

Assignment 1
2509ICT/7509ICT
Software Engineering
Lecturer: Larry Wen
Tutor: KB

Group Number: 1
Workshop: Thursday 10am

Group Members

Student number	Name
S5013937	Jaike Cambridge
S2946192	Aidan McMillan
S2944951	Wasif Chowdhury
s2807480	Alex Graham
S2930944	Haoxuan Feng

Contents

1.0 Project Planning and Documentation	1
1.3 Effort and contribution table	3
2.0 Requirements Analysis	4
Original Problem Definition	4
2.1 Requirement Behaviour Trees (RBT)	5
2.2 Use Case Diagram	14
3.0 Specification	15
3.1 Integrated Behavior Tree (IBT)	15
3.2.1 DBT - Defects Found	20
3.3 Integrated Composition Tree (ICT)	22
4.0 Design	26
4.1 Component Behaviour Trees (CBT)	26
4.1.2 OrdSystem	26
4.2 Component Interaction Network (CIN)	29
4.3 Class Diagram	29
4.4 Sequence Diagrams	30
4.4.1 Sequence Diagram (Input a new Order)	30
4.4.2 Sequence Diagram	31
4.5 Requirement Defects	32
4.5.1 Assumptions	32
4.5.2 Problems and Revision Notes	32
5.0 Error/Defect Analysis	33

1.0 Project Planning and Documentation

Student	Effort Level* (Rating from 0 – 5, the information is filled by the group)	Contribution Level* (Rating from 0 – 5, the information is filled by the group)	Justification If a student received level rating of 3 or less, your group need to give explanation for the low level rating
Alex Graham	5	5	
Aidan McMillan	5	5	
Haoxuan Feng	5	5	
Wasif Chowdhury	5	5	
Jaike Cambridge	5	5	

1.1 Time Schedule

Task	Person	Estimated Development	Estimated Verification	Estimated Correction	Estimated Hours	Actual Hours
2.0 Analysis of Problem Statement	All	1hour	<1 hour	1 hour	1 hours	1 hour
2.1 Requirement Behaviour Trees (RBT)	Haoxuan Feng(2.1-2.3)	2 hours	<1 hour	1 hour	3 hours	2.5 hours
	Wasif Chowdhury(2.4-2.5)	2 hours	<1 hour	1 hour	3 hours	2 hours
	Jaike Cambridge(2.6-2.7)	2 hours	<1 hour	1 hour	3 hours	2 hours
	Alex Graham(2.9)	1 hour	<1 hour	1 hour	2 hours	2 hours
	Aidan McMillan(2.8)	1 hour	<1 hour	1 hour	2 hours	1 hour
2.2 Use Case Diagram	Wasif Chowdhury	1 hour	<1 hour	<1 hour	1 hour	1.5 hours
3.1 Integrated Behaviour Tree (IBT)	Haoxuan Feng	4 hours	1 hour	1 hour	5 hours	5 hours
3.2 Design Behaviour Tree (DBT)	Aidan McMillan	3 hours	<1 hour	1 hour	4 hours	4 hours

3.3 Integrated Composition Tree (ICT)	Jaike Cambridge	3 hours	<1 hour	1 hour	3 hours	3 hours
3.4 Design Composition Tree (DCT)	Alex Graham	3 hours	<1 hour	1 hours	4.5 hours	5 hours
4.1 Component Behaviour Tree (CBT)	Alex Graham	2 hours	<1 hour	1 hour	3 hours	4 hours
4.2 Component Integrated Network (CIN)	Jaike Cambridge	0.5 hours	<1 hour	<1 hour	0.5 hours	<0.5 hours
4.3 Class Diagrams	Wasif Chowdhury	1.5 hours	< 1 hour	<1 hour	2 hours	1.5 hours
4.4 Sequence Diagrams	Haoxuan Feng	4 hours	<1 hour	<1 hour	5 hours	5 hours
4.5 Requirement Defects	Haoxuan Feng	2 hours	1 hour	1 hour	4 hours	2 hours
	Aidan McMillan	2 hours	1 hour	1 hour	4 hours	3 hours
5.0 Error/Defect Analysis	All	3 hours	2 hours	2 hours	7 hours	5 hours
Project Management	All	3 hours			3 hours	3 hours
Meeting	All	5 hours	-	-	5 hours	6 hours
Documentation	All	3 hours	1 hour	1 hour	5 hours	5 hours

1.2 Total Working Hours

Student Name (#ID)	Plan (hours)	Actual (hours)
Haoxuan Feng S2930944	10	10
Wasif Chowdhury S2944951	10	9
Aidan McMillan s2946192	10	9
Alex Graham s2807480	10	9
Jaike Cambridge	10	9
Total working hours	50	46
Average working hours per person	10	9

1.3 Effort and contribution table

Task		Plan				Actual		
#	Task Name	Student	Planed Time	Cumulative Time	Finished Date	Time	Cumulative Time	Finished Date
1	Draw RBTs	Haoxuan Feng	2 hours	3 hours	28/08/2016	2 hours	2 hours	27/08/2016
		Wasif Chowdhury	2 hours	3 hours	28/08/2016	2 hours	2 hours	28/08/2016
		Alex Graham	1 hour	2 hours	28/08/2016	2 hours	2 hours	29/08/16
		Aidan McMillan	1 hour	2 hours	28/08/2016	1 hour	1 hours	29/08/16
		Jaike Cambridge	2 hours	3 hours	28/08/2016	2 hours	2 hours	29/08/16
2	Use Case	Wasif Chowdhury	1 hour	3 hours	29/08/2016	1.5 hours	3.5 hours	1/09/16
3	IBT	Haoxuan Feng	5 hours	8.5 hours	02/09/2016	5 hours	8.5 hours	04/09/2016
4	DBT	Aidan McMillan	4 hours	4 hours	04/09/2016	4 hours	12.5 hours	07/09/2016
5	ICT	Jaike Cambridge	3 hours	15.5 hours	05/09/2016	3 hours	15.5 hours	09/09/2016
6	DCT	Alex Graham	4.5 hours	20 hours	06/09/2016	0 hours	15.5 hours	09/09/2016
7	CBT	Alex Graham	3 hours	22.5 hours	07/09/2016	4 hours	19.5 hours	11/09/2016
8	CIN	Jaike Cambridge	0.5 hours	23 hours	08/09/2016	0.5 hours	20 hours	12/092016
9	Class Diagram	Wasif Chowdhury	2 hours	25 hours	09/09/2016	1.5 hours	21.5 hours	14/09/2016
10	Sequence Diagram	Haoxuan Feng	5 hours	30 hours	10/09/2016	3 hours	24.5 hours	15/09/2016
11	Requirement defects	All	8 hours	38 hours	Overtime	7 hours	31.5 hours	Overtime
12	Defect Analysis	All	7 hours	45 hours	Overtime	5 hours	36.5 hours	Overtime
13	Proofreading and Editing	Haoxuan Feng	~3 hours	48 hours	11/09/2016	1 hours	37.5 hours	17/09/2016
		Aidan McMillan				2 hours	39.5 hours	17/09/2016

2.0 Requirements Analysis

Original Problem Definition

Owners of a restaurant want a customized computer system to support their take-away and home delivery service.

1. When a customer rings either for a take-away or home-delivery the person taking the call asks for their phone number.
2. This is keyed in and used to retrieve their address and credit-card details.
3. If this information is not already present it is requested and stored.
4. The system also enables users to type in the number corresponding to a menu item as it is requested.
5. Each item should be confirmed before it is added to the order.
6. The system should keep a running total of the cost of the order.
7. The system should also provide a total for the day's takings.
8. Users should be able to change details of the menu items and their cost.
9. When adding in orders it should be possible to specify how many of each item are required.

2.1 Requirement Behaviour Trees (RBT)

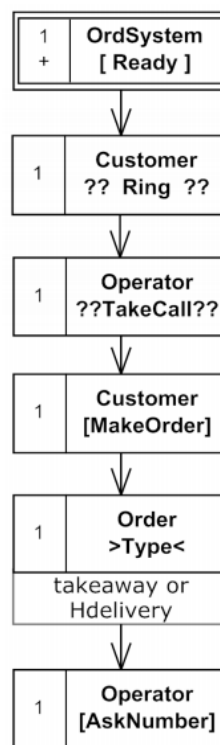
2.1.1 Requirement 1

Original Requirement 1

When a customer rings either for a take-away or home-delivery the person taking the call asks for their phone number.

Refined Requirement 1

The OrdSystem is at the ready state, when a customer rings, an operator takes the call. The customer makes an order with the type of either take-away or home-delivery, the operator then asks for the customer's phone number.



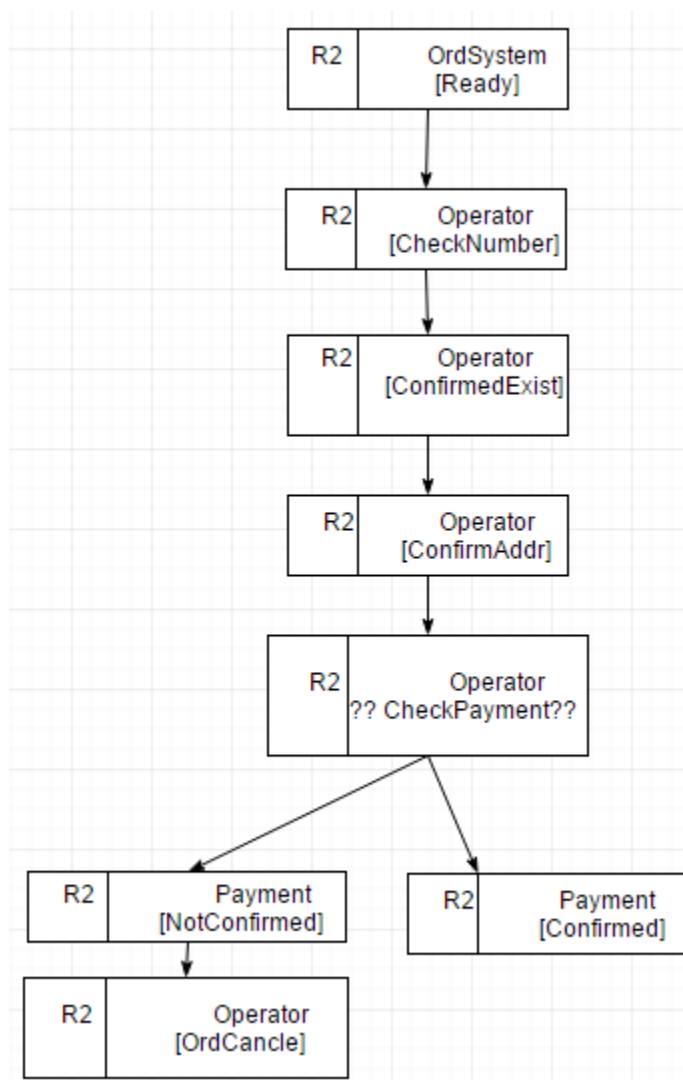
2.1.2 Requirement 2

Original Requirement 2

This is keyed in and used to retrieve their address and credit-card details.

Refined Requirement 2

The OrdSystem is at the ready state, operator check the customer's phone number and confirms customer number already exist. Operator use customer's phone number to confirm customer's payment details (either paid or not). Then operator use customer's phone number to double check customer's address and make sure the type of either take-away or home-delivery, Once confirmed the address and payment type, the order be confirmed/Cancel if there is no more new item.



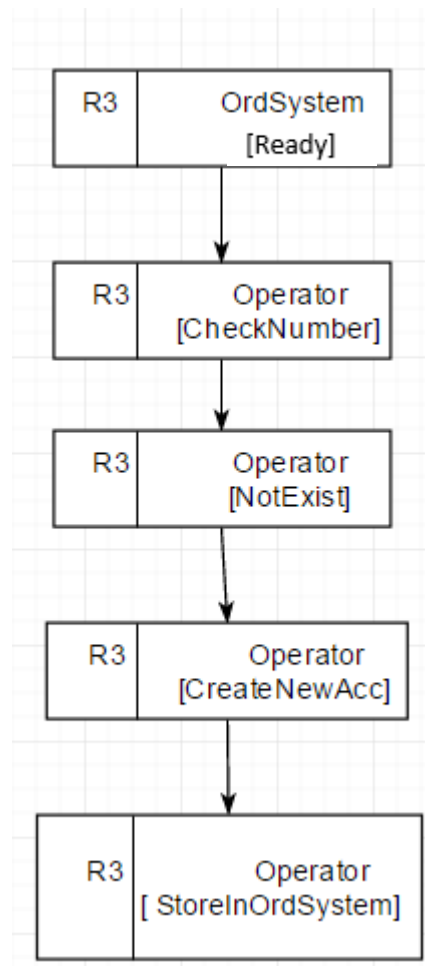
2.1.3 Requirement 3

Original Requirement 3

If this information is not already present it is requested and stored.

Refined Requirement 3

The ordSystem is in the ready state. The operator checks customer's phone number, if it does not exist in OrdSystem then operator creates a new account by using customer phone number and storing it in OrdSystem.



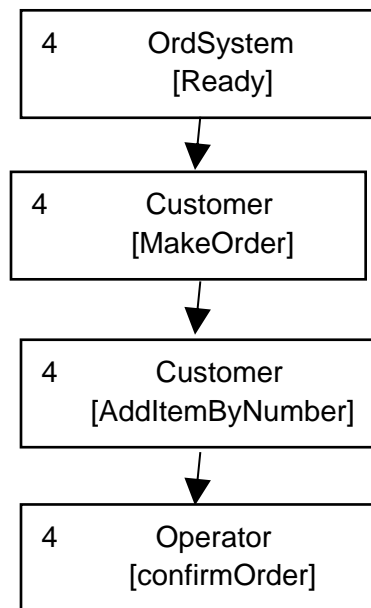
2.1.4 Requirement 4

Original Requirement 4

The system also enables users to type in the number corresponding to a menu item as it is requested.

Refined Requirement 4

The OrdSystem is in ready state. As the customer makes the order they are able to add menu items by their corresponding item number.



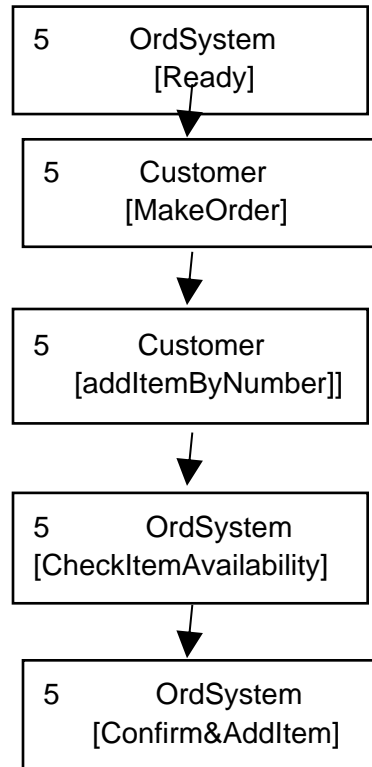
2.1.5 Requirement 5

Original Requirement 5

Each item should be confirmed before it is added to the order.

Refined Requirement 5

The ordering system will check the availability and confirm each menu item that is added by the customer to the order.



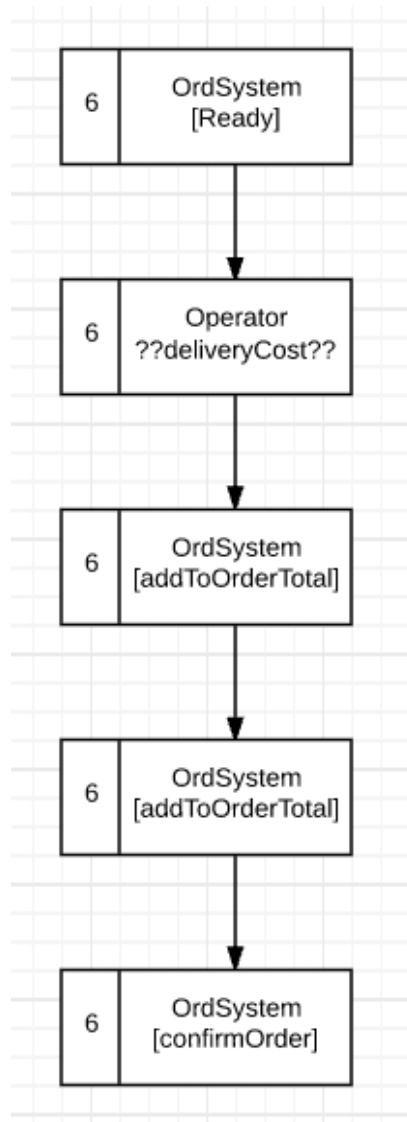
2.1.6 Requirement 6

Original Requirement 6

The system should keep a running total of the cost of the order.

Refined Requirement 6

The OrdSystem is in the ready state. It calculates the cost of an item and adds it to the order total.



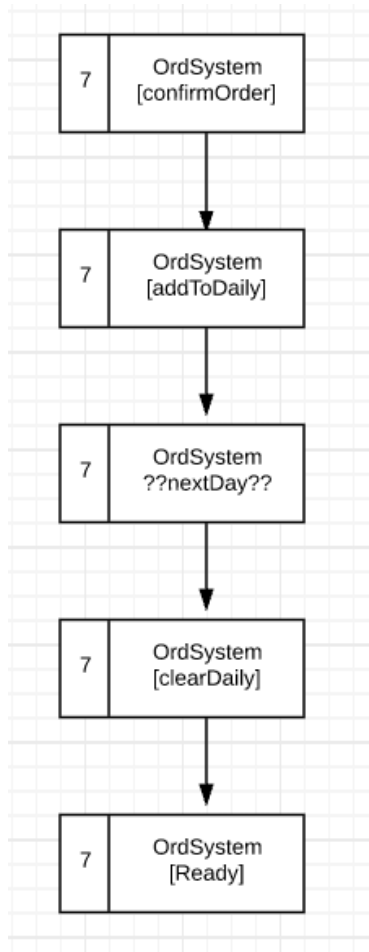
2.1.7 Requirement 7

Original Requirement 7

The system should also provide a total for the day's takings.

Refined Requirement 7

The OrdSystem adds the cost of an order to the day's takings when an order is confirmed. When it is the next day the OrdSystem resets the Daily Takings.



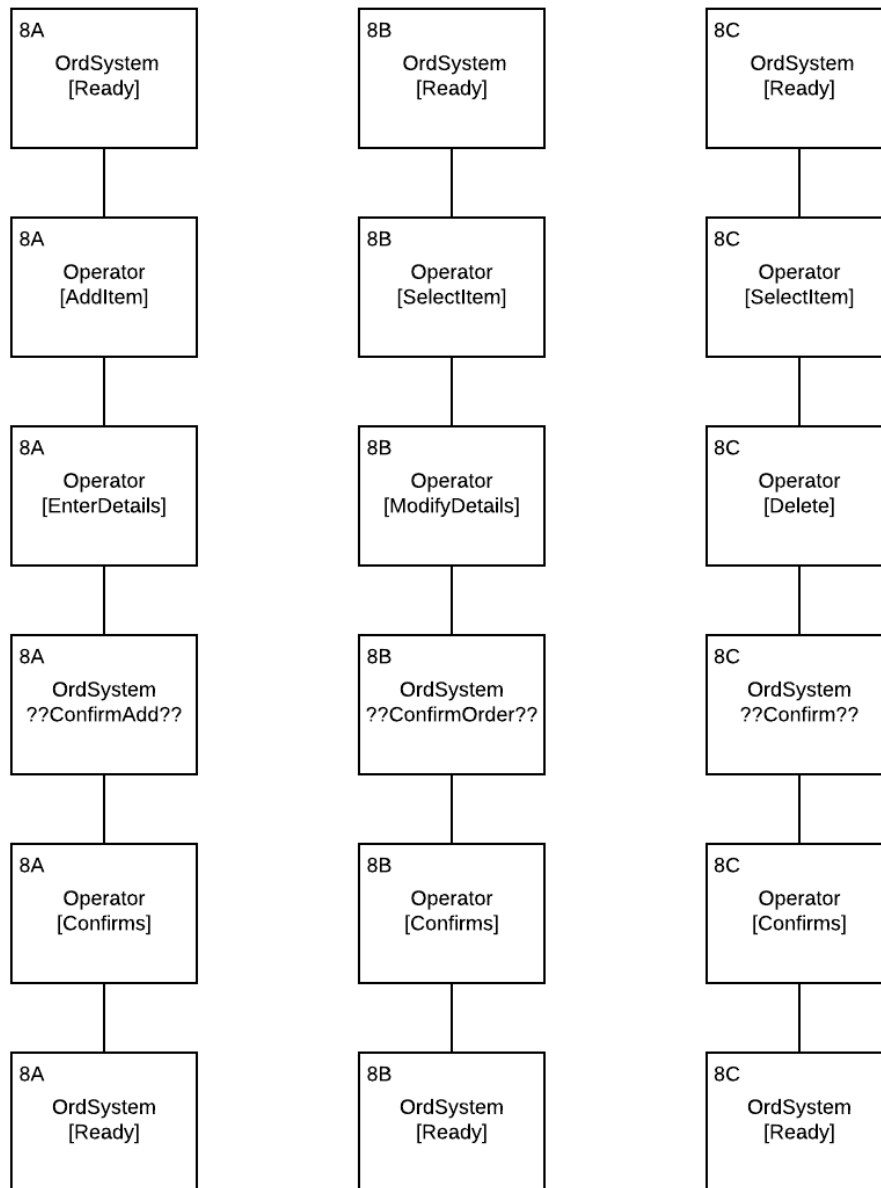
2.1.8 Requirement 8

Original Requirement 8

Users should be able to change details of the menu items and their cost.

Refined Requirement 8:

Order System is in ready state. Operator can add/remove/modify menu items. Order System prompts for confirmation and returns to ready state.



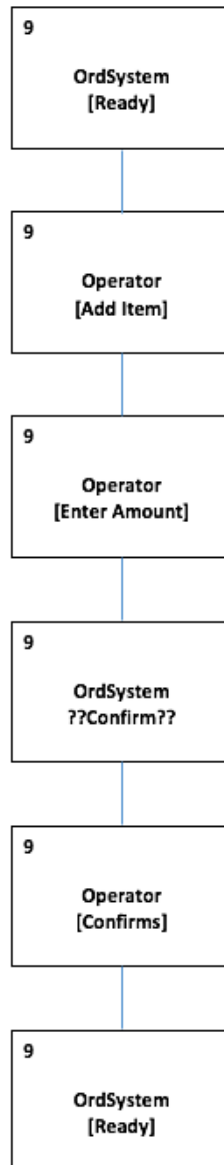
2.1.9 Requirement 9

Original Requirement 9

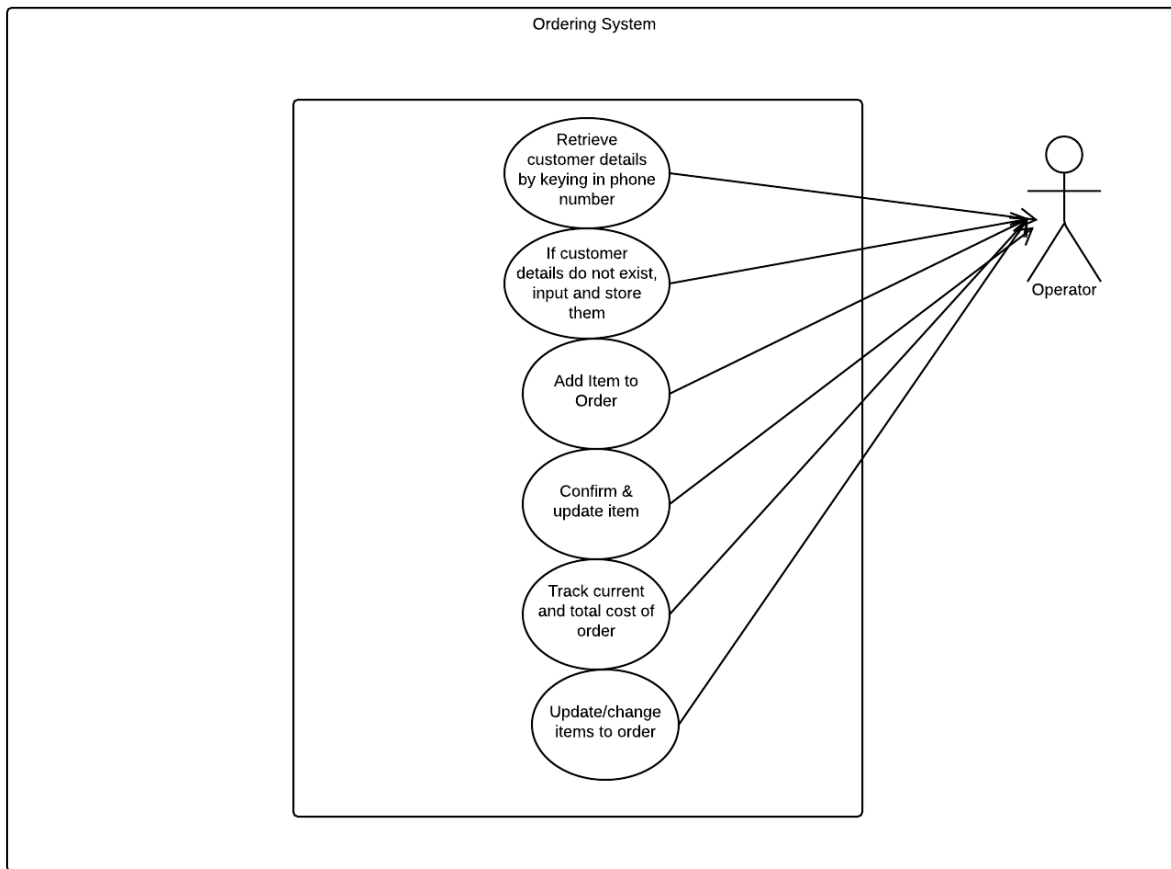
When adding in orders it should be possible to specify how many of each item are required.

Refined Requirement 9

The OrdSystem is in the ready state. An operator can add an item to the order and specify a quantity. They then confirm the addition and the OrdSystem returns to a ready state.



2.2 Use Case Diagram

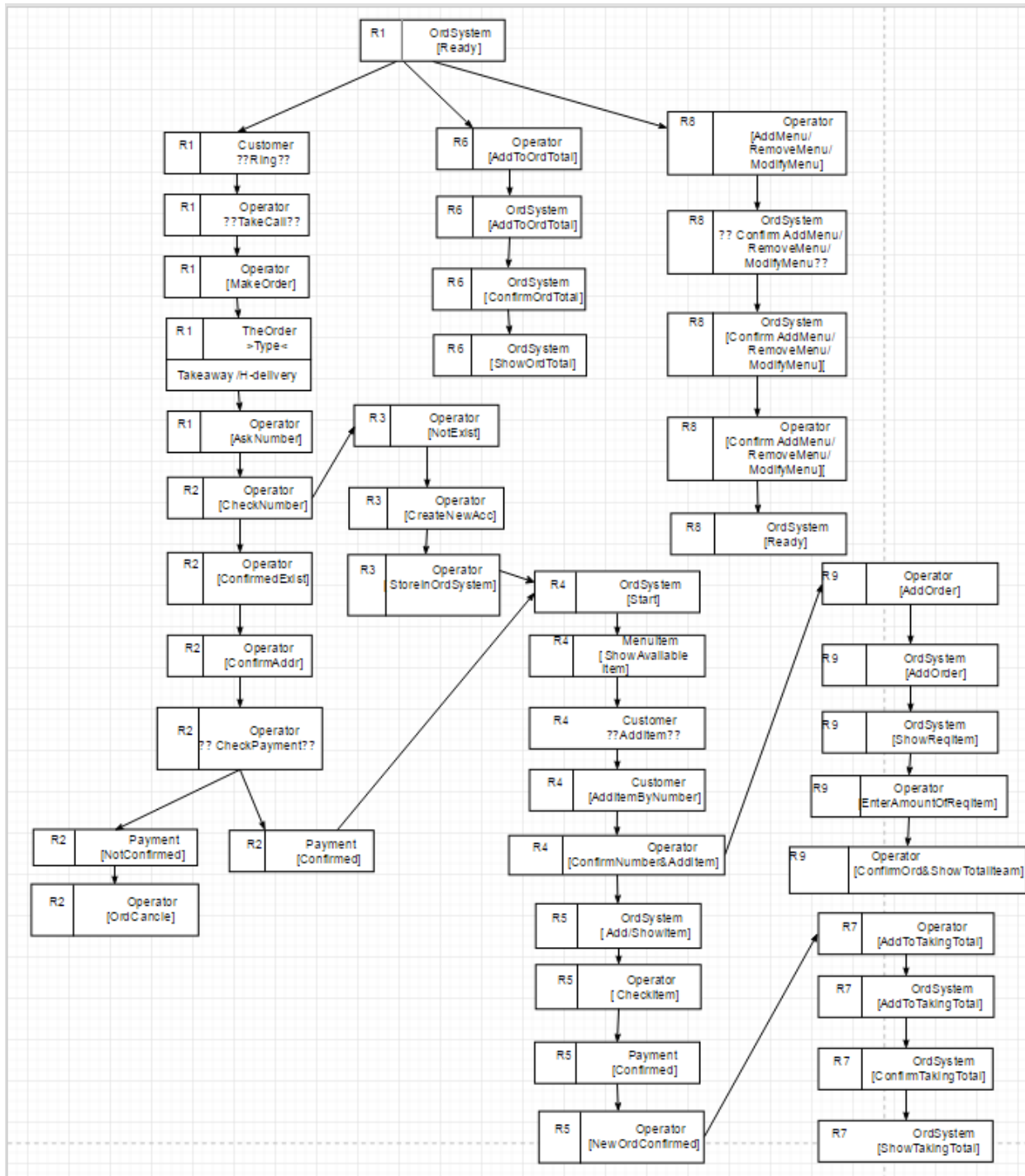


Defects

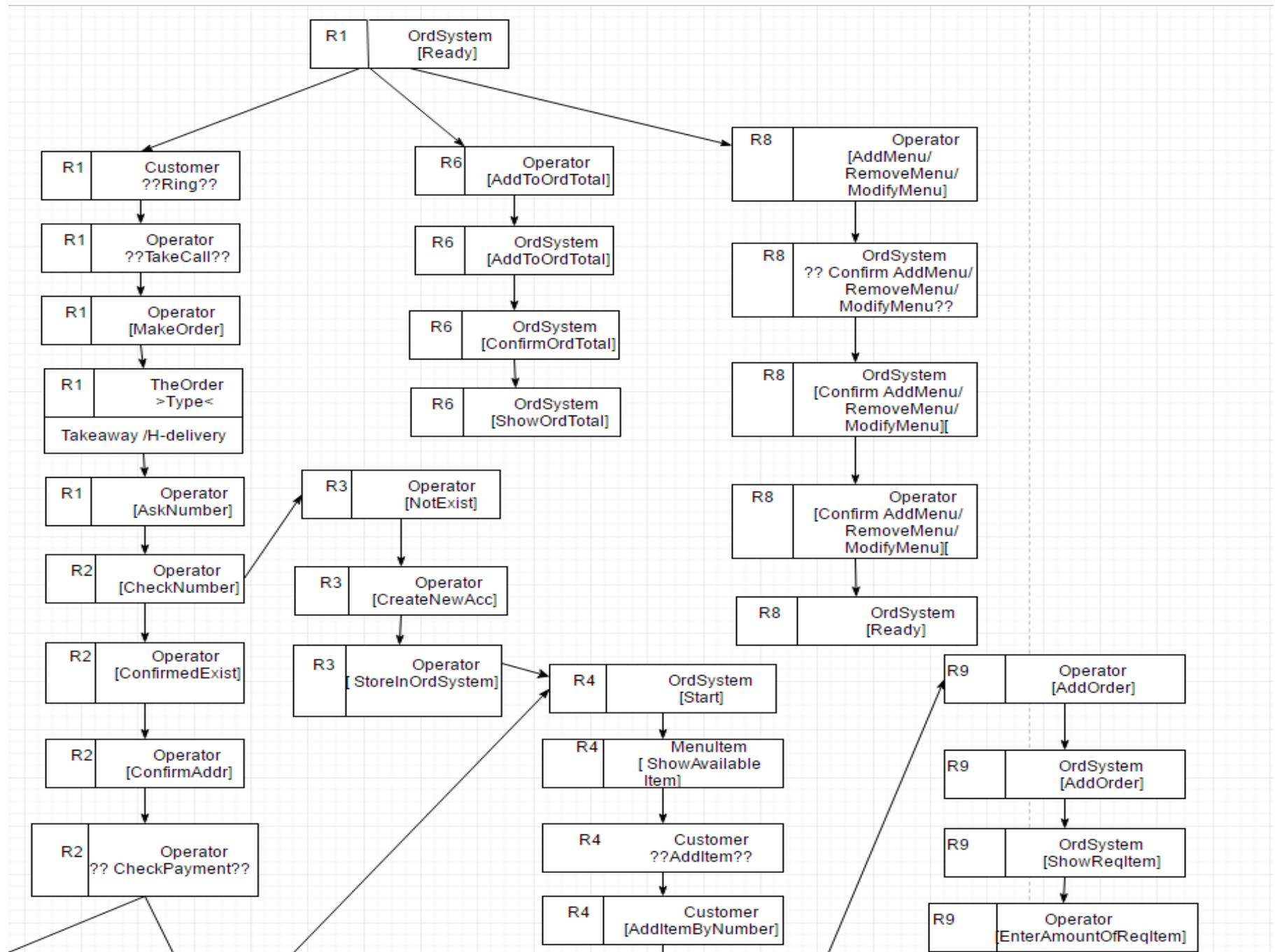
- Originally there was a customer interacting with the ordering system.
- Operator needs to be able to delete entire order.

3.0 Specification

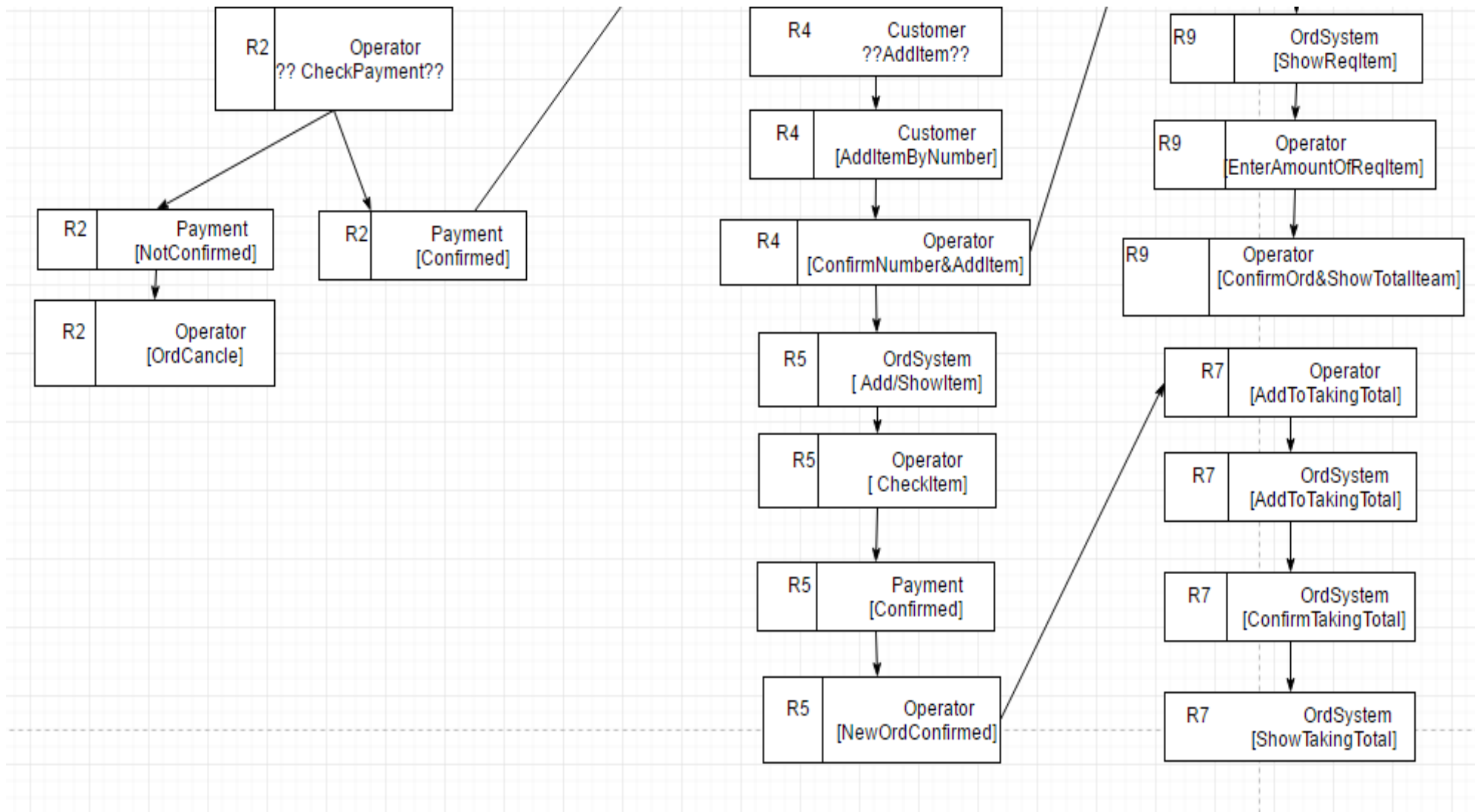
3.1 Integrated Behavior Tree (IBT)



(The entire IBT)



(IBT part 1)



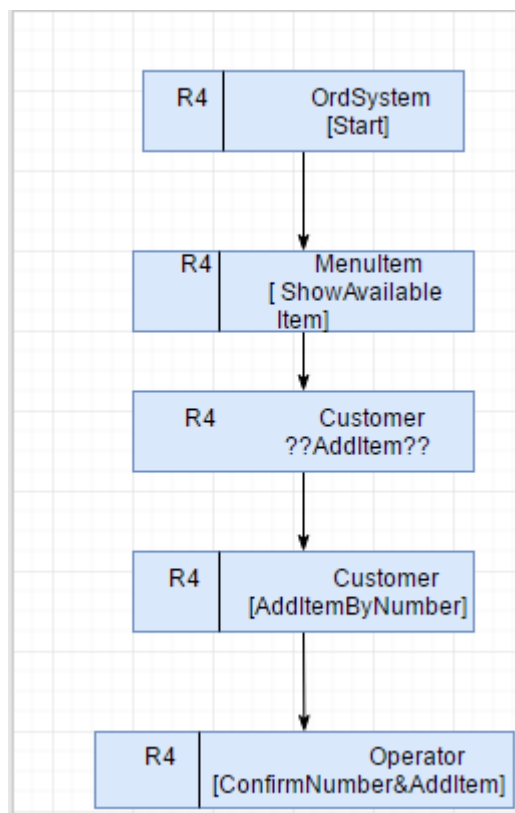
(IBT part 2)

3.1.1 Defects

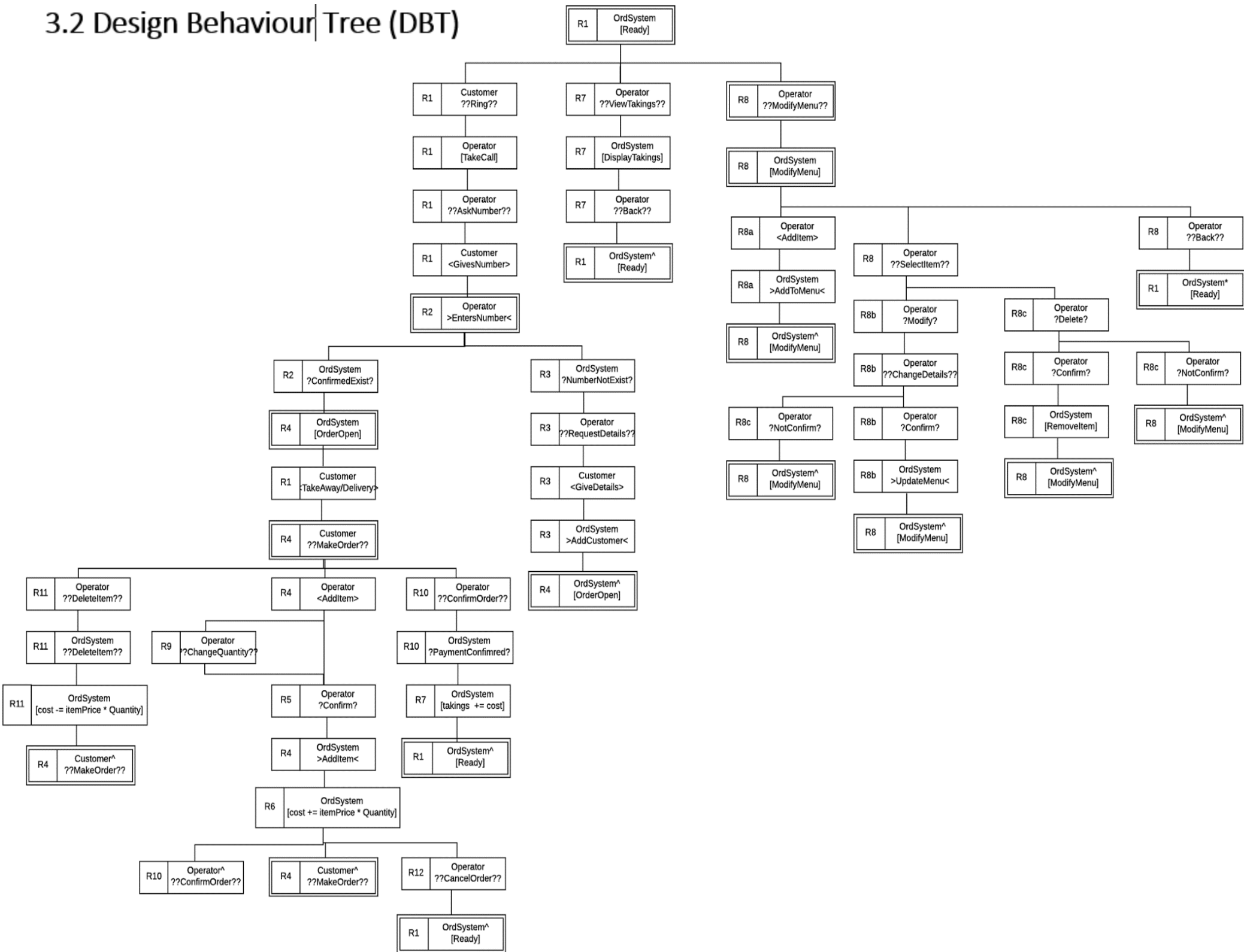
Defect NO.	Description	Req. No	Phase
1	Ambiguity /Change and refined Req 2,3,6 ,7 and 9 in IBT.	2,3,6,7,9 IBT	Translation
2	Redundancy-Should be integrated	5&4,2 IBT	Integration
3	Ambiguity /Change and refined Req 4 and redraw BT	4 BT	Ambiguity

3.1.2 Modified Requirement 4

The ordSystem in the start state as the customer makes the order they are able to use MenuItem to add Item by their corresponding item number. The MenuItem show all the available Item to customer.



3.2 Design Behaviour Tree (DBT)

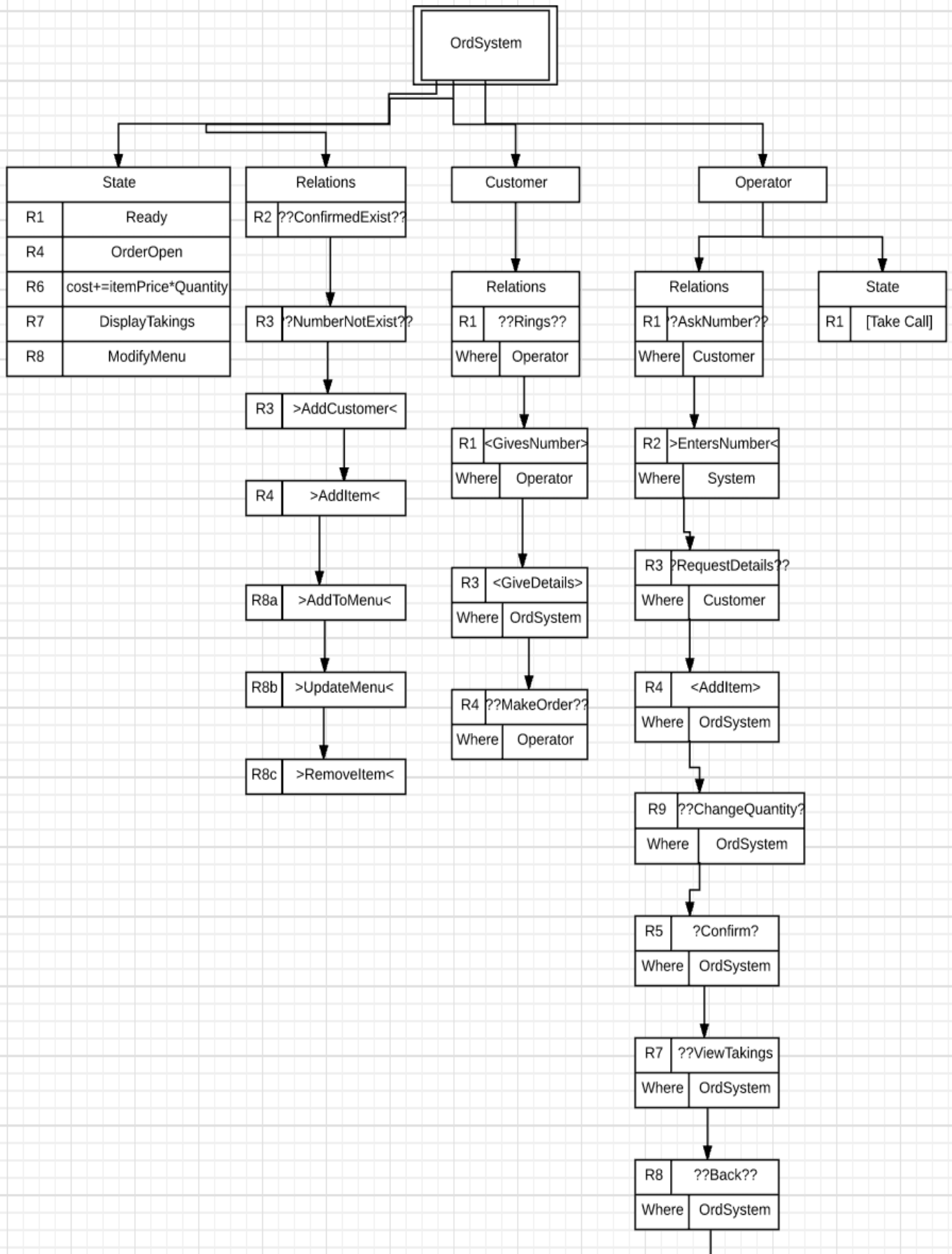


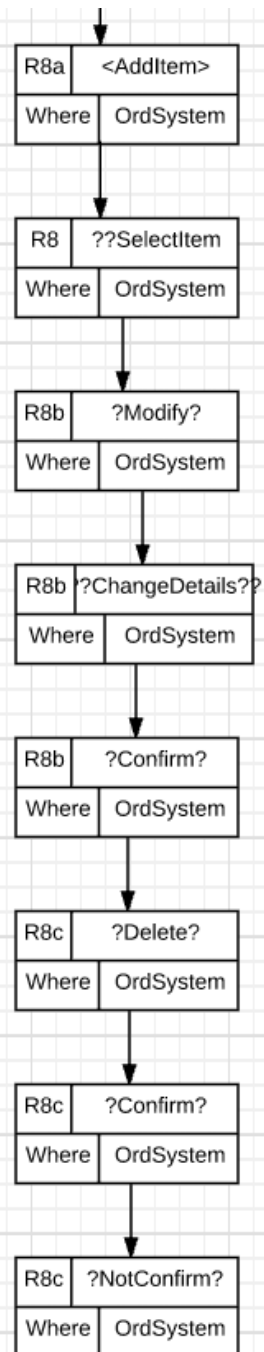
3.2.1 DBT - Defects Found

Defect No.	Description	Requirement No	Phase
1	Change in notation from state[] to input>< for customer and operator number input.	R1, R2	Integration (DBT)
2	Change in notation from state[] to condition?? For test whether number exists or not.	R2, R3	Integration (DBT)
3	Reduced ambiguity by specifying exact process followed if order doesn't exist. Changed notation to better reflect process ([]?<>)	R3	Integration (DBT)
4	Changed layout so rather than arrows pointing back steps, recursion using ^ is used.		General Change (DBT)
5	Changed position of entry of order type(takeaway/delivery) in order to better reflect the system.	R1	Integration (DBT)
6	Payment shouldn't be checked before an order is complete (can't know price)	R2	Integration, Assumption (DBT)
7	Switched payment check to end of order – defined new requirement outlining complete order checking.	R2, R10(New)	Integration (DBT)
8	Made assumption about ability to delete items from a current order.	R4, R11(New)	Integration (DBT)
9	Made assumption that an order may be cancelled.	R2, R4, R12(New)	Integration (DBT)
11	Changed AddToOrd Total to happen dynamically when adding an item to order.	R6, R4	Integration (DBT)
12	Find menu item to be redundant in regards to item.	R4	Integration (DBT)
13	Adding item changed to output, with option to change quantity if necessary.	R4, R9	Integration (DBT)

14	Operator confirming changed to if?? Which made clearer and simpler.	R5	Integration (DBT)
15	OrdSytem adding to order now takes input from operator addItem.	R4	Integration (DBT)
16	Made order dynamically update orderCost based on items and quantity. It does this both when adding and deleting items.	R6, R11	Integration (DBT)
17	Implemented Recursion in order for ability to add/delete multiple products to/from order. (allows mistake correction)	R10, R7, R4	Integration (DBT)
18	Added ability to cancel an order in progress.	R12	Integration (DBT)
19	Split R8 into its separate flows.	R8	Integration (DBT)
20	Implemented Ability to navigate through the modify sub-system as well as back to main ordering system.	R8	Integration (DBT)
21	Made changes regarding input/output/condition/event notation to better reflect process.	R8	Integration (DBT)

3.3 Integrated Composition Tree (ICT)

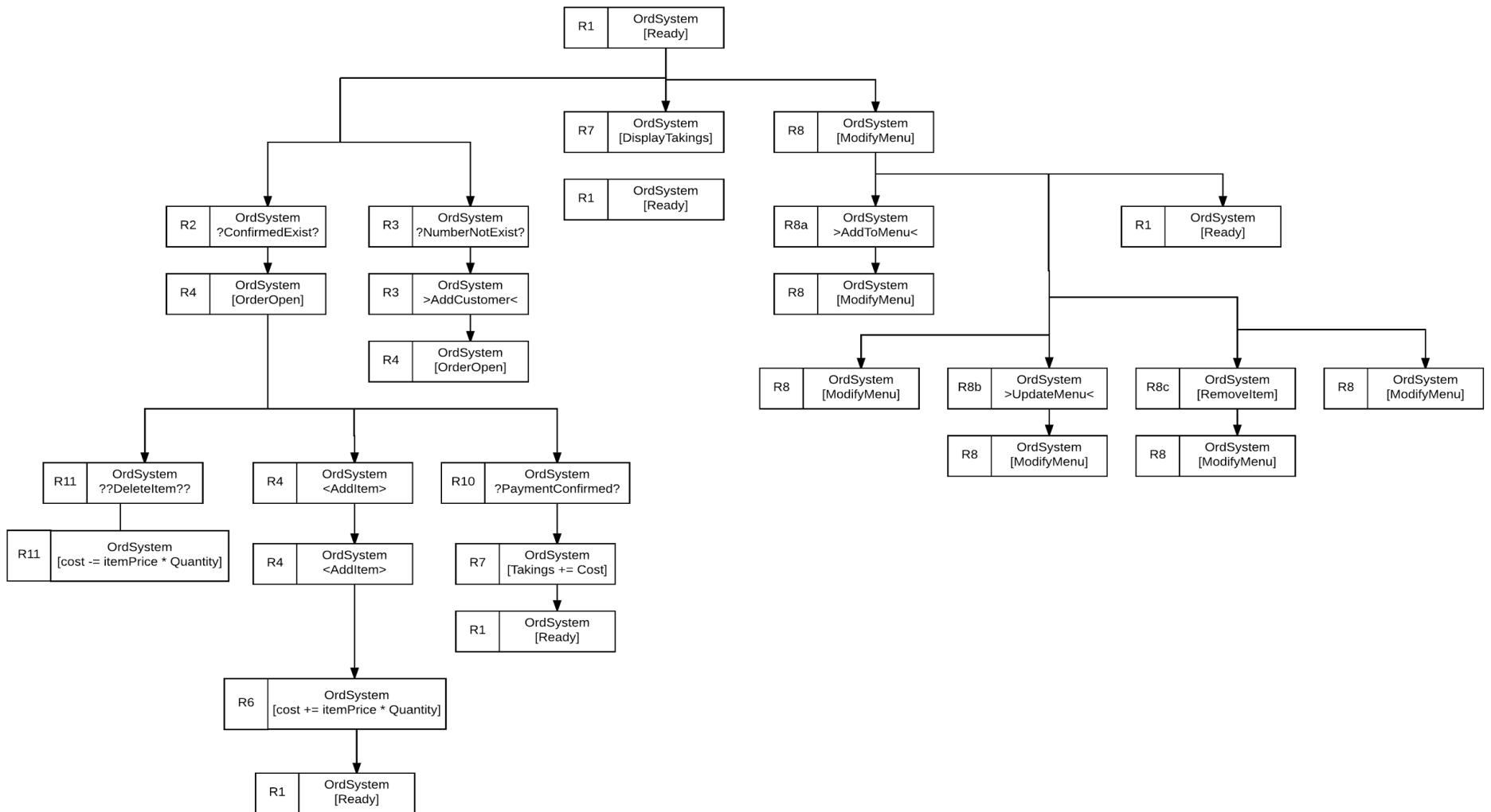




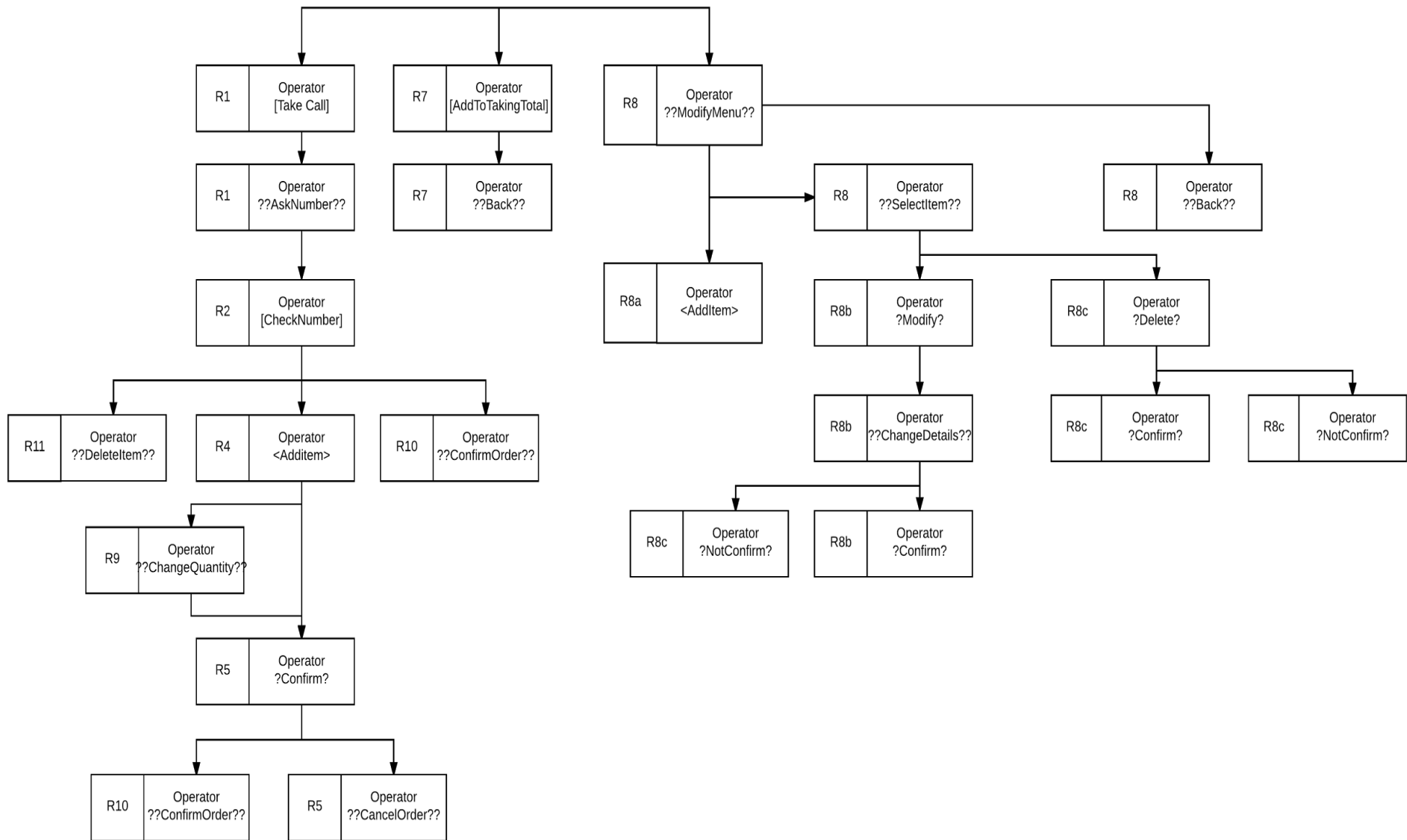
4.0 Design

4.1 Component Behaviour Trees (CBT)

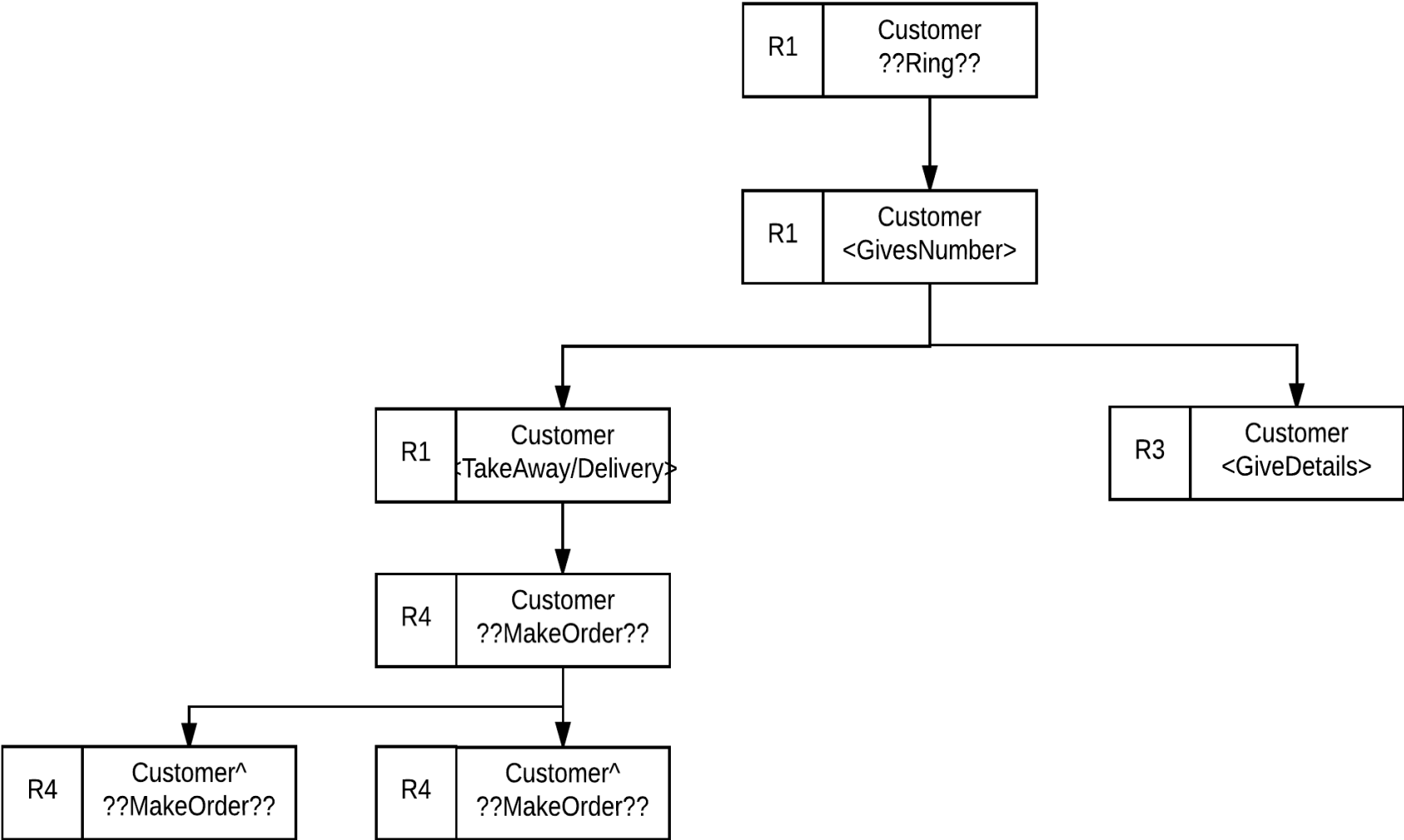
4.1.2 OrdSystem



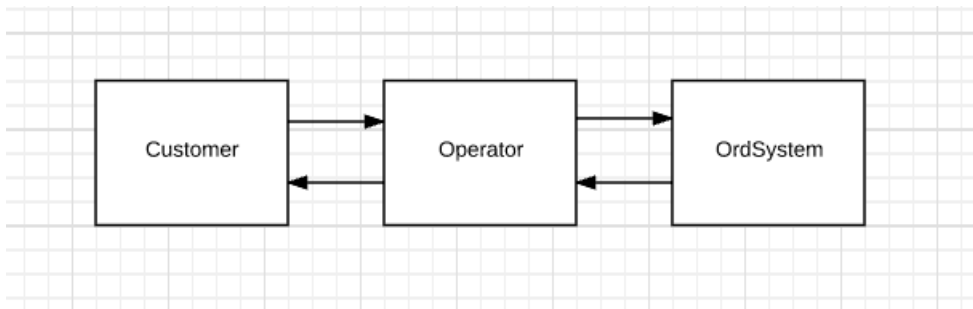
4.1.2 Operator



4.1.3 Customer

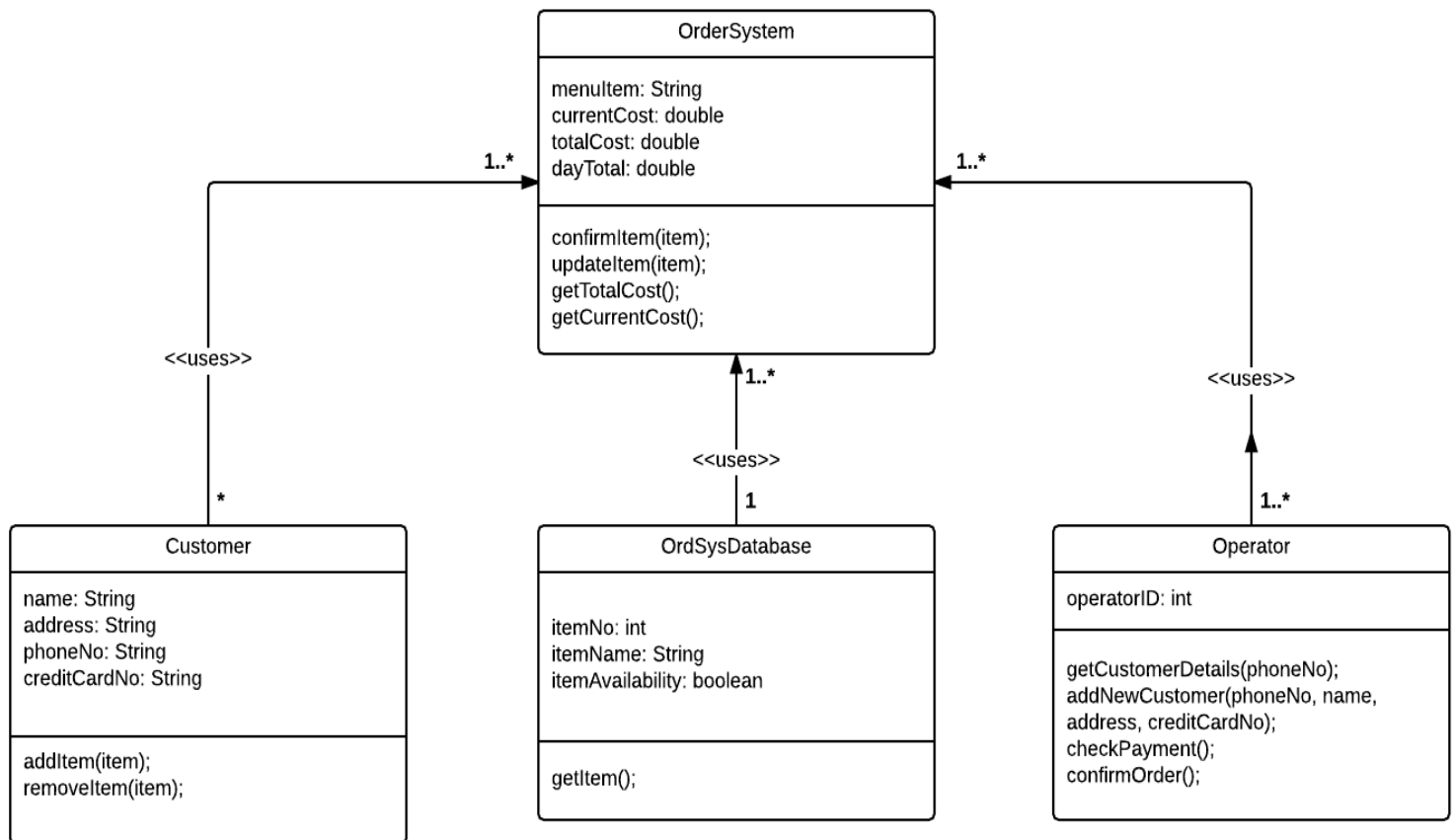


4.2 Component Interaction Network (CIN)



4.3 Class Diagram

Create a class diagram to model the high level behaviour and the detailed design of the system.



Defects:

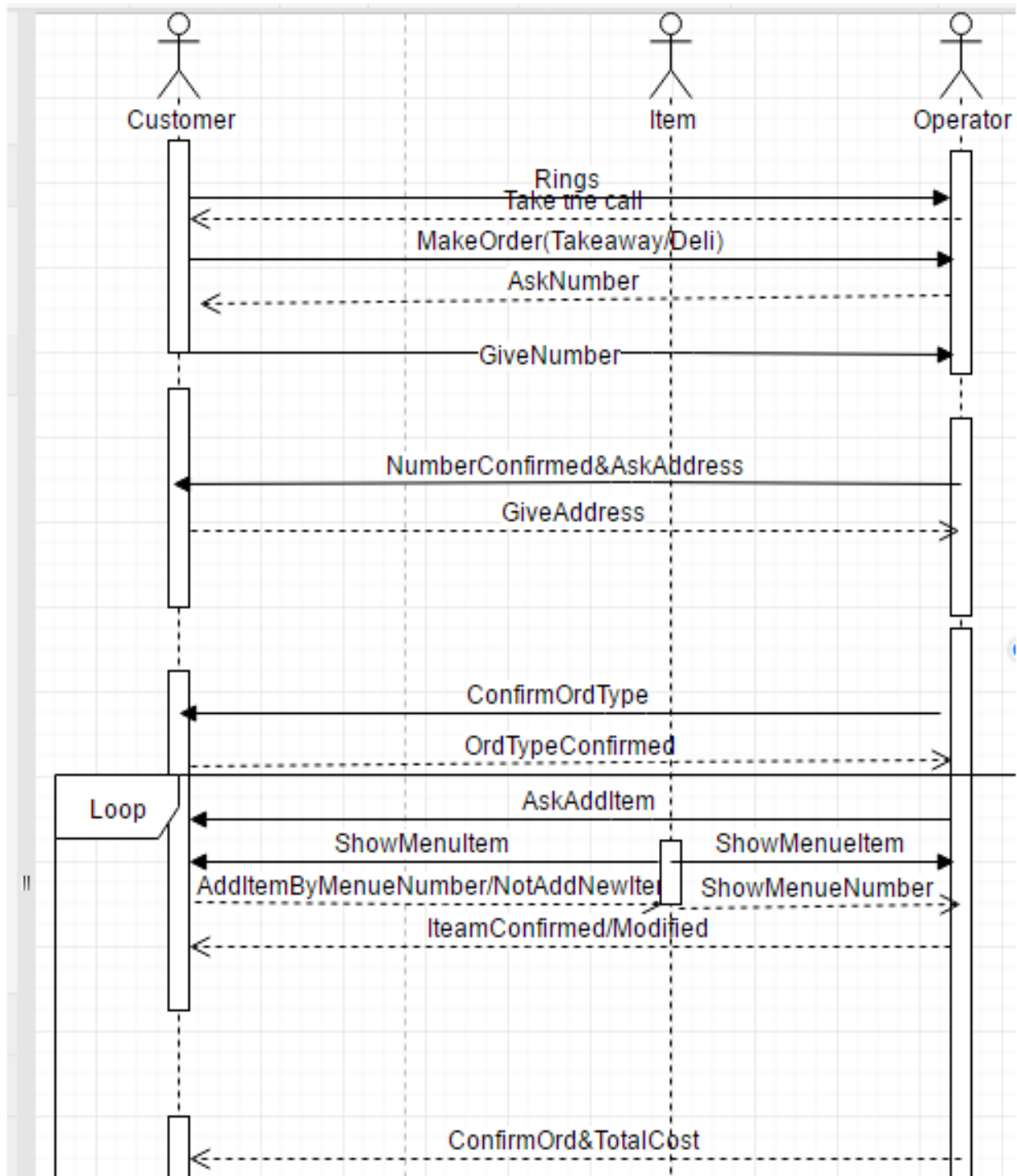
Class relationships

OrdSysDatabase requires more attributes

Add option to cancel whole order

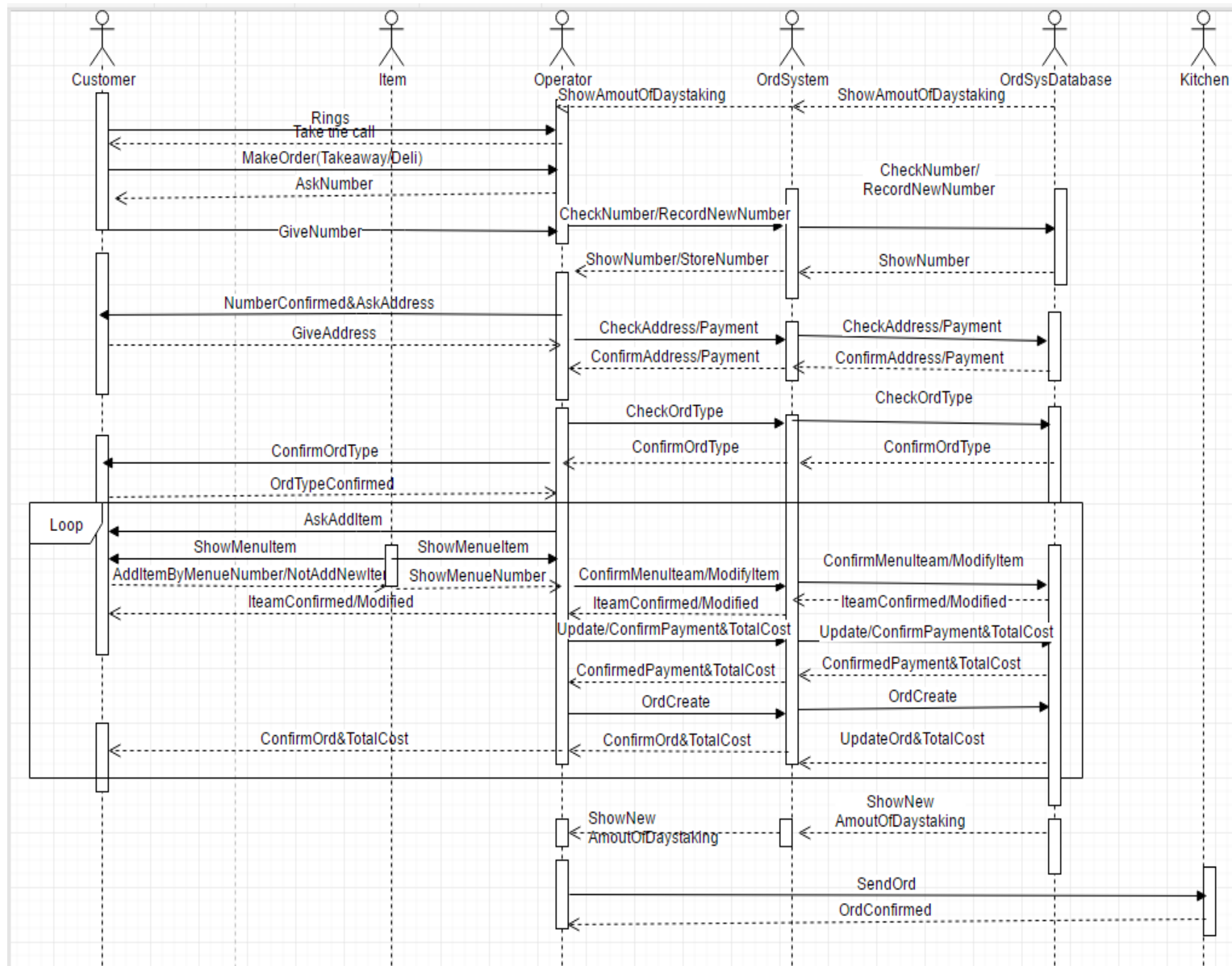
4.4 Sequence Diagrams

4.4.1 Sequence Diagram (Input a new Order)



4.4.2 Sequence Diagram

The sequence diagram represent the action and relations between customer, MenuItem, Operator, OrdSystem , OrdSystem database and Kitchen.



Defects

Defect NO.	Description	Req. No
1	Add MenuItem In each diagram	4
2	Change Loop scope	6-10
3	Add Modify Item in according Req8	8

4.5 Requirement Defects

4.5.1 Assumptions

- An operator should be able to delete an item from a current order.
- An operator should be able to cancel an order.
- All processes involving recording orders (takings) should be automated.

4.5.2 Problems and Revision Notes

In the DBT many errors were found, including how the IBT 'flowed', as well as functionality needed within the system. Details of this can be found in Section 5.

5.0 Error/Defect Analysis

Defect No.	Description	Requirement No	Phase
1(IBT)	Ambiguity /Change and refined Req 2,3,6 ,7 and 9 in IBT.	2,3,6,7,9 IBT	Translation
2(IBT)	Redundancy-Should be integrated	5&4,2 IBT	Integration
3(IBT)	Ambiguity /Change and refined Req 4 and redraw BT	4 BT	Ambiguity
4	Change in notation from state[] to input>< for customer and operator number input.	R1, R2	Integration (DBT)
5	Change in notation from state[] to condition?? For test whether number exists or not.	R2, R3	Integration (DBT)
6	Reduced ambiguity by specifying exact process followed if order doesn't exist. Changed notation to better reflect process ([]?<>)	R3	Integration (DBT)
7	Changed layout so rather than arrows pointing back steps, recursion using ^ is used.		General Change (DBT)
8	Changed position of entry of order type(takeaway/delivery) in order to better reflect the system.	R1	Integration (DBT)
9	Payment shouldn't be checked before an order is complete (can't know price)	R2	Integration, Assumption (DBT)
10	Switched payment check to end of order – defined new requirement outlining complete order checking.	R2, R10(New)	Integration (DBT)
11	Made assumption about ability to delete items from a current order.	R4, R11(New)	Integration (DBT)
12	Made assumption that an order may be cancelled.	R2, R4, R12(New)	Integration (DBT)

13	Changed AddToOrd Total to happen dynamically when adding an item to order.	R6, R4	Integration (DBT)
14	Find menu item to be redundant in regards to item.	R4	Integration (DBT)
15	Adding item changed to output, with option to change quantity if necessary.	R4, R9	Integration (DBT)
16	Operator confirming changed to if?? Which made clearer and simpler.	R5	Integration (DBT)
17	OrdSytem adding to order now takes input from operator addItem.	R4	Integration (DBT)
18	Made order dynamically update orderCost based on items and quantity. It does this both when adding and deleting items.	R6, R11	Integration (DBT)
19	Implemented Recursion in order for ability to add/delete multiple products to/from order. (allows mistake correction)	R10, R7, R4	Integration (DBT)
20	Added ability to cancel an order in progress.	R12	Integration (DBT)
22	Split R8 into its separate flows.	R8	Integration (DBT)
23	Implemented Ability to navigate through the modify sub-system as well as back to main ordering system.	R8	Integration (DBT)
24	Made changes regarding input/output/condition/event notation to better reflect process.	R8	Integration (DBT)
25	Class relationships	OrdSysDatabase requires more attributes	Add option to cancel whole order
26	Add MenuItem In each diagram	R4	
27	Change Loop scope	R6-10	
28	Add Modify Item in according Req8	Add Modify Item in according Req8	