ICS 372: Group Project 1, Iteration 1 Documentation

Business Rules

Rules for the theater system

Rule number	Rule
Rule 1	To watch a show, a person must be a member.
Rule 2	To be a member, a person must have at least one valid credit card, with a number and expiry date.
Rule 3	A customer may have multiple credit cards.
Rule 4	No two customers may have the same credit card.
Rule 5	The billing address of a credit card is assumed to be the same as that of the customer's address.
Rule 6	If a customer is removed from the member catalog, all credit cards related to the customer are also deleted.
Rule 7	If a show is scheduled for the current or a future date for a client, the client cannot be removed from the client catalog.

Use Cases

0. Exit the Application. Store the data on disk and quit the application.

Use case **Exit the Application**

Actions performed by the actor	Responses from the system
1. The clerk closes the application.	
	2.The system saves all the application data to permanent storage and closes gracefully.

1. Add Client. The system accepts the name, address, and phone number of the client. The system generates a unique id and sets the balance to 0. (The balance will remain 0 in this iteration.)

Use case **Add Client**

Actions performed by the actor	Responses from the system
1. The client fills out an application form in order to	

have their show at the theater, supplying their name, address, and phone number, and hands this form to the theater clerk.	
2. The clerk issues a request to add a new client.	
	3. The system asks for the name, address, and phone number of the new client.
4. The clerk enters the name, address, and phone number of the new client.	
	5. The system attempts to enter the information in the client catalog, and if valid, generates a unique id for the client, and sets the balance due to the client to 0. The system then informs the clerk that it was successful in adding the client.

2. Remove Client. Remove a client with the given id. If a show is scheduled for the current or a future date for this client, the client cannot be removed.

Use case **Remove Client**

Actions performed by the actor	Responses from the system
1. The clerk identifies the client to be deleted from the client catalog.	
2. The clerk issues a request to delete said client.	
	3. The system asks for the id of the client that is to be deleted.
4. The clerk enters the id for the client that is to be deleted.	
	5. The system checks if the client can be removed using Rule 7. If the client can be removed, the system marks the client as no longer being in the client catalog. The system then informs the clerk that it was successful in removing the client.

3. List all Clients. Print information about every client.

Use case **List all Clients**

Actions performed by the actor	Responses from the system
1.The clerk issues a request to display all the information of the clients to the system.	
	2. The system checks for the clients and displays

	information about everyone of them.

4. Add Customer. The system accepts the name, address, phone number, and the number and expiry date of exactly one credit card. The system generates a unique id for the customer.

Use case **Add Customer**

Actions performed by the actor	Responses from the system
1. The customer fills out an application form in order to become a member, supplying their name, address, phone number, and the number and expiry date of exactly one credit card, and hands this form to the theater clerk.	
2. The clerk issues a request to add a new member.	
	3. The system asks for the name, address, phone number, and the number and expiry date of the credit card of the new member.
4. The clerk enters the name, address, phone number, and the number and expiry date of the credit card of the new member.	
	5. The system attempts to enter the information in the customer catalog, and if valid, and generates a unique id for the new member. The system then informs the clerk it was successful in adding the customer.

5. Remove Customer. Remove a customer with the given id. All credit cards related to the customer are also deleted.

Use case **Remove Customer**

Actions performed by the actor	Responses from the system
1. The clerk identifies the customer to be deleted from the member catalog.	
2. The clerk issues a request to delete said member.	
	3. The system asks for the identifier of the member that is to be deleted.
4. The clerk enters the id for the member that is to be deleted.	
	5. The system checks if the member can be removed.

	If the member can be removed, the theater system marks the member as no longer being in the theater member catalog. Any associated credit cards are also deleted. The system then informs the clerk that it was successful in removing the customer.

6. Add a Credit Card. The system accepts the customer id, credit card number, and expiry date and remembers that the credit card belongs to this customer.

Use case **Add a Credit Card**

Actions performed by the actor	Responses from the system
1. The customer fills out a form in order to add a new credit card, supplying the card number and expiry date. Then hands the form to the theater clerk.	
2. The clerk identifies the customer and requests to add a new credit card.	
	3. The system asks for the identifier, card number and expiry date of the new credit card.
4. The clerk enters the id, card number, and expiration date of the new credit card to be added.	
	5. The system attempts to enter the information, and if valid, the credit card information is stored to said customer's file. The system then informs the clerk that it was successful in adding the credit card.

7. Remove a Credit Card. The system accepts the credit card number and removes the information related to the credit card. If this is the only credit card for the customer, it refuses to delete the credit card information.

Use case **Remove a Credit Card**

Actions performed by the actor	Responses from the system
1. The clerk identifies the credit card to be deleted from a customer.	
2. The clerk issues a request to delete said credit card.	
	3. The system asks for the customer number and

	credit card number for deletion.
4. The clerk enters the customer number and credit card number for the card to be deleted.	
	5. The system checks if the customer has more than one credit card on file. If this is the only credit card for the customer, it refuses to delete the credit card information. The system then informs the clerk about the success of the deletion or the error.

8. List all Customers. Print information about every client, including credit card information.

Use case **List all Customers**

Actions performed by the actor	Responses from the system
1.The clerk issues a request to display all the information of the customers to the system.	
	2. The system checks for the customers and displays information about each one of them.

9. Add a Show/Play. Add a new show for a client. The values accepted are the name of the show, the client id, and the period for which the client wants the theater for this play. The entire range of dates should be available, or the process fails.

Use case **Add a Show/Play**

Actions performed by the actor	Responses from the system
1. The client fills out an application form in order to add a new show, supplying the name of the show, and the period they want the theater for the play. Then hands the form to the clerk.	
2. The clerk identifies the client and requests to add a new show.	
	3. The system asks for the client id, name of the show, and duration of time needed for the new show to be added.
4. The clerk enters the client id, name of the show, and duration of time need for the new show.	
	5. The system verifies the information given is correct. If the range of dates are available, the system adds the new show to the catalog. If they are not the

	process fails. The system informs the clerk whether or not the process is successful.

10. List all Shows. List the names and dates of all shows.

Use case **List all Shows**

Actions performed by the actor	Responses from the system
1.The clerk issues a request to display all the information of the shows to the system.	
	2. The system checks for the shows and displays information about each one of them.

11. Store Data. Store all data related to the theater (everything, including customers, shows, clients, etc.) on disk.

Use case **Store Data**

Actions performed by the actor	Responses from the system
1. The clerk issues a request to save all application data to permanent storage.	
	2. The system tries to save all application data to permanent storage, and responds with whether or not it was successful.

12. Retrieve Data. Retrieve all information related to the theater. This may be done at the start of any session. If stored data is found, the user has the option to use it. The user may also invoke a command to load data, provided he/she has not yet issued any data-related commands in that session.

Use case **Retrieve Data**

Actions performed by the actor	Responses from the system
	1. If the application was just started and saved data exists, the system prompts the clerk if they would like to load it.
2. The clerk issues a request to retrieve all application data from permanent storage, either in response to (1.) or at any point prior to loading or saving data.	

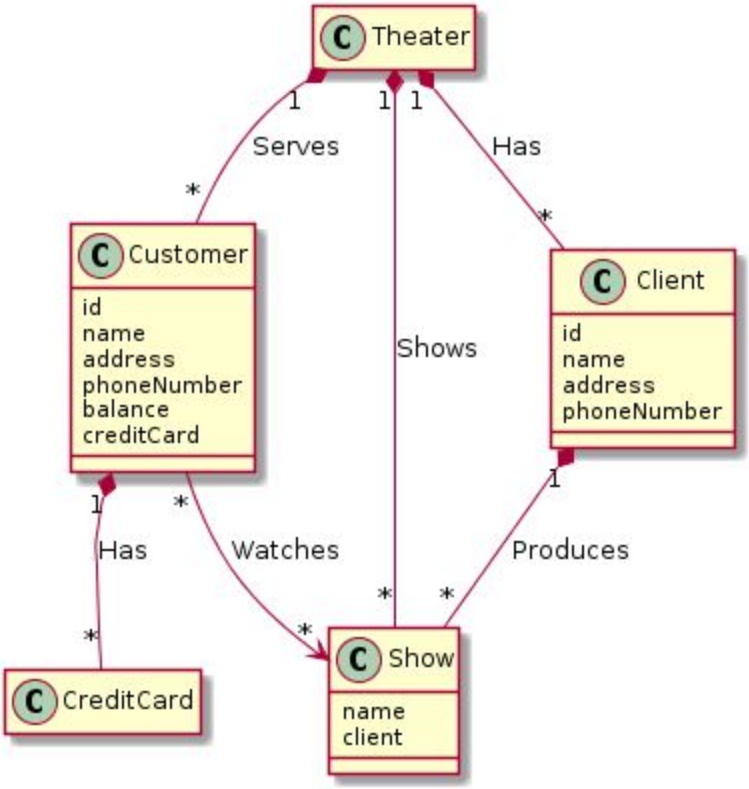
	3. The system tries to save all application data to permanent storage, and responds with whether or not it was successful.

13. Help. This should display all commands.

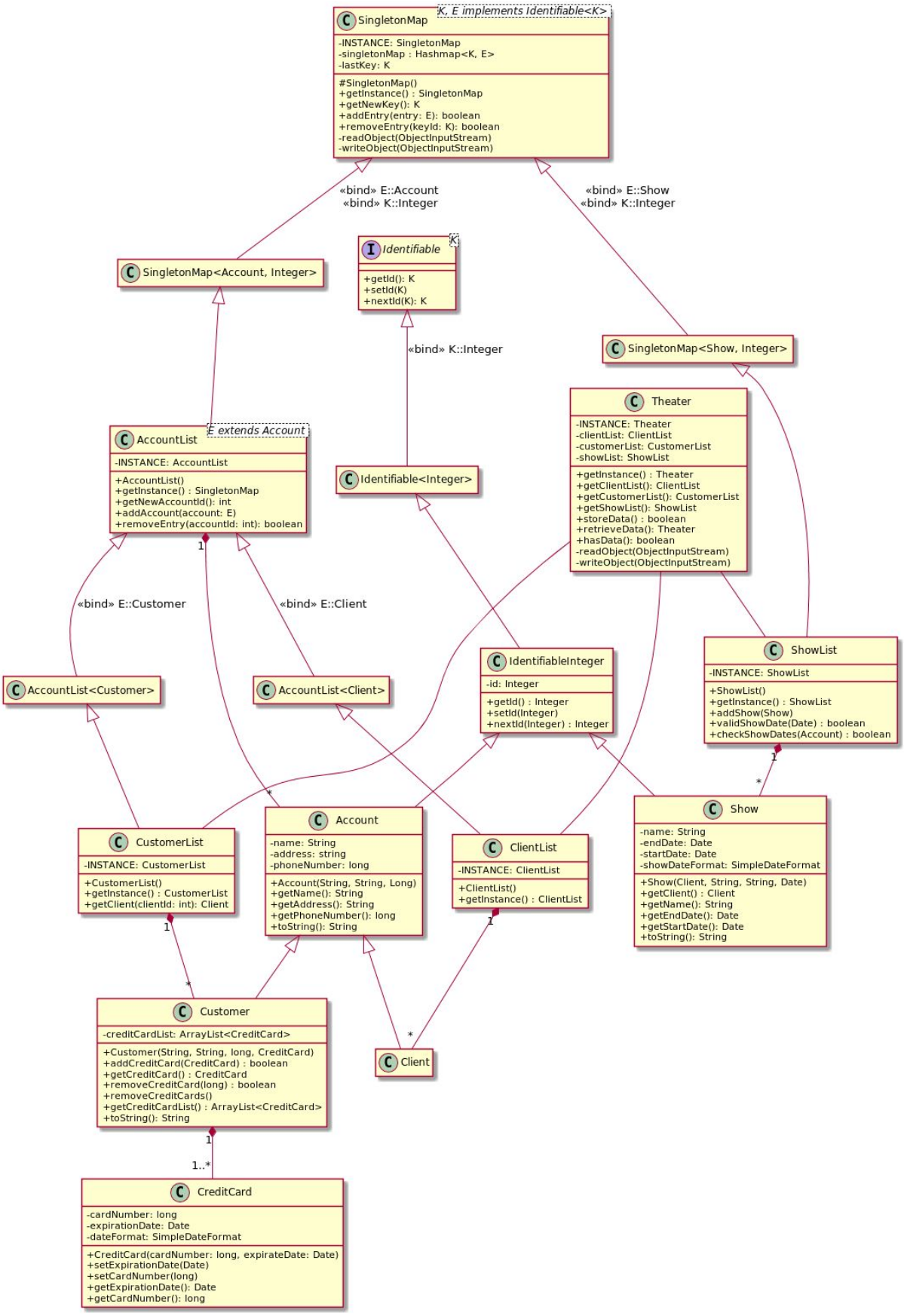
Use case **Help**

Actions performed by the actor	Responses from the system
1. The clerk requests for a list of all the available commands they can perform on the theater system.	
	2. The system echoes back a list of the thirteen available commands.

Conceptual Class Diagram

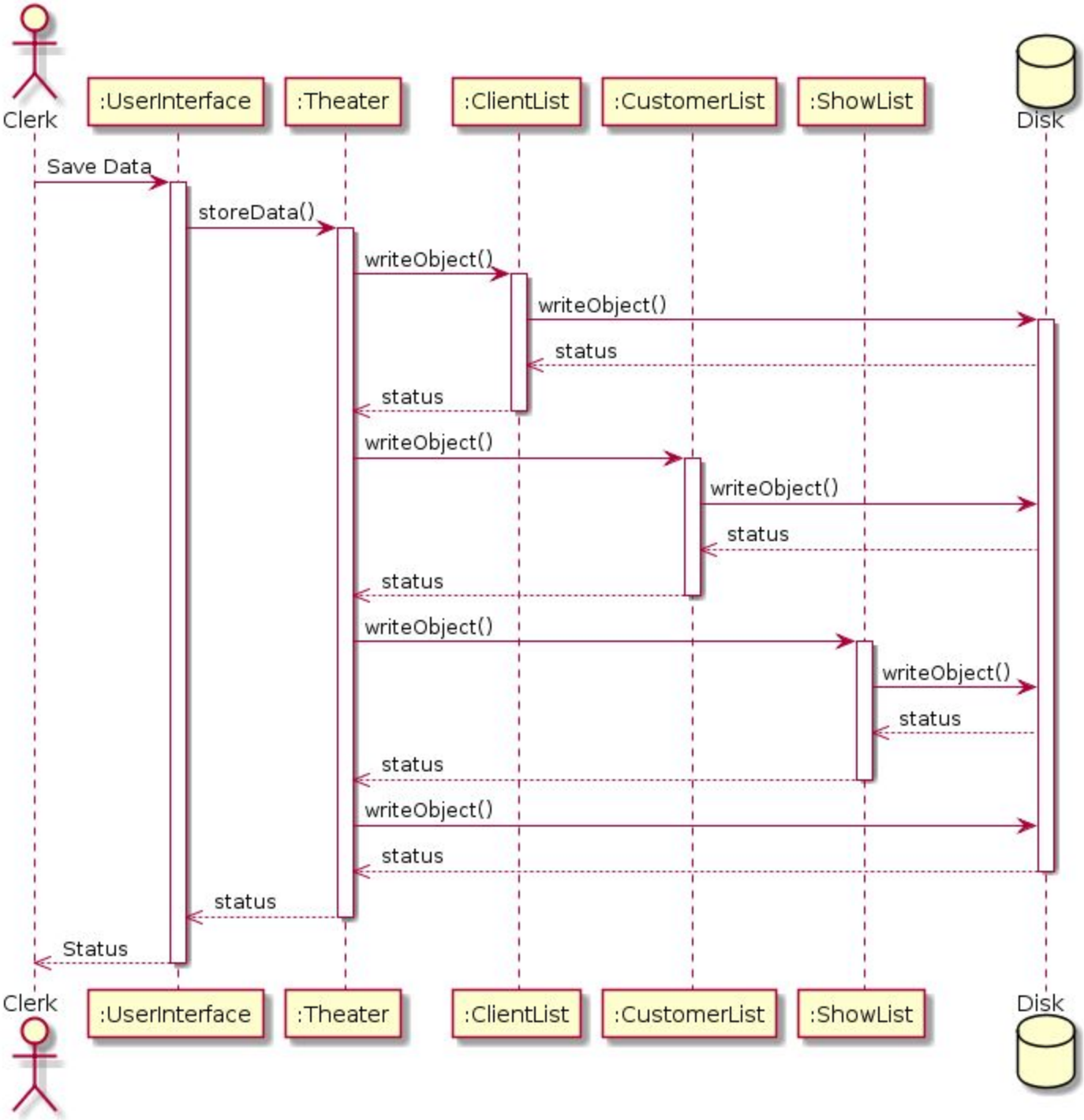


Class Diagram

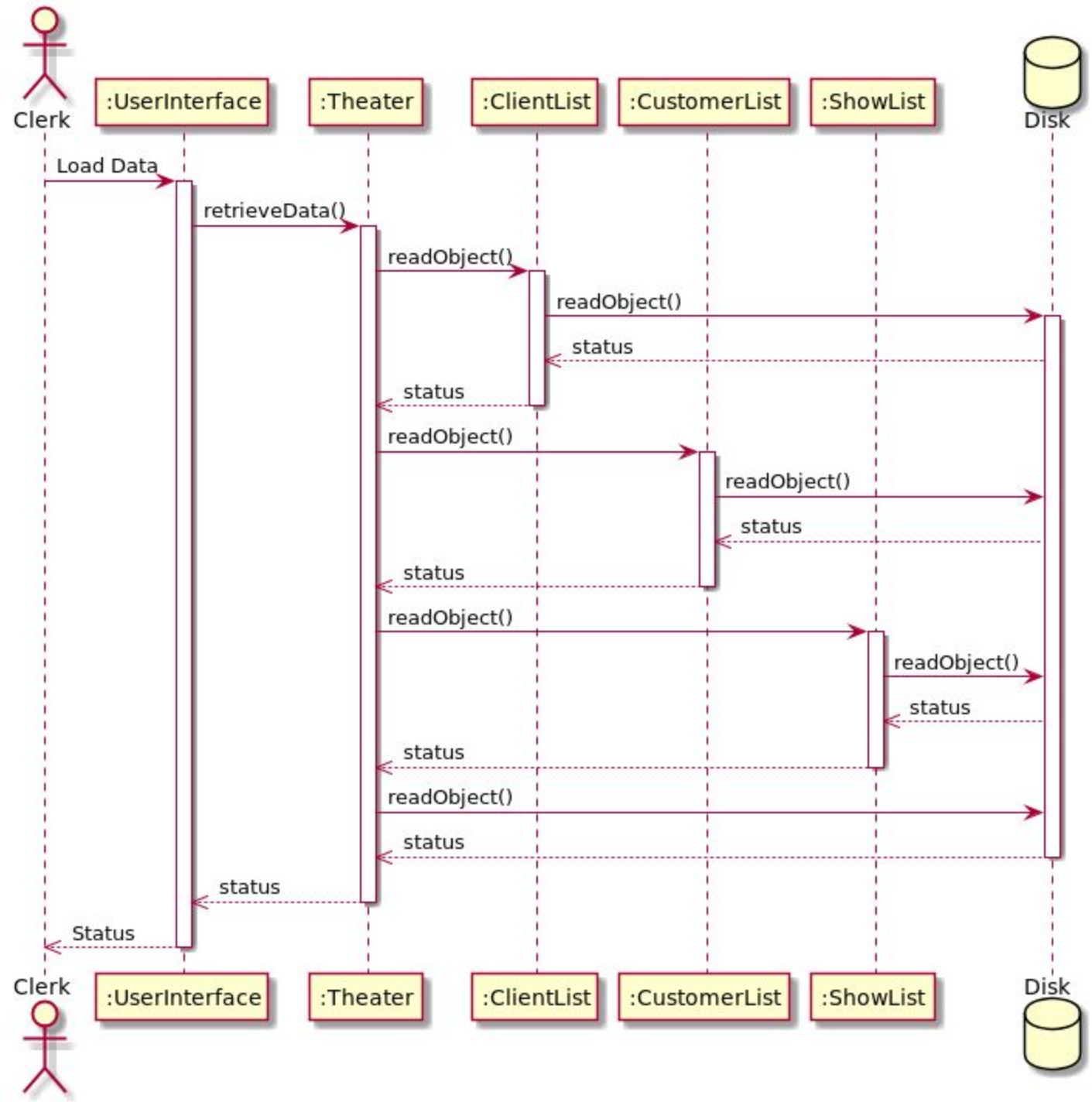


Sequence Diagrams

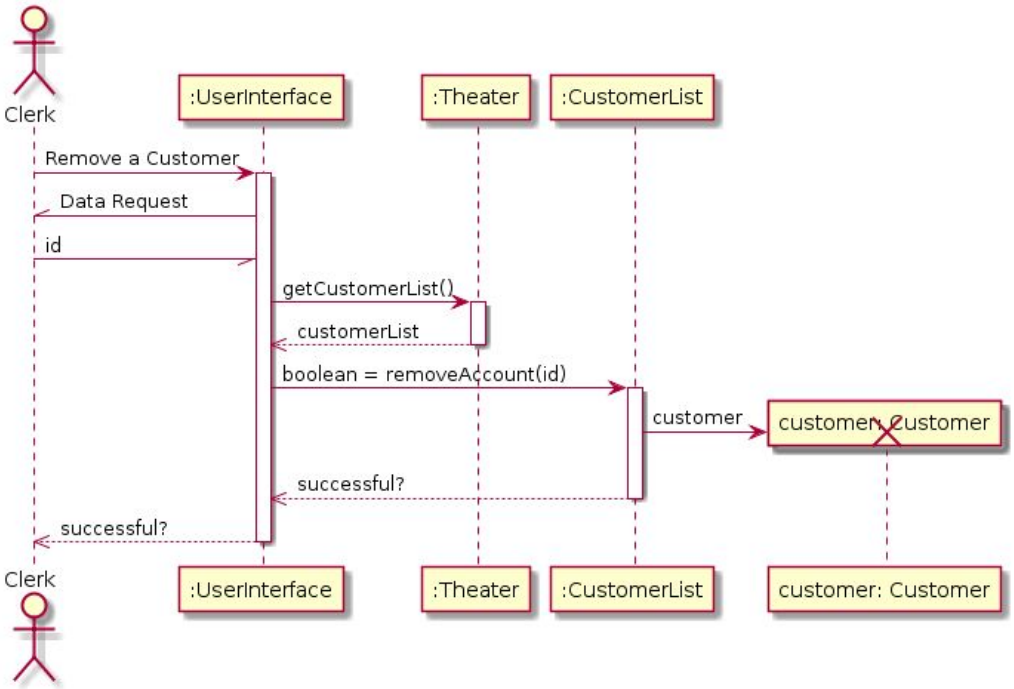
Save State



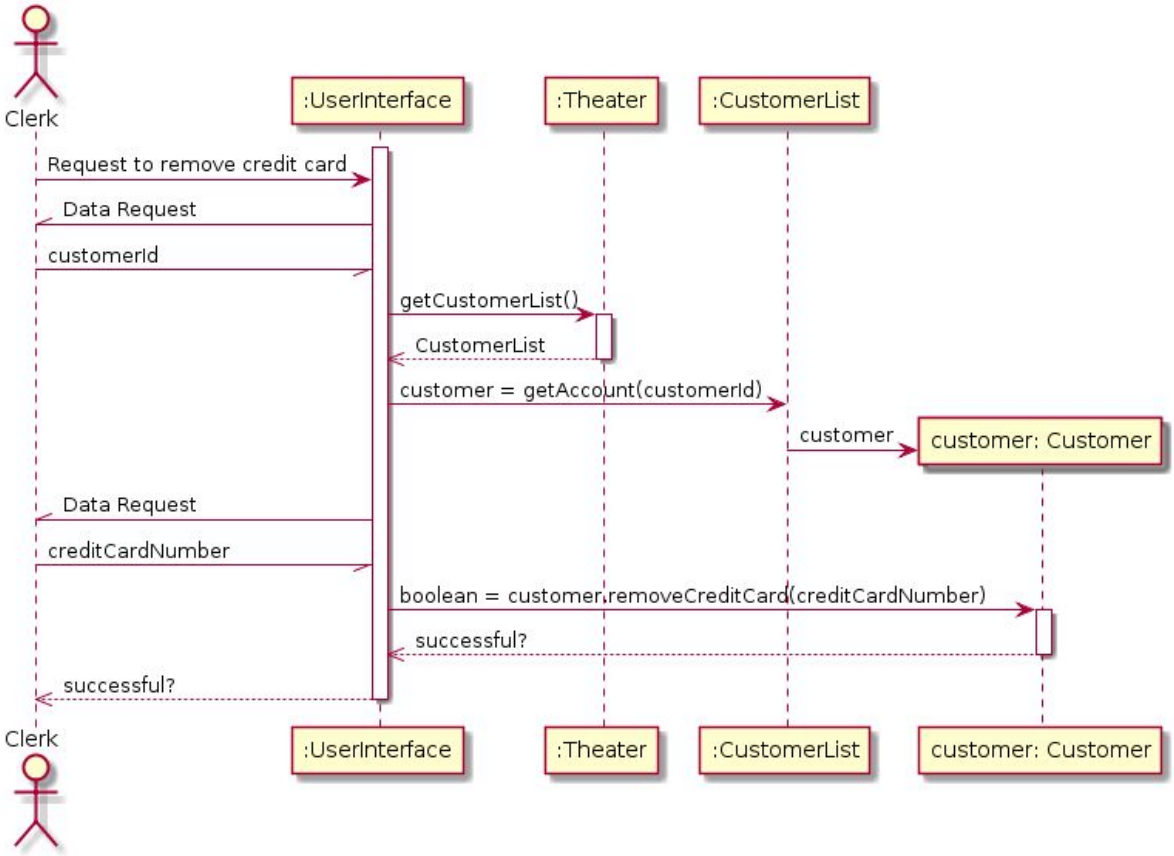
Load State


Clerk
:UserInterface
:Theater
:ClientList
:CustomerList
:ShowList
Disk
Load Data
retrieveData()
readObject()
readObject()
status
status
readObject()
readObject()
status
status
readObject()
readObject()
status
readObject()
status
status
Status
Clerk
:UserInterface
:Theater
:ClientList
:CustomerList
:ShowList
Disk

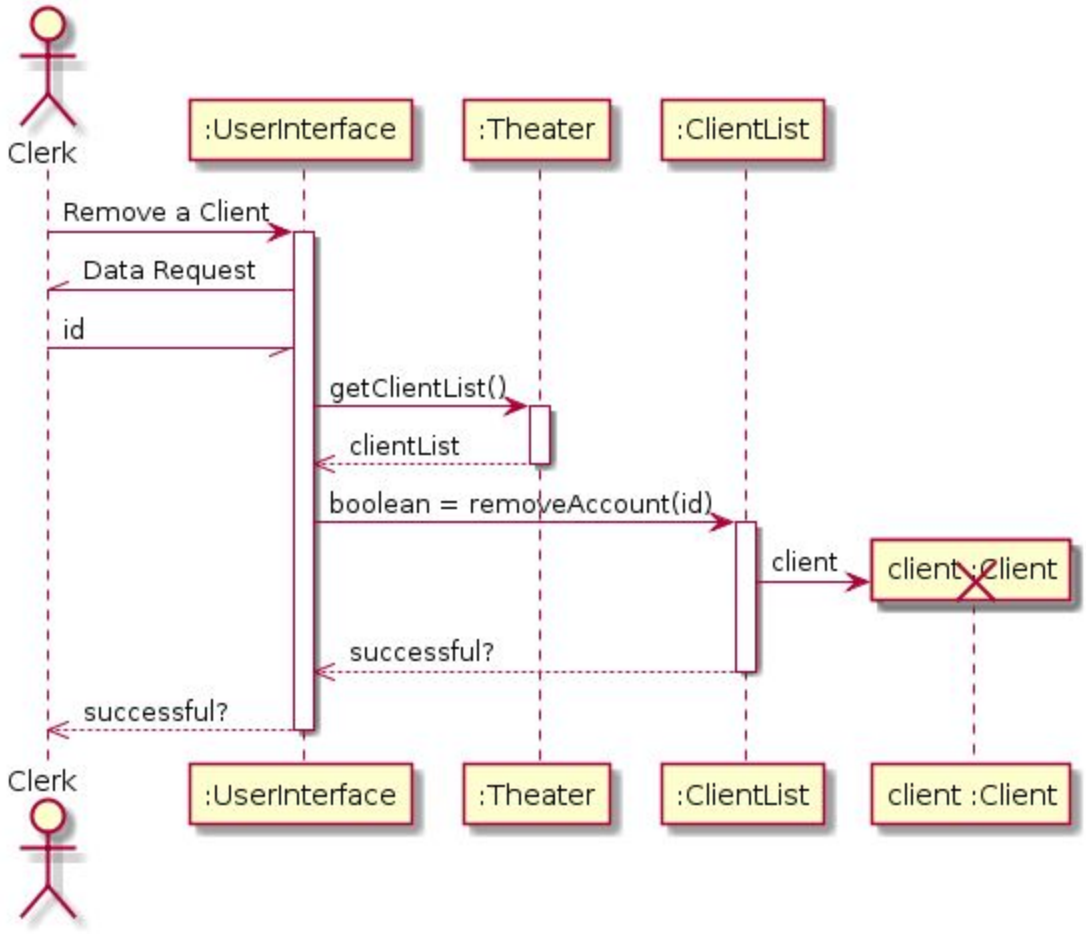
Remove A Customer



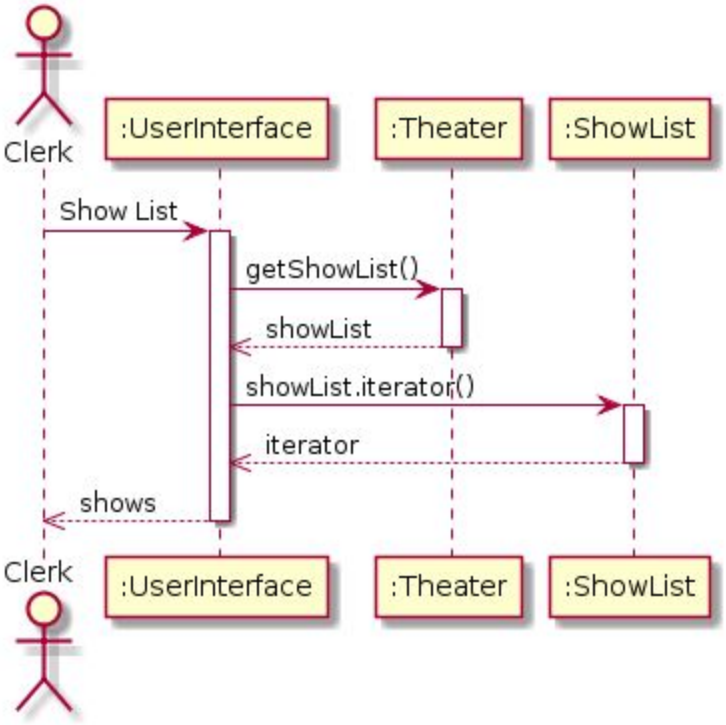
Remove a Credit Card



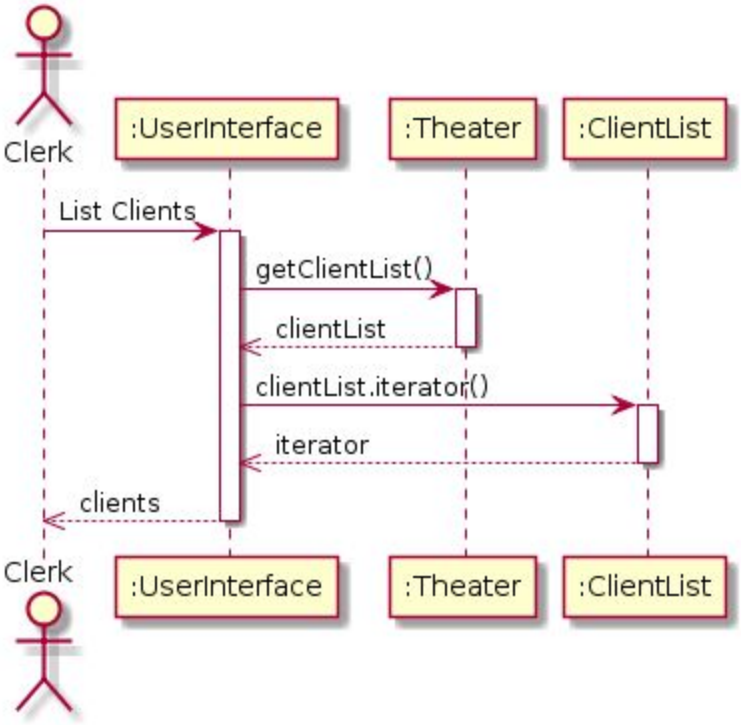
Remove a Client



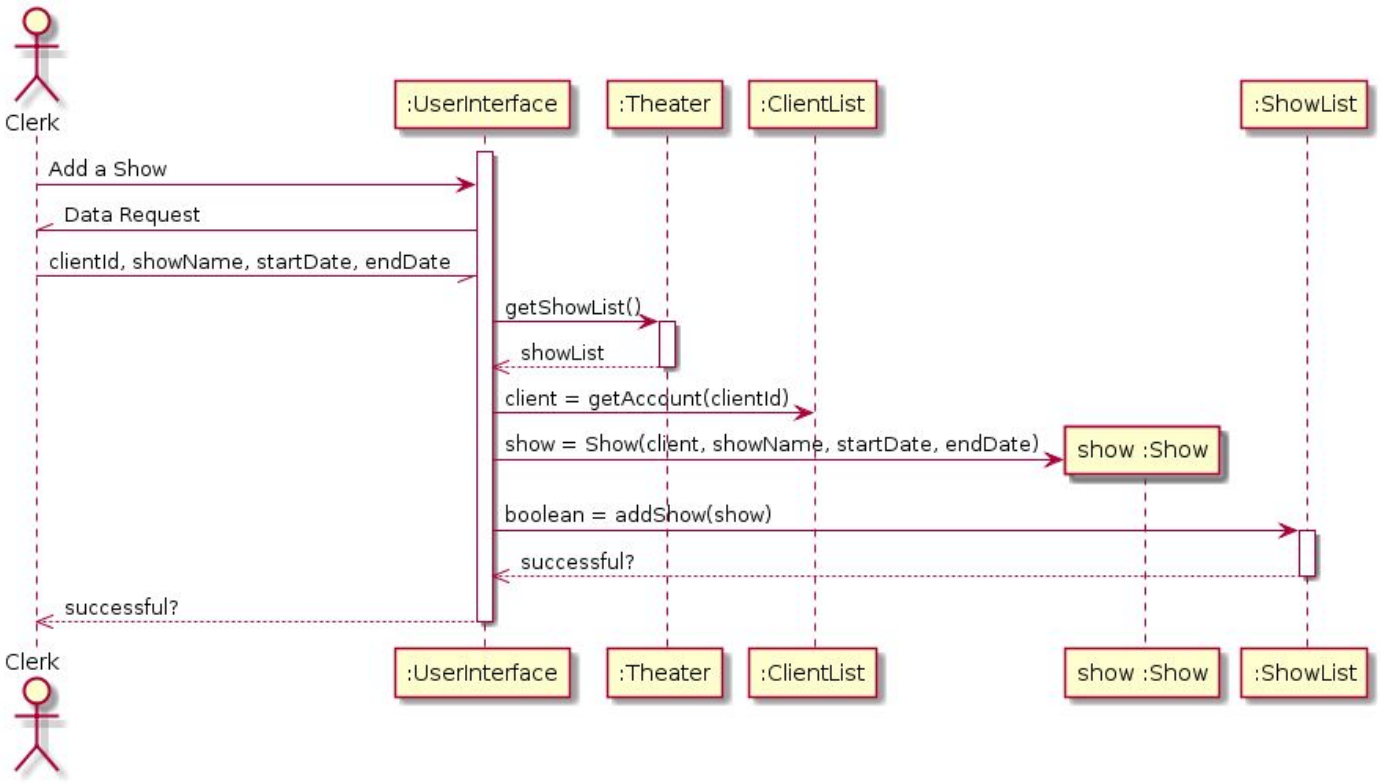
List Shows



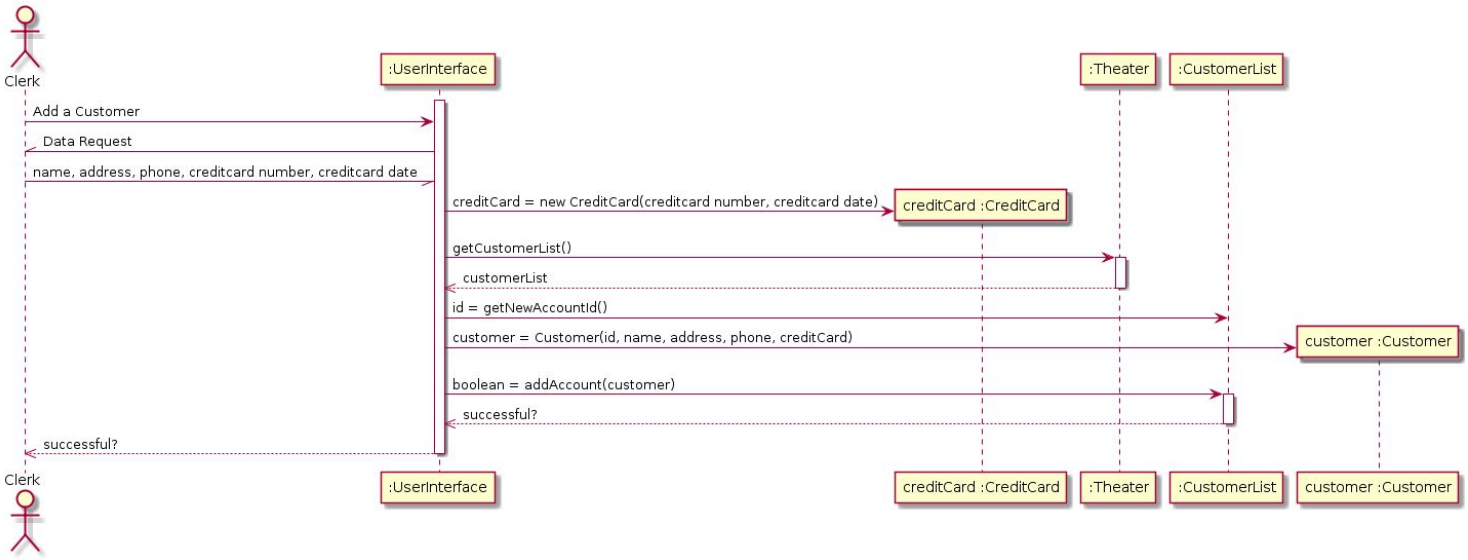
List Clients



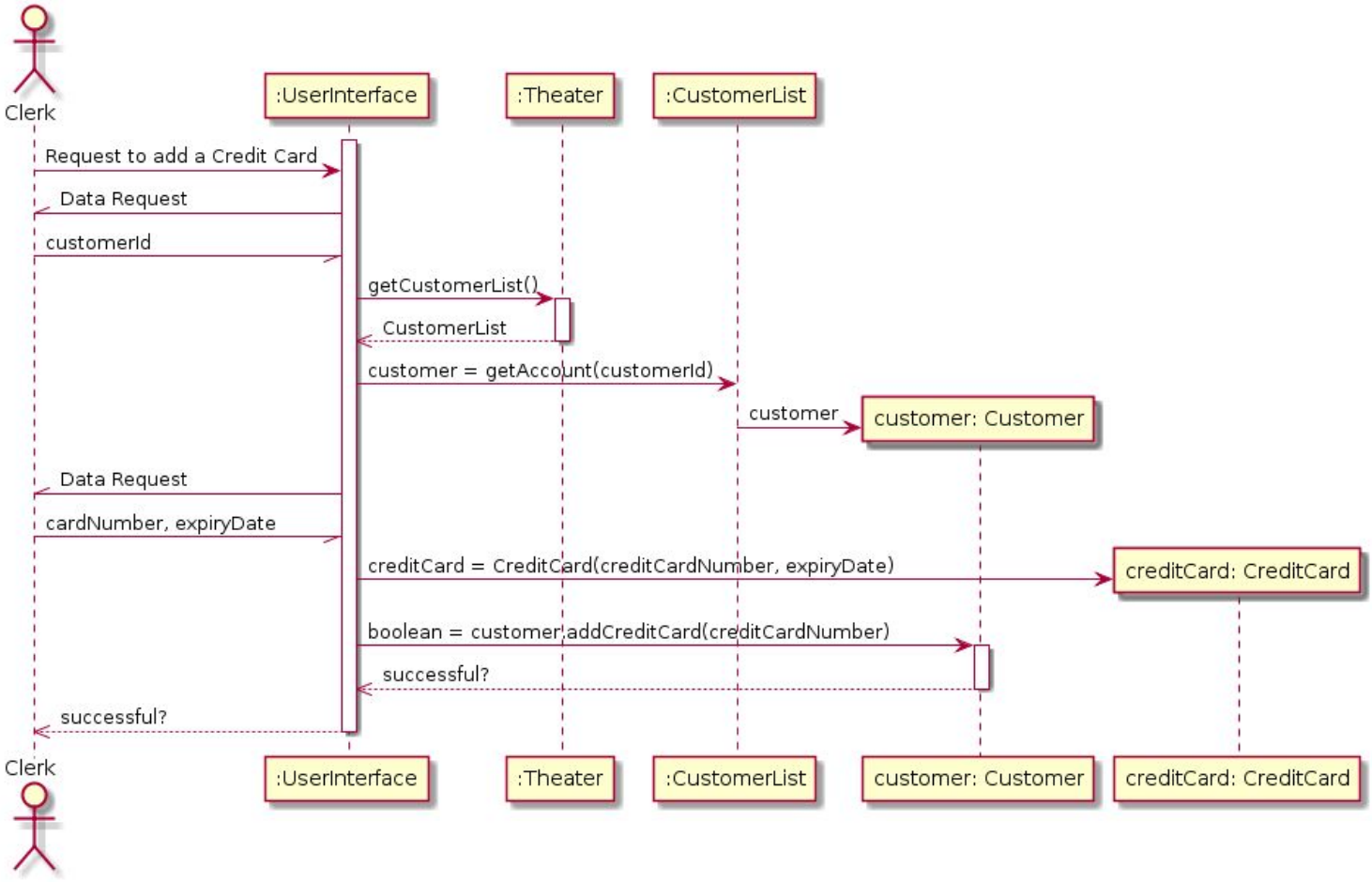
Add a Show



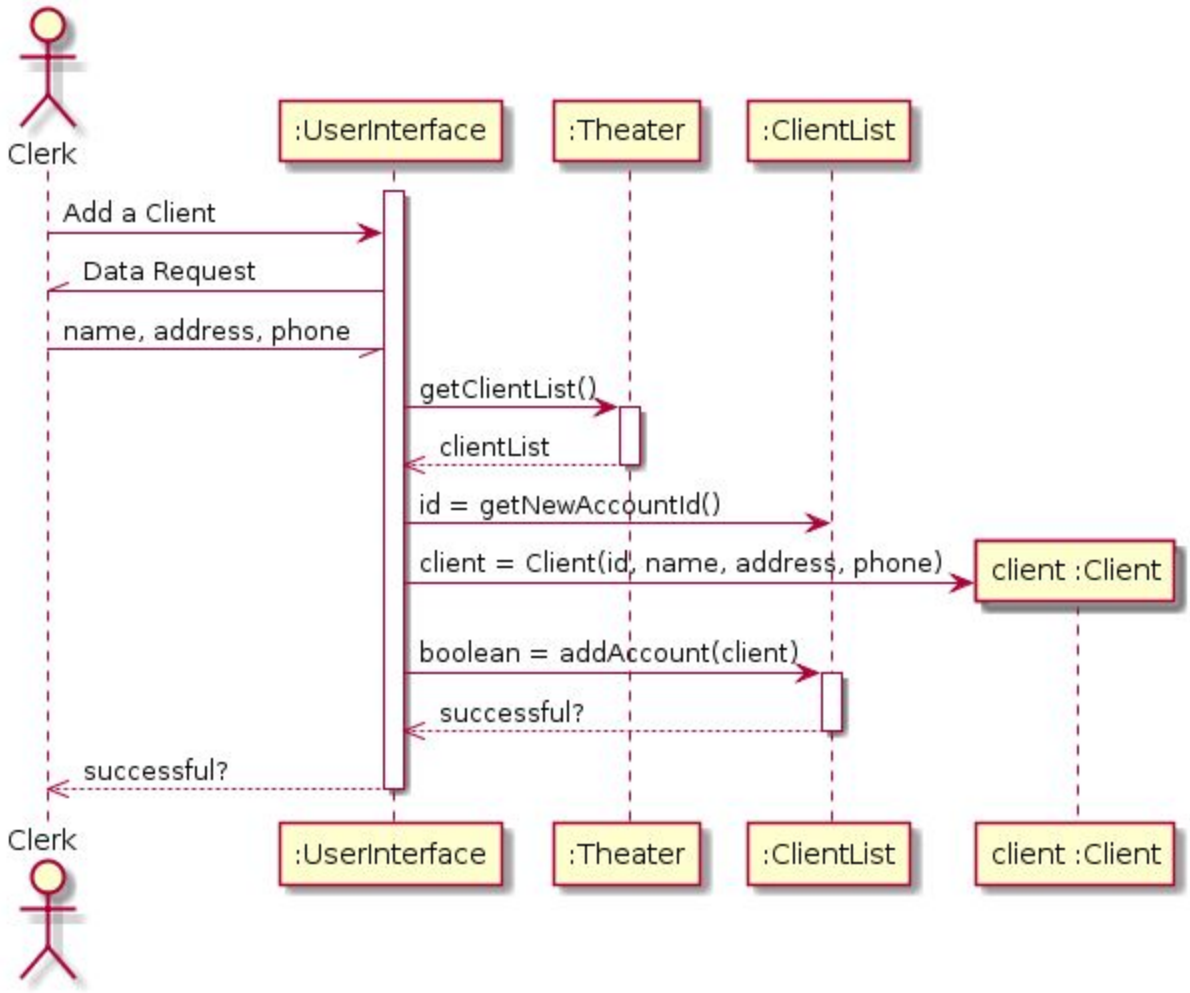
Add a Customer



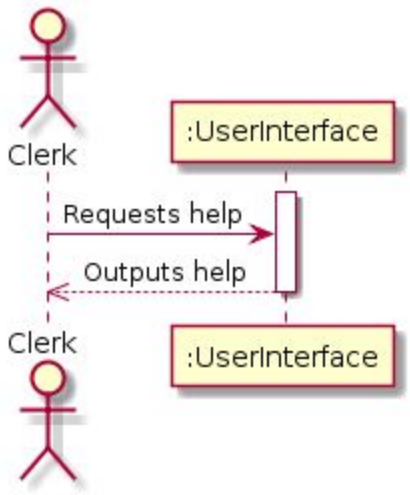
Add a Credit Card



Add a Client


Clerk
:UserInterface
:Theater
:ClientList
Add a Client
Data Request
name, address, phone
getClientList()
clientList
id = getNewAccountId()
client = Client(id, name, address, phone)
client :Client
boolean = addAccount(client)
successful?
successful?
Clerk
:UserInterface
:Theater
:ClientList
client :Client

Show Help



Exit Program