMap<Movie, Int>
        val adapter = WatchlistAdapter(this, watchlist)
        listView.adapter = adapter

        // Calculate the total price
        val totalPrice = watchlist.entries.sumOf { it.key.price * it.value }
        totalPriceTextView.text = "Total Price: Rs.${totalPrice}"
    }
}
```

activity_watchlist.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp">
    <ListView
        android:id="@+id/watchlist_list_view"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
    <TextView
        android:id="@+id/total_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Total Price: $0.00"
        android:textSize="18sp"
        android:paddingTop="20dp" />
</LinearLayout>
```

WatchlistAdapter.kt

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ArrayAdapter
import android.widget.ImageView
import android.widget.TextView

class WatchlistAdapter(context: Context, private val watchlist: Map<Movie, Int>)
    : ArrayAdapter<Map.Entry<Movie, Int>>(context, 0, watchlist.entries.toList()) {
    override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
        val view = convertView ?: LayoutInflater.from(context).inflate(R.layout.list_item_watchlist, parent, false)
        val entry = getItem(position)
        val movie = entry?.key
        val count = entry?.value ?: 0

        view.findViewById<ImageView>(R.id.watchlist_movie_image).setImageResource(movie?.imageResId ?: R.drawable.default_image)
        view.findViewById<TextView>(R.id.watchlist_movie_name).text = movie?.name ?: "Unknown"
        view.findViewById<TextView>(R.id.watchlist_movie_count).text = "Count: $count"
        view.findViewById<TextView>(R.id.watchlist_movie_total_price).text = "Price: Rs.${movie?.price?.times(count) ?: 0.0}"

        return view
    }
}
```

list_item_movie.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:orientation="horizontal"
android:padding="10dp">

<ImageView
  android:id="@+id/movie_image"
  android:layout_width="60dp"
  android:layout_height="60dp"
  android:src="@drawable/movie1" />

<LinearLayout
  android:layout_width="0dp"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:orientation="vertical"
  android:paddingStart="10dp"
  android:paddingLeft="10dp">

  <TextView
    android:id="@+id/movie_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Movie Name"
    android:textSize="16sp" />
  <TextView
    android:id="@+id/movie_price"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Price: $0.00" />
</LinearLayout>
</LinearLayout>
```

list_item_watchlist.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">

    <ImageView
        android:id="@+id/watchlist_movie_image"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:scaleType="centerCrop" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical"
        android:paddingStart="10dp">

        <TextView
            android:id="@+id/watchlist_movie_name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="16sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/watchlist_movie_count"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/watchlist_movie_total_price"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
</LinearLayout>
```

MovieDatabaseHelper.kt

```
import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class MovieDatabaseHelper(context: Context) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) {
    companion object {
        private const val DATABASE_NAME = "movies.db"
        private const val DATABASE_VERSION = 1
        private const val TABLE_MOVIES = "movies"
        private const val COLUMN_ID = "id"
        private const val COLUMN_NAME = "name"
        private const val COLUMN_PRICE = "price"
        private const val COLUMN_IMAGE = "image"
    }
    override fun onCreate(db: SQLiteDatabase) {
        val createTable = ("CREATE TABLE $TABLE_MOVIES (" +
                "$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT," +
                "$COLUMN_NAME TEXT," +
                "$COLUMN_PRICE REAL," +
                "$COLUMN_IMAGE INTEGER)")
        db.execSQL(createTable)
        // Insert default movies
        insertMovie(db, Movie(R.drawable.movie1, "Avatar", 100.00))
        insertMovie(db, Movie(R.drawable.movie2, "Avengers", 200.00))
        insertMovie(db, Movie(R.drawable.movie3, "Titanic", 500.00))
        insertMovie(db, Movie(R.drawable.movie4, "Star Wars", 100.00))
        insertMovie(db, Movie(R.drawable.movie5, "The Lion King", 300.00))
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
        db.execSQL("DROP TABLE IF EXISTS $TABLE_MOVIES")
        onCreate(db)
    }

    fun insertMovie(db: SQLiteDatabase, movie: Movie) {
        val values = ContentValues().apply {
            put(COLUMN_NAME, movie.name)
            put(COLUMN_PRICE, movie.price)
            put(COLUMN_IMAGE, movie.imageResId)
        }
        db.insert(TABLE_MOVIES, null, values)
    }

    fun getAllMovies(): MutableList<Movie> {
        val movies = mutableListOf<Movie>()
        val db = readableDatabase
        val cursor = db.query(TABLE_MOVIES, null, null, null, null, null, null)
        if (cursor.moveToFirst()) {
            do {
                val id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_ID))
                val name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAME))
                val price =
cursor.getDouble(cursor.getColumnIndexOrThrow(COLUMN_PRICE))
                val image = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_IMAGE))
                movies.add(Movie(image, name, price))
            } while (cursor.moveToNext())
        }
        cursor.close()
        return movies
    }
    fun updateMovie(id: Int, newName: String, newPrice: Double) {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COLUMN_NAME, newName)
            put(COLUMN_PRICE, newPrice)
        }
        db.update(TABLE_MOVIES, values, "$COLUMN_ID=?", arrayOf(id.toString()))
    }
    fun deleteMovie(id: Int) {
        val db = writableDatabase
        db.delete(TABLE_MOVIES, "$COLUMN_ID=?", arrayOf(id.toString()))
    }
}
```

dialog_edit_movie.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
    <EditText
        android:id="@+id/edit_movie_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Movie Name" />
    <EditText
        android:id="@+id/edit_movie_price"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Movie Price"
        android:inputType="numberDecimal" />
</LinearLayout>
```

dialog_insert_movie.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
    <EditText
        android:id="@+id/edit_movie_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Movie Name" />
    <EditText
        android:id="@+id/edit_movie_price"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Movie Price"
        android:inputType="numberDecimal" />
</LinearLayout>
```

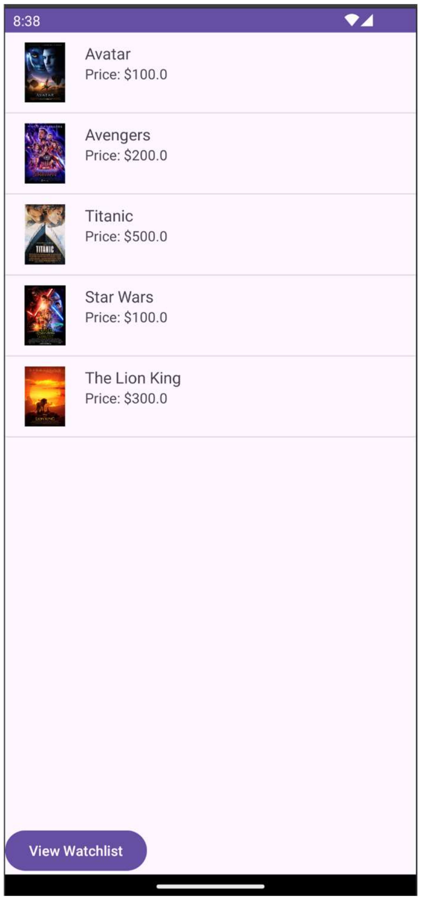
res/menu/movie_popup_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/edit" android:title="Edit" />
    <item android:id="@+id/delete" android:title="Delete" />
    <item android:id="@+id/insert" android:title="Insert" />
    <item android:id="@+id/add_to_watchlist" android:title="Add to Watchlist" />
</menu>
```

AndroidManifest.xml (added newly created activity)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Exp_3"
        tools:targetApi="31">
        <activity android:name=".WatchlistActivity"></activity>
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Output Screenshots



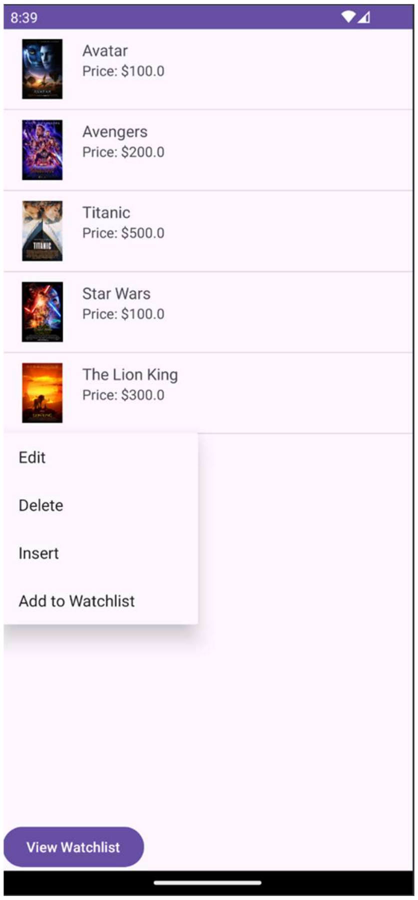
Avatar
Price: $100.0

Avengers
Price: $200.0

Titanic
Price: $500.0

Star Wars
Price: $100.0

The Lion King
Price: $300.0



Avatar
Price: $100.0

Avengers
Price: $200.0

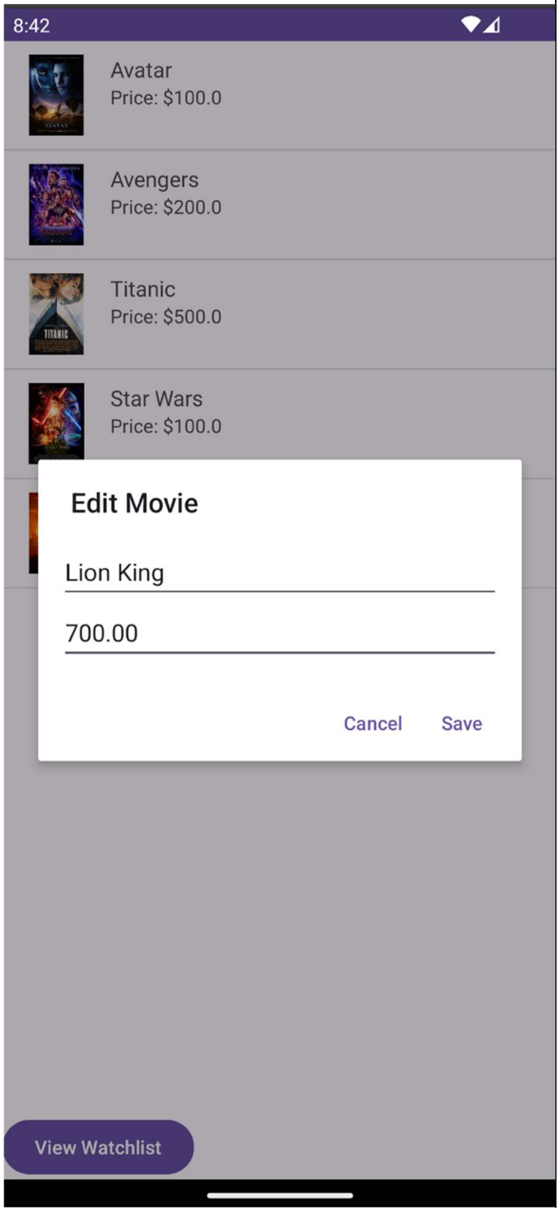
Titanic
Price: $500.0

Star Wars
Price: $100.0

The Lion King
Price: $300.0

Editing the existing Movie item:

Entering new name and price:



Avatar
Price: $100.0

Avengers
Price: $200.0

Titanic
Price: $500.0

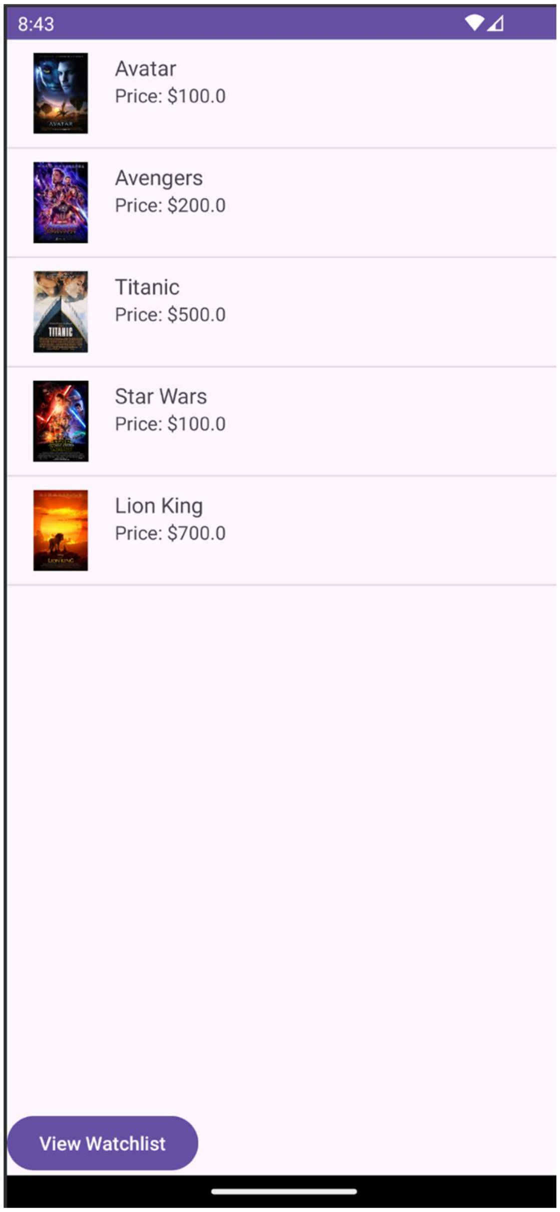
Star Wars
Price: $100.0

Edit Movie

Lion King

700.00

Movie's name and price are changed:



Avatar
Price: $100.0

Avengers
Price: $200.0

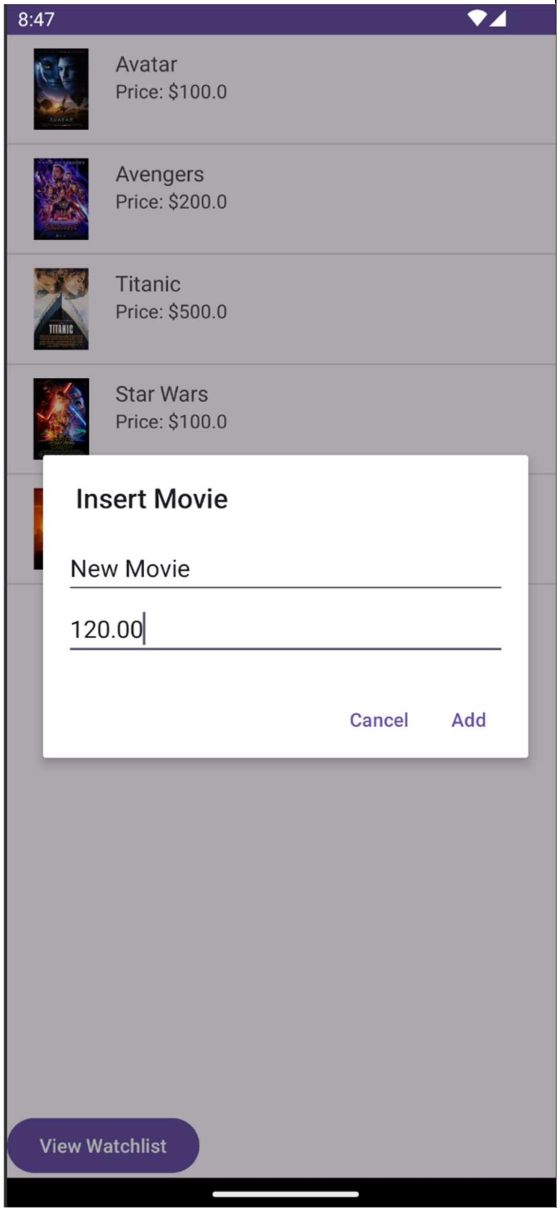
Titanic
Price: $500.0

Star Wars
Price: $100.0

Lion King
Price: $700.0

Inserting new Movie item:

Entering new movie’s name and price:



8:47

Avatar
Price: $100.0

Avengers
Price: $200.0

Titanic
Price: $500.0

Star Wars
Price: $100.0

Insert Movie

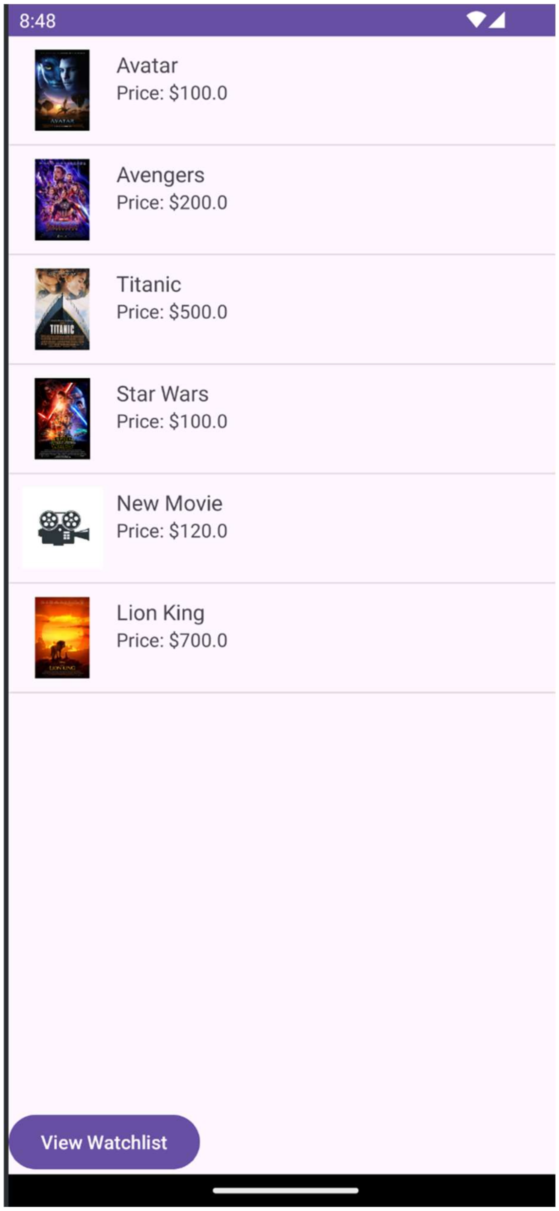
New Movie

120.00

Cancel Add

View Watchlist

New Movie being added to the list:



8:48

Avatar
Price: $100.0

Avengers
Price: $200.0

Titanic
Price: $500.0

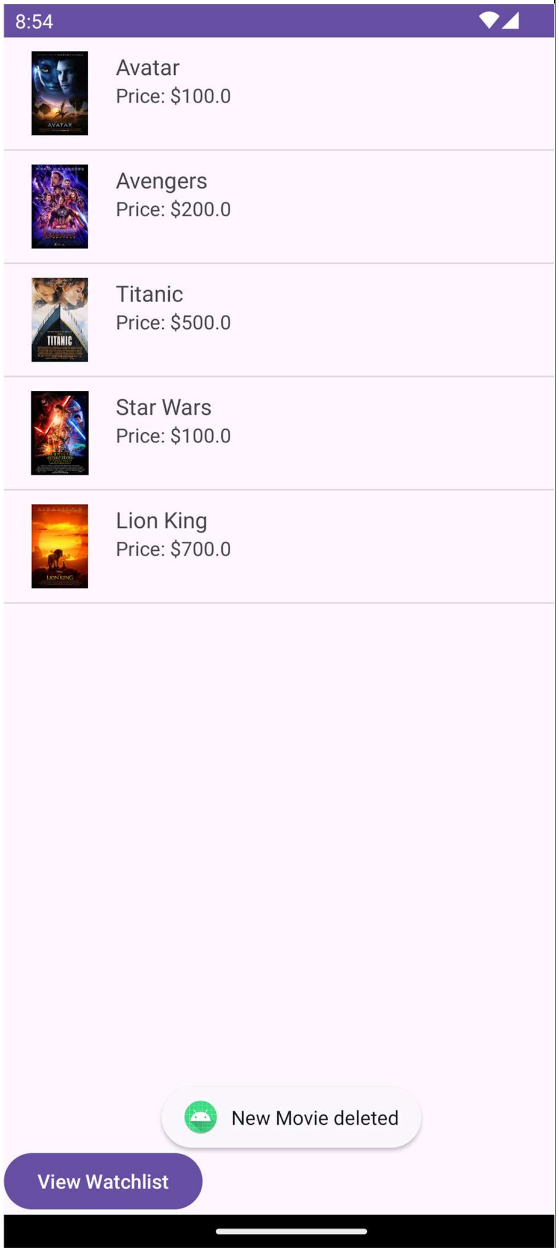
Star Wars
Price: $100.0

New Movie
Price: $120.0

Lion King
Price: $700.0

View Watchlist

Deleting newly added movie:



8:54

Avatar
Price: $100.0

Avengers
Price: $200.0

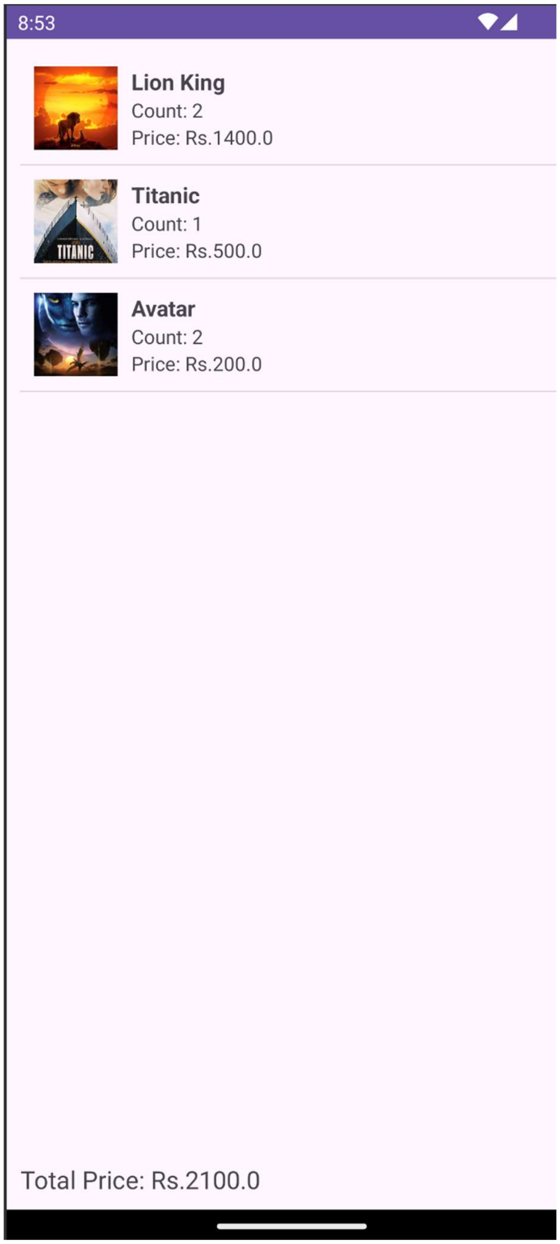
Titanic
Price: $500.0

Star Wars
Price: $100.0

Lion King
Price: $700.0

New Movie deleted

Movie Watchlist of selected movies:



8:53

Lion King
Count: 2
Price: Rs.1400.0

Titanic
Count: 1
Price: Rs.500.0

Avatar
Count: 2
Price: Rs.200.0

Total Price: Rs.2100.0

Result

The application successfully allows users to view movies, manage their watchlist, and perform actions such as editing, deleting, and inserting movies. The watchlist displays selected movies with their details, count, and total price, and the implementation has been verified to work as intended.