ANDROID PROGRAMMING
LESSON 9

ANDROID SQLITE

TABLE OF CONTENT

- What is SQLite?
- Getting Started:
  - Class SQLiteDatabase, SQLiteOpenHelper, Cursor
- Operations:
  - Create database
  - Insert, Query, Update, Delete

WHAT IS SQLITE

- Relational Database
- Embedded
- Use Structured Query Language (SQL)
- Syntax is similar to most DBMS

GETTING STARTED

Class SQLiteDatabase

- Provides interface between application code and SQLite
- Methods:
  - insert(parameters) : long
  - delete(parameters) : int
  - query(parameters) : Cursor
  - update(parameters) : int
  - execSQL(sql : String) : void
  - rawQuery(sql : String), query(sql:String) : Cursor

GETTING STARTED

Class SQLiteOpenHelper

- Helper class to “help” with operations on database
- Override methods:
  - Contructor
  - onCreate()
  - onUpgrade()

GETTING STARTED

Class Cursor

- Provides access to results of database query
- Methods:
  - moveToFirst() : boolean
  - moveToNext() : boolean
  - get<type>(columnIndex : int) : <type>
  - getColumnIndex(columnName : String) : int
  - close()

GETTING STARTED

Data Types

- NULL
- INTEGER
- REAL
- TEXT
- BLOB

OPERATION ON DATABASE

Example data

Model

Items
- id : int
- title : String
- category : String
- price: String
- date: String

Table items

id	title	category	price	Created date
1	Mua ao	Mua sam	300	12/03/3022
2	Mua xe	Mua sam	12000	08/02/2022
3	Tien dien t3	Tien dien	1200	05/04/2022
4	Tien nha t3	Tien nha	1600	07/04/2022

CREATE DATABASE CLASS

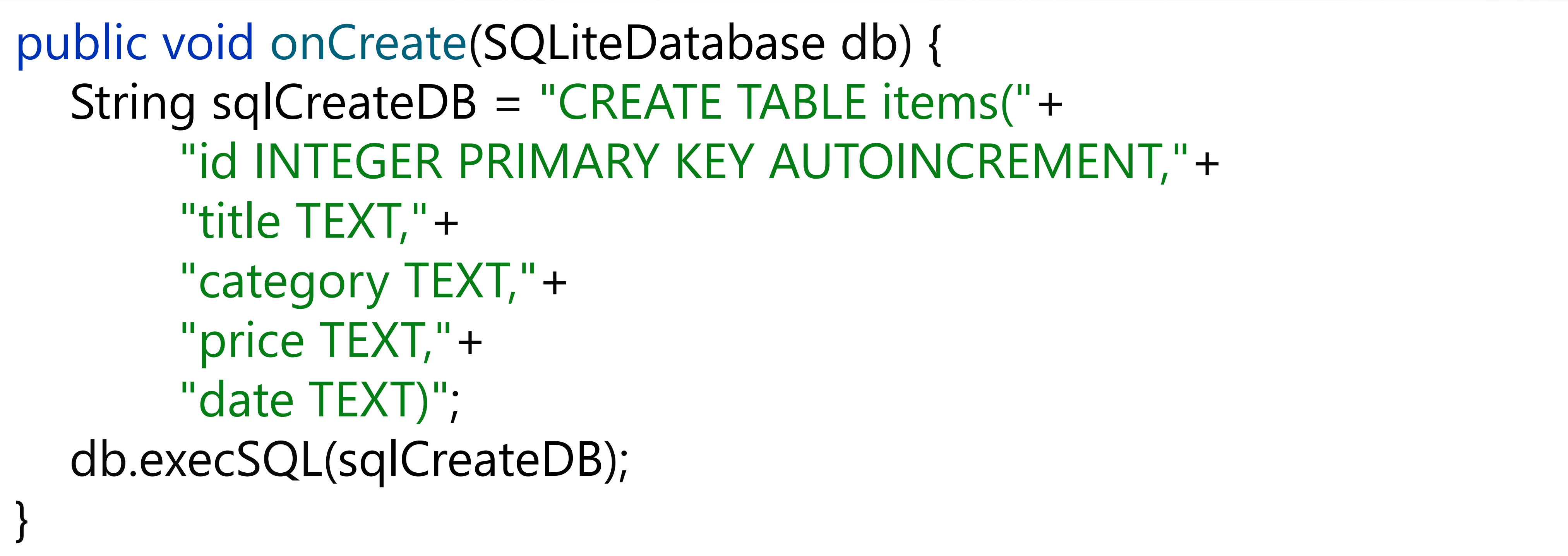
Use class **SQLiteOpenHelper** to create database

```
public class SQLiteHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "ChiTieu.db";
    private static int DATABASE_VERSION = 1;

    public SQLiteHelper(@Nullable Context context) {
        super(context,DATABASE_NAME,null,DATABASE_VERSION);
    }
```

Create Database

- When database is called the first time, run onCreate() method
- Put CREATE commands in onCreate()



```
public void onCreate(SQLiteDatabase db) {
    String sqlCreateDB = "CREATE TABLE items("+
            "id INTEGER PRIMARY KEY AUTOINCREMENT,"+
            "title TEXT,"+
            "category TEXT,"+
            "price TEXT,"+
            "date TEXT)";
    db.execSQL(sqlCreateDB);
}
```

Insert into Database (with insert)

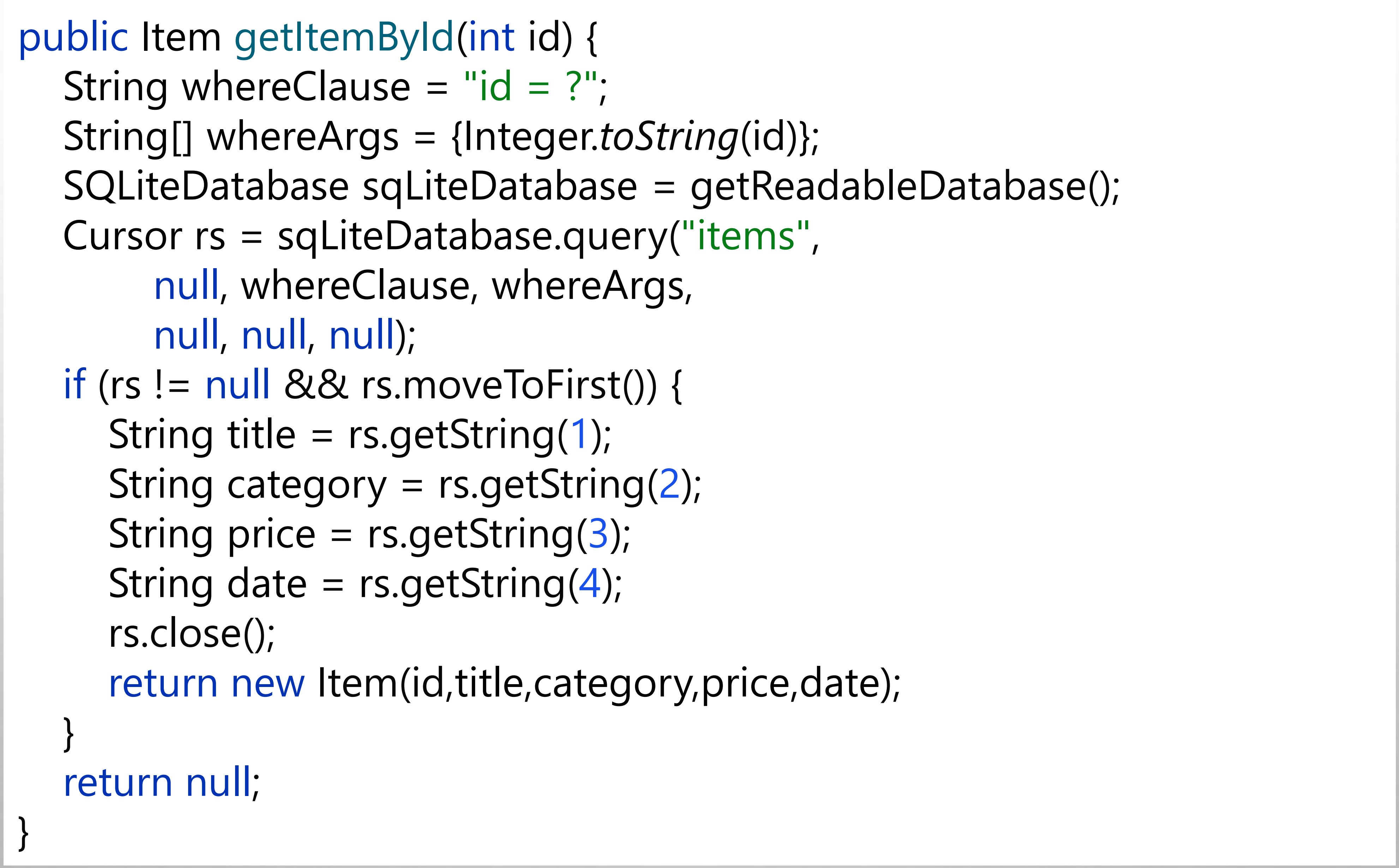
```
public void addItem(Item i){
   String sql = "INSERT INTO items(title,category,price,date)"+
         "VALUES(?,?,?,?)";
   String[] args = {i.getTitle(), i.getCategory(), i.getPrice(), i.getDate()};
   SQLiteDatabase st = getWritableDatabase();
   st.execSQL(sql,args);
}

public long addItem(Item i){
   ContentValues values = new ContentValues();
   values.put("title", i.getTitle());
   values.put("category", i.getCategory());
   values.put("price", i.getPrice());
   values.put("date", i.getDate());
   SQLiteDatabase sqLiteDatabase = getWritableDatabase();
   return sqLiteDatabase.insert("items",null, values);
}
```

ContentValues

- ContentValues
  - Map value with column name
  - Ex: INSERT INTO student(id,name,gender,mark) VALUES(1, ‘Tung’, true,8)
    - ContentValues values = new ContentValues();
    - values.put(“id”, “1”);
    - values.put(“name”, ”Tung”);
    - values.put(“gender”, true);
    - values.put(“mark”, 8);
    - sqLiteDatabase.insert(student, null, values);

Query Database (Method 1 - query)



```java
public Item getItemById(int id) {
    String whereClause = "id = ?";
    String[] whereArgs = {Integer.toString(id)};
    SQLiteDatabase sqLiteDatabase = getReadableDatabase();
    Cursor rs = sqLiteDatabase.query("items",
            null, whereClause, whereArgs,
            null, null, null);
    if (rs != null && rs.moveToFirst()) {
        String title = rs.getString(1);
        String category = rs.getString(2);
        String price = rs.getString(3);
        String date = rs.getString(4);
        rs.close();
        return new Item(id,title,category,price,date);
    }
    return null;
}
```

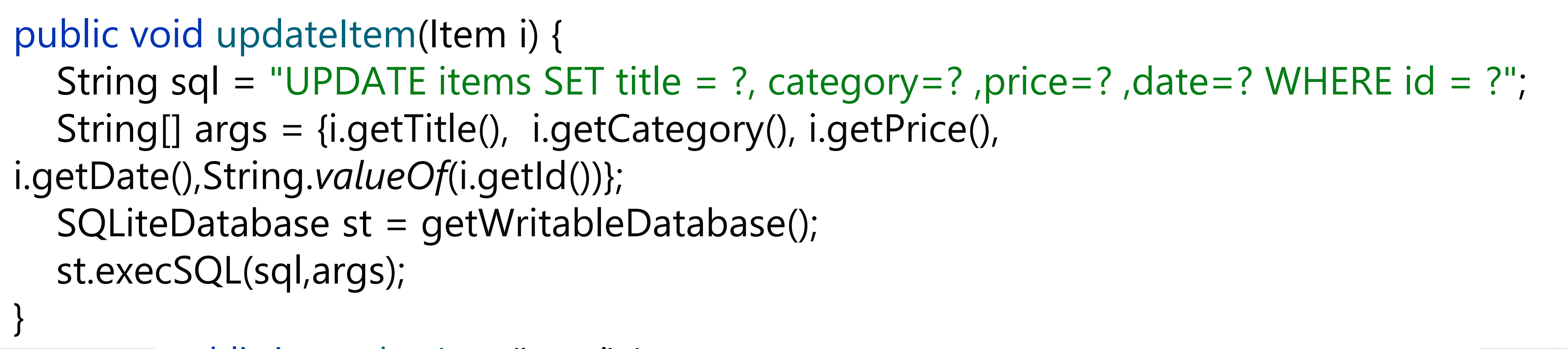
Query Database (Method 1 – query)

- sqLiteDatabase.query(table, columns[], where, whereArgs[], groupBy, having, orderBy)
    - table – name of the table to query (Ex: “student”)
    - column[] – string array with columns to query (Ex: {“id”, “name”})
    - where – conditions (Ex: “id = ?”, “name = ?”, “name LIKE %?%”)
    - whereArgs[] – arguments for where condition, replace “?”
    - groupBy
    - having
    - orderBy

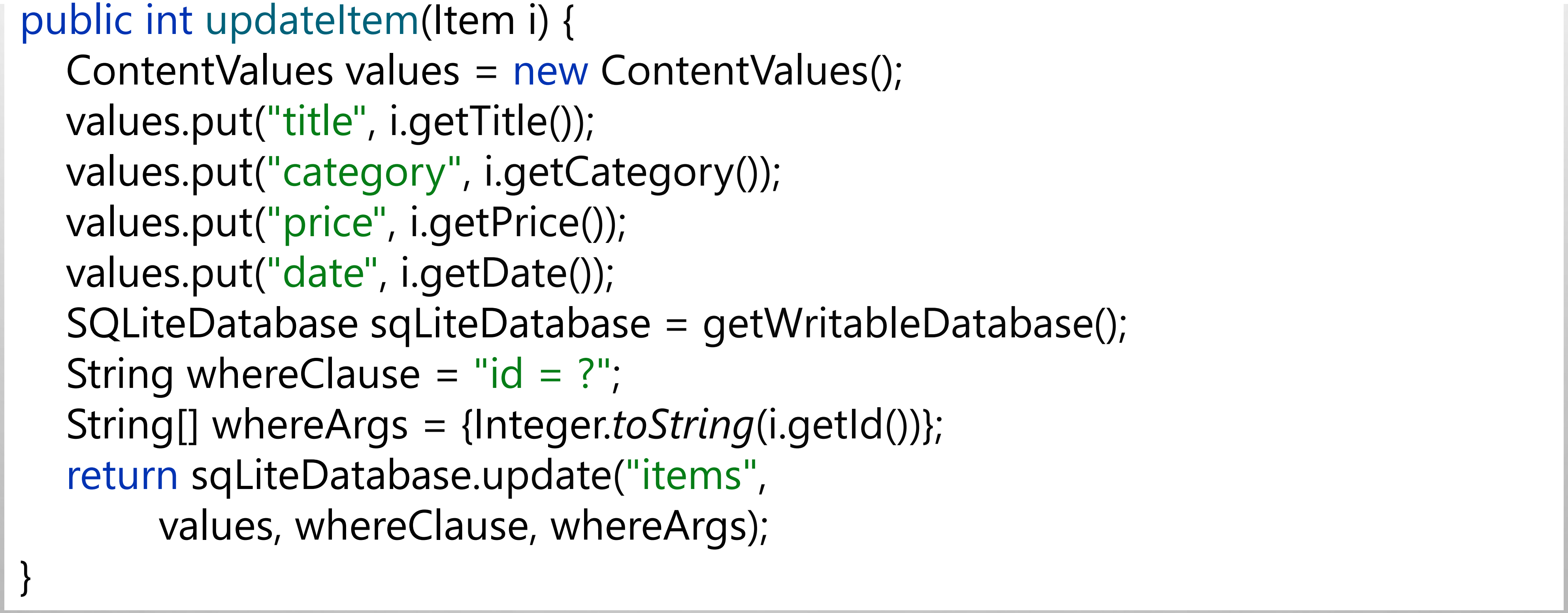
Query Database (Method 1 – query)
Get all items from database

```
public List<Item> getAll() {
    List<Item> list = new ArrayList<>();
    SQLiteDatabase sqLiteDatabase = getReadableDatabase();
    String order = "date DESC";
    Cursor rs = sqLiteDatabase.query("items",
            null, null, null,
            null, null, order);
    while ((rs != null) && (rs.moveToNext())) {
        int id= rs.getInt(0);
        String title = rs.getString(1);
        String category = rs.getString(2);
        String price = rs.getString(3);
        String date = rs.getString(4);
        list.add(new Item(id,title,category,price,date));
    }
    return list;
}
```

Update Database



```
public void updateItem(Item i) {
    String sql = "UPDATE items SET title = ?, category=? ,price=? ,date=? WHERE id = ?";
    String[] args = {i.getTitle(),  i.getCategory(), i.getPrice(),
i.getDate(),String.valueOf(i.getId())};
    SQLiteDatabase st = getWritableDatabase();
    st.execSQL(sql,args);
}
```



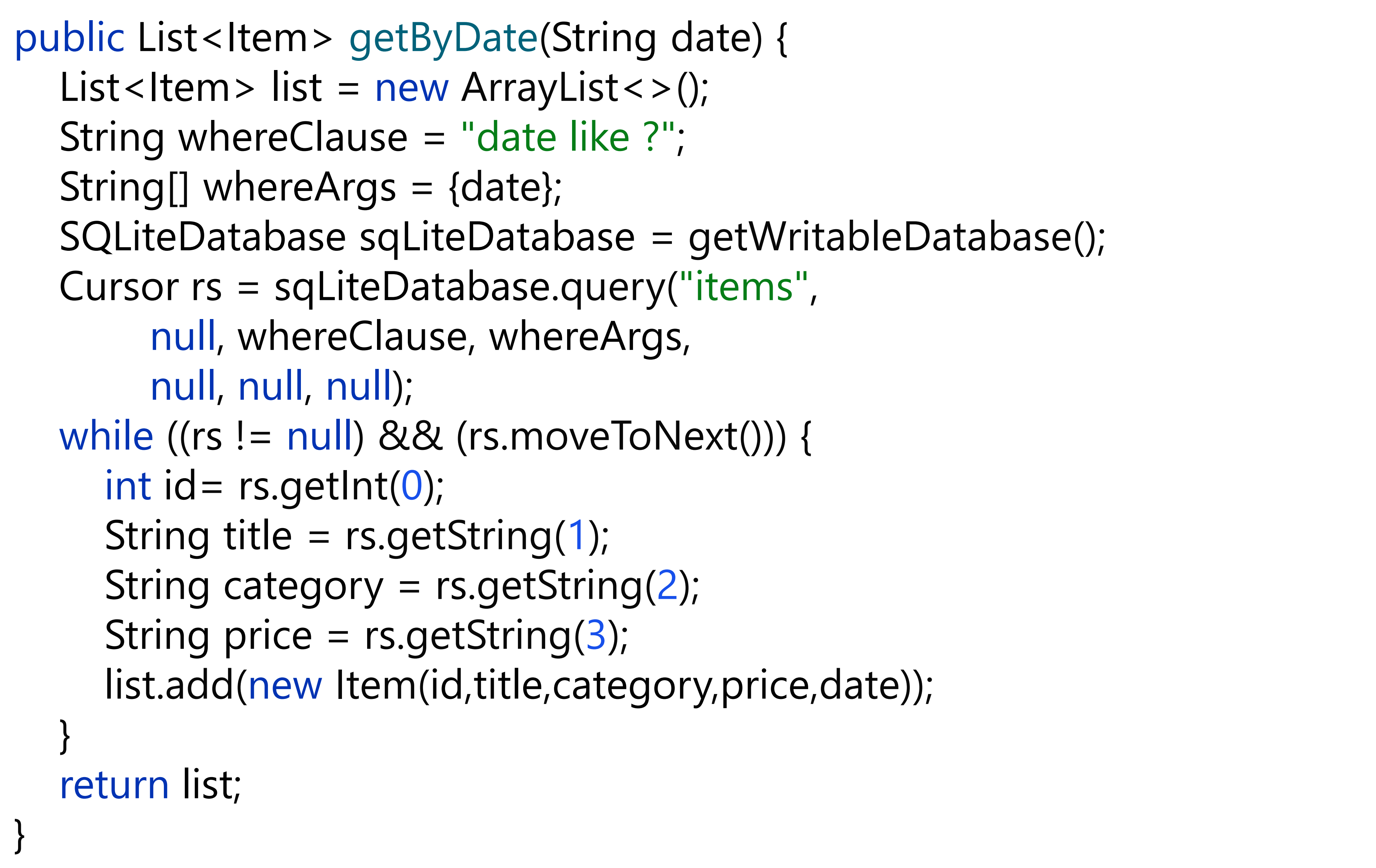
```
public int updateItem(Item i) {
    ContentValues values = new ContentValues();
    values.put("title", i.getTitle());
    values.put("category", i.getCategory());
    values.put("price", i.getPrice());
    values.put("date", i.getDate());
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    String whereClause = "id = ?";
    String[] whereArgs = {Integer.toString(i.getId())};
    return sqLiteDatabase.update("items",
            values, whereClause, whereArgs);
}
```

Delete from Database

```
public void deleteItem(int id){
    String sql = "DELETE FROM items WHERE id = ?";
    String[] args = {Integer.toString(id)};
    SQLiteDatabase st = getWritableDatabase();
    st.execSQL(sql, args);
}
```

```
public int deleteItem(int id) {
    String whereClause = "id = ?";
    String[] whereArgs = {Integer.toString(id)};
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    return sqLiteDatabase.delete("items",
            whereClause, whereArgs);
}
```

GETBY DATE



```
public List<Item> getByDate(String date) {
    List<Item> list = new ArrayList<>();
    String whereClause = "date like ?";
    String[] whereArgs = {date};
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    Cursor rs = sqLiteDatabase.query("items",
            null, whereClause, whereArgs,
            null, null, null);
    while ((rs != null) && (rs.moveToNext())) {
        int id= rs.getInt(0);
        String title = rs.getString(1);
        String category = rs.getString(2);
        String price = rs.getString(3);
        list.add(new Item(id,title,category,price,date));
    }
    return list;
}
```

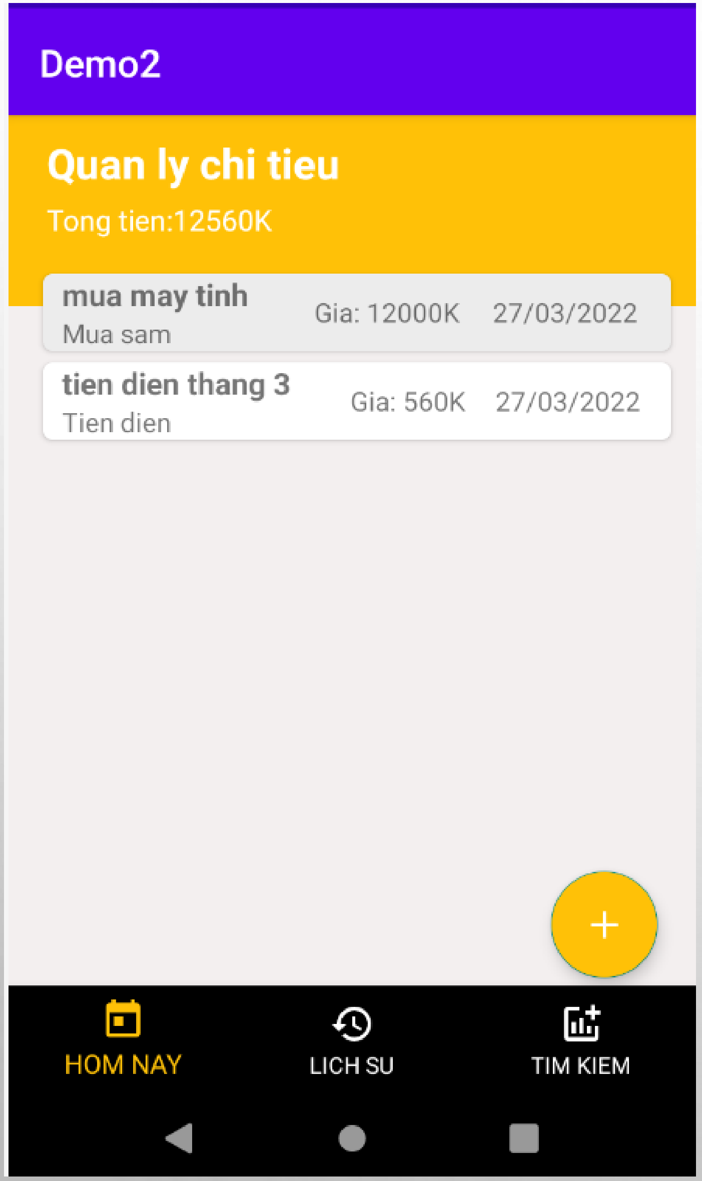
GETBY DATE FROM.. TO..

```
public List<Item> getByDateFromTo(String from,String to) {
    List<Item> list = new ArrayList<>();
    String whereClause = "date BETWEEN ? AND ?";
    String[] whereArgs = { from.trim(),to.trim()};
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    Cursor rs = sqLiteDatabase.query("items",
            null, whereClause, whereArgs,
            null, null, null);
    while ((rs != null) && (rs.moveToNext())) {
        int id= rs.getInt(0);
        String title = rs.getString(1);
        String category = rs.getString(2);
        String price = rs.getString(3);
        String date = rs.getString(4);
        list.add(new Item(id,title,category,price,date));
    }
    return list;
}
```

SEARCH BY KEY

```
public List<Item> searchByTitle(String key) {
    List<Item> list= new ArrayList<>();
    String whereClause = "title like ?";
    String[] whereArgs = {"%"+key+"%"};
    SQLiteDatabase sqLiteDatabase = getWritableDatabase();
    Cursor rs = sqLiteDatabase.query("items",
            null, whereClause, whereArgs,
            null, null, null);
    while ((rs != null) && (rs.moveToNext())) {
        int id= rs.getInt(0);
        String title = rs.getString(1);
        String category = rs.getString(2);
        String price = rs.getString(3);
        String date = rs.getString(4);
        list.add(new Item(id,title,category,price,date));
    }
    return list;
}
```

VÍ DỤ

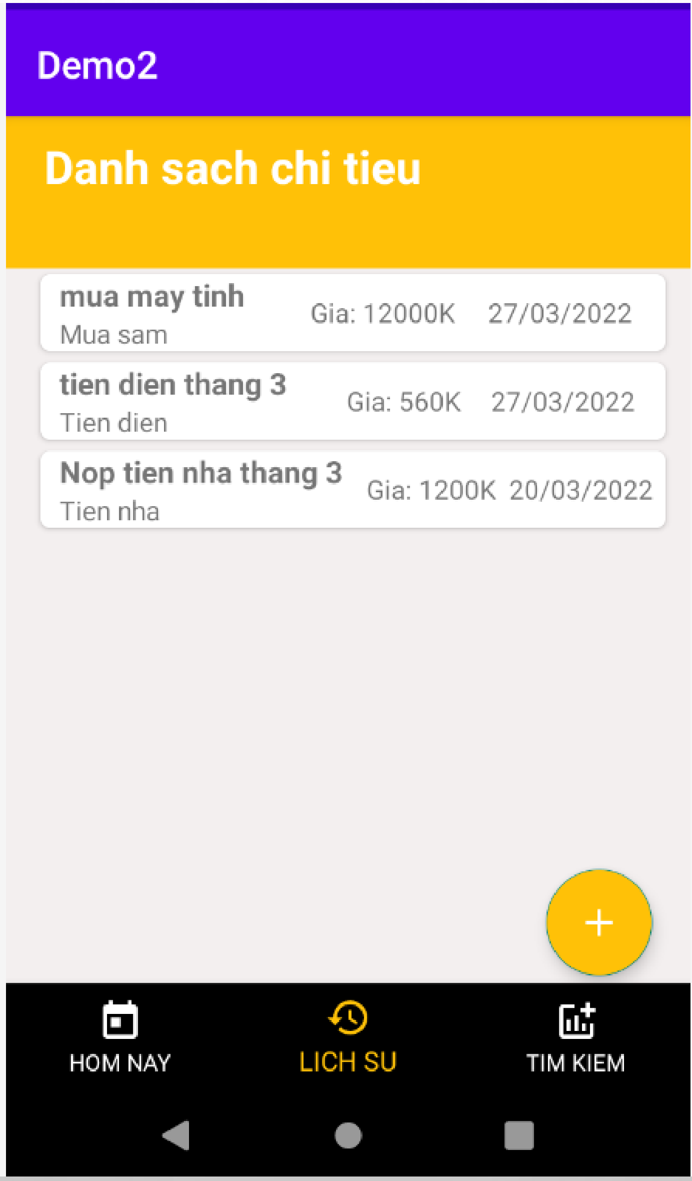


Demo2

Quan ly chi tieu

Tong tien:12560K

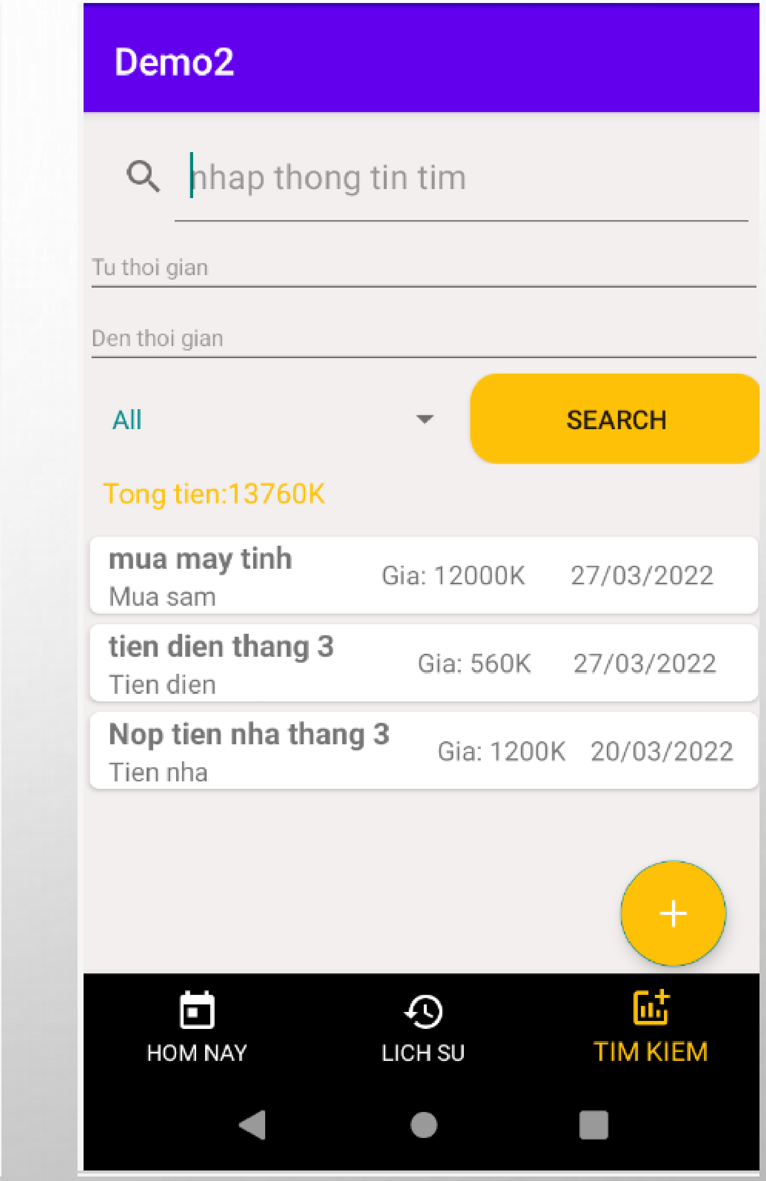
mua may tinh Mua sam	Gia: 12000K	27/03/2022
tien dien thang 3 Tien dien	Gia: 560K	27/03/2022



Demo2

Danh sach chi tieu

mua may tinh Mua sam	Gia: 12000K	27/03/2022
tien dien thang 3 Tien dien	Gia: 560K	27/03/2022
Nop tien nha thang 3 Tien nha	Gia: 1200K	20/03/2022



Demo2

nhap thong tin tim

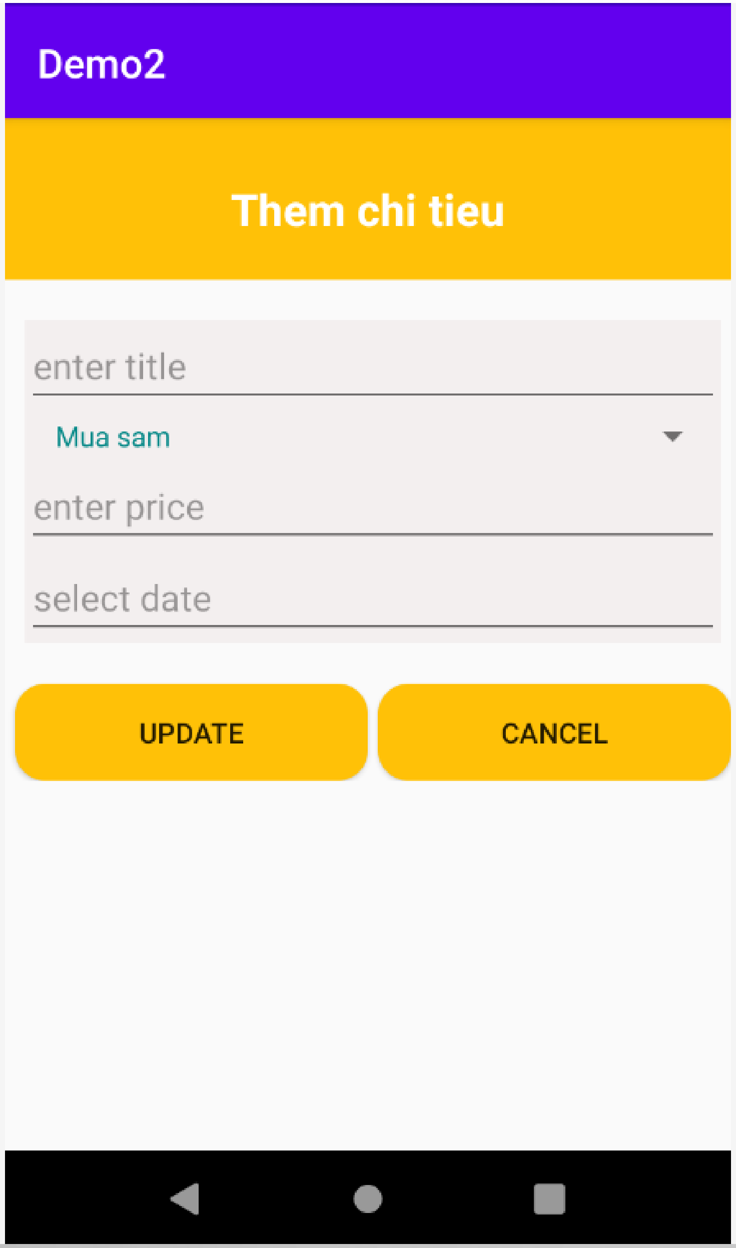
Tu thoi gian

Den thoi gian

All

Tong tien:13760K

mua may tinh Mua sam	Gia: 12000K	27/03/2022
tien dien thang 3 Tien dien	Gia: 560K	27/03/2022
Nop tien nha thang 3 Tien nha	Gia: 1200K	20/03/2022



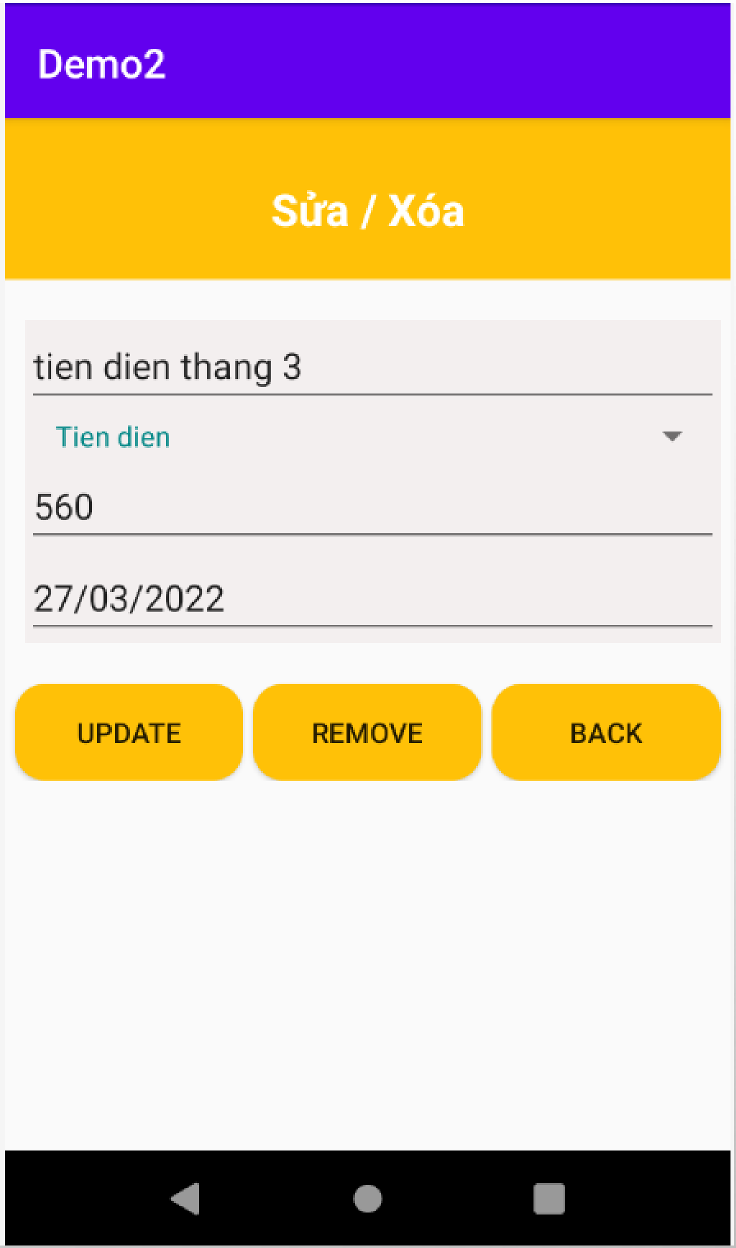
Them chi tieu

enter title

Mua sam

enter price

select date



Sửa / Xóa

tien dien thang 3

Tien dien

560

27/03/2022