

SASD & RM+

정세진
김재엽

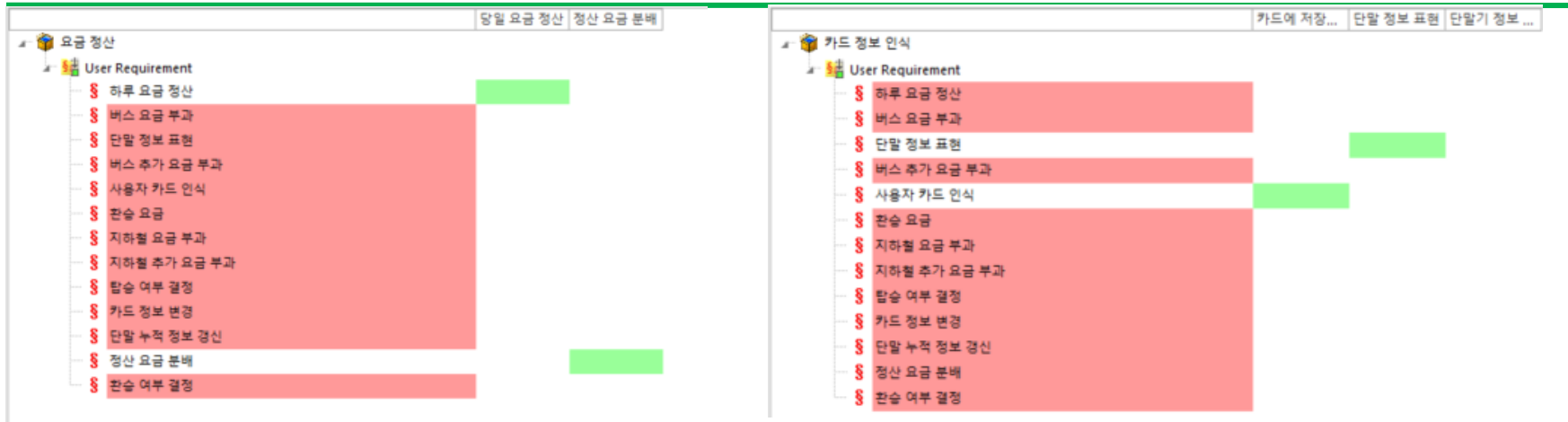
1. RM+ + SA
2. RM+ + SD
3. Test case

INDEX

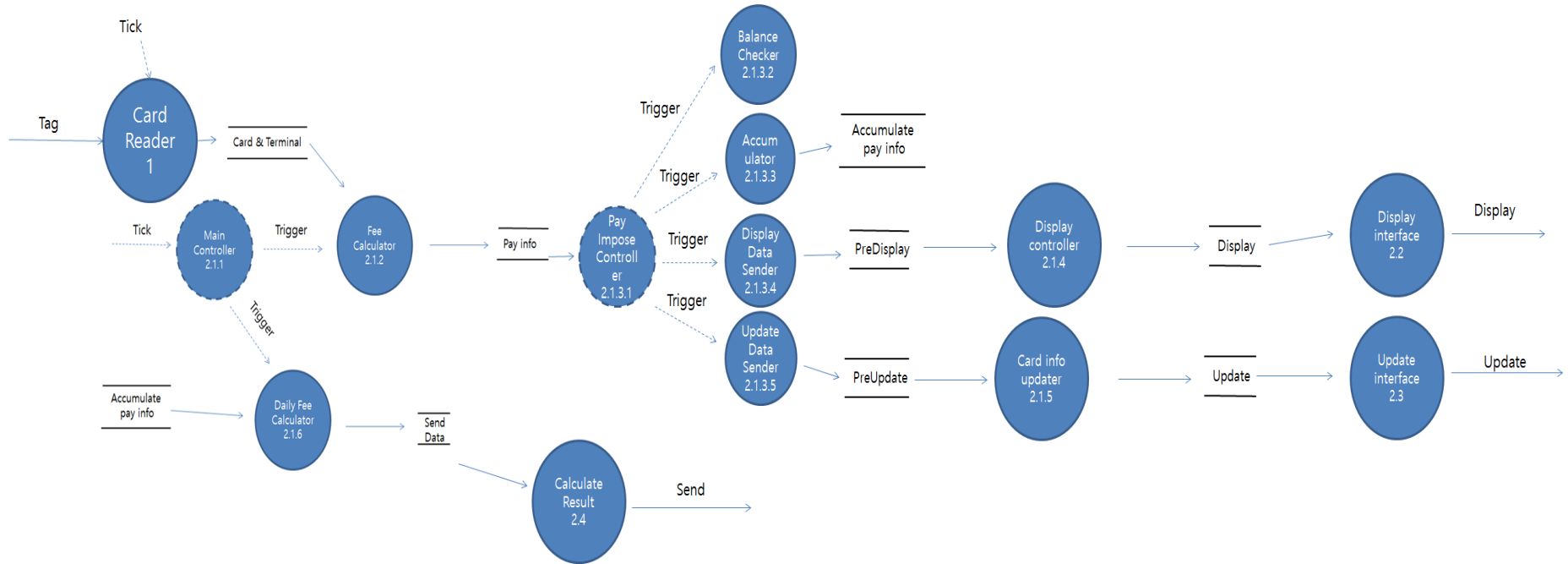
Introduce RM+ + SA

RM+ + SA

User Requirements



SA

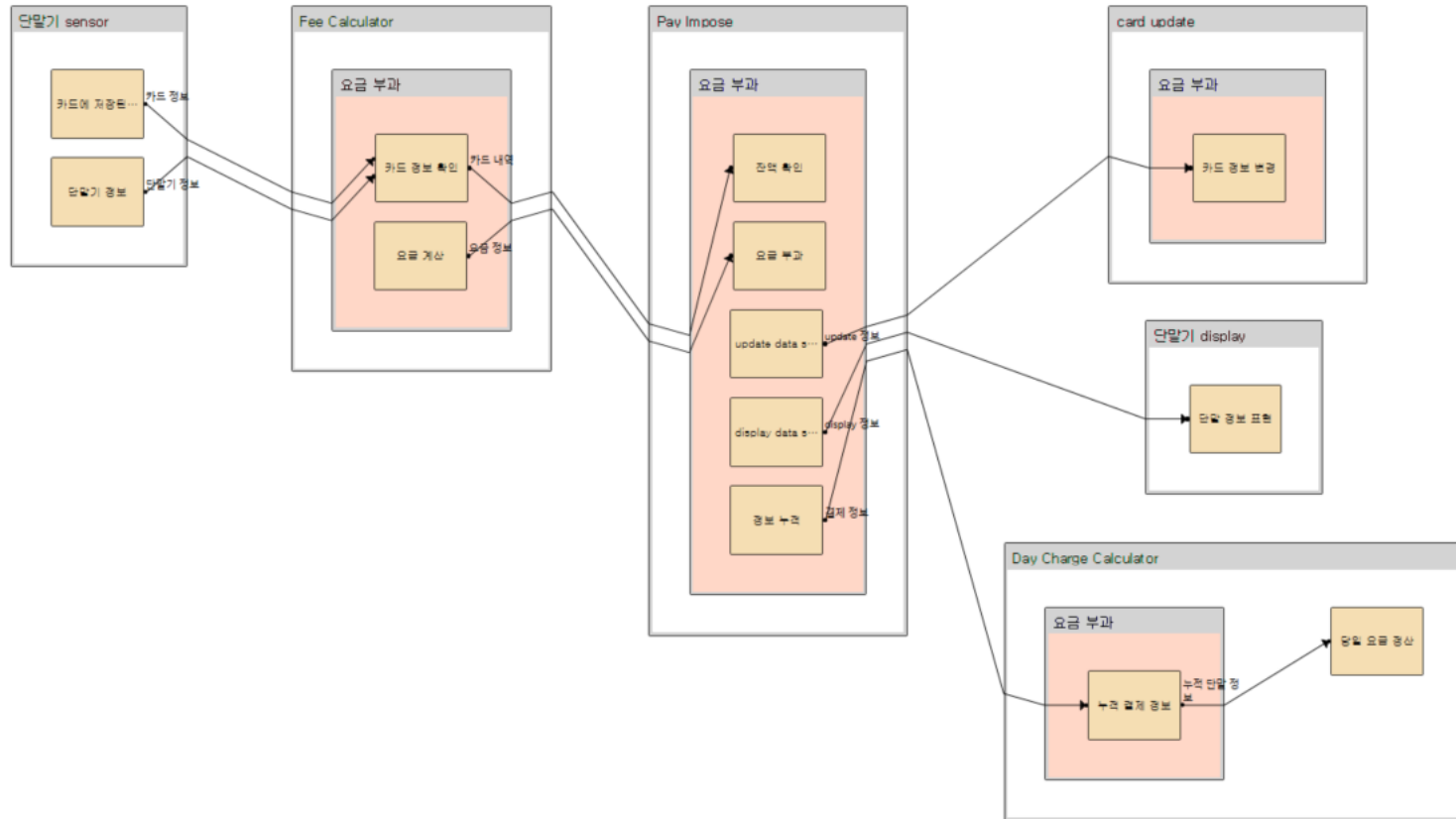


Functional Requirements & SA

Functional requirements	SA
당일 요금 정산	Daily fee calculator(2.1.6)
정산 요금 분배	Calculate result(2.4)
카드에 저장된 정보 인식	Card Reader(1)
카드 인식시 단말에 정보 표현	Display Controller(2.1.4), Display interface(2.2)
단말기 정보	Card Reader(1)
요금 계산	Fee Calculator(2.1.2)
요금 부과	Pay Impose Controller(2.1.3.1)
탑승 여부 알림	Display Data Sender(.2.1.3.4), Display Controller(2.1.4) Display interface(2.2)
잔액 확인	Balance Checker(2.1.3.2)
카드 정보 확인	Fee Calculator(2.1.2)
누적 결제 정보	Accumulator(2.1.3.3)
카드 정보 변경	Card Info Updater(2.1.5), Update interface(2.3)
환승 여부 결정	Fee Calculator(2.1.2)

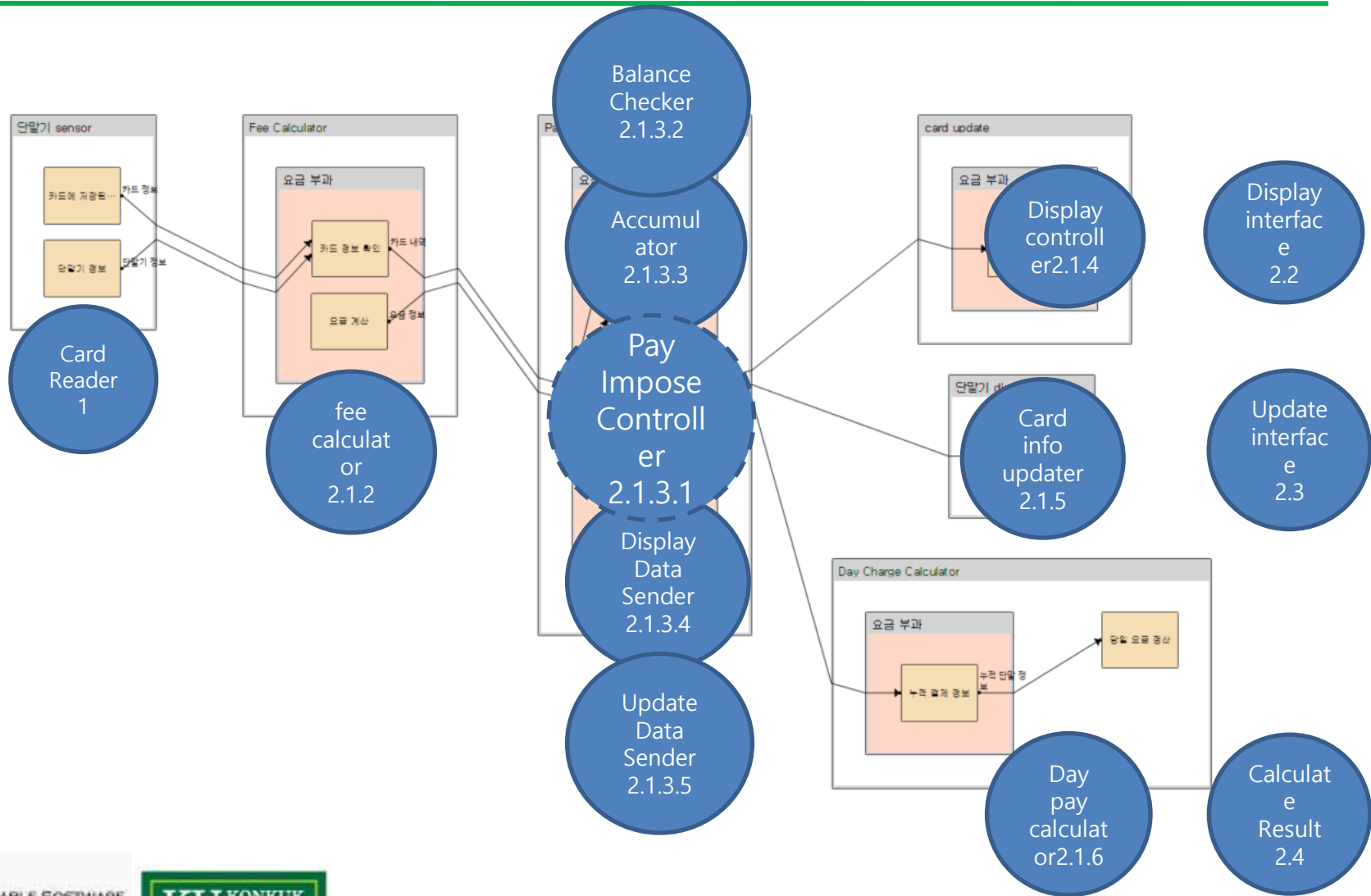
RM+

- Logical Design Model



RM+ + SA

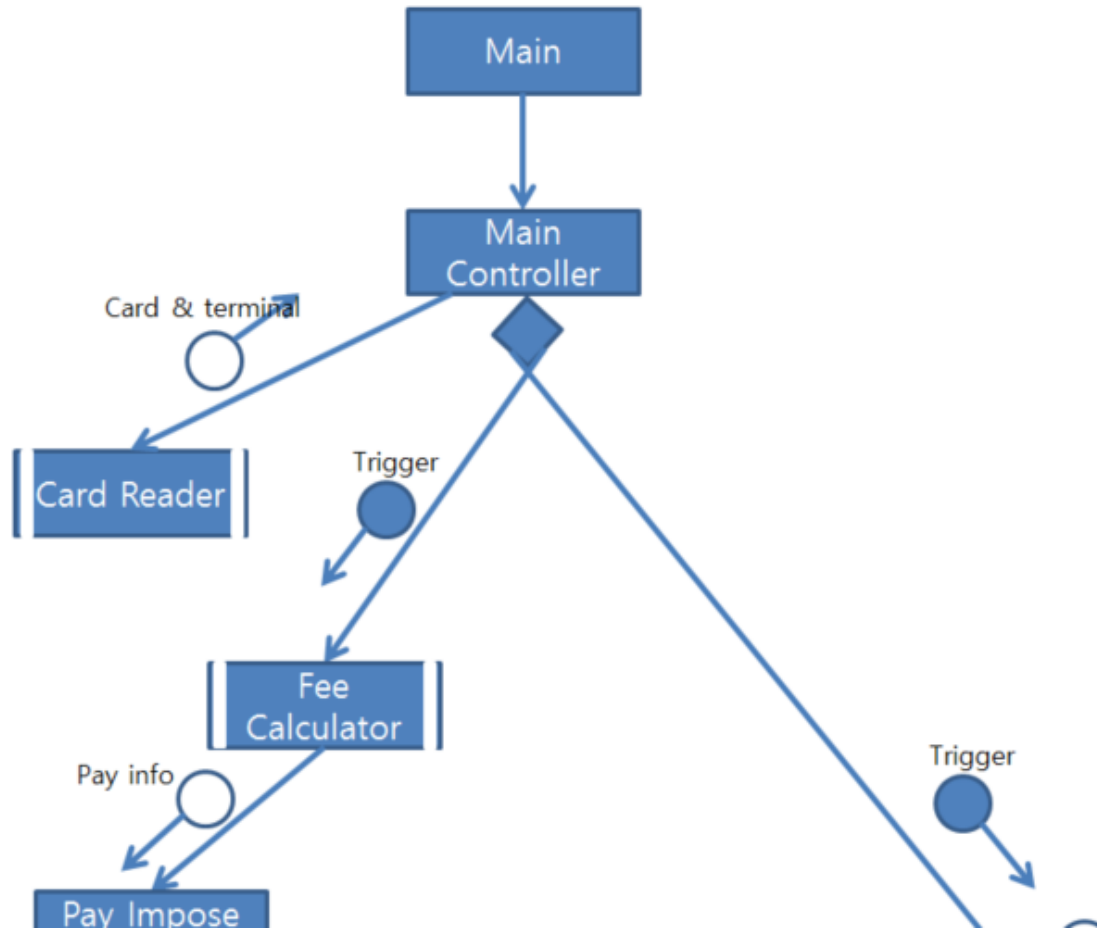
- Logical Design Model



Introduce RM+ + SD

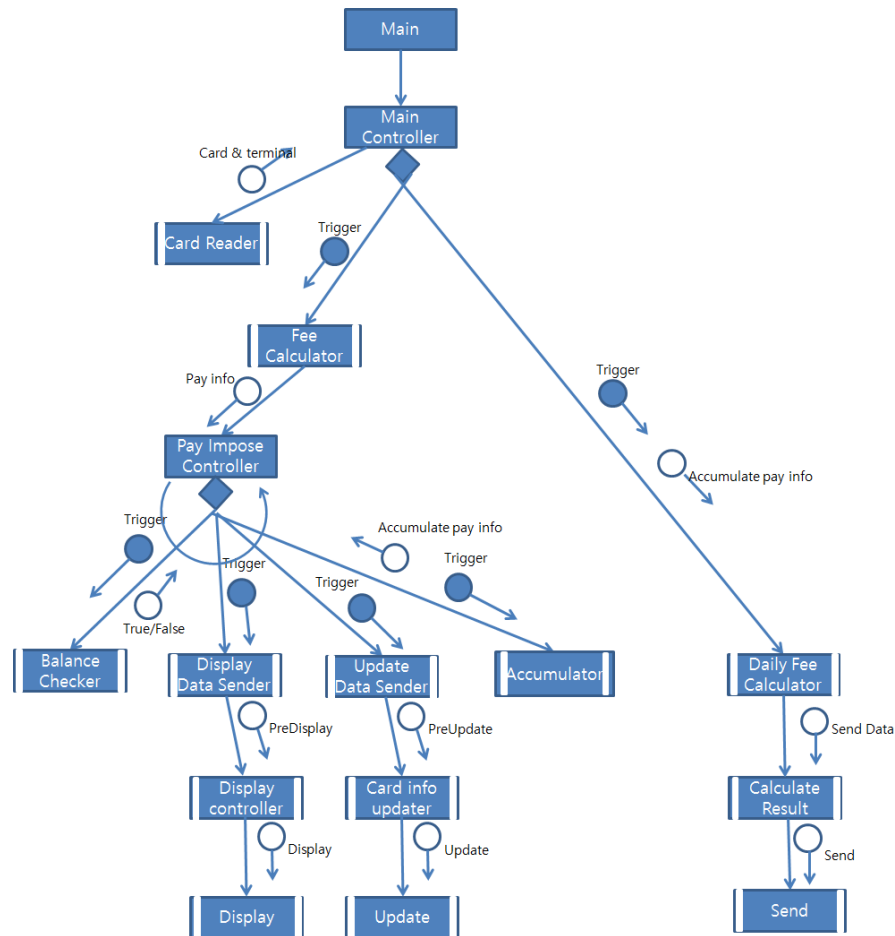
RM+ + SD

SD



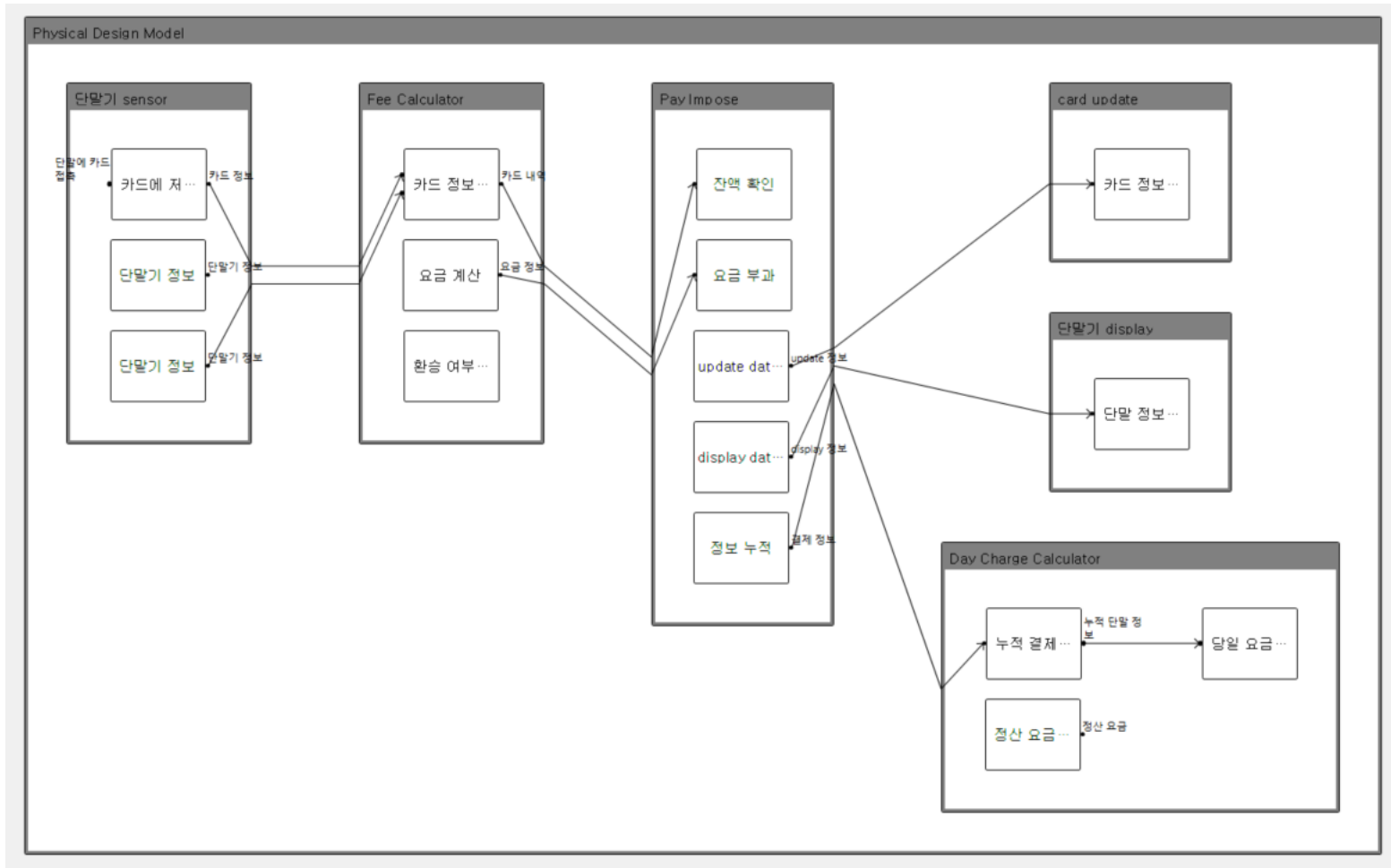
SD

- Overall



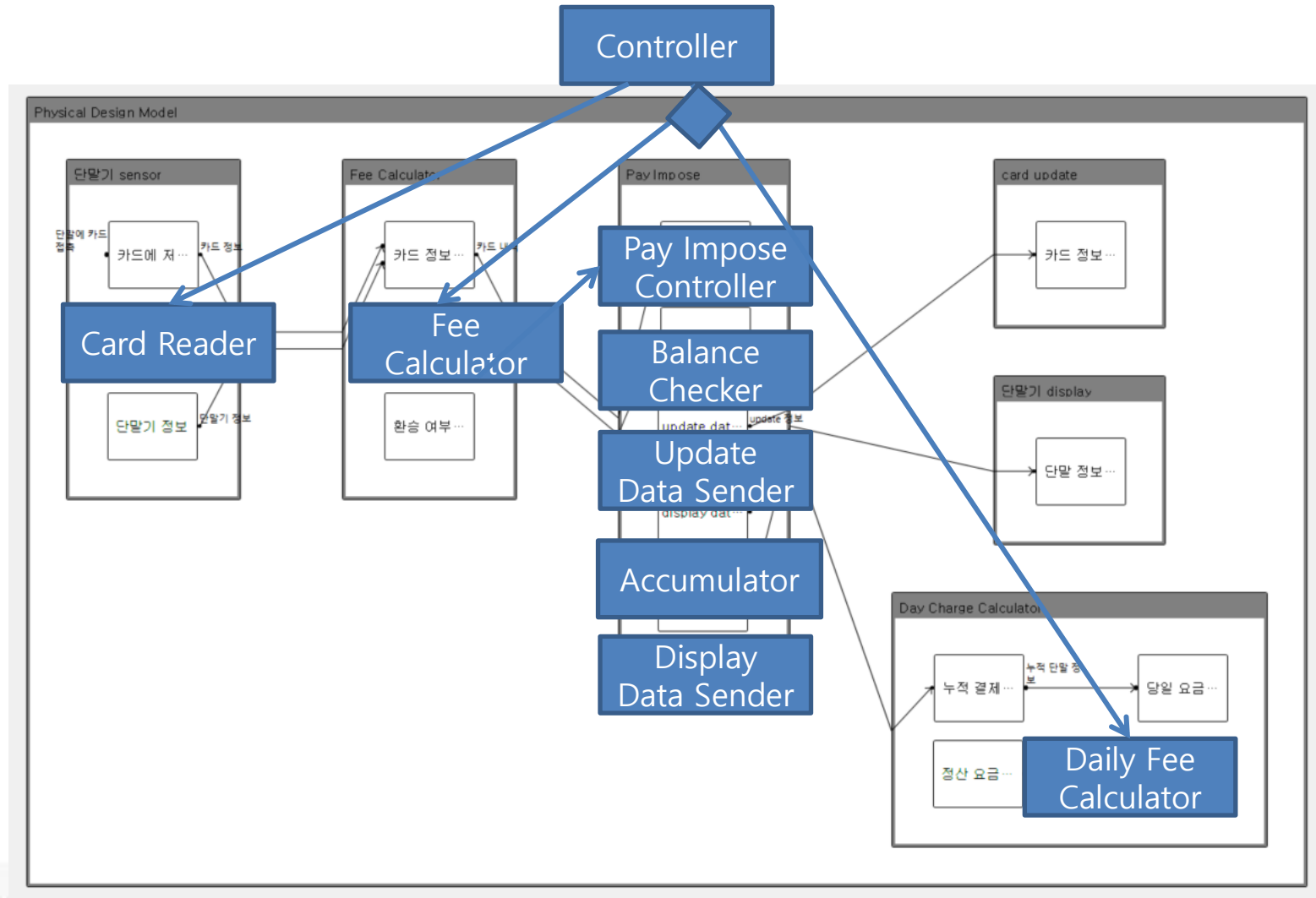
RM+

- Physical Design Model



RM+ + SD

- Physical Design Model



Introduce SD + RM+

TEST CASE

Test case – system test

- Test case를 작성할 requirements를 선택

The screenshot displays a software interface for managing requirements. On the left, a table titled "Test Specifications Requirements:" lists various requirements. On the right, a tree view titled "Specifying Item:" shows a hierarchy of functional requirements, with a "Same" relationship indicated between the two views.

Name	Id
당일 요금 정산	
정산 요금 분배	
카드에 저장된 정보 인식	
단말 정보 표현	
단말기 정보 인식	
요금 계산	
요금 부과	
탑승 여부 알림	
잔액 확인	
카드 정보 확인	
누적 결제 정보	
카드 정보 변경	
환승 여부 결정	

Specifying Item: Set Clear

Inclusion Stat...	Name
	Functional Requirements
	요금 정산
Same	당일 요금 정산
Same	정산 요금 분배
	카드 정보 인식
Same	카드에 저장된 정보 인식
Same	단말 정보 표현
Same	단말기 정보 인식
	요금 부과
Same	요금 계산
Same	요금 부과
Same	탑승 여부 알림
Same	잔액 확인
Same	카드 정보 확인
Same	누적 결제 정보
Same	카드 정보 변경
Same	환승 여부 결정

Test case – system test

- 1 하루 요금 정산 test
- 2 버스 요금 부과 test
- 3 단말 정보 표현 test
- 4 버스 추가 요금 부과 test
- 5 사용자 카드 인식 test
- 6 환승 요금 test
- 7 지하철 요금 부과 test
- 8 지하철 추가 요금 부과 test
- 9 탑승 여부 결정 test
- 10 카드 정보 변경 test
- 11 단말 누적 정보 갱신 test
- 12 환승 여부 결정 test
- 13 정산 요금 분배 test

	하루 요금 정산	하루 요금 정산 test
	버스 요금 부과	버스 요금 부과 test
	단말 정보 표현	단말 정보 표현 test
	버스 추가 요금 부과	버스 추가 요금 부과 test
	사용자 카드 인식	사용자 카드 인식 test
	환승 요금	환승 요금 test
	지하철 요금 부과	지하철 요금 부과 test
	지하철 추가 요금 부과	지하철 추가 요금 부과 test
	탑승 여부 결정	탑승 여부 결정 test
	단말 누적 정보 갱신	단말 누적 정보 갱신 test
	카드 정보 변경	카드 정보 변경 test
	정산 요금 분배	정산 요금 분배 test
	환승 여부 결정	환승 여부 결정 test

Test case – system test

Requirement name	Id	Description
System Test Specification		
하루 요금 정산 test	91	매일 요금을 정산할 수 있는지 test
버스 요금 부과 test	92	버스에 탑승한 사용자에 버스 요금을 부과할 수 있는지 test
단말 정보 표현 test	93	승객 탑승시 단말기에 탑승 정보를 표현 하는지 test
버스 추가 요금 부과 test	94	버스에 탑승한 사용자에 추가 요금을 부과하는지 test
사용자 카드 인식 test	95	단말기에 태그시 사용자의 카드 정보를 인식하는지 test
환승 요금 test	96	버스/지하철 탑승시 사용자에 환승 요금을 부과할 수 있는지 test
지하철 요금 부과 test	97	지하철에 탑승한 사용자에 지하철 요금을 부과할 수 있는지 test
지하철 추가 요금 부과 test	98	지하철에 탑승한 사용자에 추가 요금을 부과하는지 test
탑승 여부 결정 test	99	버스/지하철 탑승시 탑승 가능함지 여부 확인 가능 test
카드 정보 변경 test	100	탑승 가능한 경우 승객의 카드정보를 갱신하는지 test
단말 누적 정보 갱신 test	101	승객이 탑승하는 경우 단말기의 누적 정보를 갱신하는지 test
환승 여부 결정 test	102	버스/지하철 탑승시 환승 여부를 결정하여 올바르게 결정 하는지 test
정산 요금 분배 test	103	매일 정산된 요금을 버스/지하철에 정상적으로 분배하는지 test

Test case – functional test

- Requirements와 test case를 매핑

#	Name	Name	Id	In Test Case
1	요금 정산 test	당일 요금 정산		요금 정산 test
2	요금 분배 test	정산 요금 분배		요금 분배 test
3	카드 정보 인식 test	카드에 저장된 정보 인식		카드 정보 인식 test
4	버스 탑승 정보 표현 test	단말기 정보 표현		버스 탑승 정보 표현 test, 지하철
5	지하철 탑승 정보 표현 test	단말기 정보 인식		버스 단말기 정보 인식 test, 지하철
6	탑승 거부 표현 test	요금 계산		버스 요금 계산 test, 지하철
7	버스 단말기 정보 인식 test	요금 부과		승객에 계산된 요금 부과 test
8	지하철 단말기 정보 인식 test	탑승 여부 알림		버스/지하철 탑승 ok test, 버스
9	버스 요금 계산 test	잔액 확인		카드의 잔액 정보 확인 여부 test
10	지하철 요금 계산 test	카드 정보 확인		카드의 기본 정보 확인 여부 test
11	버스 환승 요금 계산 test	누적 결제 정보		버스 결제 정보 누적 확인 test
12	지하철 환승 요금 계산 test	카드 정보 변경		탑승 거부시 카드 정보 변경 x test
13	버스 추가 요금 계산 test	환승 여부 결정		지하철 -> 버스 탑승시 환승 o
14	지하철 추가 요금 계산 test			
15	승객에 계산된 요금 부과 test			
16	버스/지하철 탑승 ok test			
17	버스/지하철 탑승 거부 test			
18	카드의 잔액 정보 확인 여부 test			
19	카드의 기본 정보 확인 여부 test			
20	버스 결제 정보 누적 확인 test			
21	지하철 결제 정보 누적 확인 test			
22	탑승 거부시 카드 정보 변경 x test			
23	탑승 ok시 카드 정보 변경 test			
24	지하철 -> 버스 탑승시 환승 ok test			
25	버스 -> 지하철 탑승시 환승 ok test			
26	버스 -> 지하철, 지하철 -> 버스시 시간 ...			
27	환승이 아닌 경우 x 여부 test			

Test case – functional test

Requirement name	Id	Description
Functional Test Specification		
요금 계산 test	106	버스/지하철의 환승, 추가요금의 누적 정보를 이용해 정산요금을 올바르게 계산하는지 테스트 한다 누적 요금 계산은 srs문서를 따른다
요금 분배 test	107	정산한 요금을 버스/지하철로 전달하는지 테스트 한다
카드 정보 인식 test	108	단말기에 카드를 태그시 카드의 정보를 제대로 가져올 수 있는지 테스트 한다 카드의 정보는 파일의 형태로 구성되며 그 상세 내용은 SRS문서에 존재 한다 문서의 내용대로 카드의 정보를 읽어오는지 테스트 한다
버스 환승 정보 표현 test	109	단말기에 버스의 환승 정보를 올바르게 표현하는지 테스트 한다
지하철 환승 정보 표현 test	110	단말기에 지하철의 환승 정보를 올바르게 표현하는지 테스트 한다
환승 거부 표현 test	111	잔액 부족으로 인해 환승 불가일 경우 올바르게 표현하는지 테스트 한다
버스 단말기 정보 인식 test	112	단말기에서 단말기 ID를 통해 버스 단말기의 정보를 정확히 인식 하는지 테스트 한다
지하철 단말기 정보 인식 test	113	단말기에서 단말기 ID를 통해 지하철 단말기의 정보를 정확히 인식 하는지 테스트 한다
버스 요금 계산 test	114	인식한 카드 정보와 단말기 정보를 이용해 버스 환승의 요금 계산을 올바르게 하는지 테스트 한다
지하철 요금 계산 test	115	인식한 카드 정보와 단말기 정보를 이용해 지하철 환승의 요금 계산을 올바르게 하는지 테스트 한다
버스 환승 요금 계산 test	116	인식한 카드 정보와 단말기 정보를 이용해 버스 환승의 요금 계산을 올바르게 하는지 테스트 한다
지하철 환승 요금 계산 test	117	인식한 카드 정보와 단말기 정보를 이용해 지하철 환승의 요금 계산을 올바르게 하는지 테스트 한다
버스 추가 요금 계산 test	118	인식한 카드 정보와 단말기 정보를 이용해 버스 추가 요금 계산을 올바르게 하는지 테스트 한다 추가 요금은 미정산 요금을 포함한 요금이다

Test case – functional test

 지하철 추가 요금 계산 test	119	인식한 카드 정보와 단말기 정보를 이용해 지하철 추가 요금 계산을 올바르게 하는지 테스트 한다 추가 요금은 미정산 요금을 포함한 요금이다
 승객에 계산된 요금 부과 test	120	요금 계산 결과를 승객에 부과하는지 테스트 한다 요금 부과는 카드 정보 경신으로 한다
 버스/지하철 탑승 ok test	121	카드의 잔액과 계산된 요금을 이용해 탑승 가능성을 판단 하는지 테스트 한다
 버스/지하철 탑승 거부 test	122	카드의 잔액과 계산된 요금을 이용해 탑승 불가능을 판단 하는지 테스트 한다
 카드의 잔액 정보 확인 여부 test	123	카드의 정보에서 잔액 정보를 가져 와 확인하는지 테스트 한다
 카드의 기본 정보 확인 여부 test	124	단말기 태그시 얻어온 정보를 필요한 정보로 가져 오는지 테스트 한다
 버스 결제 정보 누락 확인 test	126	버스 탑승시 결제 정보가 누락되는지 테스트 한다
 지하철 결제 정보 누락 확인 test	127	지하철 탑승시 결제 정보가 누락되는지 테스트 한다
 탑승 거부시 카드 정보 변경 x test	128	잔액 확인 결과 탑승 불가의 경우 카드 정보가 변경되지 않는지 테스트 한다
 탑승 ok시 카드 정보 변경 test	129	잔액 확인 결과 탑승 가능의 경우 카드 정보가 변경 되는지 테스트 한다
 지하철 -> 버스 탑승시 환승 ok test	130	지하철 하차 후 버스 탑승시 환승 OK 가 되는지 테스트 한다
 버스 -> 지하철 탑승시 환승 ok test	131	버스 하차 후 지하철 탑승시 환승 OK 가 되는지 테스트 한다
 버스 -> 지하철, 지하철 -> 버스시 시간 초과로 환승이 아닌 경우 x 여부 test	132	환승시 시간 초과로 인해 환승 불가가 가능한지 테스트 한다
 환승이 아닌 경우 x 여부 test	133	환승이 아닌 경우에 환승 x 출력 여부를 테스트 한다

Test case – functional test

- Editor에서 연결된 requirement와 description 확인 가능

Test case editor

Input Valu	Expected R	Result	ID	TC ID
			...	106

Description

버스/지하철의 환승, 추가요금의 누적 정보를 이용해 정산요금을 올바르게 계산하는지 테스트 한다
누적 요금 계산은 srs문서를 따른다

Requirements

당일 요금 정산

매 자정이 되면 각 단말의 누적 정보를 받아서 하루동안 운행 요금을 분배 할 수 있도록 정산 되어야 한다
정산 금액은 환승시 발생하는 요금에 따라 계산하며 공식은 다음과 같다

공식: 버스→지하철 환승
버스 = 총금액 * 버스 요금 / (버스 요금 + 총금액)
지하철 = 총금액 * 총금액 / (버스 요금 + 총금액)
공식: 지하철→버스 환승
버스 = 총금액 * 총금액 / (지하철 요금 + 총금액)
지하철 = 총금액 * 지하철 요금 / (지하철 요금 + 총금액)

Q & A

END

SD

