

Software Analysis and Design (ADLC2)

Continuous Assignment

TEAM 4

Thune Htet Naing

Hninn Ei Khaing

Shi Xu

Lee Wei Xiang Vincent

Cheng Jun Long

Software Analysis and Design (ADLC2)

- Use case Diagram

Description on the purpose of use case

- Analysis Model

Sequential Diagram

Class Diagrams

- Design Model

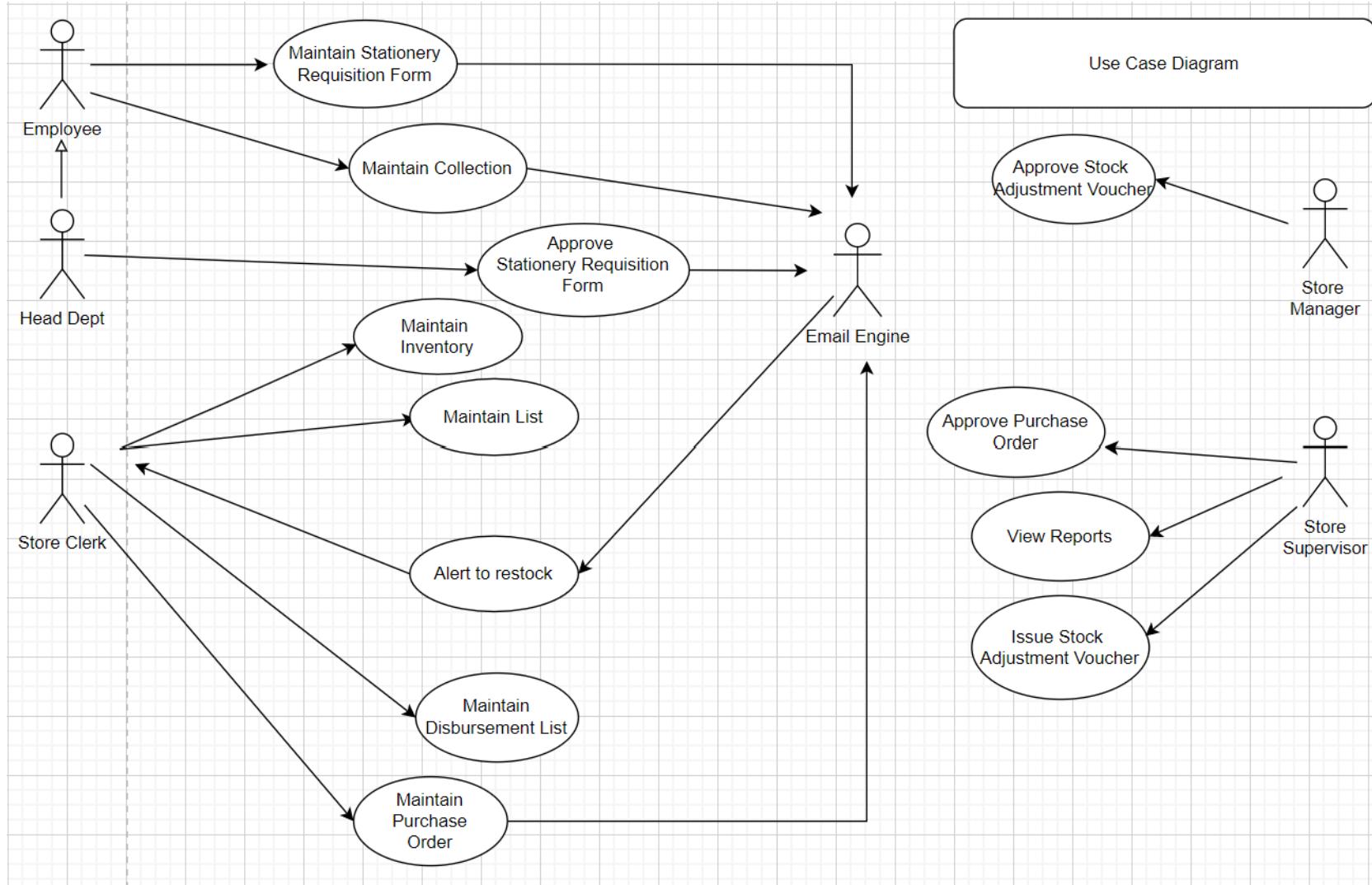
Statechart Diagram

- Data Model (Database Design)

Logical Data Model (ERD)

Data Dictionary

Use Case Diagram (Summary)



Maintain Stationery Requisition Form:

For employees in various departments to request for Stationery according to the stationery catalogue. Will be sent to the department head to approve first, after that sent to store clerk who will handle the request.

Maintain Collection:

Employees are allowed to choose the stationery collection point from six different locations. (Department still can change their preferred location point as well)

Approve Stationery Requisition Form:

Department Head to approve the stationary Requisition form.

Maintain Inventory:

Store clerk to record down the transactions from disbursement list, Delivery Order and Inventory Adjustment Voucher.

Maintain List:

(Includes Supplier, Department, Price)

Annually (or as needed), Store Clerk can go to the system to update the various list in the database.

Alert to Restock:

Alert Store Clerk when stocks inventory is low and restocking is required.

Maintain Disbursement List:

Actual quantity to be delivered (to the same Department) for each item will be noted in the disbursement list.

Maintain Purchase Order:

Store clerk will fill up the purchase order Form depending on the stock level of the items every two weeks.

Approve Purchase Order:

Alert the Store Supervisor to approve the pending Purchase Order.

View Reports:

Sales report including the amount of stationery ordered grouped by categories and stationery requisition trend of each department.

Issue Stock Adjustment Voucher:

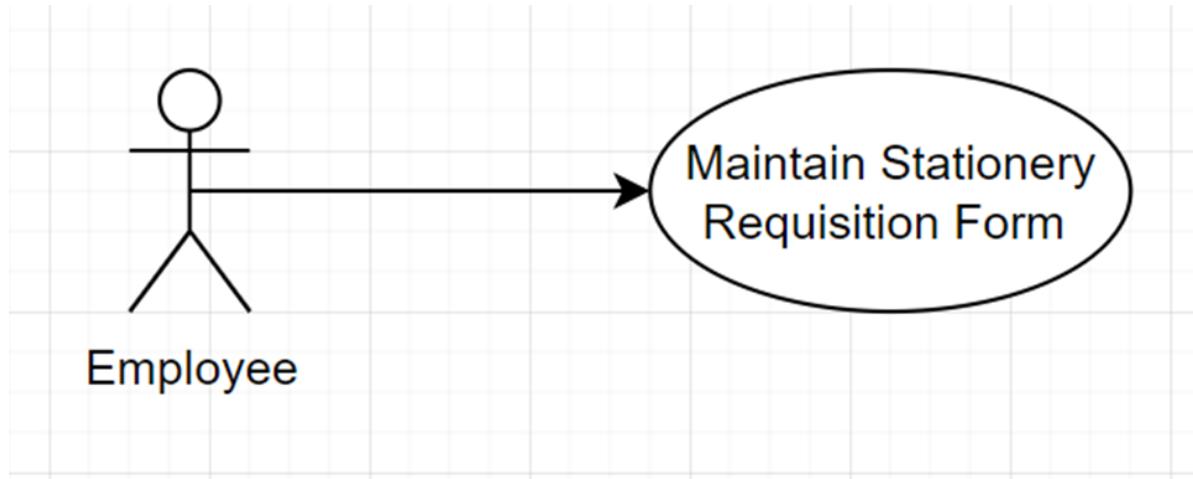
The store supervisor will issue a stock adjustment voucher for anything under \$250 per item adjustment.

Approve Stock Adjustment Voucher:

The store manager approves and issue the stock adjustment voucher that is above \$250.

Use Case: Maintain Stationery Requisition Form

For employees in various departments to request for Stationery according to the stationery catalogue. Will be sent to the department head to approve first, after that sent to store clerk who will handle the request.



Flow of Events:

The use case begins when the Employee selects the "Maintain Stationery Requisition Form" option from the Main Menu.

Basic flow – Create Requisition Form:

1. *Employee enters the following details into the create screen of Stationery Requisition Form*
 - *Dept code*
 - *Employee number*
 - *Item code*
 - *Requested quantity*
2. *System validates format of data*
3. *System retrieves dept name, employee name, employee email address, item description, name of dept head*

4. System displays dept name, employee name, employee email address, item description, name of dept head
5. System stores employee's requisition form with a unique Requisition Form Num.
6. System sets the status of this requisition form as "Pending approve"
7. System displays successful message
8. System send email to employee who prompt requisition form to inform that the status of requisition form is "Pending approve"
9. System sends email to department head(according to which Dept code on requisition form), informing that there is a new request awaiting their approval

Alternate Flow:**1.1 Update Requisition Form**

1. Employee enter Requisition Form Num
2. System retrieve Requisition Form record
3. System displays Requisition Form details
4. Employee modifies Requisition Form fields
5. Employee submits the changes
6. System updates the Requisition Form record
7. System display successful message
8. System sends email to employee, informing updating is successful.

1.2 Delete Requisition Form

1. Employee enter Requisition Form Num
2. System retrieve Requisition Form record
3. System displays Requisition Form details
4. Employee Confirms the delete operation

5. System delete the Requisition Form data
6. System display successful message
7. System send email to employee, informing delete is successful.

1.3 Retrieve Requisition histories

1. Employee enters Employee Number
2. System validates Employee Number
3. System displays current and previous stationery requisitions with its disbursement status

1.4 Requisition Form Exists

(Create Requisition Form flow) System finds an existing “Pending approve” Requisition Form with the same Employee Number and same Dept Code.

1. System displays an error message “You still have a requisition form awaiting for approve. Creation not allowed”
2. Employee can either click to send an urgent email to department head, or cancel the operation at which the use case terminates

1.5 Requisition Form Not Exists

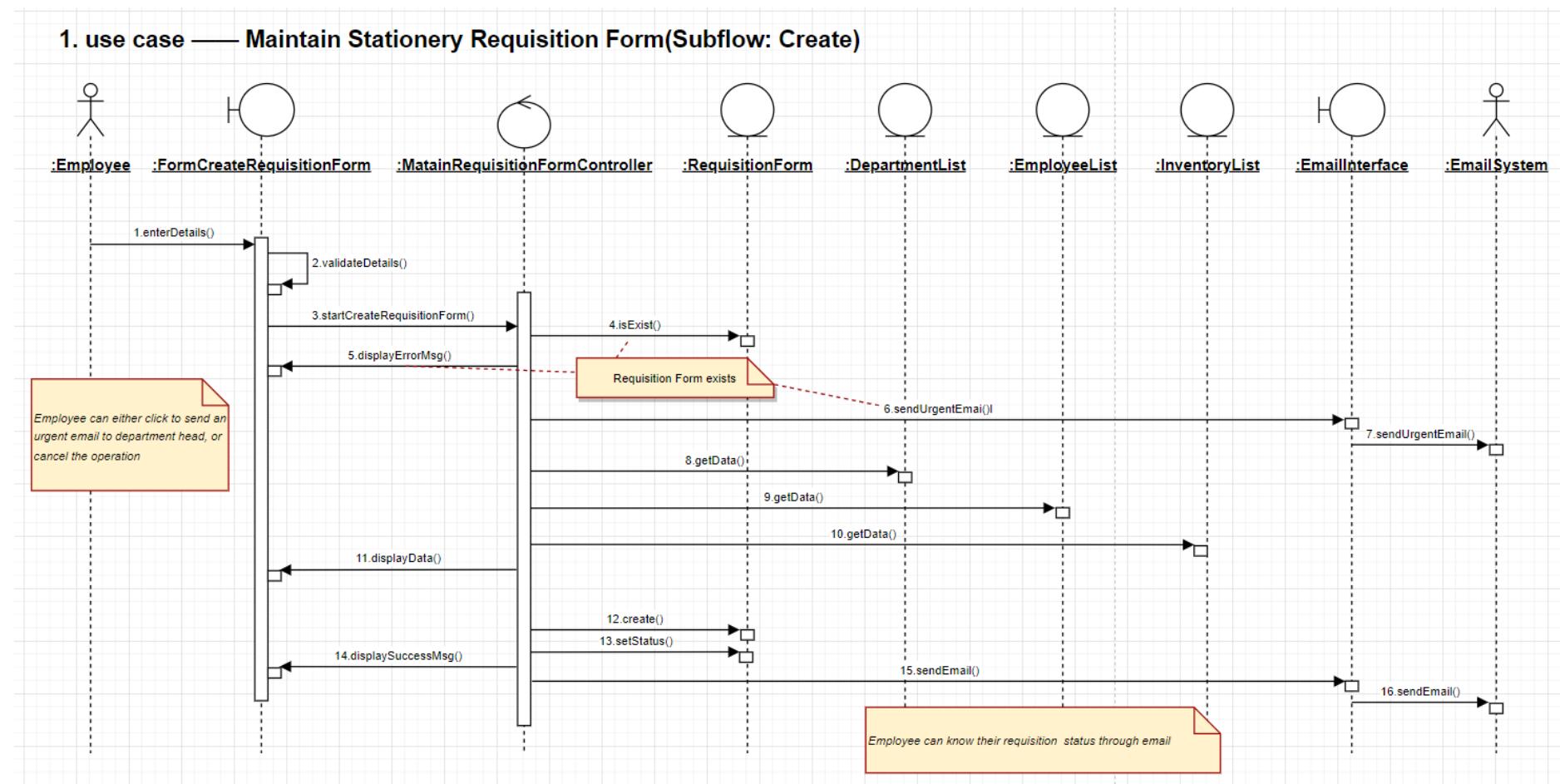
(Update or delete Requisition Form flow) System could not locate the Requisition Form or the status of this Requisition Form is not “Pending approve”(After head approved, status will be “approved”).

1. System display an error message, “Requisition Form not found”
2. Employee can either re-enter the Requisition Form Num, or cancel the operation at which the use case terminates

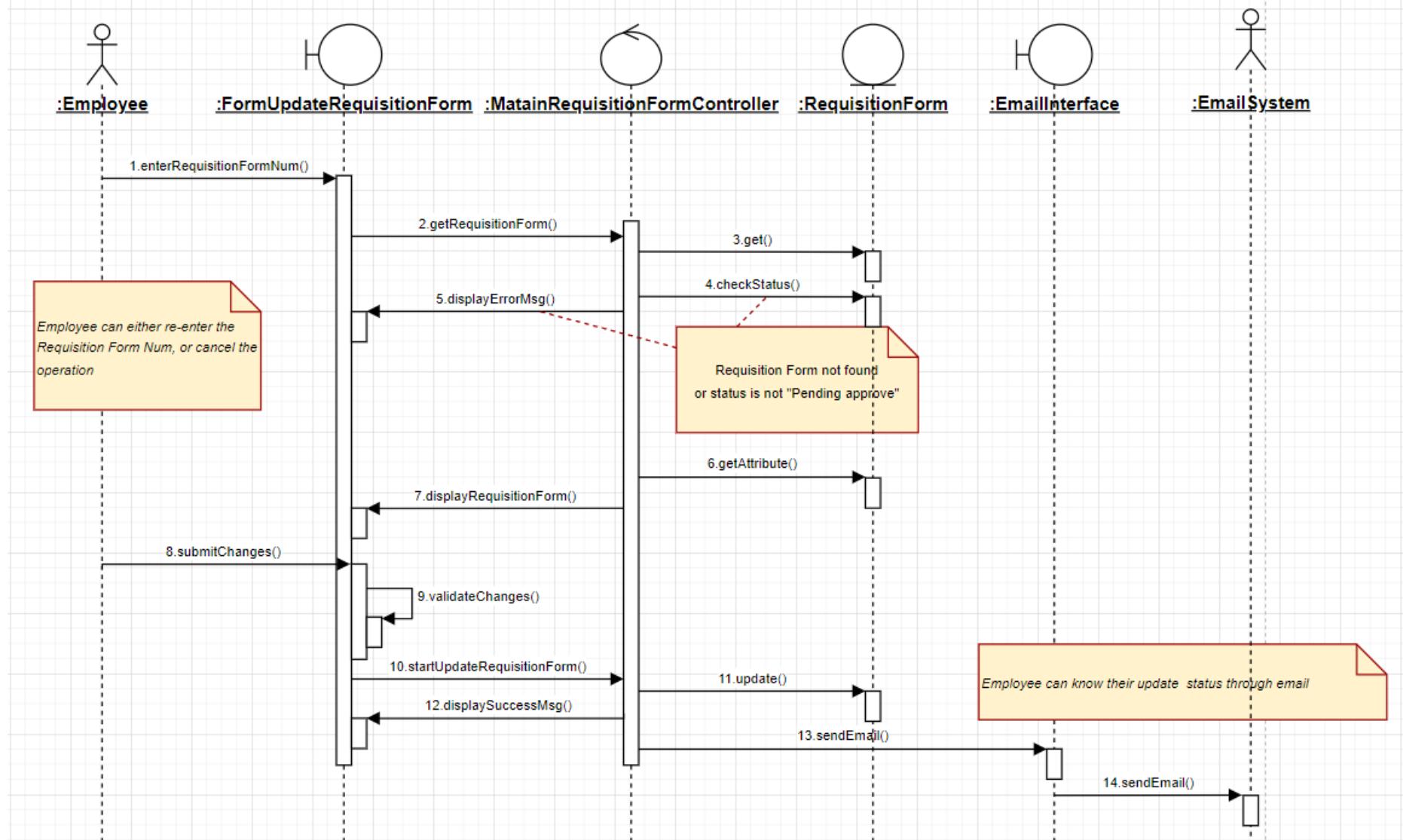
1.6 Requisition histories Not Exist

(Retrieve Requisition Histories flow) System could not locate any current or previous requisition form.

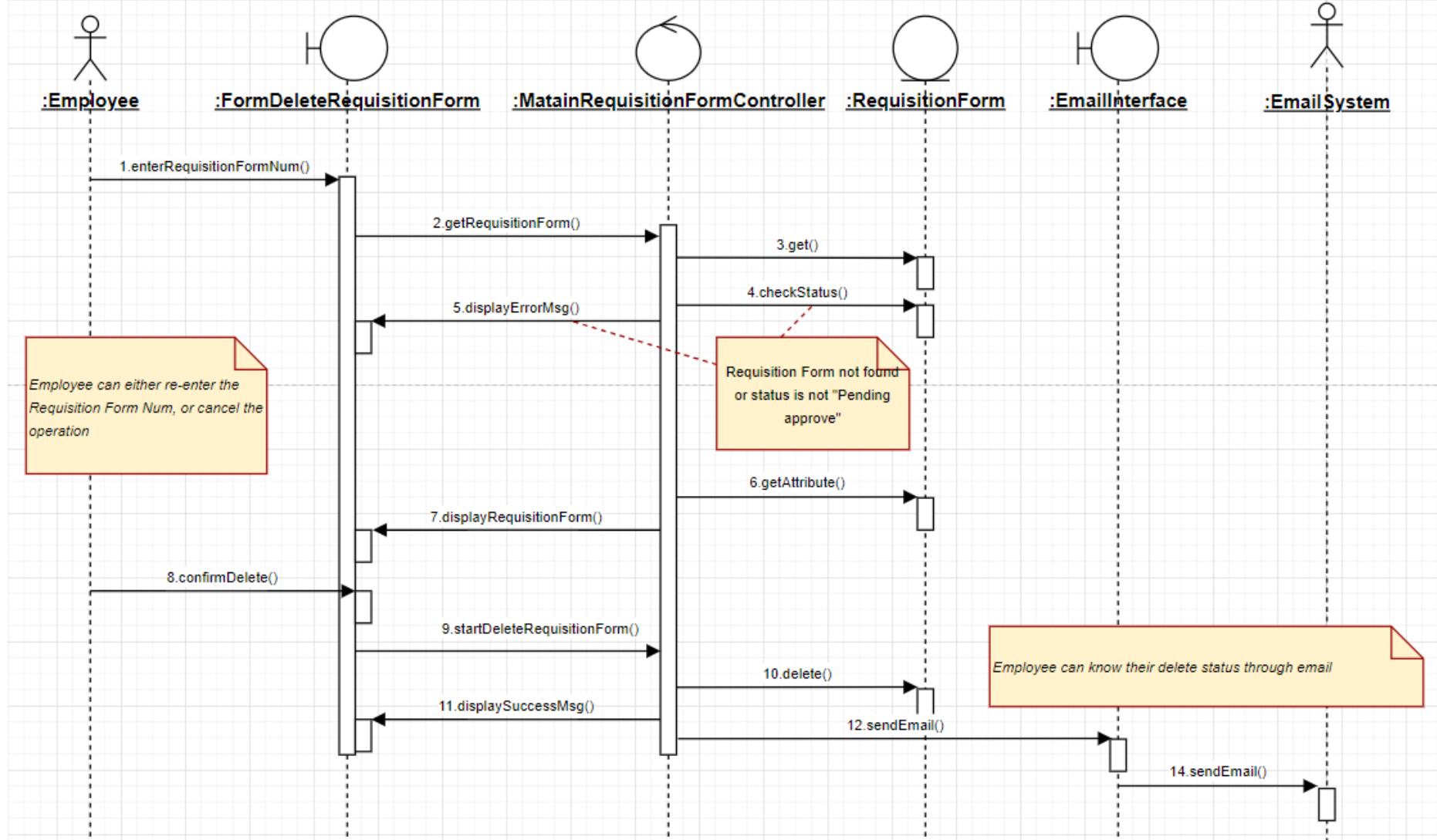
1. System display an error message, “No requisition form”
2. Employee can either create a new requisition form or cancel the operation at which the use case terminates

Sequence Diagram:**1. use case —— Maintain Stationery Requisition Form(Subflow: Create)**

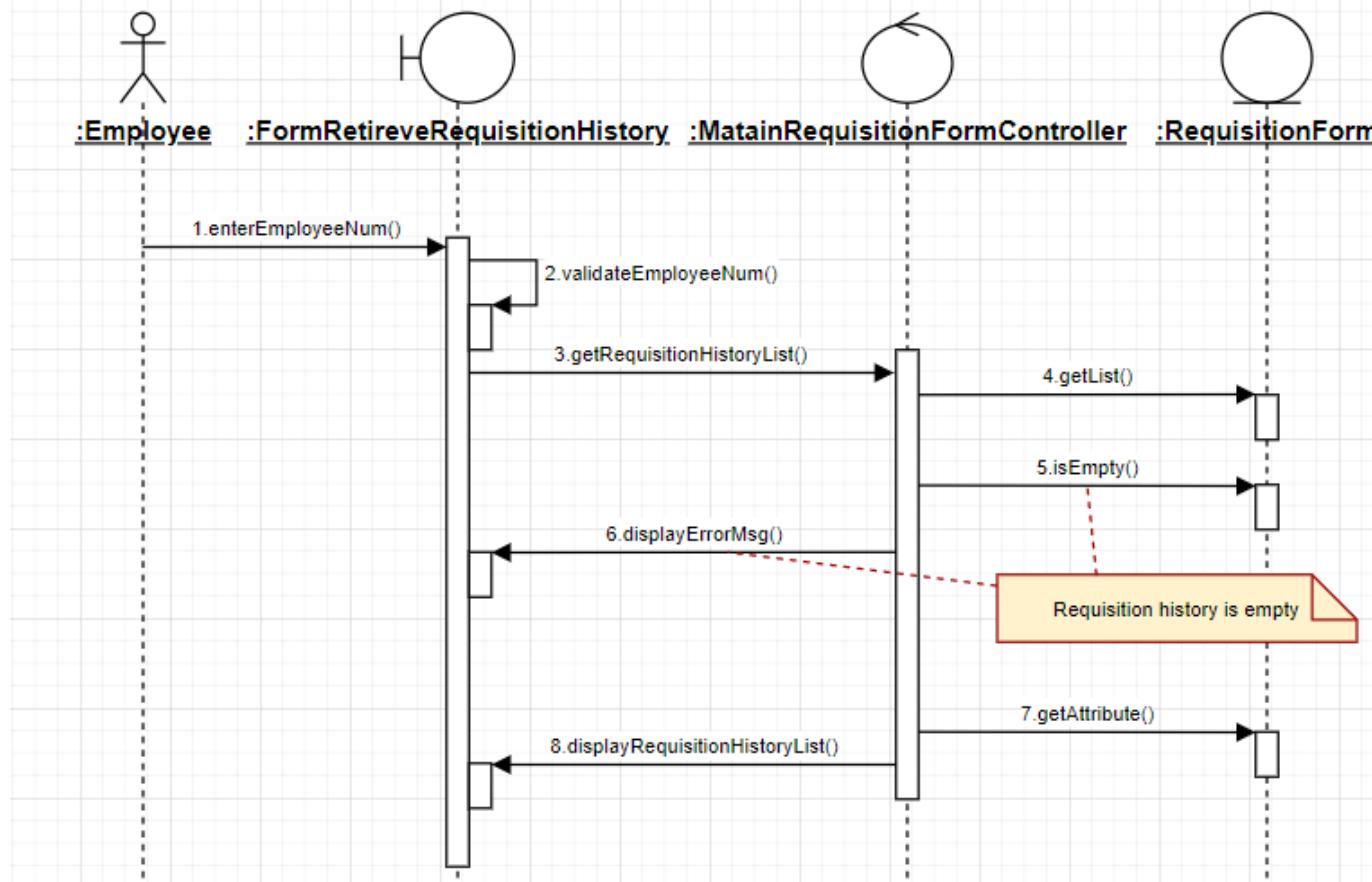
2. use case — Maintain Stationery Requisition Form(Subflow: Update)

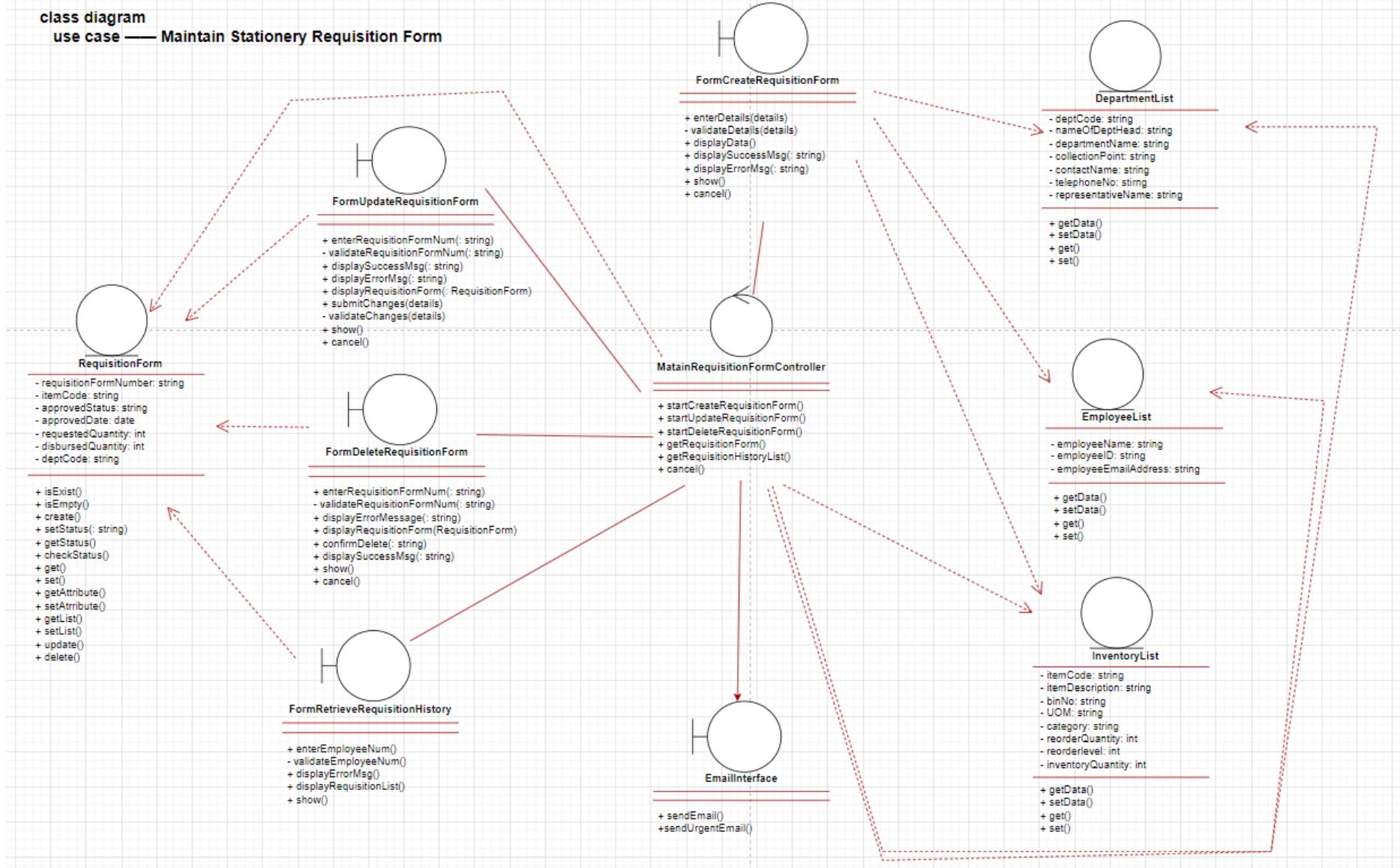


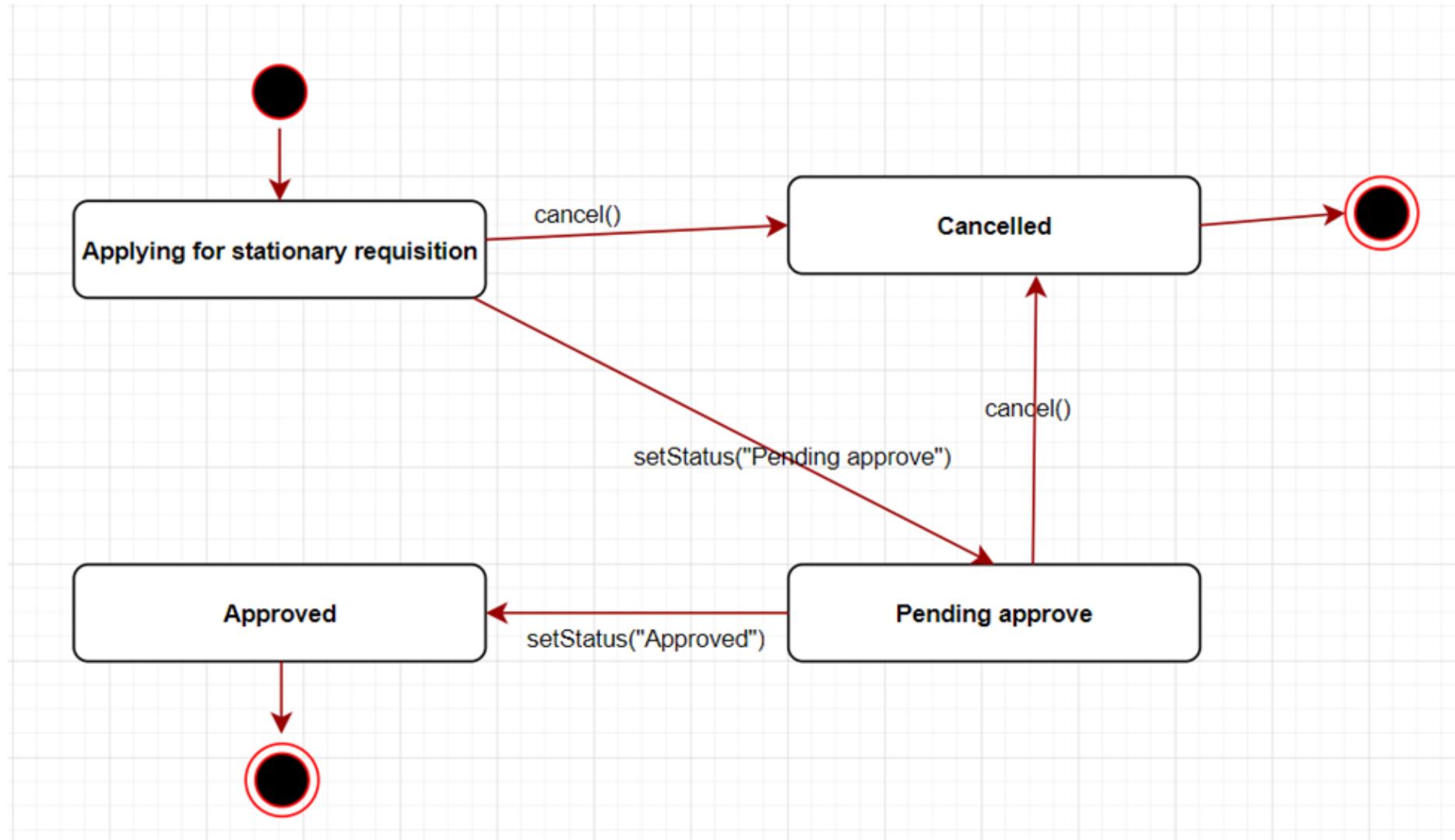
3. use case — Maintain Stationery Requisition Form(Subflow: Delete)



4. use case — Maintain Stationery Requisition Form(Subflow: Retrieve Requisition Histories)





Statechart diagram(Maintain Requisition Form):

Use Case: Maintain Collection

Employees are allowed to choose the stationery collection point from six different locations. (Department still can change their preferred location point as well)



Sub Flow: Create

Basic Flow:

1. *Employee key in the following details to the Book Pick-Up Location Screen*
 - *Requisition Number*
 - *Pick-Up location*
 - *Representatives Name*
2. *System validates format of data*
3. *System retrieves the Requisition Number*
4. *System stores the Representative Name*
5. *System create & stores the CollectionPoint*
6. *System send CollectionPoint record to Store Clerk via Email*
7. *System displays successful message*

Alternate Flow:

Requisition Number Not Found

At Step 3 of Basic Flow

System Display a Message that Requisition Number is Not Found

****Employee can only Book/Update Pick-Up Location Screen if:**

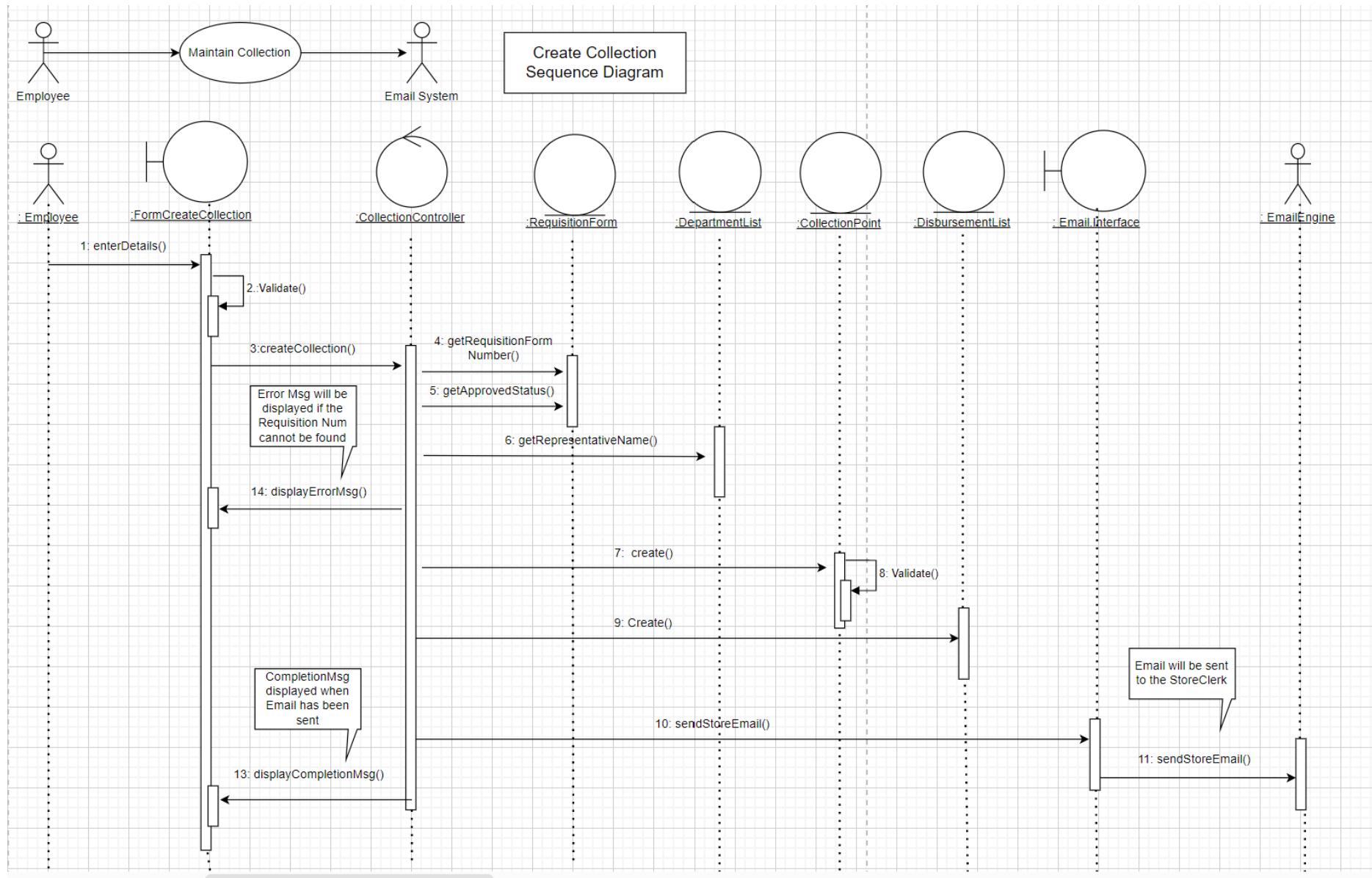
- 1) Once the Store Clerk has processed the order and ready to dispatch:
 - a) A 'null' location will be created, when employee comes and book, they will be updating the system
- 2) Store clerk has already processed the order and ready to dispatch
 - a) Error Message of Requisition Number Not Found if Employee tries to Book a Pick-Up location
- 3) Can only update 24 hours before the stipulated Dispatch time
 - a) Error Message of Unable to Update Pick Up Location

After Submitting the Stationery Requisition Form, Employee will have to select the pick up location

There will be a set of 6 places where the employee can choose from with the stipulated time

Also nomination of the POC to collect the stationery needs to be included.

Upon submission, if the employee likes to amend the Pick-Up location prior to the 2 hours before delivery, he can go online and update.



SubFlow: Update

Basic Flow:

8. Employee key in the following details to the Book Pick-Up Location Screen
 - Requisition Number
 - Pick-Up location
 - Representatives Name
9. System validates format of data
10. System retrieves the Requisition Number
11. System stores the Representative Name
12. System update & stores the CollectionPoint
13. System send CollectionPoint record to Store Clerk via Email
14. System displays successful message

Alternate Flow:

Requisition Number Not Found

At Step 3 of Basic Flow

System Display a Message that Requisition Number is Not Found

**Employee can only Book/Update Pick-Up Location Screen if:

- 4) Once the Store Clerk has processed the order and ready to dispatch:
 - a) A 'null' location will be created, when employee comes and book, they will be updating the system
- 5) Store clerk has already processed the order and ready to dispatch
 - a) Error Message of Requisition Number Not Found if Employee tries to Book a Pick-Up location
- 6) Can only update 24 hours before the stipulated Dispatch time

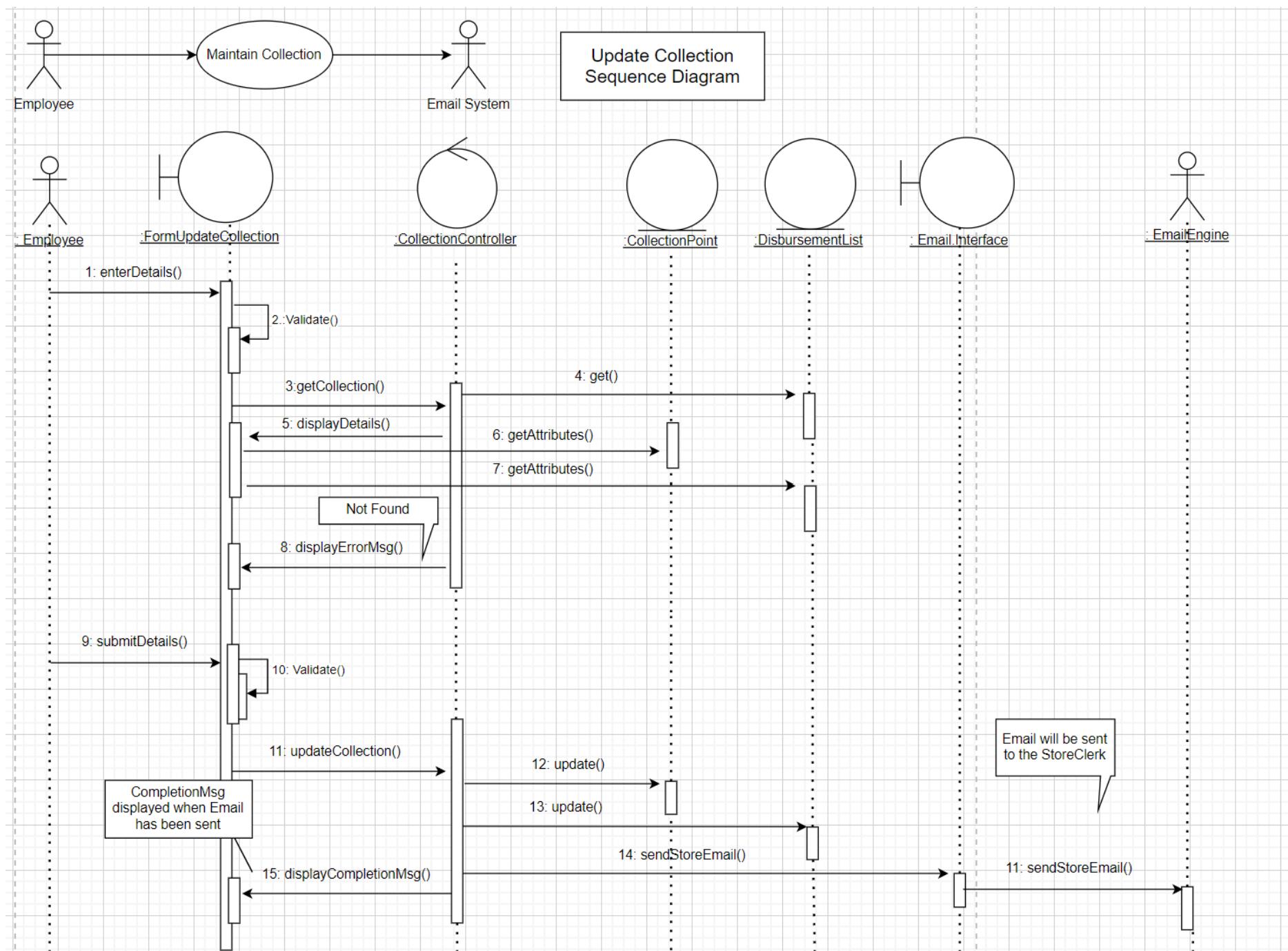
a) Error Message of Unable to Update Pick Up Location

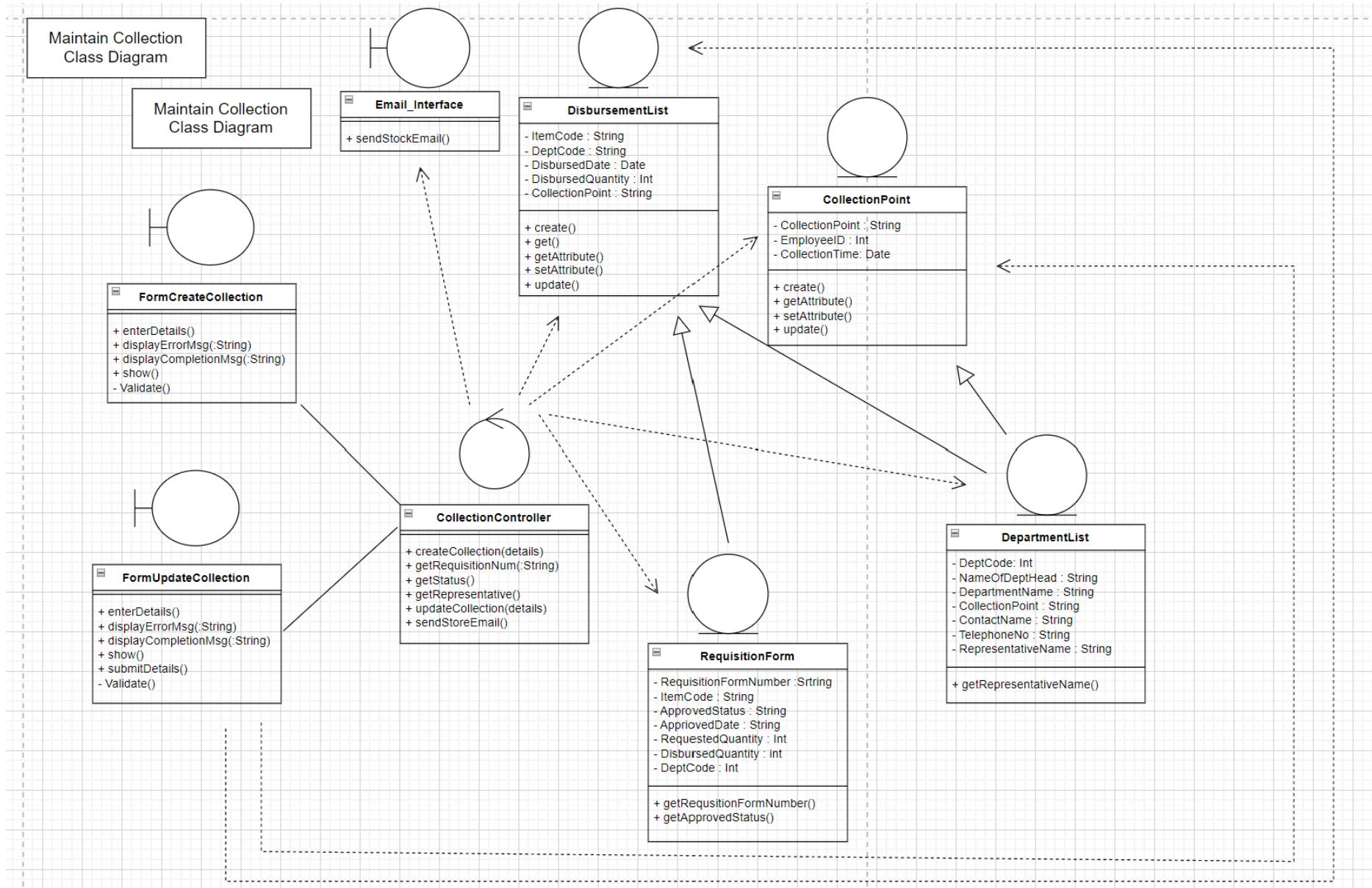
After Submitting the Stationery Requisition Form, Employee will have to select the pick up location

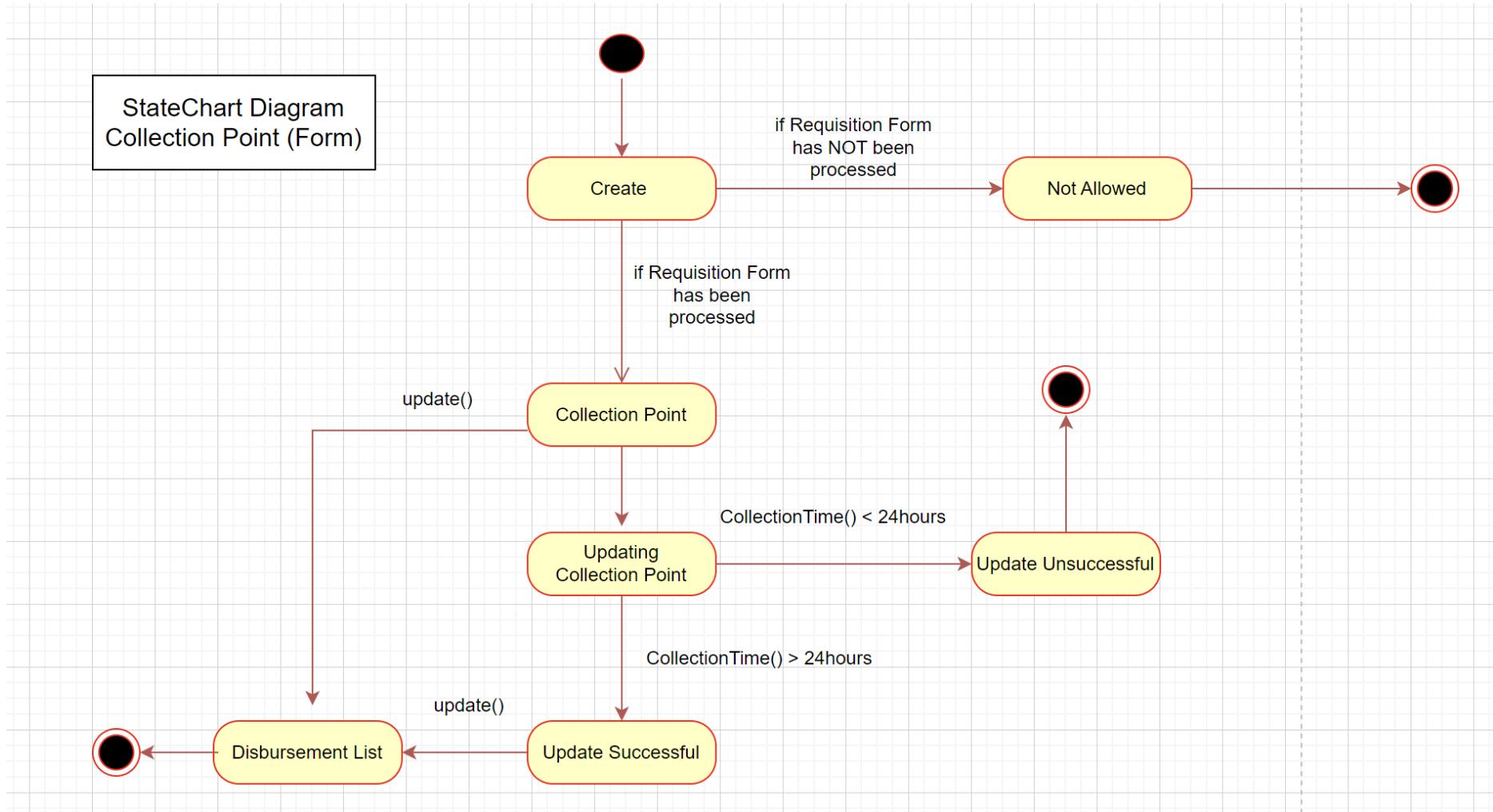
There will be a set of 6 places where the employee can choose from with the stipulated time

Also nomination of the POC to collect the stationery needs to be included.

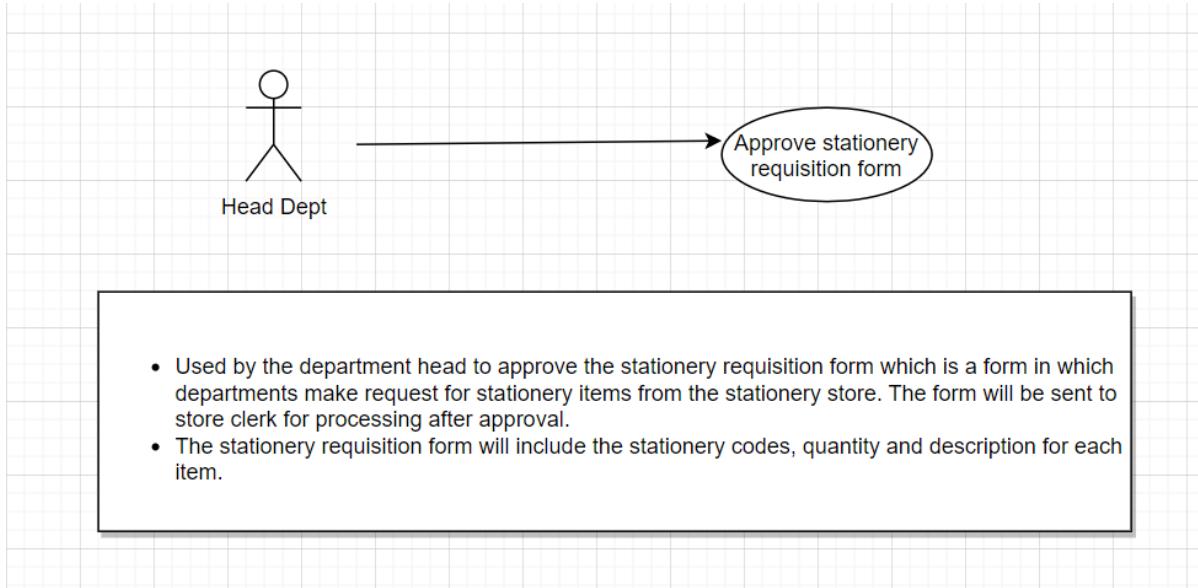
Upon submission, if the employee like to amend the Pick-Up location prior to the 2 hours before delivery, he can go online and update



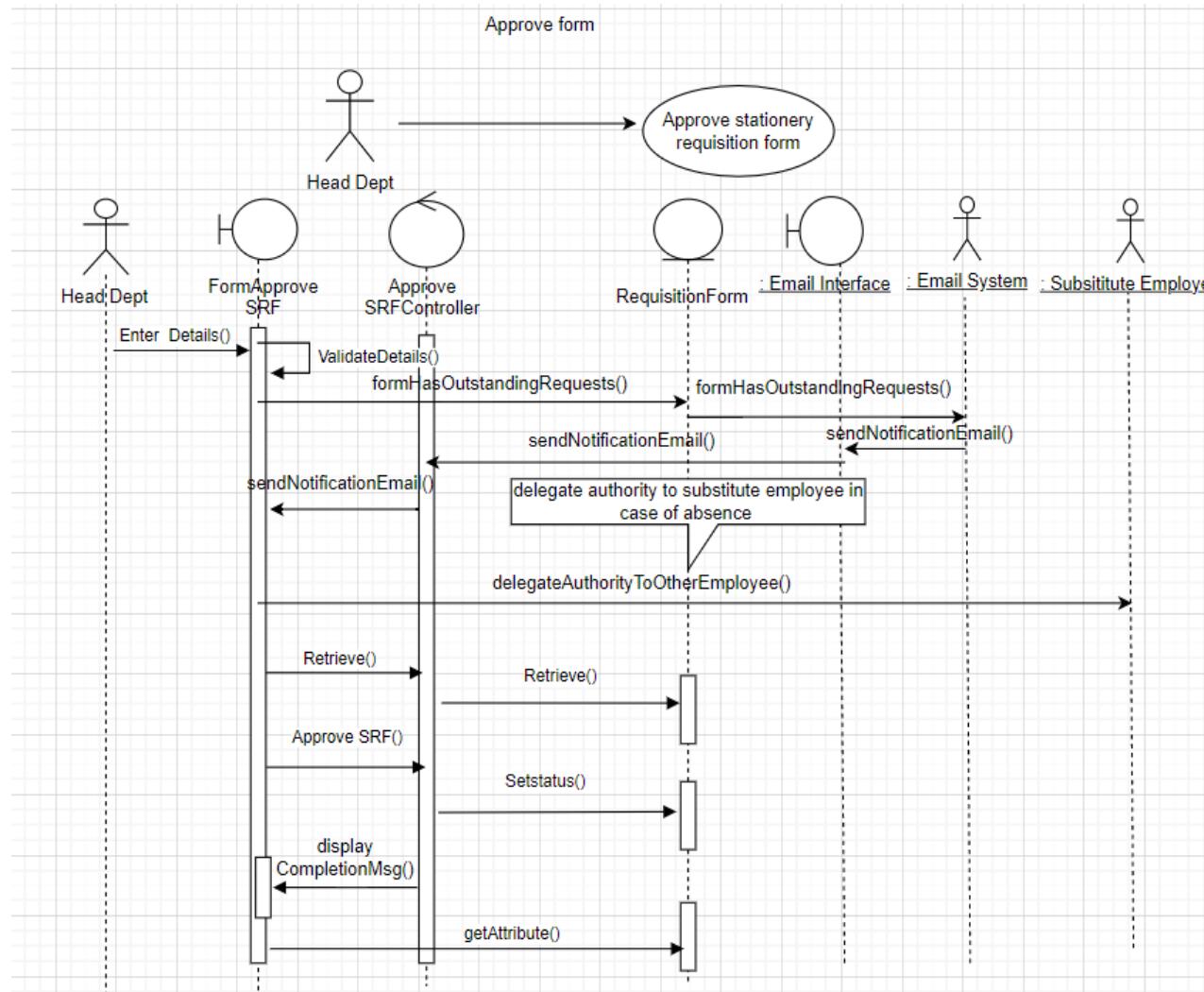




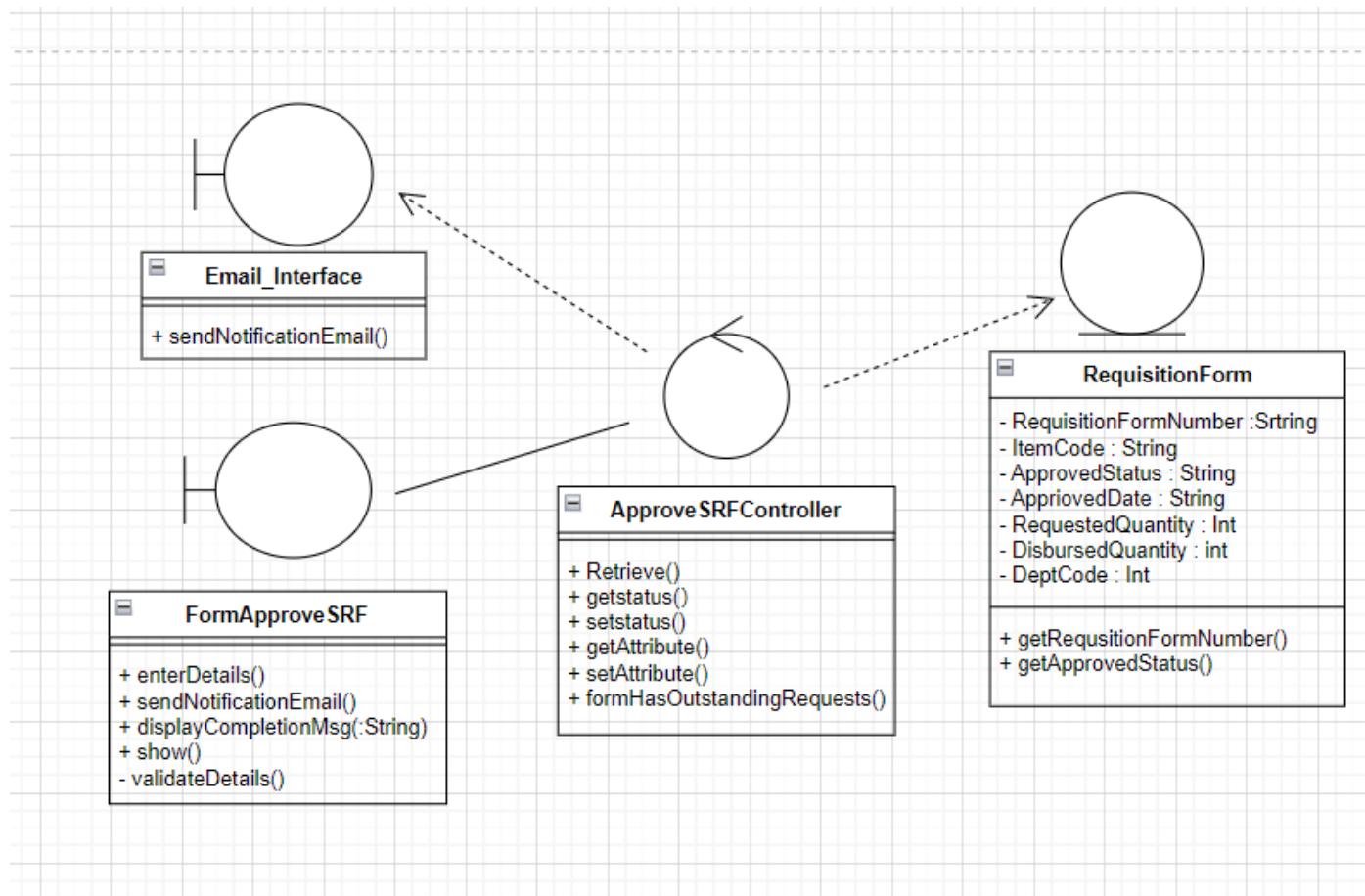
Use Case: Approve stationery Requisition Form



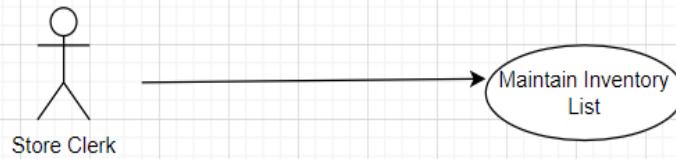
Sequence diagram for Approve Stationery Requisition form



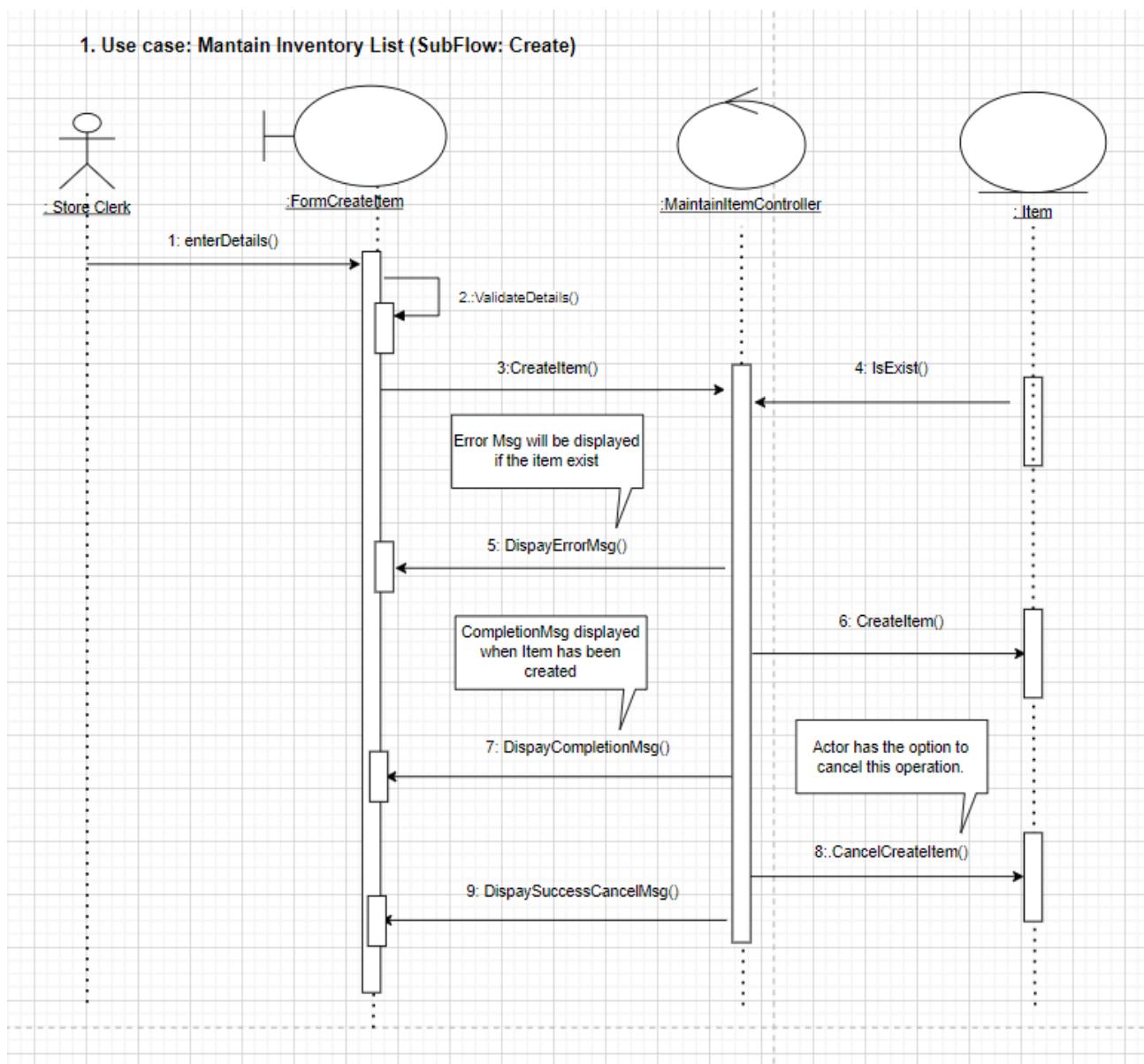
Class diagram for Approve Stationery Requisition form



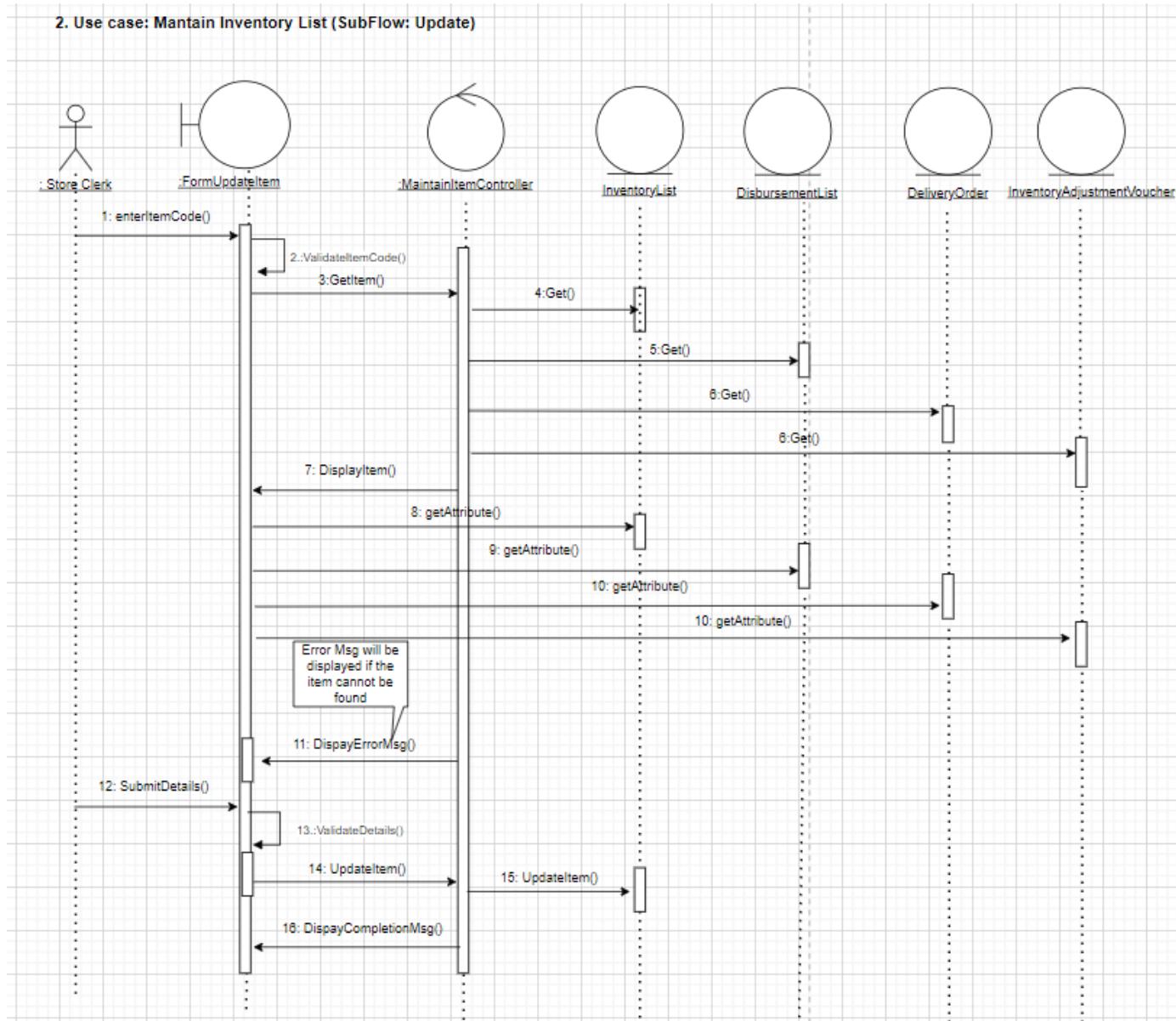
Use Case: Maintain Inventory List



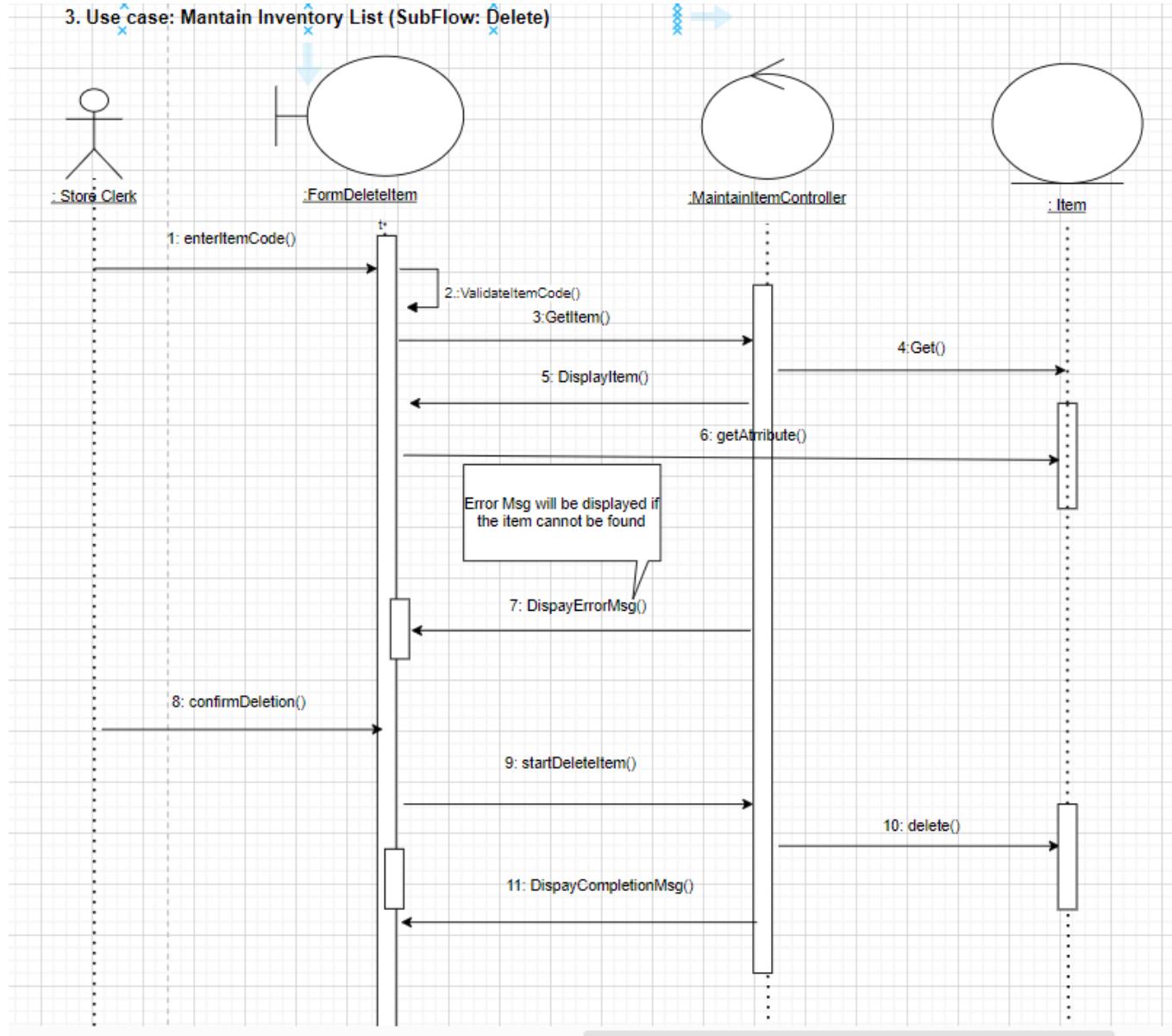
- This use case has CRUD functionality.
- Used by the store clerk to note down the transactions based on the disbursement list and delivery order shipments to be recorded in stock cards (inventory list). It allows for employees to create, update and delete records for all transactions of each item in the inventory list.
- Actual quantity issued for each item, the balance of each item, date and where the item came from will be recorded in the inventory list.

1. Use case: Mantain Inventory List (SubFlow: Create)


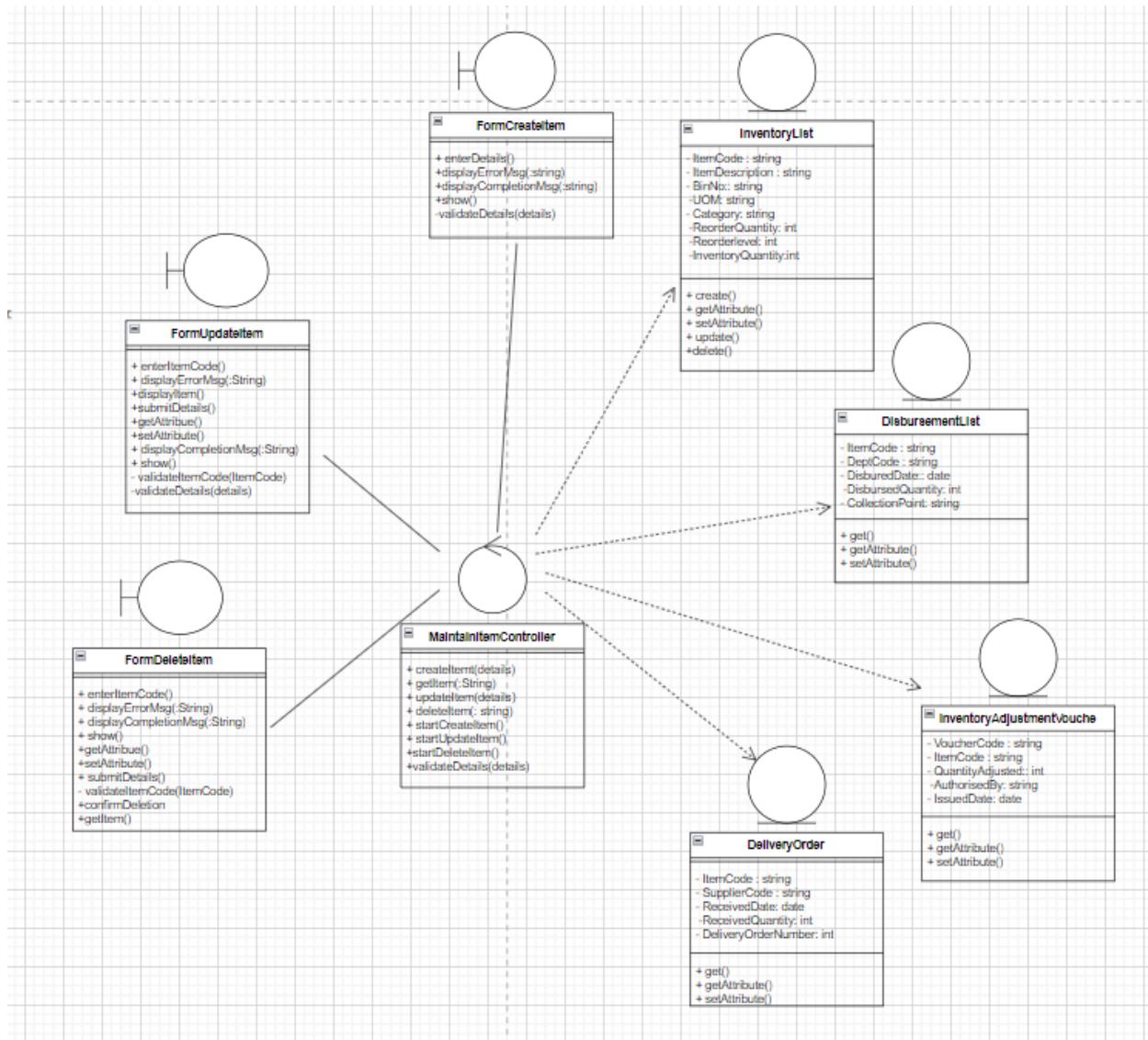
2. Use case: Maintain Inventory List (SubFlow: Update)



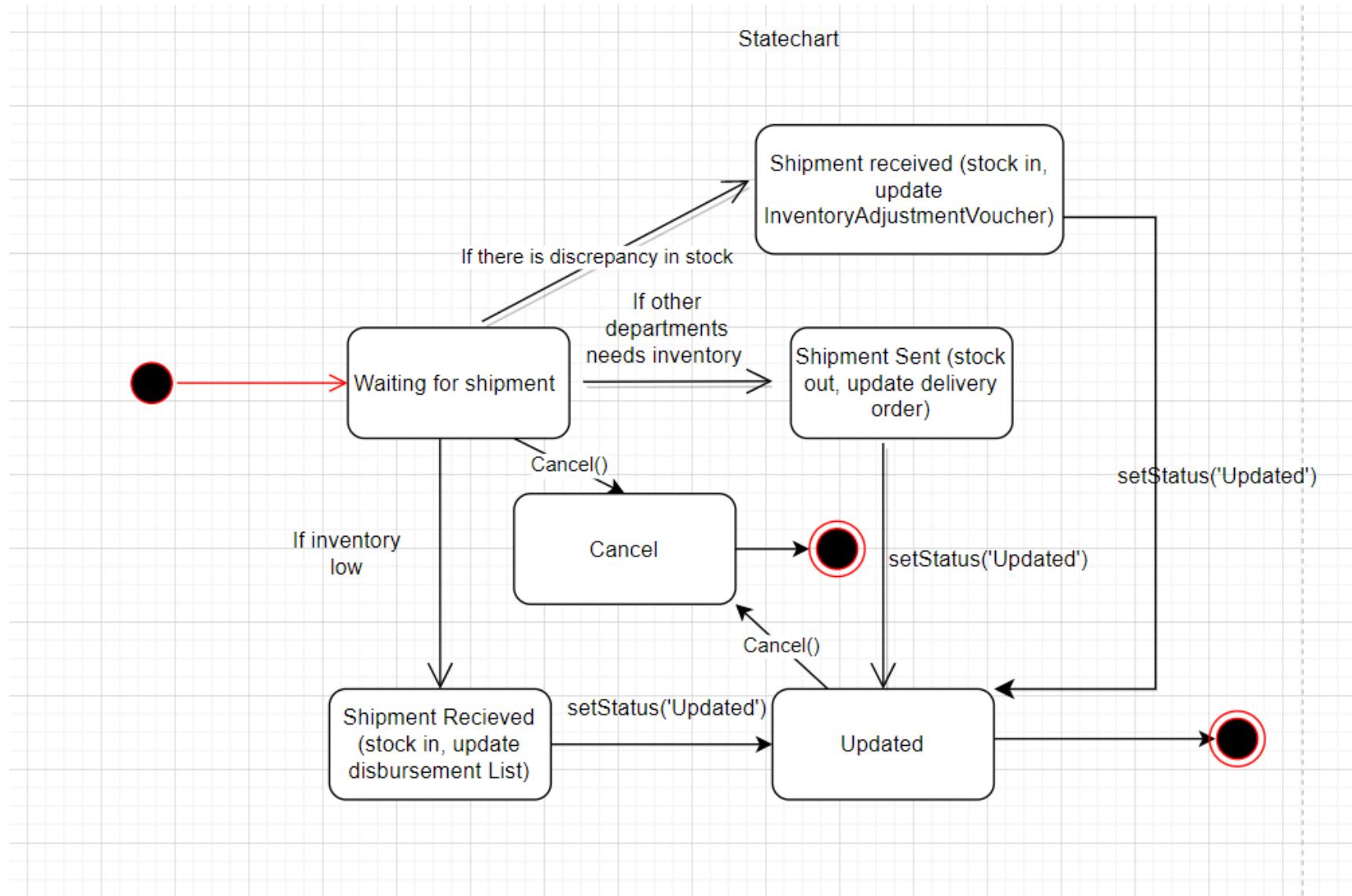
3. Use case: Maintain Inventory List (SubFlow: Delete)



4. Class Diagram for Use Case: Maintain Inventory List

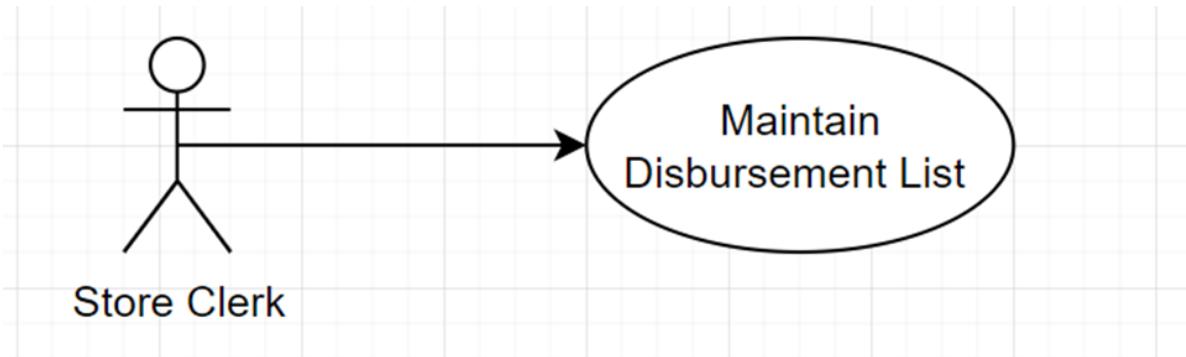


5. Statechart for Use Case: Maintain Inventory List



Use Case: Maintain Disbursement List

Actual quantity to be delivered (to the same Department) for each item will be noted in the disbursement list.



Flow of Events:

The use case begins when the store clerk selects the “Maintain Disbursement List” option from the Main Menu.

Basic flow – Create Disbursement List:

1. *Store clerk enters the following details into the create screen of Disbursement List:*
 - *Date of disbursement(DDMMYY)*
 - *Department code, Collection point*
 - *Item code, Quantity*
2. *System validates format of details*
3. *System retrieves Department name, Representative name, Stationery description*
4. *System displays Department name, Representative name, Stationery description*
5. *System stores disbursement list with a unique disbursement number*
6. *System sets the status of this disbursement list as “Pending for delivery”*
7. *System displays a successful message*

Alternate flow:**1.1 Update Disbursement List:**

1. Store clerk enters disbursement number
2. System retrieve disbursement list record
3. System display disbursement list
4. Store clerk modifies disbursement list
5. Store clerk submits changes
6. System validate data format
7. System update disbursement list
8. System displays a successful message

1.2 Delete Disbursement List:

1. Store clerk enters disbursement number
2. System retrieve disbursement list
3. System displays disbursement list
4. Store clerk confirms the delete operation
5. System delete disbursement list
6. System display successful message

1.3 Print Disbursement List(*Print type: print for Delivering*):

1. Store clerk enters disbursement number
2. System retrieve disbursement list
3. System displays disbursement list
4. Store clerk chose print type: print for Delivering
5. Store clerk confirms the print operation

6. System sends disbursement list to print
7. System sets status to “Delivering”
8. System displays a successful message

1.4 Print Disbursement List(*Print type: print for Testing*):

1. Store clerk enters disbursement number
2. System retrieve disbursement list
3. System displays disbursement list
4. Store clerk chose print type: *print for Testing*
5. Store clerk confirms the print operation
6. System sends disbursement list to print
7. System displays a successful message

1.5 Disbursement List Exists

(Create Disbursement List flow) System finds an existing disbursement list with the same disbursement date and department code.

1. System displays an error message, “Already has a pending delivery list on this day for this department. Confirm to create another disbursement list ”
2. Store clerk can either click to create a new disbursement list, or cancel the operation at which the use case terminates

1.6 Disbursement List Not Exists

(Update, Delete or two types of Print Disbursement List flow) System could not locate the disbursement list.

1. System displays an error message, “Disbursement List Not Found”
2. Store clerk can either re-enter disbursement number, or cancel the operation at which the use case terminates

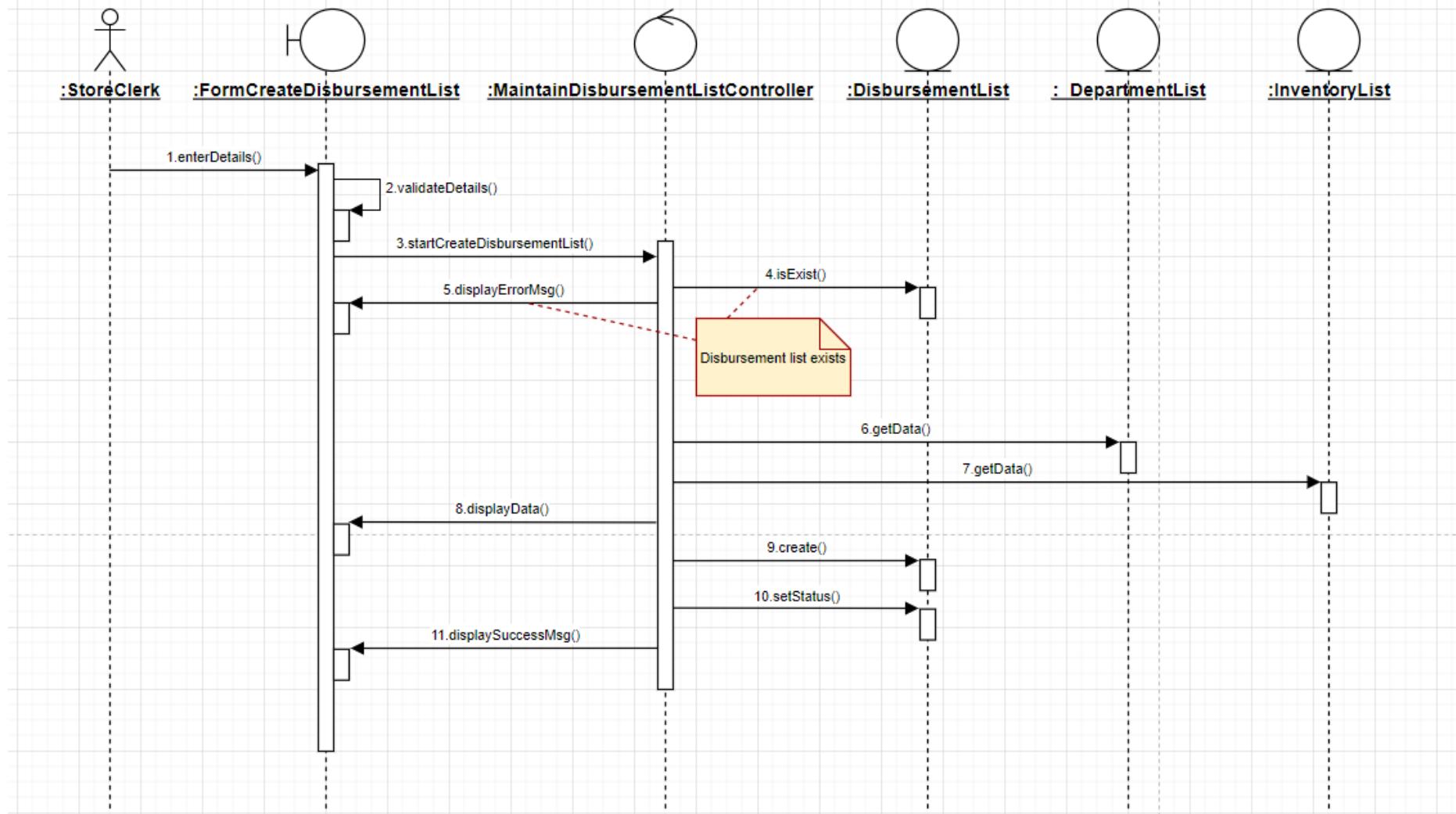
1.7 Disbursement List Status Changed

(Update, Delete or Print Disbursement List(Print type: print for delivering) flow) The statuses of this disbursement list is not “Pending for delivery”. After printing, status will be “Delivering”. After representative employee signs, status will be “Successfully delivered”)

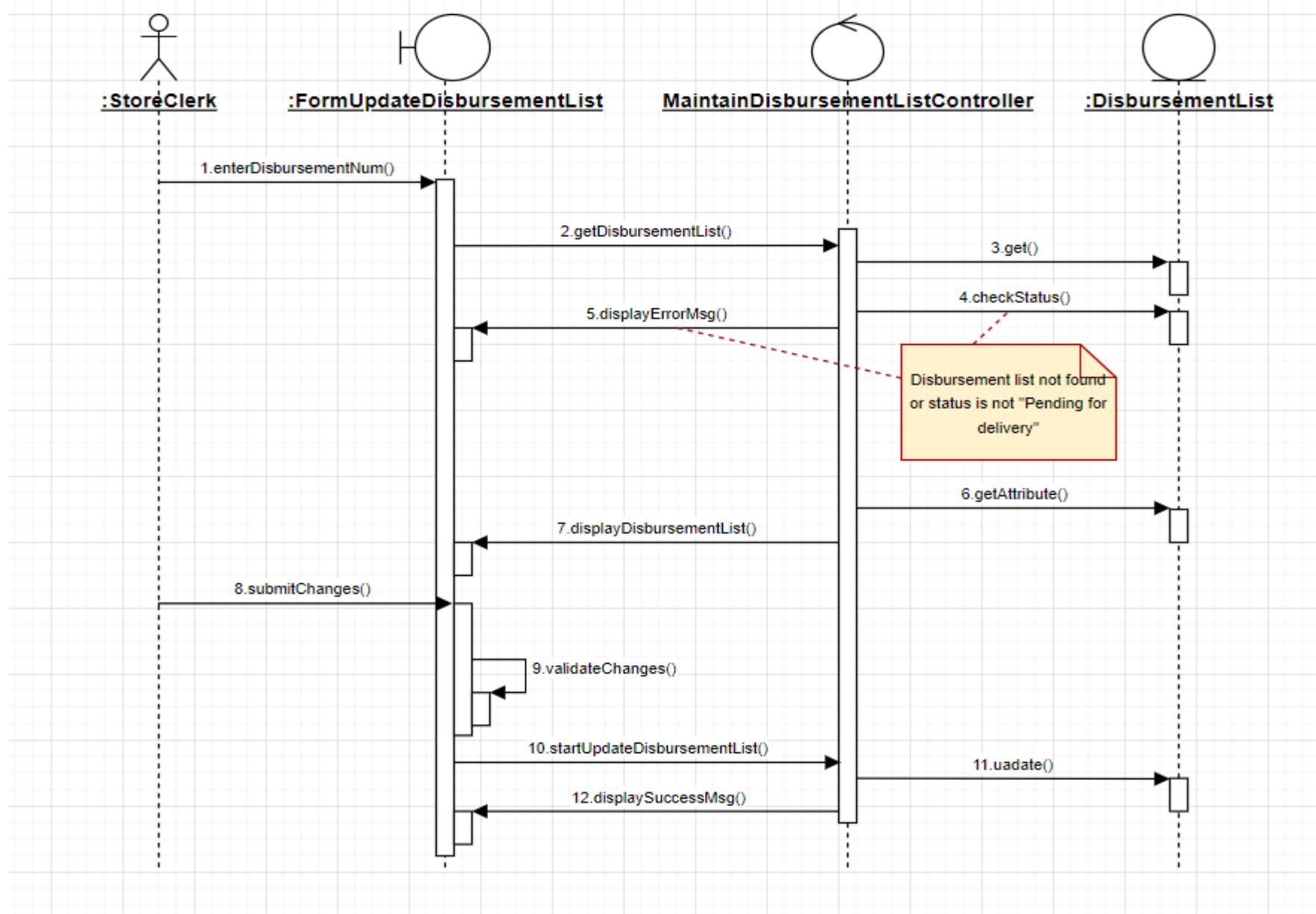
1. System display an error message, “Status changed. Can not modify or print.”

Store clerk can either prompt a operation request to store supervisor (cannot modify disbursement list until supervisor approve), or cancel the operation at which the use case terminates

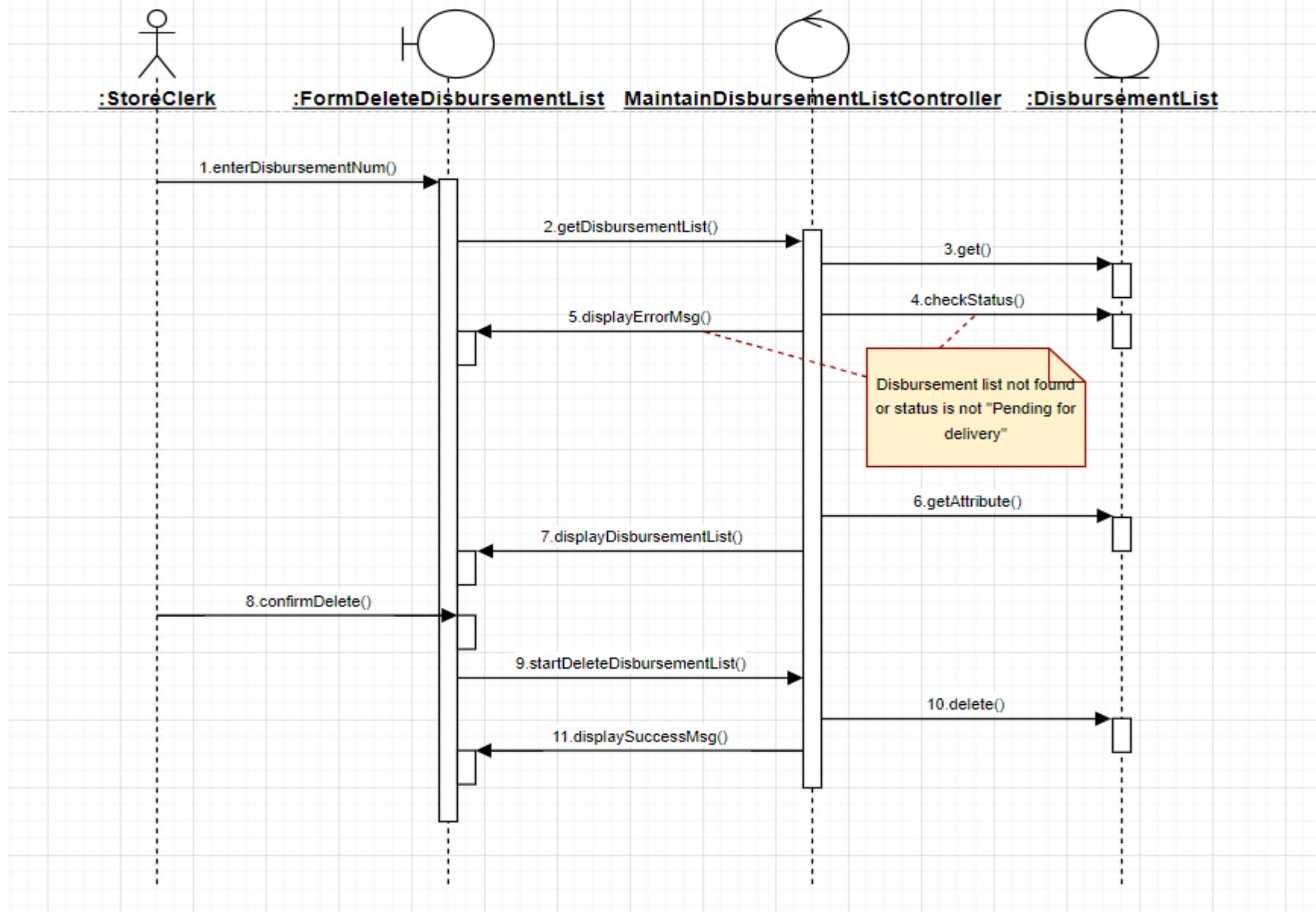
1. Use Case: Maintain Disbursement List (Subflow: Create)



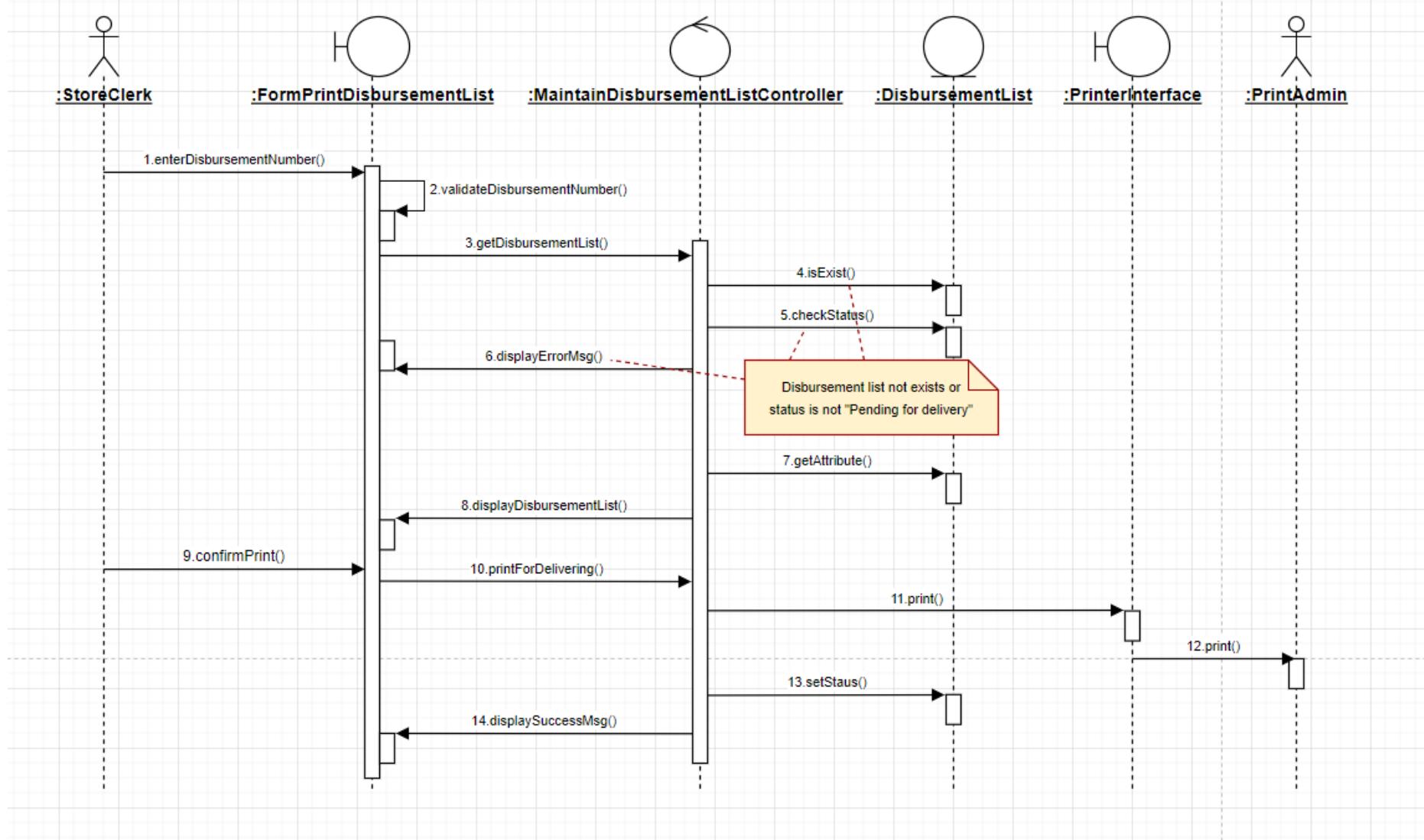
2. Use Case: Maintain Disbursement List (Subflow: Update)



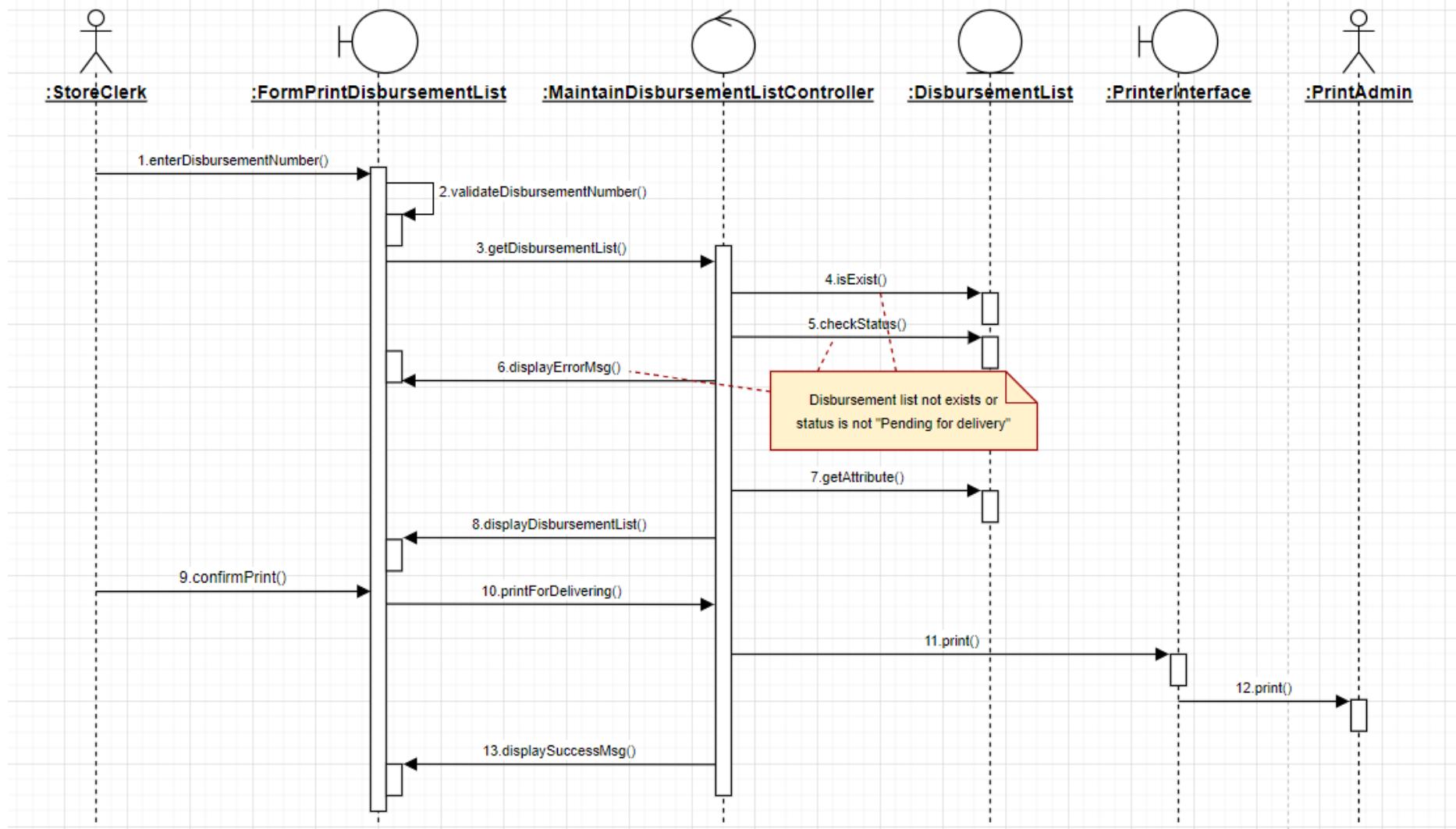
3. Use Case: Maintain Disbursement List (Subflow: Delete)

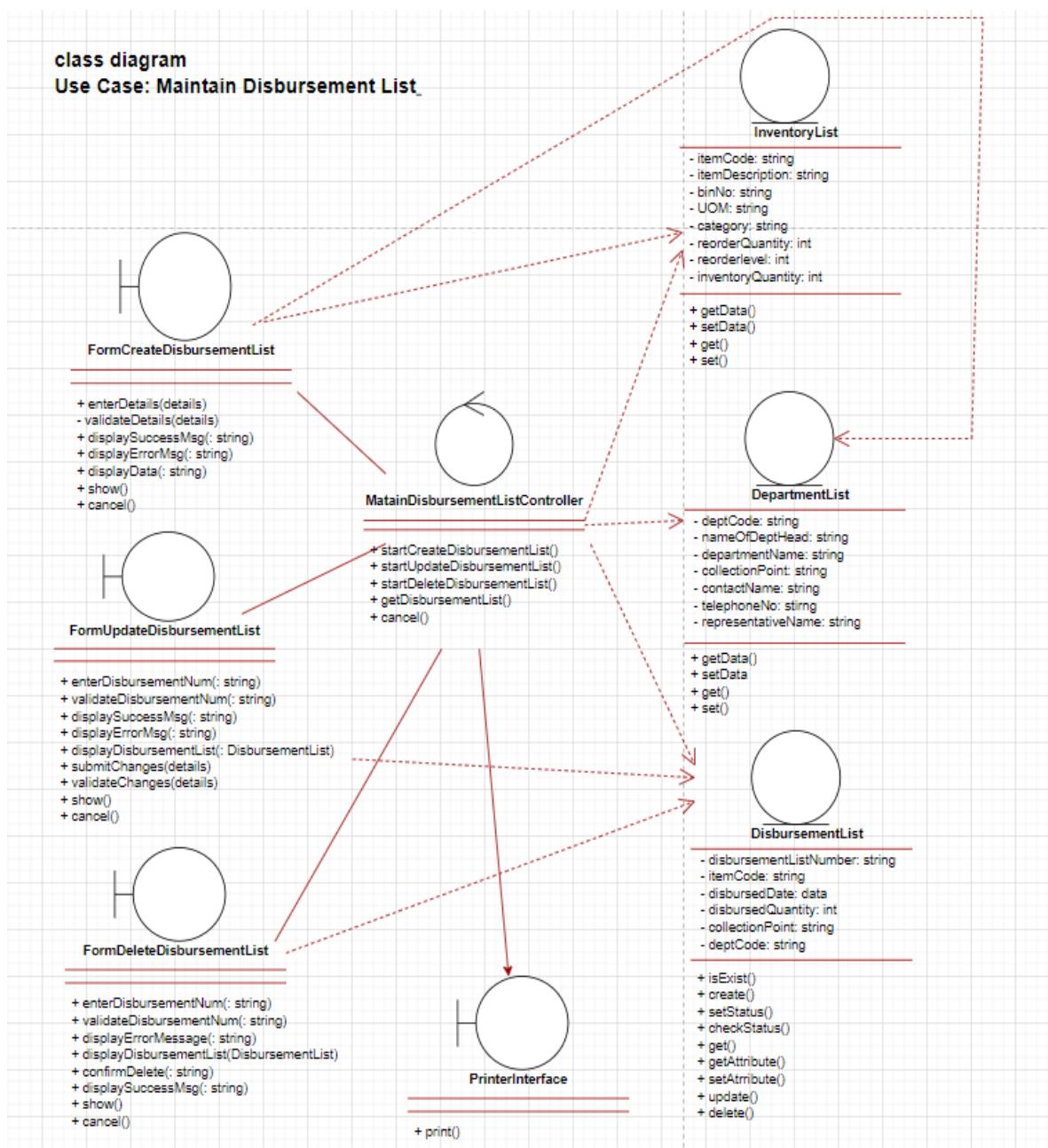


4. Use Case: Maintain Disbursement List (Subflow: Print for Delivering)



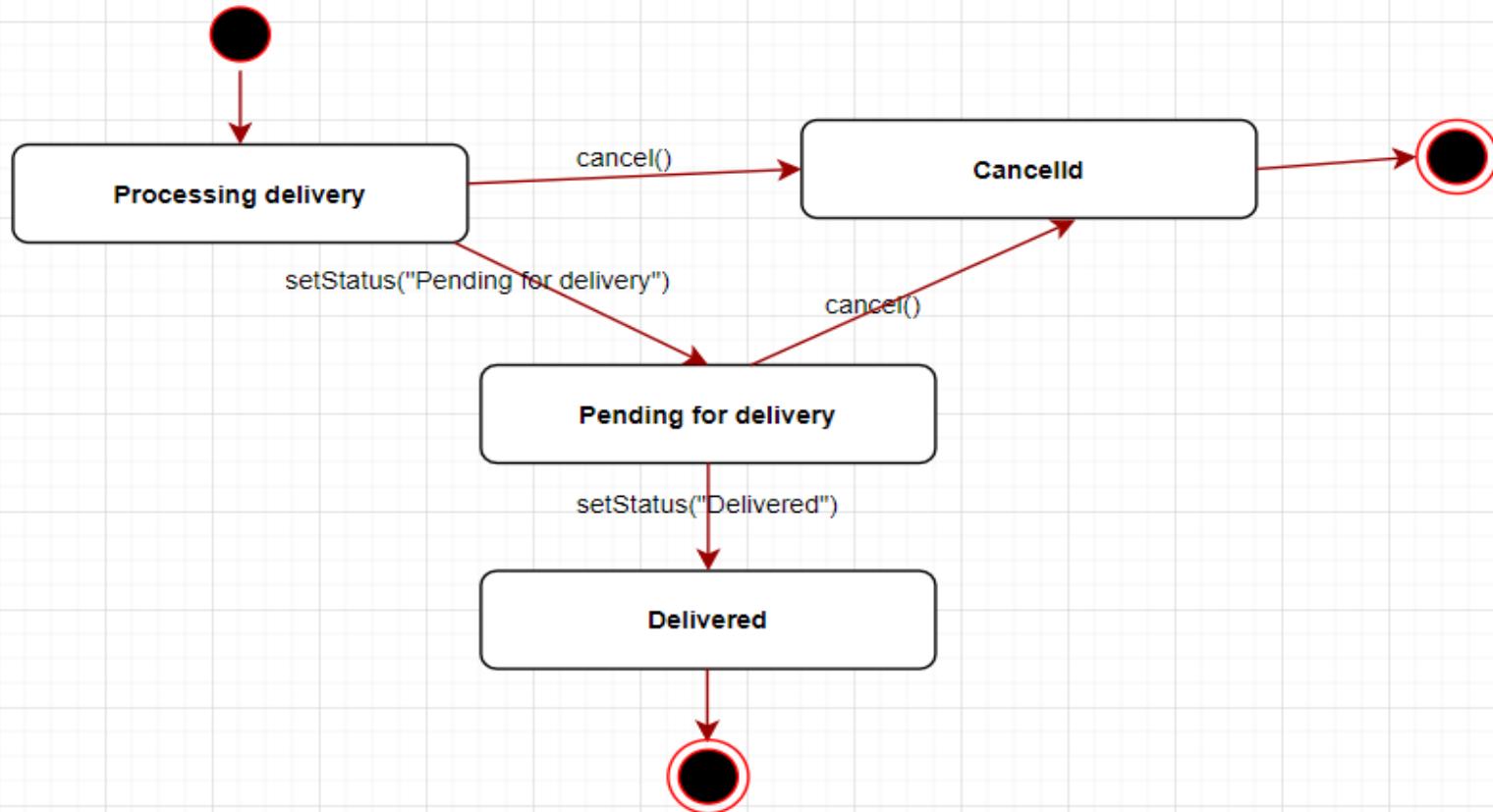
5. Use Case: Maintain Disbursement List (Subflow: Print for Testing)





Statechart Diagram

Use Case: Maintain Disbursement List



Use Case: Maintain Purchase Order

Store clerk will fill up the purchase order Form depending on the stock level of the items every two weeks.



Use case Description

Maintain Purchase Order (Create)

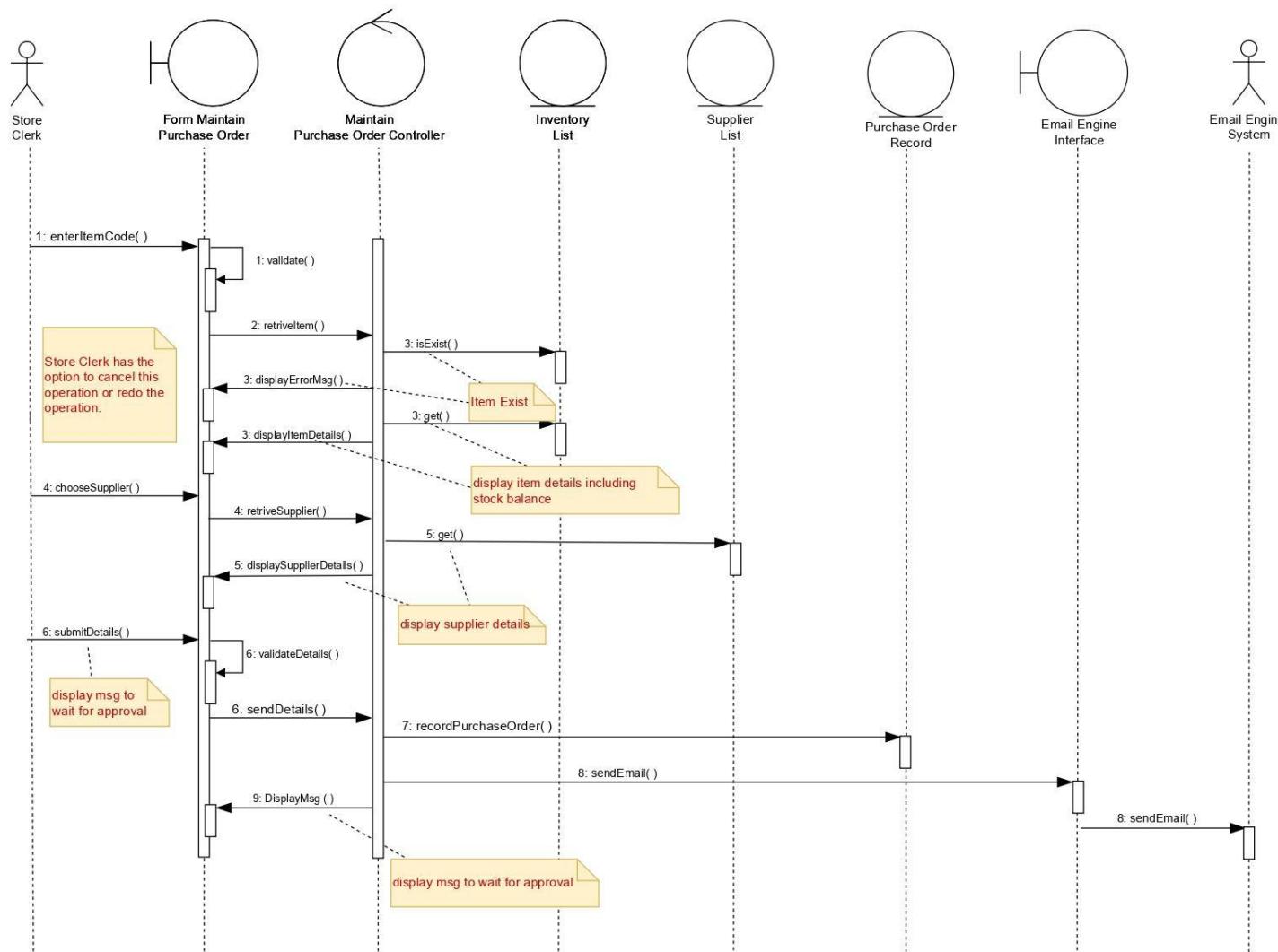
Flow of Events

This usecase is initiated when store clerk start to create purchase order.

1. Store Clerk key in item code to validate.
2. System retrieves and validate item code from inventory list.
3. System retrieves related items details and stock balance from inventory list and display.
4. Store Clerk choose Supplier from Supplier details.
5. Systems retrieves and display Supplplier Details from Supplier Record.
6. Store Clerk submit details of purchase order.
7. System records Purchase order details in Purchase order Record.
8. System infroms the Supervisor about Purchase order details through email.
9. System infroms store clerk to wait for an approval from the supervisor.

Maintain Purchase Order (Create)

Sequence Diagram



Use case Description

Maintain Purchase Order (update)

Flow of Events

This use case is initiated when store clerk start to update after he/she create purchase order.

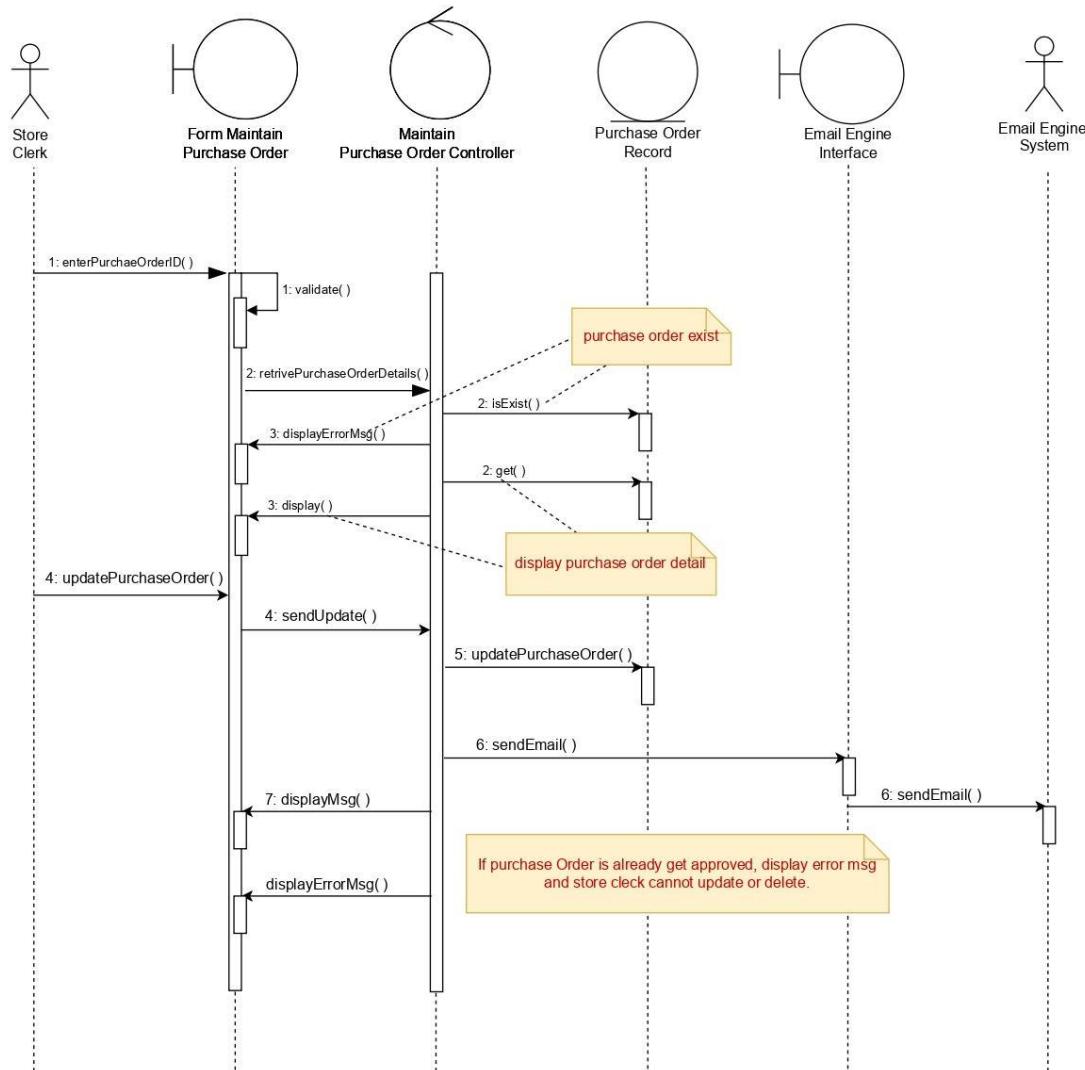
1. Store Clerk key in item purchase order ID to validate.
2. System retrieves and validate purchase order from purchase order record.
3. Systems display Purchase Order Details from purchase order Record.
4. Store Clerk submit update of purchase order.
5. System update Purchase order details in Purchase order Record.
6. System informs the Supervisor about updated Purchase order details through email.
7. System informs store clerk to wait for an approval from the supervisor.

Alternate Flow>>>>

If purchase order status is approved, store clerk cannot update or delete anything.

Maintain Purchase Order (Update)

Sequence Diagram



Use case Description

Maintain Purchase Order (delete)

Flow of Events

This use case is initiated when store clerk start to delete after he/she create purchase order.

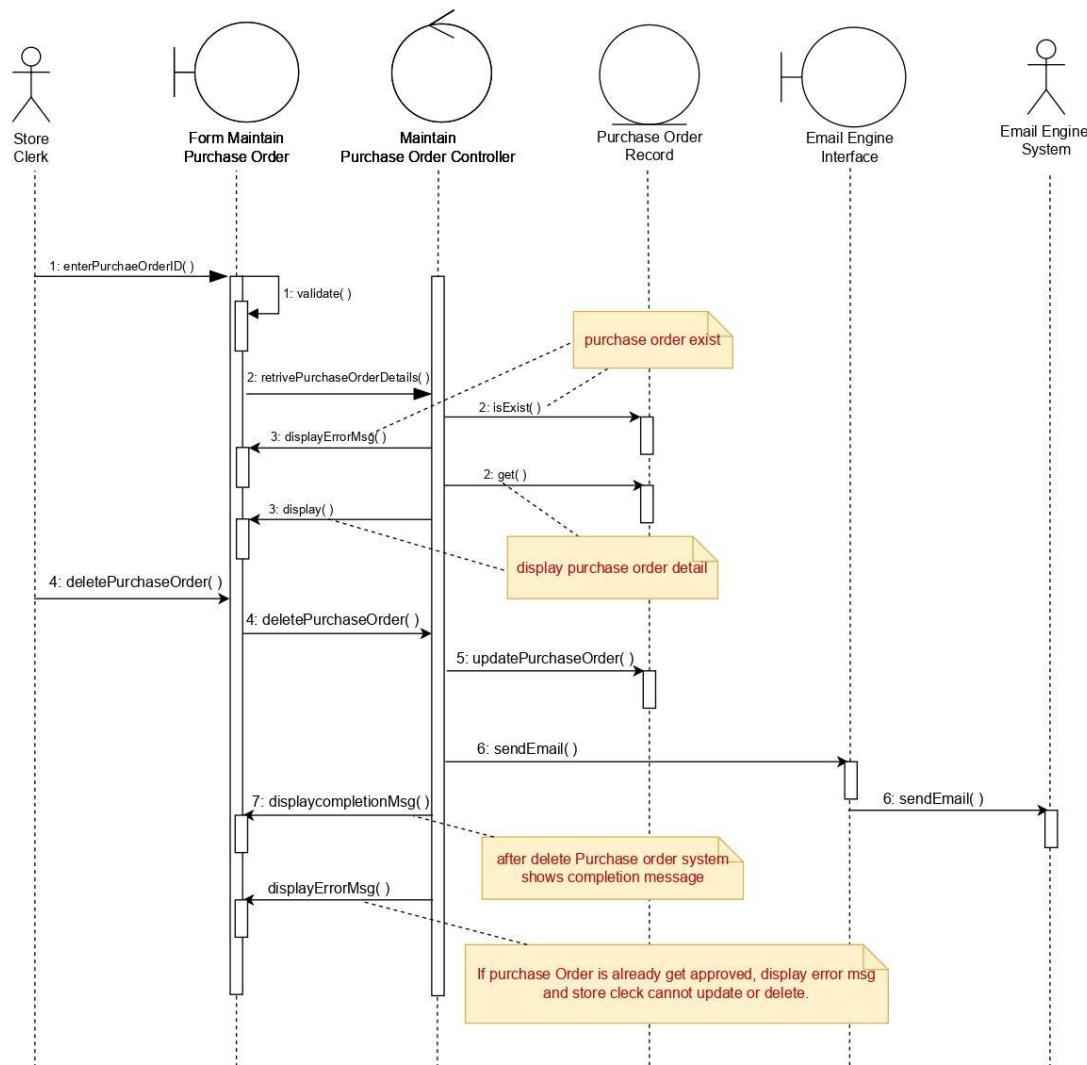
1. Store Clerk key in item purchase order ID to validate.
2. System retrieves and validate purchase order from purchase order record.
3. Systems display Purchase Order Details from purchase order Record.
4. Store Clerk delete of purchase order.
5. System update Purchase order details in Purchase order Record.
6. System informs the Supervisor about updated Purchase order details through email.
7. system shows completion message and use case is end here.

Alternate Flow>>>>

If purchase order status is already approved, store clerk cannot update or delete anything.

Maintain Purchase Order (Delete)

Sequence Diagram



Use case Description

Maintain Purchase Order (send purchase order to supplier)

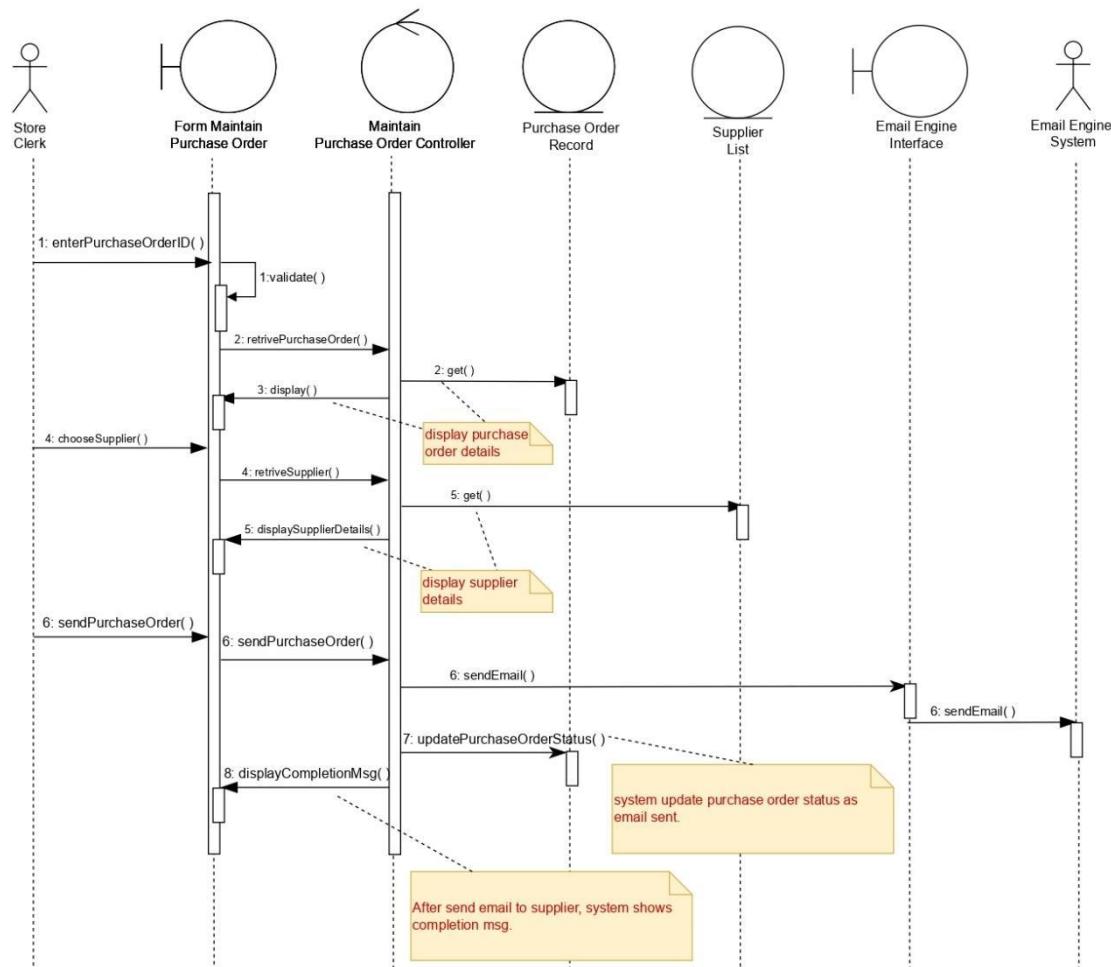
Flow of Events

This use case is initiated when store clerk receive approved purchase order.

1. Store Clerk key in purchase order ID.
2. System retrieves and validate purchase order ID from purchase order record.
3. System displays related purchase order details from purchase order record.
4. Store Clerk choose Supplier from Supplier details.
5. Systems retrieves and display Supplier Details from Supplier Record.
6. Store Clerk send details of purchase order to supplier through email.
7. System update Purchase order details in Purchase order Record.
8. System display completion message. use case end here.

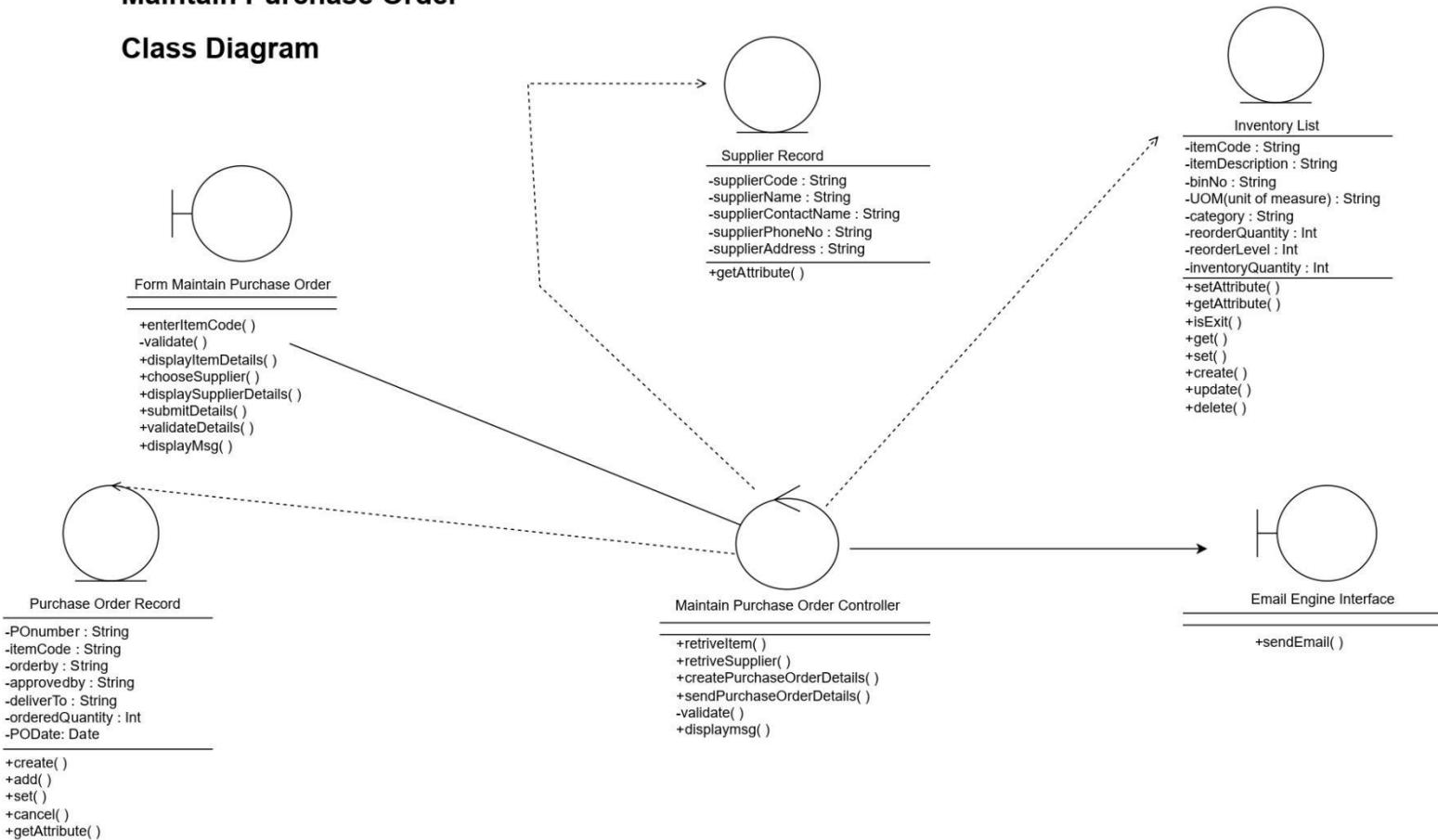
Maintain Purchase Order (send purchase order to supplier)

Sequence Diagram



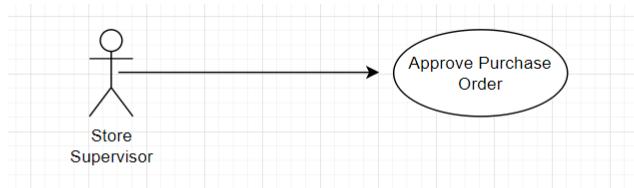
Maintain Purchase Order

Class Diagram



Use Case: Approve Purchase Order

Alert the Store Supervisor to approve the pending Purchase Order.



Approve Purchase Order

Approve Purchase Order

This usecase is initiated when Supervisor received notification about purchase order request.

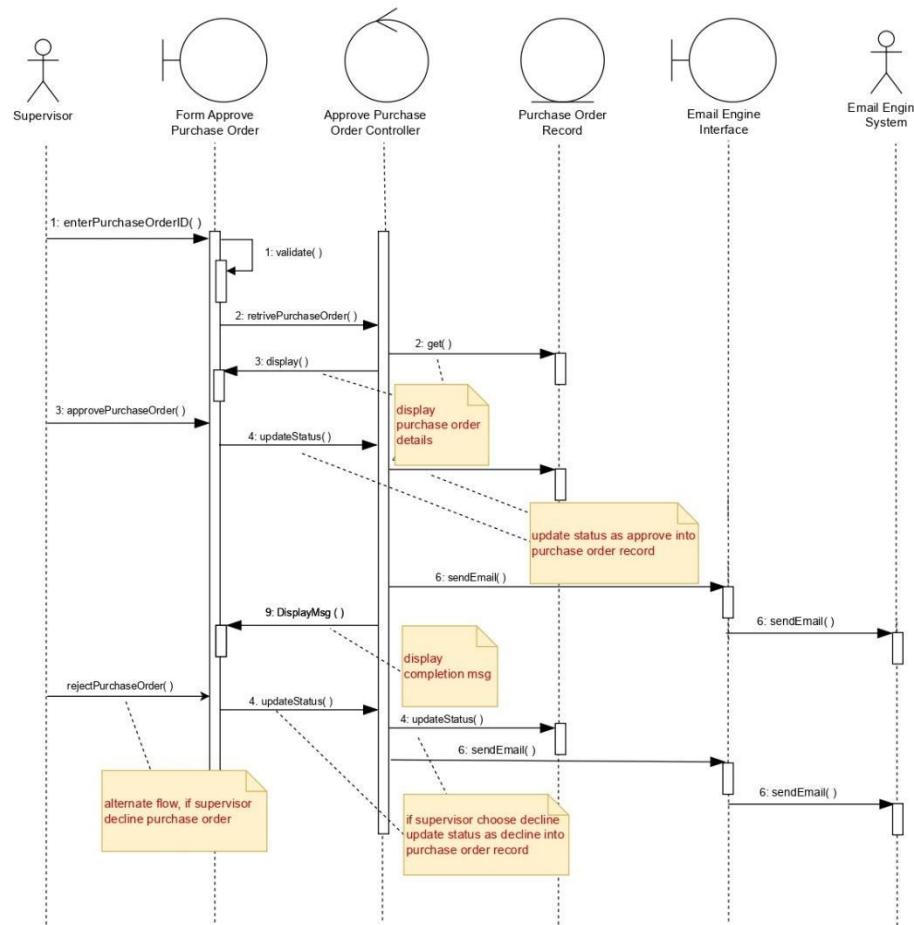
1. Supervisor key in purchase order ID.
2. System validate purchase order ID and retrieve details from the purchase order record list.
3. Supervisor select approve purchase order.
4. System update to purchase order record.
6. System sends approve notification to store clerk through email.

Alternate Flow>>

If supervisor select reject purchase order, system sends notification to store clerk through email. And update to the purchase order record.

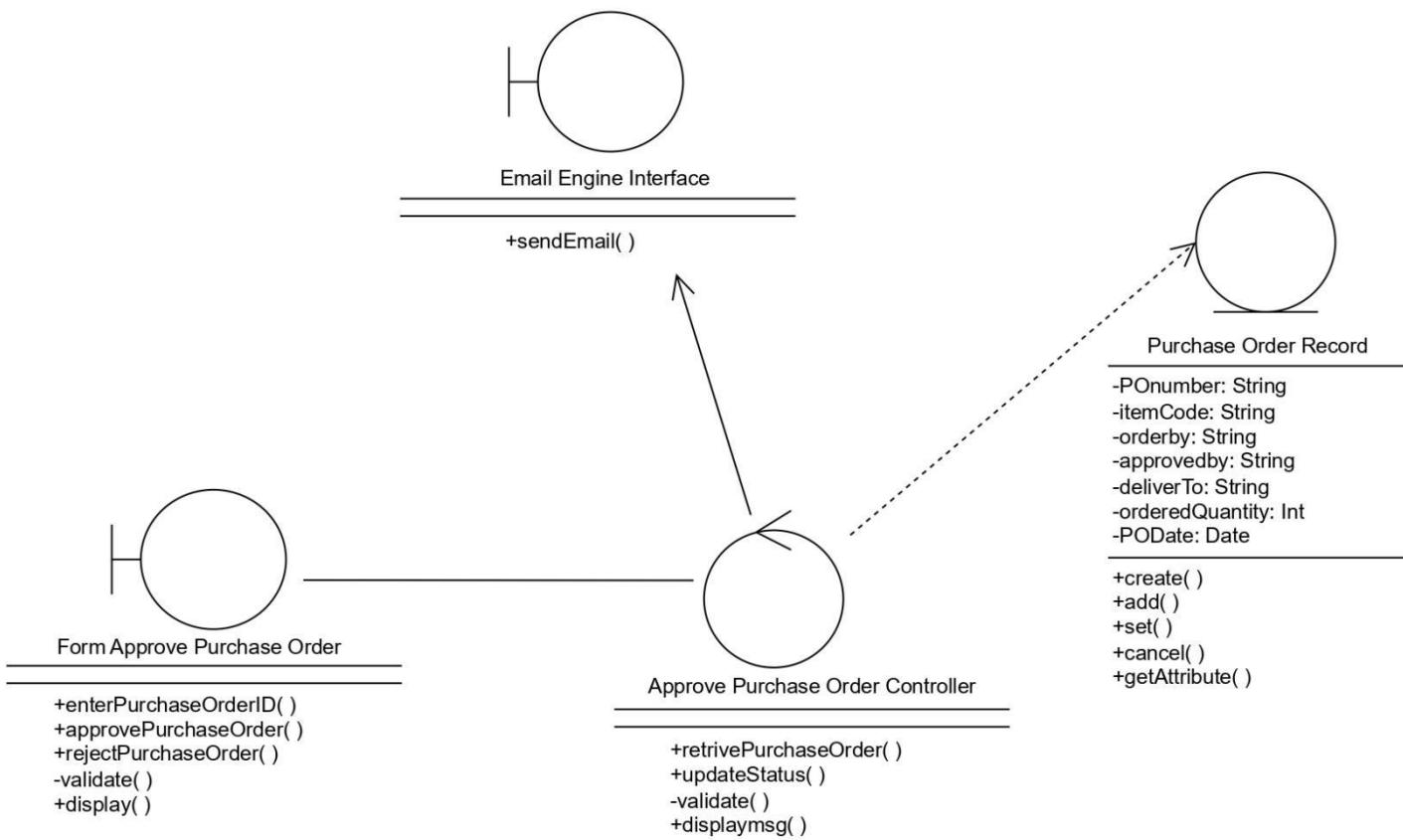
Approve Purchase Order

Sequence Diagram



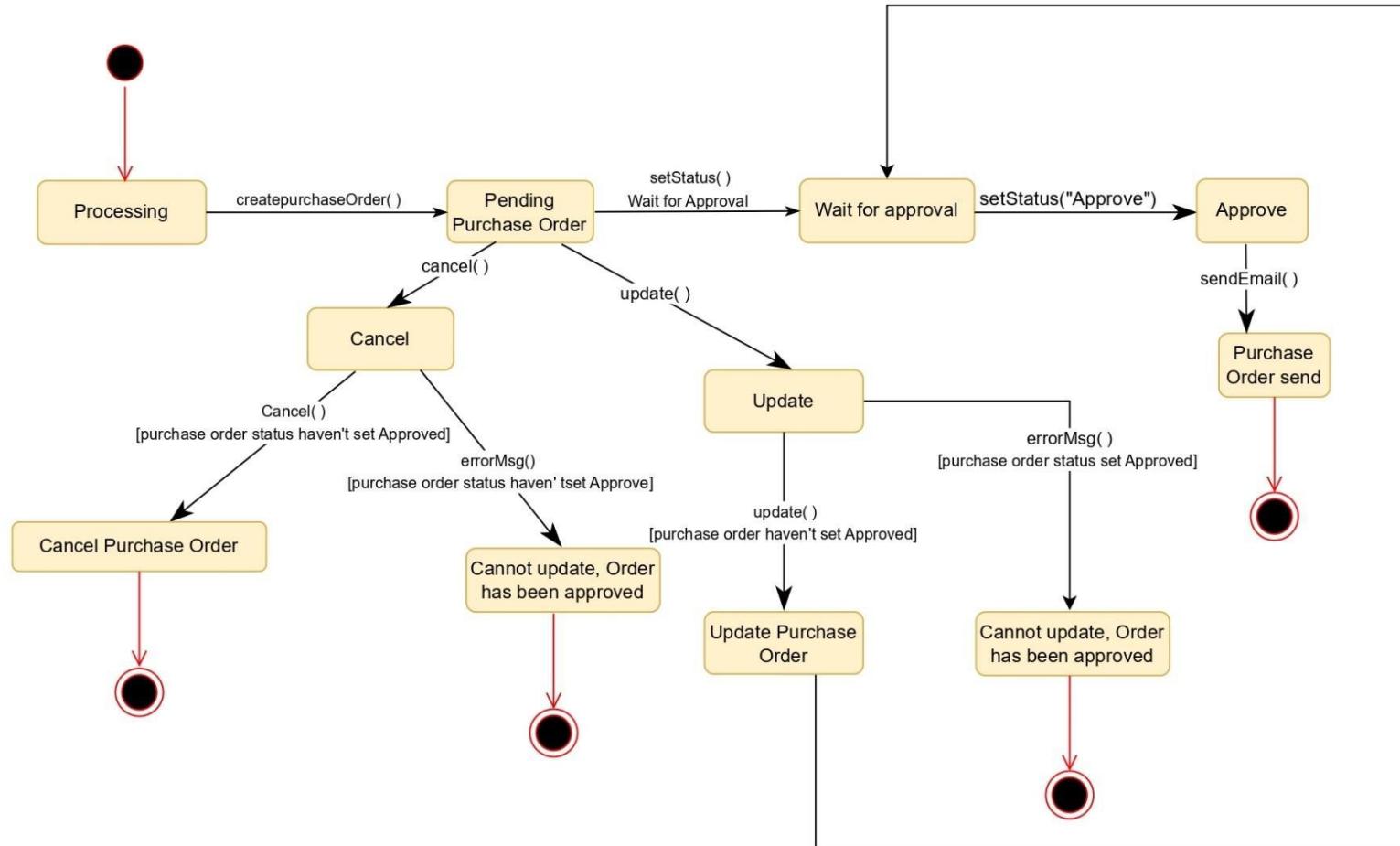
Approve Purchase Order

Class Diagram



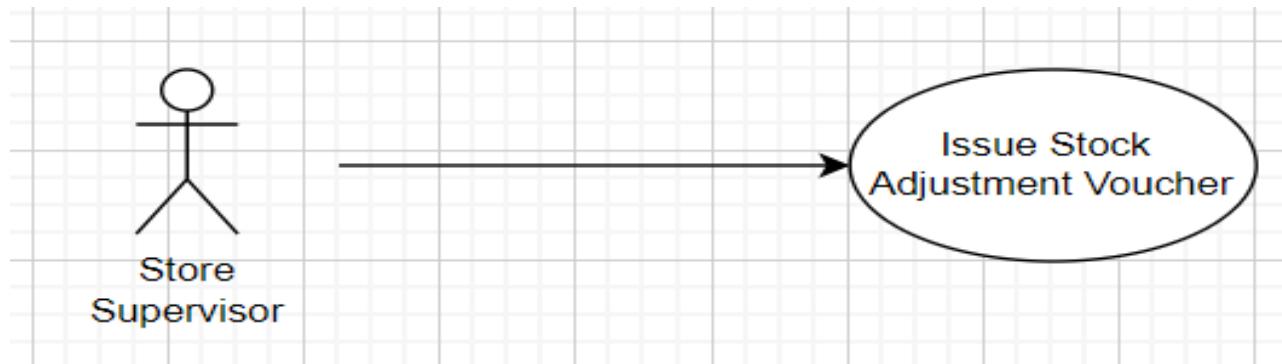
Purchase Order

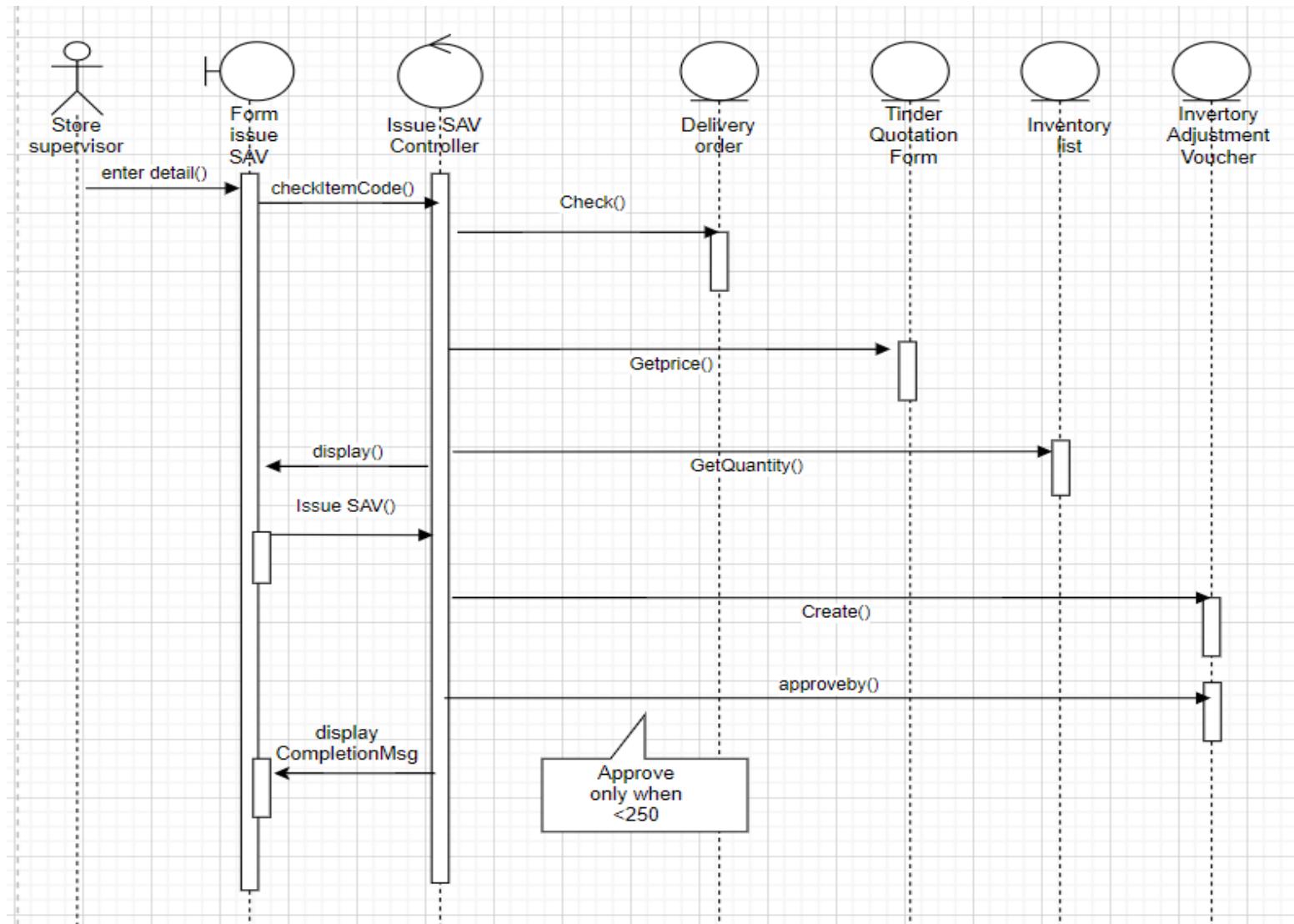
Statechart Diagram



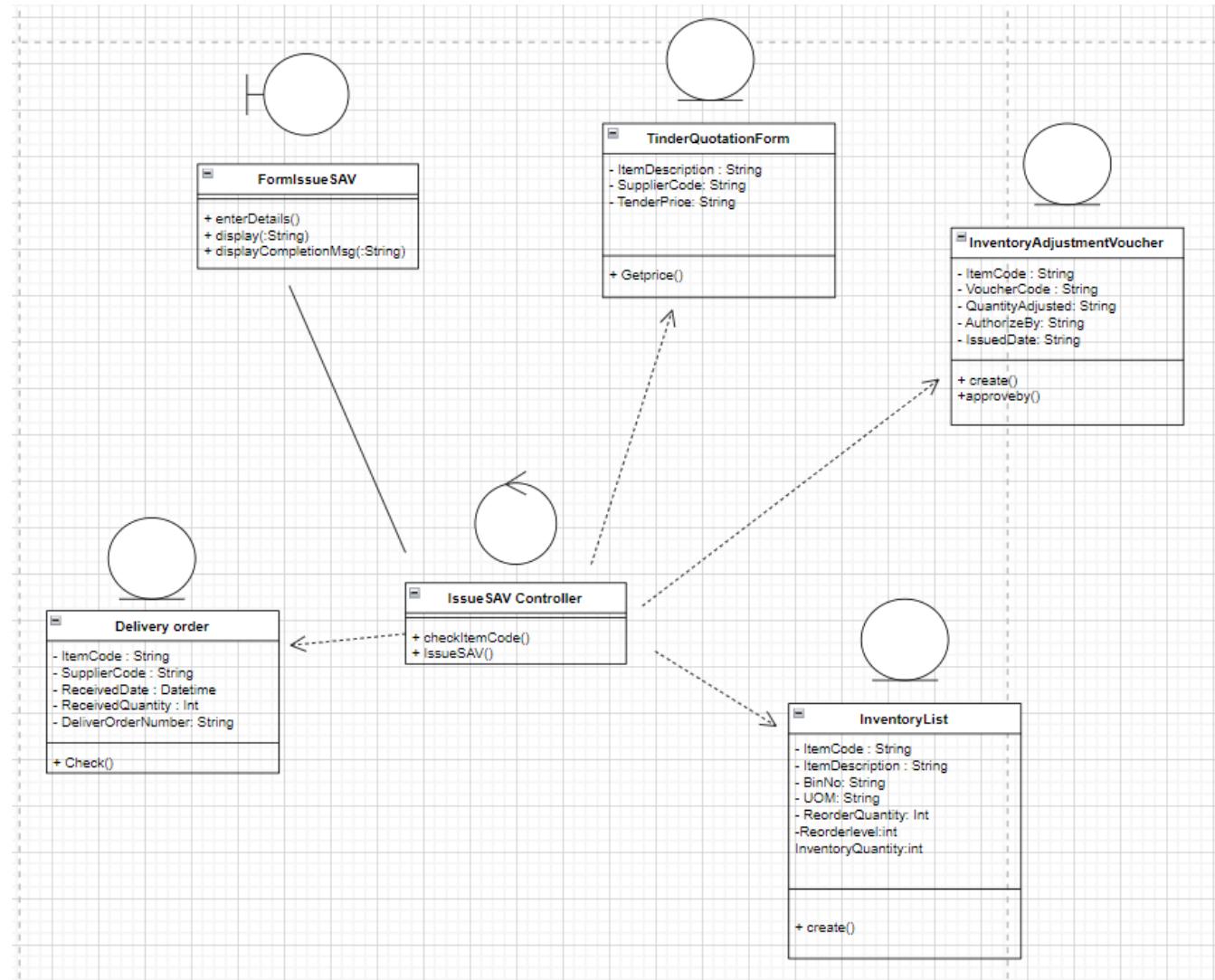
Use Case: Issue Stock Adjustment Voucher

The store supervisor will issue a stock adjustment voucher for anything under \$250 per item adjustment.



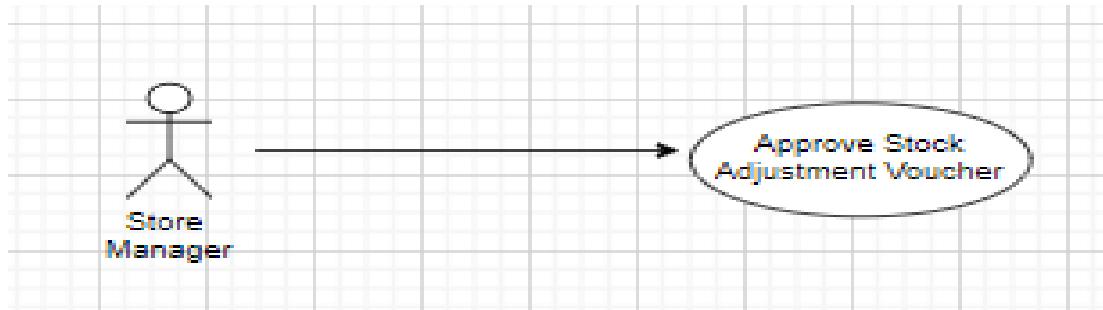
Issue Stock Adjustment Voucher : Sequence Diagram


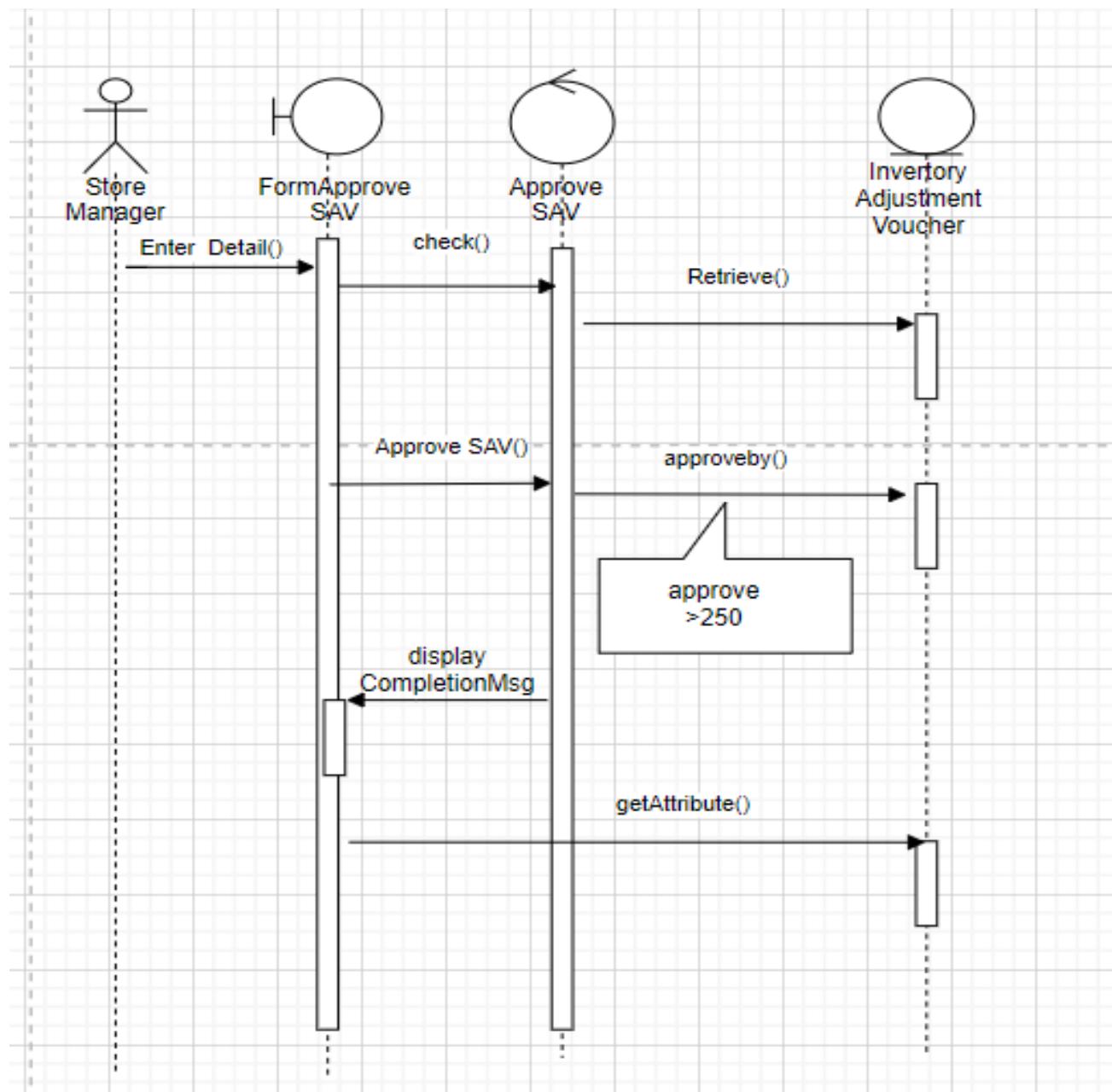
Issue Stock Adjustment Voucher : Class Diagram



Use Case: Approve Stock Adjustment Voucher

The store manager approves and issue the stock adjustment voucher that is above \$250.

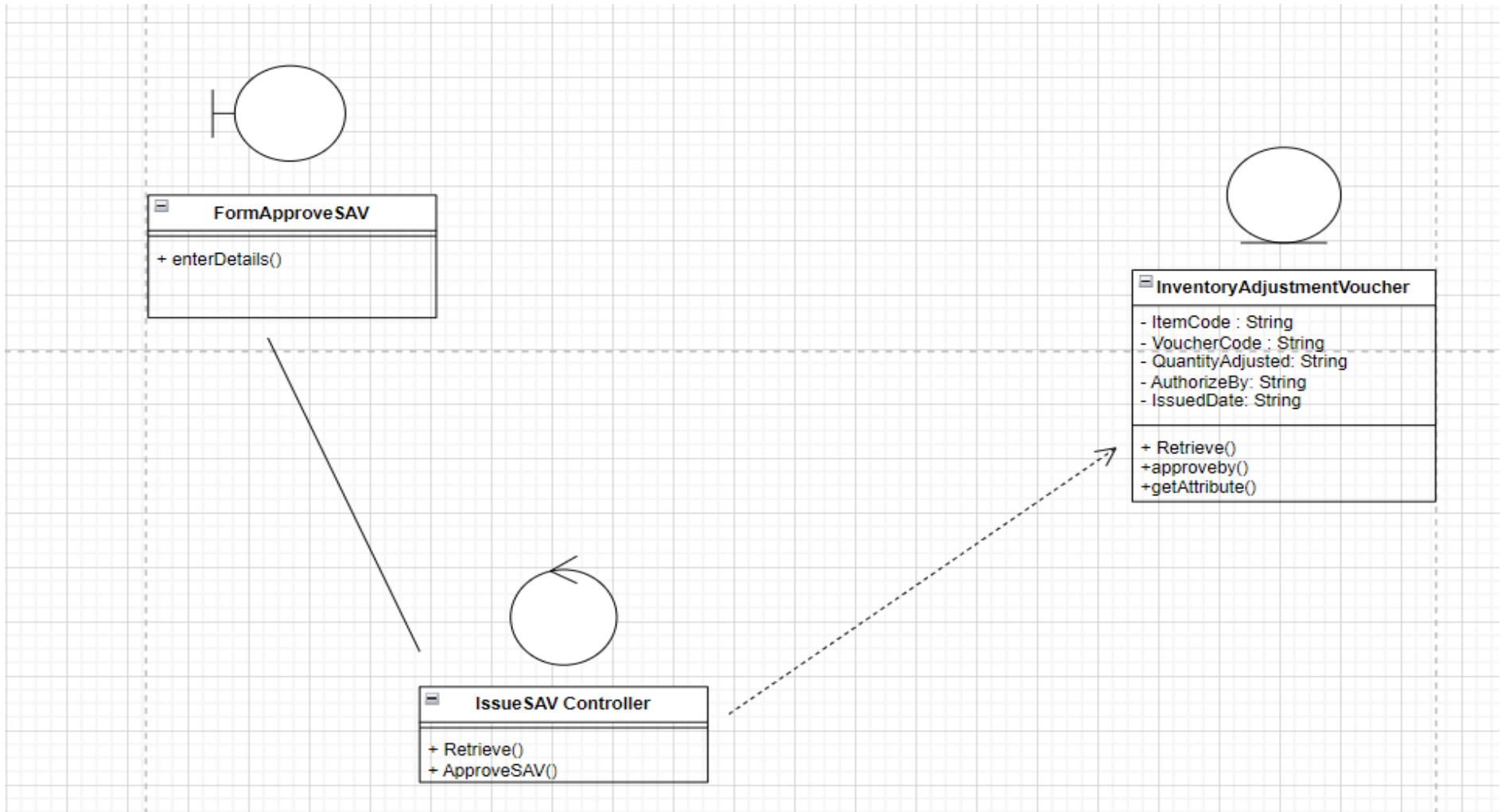




Approve Stock Adjustment Voucher

Sequence Diagram

Approve Stock Adjustment Voucher : Class Diagram

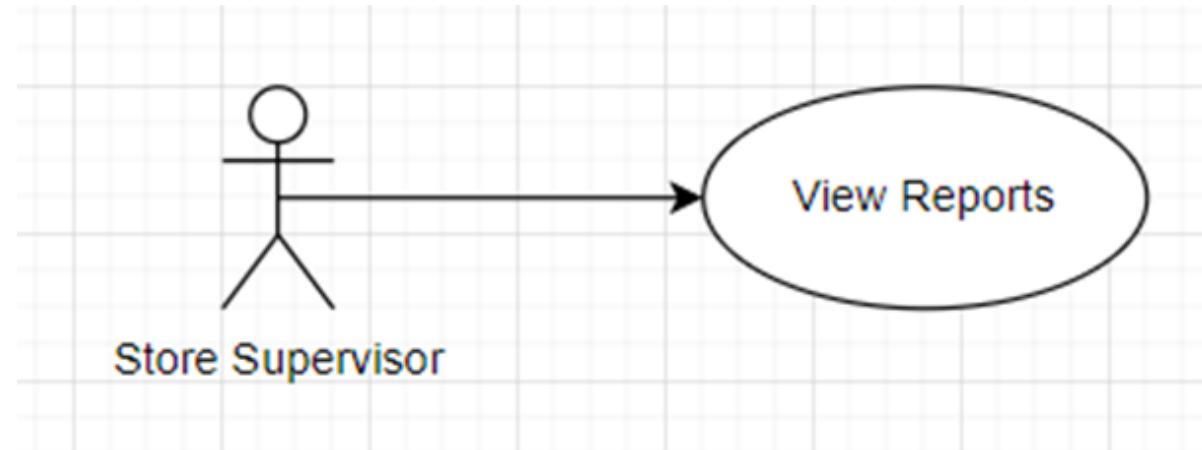


Use Case: View Reports

Sales report including the amount of stationery ordered grouped by categories and stationery requisition trend of each department.

Report to be generated with current and past 2 months.

Store manager and store supervisor are authorised to see reports.



Flows and Events:

The use case begins when the manager selects the “Report” option from the Main Menu.

Basic flow:

For retrieve total amount analysis report

1. System displays total amount analysis report in current month and past 2 months

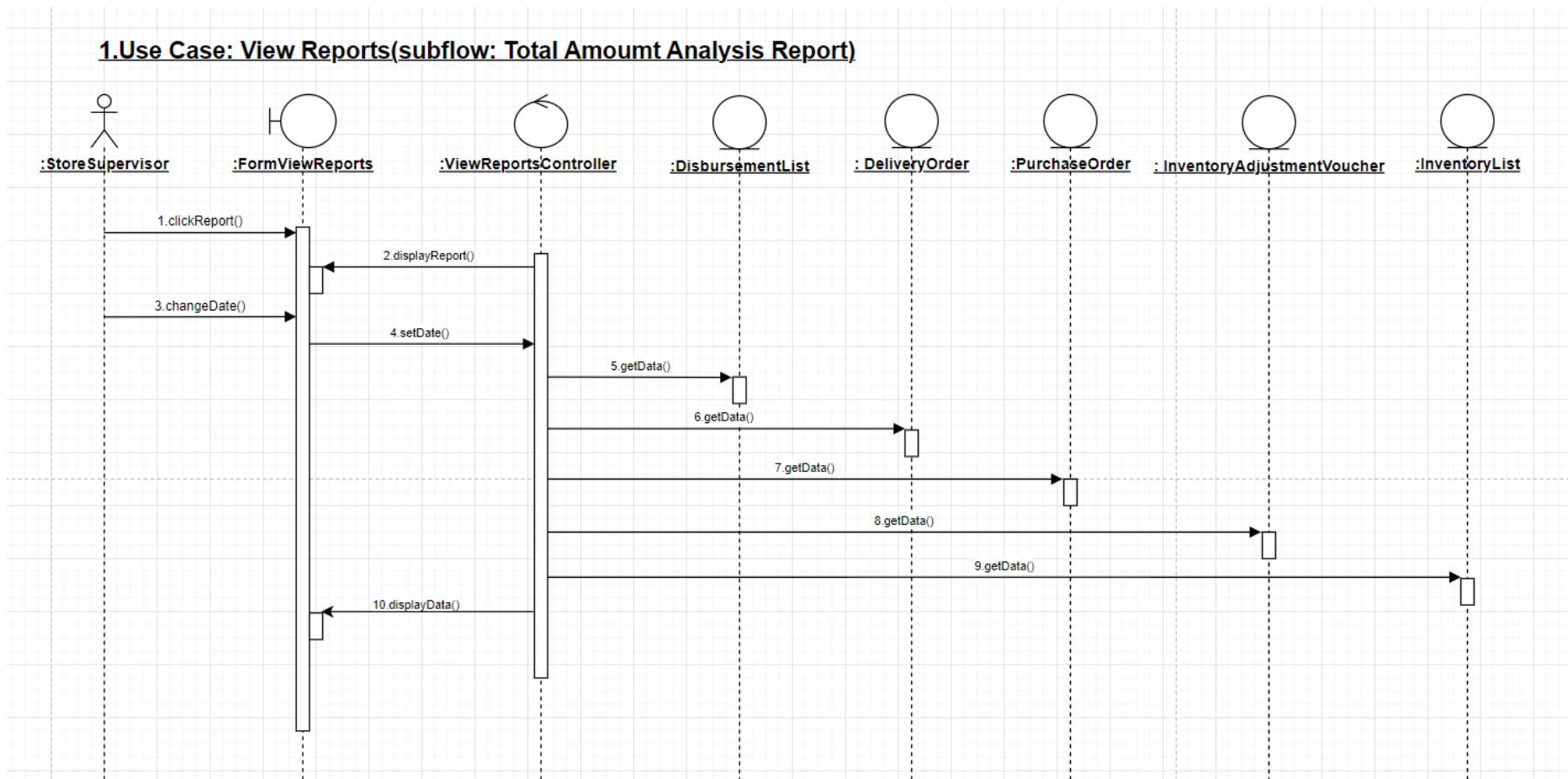
2. Manager can select any month within past 2 years to retrieve and compare data
3. System display 3 months data analysis

Alternate flows:**1.1 Retrieve stationery requisition trend report**

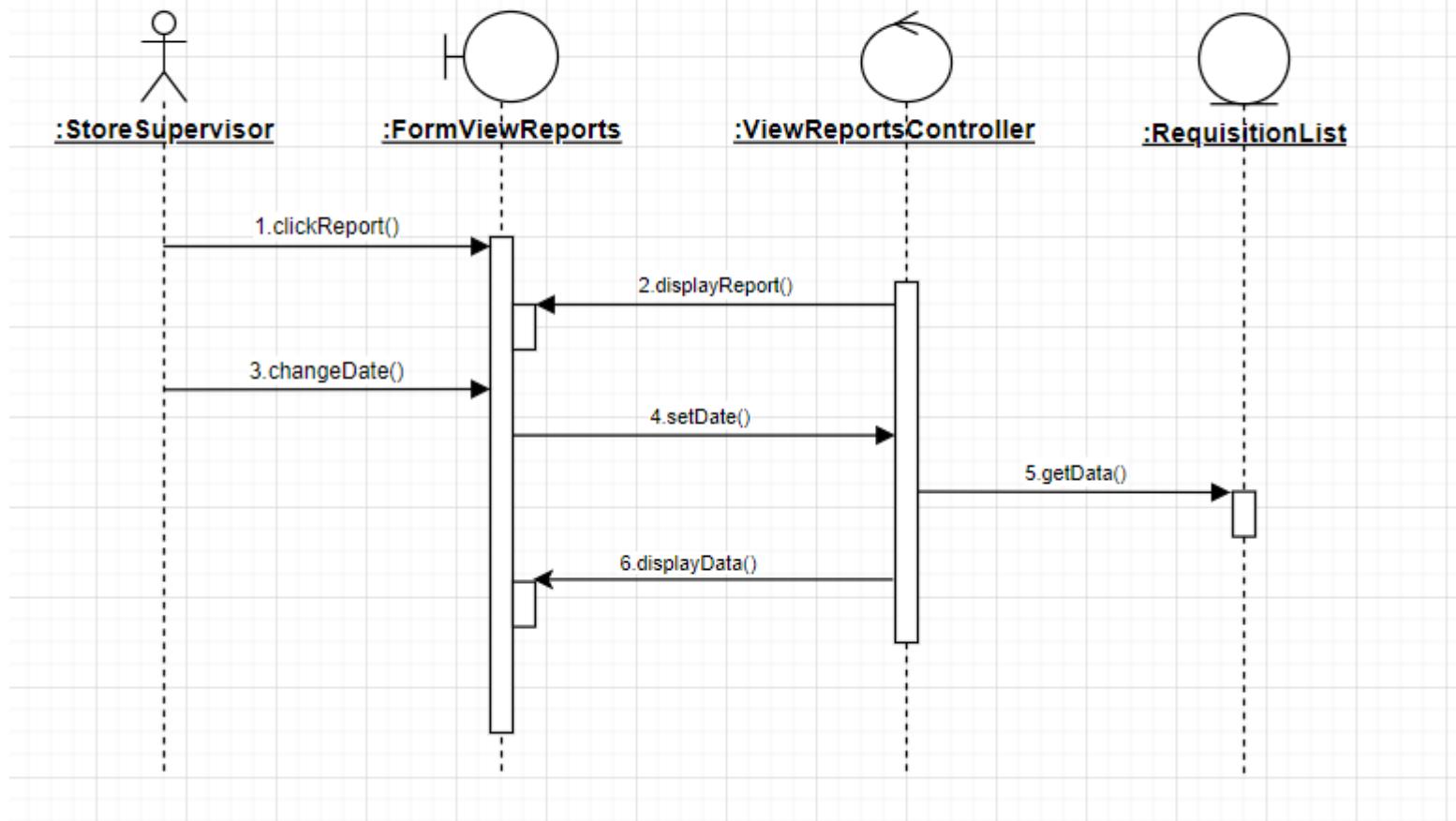
1. System displays stationery requisition trend of each department in current month and past 2 months
2. Manager can select any month within past 2 years to retrieve and compare data
3. System display 3 months data analysis

1.2 Retrieve stock item movements trend report

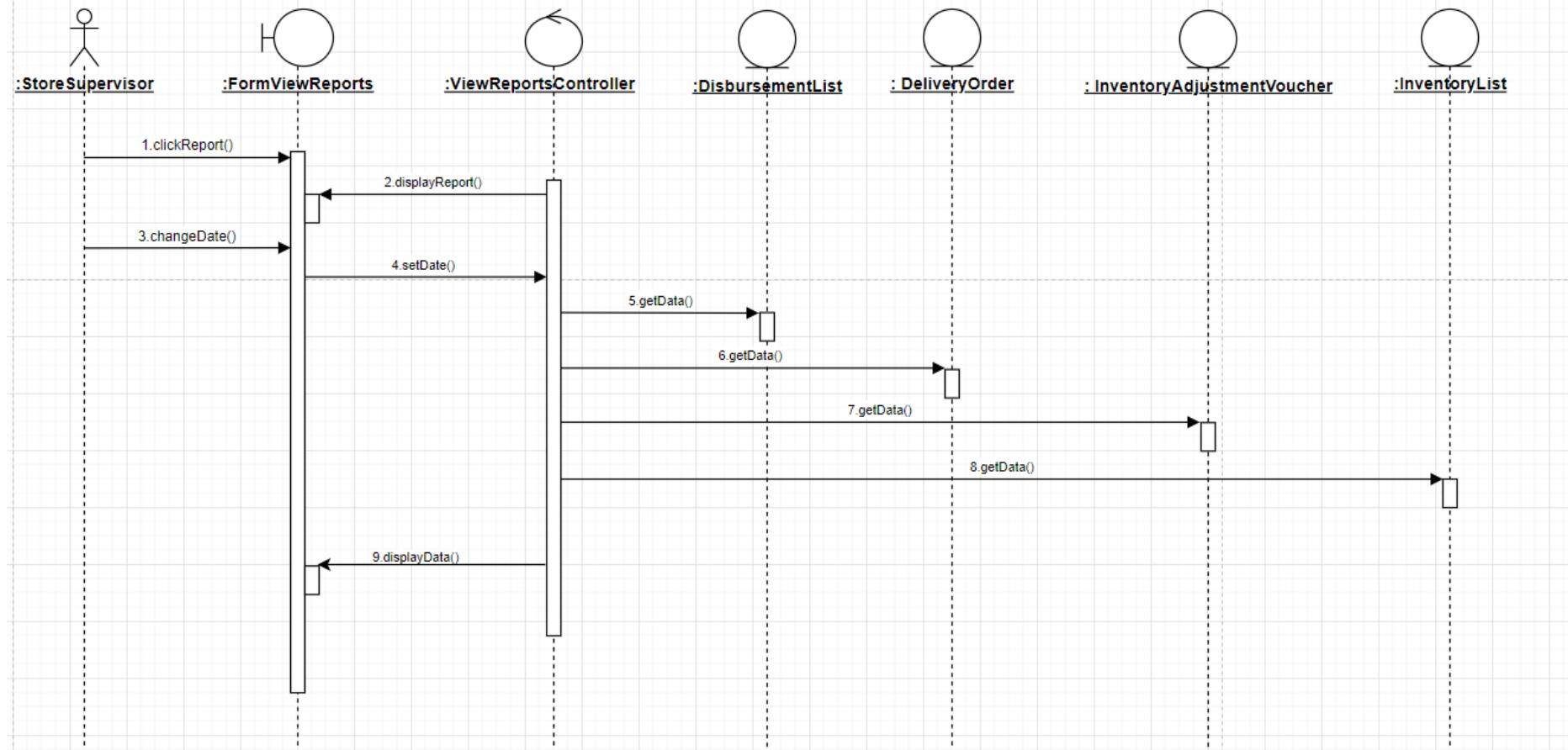
1. System displays stock item movements trend in current month and past 2 months
2. Manager can select any month within past 2 years to retrieve and compare data
3. System display 3 months data analysis

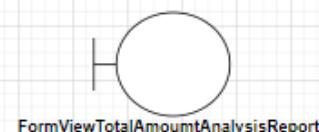
1. Use Case: View Reports(subflow: Total Amount Analysis Report)

2. Use Case: View Reports(subflow: Requisition Trend Report)

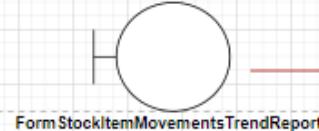


3. Use Case: View Reports(subflow: Stock Item Movement Trend Report)

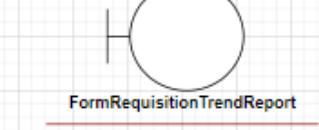


class diagram
Use Case: View Reports


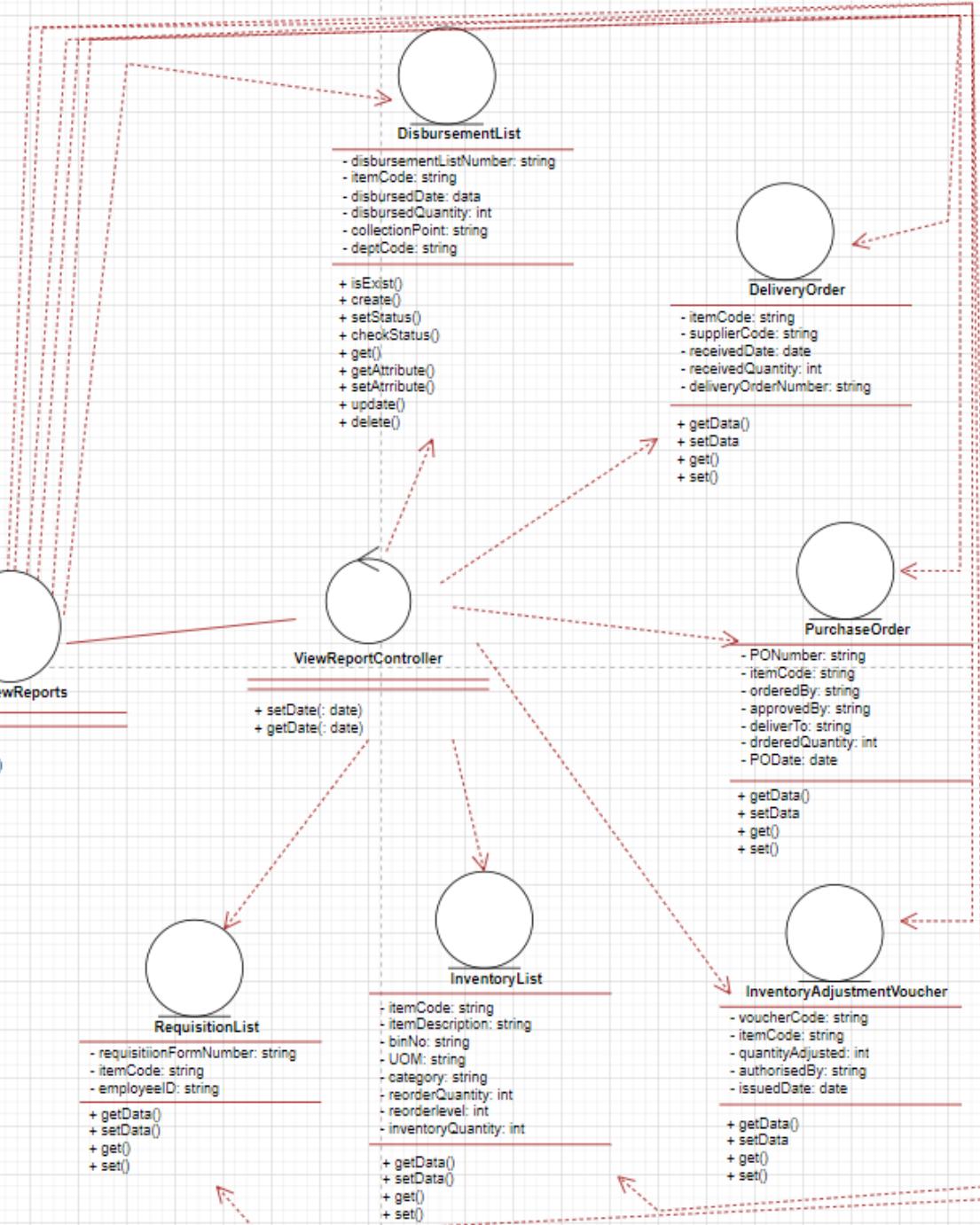
+ clickReport()
 + displayReport()
 + changeDate()
 + displayData()
 + show



+ clickReport()
 + displayReport()
 + changeDate()
 + displayData()
 + show



+ clickReport()
 + displayReport()
 + changeDate()
 + displayData()
 + show()



- disbursementListNumber: string
 - itemCode: string
 - disbursedDate: date
 - disbursedQuantity: int
 - collectionPoint: string
 - deptCode: string

+ isExist()
 + create()
 + setStatus()
 + checkStatus()
 + get()
 + getAttribute()
 + setAttribute()
 + update()
 + delete()

- itemCode: string
 - supplierCode: string
 - receivedDate: date
 - receivedQuantity: int
 - deliveryOrderNumber: string

+ getData()
 + setData()
 + get()
 + set()

- PONumber: string
 - itemCode: string
 - orderedBy: string
 - approvedBy: string
 - deliverTo: string
 - orderedQuantity: int
 - PODate: date

+ getData()
 + setData()
 + get()
 + set()

- itemCode: string
 - itemDescription: string
 - binNo: string
 - UOM: string
 - category: string
 - reorderQuantity: int
 - reorderLevel: int
 - inventoryQuantity: int

+ getData()
 + setData()
 + get()
 + set()

RequisitionList

- requisitionFormNumber: string
 - itemCode: string
 - employeeID: string

+ getData()
 + setData()
 + get()
 + set()

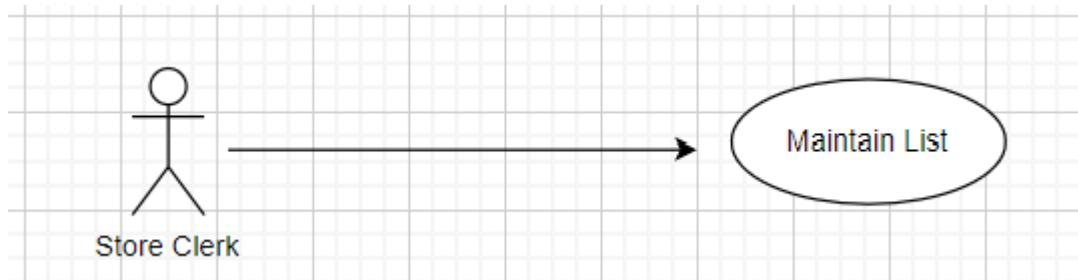
- voucherCode: string
 - itemCode: string
 - quantityAdjusted: int
 - authorisedBy: string
 - issuedDate: date

+ getData()
 + setData()
 + get()
 + set()

Use Case: Maintain List

(includes Supplier, Department, Price)

Annually (or as needed), Store Clerk can go to the system to update the various list in the database.



Basic Flow: (SubFlow: Create)

1. *StoreClerk select the type of forms and enters details of the form*
2. *System Validates the format of StoreClerk details*
 - a. *Supplier Forms (Link information to Dictionary)*
 - b. *Department Forms (Link information to Dictionary)*
 - c. *Price Forms (Link information to Dictionary)*
3. *If there are no existing records*
4. *System creates the information*
5. *System Displays Successful message*

Alternative Flow:

For Step 2, if data already exist, the information in the form,

An error message will be thrown

Basic Flow: (SubFlow: Update)

1. StoreClerk select the type of forms and enters details of of the form
2. System Validates the format of StoreClerk details
 - a. Supplier Forms (Link information to Dictionary)
 - b. Department Forms (Link information to Dictionary)
 - c. Price Forms (Link information to Dictionary)
3. System Retrieve the form with the information of the existing data
4. System Updates the information
5. System Displays Successful message

Alternative Flow:

For Step 2, if unable to locate the information in the form,

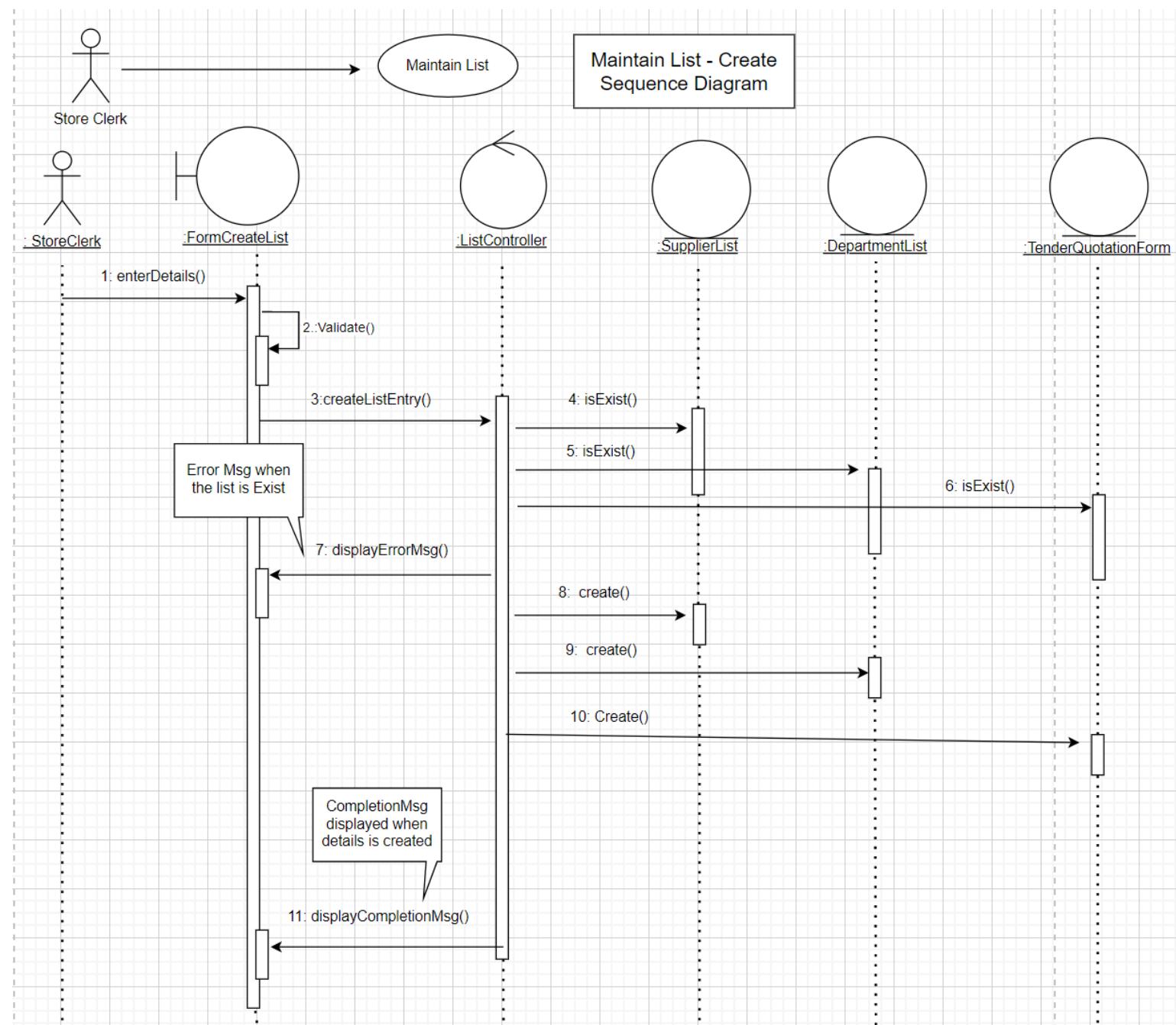
An error message will be thrown

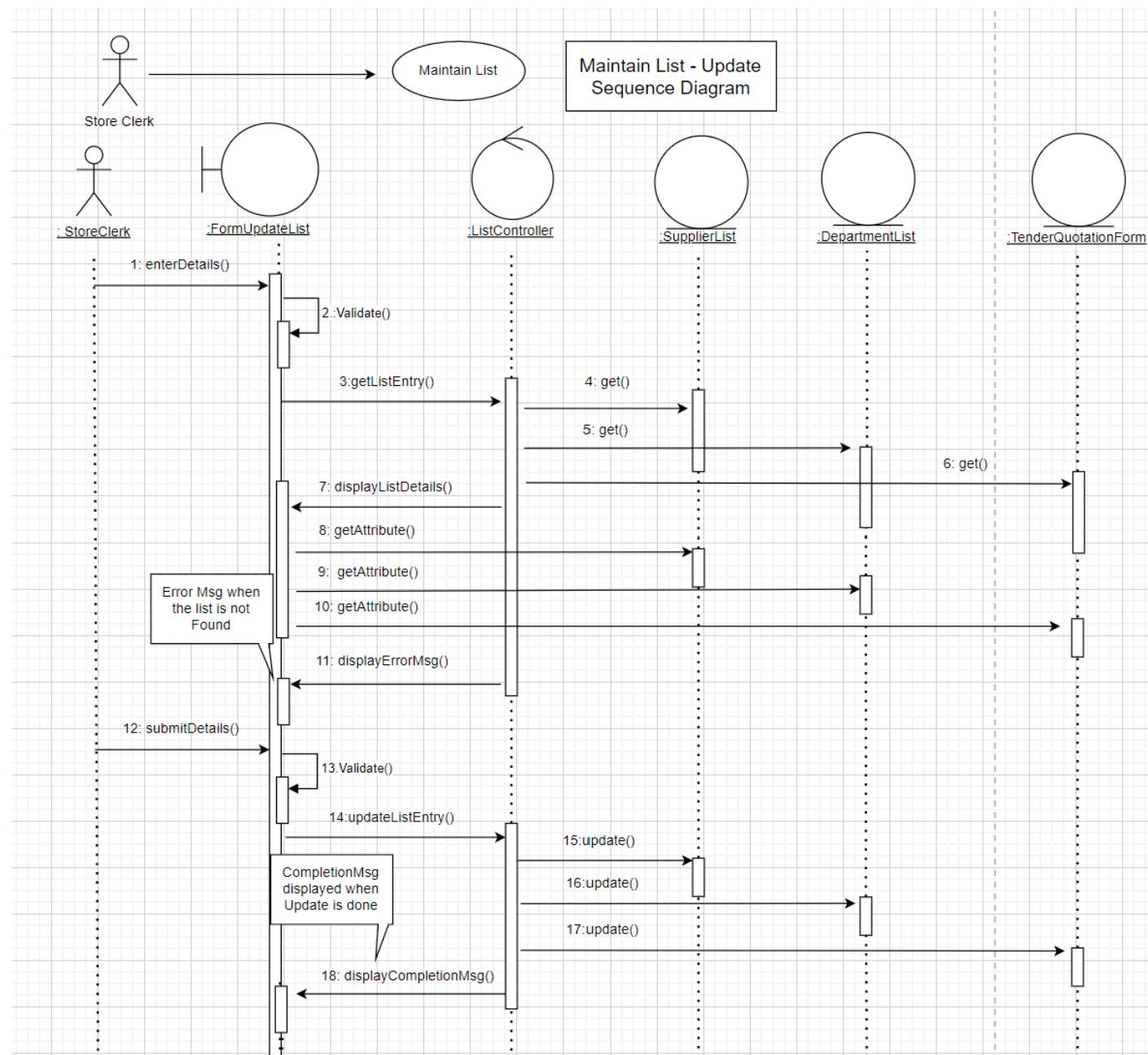
Basic Flow: (SubFlow: Delete)

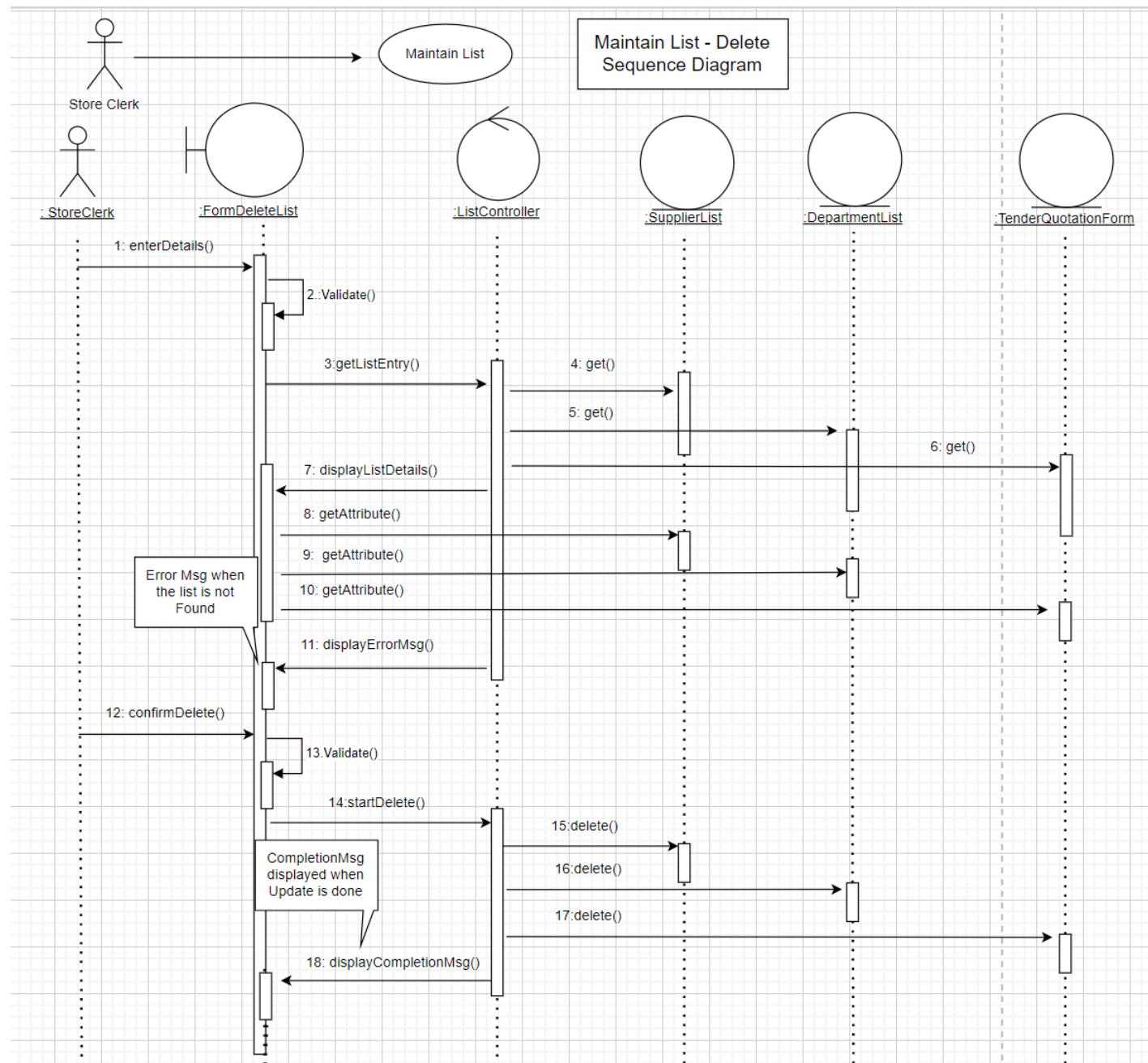
1. StoreClerk select the type of forms and enters details of of the form
2. System Validates the format of StoreClerk details
 - a. Supplier Forms (Link information to Dictionary)
 - b. Department Forms (Link information to Dictionary)
 - c. Price Forms (Link information to Dictionary)
3. System Retrieve the form with the information of the existing data
4. System Deletes the information
5. System Displays Successful message

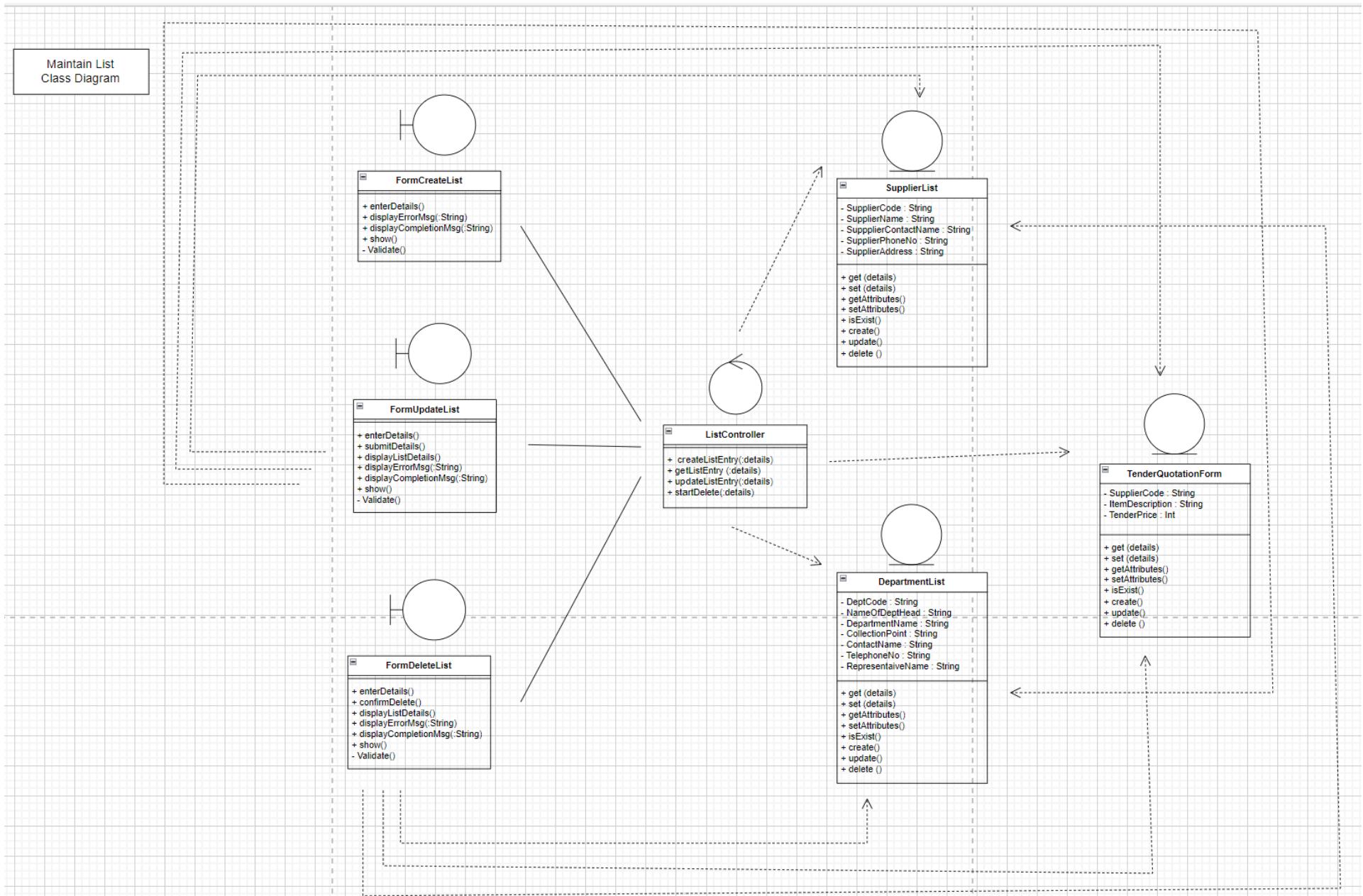
Alternative Flow:

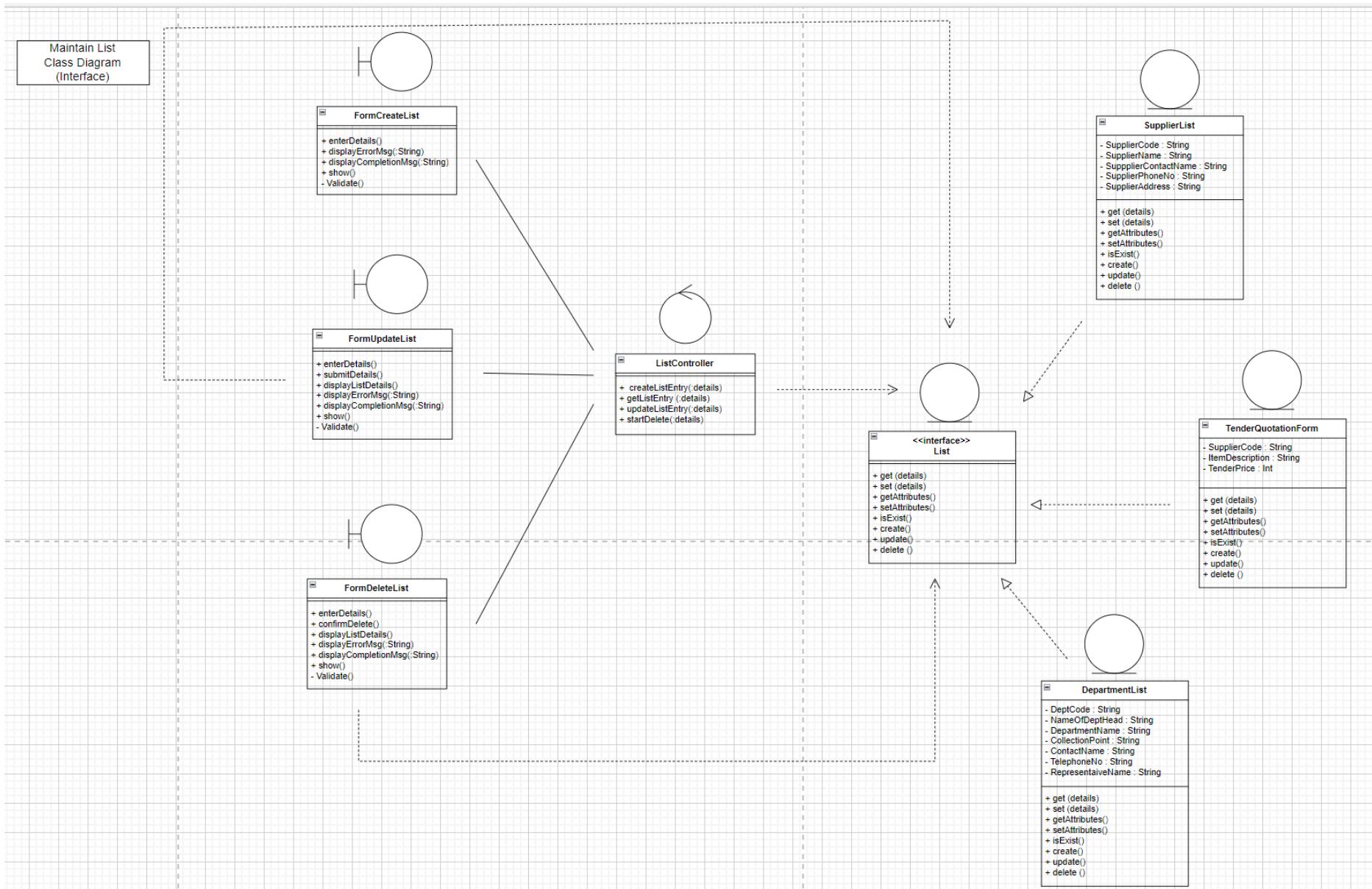
For Step 2, if unable to locate the information in the form, An error message will be thrown











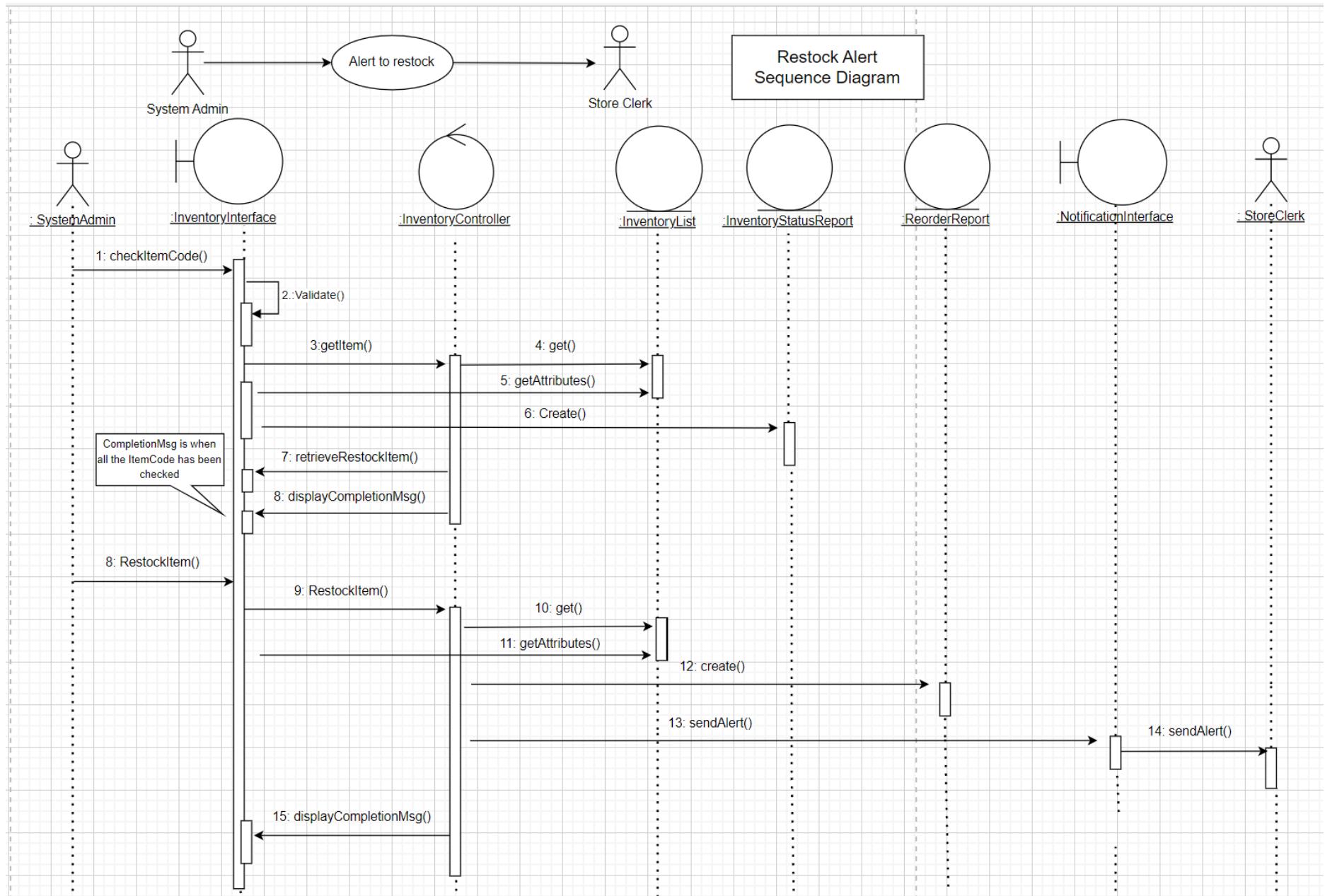
Use Case: Alert to Restock

Alert Store Clerk when stocks inventory is low and restocking is required.

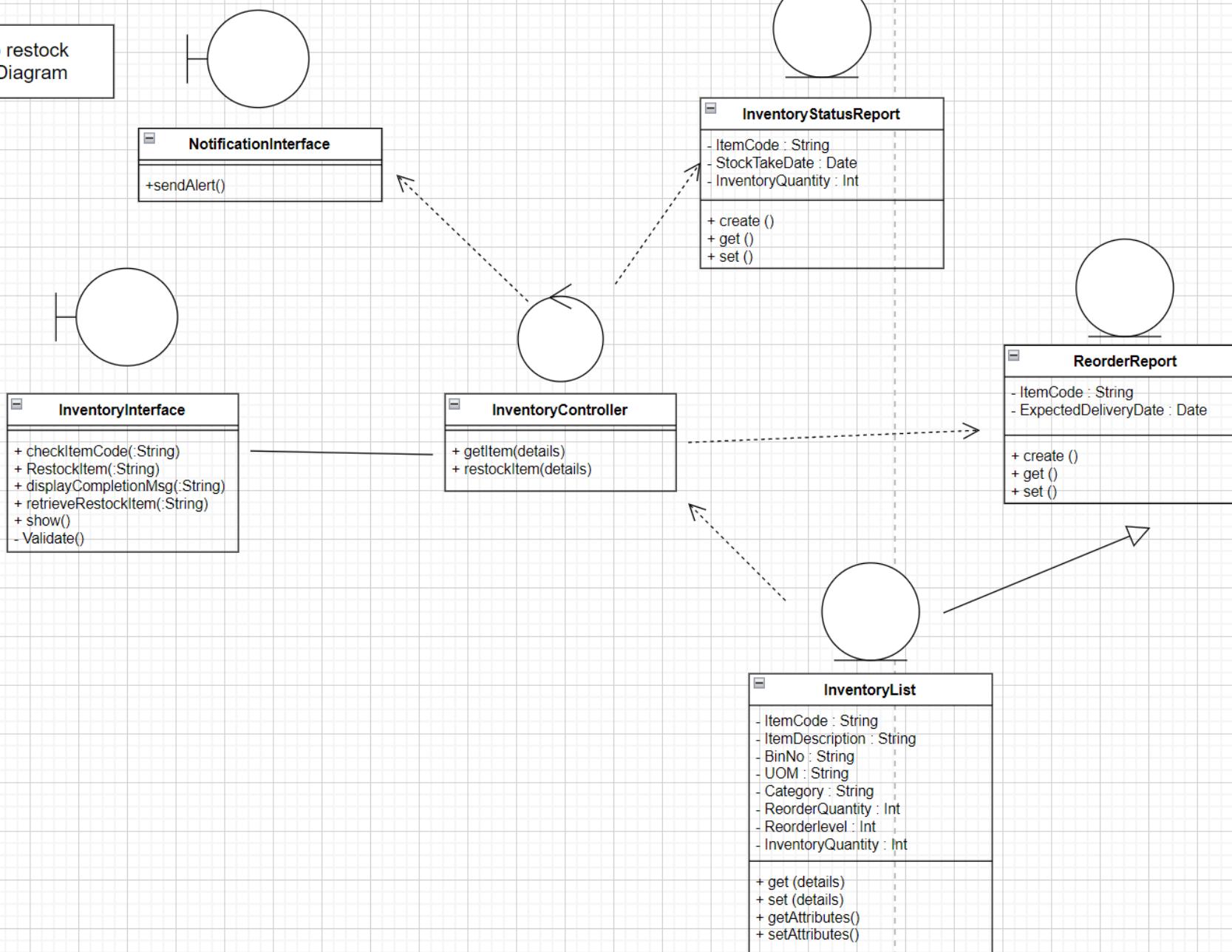


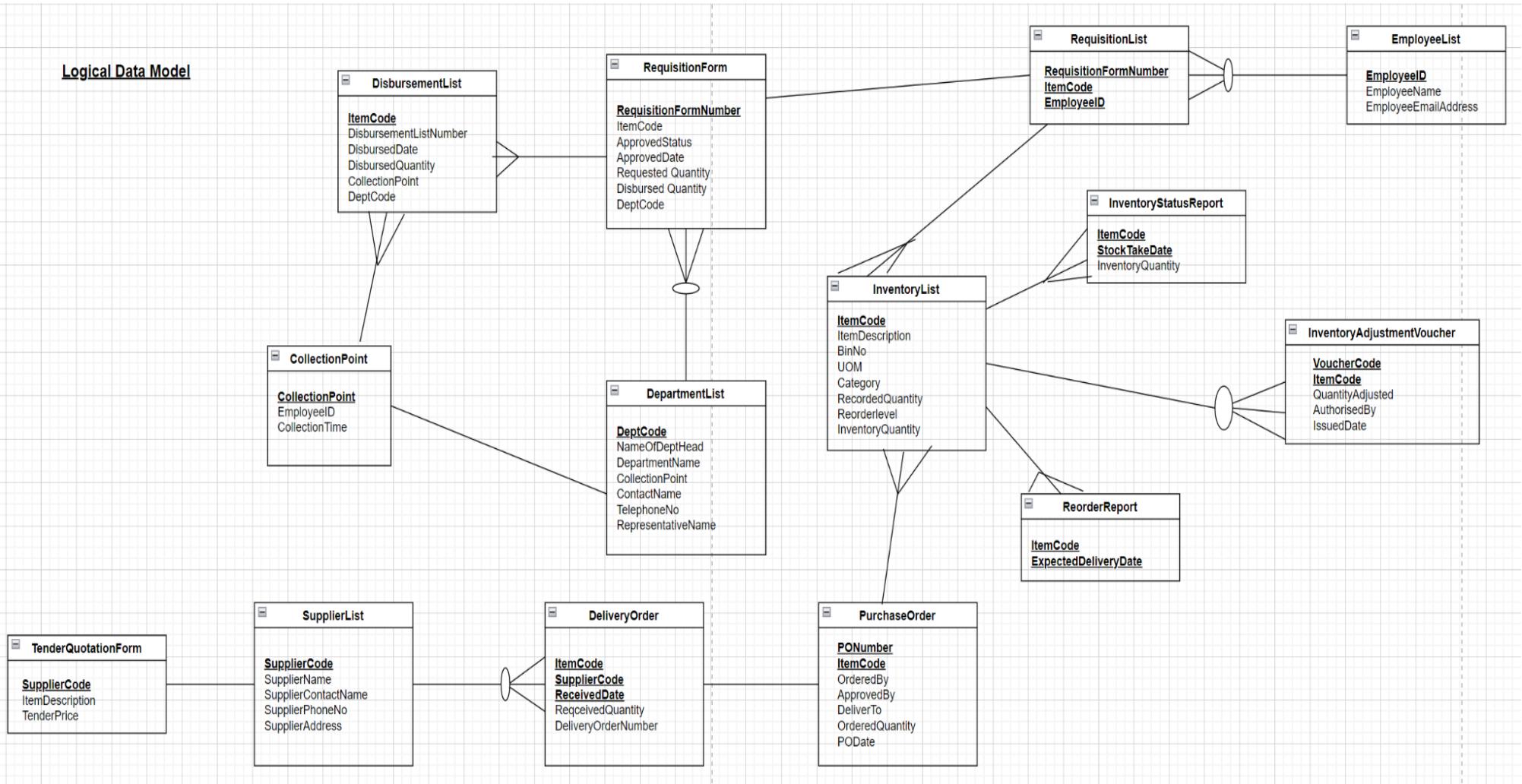
Basic Flow:

1. *SystemAdmin Enter each product ID*
2. *System retrieve product ID stocks status*
3. *SystemAdmin select and send notification of the product that require to reorder*
4. *System retrieves the List of items*
5. *System create a Reorder Record*
6. *System send data to "Notification"*
7. *"Notification" will be displayed on Store Clerk screen*



Alert to restock
Class Diagram



Logical Data Model

Data Dictionary (Final Tables)

ENTITY	ATTRIBUTE NAME	Type	LEN	DEFINITION AND BUSINESS RULES
RequestForm	RequisitionFormNumber	string	15	Unique name for requisition form
	ItemCode	string	10	Unique name for item. Start with one alphabet and follow by 3 numbers.
	ApprovedStatus	string	50	Head Of Department to approved (if head of department not around), the person who sign need to input his/her name
	ApprovedDate	date	8	Format: DDMMYY Date when the Head of Department Approved the form
	RequestQuantity	int	5	Quantity needed
	DisbursedQuantity	int	5	Quantity provided by Store
	DeptCode	string	10	Department ID
ReorderReport	ItemCode	string	10	Unique name for item. Start with one alphabet and follow by 3 numbers.
	ExpectedDeliveryDate	date	8	Format: DDMMYY Estimated Time of Receipt of the Reorder Item
InventoryAdjustmentVoucher	VoucherCode	string	10	Unique code for the Adjustment Voucher
	ItemCode	string	10	Unique name for item. Start with one alphabet and follow by 3 numbers.

	QuantityAdjusted	int	5	Quantity Discrepancies (can be positive or negative value)
	AuthorisedBy	string	50	Supervisor will sign off if the total discrepancies is less than or equal SGD 250.00 Manager will sign off if the total discrepancies is more than SGD 250.00
	IssuedDate	date	8	Format DDMMYY Date of issue of the Adjustment Voucher
DisbursementList	ItemCode	string	5	Unique name for item. Start with one alphabet and follow by 3 numbers.
	DisbursementListNumber	string	10	Unique name for each Disbursement List
	DisbursedDate	date	8	Format: DDMMYY Date of creation of the Disbursement List
	DisbursedQuantity	int	5	Quantity to be delivered the Collection Point
	CollectionPoint	string	100	Name of Place where to collect item delivery
	DeptCode	string	5	Department ID
InventoryStatusReport	ItemCode	string	5	Unique name for item. Start with one alphabet and follow by 3 numbers.
	StockTakeDate	date	8	Format: DDMMYY Date which physical stocktake was done
	InventoryQuantity	int	5	Number of quantity in store
CollectionPoint(CS)	CollectionPoint	string	100	Name of Place where to collect item delivery

	EmployeeID	string	10	Employee ID
	CollectionTime	date	8	Collection time for store delivery
SupplierList	SupplierCode	string	15	Unique character for Supplier
	SupplierName	string	15	Supplier Company Name
	SupplierContactName	string	15	Supplier Contact Name
	SupplierPhoneNo	int	10	Supplier Phone Number
	SupplierAddress	string	100	Supplier Address
TenderQuotationForm	SupplierCode	string	5	Unique character for supplier
	ItemDescription	string	250	Item description
	TenderPrice	int	10	Price of tender price for each item
DeliveryOrder	ItemCode	string	5	Unique name for item. Start with one alphabet and follow by 3 numbers.
	SupplierCode	string	5	Unique character for supplier
	ReceivedDate	date	8	Format : DDMMYY Date of Delivery
	ReceivedQuantity	int	5	Quantity Received
	DeliveryOrderNumber	string	50	Unique reference number from Supplier

RequisitionList	RequisitionFormNumber	string	50	Unique name for requisition form
	ItemCode	string	5	Unique name for item. Start with one alphabet and follow by 3 numbers.
	EmployeeID	string	10	Employee ID
DepartmentList	DeptCode	string	10	Department ID
	NameOfDeptHead	string	50	Name of department head
	DepartmentName	string	50	Department Name
	CollectionPoint	string	100	Name of Place where to collect item delivery
	ContactName	string	50	The point of contact for the department
	TelephoneNo	string	10	Department telephone no.
	RepresentativeName	string	50	The nominated person to collect the disbursed item at the Collection Point
PurchaseOrder	PONumber	string	10	Unique name for the PO (Format: ##-#####/A)
	ItemCode	string	5	Unique name for item. Start with one alphabet and follow by 3 numbers.
	Orderedby	string	50	Name of person who raised the PO
	Approvedby	string	50	Name of person who approved the PO (Tentatively the Supervisor or above)
	DeliverTo(University)	string	50	Address of University

	OrderedQuantity	int	10	Quantity to purchase
	PODate	date	8	Date Format (DDMMYY)
InventoryList	ItemCode	string	5	Unique name for item. Start with one alphabet and follow by 3 numbers.
	ItemDescription	string	250	Item description
	BinNo	string	5	Location of stock in store
	UOM	string	10	Unit of measurement
	Category	string	50	Category of item
	ReorderQuantity	int	10	Number of reorder item
	Reorderlevel	int	10	'Healthy' stock level of the inventory
	InventoryQuantity	int	10	Total number of inventory
EmployeeList	EmployeeName	string	50	Name of Employee in department
	EmployeeID	string	10	EmployeeID
	EmployeeEmailAddress	string	100	Employee email