

2017 Introduction to Software Engineering



Final Presentation

201611284 이유진

201611306 최병규

201611304 채민형

201411318 함형준

Table of Contents

- System Test Result
- Reaction to System Test Result
- Team 2's Specialities
- Feelings

System Test Result

Overall Result - T1's Plan

20 / 32 of Test Cases Failed

Test Coverage : 27%

Overall Result - T2's Plan

12 / 25 of Test Cases Failed

Test Coverage : 49%

Reaction to Result

Failures - Sell

POS_STC_008_000	상품이 판매 목록에 없는 바코드 입력	입력한 바코드 판매 목록에 추가	Fail
POS_STC_009_000	상품이 판매 목록에 있는 바코드 입력	입력한 바코드 판매 목록의 상품 개수 증가	Fail
POS_STC_010_000	재고 개수 보다 많은 판매 요청	"XX 개수가 부족합니다" 라는 메시지 띄움.	Fail
POS_STC_011_000	캐시 화면(터치 스크린)을 통해 금액 입력	금액이 캐시 화면에 제대로 입력되는지 확인	Fail
POS_STC_011_001	총 판매액보다 작은 금액 입력	결제 불가로 아무 일도 일어나지 않는다.	Fail
POS_STC_011_002	총 판매액보다 큰 금액 입력	거스름돈이 캐시 화면에 제대로 출력 되고 판매 완료 정보 , 영수증 , 상품파일 업데이트	Fail
POS_STC_012_000	판매 완료	캐시 화면(터치 스크린)와 고객 화면에 각각 판매 정보 띄움	Fail
POS_STC_014_000	판매 완료	당일 상품 파일 (YYYYMMDD_product.txt)의 상품의 개수 업데이트 확인	Fail
POS_STC_015_000	판매 완료	당일 상품 파일 (YYYYMMDD_management.txt)의 판매 정보를 추가한다. (재고 서버	Fail

Revisions - Sell(1)

바코드 입력

바코드를 입력해도 상품이 추가가 되지 않는다 ==>
소스 코드를 수정하여 추가되도록 수정

<pre>62 else if (cashier_input == BARCODE_SELL) 63 { 64 barcode_input = get_barcode_procdct_input(); 65 selected_product = get_product_barcode(barcode_input); 66 if (selected_product == NULL) 67 { 68 fputs("바코드 번호가 유효하지 않습니다. \n", stderr); 69 70 continue; 71 } 72 if (find_node(list, selected_product->name) != NULL) 73 { 74 - product_quantity_plus(list, find_node(list, selected_product->name)); 75 76 } 77 else 78 { 79 add_sell_node(list, selected_product); 80 } 81 }</pre>	<pre>68 else if (cashier_input == BARCODE_SELL) 69 { 70 barcode_input = get_barcode_procdct_input(); 71 selected_product = get_product_barcode(barcode_input); 72 if (selected_product == NULL) 73 { 74 fputs("바코드 번호가 유효하지 않습니다. \n", stderr); 75 + sleep(1); 76 continue; 77 } 78 79 if (find_node(list, selected_product->name) != NULL) 80 { 81 + product_quantity_plus(list, find_node(list, selected_product->name), 82 + find_inventory_node(current_inventory, 83 selected_product->name)); 84 } 85 else 86 { 87 + add_sell_node(list, selected_product); 88 + product_quantity_plus(list, find_node(list, selected_product->name), 89 + find_inventory_node(current_inventory, 90 selected_product->name)); 91 }</pre>
---	--

Revisions - Sell(1) (cont'd)

바코드 입력

바코드를 입력해도 상품이 추가가 되지 않는다 ==>
소스 코드를 수정하여 추가되도록 수정

```
131 -int product_quantity_plus(SELL_LIST *list, SELL_NODE *node)
```

```
132 {  
133     node->product_quantity += 1;  
134     list->total_price += node->product_price;
```

```
135 -  
136     return 1;  
137 }
```

```
138
```

```
136 +int product_quantity_plus(SELL_LIST *list, SELL_NODE *node, INVEN_NODE *inven)
```

```
137 {  
138     node->product_quantity += 1;  
139     list->total_price += node->product_price;
```

```
140 + inven->product->stock_amount -= 1;
```

```
141     return 1;
```

```
142 }
```

```
143
```

Revisions - Sell(2)

결제

금액 입력 포함 모든 결제 과정이 이루어지지 않는다 ==>
소스 코드를 수정하여 결제가 진행될 수 있도록 수정

```
else if (cashier_input == RECEIPT_SELL)
{
    // TODO purchasing

    print_sell_receipt(list);

    return 1;
}
```

```
91     else if (cashier_input == RECEIPT_SELL)
92     {
93         // TODO purchasing
94 +     perform_purchase(list);
95     print_sell_receipt(list);
96 +     // TODO add sale management
97 +     update_stock(current_inventory);
98     return 1;
99     }
```

Revisions - Sell(2) (cont'd)

결제

금액 입력 포함 모든 결제 과정이 이루어지지 않는다 ==>
소스 코드를 수정하여 결제가 진행될 수 있도록 수정

```
144 +void perform_purchase(SELL_LIST *list)
145 +{
146 +   int price_input;
147 +   int additional_price_input;
148 +   printf("지불할 금액을 입력하세요 : ");
149 +   scanf("%d", &price_input);
150 +
151 +   while (price_input < list->total_price)
152 +   {
153 +       printf("금액이 부족합니다. 추가로 더 지불하세요 : ");
154 +       scanf("%d", &additional_price_input);
155 +       price_input += additional_price_input;
156 +   }
157 +
158 +   system("clear");
159 +
160 +   display_cashier_screen(list, price_input);
161 +   display_customer_screen(list->total_price, price_input);
162 +   sleep(1);
163 +}
```

Revisions - Sell(3)

화면 출력

상품 리스트, 결제 화면 등의 모든 화면 출력이 없다 ==>
소스 코드를 수정하여 화면에 출력되도록 수정

```
46 +void display_sell_cashier_screen(SELL_LIST *list)
47 +{
48 +   SELL_NODE *temp = list->head->next;
49 +   printf("----- \n");
50 +   printf("|          CASHIER SCREEN          | \n");
51 +   printf("----- \n\n");
52 +
53 +   printf(" 상품명          \t수량 \t단가 \t총 판매액 \n");
54 +   printf("----- \n");
55 +   while (temp != list->tail)
56 +   {
57 +       printf(" %11s\t%2d\t%5d\t%d \n", temp->product_name, temp->product_quantity, temp-
58 +>product_price, temp->product_price * temp->product_quantity);
59 +       temp = temp->next;
60 +   }
61 +   printf("\nTotal Price : %5d \n\n", list->total_price);
62 +}
```

```
69 +void display_sell_customer_screen(SELL_LIST *list)
70 {
71 +   SELL_NODE *temp = list->head->next;
72 +   printf("----- \n");
73 +   printf("|          CUSTOMER SCREEN          | \n");
74 +   printf("----- \n\n");
75 +
76 +   printf(" 상품명          \t수량 \t단가 \t총 판매액 \n");
77 +   printf("----- \n");
78 +   while (temp != list->tail)
79 +   {
80 +       printf(" %11s\t%2d\t%5d\t%d \n", temp->product_name, temp->product_quantity, temp-
81 +>product_price, temp->product_price * temp->product_quantity);
82 +       temp = temp->next;
83 +   }
84 +   printf("\nTotal Price : %5d \n\n", list->total_price);
85 +}
```

Revisions - Sell(3) (cont'd)

화면 출력

상품 리스트, 결제 화면 등의 모든 화면 출력이 없다 ==>
소스 코드를 수정하여 화면에 출력되도록 수정

```
-----  
|          CUSTOMER SCREEN          |  
-----  
상품명      수량   단가   총 판매액  
-----  
음료수      3     1200  3600  
라면        3     800   2400  
Total Price : 6000  
  
-----  
|          CASHIER SCREEN          |  
-----  
상품명      수량   단가   총 판매액  
-----  
음료수      3     1200  3600  
라면        3     800   2400  
Total Price : 6000  
  
*****  
*          SELL STATE          *  
*****  
*  1. Increase Quantity      *  
*  2. Decrease Quantity     *  
*  3. BARCODE                *  
*  4. RECEIPT                *  
*****  
4  
지불할 금액을 입력하세요 : 5000  
금액이 부족합니다. 추가로 더 지불하세요 : 1000
```

```
-----  
|          CASHIER SCREEN          |  
-----  
총 판매액 : 6000  
현금 : 6000  
거스름돈 : 0  
상품명      수량   단가   총 판매액  
-----  
음료수      3     1200  3600  
라면        3     800   2400  
  
-----  
|          CUSTOMER SCREEN          |  
-----  
총 판매액 : 6000  
현금 : 6000  
거스름돈 : 0
```

Revisions - Sell(4)

서버 업데이트

재고 서버의 상품 개수가 변경되지 않는다 ==>

소스 코드를 수정하여 변경되도록 수정

재고 서버의 판매 내역이 추가되지 않는다 ==>

소스 코드를 수정하여 추가되도록 수정

```
81     else if (cashier_input == RECEIPT_SELL)
82     {
83         // TODO purchasing
84
85         print_sell_receipt(list);
86
87         return 1;
88     }
89     else if (cashier_input == RECEIPT_SELL)
90     {
91         // TODO purchasing
92         perform_purchase(list);
93         print_sell_receipt(list);
94         // TODO add sale management
95         update_stock(current_inventory);
96         return 1;
97     }
98 }
```

```
91     else if (cashier_input == RECEIPT_SELL)
92     {
93         // TODO purchasing
94         + perform_purchase(list);
95         print_sell_receipt(list);
96         + // TODO add sale management
97         + update_stock(current_inventory);
98         return 1;
99     }
100     else if (cashier_input == RECEIPT_SELL)
101     {
102         perform_purchase(list);
103         print_sell_receipt(list);
104         + add_sale_management(list);
105         update_stock(current_inventory);
106         return 1;
107     }
108 }
```

Revisions - Sell(4) (cont'd)

서버 업데이트

재고 서버의 상품 개수가 변경되지 않는다 ==>
소스 코드를 수정하여 변경되도록 수정

```
55 +void update_stock(INVEN_LIST *new_inven)
56 +{
57 +   char filename[30];
58 +   INVEN_NODE *curr;
59 +   FILE *fp;
60 +
61 +   strcpy(filename, get_current_date());
62 +   strcat(filename, "_product.txt");
63 +
64 +   fp = fopen(filename, "w");
65 +   if (fp == NULL)
66 +   {
67 +       fputs("ERROR: Cannot open file. \n", stderr);
68 +       return;
69 +   }
70 +
71 +   curr = new_inven->head->next;
72 +   while (curr != new_inven->tail)
73 +   {
74 +       fprintf(fp, "%s %d %s %d \n", curr->product->name, curr->product->price,
75 +             curr->product->barcode_number, curr->product->stock_amount);
76 +       curr = curr->next;
77 +   }
78 +}
79 +
```

Revisions - Sell(4) (cont'd)

서버 업데이트

재고 서버의 판매 내역이 추가되지 않는다 ==>
소스 코드를 수정하여 추가되도록 수정

```
234 +void add_sale_management(SELL_LIST *list)
235 +{
236 +   SELL_NODE *curr;
237 +   char filename[30];
238 +   FILE *fp;
239 +
240 +   strcpy(filename, get_current_date());
241 +   strcat(filename, "_sale_management.txt");
242 +
243 +   fp = fopen(filename, "a");
244 +   if (ftell(fp) == 0)
245 +   {
246 +       fprintf(fp, "%s %s", "날짜 : ", get_current_date());
247 +       fputs("\n\n", fp);
248 +   }
249 +
250 +   fputs("판매 상품 \n", fp);
251 +
252 +   curr = list->head->next;
253 +   while (curr != list->tail)
254 +   {
255 +       fprintf(fp, "%s %d %d %d\n", curr->product_name, curr->product_price, curr->product_quantity, curr-
>product_price * curr->product_quantity);
256 +       curr = curr->next;
257 +   }
258 +   fprintf(fp, "%s %d", "판매 금액 : ", list->total_price);
259 +   fputs("\n\n", fp);
260 +
261 +   fclose(fp);
```


Failures - Refund

POS_STC_017_000	환불 가능한 영수증 입력 후 환불 취소	환불 완료 안됨	Fail
POS_STC_017_001	환불 가능한 영수증 입력 후 환불	환불 완료	Fail
POS_STC_019_000	환불 완료	캐시 화면(터치 스크린)와 고객 화 면에 각각 환불 정보를 띄운다	Fail
POS_STC_020_000	환불 완료	환불 영수증(refund_ YYYYMMDDhhmm.txt)을 생성	Fail
POS_STC_021_000	환불 완료	당일 상품 파일 (YYYYMMDD_product.txt)의 상품 의 개수를 업데이트	Fail

Revisions - Refund(1)

환불 취소

환불 취소가 불가능하다 ==>

화면 출력이 안되서 확인이 불가능한 문제임을 파악
(실제로는 취소 되고 있었음)

환불 완료

환불이 정상적으로 완료되지 않는다 ==>

화면 출력이 안되서 확인이 불가능한 문제임을 파악
(실제로는 환불이 완료되었음)

Revisions - Refund(2)

화면 출력

환불 관련 정보가 화면에 출력되지 않는다 ==>
소스 코드를 수정하여 출력되도록 수정

```
168 + system("clear");
169 + display_refund_customer_screen(to_refund);
170 + display_refund_cashier_screen(to_refund);
```

```
98 +void display_refund_cashier_screen(SELL_LIST *list)
99 +{
100 + SELL_NODE *temp = list->head->next;
101 + printf("----- \n");
102 + printf("|          CASHIER SCREEN          | \n");
103 + printf("----- \n\n");
104 +
105 + printf(" 상품명          \t수량 \t단가 \t총 환불액 \n");
106 + printf("----- \n");
107 + while (temp != list->tail)
108 + {
109 +     printf(" %11s\t%2d\t%5d\t%d \n", temp->product_name, temp->product_quantity, temp->product_price, temp-
>product_price * temp->product_quantity);
110 +     temp = temp->next;
111 + }
112 + printf("\nTotal Price : %5d \n\n", list->total_price);
113 }
```

```
121 +void display_refund_customer_screen(SELL_LIST *list)
122 {
123 + SELL_NODE *temp = list->head->next;
124 + printf("----- \n");
125 + printf("|          CUSTOMER SCREEN          | \n");
126 + printf("----- \n\n");
127 + printf(" 상품명          \t수량 \t단가 \t총 환불액 \n");
128 + printf("----- \n");
129 + while (temp != list->tail)
130 + {
131 +     printf(" %11s\t%2d\t%5d\t%d \n", temp->product_name, temp->product_quantity, temp->product_price, temp-
>product_price * temp->product_quantity);
132 +     temp = temp->next;
133 + }
134 + printf("\nTotal Price : %5d \n\n", list->total_price);
```

Revisions - Refund(2) (cont'd)

화면 출력

환불 관련 정보가 화면에 출력되지 않는다 ==>
소스 코드를 수정하여 출력되도록 수정

```
-----  
|           CUSTOMER SCREEN           |  
-----  
상품명      수량   단가   총 환불액  
-----  
   커피         5     2000   10000  
  
Total Price : 10000  
  
-----  
|           CASHIER SCREEN           |  
-----  
상품명      수량   단가   총 환불액  
-----  
   커피         5     2000   10000  
  
Total Price : 10000  
  
*****  
*           REFUND STATE           *  
*****  
환불을 진행하시겠습니까? (1: 예, 2: 아니오) : █
```

Revisions - Refund(3)

영수증 생성

환불 완료 후 영수증이 생성되지 않는다 ==>

환불 완료하도록 수정하니 영수증이 정상 생성됨을 확인

```
187 +
188 +     strcpy(filename, "refund_");
189 +     strcat(filename, get_current_datetime());
190 +     strcat(filename, ".txt");
191 +
192 +     fp = fopen(filename, "w");
193 +     if (fp == NULL)
194 +     {
195 +         fputs("ERROR: Cannot open file \n", stderr);
196 +         return 0;
197 +     }
198 +
199 +     fprintf(fp, "%s %s", "영수증 번호 : ", get_current_datetime());
200 +     fputs("\n", fp);
201 +     fprintf(fp, "%s %s", "날짜 : ", get_current_date());
202 +     fputs("\n", fp);
203 +     fputs("환불 상품 \n", fp);
204 +
205 +     curr = to_refund->head->next;
206 +     while(curr != to_refund->tail)
207 +     {
208 +         fprintf(fp, "%s %d %d %d", curr->product_name, curr->product_price, curr->product_quantity, curr-
>product_price * curr->product_quantity);
209 +         fputs("\n", fp);
210 +         curr = curr->next;
211 +     }
212 +     fprintf(fp, "%s %d", "총 환불액 : ", to_refund->total_price);
213 +     fputs("\n", fp);
214 +
215 +     fclose(fp);
216 + }
```

Revisions - Refund(4)

서버 업데이트

환불 완료 후 재고 서버의 상품 개수가 변경되지 않는다 ==>
환불 완료하도록 수정 후, 소스 코드를 수정하여 변경되도록 수정

```
175 +   if (get_cashier_input())
176 +   {
177 +       SELL_NODE *curr;
178 +       char filename[30];
179 +
180 +       /* restore server */
181 +       curr = to_refund->head->next;
182 +       while(curr != to_refund->tail)
183 +       {
184 +           find_inventory_node(current_inventory, curr->product_name)->product->stock_amount += curr->product_quantity;
185 +           curr = curr->next;
186 +       }
```

Failures - Daily Report

POS_STC_022_000	3분 대기	하루 동안 판매 했던 내역을 settle_YYYYMMDD.txt로 파일을 생성한다.	Fail
POS_STC_023_000	판매 상태에서 대기	정산 불가 확인	Fail
POS_STC_023_001	환불 상태에서 대기	정산 불가 확인	Fail

Revisions - Daily Report(1)

정산 파일 생성

하루가 지나도 정산 파일을 생성하지 않는다 ==>
소스 코드를 수정하여 생성하도록 수정

```
후 @@ -113,8 +119,11 @@ void print_daily_report()
```

```
113     /* generating command */  
114     strncpy(command, "cp ", 3);  
115     strcat(command, datetime);
```

```
116     strcat(command, file_name);  
117
```

```
118     /* executing command */  
119     system(command);  
120 }
```

```
119     /* generating command */  
120     strncpy(command, "cp ", 3);  
121     strcat(command, datetime);  
122 +   strcat(command, " ");  
123     strcat(command, file_name);  
124
```

```
125 +   printf("%s \n", command);  
126 +
```

```
127     /* executing command */  
128     system(command);  
129 }
```


Revisions - Daily Report(2)

판매 / 환불 상태에서 대기

판매 / 환불 상태에서도 정산이 이루어진다 ==>

소스 코드를 수정하여 정산 지연하도록 수정

```
14 src/daily.c
@@ -41,7 +41,9 @@ void *time_function(void *data)
41     }
42     if (hour > 23)
43     {
44 -     print_daily_report();
45     flag = 1;
46     date++;
47     hour = 0;
@@ -75,11 +77,11 @@ void *time_function(void *data)

5 src/main.c
@@ -119,6 +119,11 @@ int main(int argc, char **argv)
119     fputs("ERROR : Invalid Input \n", stderr);
120     }
121
122     sleep(1);
123 }
124

@@ -119,6 +119,11 @@ int main(int argc, char **argv)
119     fputs("ERROR : Invalid Input \n", stderr);
120     }
121
122 +     if (status == 2)
123 +     {
124 +     print_daily_report();
125 +     }
126 +
127     sleep(1);
128 }
129
```

Failures - Stock

POS_STC_025_000	재고정보확인 입력	재고 정보상태 캐쉬화면에 출력	Fail
POS_STC_026_000	재고정보확인 입력	당일 재고 정보 stock_YYYYMMDD.txt 파일을 생성 한다	Fail

Revisions - Stock

화면 출력

재고 확인 상태가 화면에 출력되지 않는다 ==>
테스트 재 실행 결과 정상 출력 확인

파일 생성

재고 확인 파일이 생성되지 않는다 ==>
테스트 재 실행 결과 정상 출력 확인

Revisions - Stock (cont'd)

재고 확인 결과
캐시 화면 출력

파일 정상 생성 확인

```
*****  
* POINT OF SALE TERMINAL *  
*****  
* 1. Turn Off machine *  
* 2. SELL *  
* 3. REFUND *  
* 4. CHECK STOCK *  
*****  
4  
상품명 : 과자 -> 수량 : 100  
상품명 : 아이스크림 -> 수량 : 100  
상품명 : 과일 -> 수량 : 100  
상품명 : 물 -> 수량 : 100  
상품명 : 라면 -> 수량 : 100  
상품명 : 음료수 -> 수량 : 100  
상품명 : 커피 -> 수량 : 94
```

```
gyukebox ~/s/2/s/t/src > master • ls -a | grep stock  
stock_20171203.txt  
stock_20171204.txt  
stock_20171206.txt  
gyukebox ~/s/2/s/t/src > master • cat stock_20171206.txt  
상품명 : 과자 가격 : 1000 수량 : 100  
상품명 : 아이스크림 가격 : 1500 수량 : 100  
상품명 : 과일 가격 : 3000 수량 : 100  
상품명 : 물 가격 : 500 수량 : 100  
상품명 : 라면 가격 : 800 수량 : 100  
상품명 : 음료수 가격 : 1200 수량 : 100  
상품명 : 커피 가격 : 2000 수량 : 94  
gyukebox ~/s/2/s/t/src > master •
```

build passing

coverage 100%

Team 2's Specialties

Makefile

Script to generate executable file

```
gyukebox ~/s/2/s/t/src > master • make post
gcc -g main.c display.c input.c control.c daily.c sell.c refund.c inven.c -o post_executable
gyukebox ~/s/2/s/t/src > master • make clean;
rm -rf post_executable post_executable.dSYM
gyukebox ~/s/2/s/t/src > master • cat Makefile
.SUFFIXES = .c .o

SRCS = main.c display.c input.c control.c daily.c sell.c refund.c inven.c

CC = gcc
CFLAGS = -g

post:
    $(CC) $(CFLAGS) $(SRCS) -o post_executable

clean:
    rm -rf post_executable post_executable.dSYM
gyukebox ~/s/2/s/t/src > master •
```

Collaboration with git

66 commits made by 3 people

No description, website, or topics provided.

Edit

[Add topics](#)

66 commits

4 branches

0 releases

3 contributors

GPL-3.0

Branch: master ▾


New pull request

Create new file

Upload files

Find file

Clone or download ▾

 gyukebox 서버 연동 추가, 결제 모듈 추가

Latest commit 85e9550 5 hours ago

 images	README 수정 & 사진 추가	19 days ago
 src	서버 연동 추가, 결제 모듈 추가	5 hours ago
 tests	유닛 테스트 커버리지 100%!!	20 days ago
 .gitignore	modified test code	22 days ago
 LICENSE	merge from origin/master	22 days ago
 README.md	Update README.md	19 days ago

Collaboration with git (cont'd)

Issues to report problem(s)

<> Code

🔔 Issues 1

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📊 Insights

⚙️ Settings

하루 지난 후 서버 연동이 안됨 #9

🔔 Open

gyukebox opened this issue 5 hours ago · 0 comments



gyukebox commented 5 hours ago

Owner



하루가 지난 후(정산 출력한 후)
예를들어 라면 5개를 팔았다 하면,
라면이 100개에서 95개로 바뀌어야 하는데 그렇지 못함

Collaboration with git (cont'd)

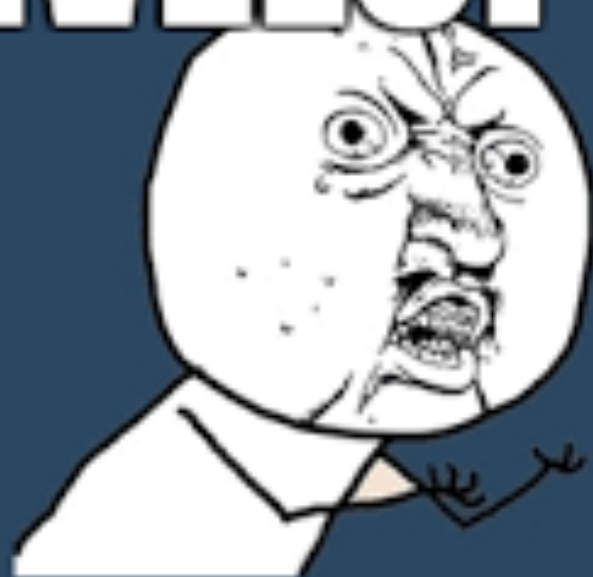
Pull Requests to submit works

Merging to accept others' works

The screenshot shows a GitHub Pull Request interface for a repository named 'check_stock #7'. The navigation bar at the top includes links for Code, Issues (1), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The pull request title is 'check_stock #7' with an 'Edit' button. A status bar indicates it is 'Merged' by 'gyukebox' 22 days ago, merging 5 commits from 'mh_branch' into 'master'. Below this, there are statistics for Conversation (0), Commits (5), and Files changed (6), along with a color-coded diff indicator showing +64 lines added and -76 lines removed. The main content area shows a comment from 'MinHyungChae' 22 days ago with no description provided. Below the comment is a commit history for Nov 4, listing five commits: 'inventory Check 미완성' (8c61209), 'no message' (4543549), '재고 확인' (1236d61), 'Merge branch 'master' into mh_branch' (064f18a), and 'README / LICENSE 재추가' (86b2a25). A merge summary shows 'gyukebox merged commit add1966 into master' 22 days ago, with a 'Revert' button. At the bottom, a notification states 'Pull request successfully merged and closed' and 'You're all set—the mh_branch branch can be safely deleted.', with a 'Delete branch' button. On the right side, there are settings for Reviewers (No reviews—request one), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), and Notifications (Unsubscribe).

Feelings

DEVELOPER



Y U NO WRITE TESTS?

Demo