

2017 소프트웨어 공학 개론

POS SYSTEM SDS

201211337 김재현
201112052 방민석
201312259 백만일
201211383 조영래

1. DFD faults

2. DFD supplement

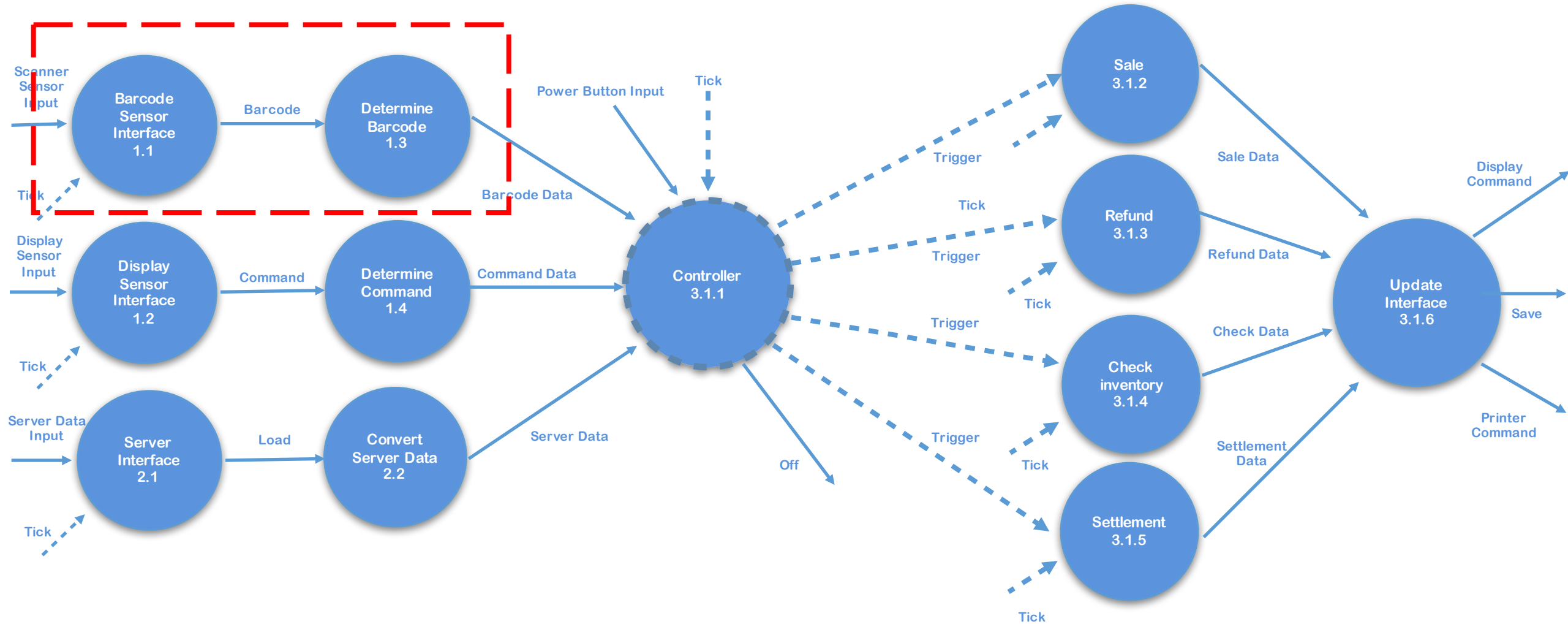
3. UTP/UTR

4. Q&A



DFD faults

1. Lack of Purchase Data List



1. DFD faults

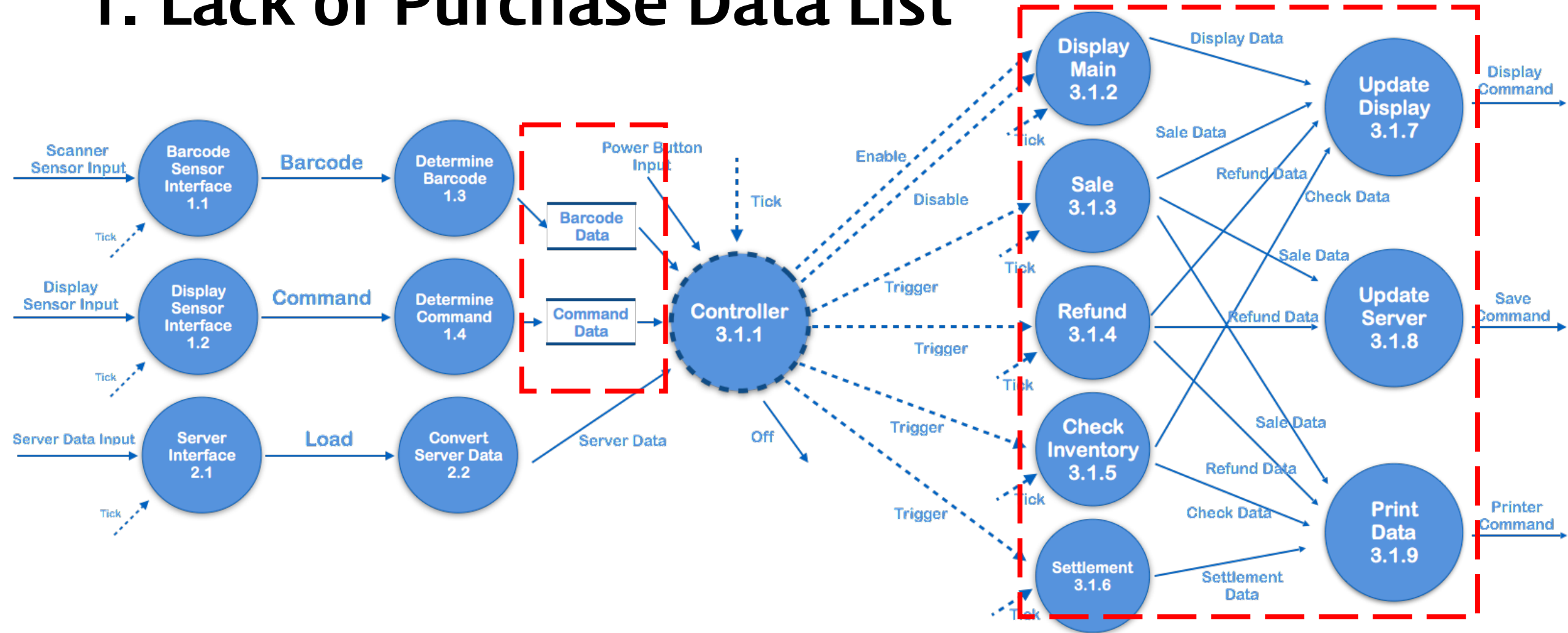
2. DFD supplement

3. UTP/UTR

4. Q&A

DFD supplement

1. Lack of Purchase Data List

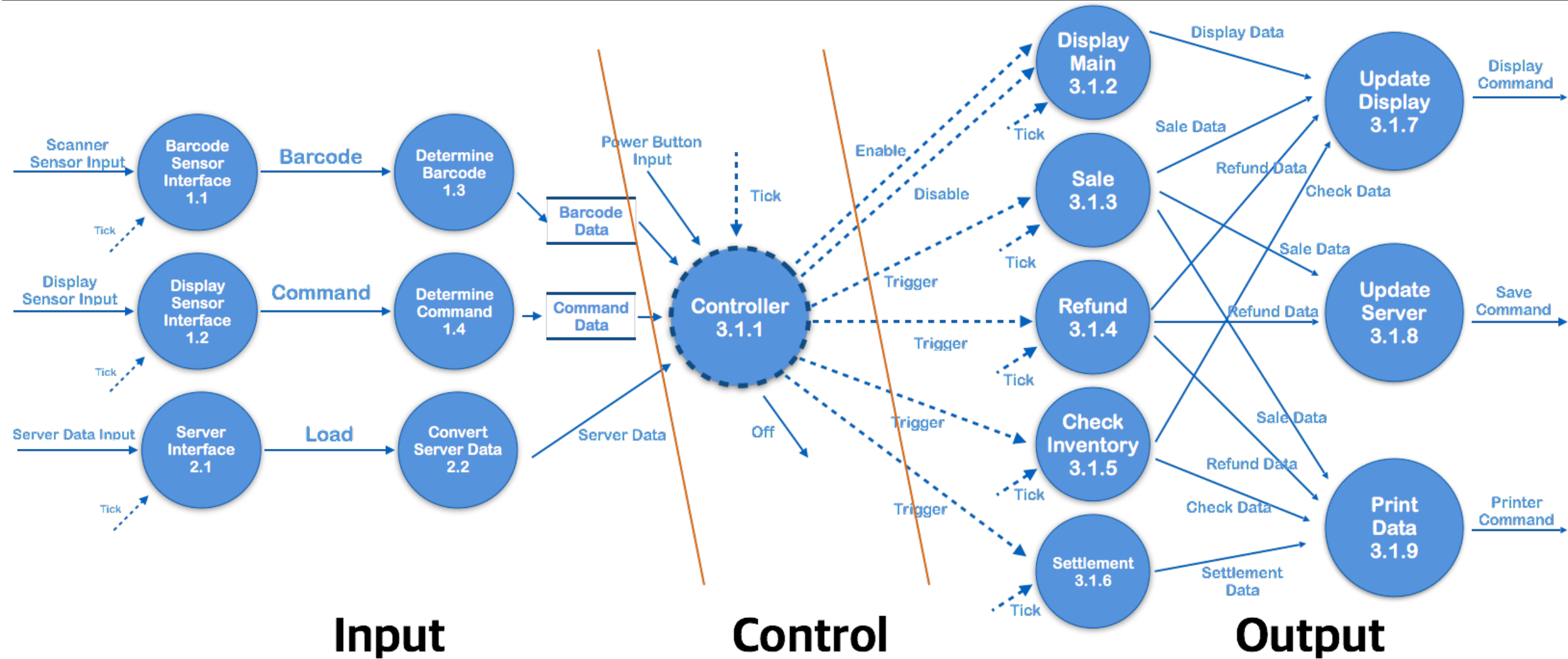


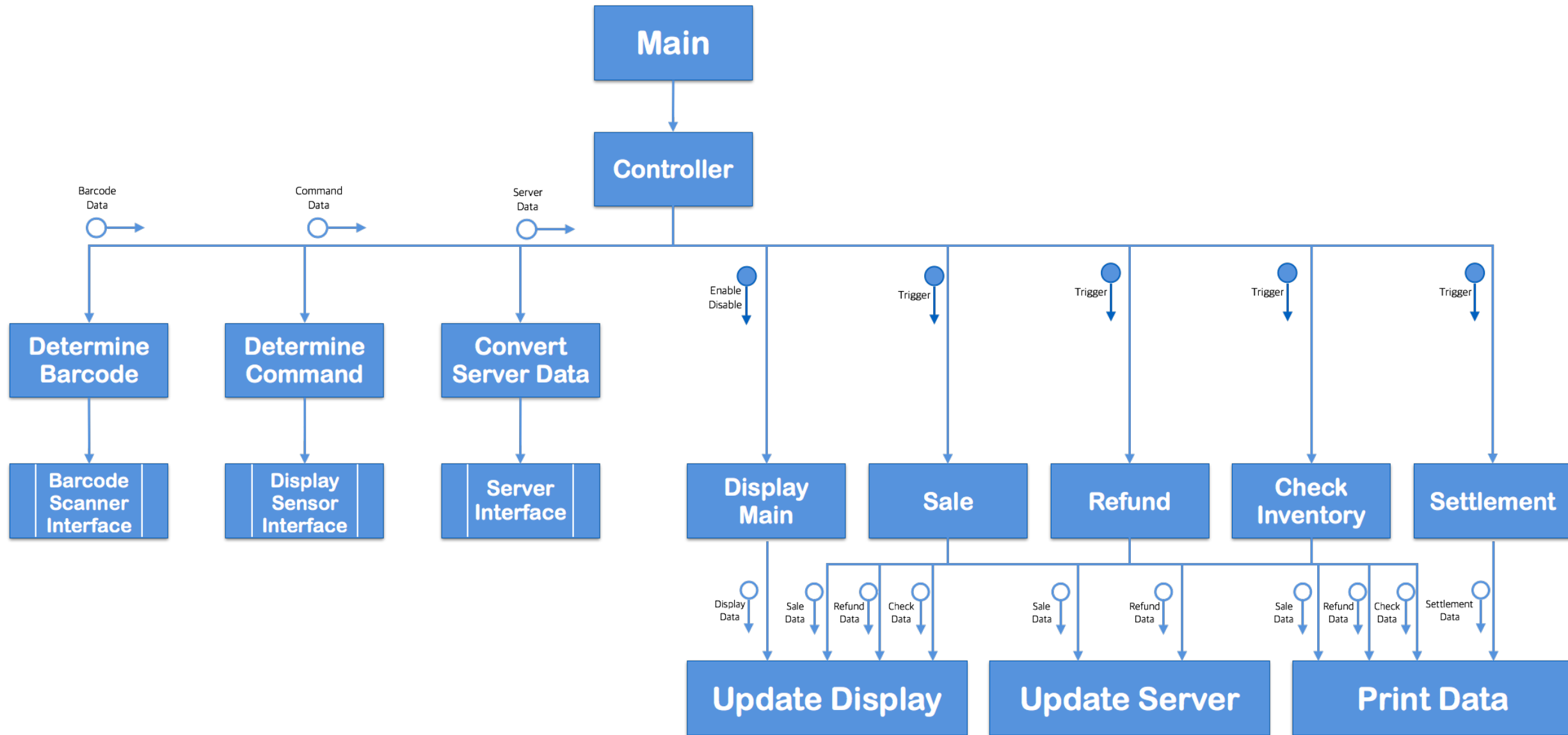
1. DFD faults

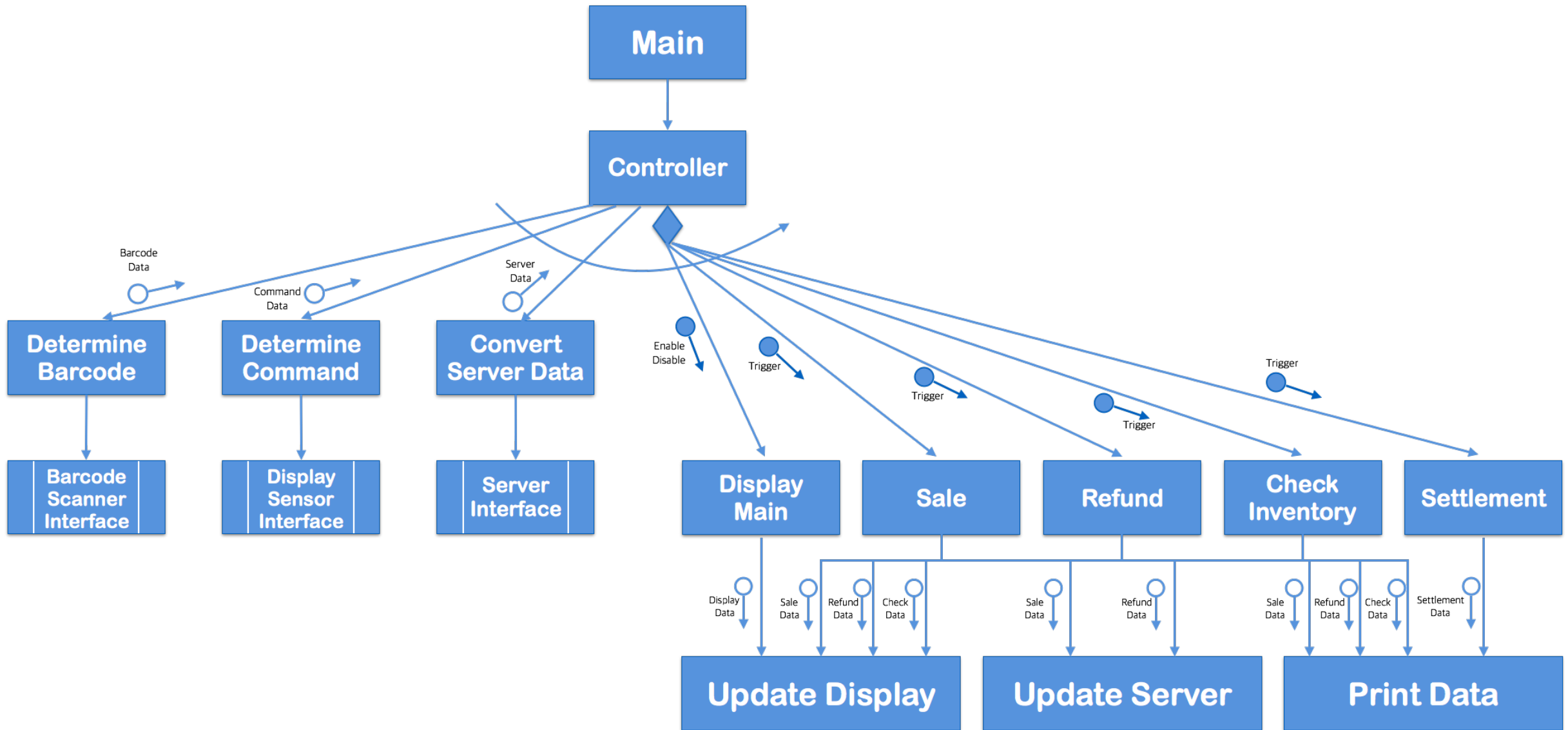
2. DFD supplement

3. UTP/UTR

4. Q&A







1. Feature to be tested

- 1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지 테스트한다.
- 2) Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를 테스트 한다.

2. DFD Test process list

| ID | Name | Description |
|-------|---------------------|---|
| 1.3 | Determine Barcode | 입력받은 데이터를 Barcode Data형태로 변환한다. |
| 1.4 | Determine Command | 디스플레이를 통해 입력 받은 데이터를 Command Data형태로 변환한다. |
| 2.2 | Convert Server Data | 서버로부터 받은 데이터를 Server Data형태로 변환한다. |
| 3.1.1 | Controller | 입력된 Data를 Command Data를 바탕으로 프로세스들을 트리거 한다. |
| 3.1.2 | Display Main | 입력 데이터를 Display Data형태로 만든다. |
| 3.1.3 | Sale | 판매 될 수량을 서버와 비교해서 판매 가능하면 Sale Data형태로 만든다. |

2. DFD Test process list(Cont.)

| | | |
|-------|-----------------|---|
| 3.1.4 | Refund | 환불 물품을 서버와 비교해서 환불 가능하면 Refund Data 형태로 만든다. |
| 3.1.5 | Check Inventory | 서버에서 재고 목록을 받아온다. |
| 3.1.6 | Settlement | 서버에서 판매, 환불 영수증을 읽어와 Settlement Data 형태로 만든다. |
| 3.1.7 | Update Display | 계산된 Data를 Display Data 형태로 만들어 Display Interface에 전달한다. |
| 3.1.8 | Update Server | 계산된 Data를 Server Data 형태로 만들어 Server Interface에 전달한다 |
| 3.1.9 | Print Data | 계산된 Data를 Printer Interface에 전달한다 |

3. Feature not to be tested

- 1) 전달 역할, 단순한 프로세스, UI 구성 프로세스, 특수한 상황에서만 작동하는 프로세스는 테스트에서 제외한다.

4. UTP (Test Design Identification)

| Identifier | Feature (Process DFD) | Valid / Invalid value |
|-----------------|-----------------------|---|
| POS_BAR_DATA_01 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "001", -1} |
| POS_BAR_DATA_02 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "010", -1} |
| POS_BAR_DATA_03 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "011", -1} |
| POS_BAR_DATA_04 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "100", -1} |
| POS_BAR_DATA_05 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "101", -1} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|-----------------|-----------------------|---|
| POS_BAR_DATA_06 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "110", -1} |
| POS_BAR_DATA_07 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "111", -1} |
| POS_BAR_DATA_08 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "000", -1} |
| POS_BAR_DATA_09 | 1.3 Determine Barcode | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "r", barcode, -1} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|---------------------|-----------------------|---|
| POS_COMMAND_DATA_01 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'b', &server, &list, -1} |
| POS_COMMAND_DATA_02 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'s', &server, &list, 1} |
| POS_COMMAND_DATA_03 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'r', &server, &list, 2} |
| POS_COMMAND_DATA_04 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'C', &server, &list, 3} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|---------------------|-----------------------|---|
| POS_COMMAND_DATA_05 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'i', &server, &list, -1} |
| POS_COMMAND_DATA_06 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'c', &server, &list, -1} |
| POS_COMMAND_DATA_07 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'e', &server, &list, 0} |
| POS_COMMAND_DATA_08 | 1.4 Determine Command | determine_command(COMMAND, SERVER, LIST, state) : {'x', &server, &list, -1} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|--------------------|-------------------------|--|
| POS_SERVER_DATA_01 | 2.2 Convert Server Data | <code>convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,0}</code> |
| POS_SERVER_DATA_02 | 2.2 Convert Server Data | <code>convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,1}</code> |
| POS_SERVER_DATA_03 | 2.2 Convert Server Data | <code>convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,2}</code> |

4. UTP (Test Design Identification)(Cont.)

| | | |
|-------------------|------------------|--------------------------|
| POS_Controller_01 | 3.1.1 Controller | selected_state {-1} |
| POS_Controller_02 | 3.1.1 Controller | selected_state {0} |
| POS_Controller_03 | 3.1.1 Controller | selected_state {1} |
| POS_Controller_04 | 3.1.1 Controller | selected_state {2} |
| POS_Controller_05 | 3.1.1 Controller | selected_state {3} |
| POS_Controller_06 | 3.1.1 Controller | selected_state {4} |
| POS_Controller_07 | 3.1.1 Controller | selected_state {int_max} |

4. UTP (Test Design Identification)(Cont.)

POS_Display_Main_01

3.1.2 Display Main

```
display_main(DISPLAY) : {  
&display}
```

4. UTP (Test Design Identification)(Cont.)

| | | |
|------------------|------------|---|
| POS_SALE_DATA_01 | 3.1.3 Sale | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, list, server, sale, 1, display} |
| POS_SALE_DATA_02 | 3.1.3 Sale | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {0, 0, server, sale, 1, display} |
| POS_SALE_DATA_03 | 3.1.3 Sale | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, 0, server, sale, 1, display} |
| POS_SALE_DATA_04 | 3.1.3 Sale | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, 0, 0, sale, 1, display} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|--------------------|-----------------------|---|
| POS_REFUND_DATA_01 | 3.1.4 Refund | Refund(LIST, SERVER, REFUND, state, DISPLAY) : {&list, &server, ,&refund, 2, display} |
| POS_CHECK_DATA_01 | 3.1.5 Check Inventory | Check(SERVER, state, DISPLAY) : {&server, 3 , &display} |
| POS_SETTLEMENT_01 | 3.1.6 Settlement | settlement(LIST, SERVER, SETTLEMENT, state) : {&list, &server, &settlement, 4} |
| POS_SETTLEMENT_02 | 3.1.6 Settlement | settlement(LIST, SERVER, SETTLEMENT, state) : {0, &server, &settlement, 4} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|-----------------------|----------------------|--|
| POS_Update_Display_01 | 3.1.7 Update Display | update_display(SALE, REFUND, SERVER, DISPLAY) : {1, 0, 0, 0} |
| POS_Update_Display_02 | 3.1.7 Update Display | update_display(SALE, REFUND, SERVER, DISPLAY) : {0, 1, 0, 0} |
| POS_Update_Display_03 | 3.1.7 Update Display | update_display(SALE, REFUND, SERVER, DISPLAY) : {0, 0, 1, 0} |
| POS_Update_Server_01 | 3.1.8 Update Server | update_server(SERVER, LIST) : {&server, &list} |
| POS_Update_Server_02 | 3.1.8 Update Server | update_server(SERVER, LIST) : {&server, 0} |

4. UTP (Test Design Identification)(Cont.)

| | | |
|-------------------|------------------|---|
| POS_PRINT_DATA_01 | 3.1.9 Print Data | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {&sale, 0, 0, 0} |
| POS_PRINT_DATA_02 | 3.1.9 Print Data | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, &refund, 0, 0} |
| POS_PRINT_DATA_03 | 3.1.9 Print Data | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, &server, 0} |
| POS_PRINT_DATA_04 | 3.1.9 Print Data | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, 0, &settlement} |

4. UTR (Test Case Identification)

| Identifier | Input Specification | Output Specification |
|-----------------|---|-----------------------|
| POS_BAR_DATA_01 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "001", -1} | Barcode Data(product) |
| POS_BAR_DATA_02 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "010", -1} | Barcode Data(product) |
| POS_BAR_DATA_03 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "011", -1} | Barcode Data(product) |
| POS_BAR_DATA_04 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "100", -1} | Barcode Data(product) |

4. UTR (Test Case Identification)(Cont.)

| | | |
|-----------------|---|---------------------------|
| POS_BAR_DATA_05 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "101", -1} | Barcode Data(product) |
| POS_BAR_DATA_06 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "110", -1} | Barcode Data(product) |
| POS_BAR_DATA_07 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "111", -1} | Barcode Data(product) |
| POS_BAR_DATA_08 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "b", "000", -1} | Error(Exception Handling) |
| POS_BAR_DATA_09 | determine_barcode(LIST, SERVER, COMMAND, barcode, state) : {&list, &server, "r", barcode, -1} | Barcode Data(receipt) |

4. UTR (Test Case Identification)

| | | |
|---------------------|---|-------------------------|
| POS_COMMAND_DATA_01 | determine_command(COMMAND, SERVER, LIST, state) : {'b', &server, &list, -1} | Command = Input Barcode |
| POS_COMMAND_DATA_02 | determine_command(COMMAND, SERVER, LIST, state) : {'s', &server, &list, 1} | Command = Sale |
| POS_COMMAND_DATA_03 | determine_command(COMMAND, SERVER, LIST, state) : {'r', &server, &list, 2} | Command = Refund |
| POS_COMMAND_DATA_04 | determine_command(COMMAND, SERVER, LIST, state) : {'C', &server, &list, 3} | Command = Check |

4. UTR (Test Case Identification)

| | | |
|---------------------|---|------------------------------|
| POS_COMMAND_DATA_05 | determine_command(COMMAND, SERVER, LIST, state) : {'i', &server, &list, -1} | Command = Input Cash |
| POS_COMMAND_DATA_06 | determine_command(COMMAND, SERVER, LIST, state) : {'c', &server, &list, -1} | Command = product_amt change |
| POS_COMMAND_DATA_07 | determine_command(COMMAND, SERVER, LIST, state) : {'e', &server, &list, 0} | Command = Exit |
| POS_COMMAND_DATA_08 | determine_command(COMMAND, SERVER, LIST, state) : {'x', &server, &list, -1} | Error(Exception Handling) |

4. UTR (Test Case Identification)

| | | |
|---------------------|---|------------------------------|
| POS_COMMAND_DATA_05 | determine_command(COMMAND, SERVER, LIST, state) : {'i', &server, &list, -1} | Command = Input Cash |
| POS_COMMAND_DATA_06 | determine_command(COMMAND, SERVER, LIST, state) : {'c', &server, &list, -1} | Command = product_amt change |
| POS_COMMAND_DATA_07 | determine_command(COMMAND, SERVER, LIST, state) : {'e', &server, &list, 0} | Command = Exit |
| POS_COMMAND_DATA_08 | determine_command(COMMAND, SERVER, LIST, state) : {'x', &server, &list, -1} | Error(Exception Handling) |

4. UTR (Test Case Identification)

| | | |
|--------------------|--|-------------|
| POS_SERVER_DATA_01 | <code>convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,0}</code> | Server Data |
| POS_SERVER_DATA_02 | <code>convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,1}</code> | Server Data |
| POS_SERVER_DATA_03 | <code>convert_server(SERVER, LIST, receipt_num, type) : {&server, &list, receipt_num,2}</code> | Server Data |

4. UTR (Test Case Identification)

| | | |
|-------------------|--------------------------|---------------------------|
| POS_Controller_01 | selected_state {-1} | Input Barcode |
| POS_Controller_02 | selected_state {0} | Wait |
| POS_Controller_03 | selected_state {1} | Sale |
| POS_Controller_04 | selected_state {2} | Refund |
| POS_Controller_05 | selected_state {3} | Check |
| POS_Controller_06 | selected_state {4} | Settlement |
| POS_Controller_07 | selected_state {int_max} | Error(Exception Handling) |

4. UTR (Test Case Identification)

| | | |
|-------------------|--------------------------|---------------------------|
| POS_Controller_01 | selected_state {-1} | Input Barcode |
| POS_Controller_02 | selected_state {'0} | Wait |
| POS_Controller_03 | selected_state {1} | Sale |
| POS_Controller_04 | selected_state {2} | Refund |
| POS_Controller_05 | selected_state {3} | Check |
| POS_Controller_06 | selected_state {4} | Settlement |
| POS_Controller_07 | selected_state {int_max} | Error(Exception Handling) |

4. UTR (Test Case Identification)

POS_Display_Main_01

display_main(DISPLAY)
: {&display}

Display Data

4. UTR (Test Case Identification)

| | | |
|--------------------|---|--|
| POS_SALE_DATA_01 | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, list, server, sale, 1, display} | Sale Data |
| POS_SALE_DATA_02 | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {0, list, server, sale, 1, display} | Error(Exception Handling) r_price < total_price |
| POS_SALE_DATA_03 | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, 0, server, sale, 1, display} | Error(Exception Handling) List x |
| POS_SALE_DATA_04 | sale(r_price, LIST, SERVER, SALE, state, DISPLAY) : {r_price, list, 0, sale, 1, display} | Error(Exception Handling) p_amt > server_stock |
| POS_REFUND_DATA_01 | Refund(LIST, SERVER, REFUND, state, DISPLAY) : {&list, &server, ,&refund, 2, display} | Refund Data |

4. UTR (Test Case Identification)

| | | |
|-------------------|---|-------------------------------------|
| POS_CHECK_DATA_01 | Check(SERVER, state, DISPLAY) : {&server, 3 , &display} | Check Data |
| POS_SETTLEMENT_01 | settlement(LIST, SERVER, SETTLEMENT, state) : {list, &server, &settlement, 4} | Settlement Data |
| POS_SETTLEMENT_02 | settlement(LIST, SERVER, SETTLEMENT, state) : {0, &server, &settlement, 4} | Error(Exception Handling) List x |

4. UTR (Test Case Identification)

| | | |
|---------------------|---|------------------------------|
| POS_COMMAND_DATA_05 | determine_command(COMMAND, SERVER, LIST, state) : {'i', &server, &list, -1} | Command = Input Cash |
| POS_COMMAND_DATA_06 | determine_command(COMMAND, SERVER, LIST, state) : {'c', &server, &list, -1} | Command = product_amt change |
| POS_COMMAND_DATA_07 | determine_command(COMMAND, SERVER, LIST, state) : {'e', &server, &list, 0} | Command = Exit |
| POS_COMMAND_DATA_08 | determine_command(COMMAND, SERVER, LIST, state) : {'x', &server, &list, -1} | Error(Exception Handling) |

4. UTR (Test Case Identification)

| | | |
|-----------------------|--|-------------------------------------|
| POS_Update_Display_01 | update_display(SALE,REFUND, SERVER , DISPLAY) : {1, 0, 0, 0} | Display Command |
| POS_Update_Display_02 | update_display(SALE,REFUND, SERVER , DISPLAY) : {0, 1, 0, 0} | Display Command |
| POS_Update_Display_03 | update_display(SALE,REFUND, SERVER , DISPLAY) : {0, 0, 1, 0} | Display Command |
| POS_Update_Server_01 | update_server(SERVER, LIST) : {&server, &list} | Save |
| POS_Update_Server_02 | update_server(SERVER, LIST) : {&server, 0} | Error(Exception Handling) list x |

4. UTR (Test Case Identification)

| | | |
|-------------------|---|-----------------|
| POS_PRINT_DATA_01 | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {&sale, 0, 0, 0} | Printer Command |
| POS_PRINT_DATA_02 | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, &refund, 0, 0} | Printer Command |
| POS_PRINT_DATA_03 | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, &server, 0} | Printer Command |
| POS_PRINT_DATA_04 | print_data(SALE, REFUND, SERVER, SETTLEMENT) : {0, 0, 0, &settlement} | Printer Command |

5. Result

Total test case : 46개

Passed : 46개

Failed : 0개

1. DFD faults

2. DFD supplement

3. SD of POS system

4. Q&A

Q&A