# System Implementation & Test

## Introduction

### Overview

In this section, contains information about the test plan, test approach, test process, test environment, test pass/fail criteria, checklist to check test this system.

### Test Approach

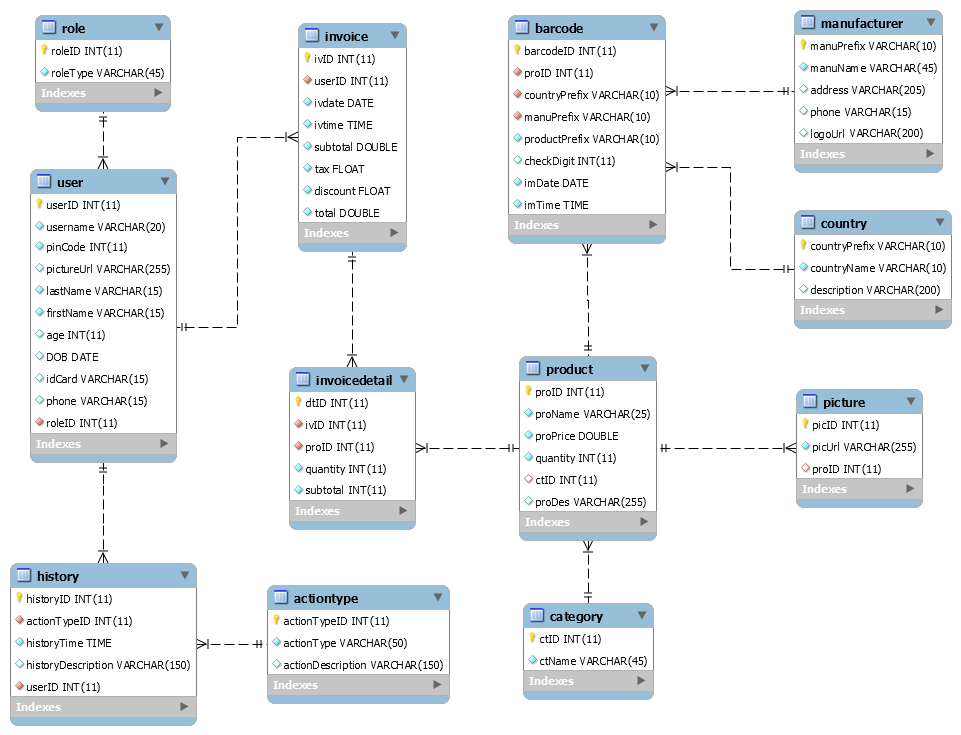
**White box testing**: Developers self-test code that they wrote and fix themselves

**Black box testing:** Test on each function of each module in system to ensure each module execute the right function. Then integration testing to test the function in system when integrate. Finally, system test to test the whole system.

**Goal:** Discover bug in system 🡪 fix bug 🡪 regression test 🡪 completed system.

## Database Relationship Diagram

* 1. **Physical Diagram**

****

* 1. **Data Dictionary**

|  |  |
| --- | --- |
| **Data dictionary: descript be content of all tables** | |
| **Table Name** | **Description** |
| Role | Describe user’s role in the system (Manager and Shop Assistant). One user has one role. |
| User | Describe account of user of system. |
| History | Describe all history in the system. |
| ActionType | Describe all action type of user in the system. |
| Picture | Describe all pictures in the system. |
| Invoice | Describe all invoice in the system. |
| InvoiceDetail | Describe all detail of invoice in the system. |
| Barcode | Describe all product barcode in the system. |
| Product | Describe all product in the system. |
| Category | Describe all category in the system. |
| Manufacturer | Describe all manufacturer in the system. |
| Country | Describe all country in the system. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Domain** | **Null** |
| **Role** | roleID {PK} | ID of role. Unique identifier of role, auto increment. | Integer | No |
| roleType | Type of Role. | Nvarchar(45) | No |
| **User** | userID {PK} | ID of user. Unique identifier of user, auto increment. | Integer | No |
| username | Unique identifier, username of user. | Nvarchar(20) | No |
| pwd | Password of user. | Nvarchar(20) | No |
| pinCode | Unique identifier, information user | Integer | No |
| lastName | Information user | Text | No |
| firstName | Information user | Text | No |
| age | Information user | Integer | No |
| DOB | Information user | Date | No |
| idCard | Information user | Text | Yes |
| phone | Information user | Text | Yes |
| roleID | Information user | Integer | No |
| **History** | historyID {PK} | ID of history. Unique identifier of history, auto increment. | Integer | No |
| actionTypeID | ID of actionType. | Integer | No |
| historyTime | Time of history. | Time | No |
| historyDescription | Description of history. | Nvarchar(150) | Yes |
| userID | ID of user. | Integer | No |
| **ActionType** | actionTypeID | ID of actionType. Unique identifier of actionType, auto increment. | Integer | No |
| actionType | Type of action. | Nvarchar(50) | No |
| actionDescription | Description of action. | Nvarchar(150) | Yes |
| **Picture** | picID {PK} | ID of picture. Unique identifier of picture, auto increment. | Integer | No |
| picUrl | Url of picture. | Integer | No |
| proID | ID of product. | Integer | Yes |
| userID | ID of user. | Integer | Yes |
| **Invoice** | ivID {PK} | ID of invoice. Unique identifier of invoice, auto increment. | Integer | No |
| userID | ID of user. | Integer | No |
| ivdate | Date of invoice. | Date | No |
| ivtime | Time of invoice. | Time | No |
| total | Total of invoice. | Double | No |
| tax | Tax of invoice. | Float | No |
| discount | Discount of invoice. | Float | No |
| payment | Payment | Double | No |
| **InvoiceDetail** | dtID {PK} | ID of Invoice Detail.  Unique identifier of role, auto increment. | Integer | No |
| ivID | ID of invoice. | Integer | No |
| proID | ID of product. | Integer | No |
| quantity | Quantity of product. | Integer | No |
| total | Total of Invoice Detail. | Integer | No |
| **Barcode** | barcodeID {PK} | ID of barcode. Unique identifier of role, auto increment. | Integer | No |
| proID | ID of product. | Integer | No |
| countryPrefix | Prefix of country. | Nvarchar(10) | No |
| manuPrefix | Prefix of manufacturer. | Nvarchar(10) | No |
| productPrefix | Prefix of product. | Nvarchar(10) | No |
| checkDigit | Check digit of barcode. | Integer | Yes |
| imDate | Date of import. | Date | No |
| imTime | Time of import. | Time | No |
| **Manufacturer** | manuPrefix {PK} | Prefix of manufacturer. | Nvarchar(10) | No |
| manuName | Name of manufacturer. | Nvarchar(45) | No |
| address | Address of manufacturer. | Nvarchar(205) | Yes |
| phone | Phone of manufacturer. | Nvarchar(15) | Yes |
| logo | Logo of manufacturer. | Nvarchar(200) | Yes |
| **Country** | countryPrefix {PK} | Prefix of country. | Nvarchar(10) | No |
| countryName | Name of country. | Nvarchar(10) | No |
| description | Country’s information | Nvarchar(200) | Yes |
| **Category** | ctID | ID of category. Unique identifier of category, auto increment. | Interger | No |
| ctName | Name of category. | Nvarchar(45) | No |
| **Product** | proID | ID of product. Unique identifier of category, auto increment. | Interger | No |
| proName | Name of product. | Nvarchar(25) | No |
| proPrice | Price of product. | Double | No |
| quantity | Quantity of product. | Interger | No |
| ctID | ID of category. | Interger | Yes |
| proDes | Description of product. | Nvarchar(255) | Yes |

1. **Performance Measures**
   1. **Clustering Performance**

Clustering is performed by running K Mean Algorithm which has complexity of : O(n \* k \* I \* d)

* n : number of points
* k : number of cluster
* I : number of iteration
* d : number of attributes (3)

Clustering take almost the time of process that we can ignore the time needed to load data from database, digitalize data.

The speed of clustering will vary and increase dramatically when n increase. The purpose of this project is not about optimizing K-Mean Algorithm so it is accepted to let the process run till it completes. Moreover, the clustering is designed to run by staff, wait time is acceptable.

1. Test plan

We have two main test phases: Function test (Integration test) and System test.

For Integration test, we use the functional test. This kind of test shows us the working of each function right or wrong. We focus on the result of the function in different cases.

The System test check all the activities the Smart lock after setup such as: what the system do when it is turned on and if user do somethings, what will response. We create test case to check each function’s activity. Then, we observe and record the result.

* 1. Features to be tested

- Functional Testing:

+ Integration testing: Test all function on Raspberry Application and each external module

+ System testing: Test all function for Raspberry Application – external modules

- UI Testing: Test UI on Raspberry Application

* 1. Features not to be tested

- Hardware module: Raspberry Pi B3.

* 1. Testing tools and environment

- Testing tools:

+ A raspberry board running Rasbian OS.

+ QT Creator.

- Environment:

+ Advanced Serial Port Terminal Application.

* 1. Test past/ fail criteria

For system testing, the criteria are:

* 90% of the test cases must pass.
* 100% of test cases about hardware module must pass.
* All test cases dealing with critical functionality must pass.
* All medium and high severity defects must be fixed.
* Test coverage must be at least 90%.

1. System Testing Test Case
   1. **Component Testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Test Case Procedure | Expected Output | Enter-test Case Dependence | Result | Test Date | Note |
| 1 | Login with role manager | - On Login screen, enter a manager’s password.  - Press “Login” button. | - System will display “Manager menu” screen. | - N/A | - Pass | 18/10/2016 |  |
| 2 | Login with role shop assistant | - On Login screen, enter a shop assistant’s password.  - Press “Login” button. | - System will display “Shop Assistant” screen. | - N/A | - Pass | 18/10/2016 |  |
| 2 | Login with wrong password | - On Login screen, enter a wrong password.  - Press “Login” button. | - System show error message: “Wrong password. Please try again”. | - N/A | - Pass | 18/10/2016 |  |
| 3 | Login with invalid password | - On Login screen, enter an invalid password.  - Press “Login” button. | - System show error message: “Invalid password. Please try again”. | - N/A | - Pass | 18/10/2016 |  |
| 4 | Logout | - On Menu screen, press “Logout” button. | - System will display “Login” screen. | - N/A | - Pass | 18/10/2016 |  |
| 5 | Capture when connected camera | - Not connect camera with raspberry Pi.  - Press “Scan” button. | - Capture success. | - N/A | - Pass | 18/10/2016 |  |
| 6 | Capture when not connect camera | - Not connect camera with raspberry Pi.  - Press “Scan” button. | - System show error message: “Please connect camera”. | - N/A | - Pass | 18/10/2016 |  |
| 7 | Read barcode when focal length is ??? cm | - Put camera in front of the barcode with distance about ?? cm.  - Press “Scan” button. | - System will display barcode’s image on “Search screen”. | 5 | - Pass | 18/10/2016 |  |
| 8 | Read barcode at near distance | - Put camera in front of the barcode with distance about ?? cm.  - Press “Scan” button. | - System will show error message. | 5 | - Pass | 18/10/2016 |  |
| 9 | Read barcode at far distance | - Put camera in front of the barcode with distance about ?? cm.  - Press “Scan” button. | - System will show error message. | 5 | - Pass | 18/10/2016 |  |
| 10 | Read barcode in low- brightness environment | - In low- brightness room, we put camera in front of the barcode with distance about ?? cm.  - Press “Scan” button. | - System will show error message. | 7 | - Pass | 18/10/2016 |  |
| 11 | Read barcode in high- brightness environment | - In high- brightness room, we put camera in front of the barcode with distance about ?? cm.  - Press “Scan” button. | - System will show error message. | 7 | - Pass | 18/10/2016 |  |
| 12 | Validate correct barcode | - On “Search” screen, enter correct number format of barcode in barcode’s text field.  - Press “Search”. | - Validate barcode success.  - System will display “Search screen”. | 7 | - Pass | 18/10/2016 |  |
| 13 | Validate invalid barcode | - On “Search” screen, enter correct format of barcode in barcode’s text field.  - Press “Search”. | - System will show error message: “Invalid barcode. Please try again.” | 7 | - Pass | 18/10/2016 |  |
| 14 | Search available barcode in database | - On “Search” screen, enter available barcode in database.  - Press “Search” button. | - System will display product’s information on “Result Search” screen. | 12 | - Pass | 18/10/2016 |  |
| 15 | Search not exist barcode in database | - On “Search” screen, enter not exist barcode in database.  - Press “Search” button. | - System will show error message: “Not found”. | 12 | - Pass | 18/10/2016 |  |
| 16 | Order in range product’s quantity in database | - On “Result Search” screen, input quantity in range product’s quantity in database.  - Press “Enter” button. | - System will display “View Cart” screen. | 12 | - Pass | 18/10/2016 |  |
| 17 | Order out of range product’s quantity in database | - On “Result Search” screen, input quantity out of range product’s quantity in database.  - Press “Enter” button. | - System will show error message: “Not enough product”. | 14 | - Pass | 18/10/2016 |  |
| 18 | Add item | - On “Result Search” screen, input quantity.  - Press “Add Order” button. | - System will display “Search” screen with currently item’s information. | 17 | - Pass | 18/10/2016 |  |
| 19 | Add same item | - After add item, we continue add same item.  - Press “View Cart” button. | - System will display “Search” screen, it contains quantity after 2th added item. | 18 | - Pass | 18/10/2016 |  |
| 20 | Delete | 1. On “View Cart” screen, choose item, press “Delete Item” button.  2. Press “View Cart” button. | - After step 1: System will display “Search” screen.  - After step 2: System will display “View Cart” screen without deleted item. | 14 | - Pass | 18/10/2016 |  |
| 21 | Search item available in “View Cart” screen. | - On “View Cart” screen, - Input item available in Cart.  -Press “Search” button. | - System will show product information on “View Cart” screen. | 14 | - Pass | 18/10/2016 |  |
| 22 | Search item not exist in “View Cart” screen. | - On “View Cart” screen, - Input item not exist in Cart.  -Press “Search” button. | - System will show error message: “Product not found”. | 14 | - Pass | 18/10/2016 |  |
| 23 | Enter number of cash more than or equal to number of total on “Payment” screen. | - Enter number of cash more than number of total on “Payment” screen.  - Press “Submit” button. | - System will display “Payment” screen, it contains change  (change = cash - total) | - N/A | - Pass | 18/10/2016 |  |
| 24 | Enter number of cash less than number of total on “Payment” screen. | - Enter number of cash less than number of total on “Payment” screen.  - Press “Submit” button. | - System show error message: “ Cash is not enough.” on “Payment” screen. | - N/A | - Pass | 18/10/2016 |  |
| 25 | Not enter number of cash on “Payment” screen. | - No input cash on “Payment” screen.  - Press “Submit” button. | - System show error message: “Please enter cash” on “Payment” screen. | - N/A | - Pass | 18/10/2016 |  |
| 26 | Enter character in cash field on “Payment” screen. | - Enter character in cash field on “Payment” screen.  - Press “Submit” button. | - System show error message: “Please enter number of cash” on “Payment” screen. | - N/A | - Pass | 18/10/2016 |  |
| 27 | Print a receipt when printer is not connect | - Not connect printer with system.  - On “Payment” screen, press “Print” button. | System show error message: “Please connect printer” on “Payment” screen. | 23 | - Pass | 18/10/2016 |  |
| 28 | Print a receipt when printer was connected | - Connect printer with system.  - On “Payment” screen, press “Print” button. | - Print receipt success. | 23 | - Pass | 18/10/2016 |  |

* 1. **Integration Testing**

G. Appendix

1. Rasbian operating system:

http://www.raspbian.org/RaspbianAbout

2. OpenCV library to detect object: <http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html>

3. Raspberry Pi Camera module document:

<http://www.ics.com/blog/raspberry-pi-camera-module#.VDqccvldX6U>

4. GPIO library for Raspberry:

<http://www.airspayce.com/mikem/bcm2835/>

5. Setup rasbian OS:

http://chiaseprojects.blogspot.com/2014/06/huong-dan-cai-at-he-ieu-hanh-raspbian.html

6. Development in/out with raspberry:

http://codientu.org/threads/10519/

7. UART raspberry guideline:

<http://www.raspberry-projects.com/pi/programming-in-c/uart-serial-port/using-the-uart>