



Tribhuvan University

Faculty of Humanities and Social Sciences

Kirtipur, Nepal

Mini Project of Inventory Management App [Mobile Programming (CACS 351)]

*For the partial fulfillment for the Bachelor in Computer Application
under Tribhuvan University*

Submitted By

Name: Prama Pandit

TU Registration No: 6-2-240-313-2018

Submitted To

Saptagandaki Multiple Campus

Faculty of Humanities and social sciences

Affiliated To Tribhuwan University

Bharatpur-10, Chitwan

May, 2022

List of Figure

Fig 1.1	Product List	11
Fig 1.2	Add New Product Details	11
Fig 1.3	View Product Details	12
Fig 1.4	Update Product Details	12
Fig 1.5	Delete Product Details	13
Fig 1.6	Add Product Movement	13

Table of Contents

1. INTRODUCTION	1
1.1 Introduction to Inventory Management System	1
1.2 Problem of Statement	1
1.3 Features of Project.....	2
1.4 Scope of the Application	2
2. ANALYSIS AND DESIGN	3
2.1 Background Research.....	3
2.2 Requirement Analysis	3
2.3 Feasibility Analysis	3
2.1.1 Economic Feasibility	3
2.2.2 Technical Feasibility	4
2.3.3 Operational Feasibility	4
2.4 Source Code	4
2.5 UI Interface.....	11
3. CONCLUSION	14

1. INTRODUCTION

1.1 Introduction to Inventory Management System

The project Inventory Management System is a complete Mobile based application designed on java technology using Android Studio. The main aim of the project is to develop Inventory Management System Model software in which all the information regarding the stock of the organization will be presented.

We can manage Store, Bill, Product Quality, Customer, Stock and Product from this project. The main objective to develop Android project on Stock Management System is to provide android app on Inventory Management System to customer, from where user can use it from his mobile device. Android project on Stock Management System is compatible with all android mobiles. So user can install and configure it on their mobile devices, if you have any specific requirements then, we can develop major android projects on Stock Management System according to requirements.

We have implemented advance search feature for searching records on various criteria for Bill, Store, Stock also admin can perform Create, read, update and delete (CRUD) operations on Product Quality, Customer, Product. In this project all the modules like Store, Customer, and Bill are tightly coupled and we can track the information easily.

1.2 Problem of Statement

After analyzing many existing IMS we have now the obvious vision of the project to be developed. Before we started to build the application team had many challenges. We defined our problem statement as:

- i. To make desktop based application of IMS for small organization.
- ii. To make the system easily managed and can be secured.
- iii. To cover all the areas of IMS like purchase details, sales details and stock management.

1.3 Features of Project

This application is used to show the stock remaining and details about the sales and purchase. It gives the details about the stock on daily based and weekly based. The details components are described below:

View All Product List:

We open the app and we see product details list.

Add Product:

We can add new product Details. if we need to extend or we have more than one product. We can add the new product along with the date.

Sales details:

It shows the details about the sales and the remaining stock of sales. It also show the details about the sales in return.

Purchase details:

It shows the details about the purchase made by the organization along with the price and dates.

1.4 Scope of the Application

Inventory Management System (IMS) is targeted to the small or medium organization **which doesn't have many** Product or warehouses i.e. only to those organization that has single power of authority. Some of the scopes are:

- i. Only one person is responsible in assigning the details or records
- ii. It is security driven.
- iii. Product can be added as per the requirement.

2. ANALYSIS AND DESIGN

2.1 Background Research

We started research by identifying the need of IMS in the organization. Initially we bounded our research to find the general reasons that emerged the needs of Inventory Management System. We used different techniques to collect the data that can clearly give us the overall image of the application. The techniques we used were interview with the developers, visiting online websites that are presented as the templates and visiting some organization to see their IMS application. Basically the following factors forced us to develop IMS application:

- i. Cost and affordability
- ii. Lack of stock management.
- iii. Effective flow of stock transfer and management.
- iv. Difficulty in monitoring the stock management.

2.2 Requirement Analysis

We collected a number of requirements for project from our primitive research, website visits, and interview to the concerned personnel and their experiences regarding the concepts of its development. We have even visited some organization in Kathmandu valley and analyze its importance and try to develop the project by fulfilling all the weakness that were found in the application. We then decided to build same type of application with different logic flow and new language which will be suitable for the small organization.

2.3 Feasibility Analysis

This software has been tested for various feasibility criterions from various point of views.

2.1.1 Economic Feasibility

The system is estimated to be economically affordable. The system is medium scale desktop application and has affordable price. The benefits include increased efficiency,

effectiveness, and the better performance. Comparing the cost and benefits the system is found to be economically feasible.

2.2.2 Technical Feasibility

Development of the system requires tools like:

- i. Android Studio
- ii. Java Programming
- iii. SQLite Database

Which are easily available within the estimated cost and schedule.

2.3.3 Operational Feasibility

The system provides better solution to the libraries by adding the typical requirement and necessities. The solution provided by this system will be acceptable to ultimate solution for the stock management.

2.4 Source Code

AddProduct.java

```
package com.example.inventorymanagement;

import java.util.ArrayList;
import java.util.List;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase;

public class DBManager extends SQLiteOpenHelper {

    static final String dbName = "product_db";
```

```

// PRODUCT TABLE
static final String product_table = "product";
static final String card_col = "card_no";
static final String model_col = "model_no"; // primary key
static final String price_buy_col = "buy_price";
static final String price_sell_col = "sell_price";
static final String description_col = "description";
static final String place_col = "place";
static final String product_name_col = "product_name";
static final String quantity_col = "quantity";

// LOG TABLE
static final String product_log_table = "product_log";
static final String product_id_col = "product_id";
static final String customer_col = "customer";
static final String quantity_change_col = "quantity";
static final String sell_or_buy_col = "SoB";
static final String created_at_col = "created_at";

public DBManager(Context context){
    super(context, dbName , null, 1);
}

@Override
public void onCreate(SQLiteDatabase db) {

    // PRODUCT TABLE
    db.execSQL("CREATE TABLE " +product_table+ "(" +model_col+ "
VARCHAR(255), " +
        card_col+ " VARCHAR(255) NOT NULL PRIMARY KEY, "
+price_buy_col+ " DECIMAL(10,5) NOT NULL, " +
        price_sell_col+ " DECIMAL(10,5) NOT NULL, " +description_col+ "
VARCHAR(255), " +

```



```

        place_col+ " VARCHAR(255) NOT NULL, " +product_name_col+ "
VARCHAR(255) NOT NULL, " +
        quantity_col+ " INTEGER DEFAULT 0);");

// LOG TABLE
db.execSQL("CREATE TABLE " +product_log_table+ "(" +product_id_col+ "
INTEGER PRIMARY KEY AUTOINCREMENT, "+
        customer_col+ " VARCHAR(255), " +created_at_col+ " DATE DEFAULT
CURRENT_DATE, " +
        quantity_change_col+ " INTEGER NOT NULL, "+
        sell_or_buy_col+ " VARCHAR(1) NOT NULL, " +card_col+ "
VARCHAR(255) NOT NULL, "+
        "FOREIGN KEY (" +card_col+") REFERENCES "+product_table+
("+"card_col+"));");

```

```

}

```

@Override

```

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS "+product_log_table);
    db.execSQL("DROP TABLE IF EXISTS "+product_table);
    onCreate(db);
}

```

@Override

```

public void onOpen(SQLiteDatabase db) {
    super.onOpen(db);
    if(!db.isReadOnly()){
        //Enable foreign key constraints
        db.execSQL("PRAGMA foreign_keys=ON;");
    }
}

```

```

// Register a new product to inventory

```

```

    public int addNewProduct(String model_no, String card_no, Double p_buy, Double
p_sell, String place, String desc, String p_name, int qnt){

        try {
            SQLiteDatabase db = this.getWritableDatabase();
            db.execSQL("INSERT INTO " +product_table+ "(" +model_col+ "," +card_col+
"," +price_buy_col+ "," +
                price_sell_col+ "," +place_col+ "," +description_col+ ","
+product_name_col+ "," +quantity_col+ ") VALUES("+""" +model_no+ """," +card_no+
"""," +p_buy+ "," +
                p_sell+ """," +place+ """," +desc+ """," +p_name+ """," +qnt+ ");");
            db.close();
            return 1; // success
        }catch (SQLException e){
            e.printStackTrace();
            return 0; // fail
        }
    }
}

```

```

// Update a single Product in inventory
    public int updateProduct(String model_no, String card_no, Double p_buy, Double
p_sell, String place, String desc, String p_name, int q) {
        try {
            SQLiteDatabase db = this.getWritableDatabase();
            db.execSQL("UPDATE " +product_table+ " SET "+model_col+ "="" +model_no+
"" , " +quantity_col+ "="" +q+ " , " +price_sell_col+ "="" +p_sell+ " , " +
                price_buy_col+ "="" +p_buy+ " , " +description_col+ "="" +desc+ "" , "
+place_col+ "="" +place+ "" , " +product_name_col+ "="" +p_name+ "" WHERE " +
                card_col+ "="" +card_no+ """);
            db.close();
            return 1; // success
        }catch (SQLException e){
            e.printStackTrace();
            return 0; // fail
        }
    }
}

```

```

    }
}

// Delete product from inventory
public int deleteProduct(String card_no){
    try {
        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL("DELETE FROM " +product_log_table+ " WHERE " +card_col+
"=" +card_no+ ";""); // delete from log table
        db.execSQL("DELETE FROM " +product_table+ " WHERE " +card_col+ "="
+card_no+ ""); // delete from product table
        db.close();
        return 1; // success
    }catch (SQLException e){
        e.printStackTrace();
        return 0; // fail
    }
}

// Sell or Buy Product
public int sellorbuyProduct(String card_no, String customer, String sob, int quantity){
    try {
        SQLiteDatabase db = this.getWritableDatabase();
        String selectQuery = "SELECT * FROM " +product_table+ " WHERE "
+card_col+ "=" +card_no+"";
        Cursor c = db.rawQuery(selectQuery,null);
        int n = 0 ;
        if(c.getCount() > 0){
            c.moveToFirst();
            n = c.getInt(7);
            c.close();
        }

        if(sob.equals("s")){

```

```

        if(n < quantity){
            db.close();
            return 0; // fail getColumnIndex
        }
        n = n-quantity;
        db.execSQL("UPDATE " +product_table+ " SET " +quantity_col+ "=" +n+ "
WHERE " +card_col+ "=" +card_no+ """);
        db.execSQL("INSERT INTO " +product_log_table+ "(" +customer_col+ ", "
+card_col+ ", " +
            quantity_change_col+ ", " +sell_or_buy_col+ ") VALUES(" +customer+
            ", " +card_no+ ", " +quantity+ ", 's')");

    }
    else if(sob.equals("b")){
        n = n+quantity;
        db.execSQL("UPDATE " +product_table+ " SET " +quantity_col+ "=" +n+ "
WHERE " +card_col+ "=" +card_no+ """);
        db.execSQL("INSERT INTO " +product_log_table+ "(" +customer_col+ ", "
+card_col+ ", " +
            quantity_change_col+ ", " +sell_or_buy_col+ ") VALUES(" +customer+
            ", " +card_no+ ", " +quantity+ ", 'b')");
    }
    db.close();
    return 1; // success
} catch (SQLException e){
    e.printStackTrace();
    return 0; // fail
}
}

// Get all details about a single product
public void getProduct(Product p, String c_no){
    SQLiteDatabase db = this.getReadableDatabase();

```

```
String selectQuery = "SELECT * FROM " +product_table+ " WHERE " +card_col+
"=" +c_no+""; // change from product_name to card_no
```

```
Cursor cursor = db.rawQuery(selectQuery,null);
```

```
if(cursor.getCount(>0) {
```

```
    cursor.moveToFirst();
```

```
    p.setProduct_name(cursor.getString(6));
```

```
    p.setModel_no(cursor.getString(0));
```

```
    p.setCard_no(cursor.getString(1));
```

```
    p.setPrice_buy(cursor.getDouble(2));
```

```
    p.setPrice_sell(cursor.getDouble(3));
```

```
    p.setDescription(cursor.getString(4));
```

```
    p.setPlace(cursor.getString(5));
```

```
    p.setQuantity(cursor.getInt(7));
```

```
    cursor.close();
```

```
}
```

```
db.close();
```

```
}
```

```
// Get log cursor
```

```
public Cursor getLogCursor(String card_no){
```

```
    SQLiteDatabase db = this.getReadableDatabase();
```

```
    String selectQuery = "SELECT * FROM " +product_log_table+ " WHERE "
+card_col+ "=" +card_no+"";
```

```
    Cursor c = db.rawQuery(selectQuery,null);
```

```
    return c;
```

```
}
```

```
}
```

2.5 UI Interface

MacBook Pro		
Product No	Place	Piece
1	Bharratpur,Chit	20
Galaxy A53 5G		
Product No	Place	Piece
2	Kathmandu	50
iPhone 13 Pro		
Product No	Place	Piece
3	Hetauda	20
Redmi Note 10 Pro Max		
Product No	Place	Piece
4	Tadi	304
Samsung Z Fold		
Product No	Place	Piece
5	Nawalparasi	20

Fig 1.1: Product List

Add Product

product	Dell Inspire 13
Product No:	7
Brand/Model:	Dell
Piece:	40
Place:	Hakimchok
Purchase price	Sale price
120000	140000

Dell inspire 13 is the best flagship laptop in this category

ADD

category | category's

q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123 , .

Fig 1.2: Add New Product Details

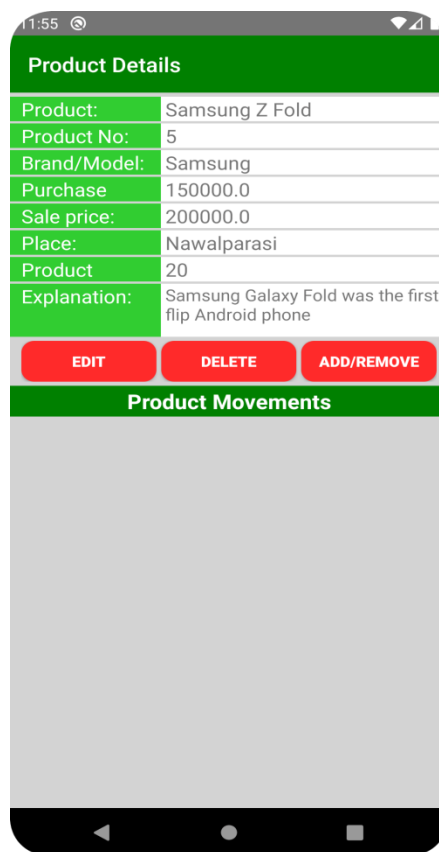


Fig 1.3: View Product Details

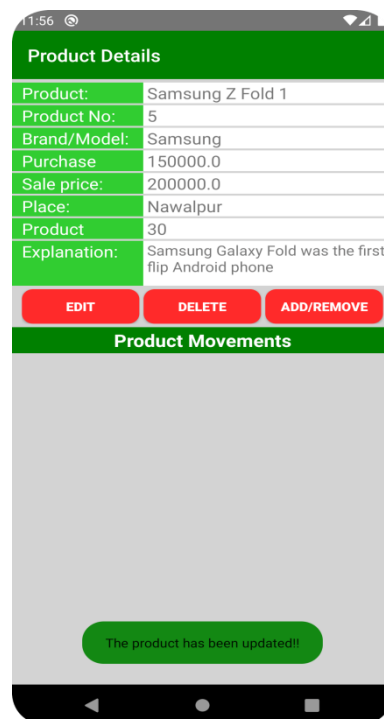


Fig 1.4: Update Product Details

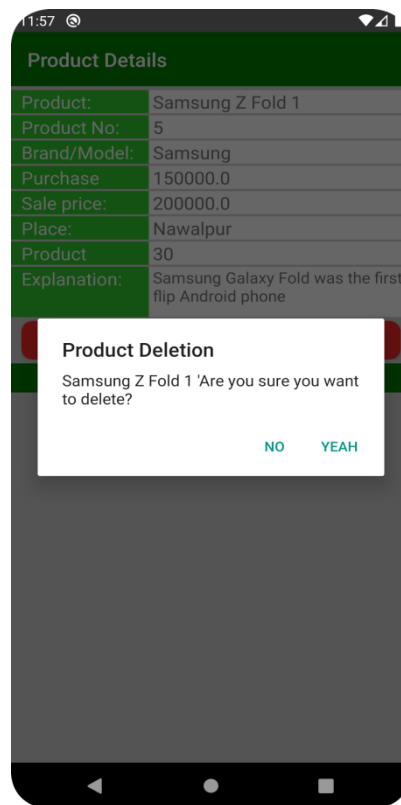


Fig 1.5: Delete Product Details

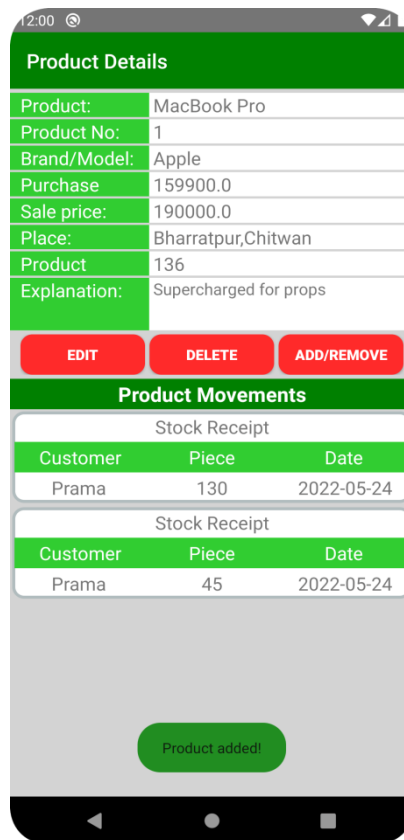


Fig 1.6: Add Product Movement

3. CONCLUSION

To conclude, Inventory Management System is a simple desktop based application basically suitable for small organization. It has every basic items which are used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. This application also provides a simple report on daily basis to know the daily sales and purchase details.

This application matches for small organization where there small limited if Product.

Through it has some limitations; our team strongly believes that the implementation of this system will surely benefit the organization.