Kiichigo

A Point-of-Sale system for the Raspberry Pi

Grant Wade

June 24, 2016

Contents

Chapter 1

Functional Requirements

1.1 Client-User

1. Track Inventory, including serial numbers
2. Create Transactions
   On Transaction: Customer, Taxable, Discount, Custom Fields
   Per Item: SKU, Description, Ext. Description, Qty, Price, Taxable, Total Price
3. Create Item Entries
4. Lookup Transactions
5. Lookup Customer History
6. Save Transactions (Hold and Layaway)
   Option to input item SN immediately or later
   Option to print hold/layaway sticker for each item
7. Print Receipts
8. Print Labels, including barcodes
9. Scan barcodes using USB scanner
10. Receive Inventory
11. Transfer Inventory Out
12. z report
13. x report
14. Input opening amounts (if static not defined by admin)
15. Input closing amounts

16. Generate Specific reports (Saved by manager)
17. cash drop
18. cash add (put money in register...)
19. Lookup item

    option for having levels of lookup

    first, show categories, then subcat, then subcat... until user doesn't select any subcat, then will show list of all items that fit all cat and subcat filters. will search by name (starting from beginning) from there, or can switch to "search by keyword" mode

1.2 Client-Manager

1. Create Users
2. Restrict User Privileges

    z/x reports

    Information Shown on x/z reports

    Show over/short amounts

    cash drop/add

    Receive Inventory

    Transfer Inv. Out

    lookup customer history

    Item Creation

    Allow refunds/voids

    define which transaction fields user can modify
3. Delete Users
4. Generate Reports (custom too)

    Recommended items to order

1.3 Client-Admin

1. Add New Terminal to Cluster
2. Remove Terminal from Cluster
3. e-mail alerts to admin about problems

    Terminal stops responding

    Suspicious Tx Void

4. Customer Information Fields name rename
5. Item Information Fields name rename
6. Define Custom Fields for Transaction
7. Set standard amount for till's to have on open

1.4 Server Database

1. SQL Changes are done local, then broadcast
2. When client is idle, then synchronize received changes
3. big changes need to roll between clients so the load is spread (1->2; 1->3, 2->4; 1->5, 2->6, 3->7, 4->8, 5->9)
4. Backup mode for terminals (Just holds database and waits for a terminal to fail, when it will ask what terminal it should replace.)

1.5 Hardware

Chapter 2

Non-Functional Requirements

2.1 Client-User

2.2 Client-Manager

2.3 Client-Admin

1. Setup for Terminals is simple (plug it in and it asks if it's the first or not, etc)

2.4 Server Database

2.5 Hardware

1. Compatible with wireless
2. Compatible with various printers
3. Compatible with barcode scanners
4. Compatible with Touchscreen (maybe...)
5. Compatible with employee card scanner

Chapter 3

Use Cases

Chapter 4

State Diagrams

Chapter 5

Database Schematic

Chapter 6

Class Diagrams

Chapter 7

Sequence Diagrams

Chapter 8

Test Cases