Software Architecture Document

for

*POS Undone*

Prepared by Undone Team

Table of Contents

1. Overview........................................................................................................................... 1

1.1 Purpose................................................................................................................. 1

1.2 Architecture Representation................................................................................ 1

2. Architectural Factors......................................................................................................... 1

3. Architectural Decisions and Rationale............................................................................. 2

3.1 Design of Inventory and Stock Records............................................................. 2

3.2 Design of Process Sale....................................................................................... 2

3.3 Design of Report…………………………………………………………………... 3

4. Logical View...................................................................................................................... 4

4.1 Domain Vocabulary............................................................................................... 6

5. Data View........................................................................................................................... 7

6. Process View..................................................................................................................... 7

7. Use Case View.................................................................................................................. 8

8. Deployment View.............................................................................................................. 9

9. References........................................................................................................................ 9

Revision History

|  |  |  |
| --- | --- | --- |
| **Name** | **Date** | **Reason For Change** |
| UnDone | 3/12/2013 | start doing SAD |
| UnDone | 6/12/2013 | continue doing SAD |

# **1.** **Overview**

## 1.1 Purpose

This document is an overview of the architecture system comprehensively. So readers will get more information about architecture details and understand the design of this application. This document will include many interesting and useful information for developers such as technical details, and code fragments.

## 1.2 Architecture Representation

The architecture is described by:

**Architectural Factors:** Why we develop on android and use JAVA language to implement this application.

**Architectural Decisions:** How to design about Inventory, Process Sale ,and Report of the POS.

**Logical View:** Logical Layer, Domain class diagram, Domain model, and Domain Vocabulary

**Data View:** How database is working

**Process View:** process of UnDone POS

**Use Case View:** the requirements about Inventory, Process Sale ,and Report of this POS

**Deployment View:** What format of our POS is being

# **2.** **Architectural Factors**

The reason that we develop in mobile POS because it’s portable and we can use it in outdoor and also sending data using cellular so it can use any where and we can show the catalog to the customer immediately.

We choose to implement in android OS because android has various of devices and allow to develop application and install without permission from the store.

Also we have base in JAVA language so it is more easy to learn how to implement in android instead of iOS

# **3.** **Architectural Decisions and Rationale**

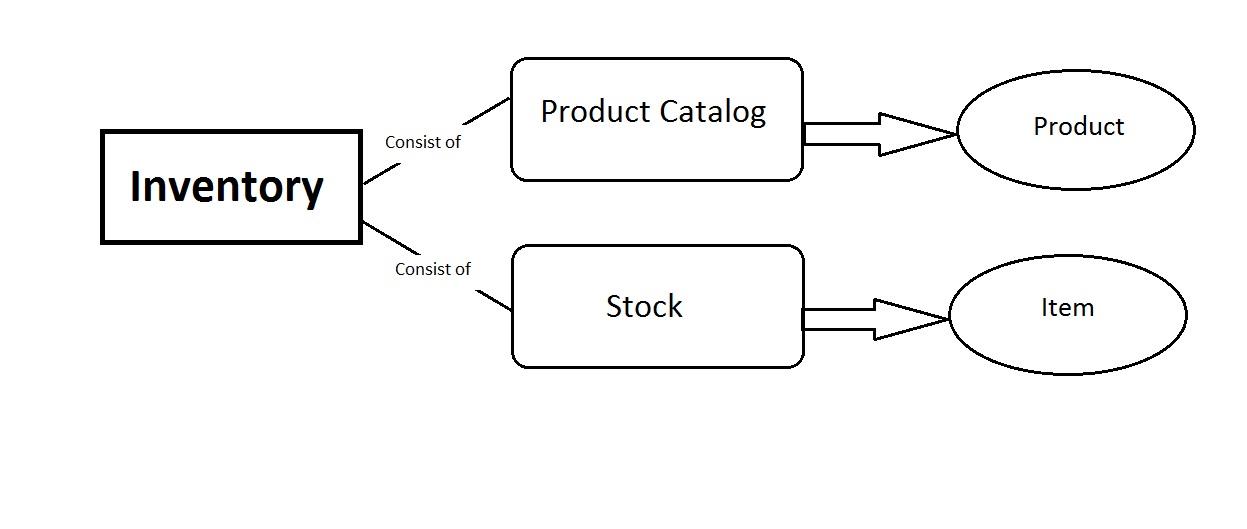
## 3.1 Design of Product Catalog and Stock Records

We separate inventory into 2 parts. They are Product Catalog and Stock. Two of these have their own database.

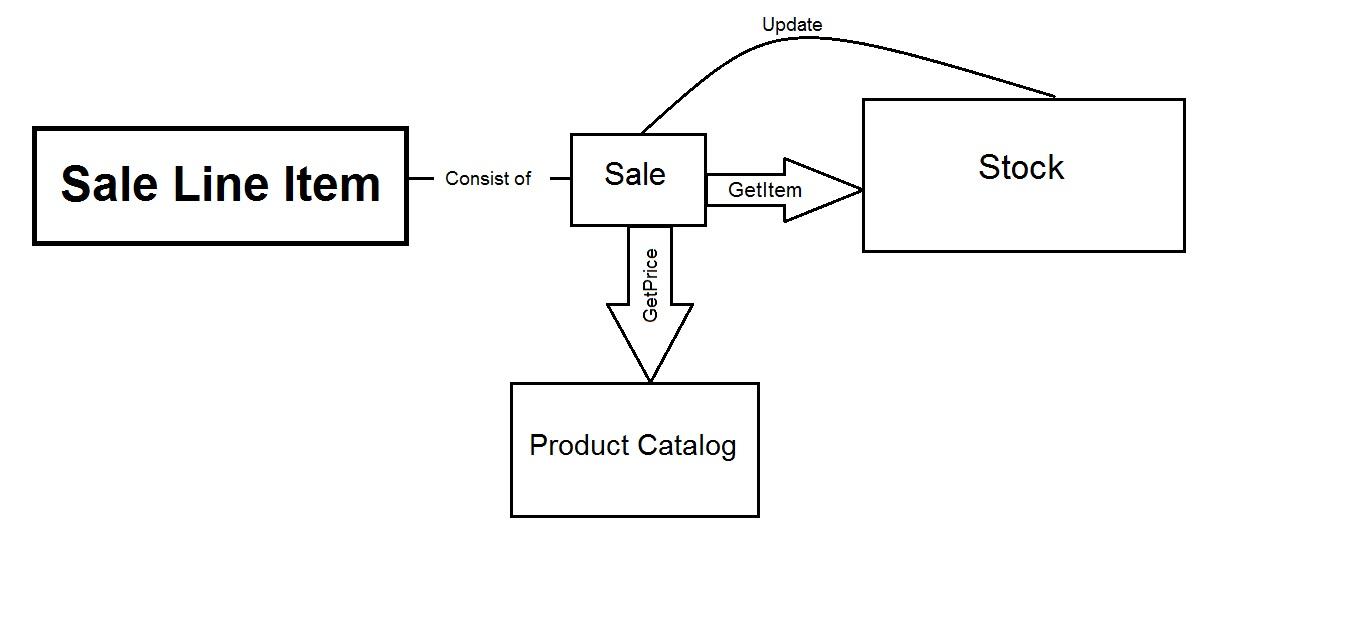
Product Catalog record : id, name, price, last edited

Stock record : id, name, cost, quantity, last edited

We can remove product out from product catalog, but we cannot remove item from stock because each item in stock represent each lots of item so it will be record permanently.

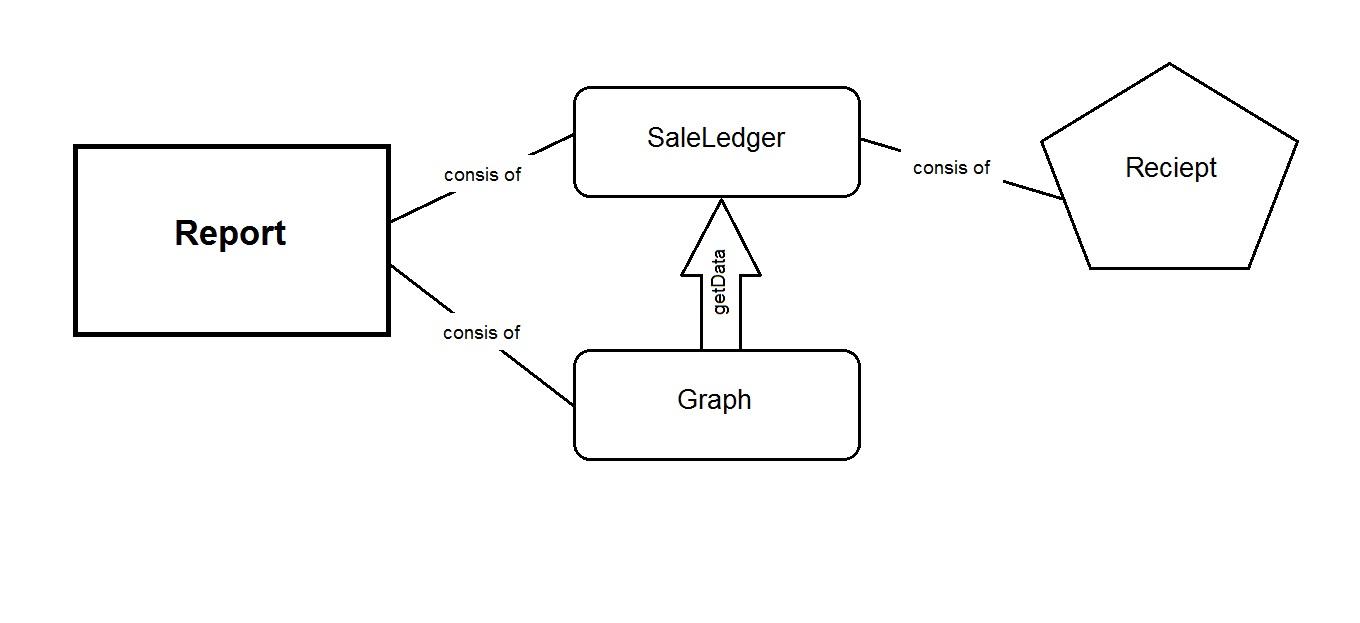


**3.2** **Design of Process Sale**

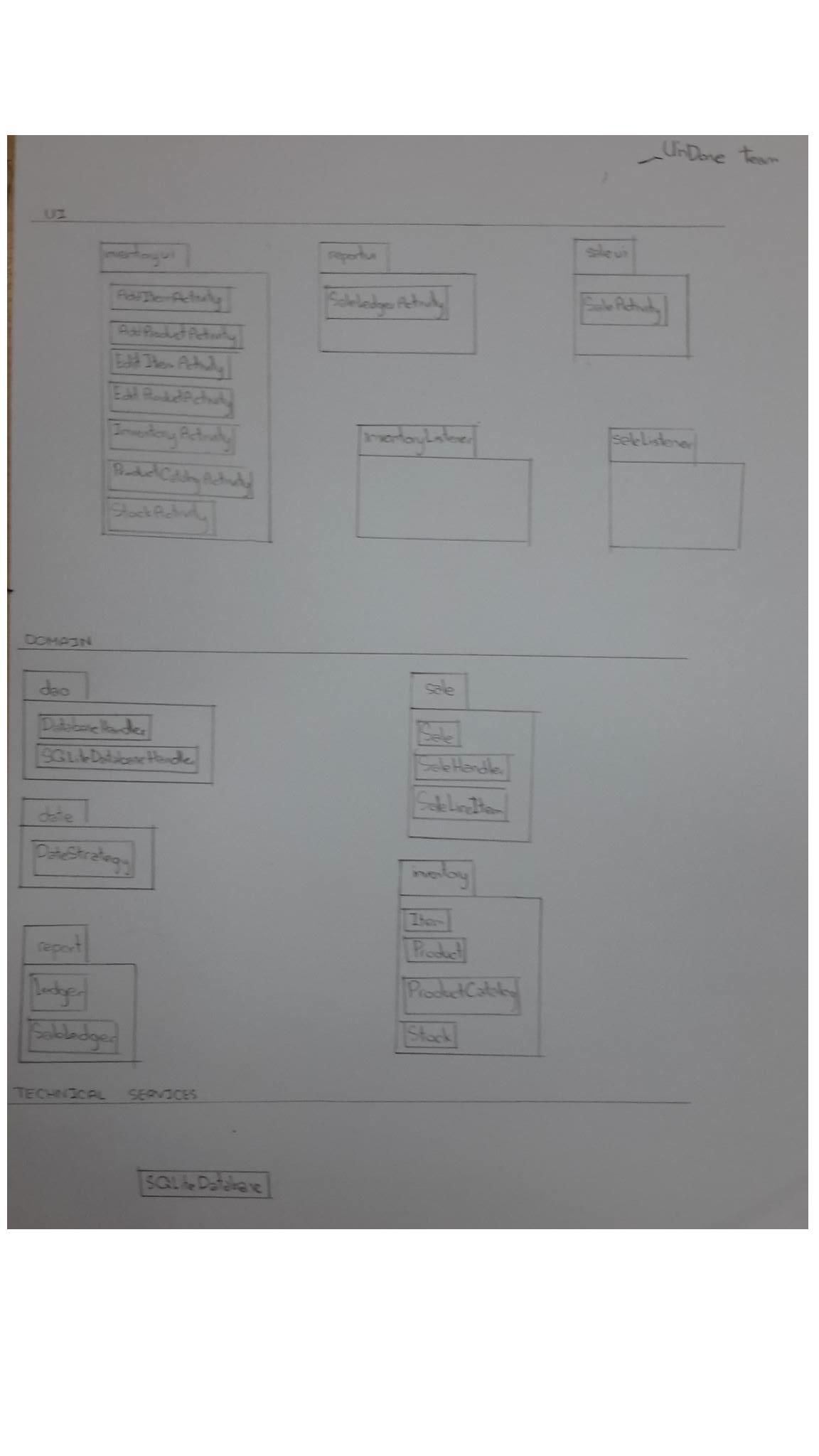
In process sale, when we add item, item will be added to Sale LineItem. After finish making sale, item will be decrease from stock. The item price will get from product catalog. In calculate total price, the system is already include calculate tax.

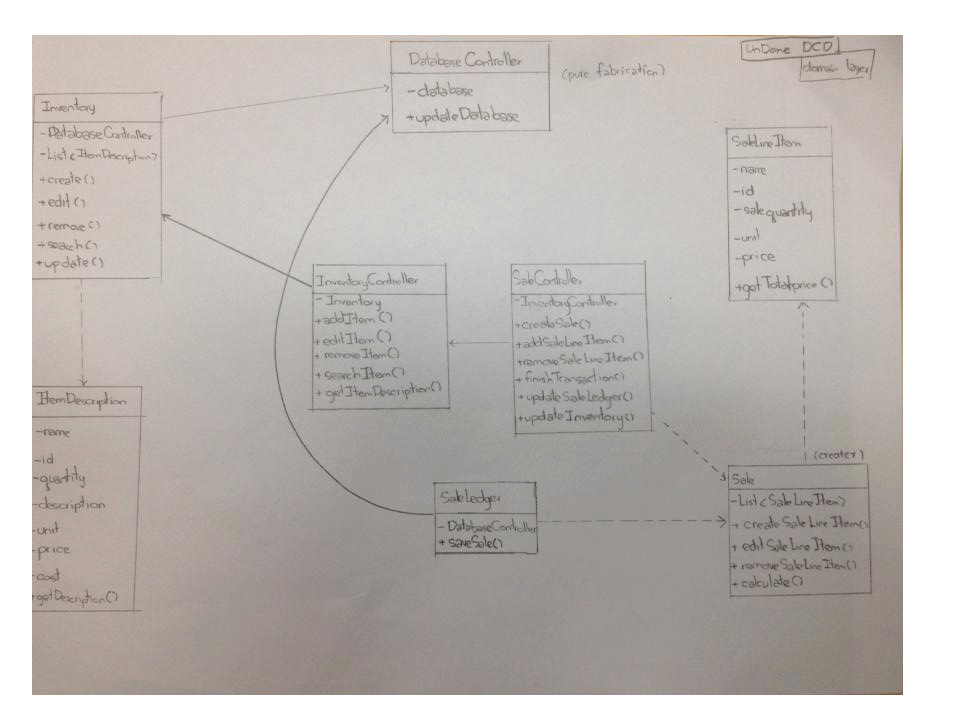
**3.3** **Design of Report**

Report will show 2 things. They are receipt and graph. Sale report get date from Sale Ledger and show sale history order by time. In the top it has datepicker bar that you can choose date to view profit graph of that day.

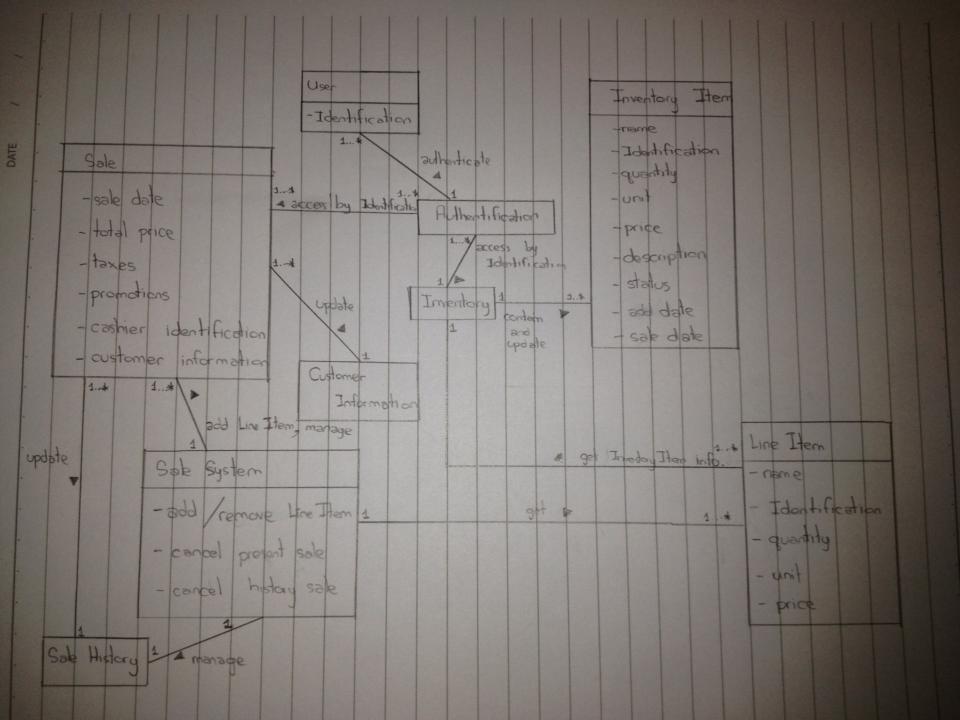


**4.** **Logical View**

Logical Layers

Domain Class DIagram

Domain Model



## 4.1 Domain Vocabulary

Data Base - collect all data. separate into 2 tables: Stock database, Product Catalog database. Each of them are not connect together.

Product Catalog - show product in store. include :

Stock - collect items to stock separate from product catalog

Sale - calculate total payment (include tax), receive money, and return a change.

Sale Line Item - show item which is was selected in a sale

Sale Ledger - record sale history of each sale

Report - show sale receipt , and profit graph

Receipt - Show history of each sale.

Graph - Show item profit in type of graph(day, month, year)

# **5.** **Data View**

# **6.** **Process View**

# **DataBaseHandler:** Control interface of the SQLiteDataBaseHandler.

**SQLiteDatabaseHandler(Content Provider):** Support the use case by add, edit, delete or get the product from the database.

**Product Catalog (Service)**: Manage the product catalog database functionally in add, edit, delete or get the product from product catalog database

**Stock(Service):** Manage the stock database functionally in add or edit get the product from stock database

**Sale(Content Provider):** Support use case in making sale by store SaleLineItem to provide to SaleHandler.

**SaleHandler(Service):** Manage the Sale functionally by coordinate between user interface and sale system.

**SaleLedger(Service):** Manage the ledger database functionally in add and get all ledger in the ledger database.

# **7.** **Use Case View**

**UC1 :** Add new Product to Product Catalog

**UC2 :** Edit product description in Product catalog

**UC3 :** Remove Product in Product Catalog.

**UC4 :** Add item to Stock

**UC5 :** Edit item in Stock

**UC6 :** Remove item in stock

**UC7 :** Make Sale

**UC8 :** Scan Barcode

**UC9** **:** Calculate Subtotal price

**UC10** **:** Calculate Total price (Already include tax)

**UC11** **:** Change price of product

**UC12** **:** Add SaleLineItem

**UC13** **:** Remove SaleLineItem

**UC14** **:** Clear SaleLineItem

**UC15** **:** Update Stock after making sale

**UC16** **:** Show graph of profit

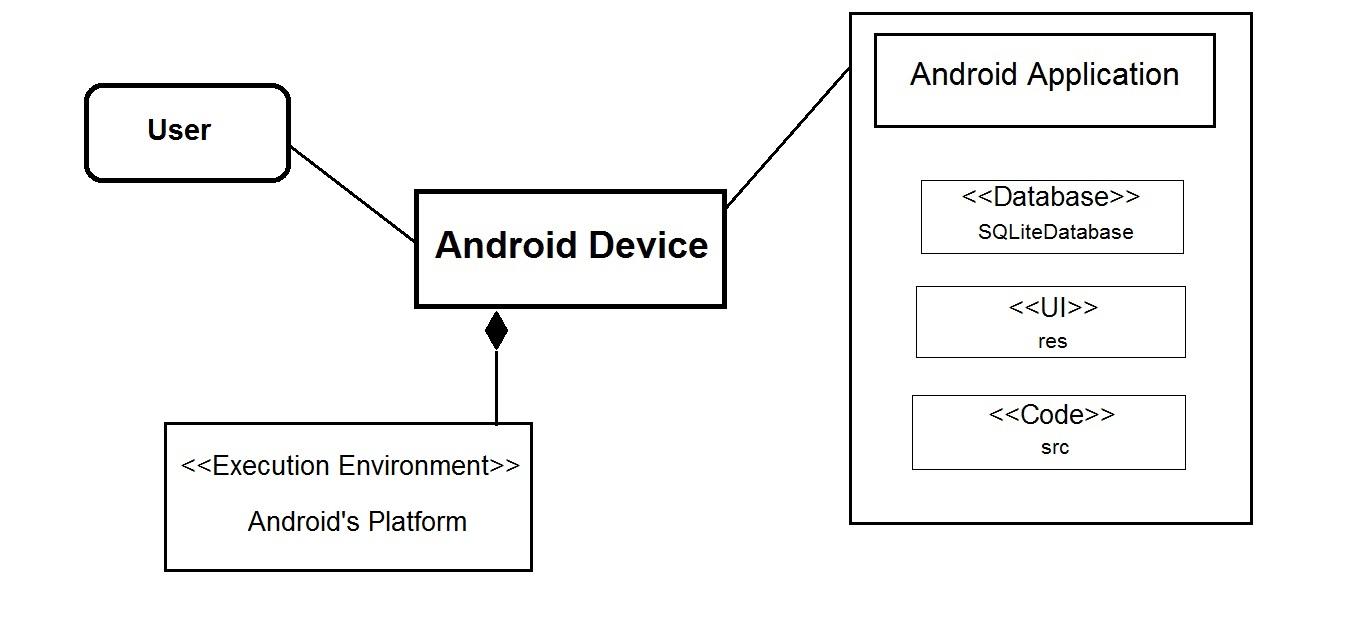
**UC17** **:** Show receipt of SaleLedger

**UC18** **:** Choose date to see report in that day

see more use case information : <https://docs.google.com/document/d/1dMQY_v6B72nVh34K3JxH_hiZqP5EqbWcosHGT1JNP8A/edit>

# 

# **8.** **Deployment View**



# 

# 

# 

# **9.** **References**

- <http://www.cs.toronto.edu/~wl/teach/407/2002/rup-sad.html>

- <http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>