

## II. Trace Iteration Plan in Requirement Document

Second, I will review the iteration plan in requirement document as Fig2 below.

### Requirement Prioritization

The most basic and critical requirements are:

- The POS system could recognize the merchandise and generate the correct bill
- The database could be updated by cashier and manager
- Basic interface for above operations

Then also important:

- The POS system should be able to synchronize with database automatically
- Cashier and manager should login in system with different authority

Then valuable:

- Tools to parse and analyze the financial data from database and generate the daily/monthly reports

Nice features:

- Interface to apply coupons in a deal
- More payment methods support, AliPay, Member Card, etc.
- Better UI experience
- Performance improvement

### System Iteration Plan

It will be basically coordinated with the requirement prioritization above:

- The first iteration aims to build a basic POS system and a linked database, and to pass the tests of basic shopping scenario.
- Then implement the important support part to fulfill the needs of daily management: Cashiering, changes back and update the inventory in both automated and manual way.
- Financial reports and some basic analysis generation supported.
- More features to be added.

Figure 2: Review of iteration plan in requirement document

**3 Basic requirement:** POS system that merchandise and generate correct bill, database update, basic interface.

**Other addition requirement:** Log in with different authority, generate reports, apply coupon, more payment methods support and better UI.

I finished all features in the requirement document, I will show detail in later sections.

## IV. Detail Development and User Manual

### 1. Version 0

In version0 , I tried to complete basic features first.

1. I start from cashier interface, added a power button and log in system. Cashiers and admins could log in with different authority and log out.

**Passport for cashiers is cashier and for admins is admin.** Two lights reflect the status of log in and power, green for cashiers and red means admins log in. In Fig3, Fig4.

2. **Cashier** could check out, apply coupon, show bill, print receipt for customers. In Fig5, Fig8, Fig9, Fig10.
3. For **Admin**, I just finish the interfaces in this version, admin could change discount, get report. In Fig11.

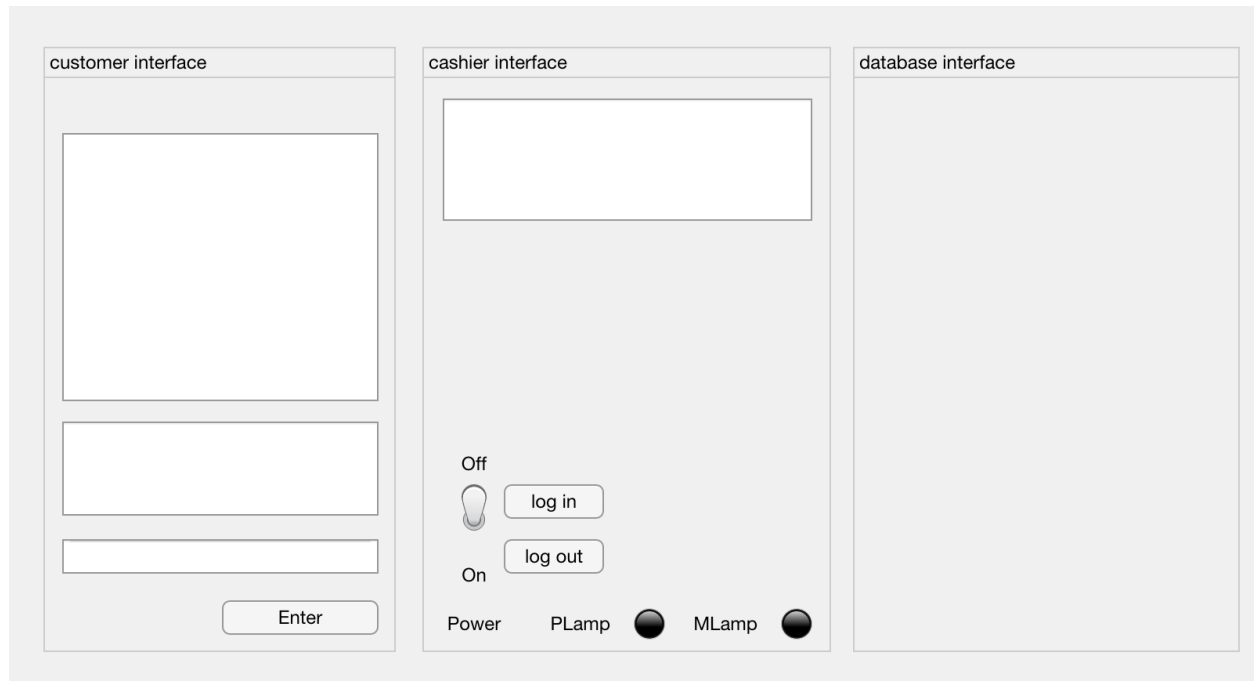


Figure 3: UI of Version0

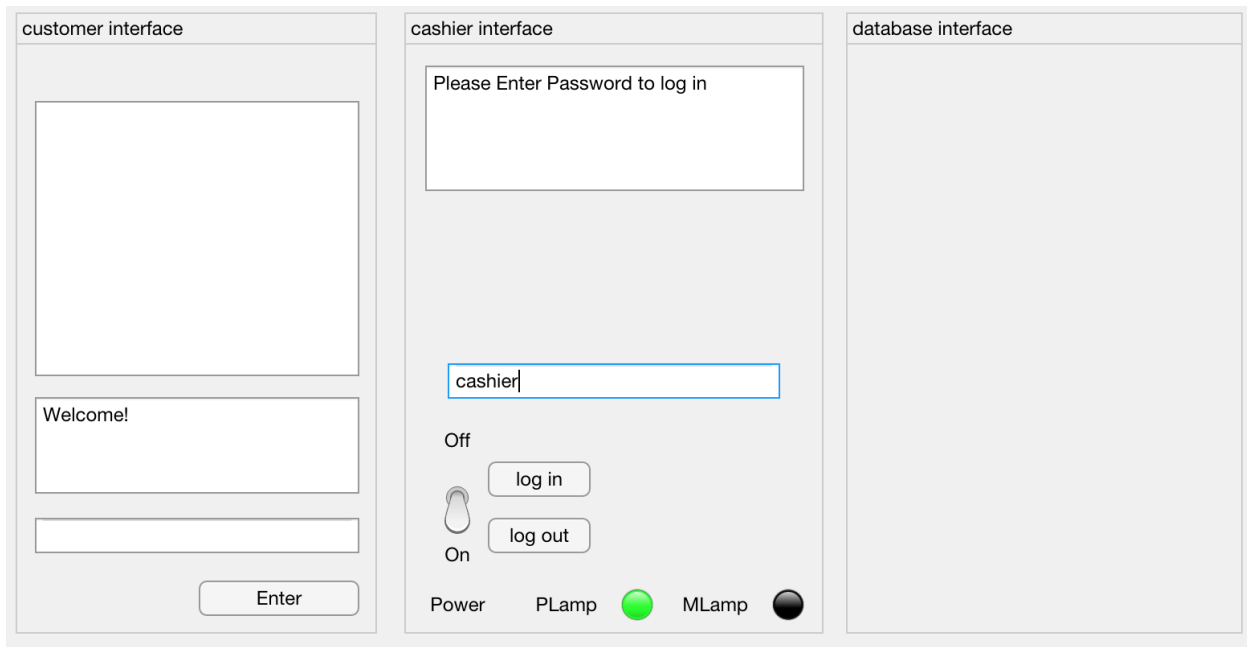


Figure 4: Cashier Log in of Version0

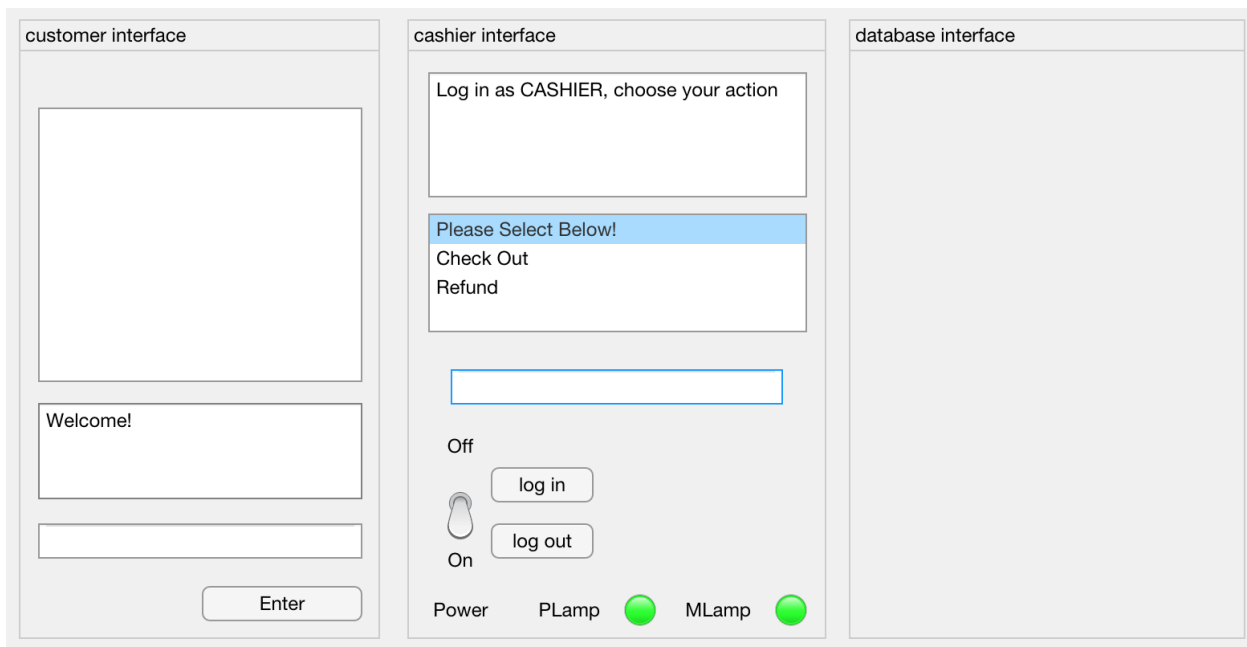


Figure 5: Main Interface of Cashier in Version0

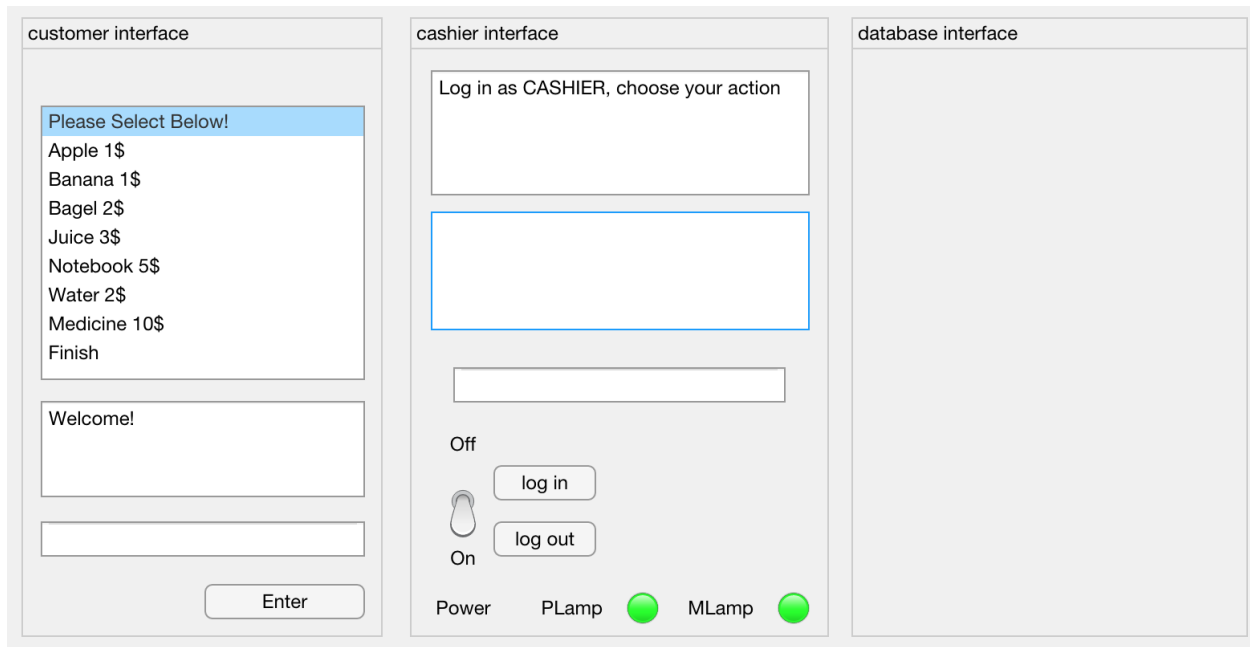


Figure 6: UI for Customers to Shopping in Version0

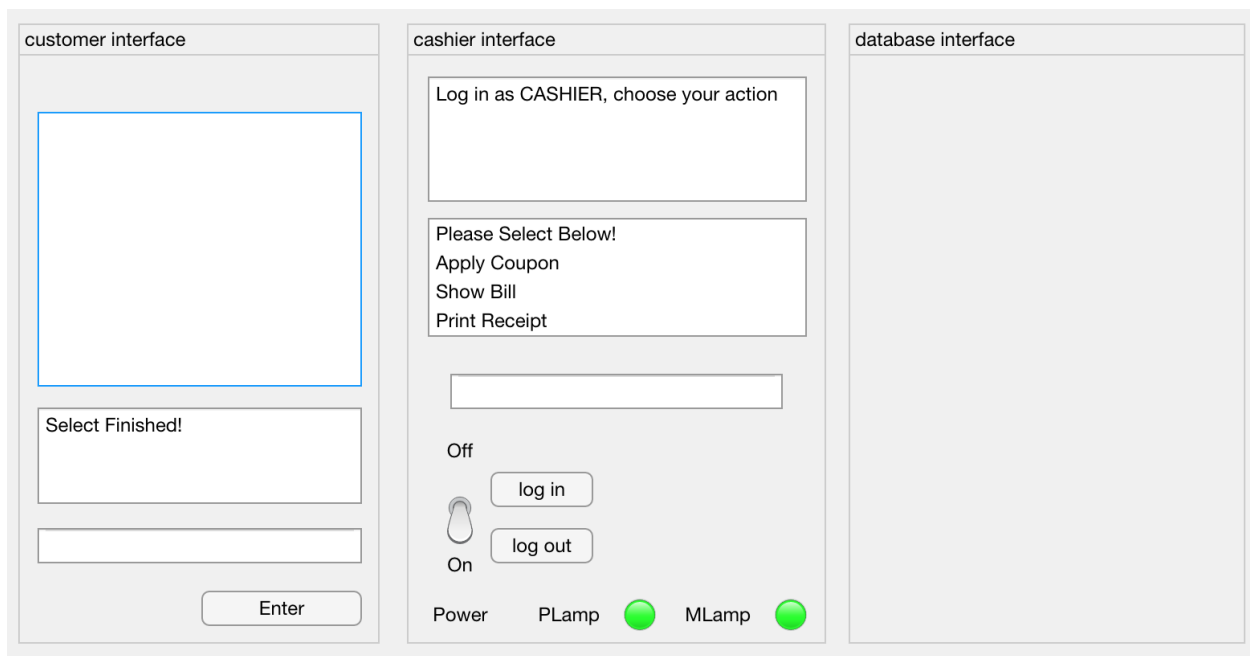


Figure 7: Customers Finish shopping in Version0

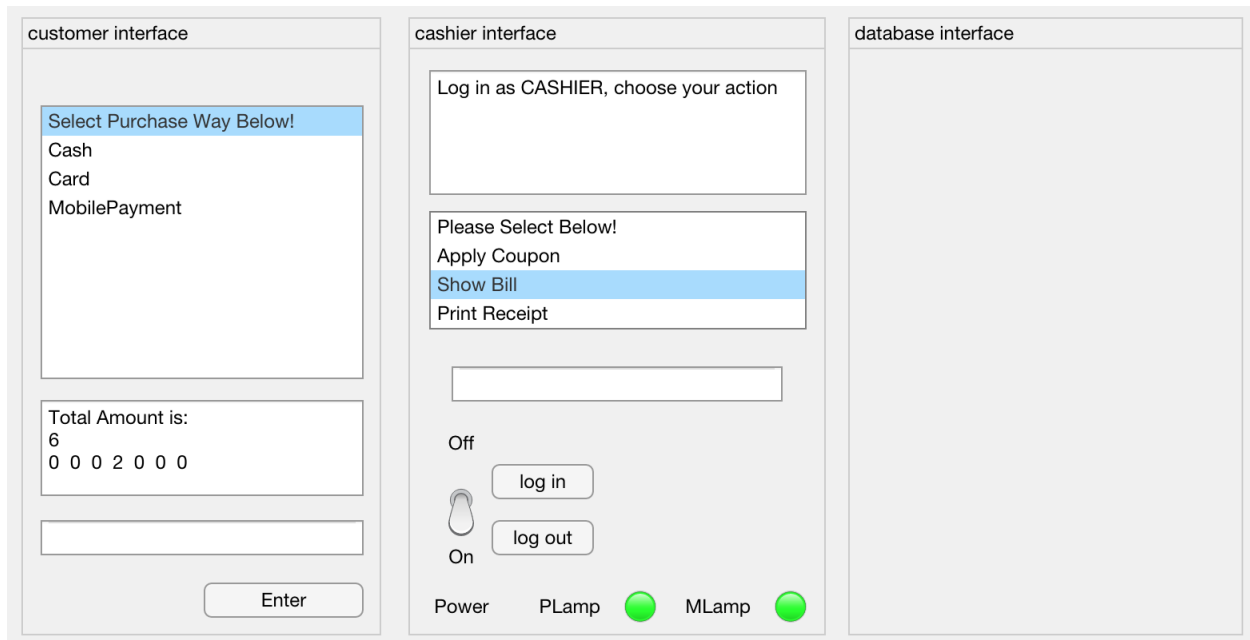


Figure 8: Cashier Show Bill to Customers in Version0

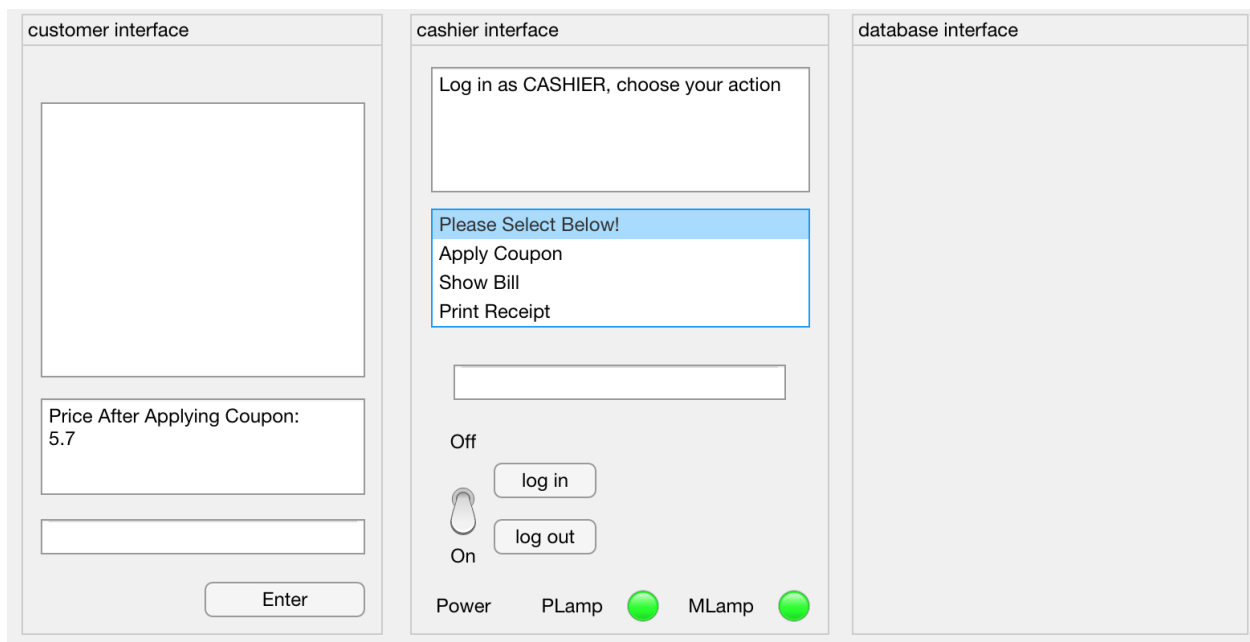


Figure 9: Cashier Apply Coupon for Customers in Version0

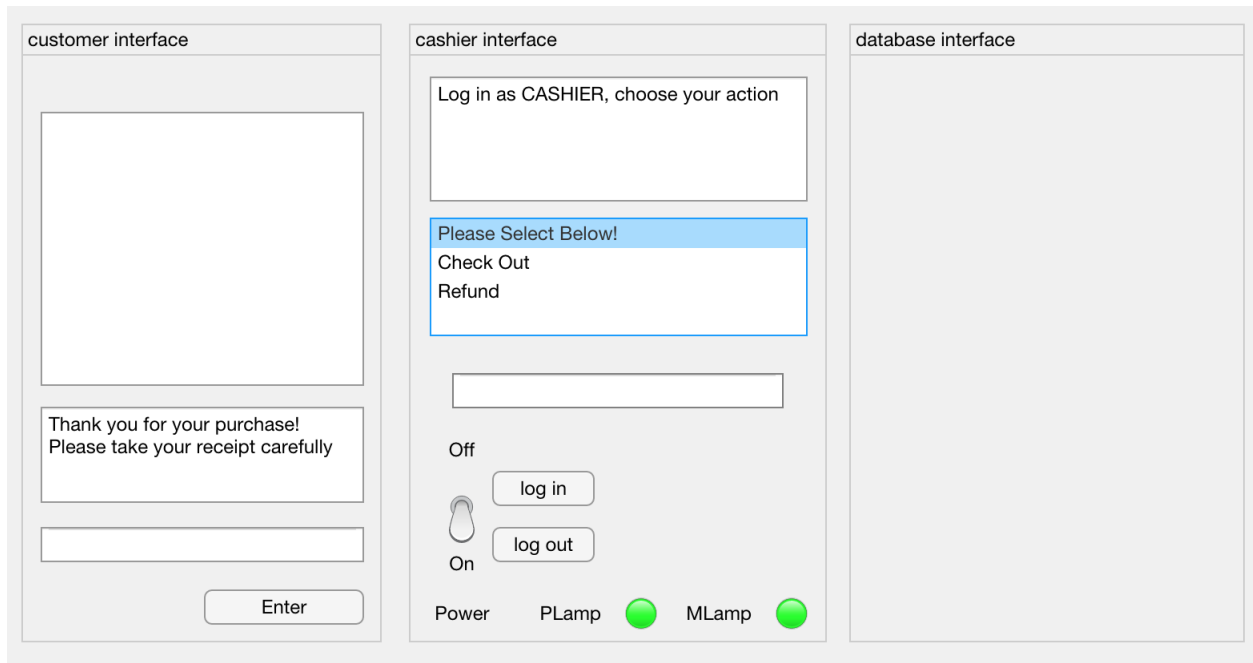


Figure 10: Cashier Print Receipt for Customers in Version0

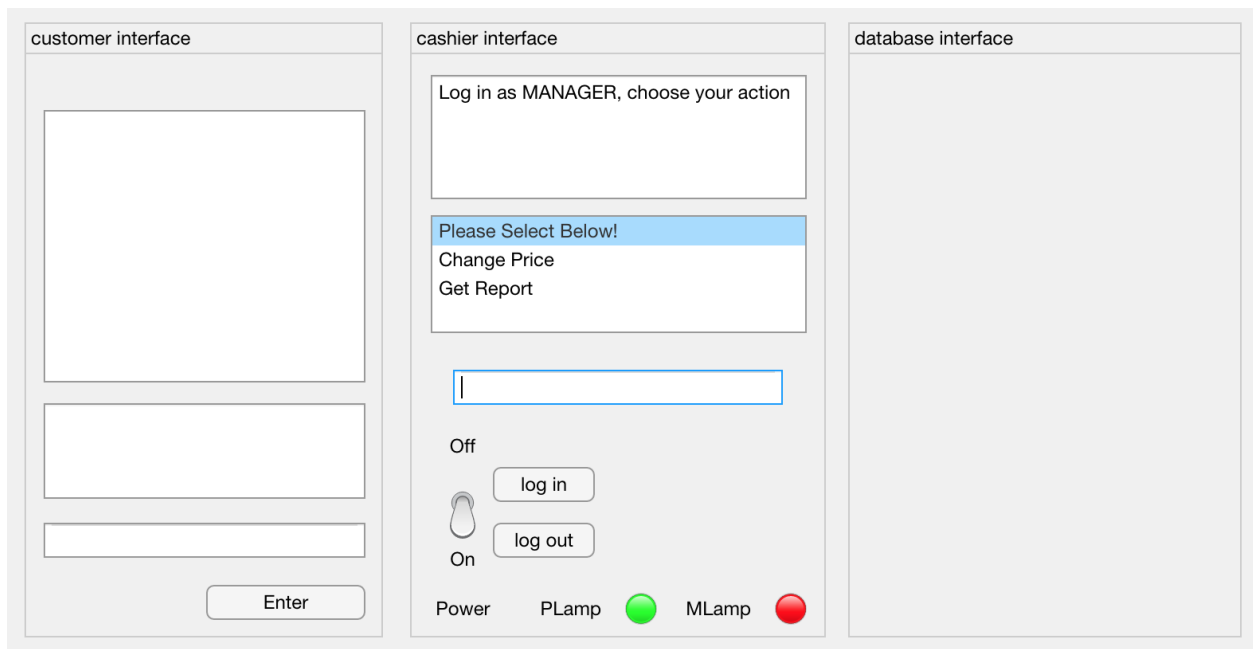


Figure 11: Main UI for admins in Version0

## 2. Version 1

In Version1, I implement database and finish all its maintenance. Also I developed more features to basic requirements and finished additional functions.

1. **Cashiers** could refund products and update inventory database. In Fig16, Fig17, Fig18.
2. **Database history**, trace and search for bill history. In Fig21, Fig22, Fig23, Fig24.
3. **Customers** could use different payment, including cash(including change), card(including card verify(correct 3-bit codes including 123,321,111,222,333)), mobile payment(QR code based). In Fig12, Fig13, Fig14, Fig15.
4. **Admins** could change the discount plan of supermarket and get total sell amount, sold products and inventory reports. In Fig19, Fig20, Fig21, Fig22, Fig23, Fig24.

**Just Show Modified Images of This Version Below!**

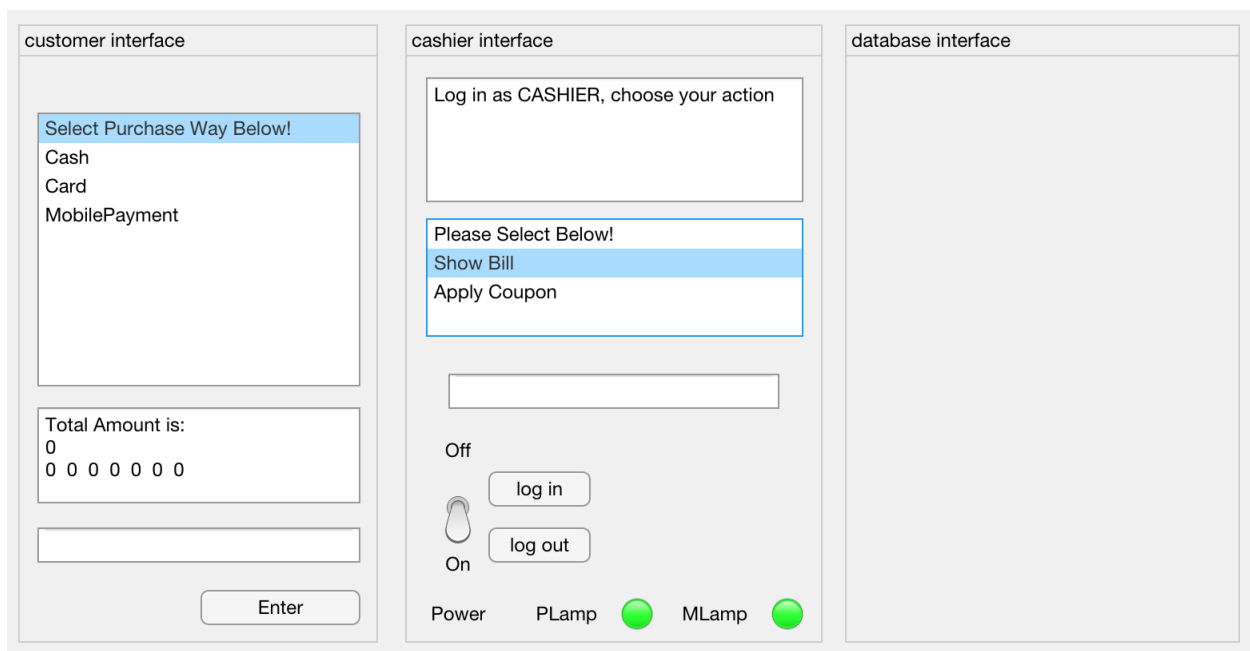


Figure 12: Multiple Support for payment in Version1

customer interface	cashier interface	database interface
<div>Select Purchase Way Below!</div> <div>Cash Card MobilePayment</div> <div>Total Amount is: 0 Please Enter your Cash Amount Below</div> <div><input type="text"/></div> <div>Enter</div>	<div>Log in as CASHIER, choose your action</div> <div>Please Select Below!</div> <div>Show Bill Apply Coupon</div> <div><input type="text"/></div> <div>Off</div> <div><input type="checkbox"/></div> <div>log in</div> <div>log out</div> <div>On</div> <div>Power      PLamp      MLamp</div>	

Figure 13: Cash payment UI in Version1

customer interface	cashier interface	database interface
<div>Select Purchase Way Below!</div> <div>Cash Card MobilePayment</div> <div>Please Enter your Card 3-digit Code Below</div> <div><input type="text"/></div> <div>Enter</div>	<div>Log in as CASHIER, choose your action</div> <div>Please Select Below!</div> <div>Show Bill Apply Coupon</div> <div><input type="text"/></div> <div>Off</div> <div><input type="checkbox"/></div> <div>log in</div> <div>log out</div> <div>On</div> <div>Power      PLamp      MLamp</div>	

Figure 14: Card payment UI in Version1



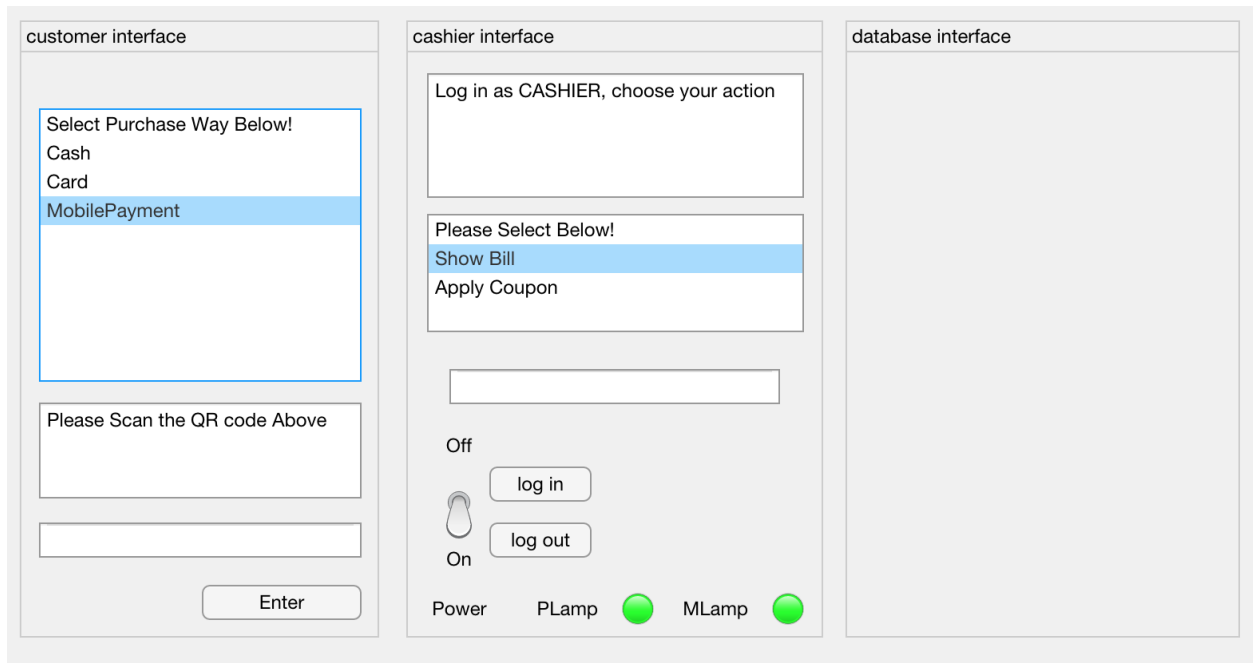


Figure 15: Mobile payment UI in Version1

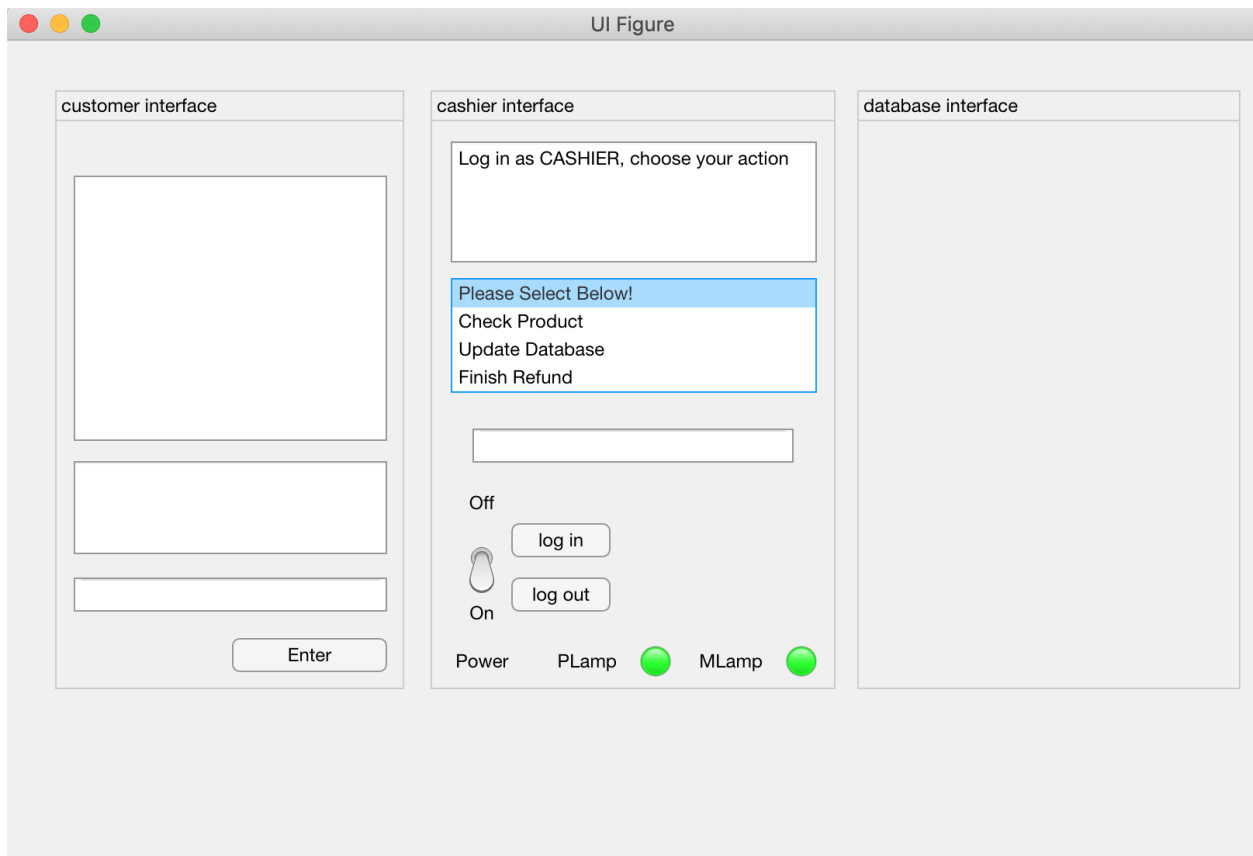


Figure 16: Refund UI for cashier in Version1

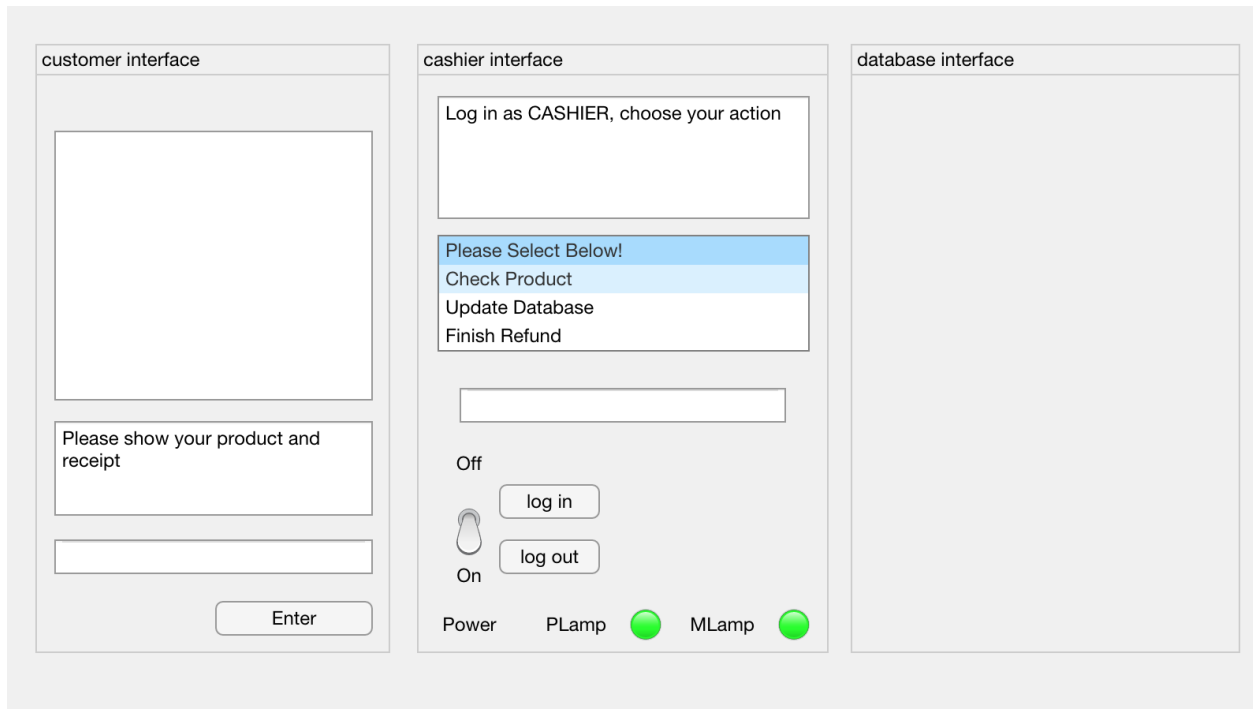


Figure 17: Cashier Check Products for customers in Version1

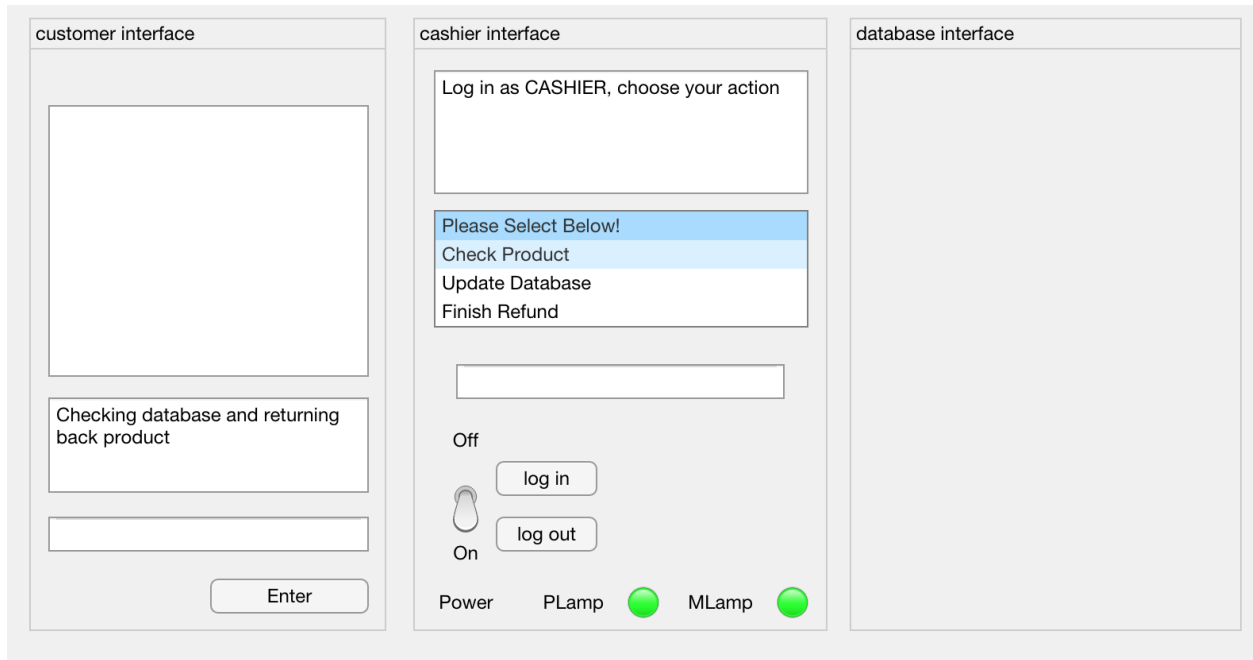


Figure 18: Cashier update database in Version1

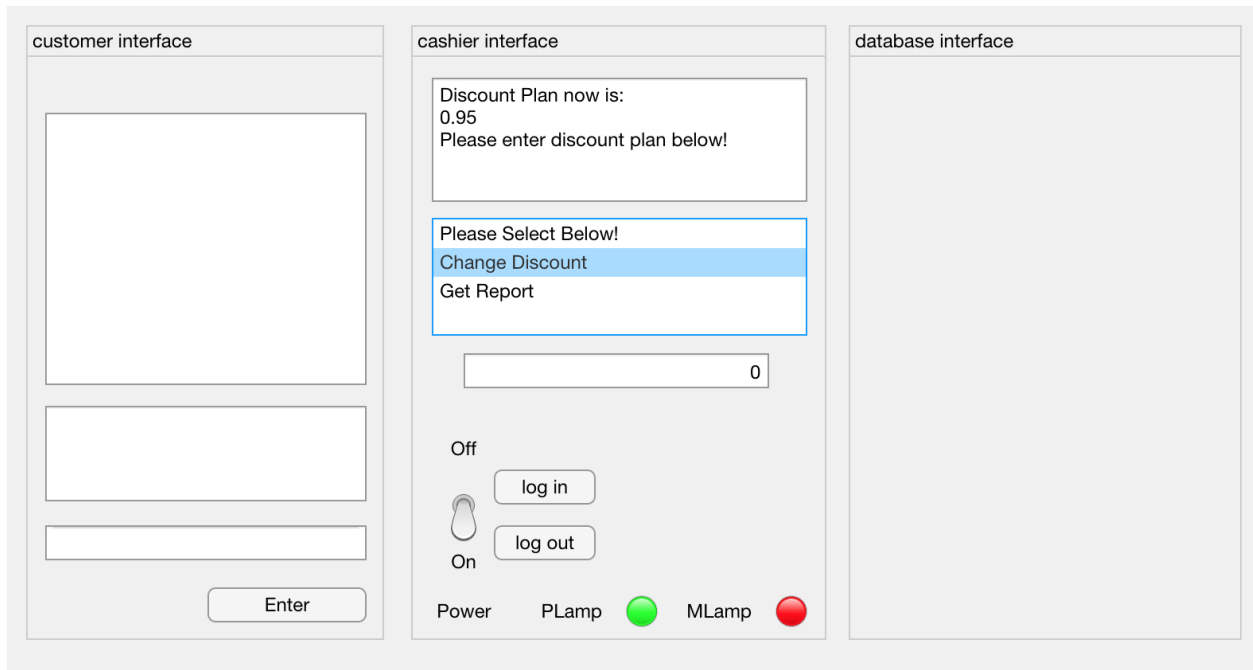


Figure 19: Admin change discount in Version1

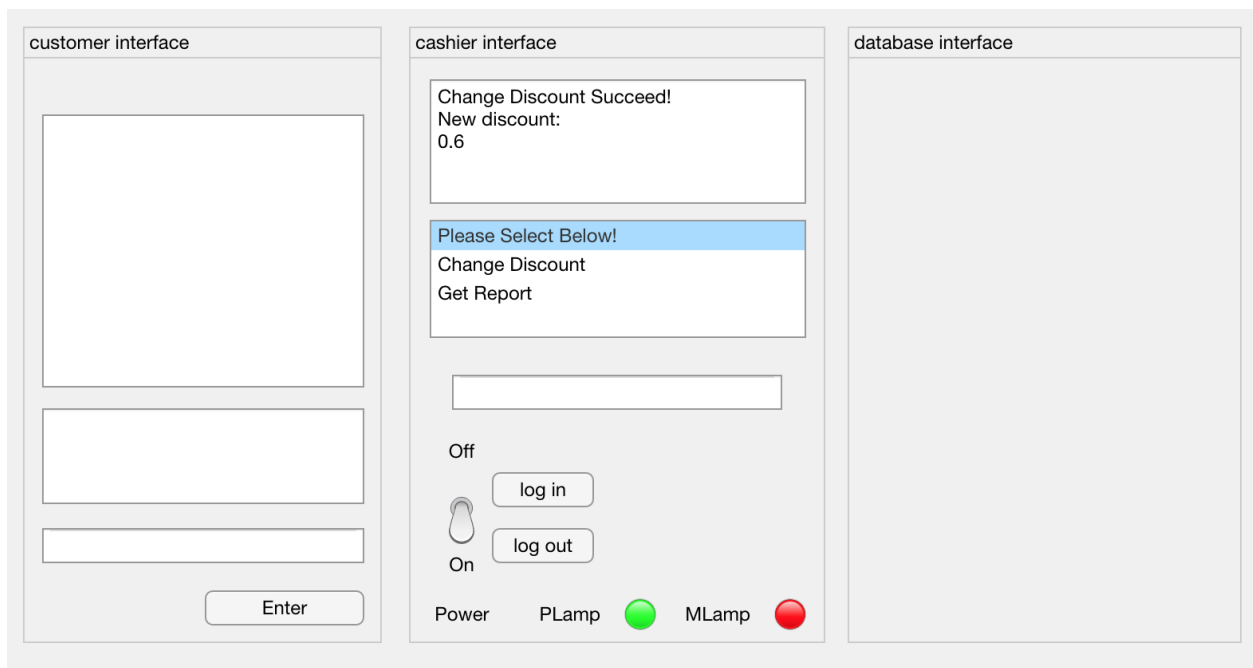


Figure 20: New discount case in Version1

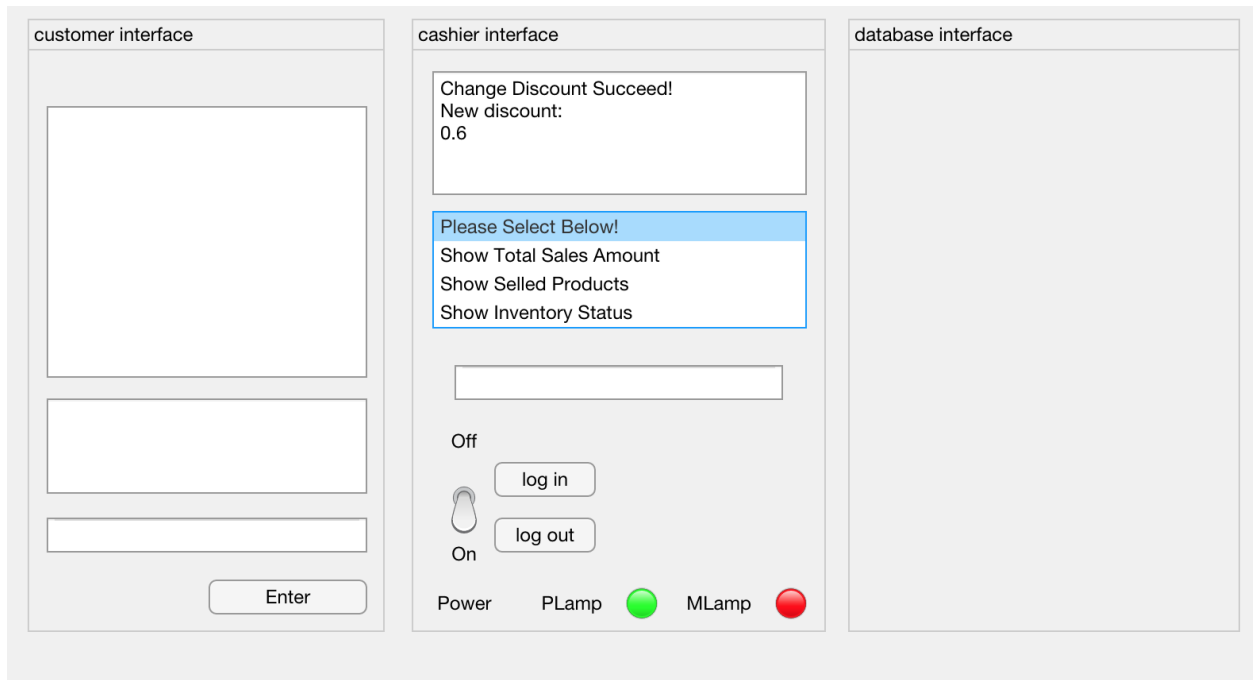


Figure 21: Report UI for Admin in Version1

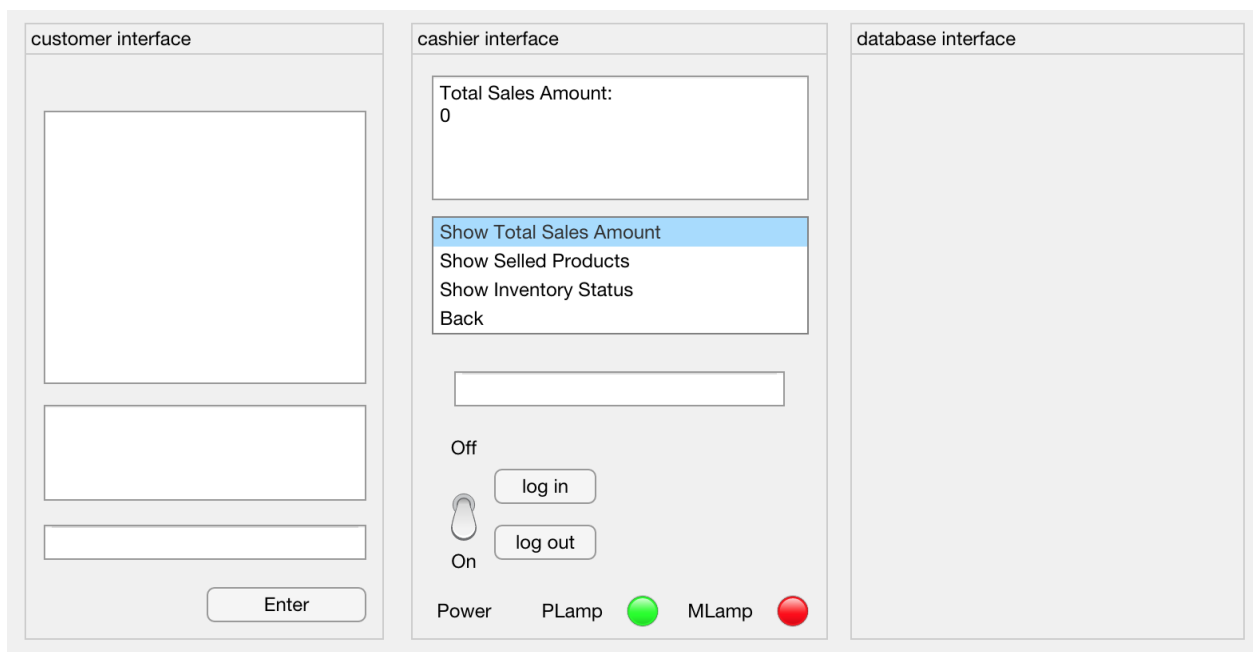


Figure 22: Report Sell Amount for Admin in Version1

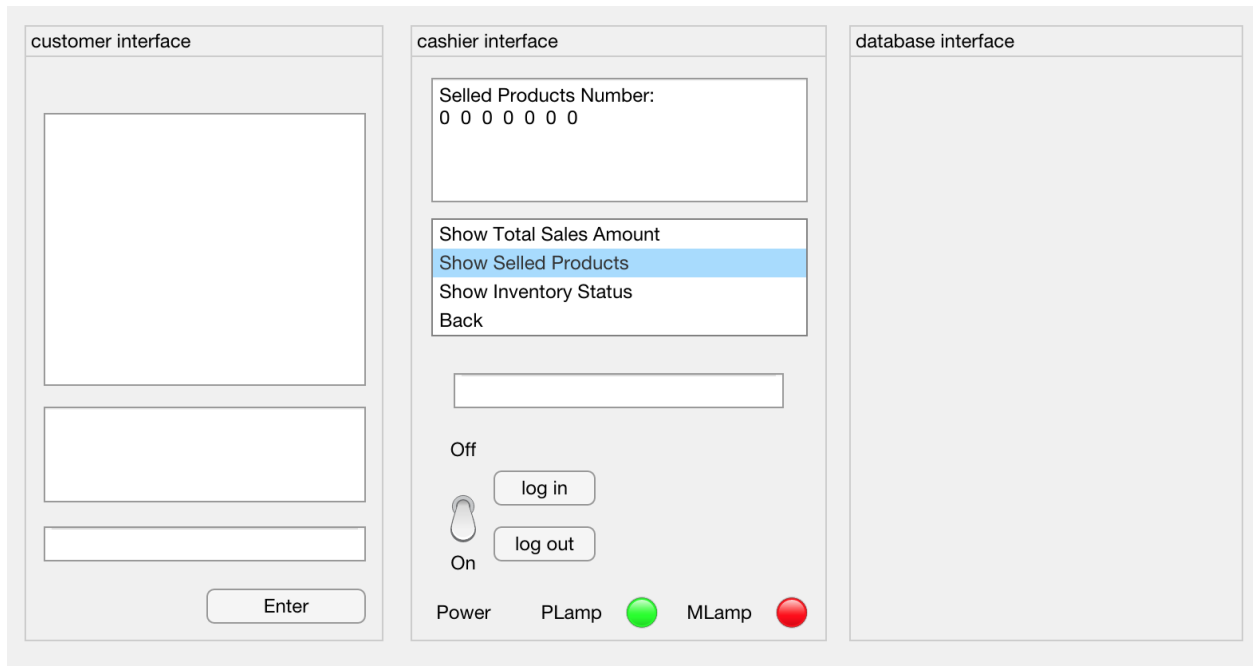


Figure 23: Report Sold Products for Admin in Version1

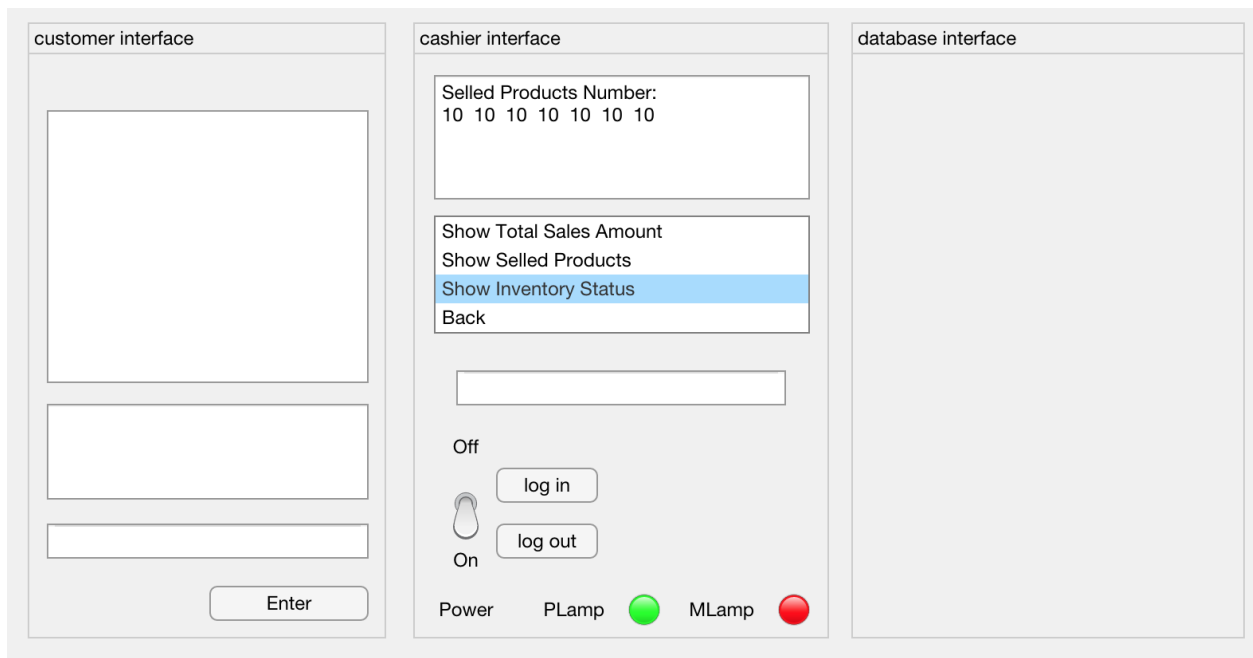


Figure 24: Report Inventory Status for Admin in Version1

### 3. Version 2

In Version 2, I implemented addition functions for admins. Also, I add more operations related to database and trace previous bills more precisely.

1. **UI for database** and build bill history and inventory status, show current bill and summary information of the supermarket. In Fig25, Fig26.
2. **Cashiers** could refund products using bill No. and update the database. In Fig27, Fig28.
3. **Admins** could add products to inventory and search bill using bill No. In Fig29, Fig30, Fig31, Fig32, Fig33.
4. **Customers** could refund use bill No. and check their bill history. In Fig27.
5. **Reduce Risk:** avoid multi-time refund and refund before purchase. In Fig27, Fig28.

**Just Show Modified Images of This Version Below!**

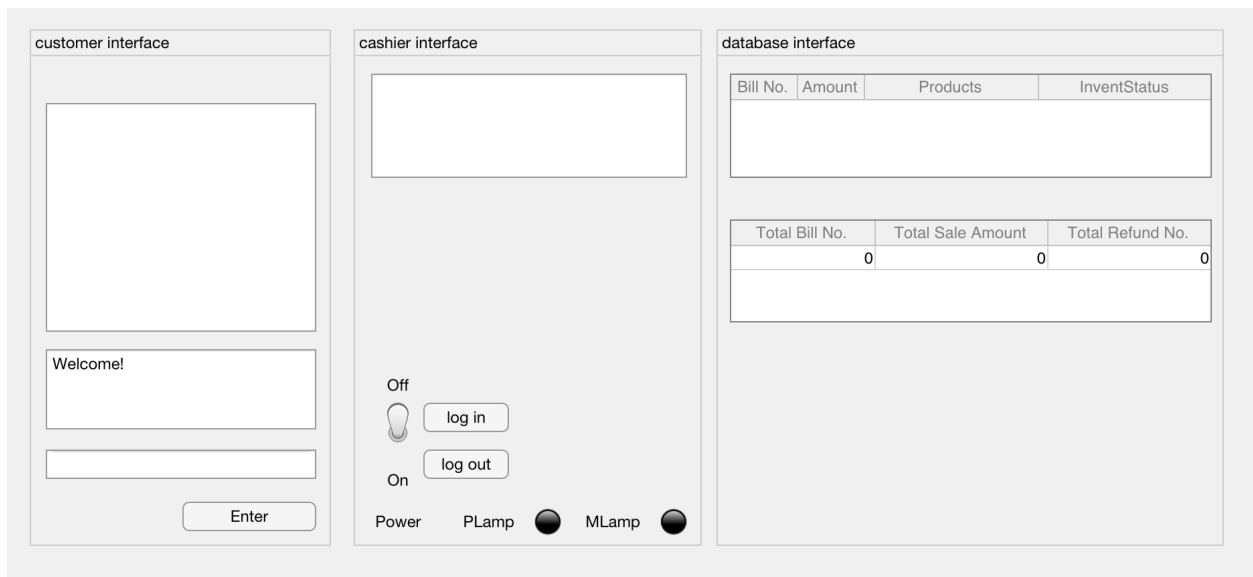


Figure 25: New Database UI in Version2

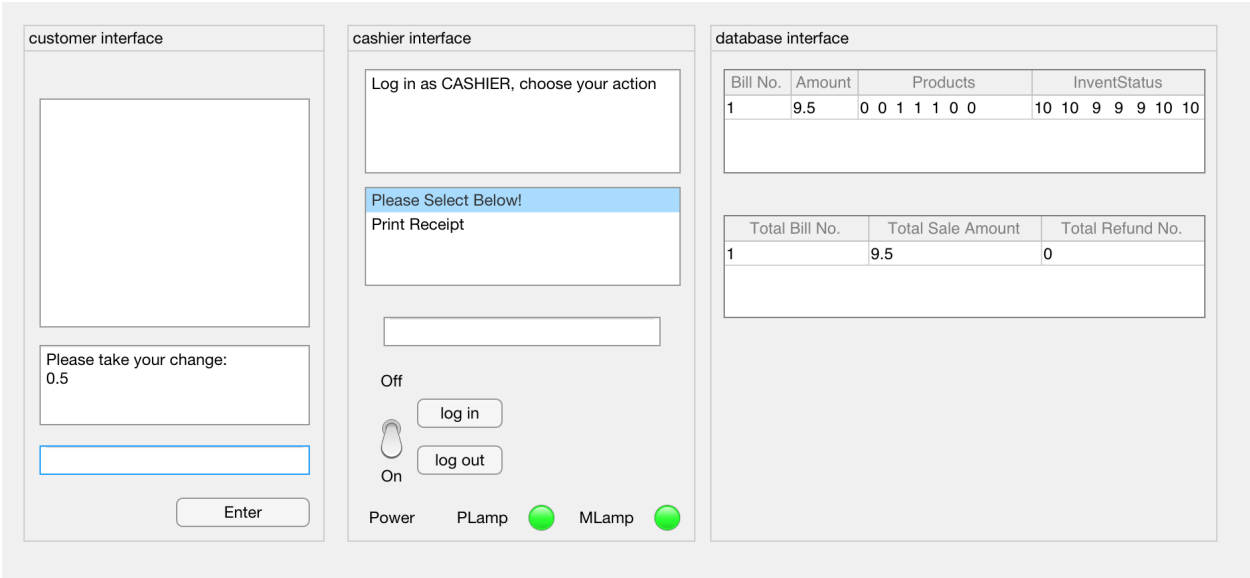


Figure 26: New Database UI in Version2

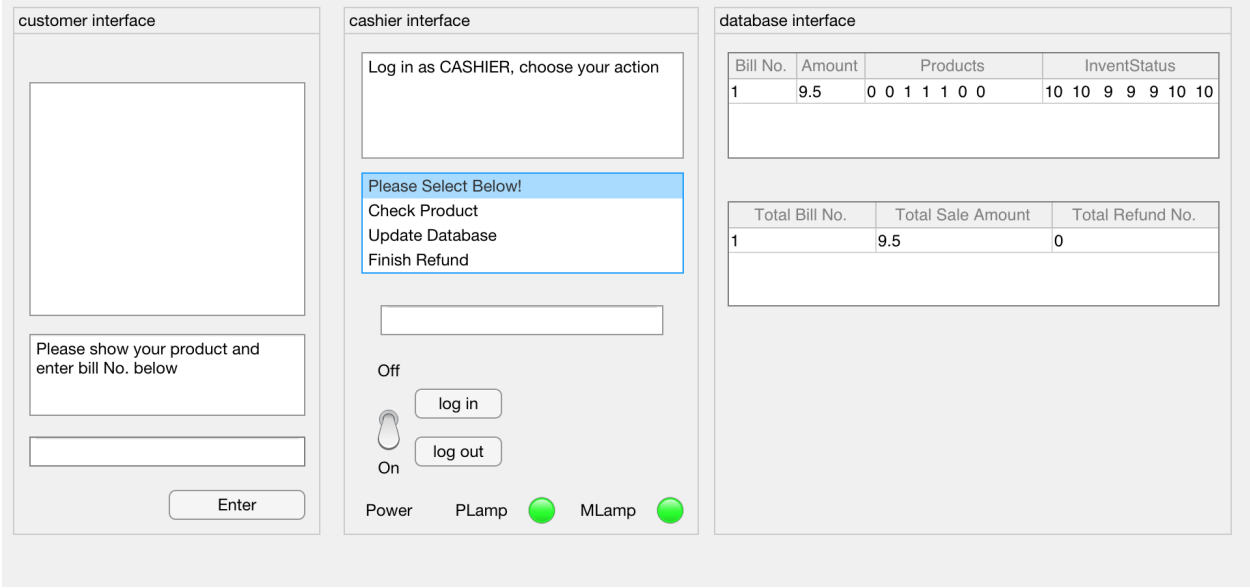


Figure 27: Cashier Check Refund Products Using Bill No. in Version2

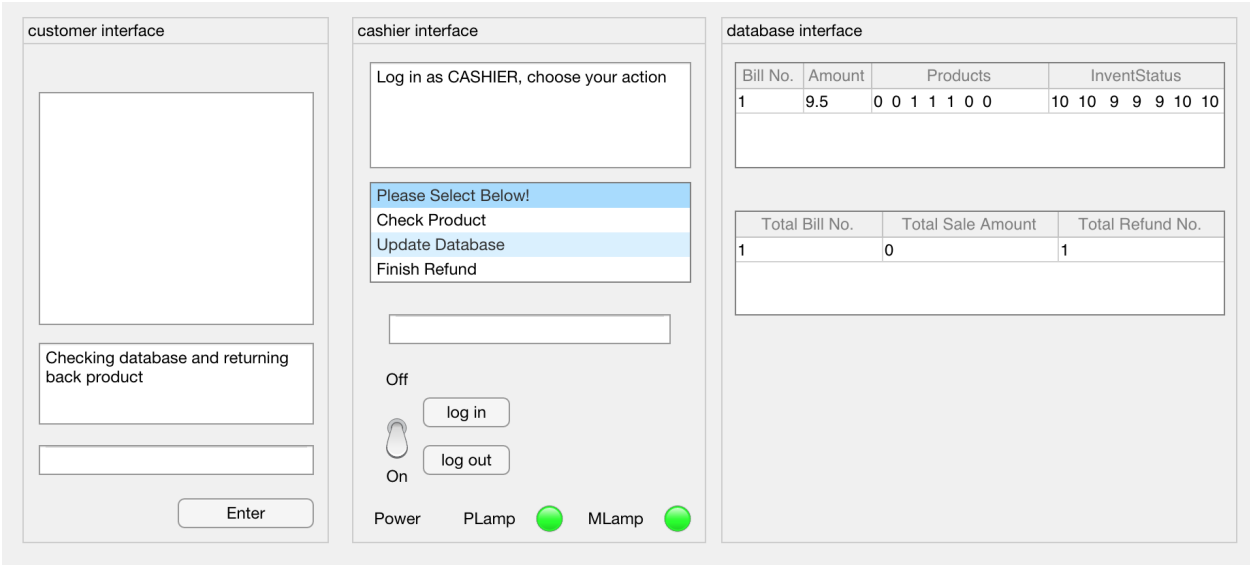


Figure 28: Cashier Update Database Real-time in Version2

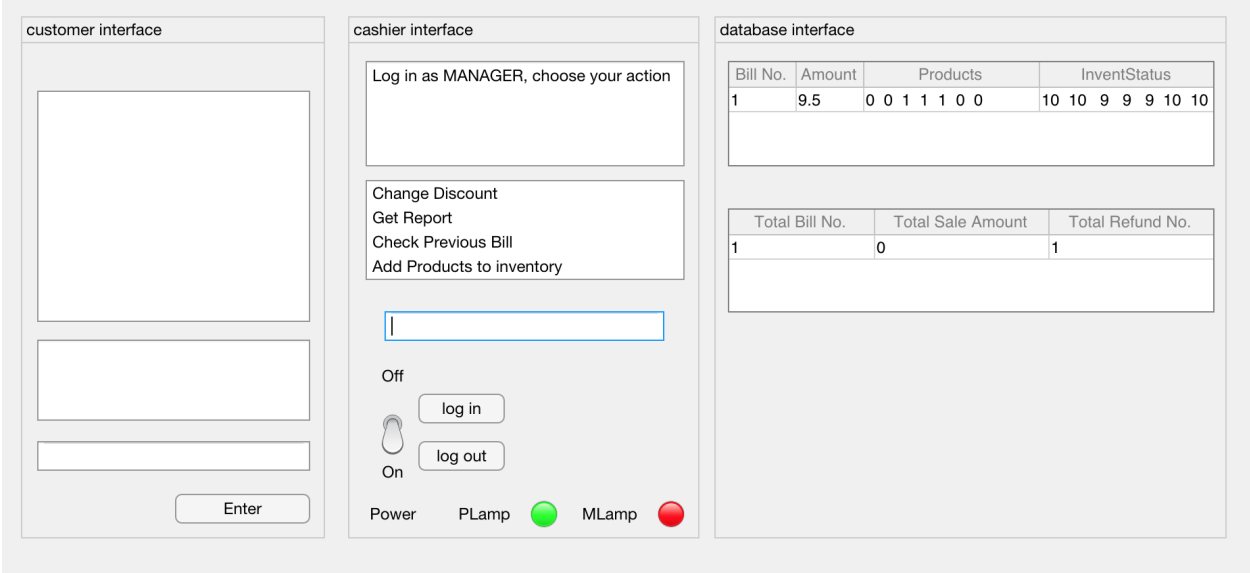


Figure 29: New Admin UI in Version2



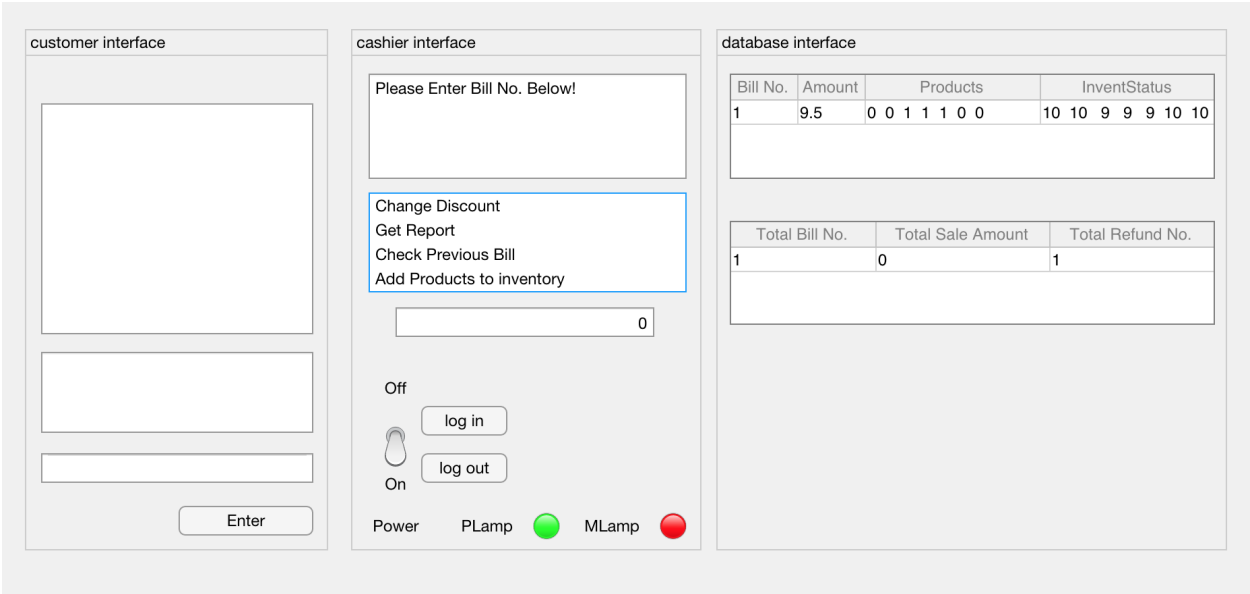


Figure 30: Admin Check Previous Bill History in Version2

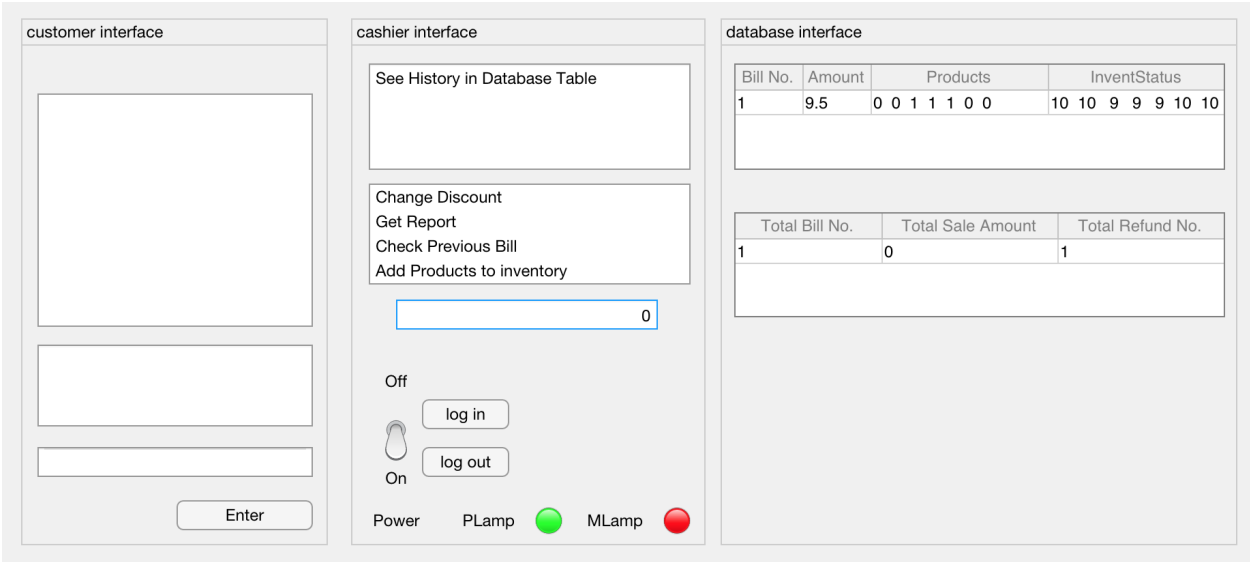


Figure 31: Admin Check Previous Bill History Result in Version2

customer interface

Enter

cashier interface

Current Inventory Products  
10 10 10 10 10 10 10  
Please Enter Add Number Below!

Change Discount  
Get Report  
Check Previous Bill  
Add Products to inventory

0

Off

log in

log out

On

Power    PLamp    MLamp

database interface

Bill No.	Amount	Products	InventStatus
1	9.5	0 0 1 1 1 0 0	10 10 9 9 9 10 10

Total Bill No.	Total Sale Amount	Total Refund No.
1	0	1

Figure 32: Admin Add More Products to Inventory in Version2

customer interface

Enter

cashier interface

New inventStatus:  
14 14 14 14 14 14 14

Change Discount  
Get Report  
Check Previous Bill  
Add Products to inventory

Off

log in

log out

On

Power    PLamp    MLamp

database interface

Bill No.	Amount	Products	InventStatus
1	9.5	0 0 1 1 1 0 0	10 10 9 9 9 10 10

Total Bill No.	Total Sale Amount	Total Refund No.
1	0	1

Figure 33: Admin Add More Products to Inventory Result in Version2

#### 4. Version 3

In Version3, Basic UI and functions of supermarket interfaces and additional features are almost completed. So this version mainly focus on bug fixed and reduce some possible risks.

1. Database could show real-time information in UI. In Fig36, Fig35.
2. Customers could not refund multiple times. In Fig36, Fig34.
3. Cashier could check refund products multiple times and recognize refunded bill. In Fig34, Fig35, Fig36.
4. Admin could trace if specific bill was funded or not, and show current report in low risk way. In Fig37, Fig38.

**Just Show Modified Images of This Version Below!**

The figure displays three distinct user interface panels for a supermarket system in Version 3.

**customer interface:** This panel features a large empty rectangular area at the top. Below it, a message box states, "You have refund this bill Do not do it again!". At the bottom, there is an input field and an "Enter" button.

**cashier interface:** This panel includes a login prompt "Log in as CASHIER, choose your action" above a text input field. Below this is a menu titled "Please Select Below!" with options "Check Product" and "Update Database". Further down is another input field, a toggle switch currently set to "Off" with "log in" and "log out" buttons, and a status bar at the bottom showing "Power", "PLamp" with a green indicator, and "MLamp" with a green indicator.

**database interface:** This panel contains two tables. The top table lists bill details:

Bill No.	Amount	Products	InventStatus
2	1.9	0 2 0 0 0 0 0	10 8 7 10 10 10

The bottom table shows summary statistics:

Total Bill No.	Total Sale Amount	Total Refund No.
2	1.9	1

Figure 34: Admin Add More Products to Inventory Result in Version3

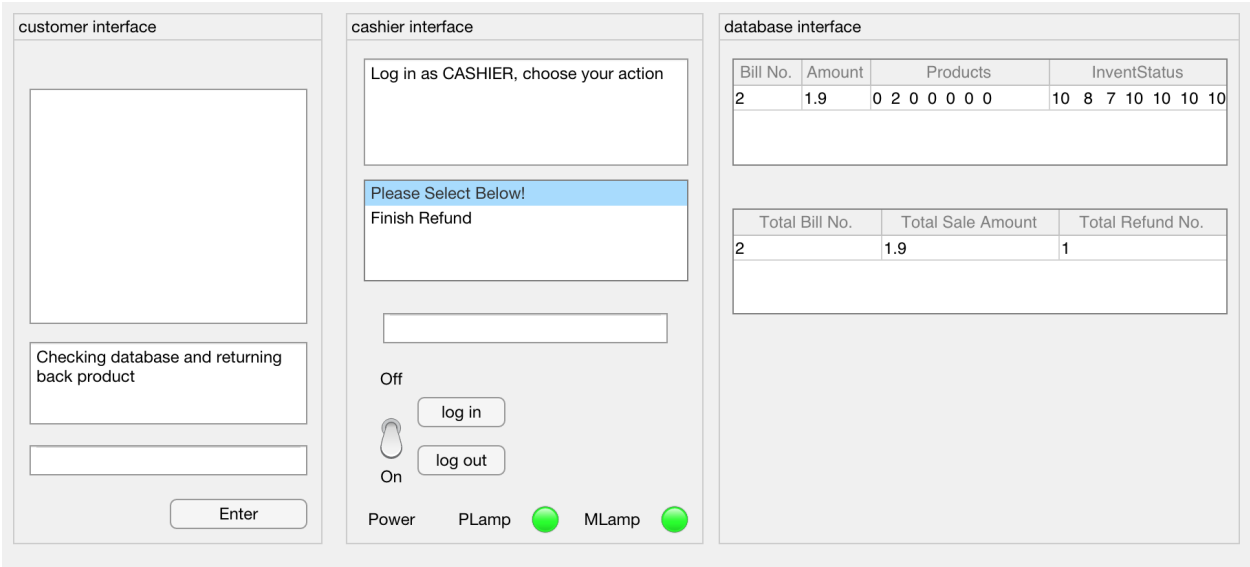


Figure 35: Cashier Could Update Database Real-time in Version3

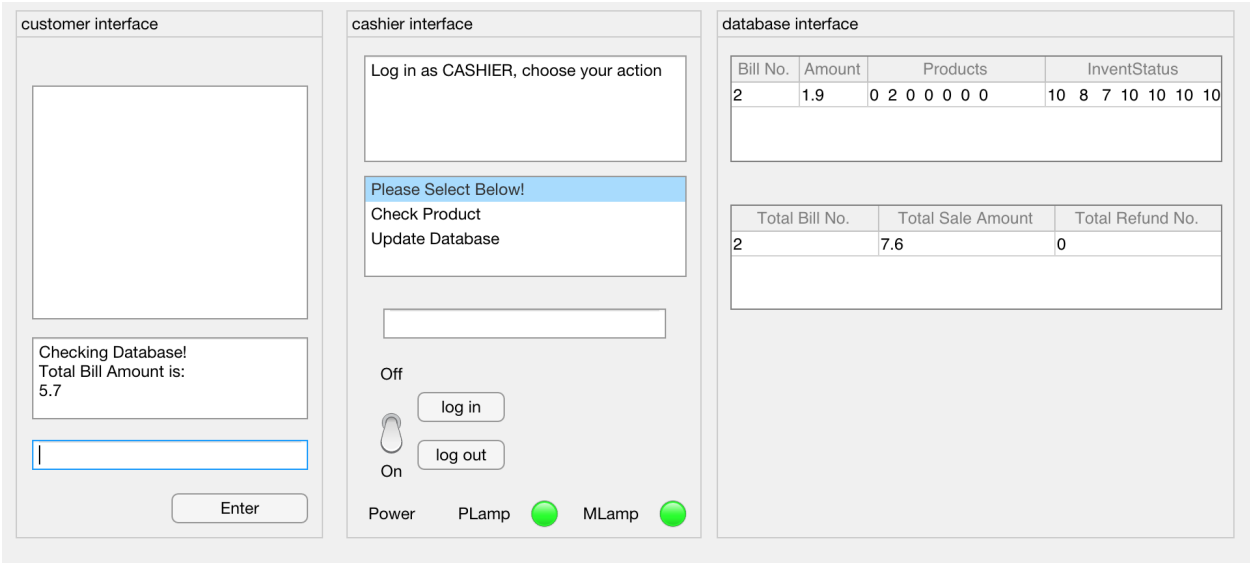


Figure 36: Detail Refund Information for Customer and Cashier in Version3

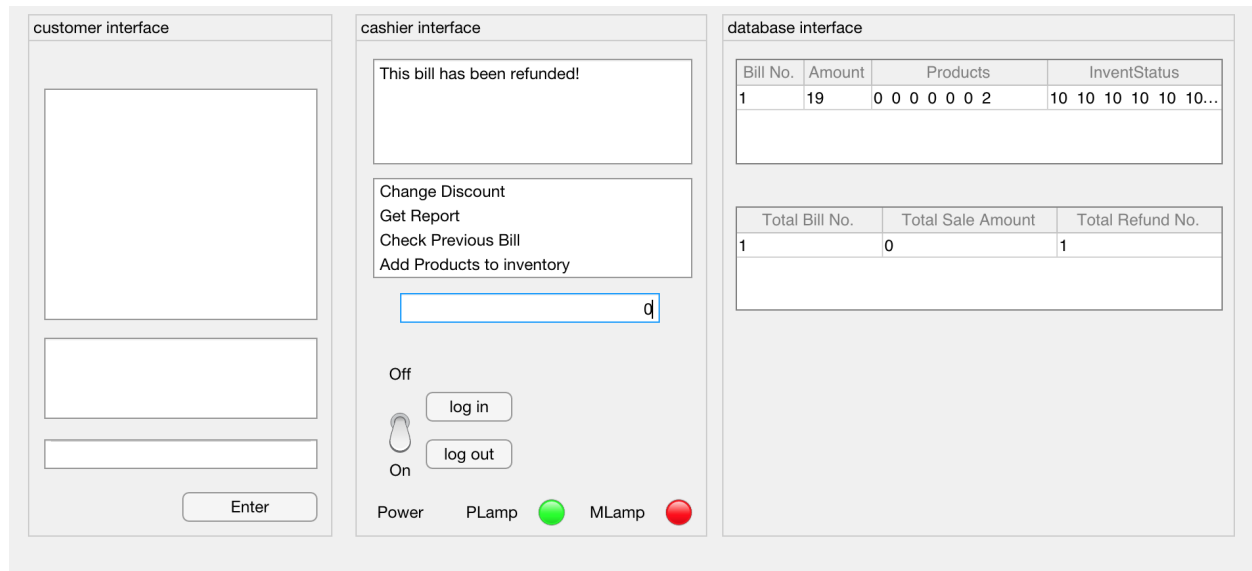


Figure 37: Admin Recognize Refunded Bill in Version3

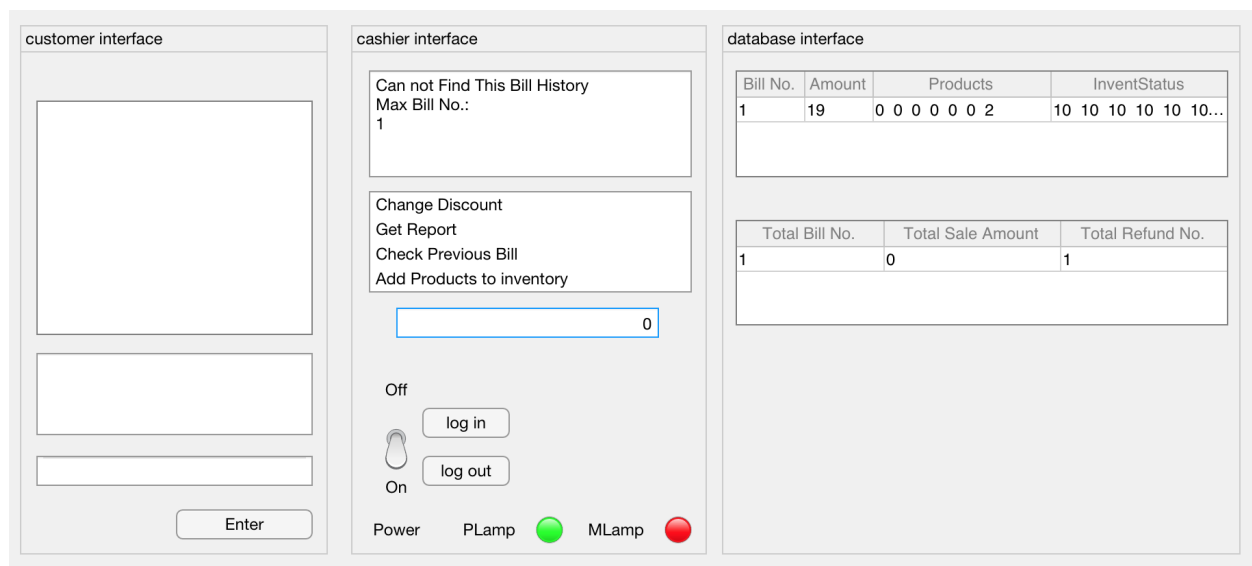


Figure 38: Admin Recognize Wrong Bill No. in Version3

## V. Validation Plan

### Test Plan

In this system, there are 3 groups of people: Cashiers, Customers, Admins.

1. We should arrange test-case among these 3 groups of people in initial.
2. Assign specific test-case to test every use case in different situations.
3. White Box and Black Box test should both be applied in this system.