

# Table of Contents

Table of Contents.....	1
0. Introduction .....	4
0.1 Glossary.....	6
0.2 Unbound Use Case Legend .....	6
1. Requirements.....	7
1.1 Functional Requirements.....	7
1.2 Non-Functional Requirements.....	9
2. Assumptions.....	10
3. Use Cases .....	11
3.1 Use Case Titles .....	11
3.2 Use Case Diagram .....	11
3.3 Use Cases .....	12
4. Use Case Maps .....	16
4.1 Responsibilities .....	16
4.2 Unbound Use Case Maps .....	18
4.2.1 UC-01 Customer Enters, Completes Shopping and Leaves the Store.....	18
4.2.2 UC-02 Manager Opens/Closes and Monitors Current Store Efficiency.....	19
4.2.3 UC-03 Manager Attempts to Open a Service Point .....	20
4.2.4 UC-04 Manager Attempts to Close a Service Point .....	21
4.2.5 UC-05 Manager Opens the Store.....	21
4.2.6 UC-06 Manager Closes the Store .....	22
4.3 Bound Use Case Maps.....	23
4.3.1 UC-01 Customer Enters, Completes Shopping and Leaves the Store.....	23
4.3.2 UC-02 Manager Opens/Closes and Monitors Current Store Efficiency.....	23
4.3.3 UC-03 Manager Attempts to Open a Service Point .....	24
4.3.4 UC-04 Manager Attempts to Close a Service Point .....	24
4.3.5 UC-05 Manager Opens the Store.....	25
4.3.6 UC-06 Manager Closes the Store .....	25
5. Interaction Diagrams.....	26

5.1.1 UC-01 Customer Enters, Completes Shopping and Leaves the Store .....	26
5.1.2 UC-01 Service Point Selection Strategy by Customer Count .....	27
5.1.3 UC-01 Service Point Selection Strategy by Item Count.....	28
5.2.1 UC-02 Manager Opens/Closes and Monitors Current Store Efficiency .....	29
5.2.2 UC-02 Manager Monitoring Through Rate Strategy .....	30
5.2.3 UC-02 Manager Monitoring Through Queue Strategy .....	31
5.3 UC-03 Manager Attempts to Open a Service Point .....	32
5.4 UC-04 Manager Attempts to Close a Service Point.....	33
5.5 UC-05 Manager Opens the Store .....	33
5.6 UC-06 Manager Closes the Store .....	34
6. Structural Model .....	35
7. Object Specifications.....	36
7.1 Customer .....	36
7.2 Manager .....	38
7.3 ServicePoint .....	40
7.4 WaitQueueSystem .....	43
7.5 Store .....	44
7.6 CustomerHandler .....	47
7.7 ManagerHandler .....	48
7.8 ServicePointHandler.....	49
7.9 ManageStoreStrategy .....	50
7.10 QueueSizeStrategy.....	51
7.11 RatesStrategy .....	52
7.12 SelectSPStrategy .....	53
7.13 SelectSPCustsStrategy.....	54
7.14 SelectSPItemsStrategy .....	55
7.15 StoreAdditionStrategy.....	56
7.16 RateAddition .....	57
7.17 BatchAddition.....	58
8. Strategies .....	59
8.1 Service Point Selection Strategies (SPSS).....	59
8.1.1 Customer Count Strategy (CCS) .....	59
8.1.2 Item Count Strategy (ICS).....	59

8.2 Store Monitoring Strategy (SMS) .....	59
8.2.1 Arrival Rate Strategy (ARS).....	59
8.2.2 Queue Size and Idle Time Strategy (QSS).....	60
9.1 Testing Explanation .....	61
9.2 Testing Expansion.....	61
9.2.1 Simulation Configuration .....	61
9.2.2 All Customer Creation Options .....	62
9.2.3 Few Customers Run .....	63
9.2.4 Creation and Use of the Wait Queue .....	63
9.2.5A Service Point Selection Strategy: Customer Count.....	64
9.2.5B Service Point Selection Strategy: Item Count .....	64
9.2.6 Simultaneous Customers .....	65
9.2.7 Full Service Points .....	65
9.2.8A Manager Opens/Closes Service Points: Arrival Rate Strategy.....	66
9.2.8B Manager Opens/Closes Service Points: Queue Size/Idle Time Strategy.....	67
9.2.9 Customer Wait Time Upon Leaving the Store and Averages Upon Store Closing.....	68
9.2.10 Service Point Idle Times Upon Closing and Average Upon Store Closing.....	68
9.2.11 Manager Dynamically Changing Monitoring Strategies .....	69
10. Additional Information.....	70
10.1 Graphics User Information Explanation.....	70
10.2 Log Information.....	72
10.3 Implemented Design Patterns .....	72
10.4 Creating Batch Files.....	72

# 0. Introduction

The following document represents the final iteration for a store simulation program. This simulation will be used in stores to help managers and store owners anticipate and manage customer flow efficiently.

This document contains the requirements, assumptions, use cases, use case maps, interaction diagrams, the structural model and strategies as well as additional information for the simulation and takes into account two basic scenarios: Customer Enters, Completes Shopping and Leaves the Store and Manager Opens/Closes and Monitors Current Store Efficiency. The purpose of this text is to understand the scenarios, interactions and objects that are required and the messages that must be sent in order for the two scenarios mentioned above to be carried out.

The requirements have been divided into two sections: functional requirements, non-functional requirements. Each requirement has been uniquely identified by a code (FR-# for functional requirements and NFR# for non-functional requirements). Functional requirements are the task or jobs that are necessary for the functioning of the simulation (e.g. opening a service queue and processing a customer) and non-functional requirements are thought to be the requirements that are not directly part of the function of the simulation (e.g. A customer must be able to efficiently choose the service point they believe will be get them out the store the fastest).

There are two main actors in this simulation program in the scenarios, the manager and the customer. The manager is responsible for opening and closing the store as well as monitoring both service points and customer arrival rate to decide whether or not to open or close a service point. The store must be able to deal with the number of customers in the store in an efficient manner and in such a way that the manager can make efficient decisions based on the information he is gathering. The Use Cases take these requirements into account and have been created to provide a basic description of the five scenarios that are being covered in this iteration, without going into heavy detail about the internal workings of how the simulation works. Alternative scenarios have been provided for cases in which the expected sequence has not been fully completed. Six Use Cases have been included: Customer Enters, Completes Shopping and Leaves the Store, Manager Opens/Closes and Monitors Current Store Efficiency, Manager Attempts to Open a Service Point, Manager Attempts to Close a Service Point, Manager Opens the Store and finally Manager Closes the Store. All Use Cases have been cross-listed to previous requirements and uniquely identified by a number.

For each Use Case, a corresponding use case map and interaction diagram has been created. The use case maps include a related path set for a list of responsibilities corresponding to each sequence of the Use Cases. Bounded use case maps have been used in this iteration in order to show the objects involved in each scenario. A list of responsibilities has been included with the Use Case Maps and the Responsibilities of each Use Case Map make use of this list in order to map out the responsibilities required for each of the scenarios described earlier. Interaction diagrams have also been created to provide a nice outline of the messages that must be sent between the objects involved for each scenario. Diagrams

were created with regards to UML 2.0. Each interaction diagram has been identified by a unique number (SD-#). A responsibility cross-reference table has also been included in the document in order to link messages in the interaction diagrams with responsibilities from the Use Case Maps for each of the scenarios.

Also included is a structural diagram. The structural diagram provides an exterior view of the objects of the system and how they interact together to create the entire simulation. The store only has one manager which allows us to apply the singleton pattern to our architecture.

The simulation outputs a log of its output in order to keep track of the customers in the store.

The main purpose of this document is to create a sound and traceable design for the system. The requirements, Use Cases, Use Case Maps and Interaction Diagrams included below should provide a sound and efficient design for the basic functions of the store simulation.

# 0.1 Glossary

---

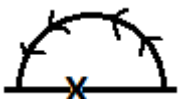
Term	Definition
Service Point	The cashier where once customers have finished their shopping will go to pay and leave the store.
Service Point Queue	The cashier's wait line
Wait Queue System	The line in which customers enter to select which service point to go to
Manager	Manager of the store. He/she opens, closes the store and service points as well as monitors the store's efficiency.
Customer	Someone who is shopping in the store.
Threshold	The limit of a rate or capacity at which the store operations should change.
Arrival Rate	The rate at which customers enter per hour
Item Processing Rate	The amount of time it takes for a service point to process an item in seconds.
Purchasing Area	The shopping area where customers gather their goods
Rate Strategy	Manager will monitor the arrival rate of the store and the service point queues
Queue Strategy	Manager will monitor the size of the service point queues and their idle times

# 0.2 Unbound Use Case Legend

---



Represents a skip



Represents a loop

# 1. Requirements

## 1.1 Functional Requirements

---

Unique-ID	Functional Requirement	Traceability
FR-01	The manager must know the maximum number of service points that can be open	C-01
FR-02	The manager must know the minimum number of service points that can be open	C-04
FR-03	The manager must know the initial number of service points to open when the store opens	C-11
FR-04	The manager must know their interval to monitor the store	C-02
FR-05	The manager must know the arrival rate threshold for the store at which they will start attempting to open service points	G-M-6, GD (12/03/12) <sup>1</sup>
FR-06	The manager will check the service point to see if the number of customers in its queue goes beyond its threshold	G-M-6, C-05
FR-07	The manager will check if the amount of idle time of a service point is beyond its threshold	C-08
FR-08	The manager will check to see if he needs to open a new service point if the arrival rate for a service point's queue is above arrival rate threshold	C-07
FR-09	The manager will check to see if he needs to close a service point if the arrival rate for that service point's queue is below the arrival rate threshold	C-10
FR-10	The customer must know the maximum number of customers in a service point queue	C-12
FR-11	The manager must know the maximum number of customers who can be in the store at a time	GD (13/03/12) <sup>2</sup>

FR-12	A customer must be able to enter the store when it's open and purchase their desired number of items	G-C-01
FR-13	If a customer has made purchases they must be able to enter the wait queue system	G-C-01
FR-14	A customer must be able to advance and leave the wait queue system	G-C-02
FR-15	A customer must be able to select a service point	G-C-02
FR-16	A customer must be able to enter the selected service point queue	G-C-03
FR-17	A customer must be able to advance in the service point queue and if they are at the front they proceed to the service point	G-C-06
FR-18	Customer must be able to be processed by a service point and leave the store	G-C-05
FR-19	The manager must be able to close a service point	G-M-01 G-M-09
FR-20	The manager must be able to open a service point	G-M-10
FR-21	The manager must be able to open the store with the initial number of service points	G-M-03 G-M-09
FR-22	The manager must be able to close the store.	G-M-10

<sup>1</sup>Justification: If a large volume of customers enter the store opening additional service points will prevent a backlog of customers at the service points when they go to pay.

Removed arrival rate threshold for manager closing service points because it does not take into account the number of customers currently in the store and could cause a backlog at the service points.

<sup>2</sup>Store should have a maximum capacity at which customers cannot longer enter to meet safety codes.



# 1.2 Non-Functional Requirements

---

Unique-ID	Non-Functional Requirements	Traceability
NF-01	A customer must be able to efficiently choose the service point they believe will be the fastest	S-C-01
NF-02	The manager must be able to minimize the total idle time of the cashiers	S-M-01
NF-03	There should be a graphical representation of the number of customers in the wait queue system and the service point queue	GD (12/03/12)
NF-04	There should be a visual representation if a service point is closed or open	GD (12/03/12)

# 2. Assumptions

Unique-ID	Description	Traceability
A1	The store opens and closes at set intervals and is not 24 hours	GD (12/03/12) <sup>3</sup>
A2	The manager checks the service points instead of being alerted by them	GD (12/03/12) <sup>4</sup>
A3	When the store is initially closed no more customers can enter but the remaining customers may finish their shopping	GD (12/03/12) <sup>5</sup>
A4	The waiting queue does not have a maximum number of customers	GD (12/03/12) <sup>6</sup>
A5	There is only one manager on duty at a time	GD (12/03/12) <sup>7</sup>

<sup>3</sup> Allows the store to close and reset instead of never reopening the store again

<sup>4</sup> Allows the manager to not always have to go and check up on everything after a certain interval

<sup>5</sup> Allows the store to be able to prepare for closing while customers are still in the store

<sup>6</sup> Customers can start lining up no matter how great the wait queue system is

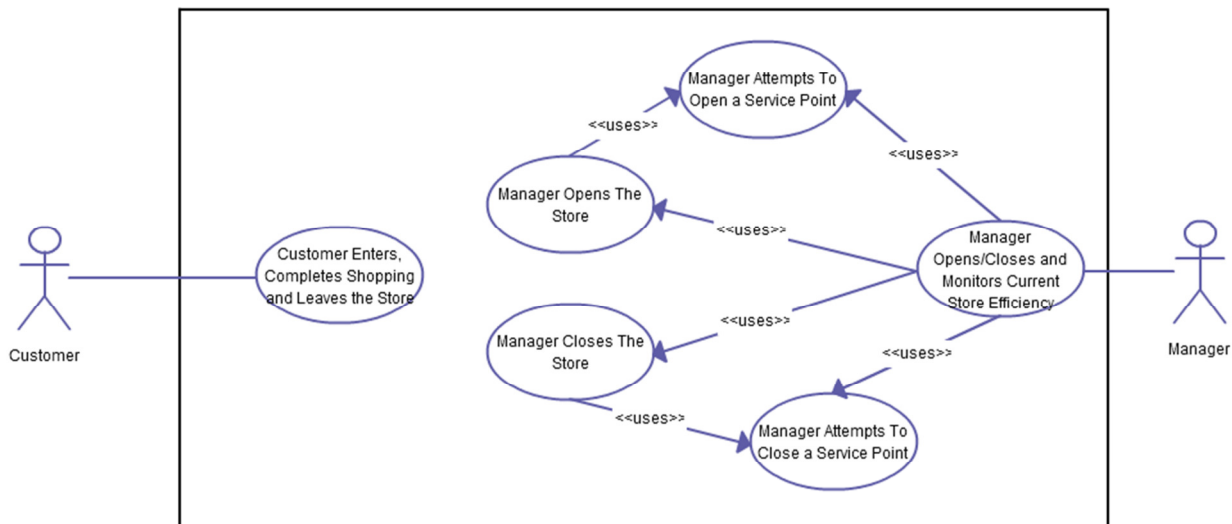
<sup>7</sup> Required for Singleton pattern

# 3. Use Cases

## 3.1 Use Case Titles

Unique-ID	Use-Case Title
UC-01	Customer Enters, Completes Shopping and Leaves the Store
UC-02	Manager Opens/Closes and Monitors Current Store Efficiency
UC-03	Manager Attempts to Open a Service Point
UC-04	Manager Attempts to Close a Service Point
UC-05	Manager Opens the Store
UC-06	Manager Closes the Store

## 3.2 Use Case Diagram



## 3.3 Use Cases

UC-01: Customer Enters, Completes Shopping and Leaves the Store	Traceability
<b>Description:</b> Customer enters the stores, collects their purchases, enters the wait queue system, selects a service point, completes transaction and leaves the store.	
<b>External Actors:</b> Customer	
<b>Related Use Cases:</b>	
<b>Precondition:</b> Store is open.	
<b>Triggering Event:</b> Customer enters the store.	
<b>Sequence:</b> <ol style="list-style-type: none"> <li>1. Customer enters the store and is assigned a number reflecting the number of items they will purchase.</li> <li>2. Customer shops in a store for a time proportional to the number of items they are buying and leaves the purchasing area.</li> <li>3. Customer enters the wait queue system and advances through it until they are at the front of the queue</li> <li>4. Customer selects a service point according to their strategy</li> <li>5. Customer leaves the wait queue system, enters the service point queue and advances through it until they are at the front</li> <li>6. Customer leaves the service point queue and enters the service point</li> <li>7. Customer is processed, pays and leaves the service point</li> <li>8. Customer leaves the store</li> </ol>	FR-12  FR-13, FR-14  FR-15 FR-14, FR-16, FR-17  FR-18
<b>Resulting Event:</b> Customer has completed shopping and leaves store	
<b>Post Condition:</b> Customer has left the store	
<b>Alternatives:</b>	
<b>Non-Functional Requirements:</b>	
<b>Comments:</b>	

UC-02: Manager Opens/Closes and Monitors Current Store Efficiency	Traceability
<b>Description:</b> Manager opens and closes the store as well as monitors the service points to determine whether or not to open or close one.	
<b>External Actors:</b> Manager	
<b>Related Use Cases:</b>	
<b>Precondition:</b> Store is closed	
<b>Triggering Event:</b> Manager wants to open the store	
<b>Sequence:</b> <ol style="list-style-type: none"> <li>1. Manager decides to open the store at opening time in accordance to UC-05.</li> </ol>	FR-21

<ul style="list-style-type: none"> <li>2. Manager monitors the store in accordance to their strategy and opens a service point see UC-03 Or closes a service point see UC-04</li> <li>3. Manager closes the store in accordance to UC-06</li> </ul>	FR-04, FR-19, FR-20 FR-22
<b>Resulting Event:</b> Manager closes the store	
<b>Post Condition:</b> Store is closed	
<b>Alternatives:</b>	
<b>Non-Functional Requirements:</b>	
<b>Comments:</b>	

<b>UC-03: Manager Attempts To Open a Service Point</b>	<b>Traceability</b>
<b>Description:</b> The manager wishes to open a service point.	
<b>External Actors:</b> Manager	
<b>Related Use Cases:</b> UC-02	
<b>Precondition:</b> The store is open or is about to open.	
<b>Triggering Event:</b> Store is being opened or manager decides to open a service point.	
<b>Sequence:</b> <ul style="list-style-type: none"> <li>1. The manager checks the number of open service points and determines it is less than the maximum.</li> <li>2. Manager finds the next available closed service point.</li> <li>3. Manager alerts this service point to open.</li> <li>4. Service point opens its queue and starts serving customers.</li> </ul>	FR-01, FR-02  FR-20
<b>Resulting Event:</b> A new service point is opened is possible.	
<b>Post Condition:</b> Store is open with one more service point available.	
<b>Alternatives:</b> <ul style="list-style-type: none"> <li>1.1: Manager checks if the number of open service points is equal to the maximum number and finds that it is. Manager cannot open another service point.</li> </ul>	FR-01, FR-02
<b>Non-Functional Requirements:</b>	
<b>Comments:</b>	

<b>UC-04: Manager Attempts To Close a Service Point</b>		<b>Traceability</b>
<b>Description:</b> The manager wishes to close a service point.		
<b>External Actors:</b> Manager		
<b>Related Use Cases:</b> UC-02		
<b>Precondition:</b> The store is open. <b>Triggering Event:</b> Store is being closed or manager decides to close a service point.		
<b>Sequence:</b> <ol style="list-style-type: none"> <li>1. The manager checks the number of open service points and determines it is greater than the minimum.</li> <li>2. Manager finds the first open service point to close.</li> <li>3. Manager alerts this service point to close.</li> <li>4. The service point closes their queue so no more customers may enter.</li> <li>5. The service point processes the remaining customers.</li> </ol>	FR-01, FR-02  FR-19	
<b>Resulting Event:</b> A service point is now closed if possible. <b>Post Condition:</b> Store is open with one less service point open or the store closed.		
<b>Alternatives:</b> <p>1.1: Manager checks if the number of open service points is equal to the minimum number and finds it is. Manager cannot close the service point.</p> <p>1.2 Manager is closing the store and so skips to step 3 without checking the number of open service points.</p>		
<b>Non-Functional Requirements:</b>		
<b>Comments:</b>		

<b>UC-05: Manager Opens The Store</b>		<b>Traceability</b>
<b>Description:</b> The manager wishes to open the store.		
<b>External Actors:</b> Manager		
<b>Related Use Cases:</b> UC-02		
<b>Precondition:</b> The store is closed. <b>Triggering Event:</b> The manager wishes to open the closed store.		
<b>Sequence:</b> <ol style="list-style-type: none"> <li>1. Manager opens the initial number of service points using UC-03.</li> <li>2. The manager sets the store to open so customer may enter.</li> </ol>	FR-03, FR-20  FR-21	
<b>Resulting Event:</b> Store is opened. <b>Post Condition:</b> Store is open.		

<b>Alternatives:</b>	
<b>Non-Functional Requirements:</b>	
<b>Comments:</b>	

<b>UC-06: Manager Closes The Store</b>	<b>Traceability</b>
<b>Description:</b> The manager wishes to close the store.	
<b>External Actors:</b> Manager	
<b>Related Use Cases:</b> UC-02	
<b>Precondition:</b> The store is opened. <b>Triggering Event:</b> The manager wishes to close the open store.	
<b>Sequence:</b> <ol style="list-style-type: none"> <li>1. Manager closes the store doors so no new customers may enter.</li> <li>2. Manager waits for the purchasing area and wait queue system to empty.</li> <li>3. The manager goes to each open service point and closes it as in UC-04.</li> <li>4. After all the customers leave the store the manager completes store closing.</li> </ol>	FR-22 FR-19
<b>Resulting Event:</b> Store is closed. <b>Post Condition:</b> Store is closed.	
<b>Alternatives:</b>	
<b>Non-Functional Requirements:</b>	
<b>Comments:</b>	

# 4. Use Case Maps

## 4.1 Responsibilities

---

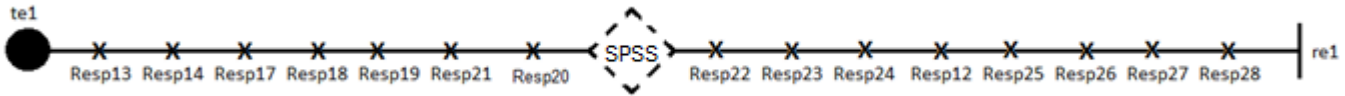
Unique-ID	Responsibility	Use Cases
Resp1	Manager is alerted it is time to open the store	UC-02
Resp2	Manager is alerted it is time to close the store	UC-02
Resp3	Manager opens the store	UC-02, UC-05
Resp4	Manager closes the store	UC-02, UC-06
Resp5	Manager monitors the store	UC-02
Resp6	Manager alerts the service point to open	UC-03
Resp7	Manager alerts service point to close	UC-04
Resp8	Manager checks the number of open service points	UC-02, UC-03, UC-04
Resp9	Manager finds next available closed service point	UC-03
Resp10	Manager finds next available open service point to close	UC-04
Resp11	Manager sets the store to closed	UC-06
Resp12	Customer is removed from the service point queue	UC-01
Resp13	Customer is assigned a number of purchases	UC-01
Resp14	Customer is added to the Purchasing Area	UC-01
Resp15	Manager checks the number of customers in the purchasing area	UC-02
Resp16	Manager checks the number of customers in the wait queue system	UC-02
Resp17	Customer is removed from the Purchasing area	UC-01
Resp18	Customer enters the wait queue system	UC-01
Resp19	Customer advances through the wait queue system	UC-01
Resp20	Customer evaluates all the service points	UC-01



Resp21	Customer selects service point	UC-01
Resp22	Customer leaves the wait queue system	UC-01
Resp23	Customer enters the service point queue	UC-01
Resp24	Customer advances in service point queue	UC-01
Resp25	Customer enters the service point	UC-01
Resp26	Customer is processed	UC-01
Resp27	Customer leaves the service point	UC-01
Resp28	Customer leaves the store	UC-01
Resp29	Service point opens its queue to customers and starts processing them	UC-03
Resp30	Service point checks the number of customers in its queue	UC-02
Resp31	Manager checks if the store is empty	UC-06
Resp32	Service point closes its queue	UC-04
Resp33	Manager's store monitoring interval expires	UC-02
Resp34	Manager's store monitoring interval resets	UC-02

# 4.2 Unbound Use Case Maps

## 4.2.1 UC-01 Customer Enters, Completes Shopping and Leaves the Store



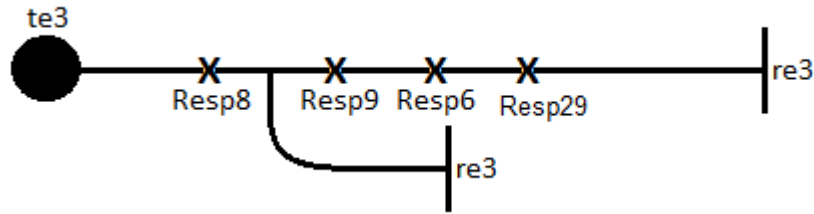
<b>te1</b>	Customer enters the store	
<b>re1</b>	Customer has completed shopping and leaves store	
<b>RespID</b>	<b>RESPONSIBILITIES</b>	<b>Use Cases Steps</b>
Resp13	Customer is assigned a number of purchases	UC(01,1)
Resp14	Customer is added to the Purchasing Area	UC(01,1)
Resp17	Customer is removed from the Purchasing area	UC(01,2)
Resp18	Customer enters the wait queue system	UC(01,3)
Resp19	Customer advances through the wait queue system	UC(01,3)
Resp20	Customer evaluates all the service points	UC-01
Resp21	Customer selects service point	UC(01,4)
Resp22	Customer leaves the wait queue system	UC(01,5)
Resp23	Customer enters the service point queue	UC(01,5)
Resp24	Customer advances in service point queue	UC(01,5)
Resp12	Customer is removed from the service point queue	UC(01,6)
Resp25	Customer enters the service point	UC(01,6)
Resp26	Customer is processed	UC(01,7)
Resp27	Customer leaves the service point	UC(01,7)
Resp28	Customer leaves the store	UC(01,8)
SPSS	Customer selects a service point depending on strategy (see section 8 for Strategy)	UC(01,4)

## 4.2.2 UC-02 Manager Opens/Closes and Monitors Current Store Efficiency



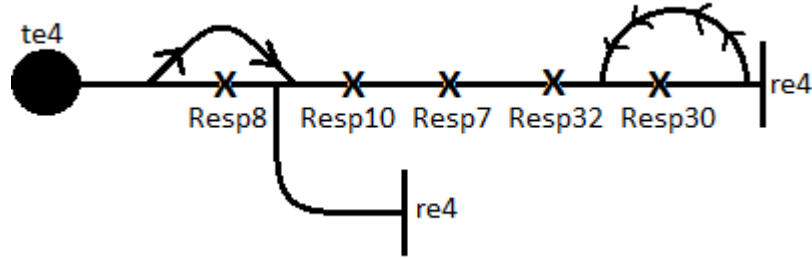
<b>te2</b>	Manager wants to open the store	
<b>re2</b>	Manager closes the store	
<b>RespID</b>	<b>RESPONSIBILITIES</b>	<b>Use Cases Steps</b>
Resp1	Manager is alerted it is time to open the store	UC(02,1)
Resp3	Manager opens the store	UC(02,1)
Resp5	Manager monitors the store	UC-02
Resp33	Manager's store monitoring interval expires	UC(02,2)
Resp34	Manager's store monitoring interval resets	UC(02,2)
Resp2	Manager is alerted it is time to close the store	UC(02,3)
Resp4	Manager closes the store	UC(02,3)
SMS	Manager monitors store in accordance to their strategy and may open or close a service point (see section 8 for Strategy)	UC(02, 2)

### 4.2.3 UC-03 Manager Attempts to Open a Service Point



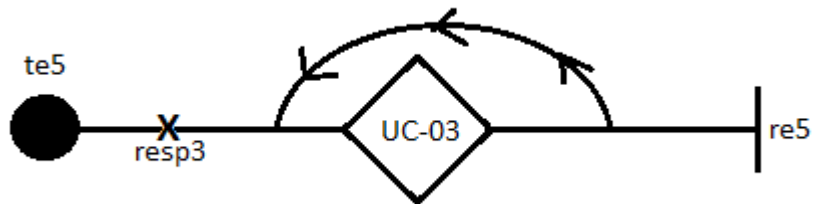
<b>te3</b>	Store is being opened or manager decides to open a service point.	
<b>re3</b>	A new service point is opened is possible.	
<b>RespID</b>	<b>RESPONSIBILITIES</b>	<b>Use Cases Steps</b>
Resp8	Manager checks the number of open service points	UC(03,1), UC(03,1.1)
Resp9	Manager finds next available closed service point	UC(03,2)
Resp6	Manager alerts the service point to open	UC(03,3)
Resp29	Service point opens its queue to customers and starts processing them	UC(03,4)

#### 4.2.4 UC-04 Manager Attempts to Close a Service Point



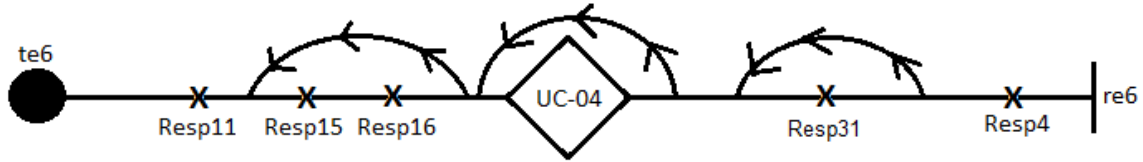
<b>te4</b>	The store is open	
<b>re4</b>	A service point is now closed if possible	
<b>RespID</b>	<b>RESPONSIBILITIES</b>	<b>Use Cases Steps</b>
Resp8	Manager checks the number of open service points	UC(04,1), UC(04,1.1), UC(04, 1.2)
Resp10	Manager finds next available open service point to close	UC(04,2)
Resp7	Manager alerts service point to close	UC(04,3)
Resp32	Service point closes its queue	UC(04,4)
Resp30	Service point checks the number of customers in its queue	UC(04,5)

#### 4.2.5 UC-05 Manager Opens the Store



<b>te5</b>	The store is closed	
<b>re5</b>	Store is opened	
<b>RespID</b>	<b>RESPONSIBILITIES</b>	<b>Use Cases Steps</b>
Resp3	Manager opens the store	UC(05,2)

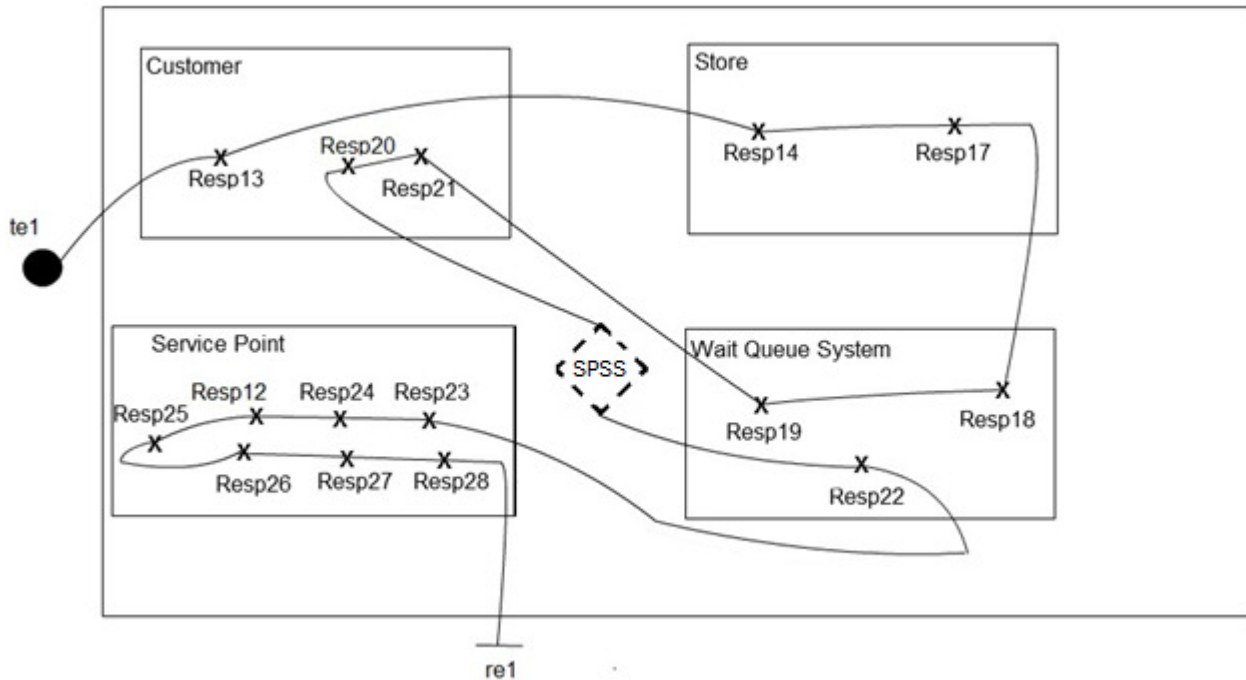
## 4.2.6 UC-06 Manager Closes the Store



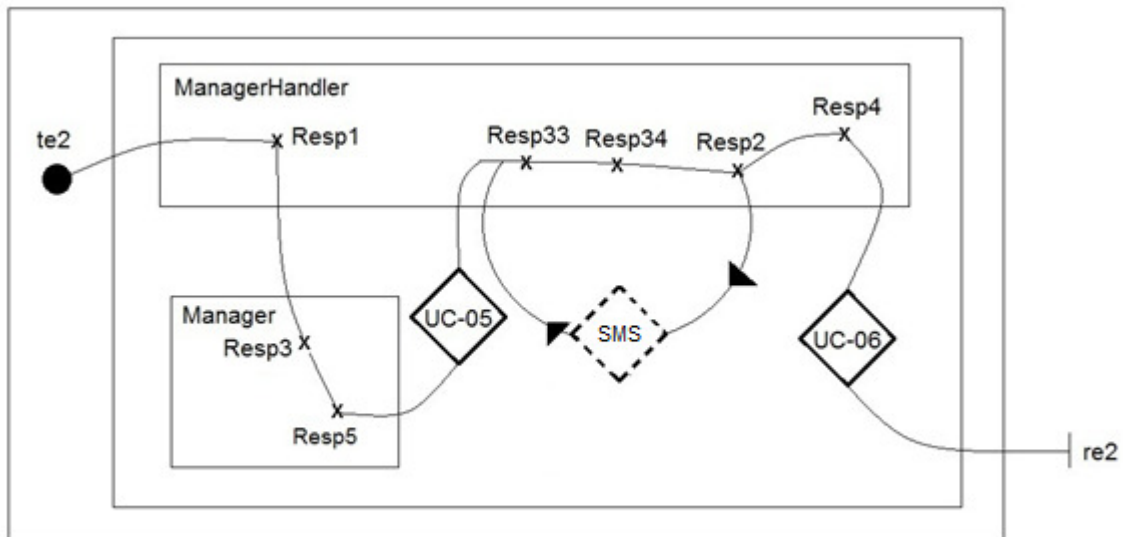
<b>te6</b>	The store is opened	
<b>re5</b>	Store is closed	
<b>RespID</b>	<b>RESPONSIBILITIES</b>	<b>Use Cases Steps</b>
Resp11	Manager sets the store to closed	UC(06,1)
Resp15	Manager checks the number of customers in the purchasing area	UC(06,2)
Resp16	Manager checks the number of customers in the wait queue system	UC(06,2)
Resp31	Manager checks if the store is empty	UC(06,4)
Resp4	Manager closes the store	UC(06,4)

# 4.3 Bound Use Case Maps

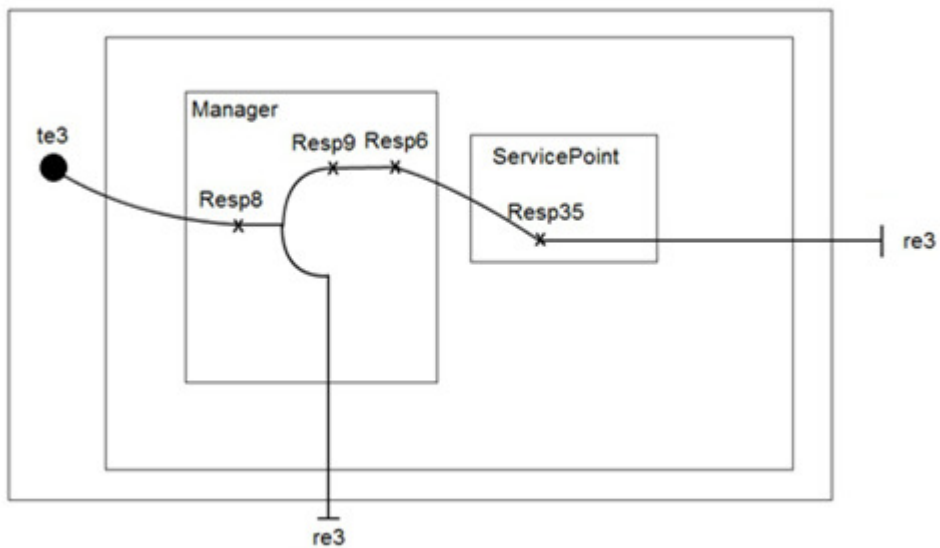
## 4.3.1 UC-01 Customer Enters, Completes Shopping and Leaves the Store



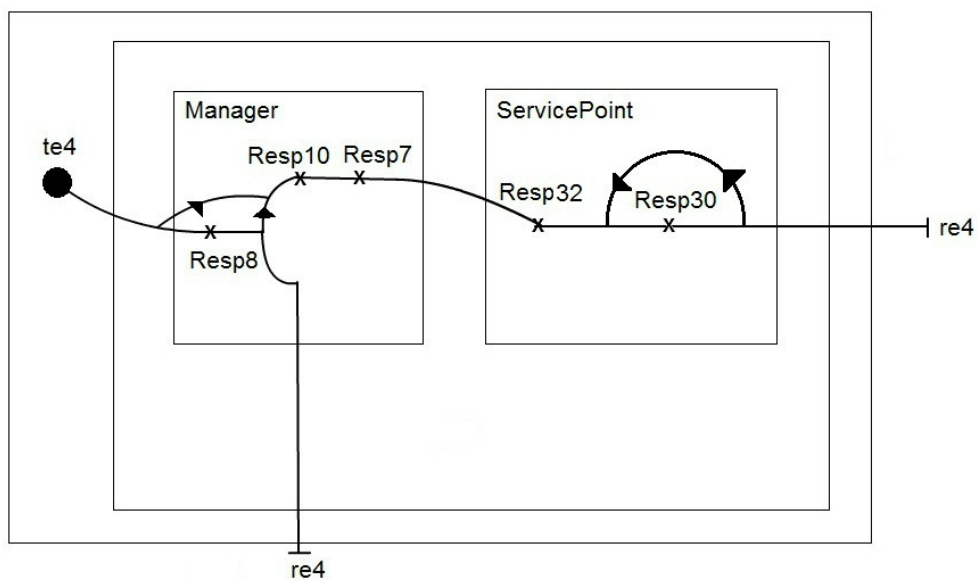
## 4.3.2 UC-02 Manager Opens/Closes and Monitors Current Store Efficiency



### 4.3.3 UC-03 Manager Attempts to Open a Service Point

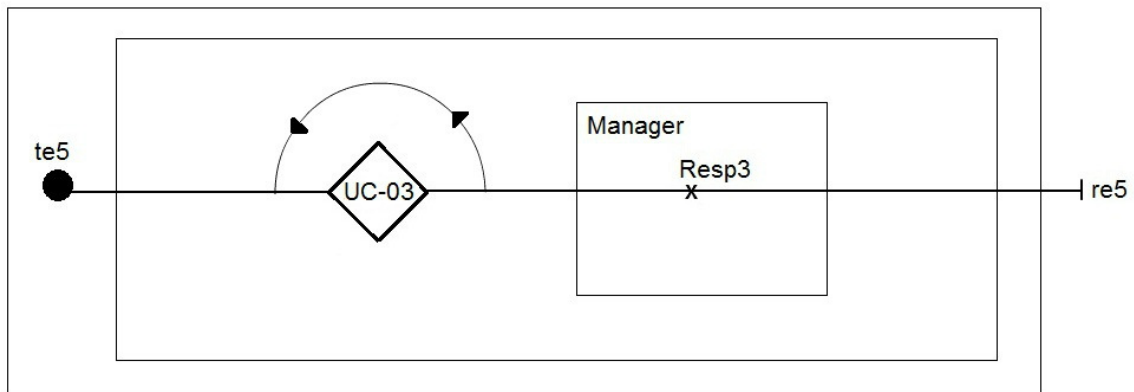


### 4.3.4 UC-04 Manager Attempts to Close a Service Point

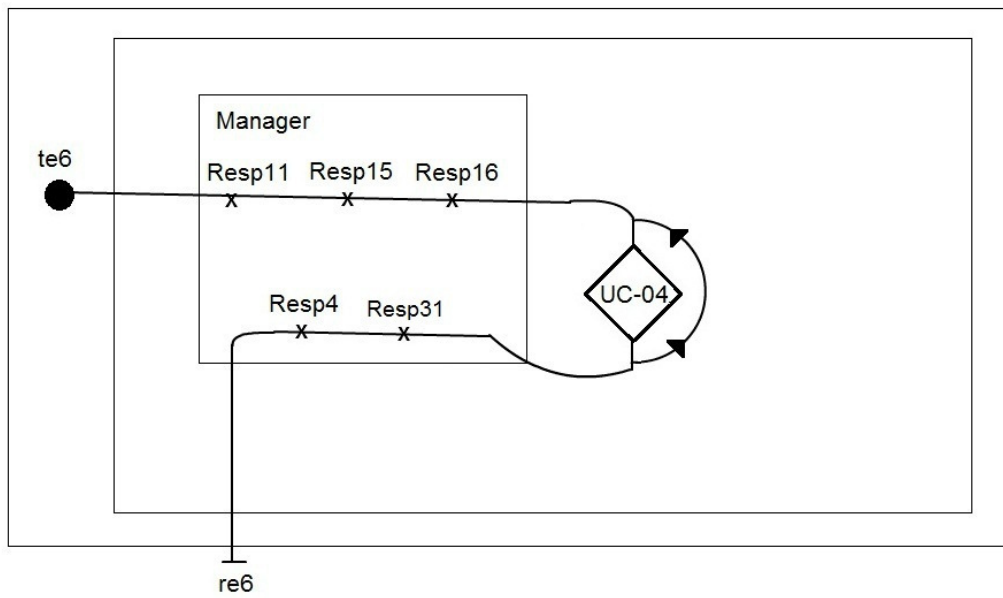




### 4.3.5 UC-05 Manager Opens the Store

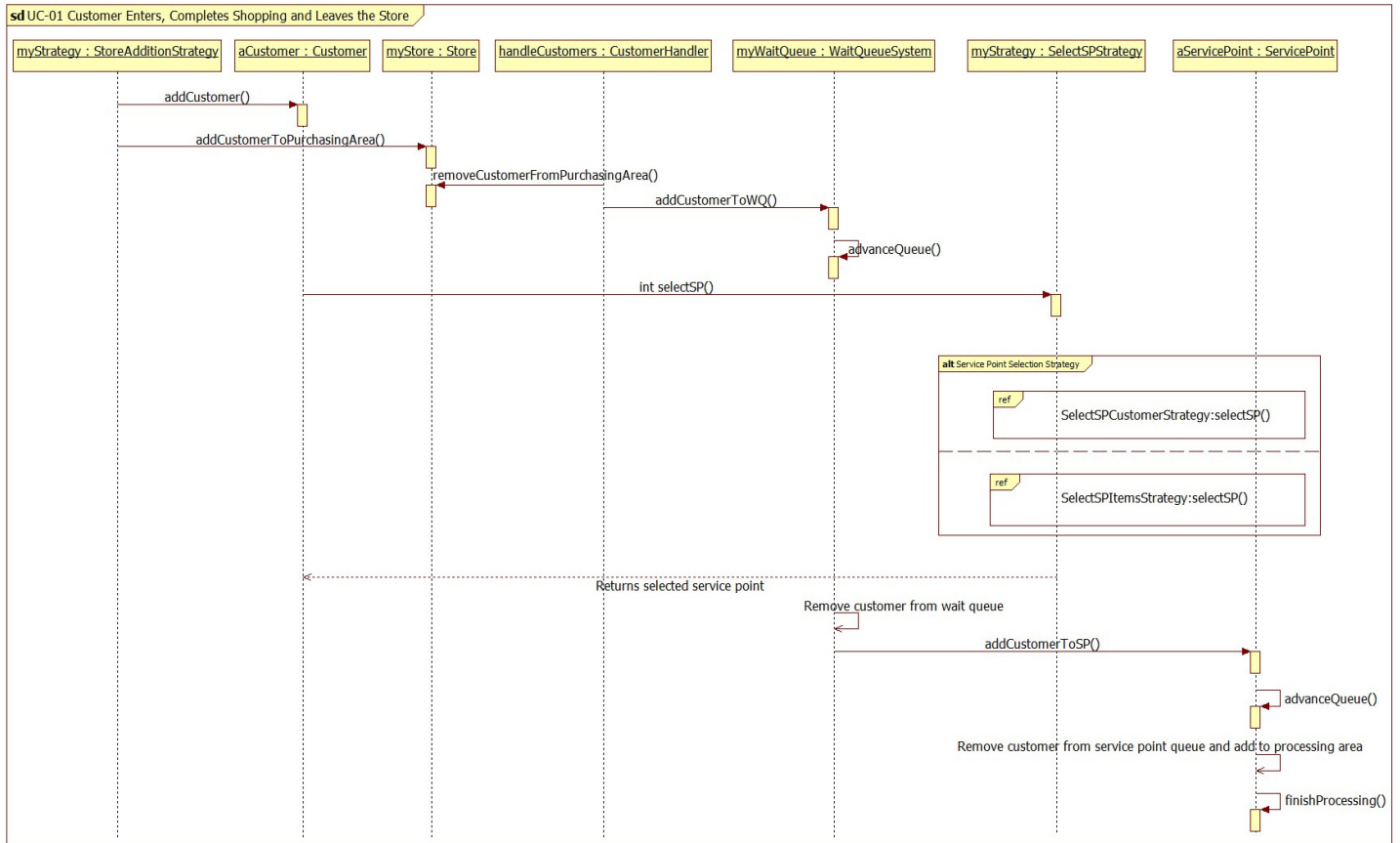


### 4.3.6 UC-06 Manager Closes the Store

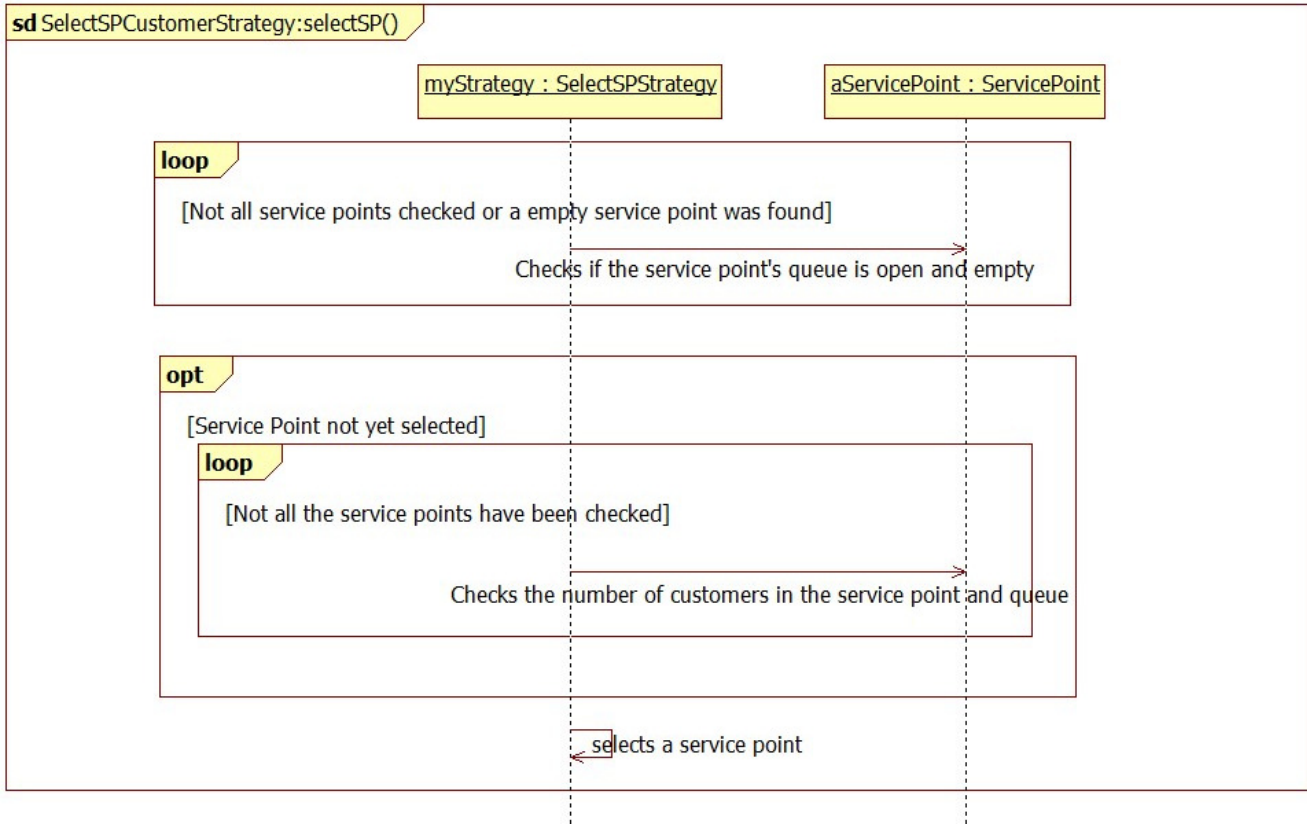


# 5. Interaction Diagrams

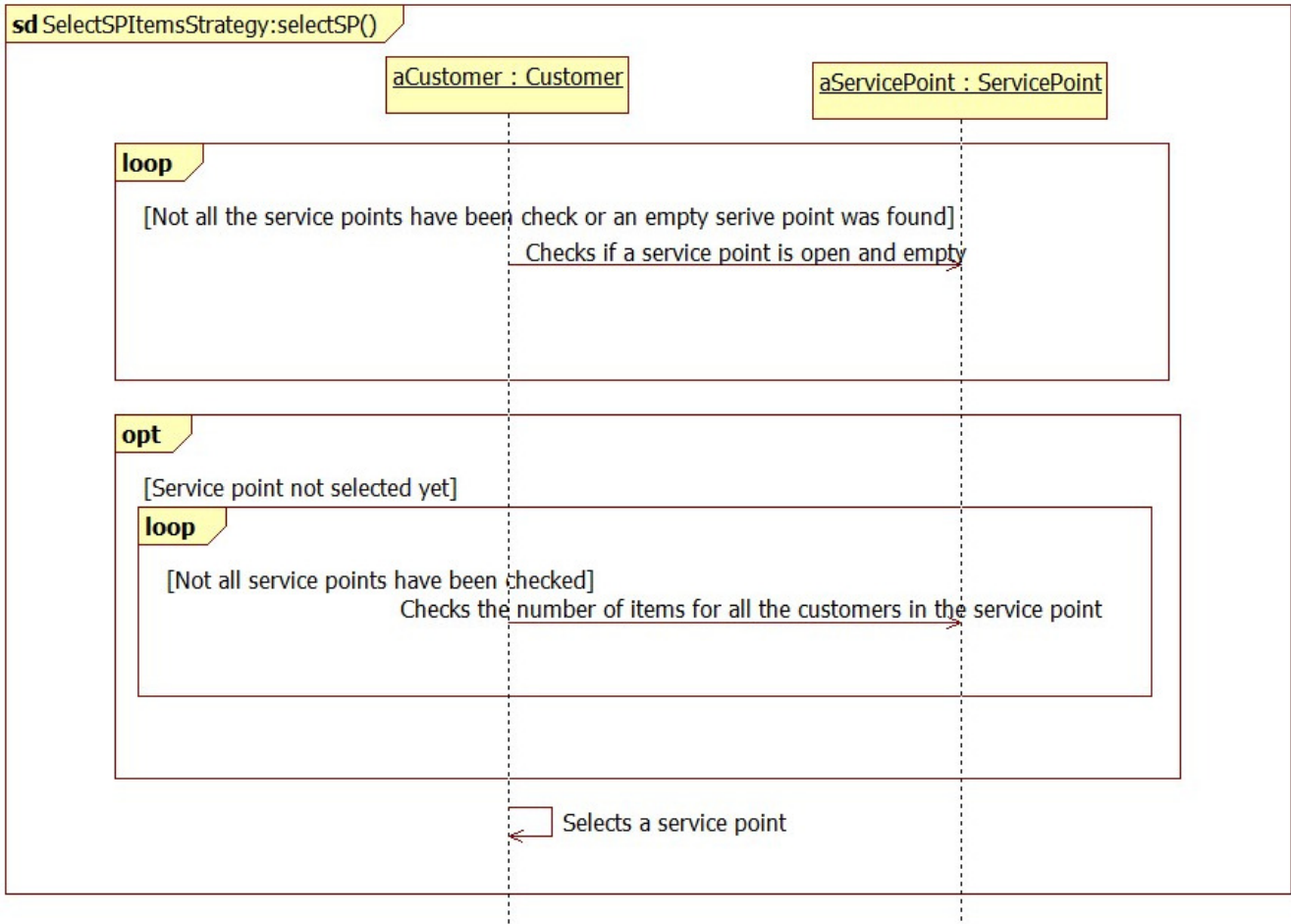
## 5.1.1 UC-01 Customer Enters, Completes Shopping and Leaves the Store



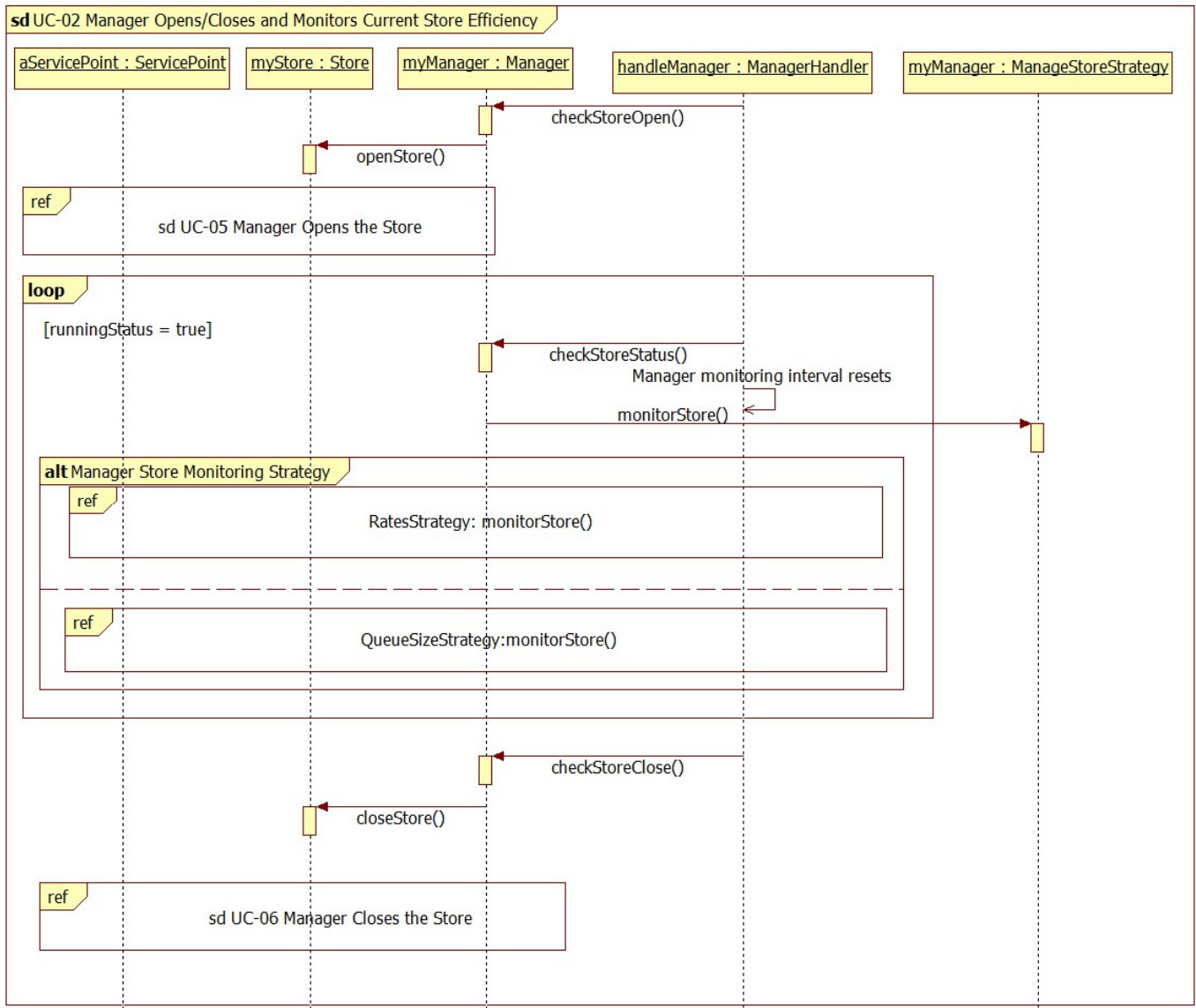
## 5.1.2 UC-01 Service Point Selection Strategy by Customer Count



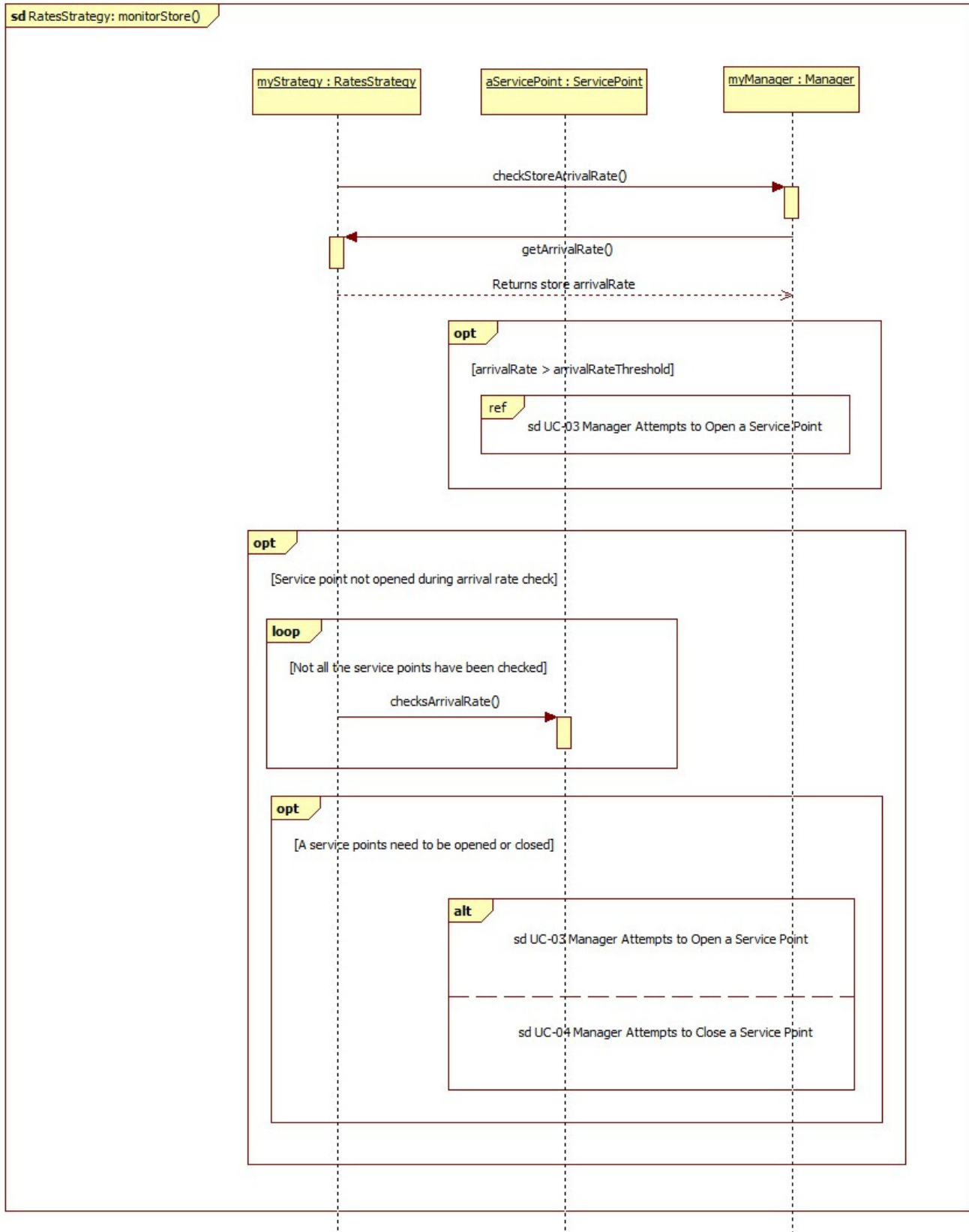
### 5.1.3 UC-01 Service Point Selection Strategy by Item Count



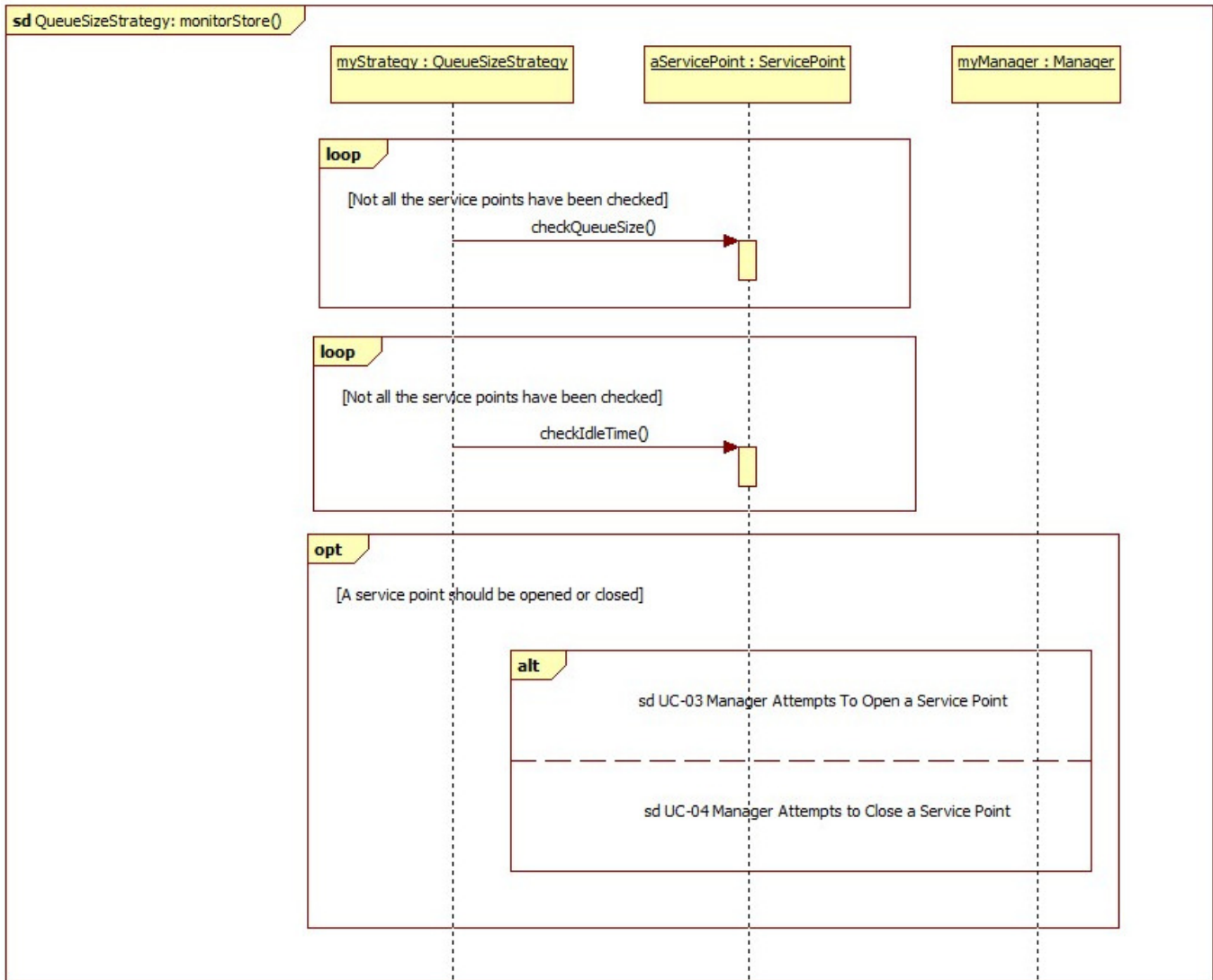
## 5.2.1 UC-02 Manager Opens/Closes and Monitors Current Store Efficiency



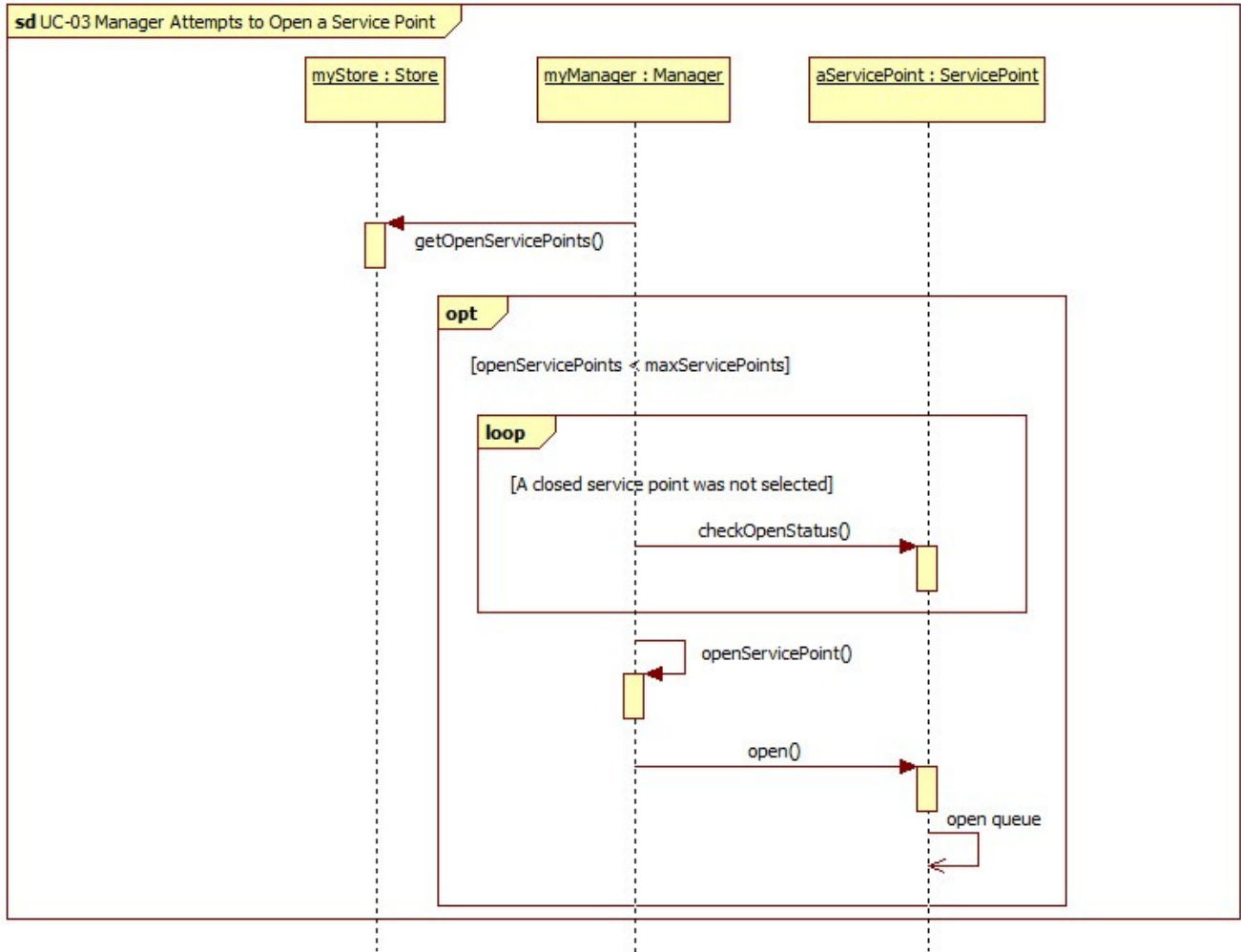
## 5.2.2 UC-02 Manager Monitoring Through Rate Strategy



### 5.2.3 UC-02 Manager Monitoring Through Queue Strategy

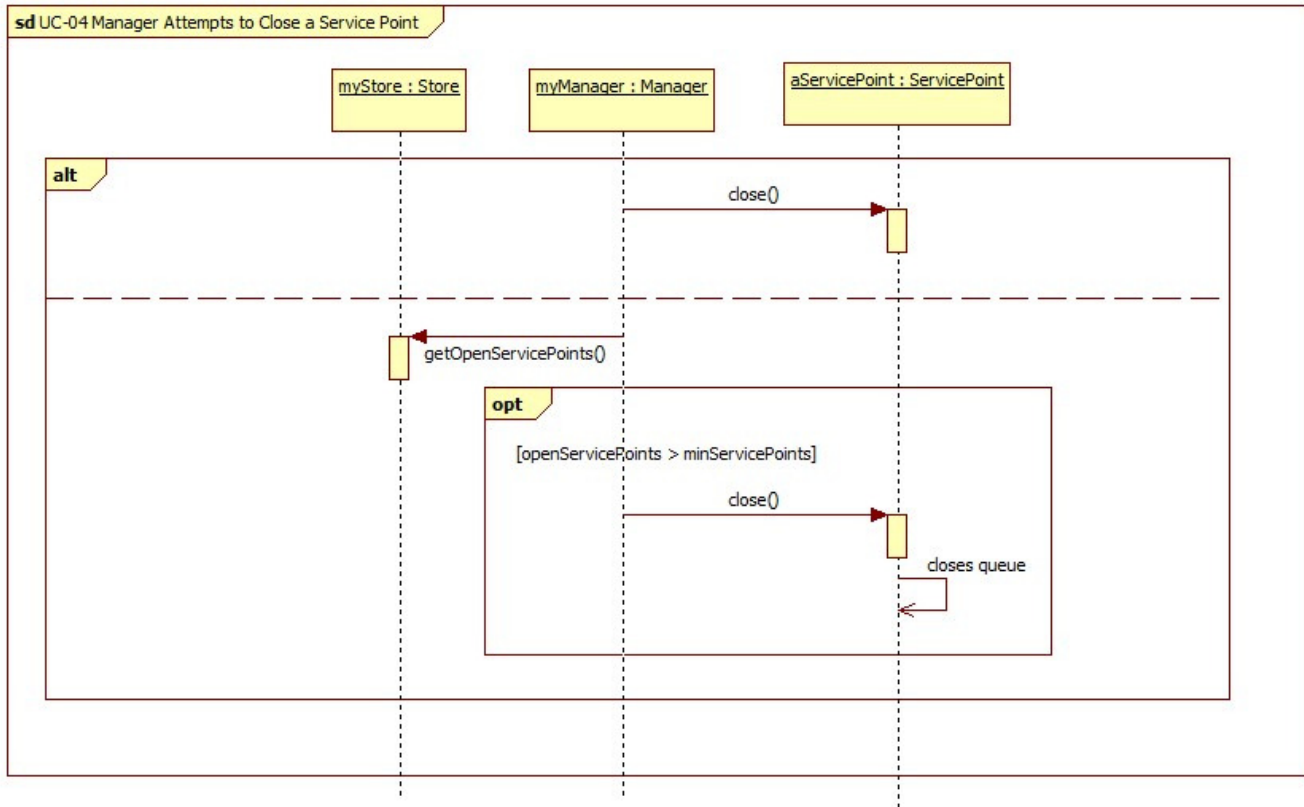


### 5.3 UC-03 Manager Attempts to Open a Service Point

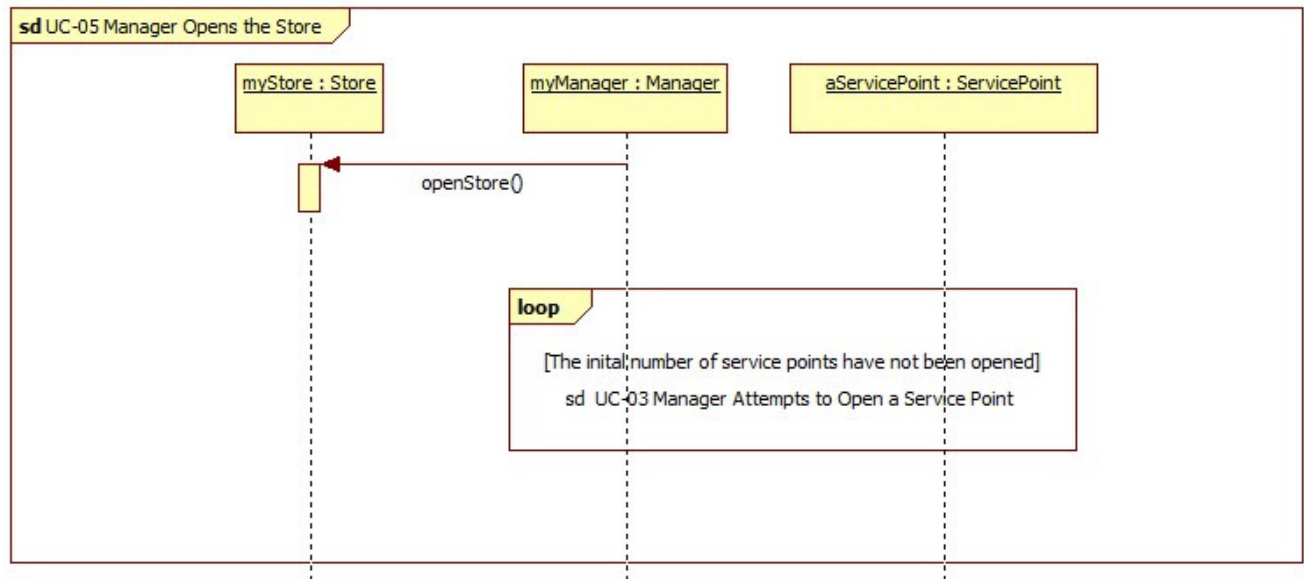




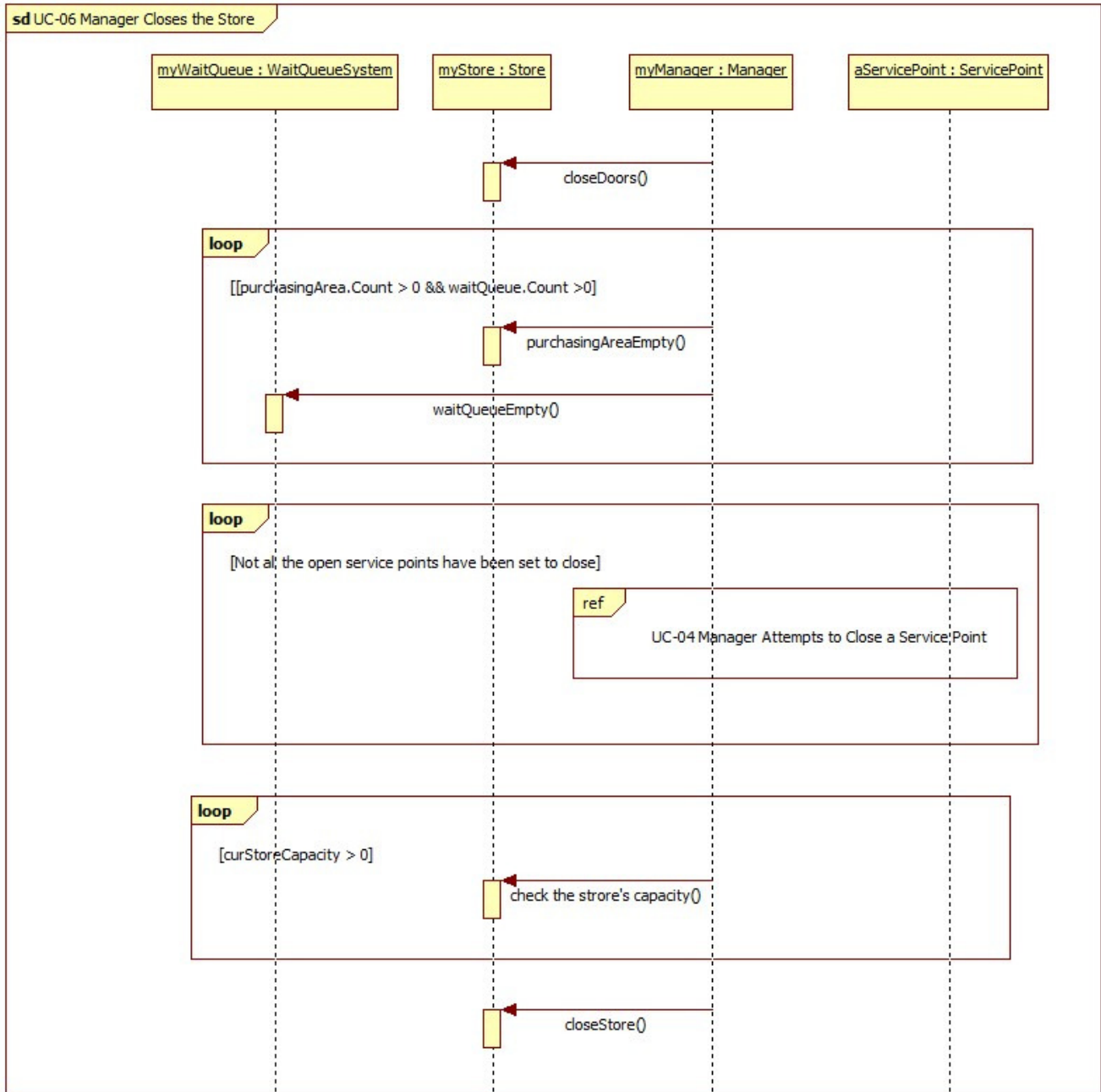
## 5.4 UC-04 Manager Attempts to Close a Service Point



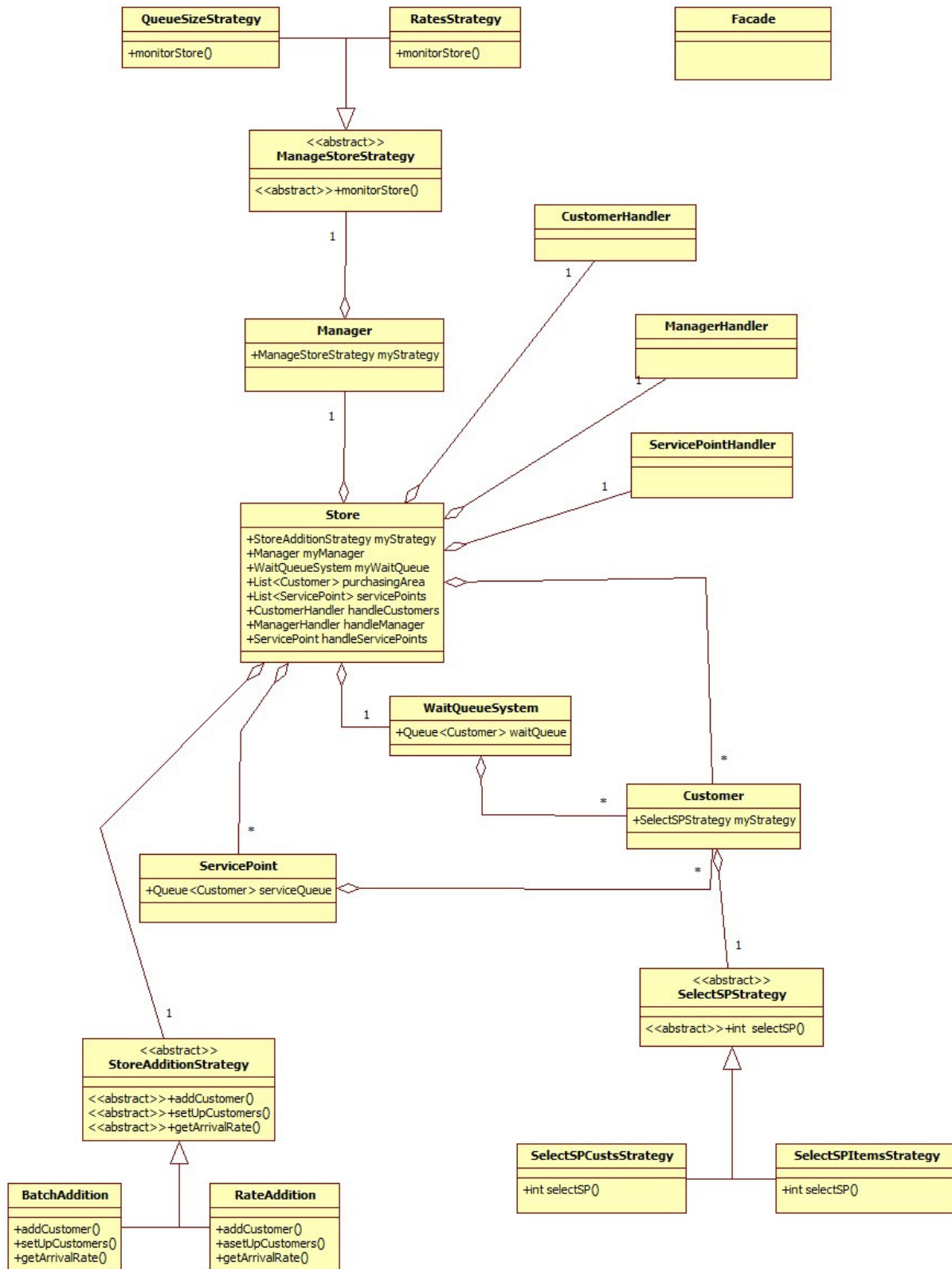
## 5.5 UC-05 Manager Opens the Store



## 5.6 UC-06 Manager Closes the Store



# 6. Structural Model



# 7. Object Specifications

## 7.1 Customer

Name	ID	Description	Inheritance
Customer	Act-1	Customer class represents the customers that enter the store. It has a set number of items to purchase, the time it was/will be added to the store, and a strategy to pick a service point. It can be created with all the attributions specified or can create a random number of items and a defaulted service point selection strategy. It is responsible for selecting the service point to entering according to its selection strategy. It is also responsible for keeping track of the amount of time it waited in the wait queue system or a service point queue.	None

Unique-ID	Responsibilities	Collaborators	Use Cases
Resp13	Customer is assigned a number of purchases		UC-01
Resp20	Customer evaluates all the service points		UC-01
Resp21	Customer selects service point		UC-01

### Data Member Dictionary

Variable Name	Type	Status	Procedures
myStrategy	SelectSPStrategy	Read Only	Customer() selectsSP()
custID	int	Read/Write	
numOfItems	int	Read/Write	Customer()
timeEntered	int	Read/Write	Customer()

waitTime	int	Read/Write	
_r	Random	Read Only	Customer()
maxNumItems	int	Read Only	Customer()
timePerPurchase	int	Read/Write	
timePerProcess	int	Read/Write	

## Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
Customer				Resp13
Customer		timeEntered items custStrategy	int int SelectSPStrategy	Resp13
selectServicePoint	Int			Resp20 Resp21
getTimePerProcess	void			
setTimePerProcess	void	int	newProcessTime	

## 7.2 Manager

---

Name	ID	Description	Inheritance
Manager	Act-2	The manager is responsible for opening, closing, and monitoring the store. It has a strategy for monitoring the store and an interval to know when to monitor the store. It monitors the store by closing and opening service point in accordance to its strategy.	None

Unique-ID	Responsibilities	Collaborators	Use Cases
Resp3	Manager opens the store		UC-02, UC-05
Resp5	Manager monitors the store		UC-02
Resp8	Manager checks the number of open service points		UC-02, UC-03, UC-04
Resp9	Manager finds next available closed service point		UC-03
Resp6	Manager alerts the service point to open		UC-03
Resp10	Manager finds next available open service point to close		UC-04
Resp7	Manager alerts service point to close		UC-04
Resp11	Manager sets the store to closed		UC-06
Resp15	Manager checks the number of customers in the purchasing area		UC-02
Resp16	Manager checks the number of customers in the wait queue system		UC-02
Resp31	Manager checks if the store is empty		UC-06
Resp4	Manager closes the store	ManagerHandler	UC-02, UC-06

## Data Member Dictionary

Variable Name	Type	Status	Procedures
myStrategy	ManagerStoreStrategy	Read Only	
arrivalInterval	int	Read/Write	
nextCheckTime	int	Read/Write	

## Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
Manager				
openStore	void			Resp3
closeStore	void			Resp11 Resp15 Resp16 Resp31 Resp4
checkStoreArrivalRate	bool			
openServicePoint	void	openMe	ServicePoint	Resp6
openServicePoint	void			Resp6 Resp10
closeServicePoint	void	closeMe	ServicePoint	Resp9 Resp7
setStrategy	void	newStrategy	ManagerStoreStrategy	
updateStoreArrivalRate	void			
monitorStore	void			Resp5 Resp8

## 7.3 ServicePoint

Name	ID	Description	Inheritance
ServicePoint	Act-3	The service point represents the cashiers the process the customers. They have a queue which is responsible for adding and removing customers that want to be processed by the service point. It is responsible for processing the customers and removing them from the store. It keeps track of its idle time and queue size and arrival rate.	None

Unique-ID	Responsibilities	Collaborators	Use Cases
Resp23	Customer enters the service point queue		UC-01
Resp24	Customer advances in service point queue		UC-01
Resp12	Customer is removed from the service point queue		UC-01
Resp25	Customer enters the service point		UC-01
Resp26	Customer is processed		UC-01
Resp27	Customer leaves the service point		UC-01
Resp28	Customer leaves the store		UC-01
Resp29	Service point opens its queue to customers and starts processing them		UC-03
Resp32	Service point closes its queue		UC-04
Resp30	Service point checks the number of customers in its queue		UC-02

### Data Member Dictionary

Variable Name	Type	Status	Procedures
---------------	------	--------	------------



numCustomerSinceCheck	int	Read/Write	addCustomerToSP checkArrivalRate
totalIdleTime	int	Read/Write	
idleTime	int	Read/Write	open advanceQueue checkIdleTime
sessionIdleTime	int	Read/Write	open close
arrivalRate	int	Read/Write	checksArrivalRate
maxIdleTime	int	Read Only	checkIdleTime
maxArrivalRate	int	Read Only	checksArrivalRate
minArrivalRate	int	Read Only	checksArrivalRate
serviceNumber	int	Read/Write	checksArrivalRate
processStartTime	int	Read/Write	advanceQueue
processingCustomer	Customer	Read/Write	advanceQueue finishProcessing
emptyStatus	bool	Read/Write	ServicePoint open advanceQueue finishProcessing
openStatus	bool	Read/Write	checkOpenStatus ServicePoint open close
everOpened	bool	Read/Write	ServicePoint open
maxQueueSize	int	Read Only	checkQueueSize
curCapacity	int	Read Only	getCapacity

			addCustomer finishProcessing
serviceQueue	Queue<Customer>	Read/Write	checkQueueSize addCustomerToSP advanceQueue

## Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
ServicePoint		assignedNumber	int	
open	void			Resp29
close	void			Resp32
getCapacity	int			
addCustomerToSP	void			Resp23
advanceQueue	void			Resp24 Resp25
finishProcessing	Void			Resp12 Resp26 Resp27 Resp28
checksArrivalRate	Int			
checkQueueSize	Int			Resp30
checkIdleTime	Int			
checkOpenStatus	bool			

## 7.4 WaitQueueSystem

Name	ID	Description	Inheritance
WaitQueueSystem	Act-4	The wait queue system is responsible for holding the customers in their time between the purchasing area and the service points.	None

Unique-ID	Responsibilities	Collaborators	Use Cases
Resp18	Customer enters the wait queue system		UC-01
Resp19	Customer advances through the wait queue system		UC-01
Resp22	Customer leaves the wait queue system		UC-01

### Data Member Dictionary

Variable Name	Type	Status	Procedures
newFirst	bool	Read/Write	advanceQueue
waitQueue	Queue<Customer>	Read/Write	addCustomerToWQ advanceQueue waitQueueEmpty

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
addCustomerToWQ	void	addMe	Customer	Resp18
advanceQueue	void			Resp19 Resp22
waitQueueEmpty	bool			

## 7.5 Store

Name	ID	Description	Inheritance
Store	Act-5	The store is responsible for holding the customers when they enter the store in the purchasing area. The store can be opened and closed by the manager. The store keeps track of and calculates its arrival rate. The store determines when to add a customer based on its customer addition strategy.	None

Unique-ID	Responsibilities	Collaborators	Use Cases
Resp14	Customer is added to the Purchasing Area		UC-01
Resp17	Customer is removed from the Purchasing area		UC-01

### Data Member Dictionary\*Missing One + check the procedures now

Variable Name	Type	Status	Procedures
myStrategy	StoreAdditionStrategy	Read Only	Store
myManager	Manager	Read Only	getManager
totalCustomerWaitTime	int	Read/Write	Store
ArrivalRateThresholds	int[]	Read Only	Store
arrivalRate	int	Read//Write	getArrivalRate
desiredArrivalRate	Int	Read/Write	
numCustomersSinceCheck	int	Read/Write	addCustomerToPurchasingArea
openTime	int	Read/Write	getOpenTime setOpenTime
closeTime	int	Read/Write	getCloseTime setCloseTime
storeDoorStatus	bool	Read/Write	Store

			closeDoors openStore
storeRunnigStatus	bool	Read/Write	Store closeStore openStore
maxServicePoints	int	Read/Write	openStore setMaxServicePoints
minServicePoints	int	Read/Write	setMinServicePoints
initServicePoints	int	Read/Write	setInitServicePoints
openServicePoints	int	Read/Write	getOpenServicePoints
maxStoreCapacity	int	Read/Write	addCustomerToPurchasingArea setStoreMax
curStoreCapacity	int	Read/Write	
myWaitQueueSystem	WaitQueueSystem	Read/Write	StoreLoop
customersToAdd	List<Customer>	Read/Write	
servicePoints	List<ServicePoint>	Read/Write	Store
purchasingArea	List<Customer>	Read/Write	addCustomerToPurchasingArea removeCustomerFromPurchasingArea
cusNumber	int	Read/Write	addCustomerToPurchasingArea
handleCustomers	CustomerHandler	Read/Write	StoreLoop
handleServicePoints	ServicePointHandler	Read/Write	StoreLoop

## Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
StoreLoop	void			
Store()				
closeDoors	void			

closeStore	void			
openStore	void			
getOpenServicePoints	int			
addCustomerToPurchasingArea	bool	addMe	Customer	Resp14
removeCustomerFromPurchasingArea	void	removeMe	Customer	Resp17
purchasingAreaEmpty	bool			
getArrivalRate	int			
getManger	Manager			
getClosingTime	int			
getOpenTime	int			
setCloseTime	void	newCloseTime	int	
setInitialServicePoints	void	newInitial	int	
setMaxServicePoints	void	newMax	int	
setMinServicePoints	void	newMin	int	
setOpenTime	void	newOPenTime	int	
setStoreMax	void	newStoreMax	int	

## 7.6 CustomerHandler

---

Name	ID	Description	Inheritance
CustomerHandler	Act-6	Class is responsible for adding customer to the store at the correct time, incrementing the wait times of the customer, and removing customer from the purchasing area.	None

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
previousTime	int	Read/Write	runCustomerHandler

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
runCustomerHandler	void			
incrementWaitTimes	void			
checkPurchasingArea	void			

# 7.7 ManagerHandler

Name	ID	Description	Inheritance
ManagerHandler	Act-7	Class is responsible for alerting the manager when the store should be opened, closed, and when the manager should check its operation efficiency.	None

Unique-ID	Responsibilities	Collaborators	Use Cases
Resp1	Manager is alerted it is time to open the store		UC-02
Resp33	Manager's store monitoring interval expires		UC-02
Resp34	Manager's store monitoring interval resets		UC-02
Resp2	Manager is alerted it is time to close the store		UC-02
Resp4	Manager closes the store	Manager	UC-02, UC-06

## Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

## Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
runManagerHandler	void			
checkStoreOpen	void			Resp1
checkStoreClose	void			Resp2 Resp4
checkStoreStatus	void			Resp33 Resp34



# 7.8 ServicePointHandler

---

Name	ID	Description	Inheritance
ServicePointHandler	Act-8	Class is responsible checking if a customer should be processed by the service point and if the service point queues should be advanced.	None

Unique-ID	Responsibilities	Collaborators	Use Cases

## Data Member Dictionary

Variable Name	Type	Status	Procedures
previousTime	int	Read/Write	runServicePointHandler

## Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
runServicePointHandler	void			
advanceServiceQueues	void			
incrementIdleTime	void			
checkCustomerProcessing	void			

## 7.9 ManageStoreStrategy

---

Name	ID	Description	Inheritance
ManageStoreStrategy	Act-9	This class is responsible for determining how the manager monitors the store.	None

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
monitorStore()	void			

## 7.10 QueueSizeStrategy

---

Name	ID	Description	Inheritance
QueueSizeStrategy	Act-10	This class is responsible for determining how the manager monitors the store. Checks the service point's idle time and queue sizes to determine which to open and close.	ManageStoreStrategy

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
monitorStore()	void			

## 7.11 RatesStrategy

---

Name	ID	Description	Inheritance
RatesStrategy	Act-11	This class is responsible for determining how the manager monitors the store. Checks the arrival rate of the store and service points to determine if service points should be opened or closed.	ManageStoreStrategy

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
monitorStore()	void			

## 7.12 SelectSPStrategy

---

Name	ID	Description	Inheritance
SelectSPStrategy	Act-12	This class is responsible for selecting which service point the customer will enter.	None

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
selectSP()	void			

## 7.13 SelectSPCustsStrategy

---

Name	ID	Description	Inheritance
SelectSPCustsStrategy	Act-13	This class is responsible for selecting which service point the customer will enter. It counts the number of customer in each service point's queue and picks the one with the smallest queue.	SelectSPStrategy

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
selectSP()	void			

## 7.14 SelectSPItemsStrategy

---

Name	ID	Description	Inheritance
SelectSPItemsStrategy	Act-14	This class is responsible for selecting which service point the customer will enter. It counts the number of customer in each service point's queue and picks the one with the smallest queue.	SelectSPStrategy

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
selectSP()	void			

## 7.15 StoreAdditionStrategy

---

Name	ID	Description	Inheritance
StoreAdditionStrategy	Act-15	This class is responsible for determine how customer are added to the store.	None

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
addCustomer()	void			
setUpCustomers()	void			
getArrivalRate()	int			



## 7.16 RateAddition

---

Name	ID	Description	Inheritance
RateAddition	Act-16	This class is responsible for selecting which service point the customer will enter. It counts the number of customer in each service point's queue and picks the one with the smallest queue.	StoreAdditionStrategy

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
addCustomer()	void			
setUpCustomers()	void			
getArrivalRate()	int			

## 7.17 BatchAddition

---

Name	ID	Description	Inheritance
BatchAddition	Act-17	This class is responsible for selecting which service point the customer will enter. It counts the items in each service point queue and picks the one with the least number of items.	StoreAdditionStrategy

Unique-ID	Responsibilities	Collaborators	Use Cases

### Data Member Dictionary

Variable Name	Type	Status	Procedures
There are no Data Members for this Class			

### Procedure Dictionary

Procedure Name	Return Type	Parameter Name	Parameter type	Responsibilities
addCustomer()	void			
setUpCustomers()	void			
getArrivalRate()	int			

# 8. Strategies

## 8.1 Service Point Selection Strategies (SPSS)

---

### 8.1.1 Customer Count Strategy (CCS)

The customer selects which service point to enter based on the number of customers in the service point and its queue. The customer will prioritize an empty service point. If no service points are open the customer will count all the customers in a service point taking into account the processing customer and all the customer's in the queue then select the service point with the smallest number of customer in it.

This strategy would benefit customers in a store that has a long constant processing time (not taking into account time for each item) since less customers would have to be processed before they reach the service point.

### 8.1.2 Item Count Strategy (ICS)

The customer selects their service point to enter based on the total number of items in the service point and its queue. The customer will prioritize an empty service point. If no service points are empty the customer counts the total number of items of the processing customer and all the customers in the queue and selects the service point with the fewest items in it.

This strategy would benefit customers in a store with a long processing time per item.

## 8.2 Store Monitoring Strategy (SMS)

---

### 8.2.1 Arrival Rate Strategy (ARS)

Upon the expiration of the manager's monitoring interval the manager first checks the arrival rate of customer into the store. The store has a set of arrival rate thresholds depending on the current number of open service points. If the arrival rate is above the current threshold for the number of open service points the manager will attempt to open a new service point to accommodate the influx of customers. If the manager opens a new service point based on the store arrival rate they do not perform any other activities until the next monitoring interval.

If the store arrival rate did not warrant a new service point being open the manager goes and checks each service point. Each service point has a threshold for alerting the manager to open a new service point or to close a service point and when checked will do so. After consulting all the service points the manager takes

the majority ruling and will attempt to open or close a new service point. A manager may only open or close one service point per monitoring interval as to avoid over reaction to a situation.

### **8.2.2 Queue Size and Idle Time Strategy (QSS)**

Upon the expiration of the manager's monitoring interval the manager goes and checks the queue sizes of each of the service points. The service points will alert the manager to open a new service point if their queue size has reached the maximum. Next the manager will go and check the idle time for each of the service points. The service points have a maximum idle time that if surpassed will alert the manager to close a service point. The manager takes the majority position and will possible attempt to open a close a service point depending on the queue sizes and idle time of the service points.

# 9. Testing

## 9.1 Testing Explanation

---

Most test cases will have a log file that has the logged output when the test is run. It has a batch file that was used in batch mode to generate the log. It lists any traceability to required tests. There is an explanation of what was being tested and specific things that should be noted.

## 9.2 Testing Expansion

---

### 9.2.1 Simulation Configuration

**Log File:** N/A

**Batch File:** N/A

**Requirement Tractability:**

Req 1 (simulation configuration (ie what is hardwired vs what can be input))  
WebCT announcement April 3, 2012 16:33

Soft-Coded (Can be changed during run time)

Configurations that can be changed while the simulation is running:

- Arrival rate<sup>1</sup>
- Shopping time per item
- Processing time per item
- Simulated second
- Add customers
- Manager strategy
- Customer strategy<sup>2</sup>

<sup>1</sup>Arrival rate can only be directly changed in Arrival Rate Mode

<sup>2</sup>Customer strategy can only be changed if the customer is added via batch mode or through the explicitly pre-prepared customer button

Configurations that can be changed during run time but before the simulation starts to run:

- Open time
- Close time
- Maximum/Minimum/Initial number of service points
- Size of service point queue
- Store capacity
- Store addition strategy
- Customer addition

Hard coded (Cannot be changed via file or graphical interface)

- Interval which manager monitors store efficiency
- Store arrival rate/service point arrival rate thresholds
- Max service point idle time

## **9.2.2 All Customer Creation Options**

**Log File:** N/A

**Batch File:** N/A

**Requirement Tractability:**

Req 2 (all customer creation options)

WebCT announcement April 3, 2012 16:33

### **9.2.2.1 Batch addition from file (Batch Mode Only)**

In the selected batch text file the user may specify what time any number of customers enters, how many items they purchase, and which service point selection strategy they will use.

### **9.2.2.2 Arrival rate (Arrival Rate Mode Only)**

User can select a desired arrival rate and simulation will try to add customers at an interval to meet this rate. Customers will be assigned a random number of items.

### **9.2.2.3 Add now option (Both Modes)**

A customer will immediately be added to the store with a random number of items to buy.

### **9.2.2.4 Explicitly pre-prepared customer addition (Both Modes)**

User selects the time the customer should be added at, how many items they should purchase, and which service point selection strategy they will use.

### **9.2.2.5 Limitations**

If the store capacity is reached no customers can be added until a customer leaves the store. No addition method can over-ride this.

## **9.2.3 Few Customers Run**

**Log File:** 923FewCustomersLog.txt

**Batch File:** 923FewCustomersBatch.txt

### **Requirement Tractability:**

Req 3 (opening and closing of store with a few customers using directly the cashier queues)  
WebCT announcement April 3, 2012 16:33

### **Explanation/Notes:**

When the store is open at 8:00:00 three customers enter with 1, 2, and 3 items respectively.

At 8:01:00 the first customer is done gathering their items, selects service point 1 and starts being processed.

At 8:01:40 the first customer is finishing being processed and leaves the store with a wait time of 0 seconds since they did not have to wait in the wait queue or a service point queue.

Similarly customer 2 and 3 enters service point 1, is processed and leaves the store with a wait time of 0 seconds.

At 8:05:00 the store closes immediately since there are no customers in the store.

## **9.2.4 Creation and Use of the Wait Queue**

**Log File:** 924WaitQueueLog.txt

**Batch File:** 924WaitQueueBatch.txt

### **Requirement Tractability:**

Req 4 (Creation and use of the wait queue)  
WebCT announcement April 3, 2012 16:33

### **Explanation/Notes:**

At 8:00:01 9 customers are added to the store. The first customer purchases 1 item, the second 2 items, and the remaining 3 items.

The first customer two customers are processed by the service point with a wait time of 0 seconds.

Customers 3 through 9 however all finishing gathering their purchases at the same time (8:03:01). Only 1 service point is open and it can hold one customer to be processed (Customer 3) and 3 customers in its queue (Customers 4-6) leaving Customers 7-9 to wait in the wait queue.

Customer 4 leaves the store at 8:04:01 at which point Customer 5 starts getting processed by service point 1 and Customer 7 enters service point 1's queue.  
Continue until all the customers are processed and the store closes at 10:00:00

### **9.2.5A Service Point Selection Strategy: Customer Count**

**Log File:** 925ACustSelectionLog.txt

**Batch File:** 925ACustSelectionBatch.txt

#### **Requirement Tractability:**

Req 5 (each customer selection strategy)  
WebCT announcement April 3, 2012 16:33

#### **Explanation/Notes:**

When Customer 1 finishing gathering their purchasing they select service point 1 since it is the first empty service point. Similarly Customer 3 and 2 select service points 2 and 3 respectively.  
Customer 4 selections service point 1 since all three service points are the same size. Customer 5 selects service point 2 since service point 1 has a capacity of 2, and Customer 6 selects service point 2 since it is the least full. Customers 7-9 continue to select service points 1 to 3 in that order.  
At 8:03:30 Customer 3 in service point 2 is finished being processed and leaves the store so now service point 2 has 1 less customer than service points 1 and 3. At 8:03:30 Customer 10 would have selected service point 1 but since service point 2 has less customer in it Customer 10 selects it instead.

### **9.2.5B Service Point Selection Strategy: Item Count**

**Log File:** 925BItemSelectionLog.txt

**Batch File:** 925BItemSelectionBatch.txt

#### **Requirement Tractability:**

Req 5 (each customer selection strategy)  
WebCT announcement April 3, 2012 16:33

#### **Explanation/Notes:**

At 8:00:15 the first customer enters trying to purchase 18 items and eventually enters service point 1.  
At 8:01:30 the second customer enters trying to purchase 10 items and eventually enters service point 2.  
At 8:03:10 the third customer enters trying to purchase 1 item and enters service point 3 since it is the only empty service point.  
When the next 4 customers go to select a service point they all select service point 3 since it still is the service point with the fewest items. After Customer 5 enters service point 3 at 8:04:30 service point 1 now



has a total of 11 items in it so when Customer 8 goes to select a service point they select service point 2 since it only has a total of 10 items in it.

## 9.2.6 Simultaneous Customers

**Log File:** 926SimAddLeaveLog.txt

**Batch File:** 926SimAddLeaveBatch.txt

### **Requirement Tractability:**

Req 12 (customers simultaneously leaving the store)

Req 13 (customers simultaneously entering AND leaving the store)

WebCT announcement April 3, 2012 16:33

### **Explanation/Notes:**

At 8:00:45 Customers 1,2,3 enter the store simultaneously.

At 8:02:45 Customers 1,2,3 simultaneously leave the store AND Customers 4,5,6 simultaneously enter the store.

## 9.2.7 Full Service Points

**Log File:** 927FullStoreLog.txt

**Batch File:** 927FullStoreBatch.txt

### **Requirement Tractability:**

Req 14 (current cashier queues full and customers in the waiting queue (can open another cashier queue))

Req 15 (ALL cashier queues of store full and customers in the waiting queue)

WebCT announcement April 3, 2012 16:33

### **Explanation/Notes:**

Initially 3 service points are open.

At 8:07:10 all three service points are full so Customer 13 remains in the wait queue.

At 8:08:00 the manager's monitoring interval expires and since all the service point queues are full he opens a new service point 4<sup>th</sup> service point queue.

At 8:08:30 all 4 queues are full and since only 4 queues can be opened the customers then again start queuing up in the wait queue.

## 9.2.8A Manager Opens/Closes Service Points: Arrival Rate Strategy

**Log File:** 928AManagerRatesLog.txt

**Batch File:** 928AManagerRatesBatch.txt

### Requirement Tractability:

Req 6 (each strategy of the manager for opening one cashier queue)

Req 7 (each strategy of the manager for opening several cashier queues over time)

Req 8 (each strategy of the manager for opening several cashier queues simultaneously) <sup>1</sup>

Req 9 (each strategy of the manager for closing one cashier queue)

Req 10 (each strategy of the manager for closing several cashier queues over time) <sup>1</sup>

Req 11 (each strategy of the manager for closing several cashier queues simultaneously)

WebCT announcement April 3, 2012 16:33

<sup>1</sup> It was decided that it should NOT be possible for the manager to open or close multiple service point during one monitoring arrival interval since it could lead to over reaction to a momentary drop or raise in customers.

### Explanation/Notes:

The manager checks the store and service point arrival rates every 4 minutes starting at 8:04:00

Initially 1 service point is opened

8:04:00 – Manager monitoring interval expires Neither the store arrival rate or the service point arrival rate were above the threshold and only the minimum number of service points were open so the manager cannot open or close any service points

8:08:00 – Manager monitoring interval expires

The store arrival rate was above the threshold for 1 open service point so the manager opens service points 2

8:12:00 – Manager monitoring interval expires

Again the store arrival rate was above the threshold for 2 open service points this time so the manager opens service point 3

8:16:00 – Manager monitoring interval expires

The store arrival rate was 0 so the manager did not open the a service point based off it

Service points 1 had an arrival rate above its threshold and no other service points had arrival rate below their thresholds so the manager opened service point 4.

8:20:00 – Manager monitoring interval expires

While service point 1 was above it's arrival rate threshold service points 2 to 4 were below so the manager closed service point 4.

8:24:00 – Manager monitoring interval expires

Again more service points had arrival rates below their thresholds than above so the manager closed service point 3

## 9.2.8B Manager Opens/Closes Service Points: Queue Size/Idle Time Strategy

**Log File:** 928BManagerQueueLog.txt

**Batch File:** 928BManagerQueueBatch.txt

### **Requirement Tractability:**

Req 6 (each strategy of the manager for opening one cashier queue)

Req 7 (each strategy of the manager for opening several cashier queues over time)

Req 8 (each strategy of the manager for opening several cashier queues simultaneously)<sup>1</sup>

Req 9 (each strategy of the manager for closing one cashier queue)

Req 10 (each strategy of the manager for closing several cashier queues over time)<sup>1</sup>

Req 11 (each strategy of the manager for closing several cashier queues simultaneously)

WebCT announcement April 3, 2012 16:33

<sup>1</sup> It was decided that it should NOT be possible for the manager to open or close multiple service point during one monitoring arrival interval since it could lead to over reaction to a momentary drop or raise in customers.

### **Explanation/Notes:**

The manager checks the store and service point arrival rates every 4 minutes starting at 8:04:00

Initially 2 service points is opened.

8:04:00 – Manager monitoring interval expires

Service point 1's idle time was too large so it is closed.

8:08:00 – Manager monitoring interval expires

Service point 2 had a max queue size and no idle time so service point 1 is reopened

8:12:00 – Manager monitoring interval expires

Both service point's queue's are full and neither have an idle time so service point 3 is opened

8:16:00 – Manager monitoring interval expires

Again the service points queue's are full so service point 4 is opened

8:20:00 – Manager monitoring interval expires

The queues are not full but the idle times are not large enough to close a service point – no change

8:24:00 – Manager monitoring interval expires

The service points have reached their max idle time so service point 1 is closed

8:28:00 – Manager monitoring interval expires

Again the service points have reached their max idle time so service point 2 is closed

8:32:00 – Manager monitoring interval expires

Service point 3 is closed due to hitting its max idle time

## 9.2.9 Customer Wait Time Upon Leaving the Store and Averages Upon Store Closing

**Log File:** 929WaitTimesLog.txt

**Batch File:** 929WaitTimesBatch.txt

### **Requirement Tractability:**

N/A

### **Explanation/Notes:**

At 8:05:00 Customer 1 leaves the store with 0 second wait time since they did not wait in the wait queue or service point queue

At 8:07:30 Customer 2 leaves the store with a 150 second wait time since they waited in the service point for Customer 2 to be processed

At 8:10:00 Customer 3 leaves the store with a 300 second wait time since they waited in the service point for Customer 1 and 2 to be processed

At 8:12:30 Customer 4 leaves the store with a 450 second wait time since they waited in the wait queue for Customer 1 to be processed then waited in the service point queue for Customers 2, and 3 to be processed

At 8:15:00 Customer 5 leaves the store with a 600 second wait time from waiting in the wait queue for Customers 1 and 2 to be processed and waiting in the service point queue for Customers 3 and 4 to be processed

When the store closed the average customer wait time was 300 seconds (1500 total / 5 customers)

## 9.2.10 Service Point Idle Times Upon Closing and Average Upon Store Closing

**Log File:** 929IdleLog.txt

**Batch File:** 929IdleBatch.txt

### **Requirement Tractability:**

N/A

### **Explanation/Notes:**

At 8:08:00 service point 1 closes with a sessional idle time of 300 seconds.

At 8:12:00 service point 1 is reopened

At 8:15:00 the manager closes the store and both the service points

Service point 1 has a second sessional idle time of 60 seconds

Service point 2 closed with a sessional idle time of 360 seconds.

The average service point idle time was 360 seconds  $((300 + 30 + 360) / 2 \text{ service points} = 360 \text{ seconds})$

## 9.2.11 Manager Dynamically Changing Monitoring Strategies

**Log File:** 9211ManagerChangeLog.txt

**Batch File:** N/A

### **Requirement Tractability:**

Bonus

WebCT announcement April 3, 2012 16:33

### **Explanation/Notes:**

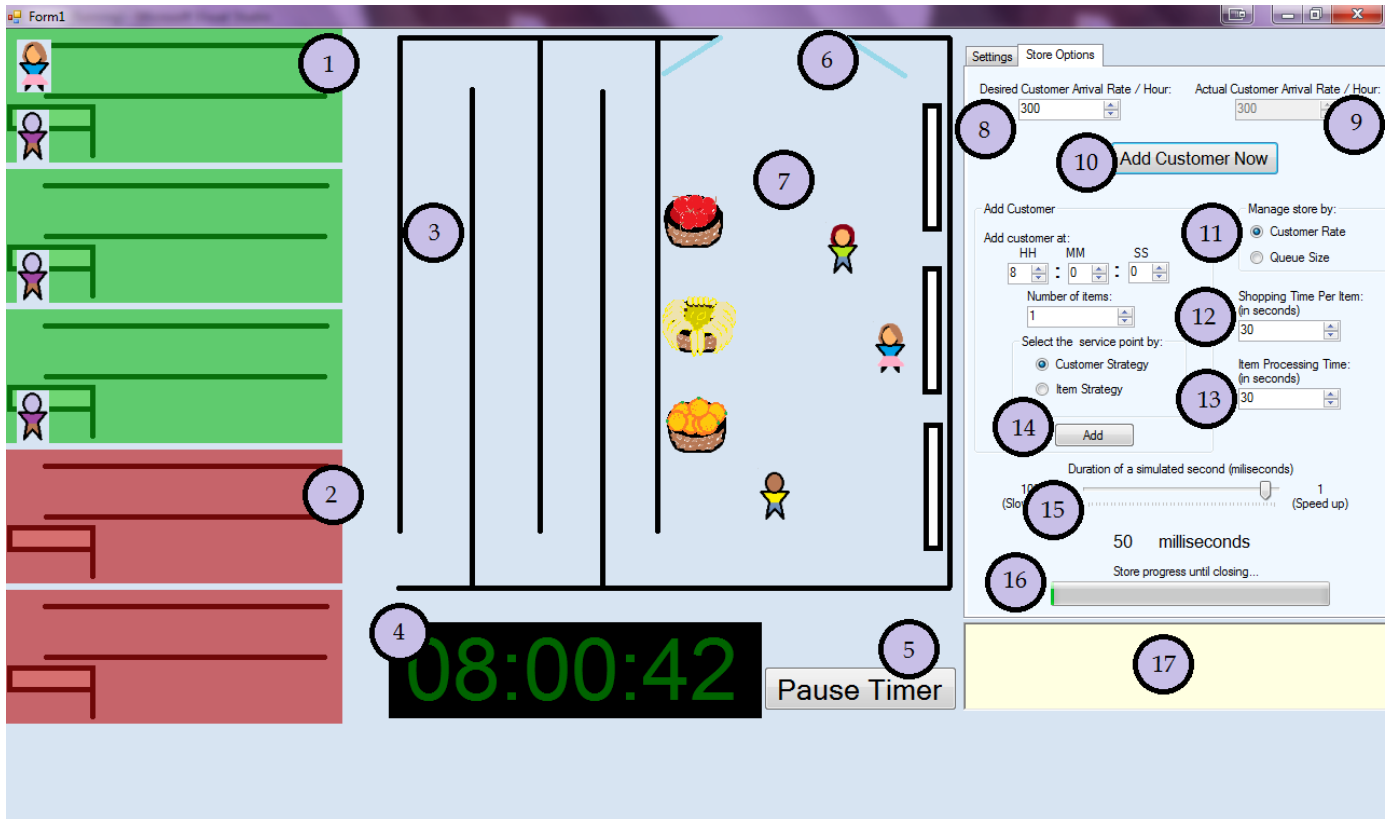
Ran the system in Addition Rate Mode. Opened the store at 8:00:00, closed the store at 8:10:00. Store started with the Customer Arrival Rate Strategy but at 8:05:00 manually switch to the Queue Size strategy.

At 8:04:00 the manger monitors the store using the arrival rate strategy

At 8:08:00 the manager monitors the store using the queue size and idle time strategy indicating their strategy has dynamically changed.

# 10. Additional Information

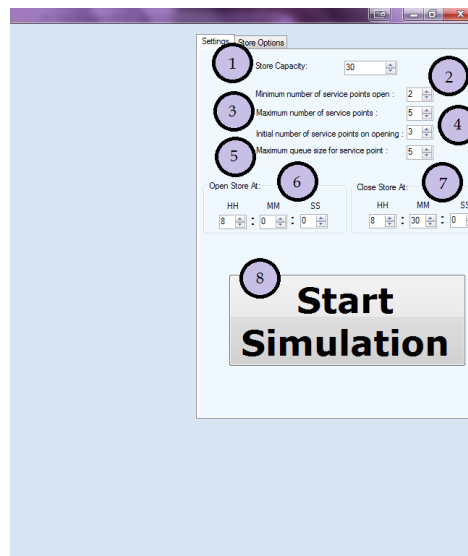
## 10.1 Graphics User Information Explanation



\*Not displayed: Yellow service point (Service point closed service queue and will no longer accept customers)

Number	Feature
1	Open service point processing a customer
2	Closed service point
3	Wait queue system
4	Timer
5	Button to pause then resume time
6	Open doors indicating store is open
7	Purchasing area
8	Desired rate at which customers will be entering the store per hour

9	The actual rate at which customers are entering the store
10	Add a customer immediately to the store
11	Change the manager's strategy pattern for monitoring the store
12	The time it takes for a customer to gather a single item
13	The time it takes for a service point to process an individual item
14	Add a customer at a specific desired time
15	Speed up or slow down time interval in the simulation
16	Store simulation progress until completion (store closes)
17	Log to notify about pertinent events regarding customers and service points



Number	Feature
1	The maximum number of customers able to enter the store
2	Minimum number of service points allowed open
3	Maximum number of service points in the store
4	Initial number of service points opened on store opening
5	The maximum queue size for a service point
6	What time will the store open at (when will simulation begin)

## 10.2 Log Information

---

The log file can be found in the following location: .....\\Team2\_It2\\Team2\_It2\\bin\\Debug

The log appends the current store run after the previous run so it is possible to back track to previous runs.

## 10.3 Implemented Design Patterns

---

### Strategy Pattern

Strategy pattern was implemented in 3 occurrences. The customer's service point selection strategy (customer count vs. item count), the store manager's store monitoring strategy (arrival rate vs. queue size/idle time), and the strategy for how customer should be added to the store (batch vs. arrival rate).

### Singleton Pattern

The store's manager was implemented using the singleton pattern.

### Façade Pattern

All of the graphical interfaces are linked to the store functionality and vice-versa through the Façade class.

## 10.4 Creating Batch Files

---

Batch files are text files that can be opened in batch mode that set up most of the parameters of the simulation. See [SampleBatchFile.txt](#) as a reference.

There can be no extra spaces or empty lines at the end of the batch file.