

Asiatic Accounting System

- ☐ **Sequence Diagram**
- ☐ **Statechart Diagram**

Group #1

#0205002
#0205003
#0205004
#0205014
#0205026
#9705065

Sequence Diagram

Create Company

- **Priority:** High
- **Primary Actor:** System Administrator
- **Other Interested Stakeholders:** Company Owner
- **Description:** The number of companies is flexible and there can be such a situation where a new company should be created . Then system administrator can do such using this use-case.
- **Precondition:** The user currently logged in must have the authorization to create company.
- **Trigger:** System administrator selects the create company command.

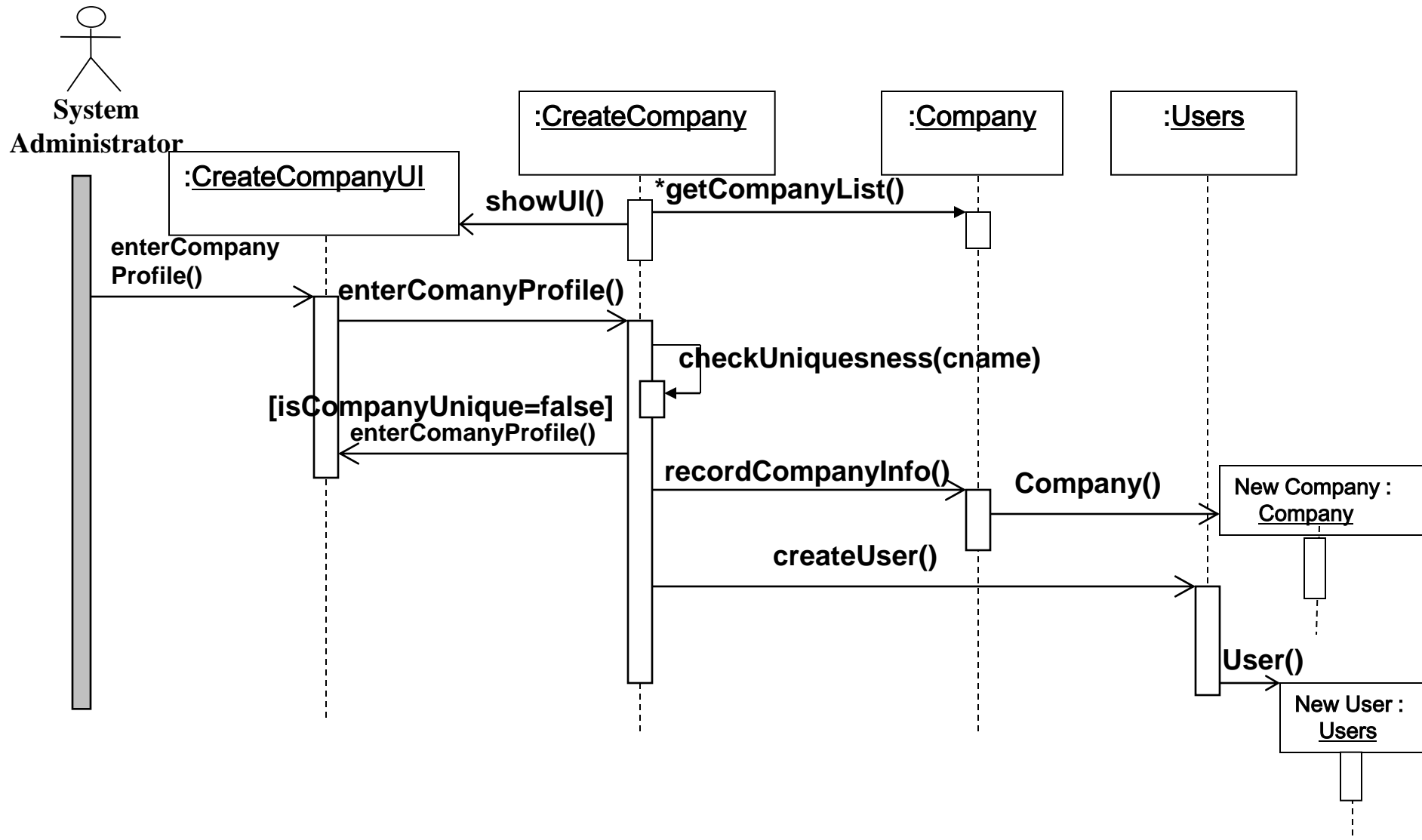
Create Company (cont...)

- Typical Course of Events:
 1. System asks the user to enter company name and other company related information like address , phone no , fax , email etc.
 2. User enters all the necessary information about company profile and commands to store them.
 3. System checks whether the company name is unique.
 4. System generates username , password and service attribute and records the new company.

Create Company (cont...)

- Alternate Course of Events:
 3. **Alt** : System finds that the company name is not unique. It then asks the user to reenter the company name again.
- **Post Condition**: A new company account is created.

Sequence Diagram : Create Company



Modify Company

- **Priority:** Low
- **Primary Actor:** Company administrator
- **Other Interested Stakeholders:** Owner
- **Description:** Some of the data of an existing company may need to be modified . Then authorized user can do such using this use-case.
- **Precondition:** The user currently logged in must have the authorization to create company.
- **Trigger:** System administrator selects the modify company command.

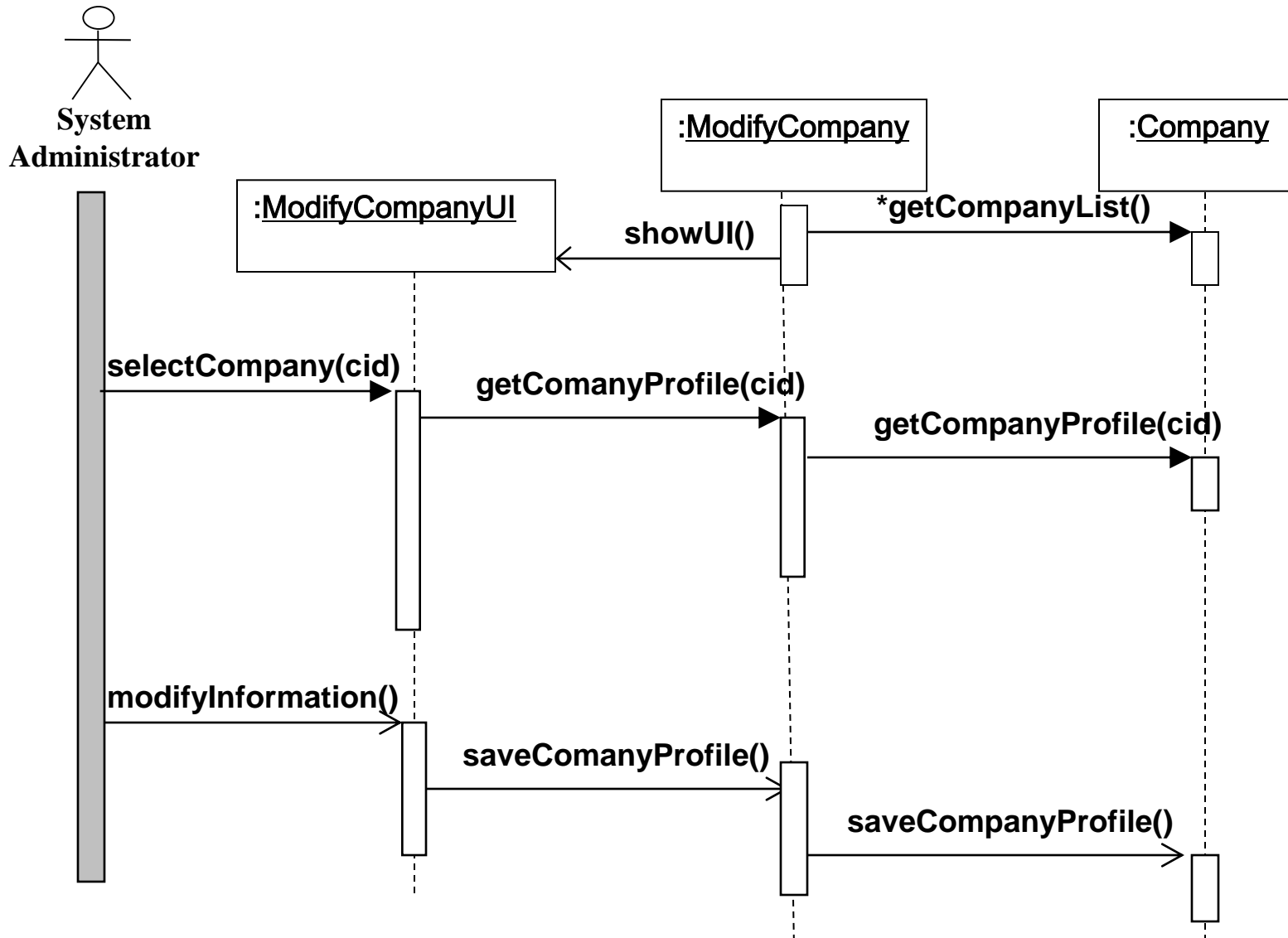
Modify Company (cont...)

- Typical Course of Events:
 1. System asks user to enter company name.
 2. User selects the company name.
 3. System get the company profile if the company name entered by the user matches any of the saved company name.
 4. System shows the form for the user to fill other company related information like address , phone no , fax , email etc with the previous value of those fields appearing.
 5. User changes necessary information and commands to store the modified information.
 6. System modifies the company account .

Modify Company (cont...)

- **Alternate Course of Events:**
 3. **Alt :** System finds that the company name entered by the user does not match with saved company name. It then rejects the entered company name and asks the user to reenter the company name again.
- **Post Condition:** The respective information's of the company account is changed.

Sequence Diagram 1: Modify Company



Create User

- **Priority:** High
- **Primary Actor:** Company administrator .
- **Other Interested Stakeholders:** New user for whom the account is being created.
- **Description:** This use-case depicts the way to create a new user account.
- **Precondition:** The user currently logged in must have the authorization to create user.
- **Trigger:** User selects the create new user command.

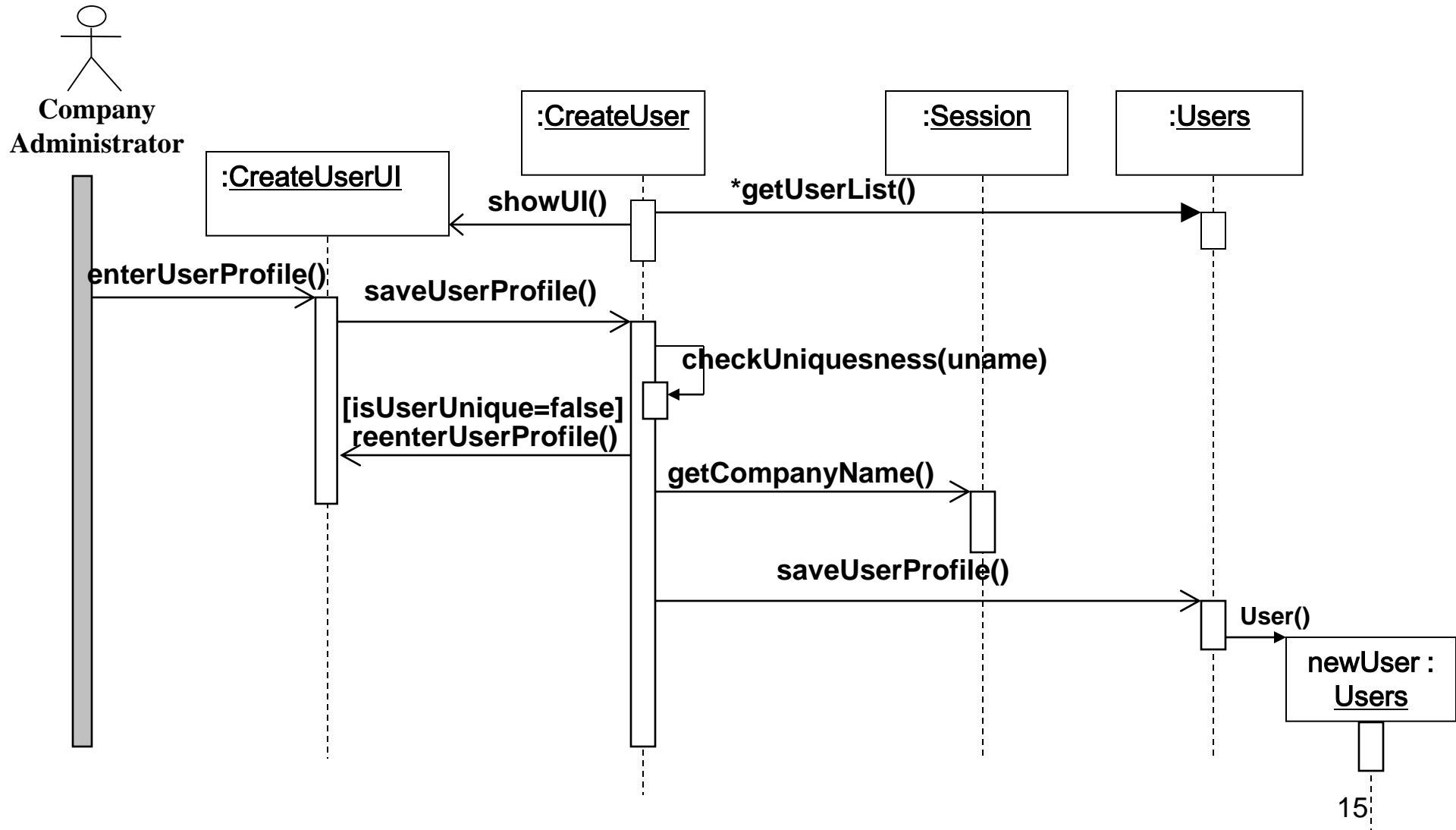
Create User (cont...)

- Typical Course of Events:
 1. System asks the privileged user to enter user profile for the new user.
 2. The privileged user enters all the necessary information for new user.
 3. System checks whether the new user name is unique.
 4. System then commands to store the record.
 5. System sets the status field to be valid and creates a new user account .

Create User (cont...)

- Alternate Course of Events:
 3. Alt : System finds that the new user name is not unique. It then asks the privileged user to reenter the new user name again.
- Post Condition: A new user account is created.

Sequence Diagram : Create User



Modify User

- **Priority:** Low
- **Primary Actor:** Company administrator
- **Other Interested Stakeholders:** The user to be modified
- **Description:** Some of the data of an existing user account may need to be modified or even a user account may need to be deleted. The authorized user can do such things using this use-case
- **Precondition:** The user currently logged in must have the authorization to create company.
- **Trigger:** User selects the modify user command.

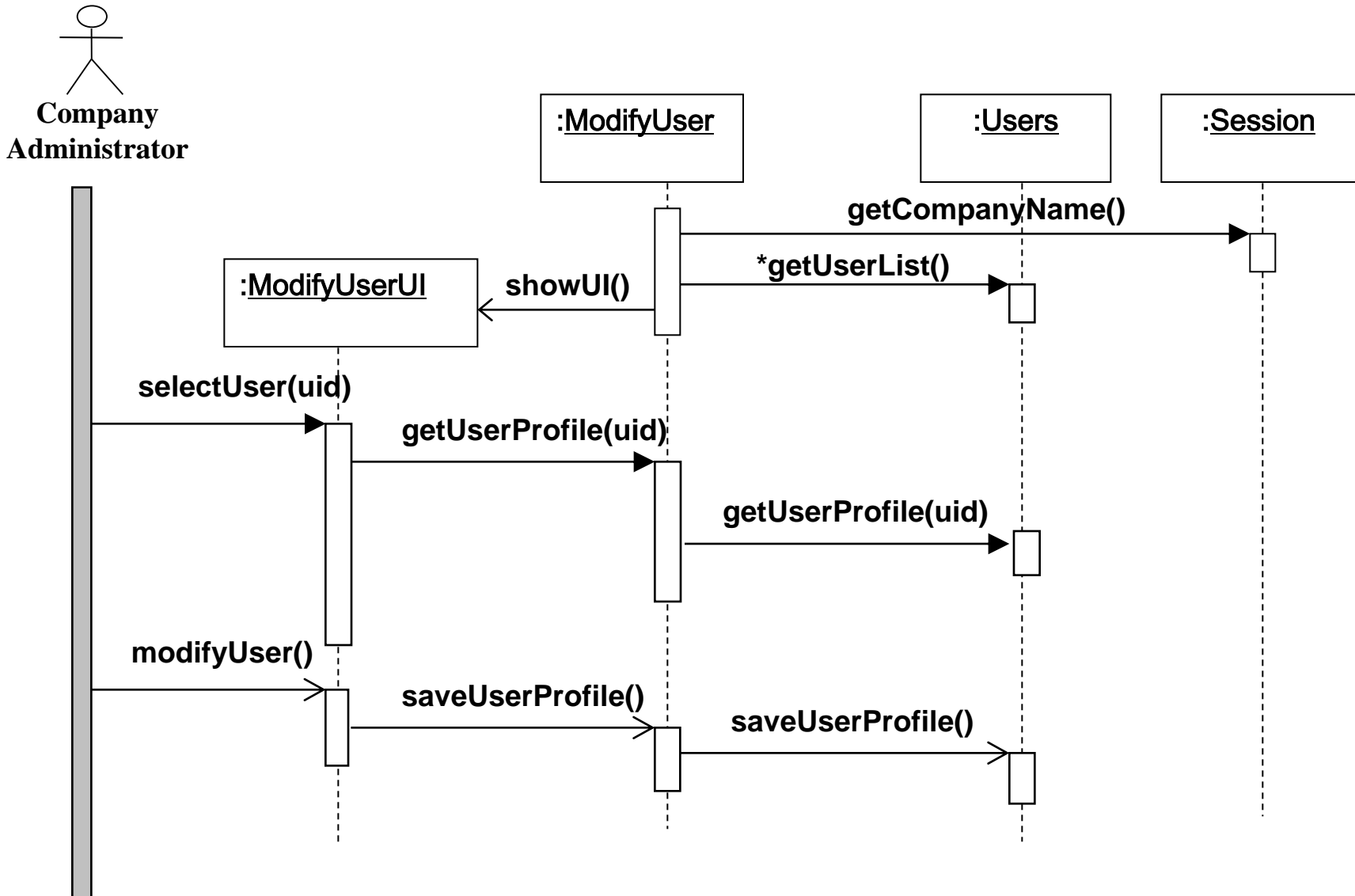
Modify User (cont...)

- Typical Course of Events:
 1. System asks the privileged user to enter the user name to be modified.
 2. The privileged user enters the that user name .
 3. System checks whether the user name entered by the privileged user matches with any saved user account.
 4. System shows the form for the privileged user to fill other related information like user_id , password , status etc with the previous value of those fields appearing.
 5. The privileged user changes necessary information and selects to save.
 6. System modifies the user account .

Modify User (cont...)

- **Alternate Course of Events:**
 3. **Alt** : System finds that the user name entered by the privileged user does not match with any saved user account. It then rejects the entered user name and asks the privileged user to reenter the user name again.
- **Post Condition**: The respective fields of the user account is changed.

Sequence Diagram : **Modify User**



Create Client

- **Priority:** High
- **Primary Actor:** System administrator.
- **Other Interested Stakeholders:** Company Owner,
The client to be created
- **Description:** When a new client comes to a statement with the organization , we need to create a new client account. Authorized user can do such using this use-case
- **Precondition:** The user currently logged in must have the authorization to create company.
- **Trigger:** User selects the create client command.

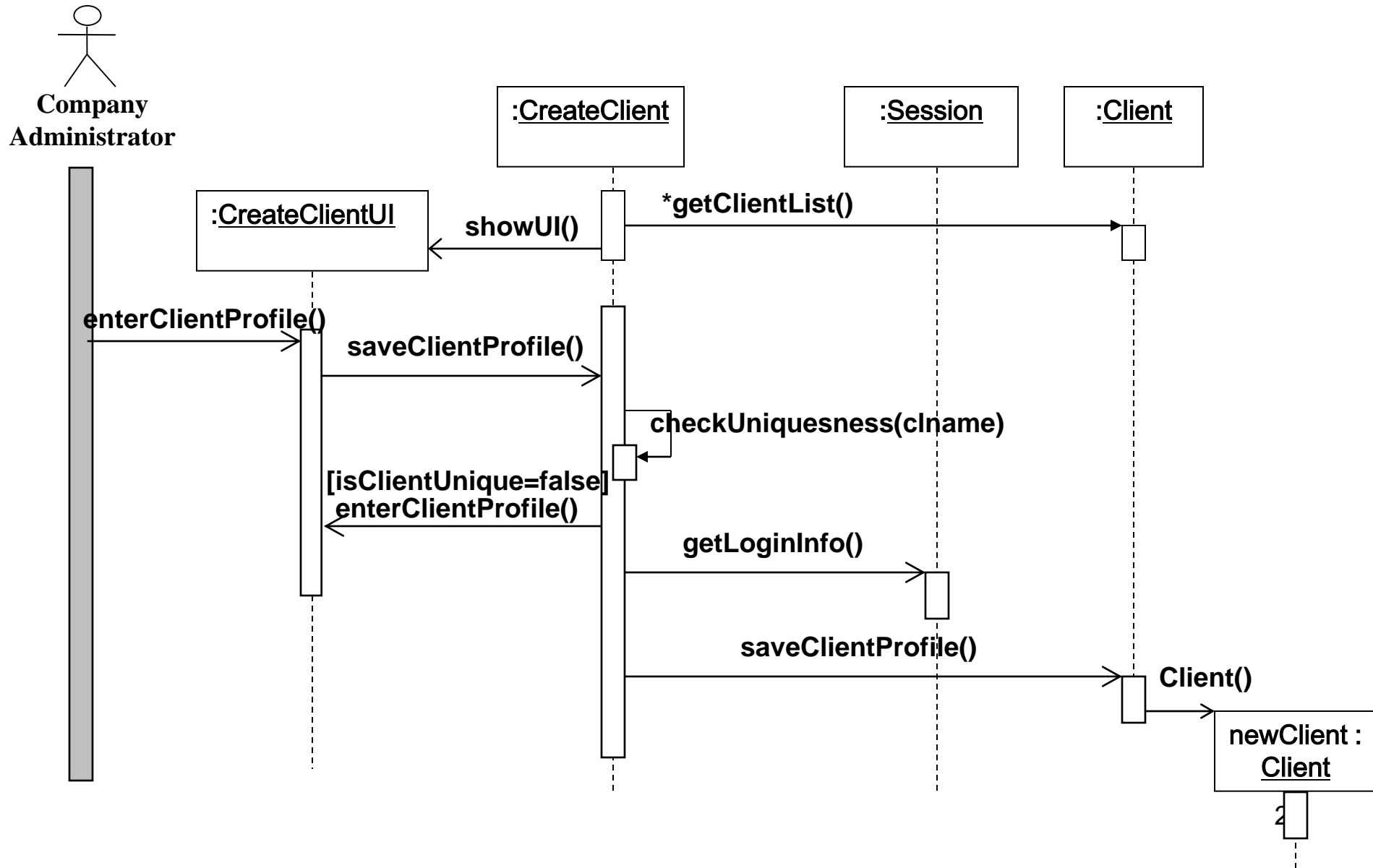
Create Client (cont...)

- Typical Course of Events:
 1. System asks the user to enter client_id and other client related information like client name, address, phone no, fax, email, contact person, agency commission etc.
 2. User enters all the necessary information about client.
 3. System checks whether the client_id is unique.
 4. System finds user_id of the current user and takes it to be the value of the client_creator field.
 5. System creates a client account.

Create Client (cont...)

- Alternate Course of Events:
 3. Alt : System finds that the client_id is not unique. It then asks the user to reenter the client_id again.
- Post Condition: A new client account is created.

Sequence Diagram : Create Client



Modify Client

- **Priority:** Low
- **Primary Actor:** System administrator
- **Other Interested Stakeholders:** Client to be modified , company owner.
- **Description:** Some of the data of an existing client may need to be modified . Then authorized user can do such using this use-case
- **Precondition:** The user currently logged in must have the authorization to create company.
- **Trigger:** User selects the modify client command.

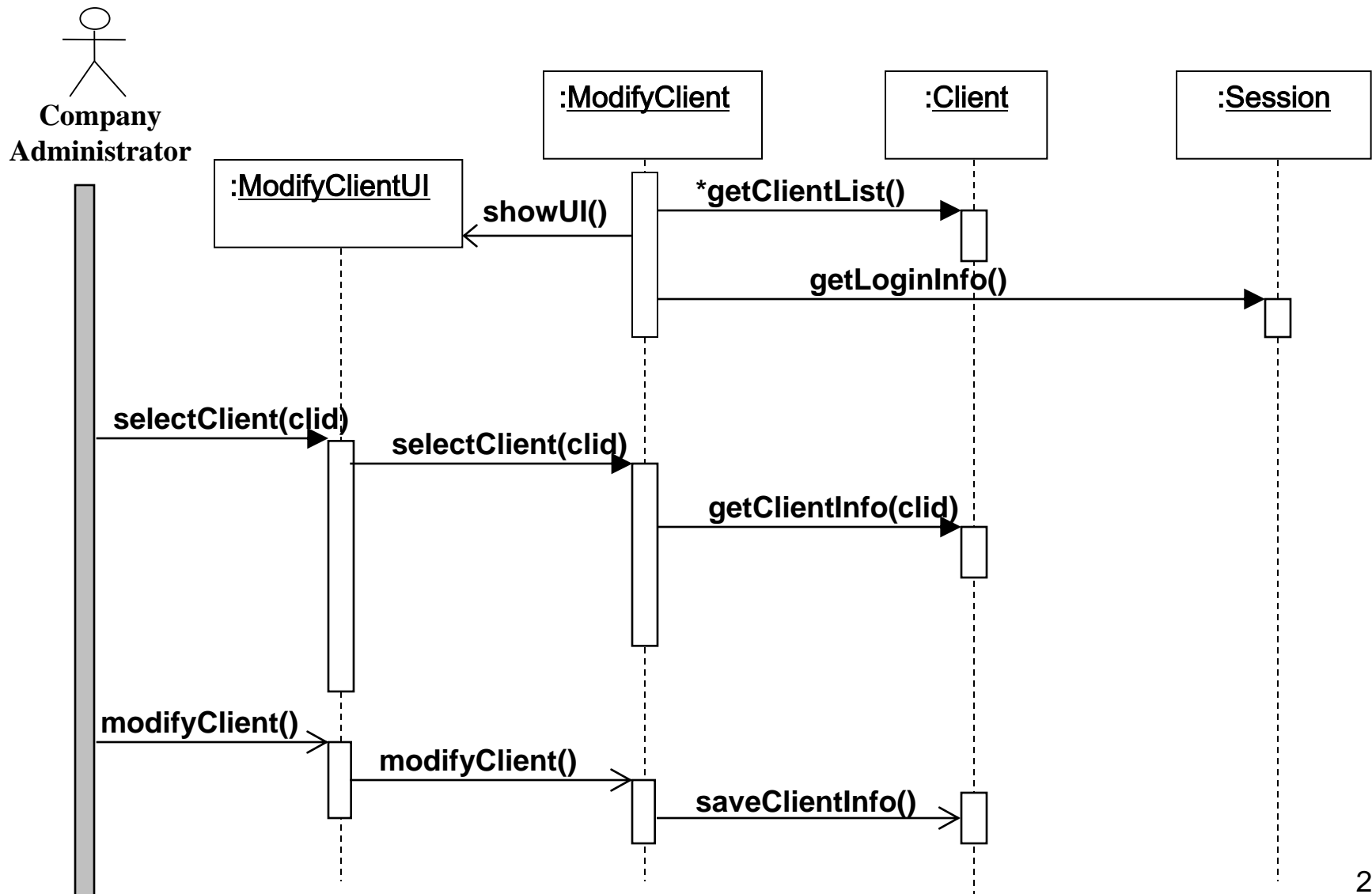
Modify Client (cont...)

- Typical Course of Events:
 1. System asks the user to enter client_id.
 2. User enters the client_id.
 3. System checks whether the client_id entered by the user matches any of the saved client account.
 4. System shows the form for the user to fill other client related information like client name , address , phone no , fax , email , contact person , agency commission etc with the previous value of those fields appearing.
 5. User changes necessary information and selects to store them.
 6. System modifies the client account .

Modify Client (cont...)

- **Alternate Course of Events:**
 3. **Alt :** System finds that the client_id entered by the user does not match with any saved client account. It then rejects the entered client_id and asks the user to reenter the client_id again.
- **Post Condition:** The respective information of the client account is changed.

Sequence Diagram : Modify Client



Create Account

- **Priority:** High
- **Primary Actor:** Company Administrator, Privileged users
- **Secondary Actor:** Planning stuffs - they may need new accounts when new client is added
- **Description:** Account is one of the main objects in a accounting system. This accounting system needs to create accounts to capture the records of money transactions. Whenever a new client, party, employee is added in the system, corresponding accounts are needed to be created.
- **Precondition:** The user needs to have privilege to create accounts.

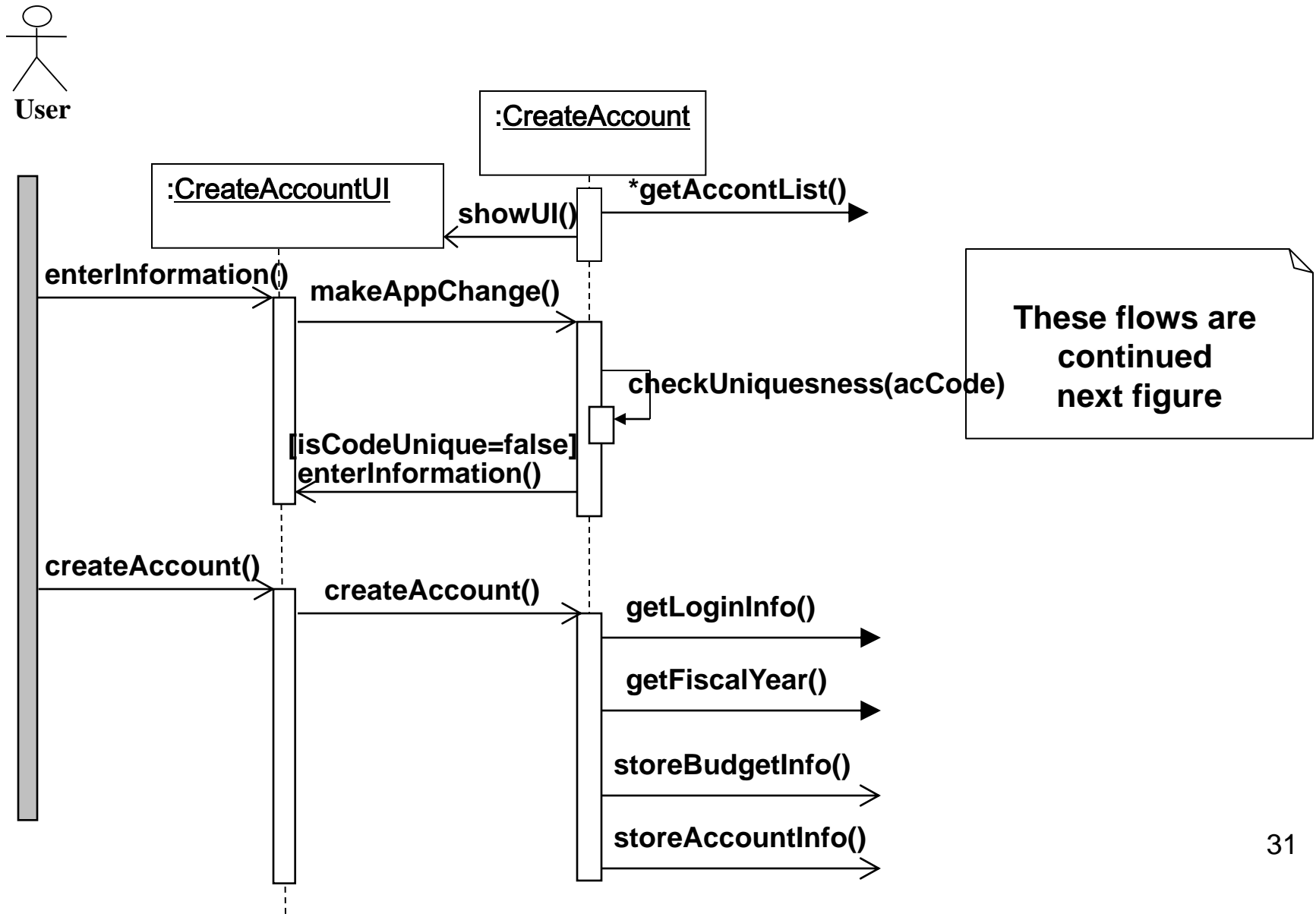
Create Account (cont...)

- Typical Course of events:
 1. The user wants to create a new account.
 2. User provides necessary information like account name, code etc.
 3. User chooses whether this account is a sub-ledger or not.
 4. If it is a sub-ledger, then he needs to choose it's parent account. Else he needs to specify it's type, that is whether it is a manufacturing account or trading account etc. and on what side it will go, that is whether it will be on the debit side or credit side.
 5. User enters the yearly budget for this account
 6. The system responds by checking that the user has entered all necessary information.

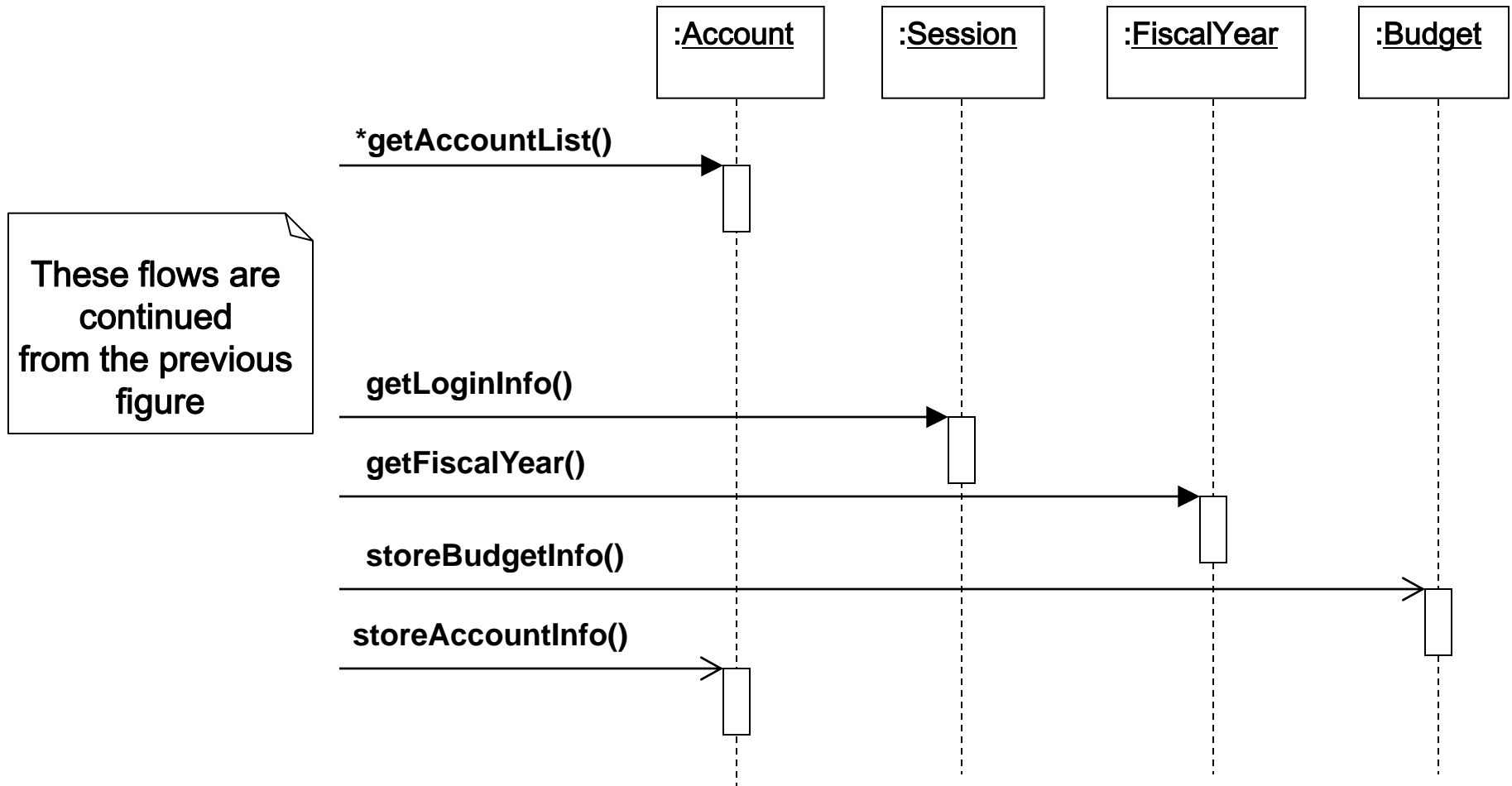
Create Account (cont...)

- **Alternate Course of Events:**
 1. The user may not have privilege to create account. System alerts him.
 2. This code may already exists. User is alerted.
 6. If the user has not provided all the information, he is alerted
 7. The user may cancel the process any time before saving the information.
- **Conclusion:** This use case concludes when the user receives a confirmation message of successful creation of an account.
- **Post Condition:** An account is successfully created.
- **Business Rules:**
 - Voucher entry is possible for sub ledgers only.
 - If a account is a sub ledger, then it's category is same as it's parent account and side is same as parent account.

Sequence Diagram 1: Create Account



Sequence Diagram 2: Create Account



Modify Account

- **Priority:** Normal
- **Primary Actor:** Company Administrator, Privileged users.
- **Description:** If the user feels that the information of an account is wrong or have changed, he may need to modify it.
- **Precondition:** The user needs to have privilege to create accounts.
- **Typical Course of events:**
 1. The user wants to modify a new account.
 2. User provides necessary information like account name, code etc.

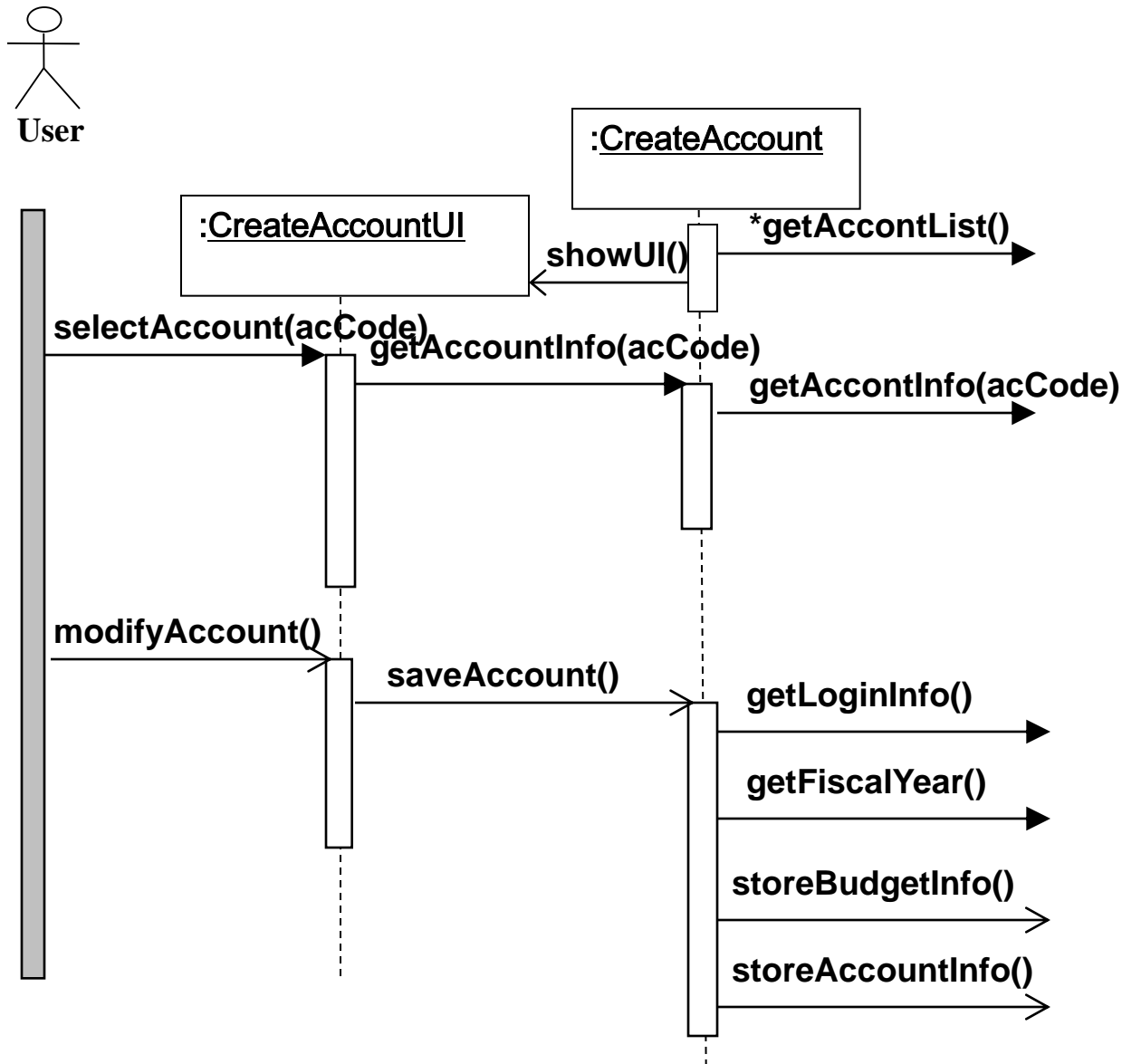
Modify Account (cont...)

- Typical Course of Events (cont.)
 3. The system responds by showing all the information of that account.
 4. The user may alter the information and chooses to save modifications.
 5. The system responds by checking that the user has entered all necessary information.
- Alternate Course of Events :
 1. The user may not have privilege to modify account. System alerts him.
 6. If the user has not provided all the information, he is alerted
 7. The user may cancel the process any time before saving the information.

Modify Account (cont...)

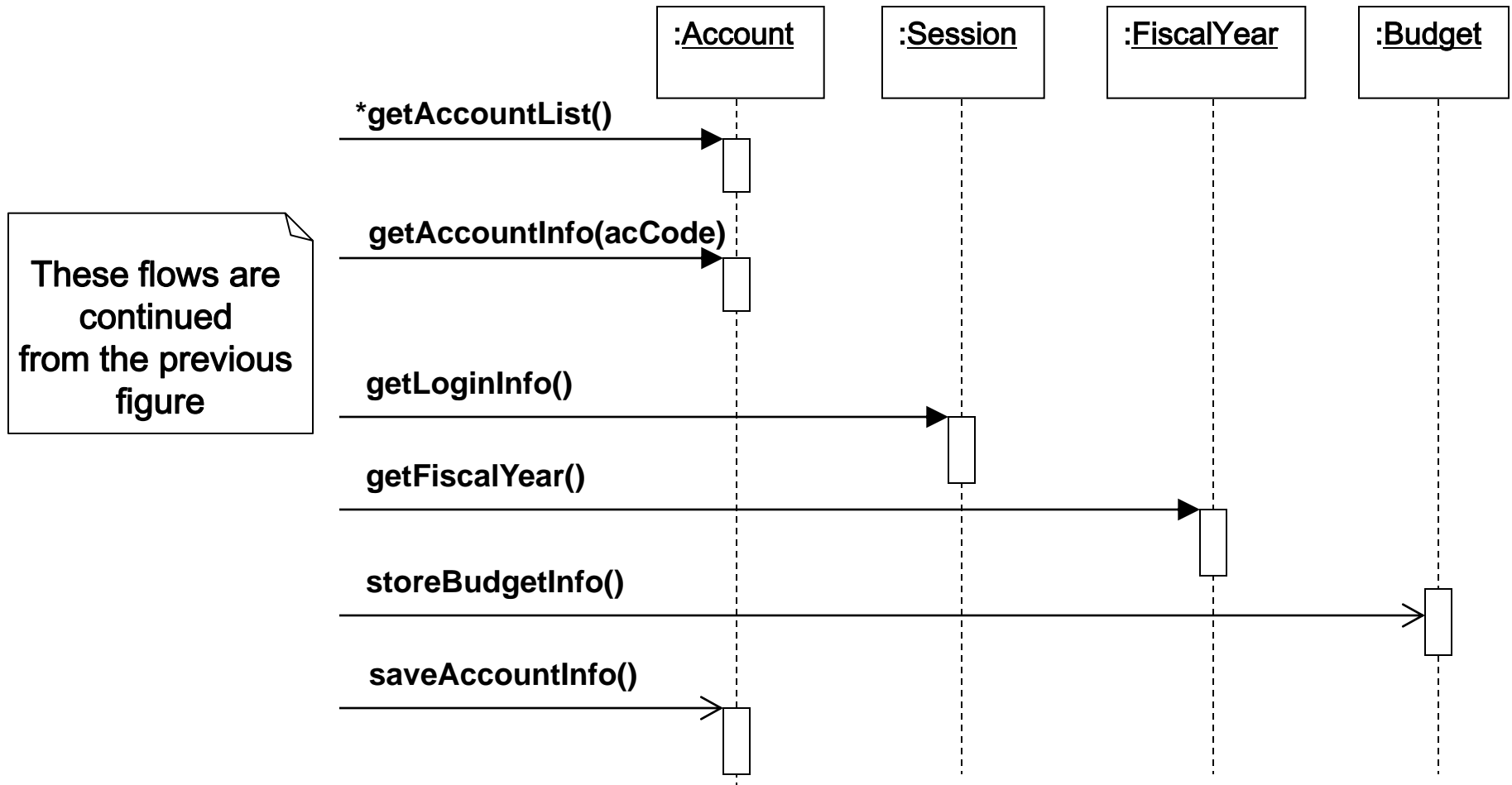
- **Conclusion:** This use case concludes when the user receives a confirmation message of successful modification of an account.
- **Post Condition:** An account is successfully modified.
- **Business Rules:**
 - If an account is so that voucher entry is possible with it, then after modification, it cannot be a parent account.
 - After modification, all the account that are child of the modified account, may need to be modified so that their type and side remains same with their new parent.

Sequence Diagram 1: **Modify Account**



These flows are
continued
next figure

Sequence Diagram 2: **Modify Account**



Create Bank Payment Voucher

- **Priority:** High
- **Primary Actor:** Accounting stuffs
- **Other Interested Stakeholders:** Finance officer, Client, Party, Employee
- **Description:** This use case describes the event of an accounting stuff creating a bank payment voucher.
- **Precondition:** The client/party/employee must have a account.
- **Trigger:** This use case is initiated when a new bank payment voucher is prepared.

Create Bank Payment Voucher (cont...)

- Typical Course of Events:
 1. System sets date to current date.
 2. System sets Voucher No. to be the last bank payment voucher no. of that month.
 3. The actor enters a description for the voucher.
 4. The actor enters a cheque no.
 5. The actor selects the bank account code.
 6. System sets the creditor details to the bank account name.
 7. The actor selects the type of payment.
 8. The actor selects a debtor account code.

Create Bank Payment Voucher (cont...)

- Typical Course of Events (cont...) :
 9. System sets the debtor details to the name of the selected debtor account.
 10. System generates a serial no.
 11. The actor enters the debit amount in taka.
 12. System asks for whether the actor wants to add more debtors.
 13. If the actor doesn't want to add more debtors , he commands the system to record the voucher.
 14. System records the voucher.

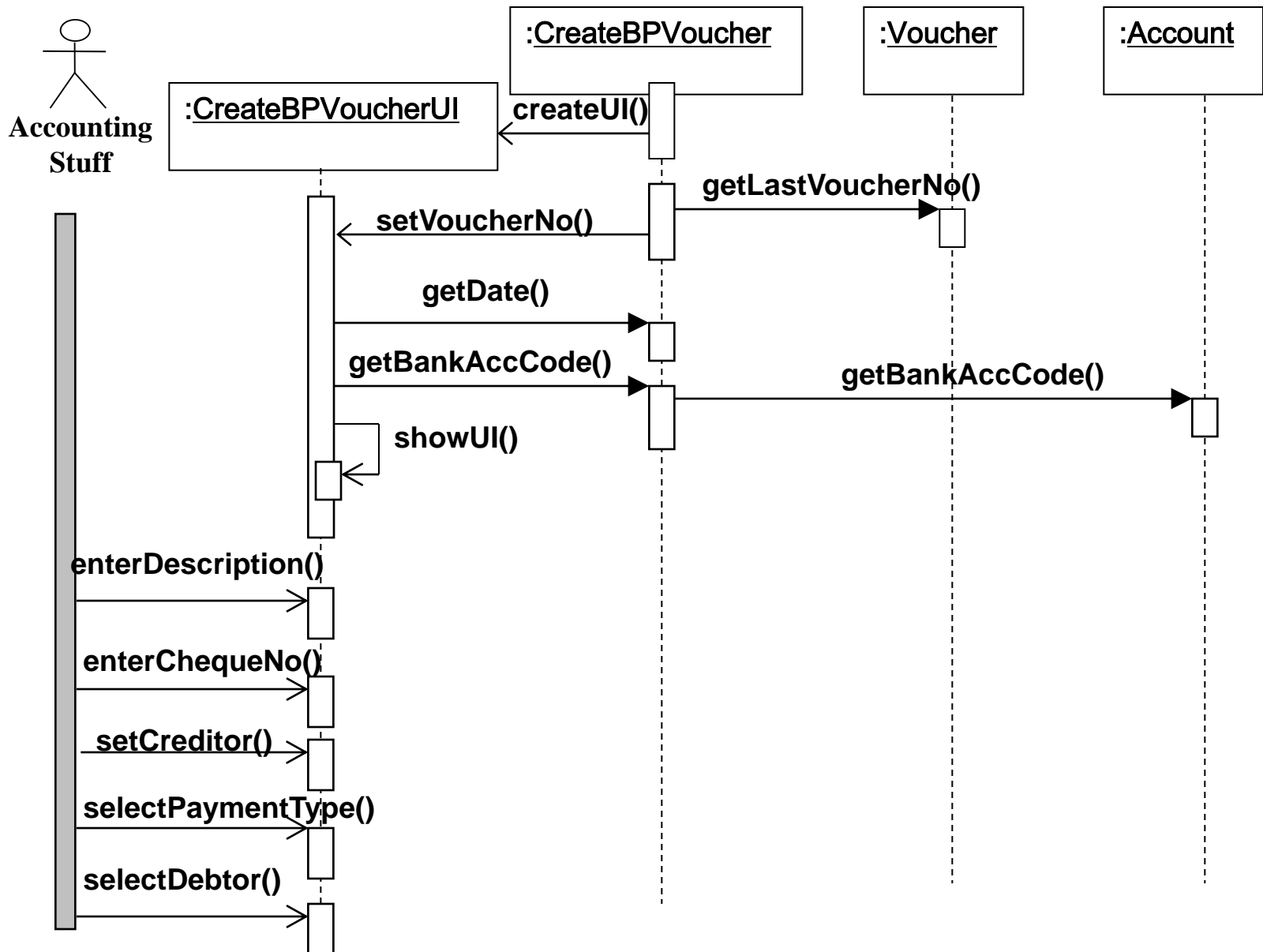
Create Bank Payment Voucher (cont...)

- **Alternate Course of Events:**
 1. The actor may change the date.
 6. The actor may change the bank account details.
 - 11.a. If the category of the debtor account code is “trading account”, then the actor may select the client code, mode of payment, IB Code/Bill no & amount for each IB Code/Bill no.
 - 11.b. If the category of the debtor account code is “P/L account”, then the actor may select the employee code, mode of payment.
 - 11.c. If the category of the debtor account code is “Balance Sheet”, then the actor may enter the mode of payment.
 13. If the actor wants to add more debtors to this voucher, he moves to step 8.

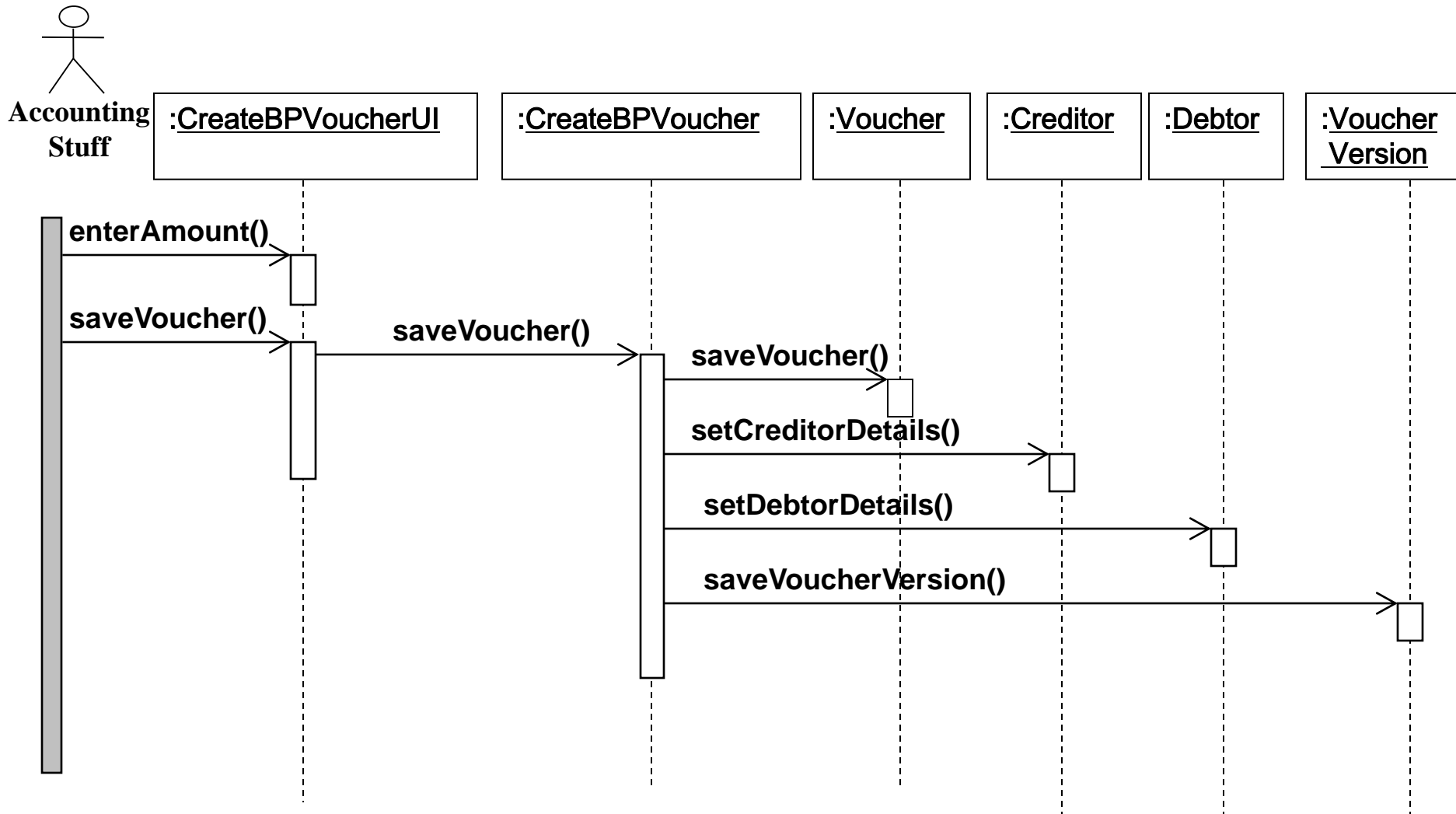
Create Bank Payment Voucher (cont...)

- **Conclusion:**
 - The Bank payment voucher is created.
- **Business Rules:**
 - Only one voucher can be prepared against one cheque no.
 - One account code can't appear more than once in a voucher.
 - Type of Payment:
 - TV
 - PRESS
 - RADIO
 - CINEMA
 - OUTDOOR
 - OTHER
 - EXPENSE
 - PROVISION

Sequence Diagram 1: Create Bank Payment Voucher



Sequence Diagram 2: Create Bank Payment Voucher



Close Account

- **Priority:** Normal
- **Primary Actor:** Company Administrator, Privileged users.
- **Description:** If the user feels that the information of an account is not necessary any more or the account is obsolete, he may close it. It may occur when they finishes business relations with a client, party or employee.
- **Precondition:** The user needs to have privilege to close accounts.

Close Account (cont...)

- Typical Course of events:
 1. The user wants to close a new account.
 2. User provides necessary information like account name, code etc.
 3. The system responds by showing all the information of that account.
 4. The user chooses to close that account.
 5. The system responds by checking that it is possible to close that account. If possible, the account is closed.

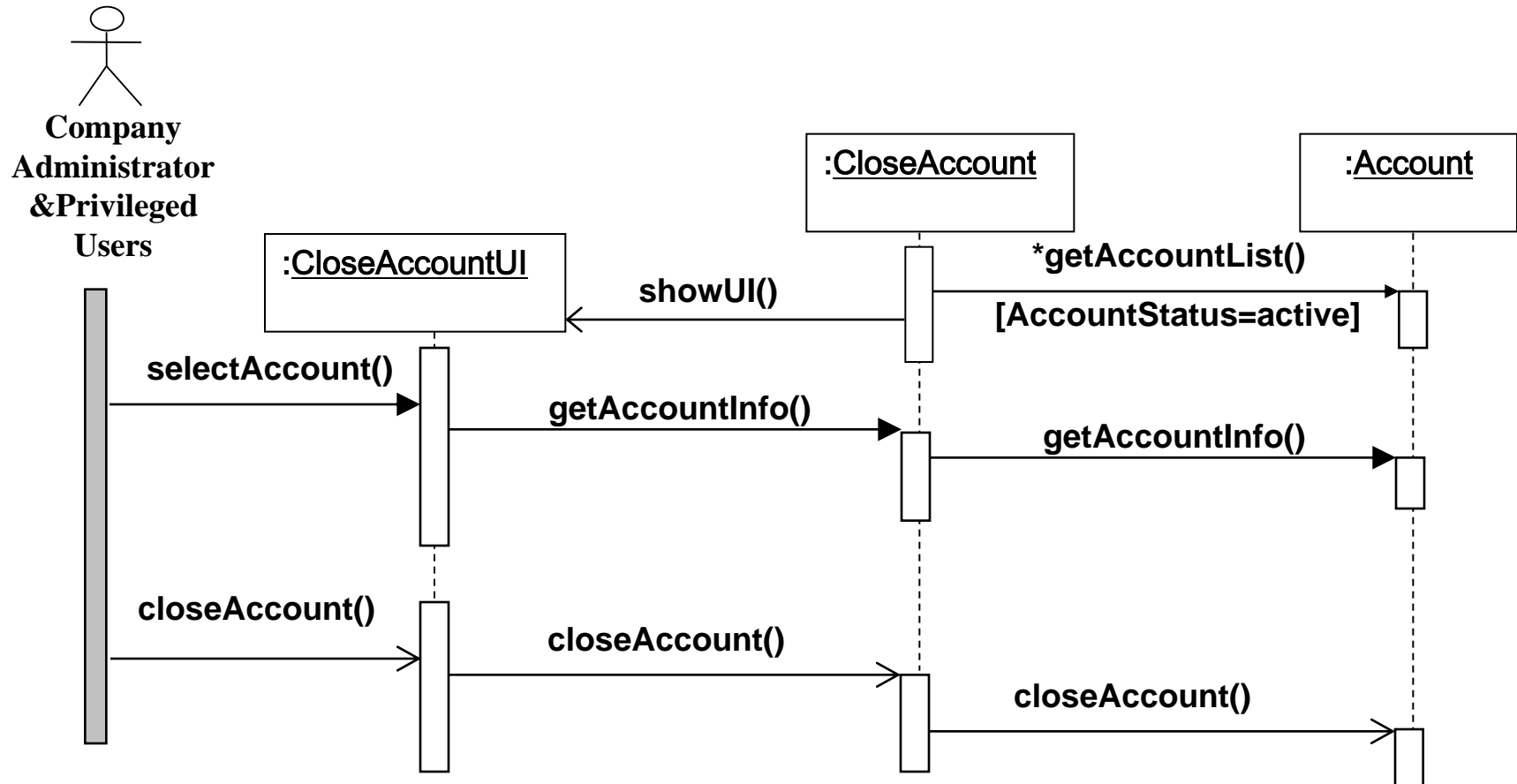
Close Account (cont...)

- Alternate Course of events:
 1. The user may not have privilege to modify account. System alerts him.
 2. This code may not exist. User is alerted.
 4. If the user has not provided all the information, he is alerted
 5. It may not be possible to close the account. He is alerted
 6. The user may cancel the process any time before saving the information.

Close Account (cont...)

- **Conclusion:** This use case concludes when the user receives a confirmation message of successful closing of an account.
- **Post Condition:** An account is successfully closed and no voucher entry or reports can be seen regarding that account.
- **Business Rules:**
 - An account with active child accounts cannot be closed. User needs to close child accounts first
 - An account with un-posted vouchers cannot be closed.
 - An account whose balance is not zero cannot be closed.

Sequence Diagram : Close Account



Reopen Account

- **Priority:** Less
- **Primary Actor:** Company Administrator, Privileged users.
- **Description:** If the user feels that a closed account is necessary again, he may reopen it. It may occur when they restarts business relations with a client, party or employee that was postponed.
- **Precondition:** The user needs to have privilege to reopen accounts.

Reopen Account (cont...)

- Typical Course of events:
 1. The user wants to reopen a account.
 2. User provides necessary information like account name, code etc.
 3. The system responds by showing all the information of that account.
 4. The user chooses to reopen that account.
 5. The system responds by checking that it is possible to reopen that account. If possible, the account is reopened.

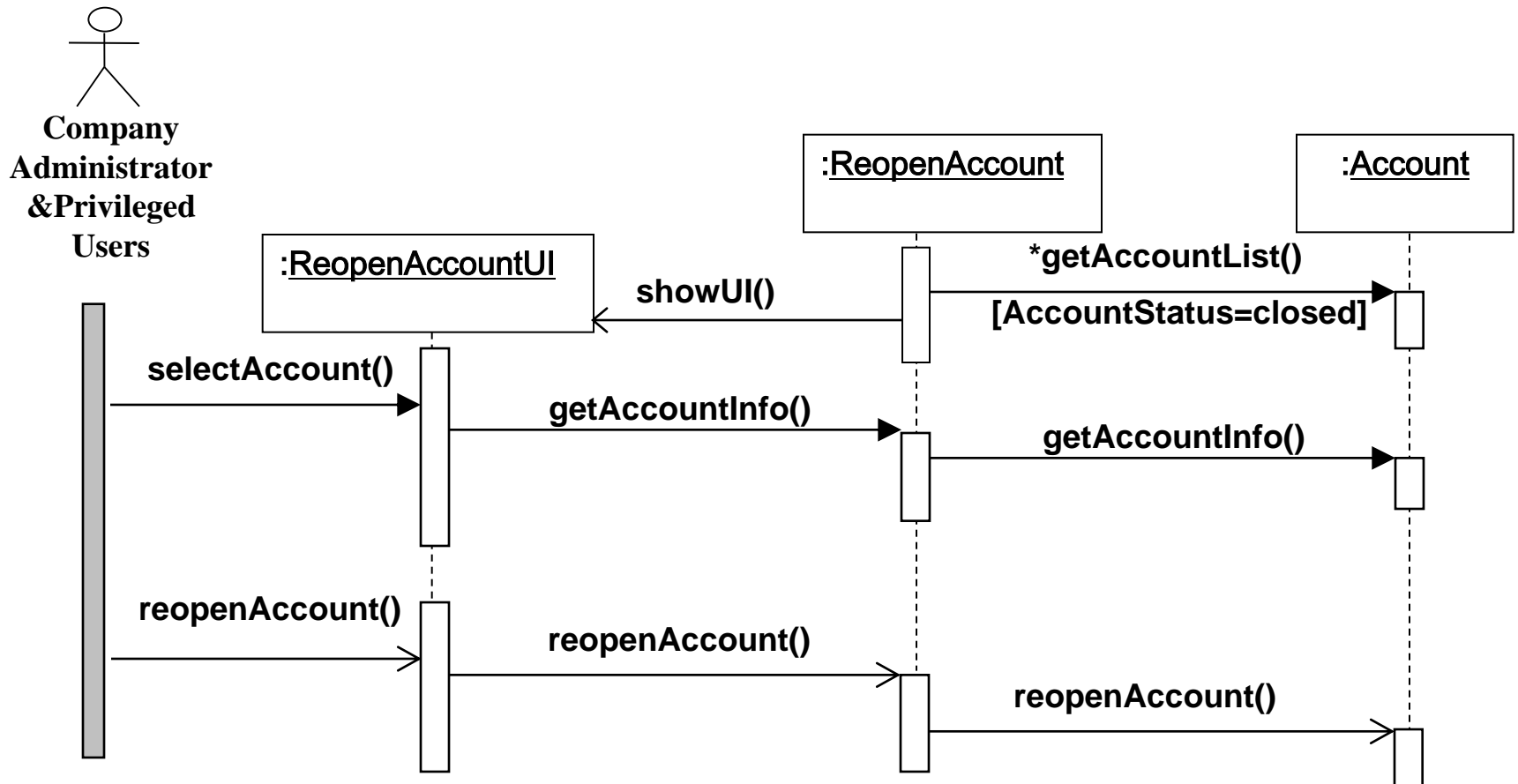
Reopen Account (cont...)

- Alternate Course of events:
 1. The user may not have privilege to reopen account. System alerts him.
 2. This code may not exist. User is alerted.
 4. If the user has not provided all the information, he is alerted
 5. It may not be possible to reopen the account. He is alerted
 6. The user may cancel the process any time before saving the information.

Reopen Account (cont...)

- **Conclusion:** This use case concludes when the user receives a confirmation message of successful reopening an account.
- **Post Condition:** An account is successfully reopened and voucher entry or reports can be seen regarding that account.
- **Business Rules:**
 - An account with closed parent accounts cannot be reopened. User needs to reopen parent accounts first.

Sequence Diagram : Reopen Account



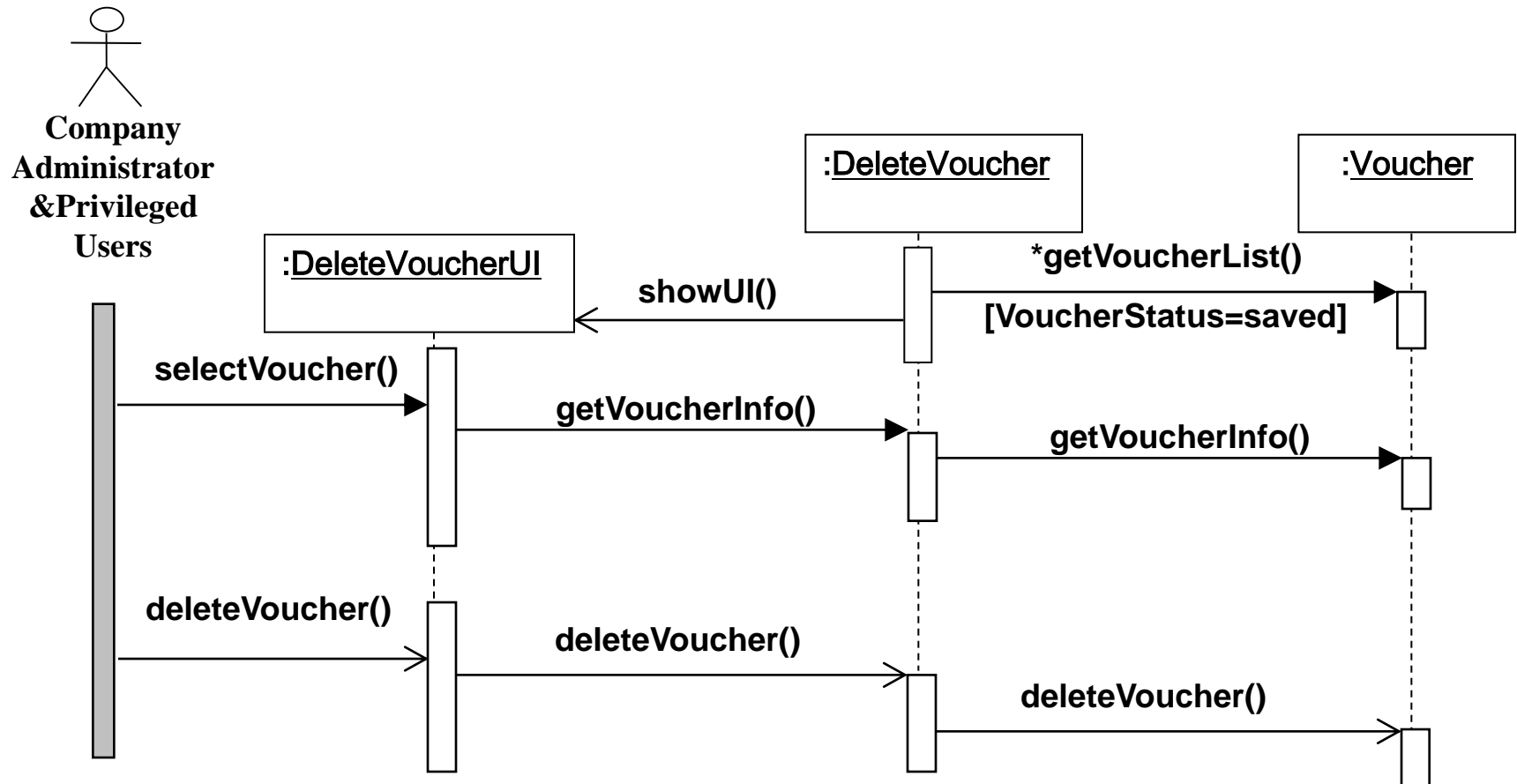
Delete Voucher

- **Priority:** Low
- **Primary Actor:** Company Administrator, Accounting Stuffs
- **Description:** Voucher is one of the important document in an Accounting System. When the voucher transaction is invalid, the voucher is deleted.
- **Precondition:** The users must have enough privilege to delete the voucher.

Delete Voucher (cont...)

- **Typical Course of Events :**
 1. System asks the user to enter voucher no and created on.
 2. User enter the necessary information about the voucher.
 3. System finds the voucher and sets its status entry false.
 4. System then finalize it by sending a confirmation message.
- **Alternate Course of Events:**
 - 4.a. System finds the voucher no and created on is not properly input by user. Return error messages to user to correct it.
 - 4.b. System can't find the specified voucher. User is informed that the specified voucher is not found & can't be deleted
- **Post Condition:** The specified voucher is deleted.

Sequence Diagram : Delete Voucher



Modify Voucher

- **Priority:** Medium
- **Primary Actor:** Company Administrator & Accounting Stuffs
- **Description:** Voucher is one of the important document in an Accounting System. When the transaction amount of a voucher is changed, then the voucher may need to be modified.
- **Precondition:** The users must have enough privilege to modify the voucher.

Modify Voucher (cont...)

□ Typical Course of Events:

1. System asks the user to enter voucher no and created on.
2. User enters voucher no and created on.
3. System then show all the information about the entered voucher.
4. User modifies necessary information and commands to store them.
5. System responds by giving a confirmation message to the user.

Modify Voucher (Cont.)

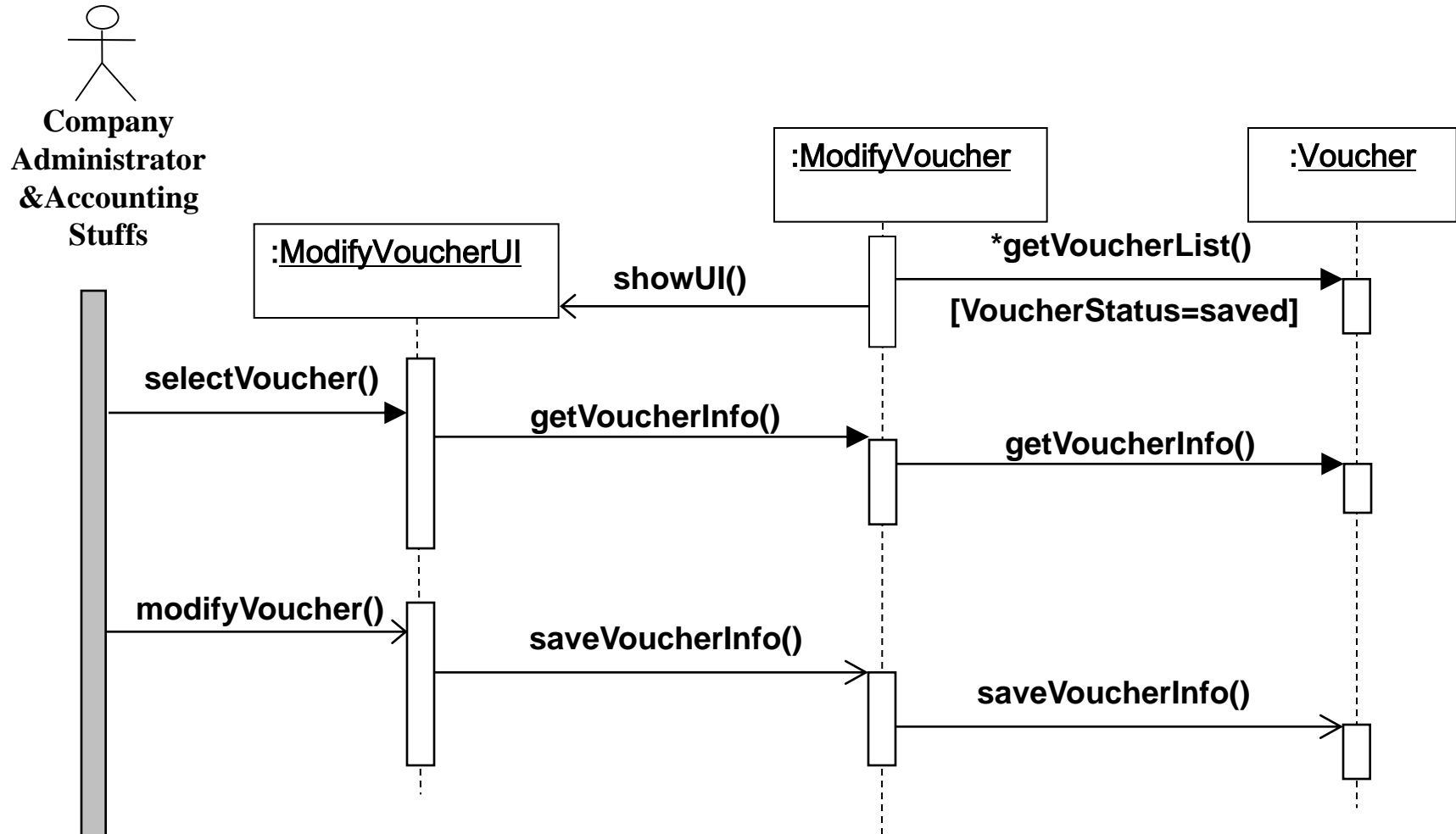
- Alternate Course of Events:

Alt-Step 4.a System finds the voucher no and created on is not properly input by user. Return error messages to user to correct it.

Alt-Step 4.b System can't find the specified voucher. User is informed that the specified voucher is not found & can't be modified.

- **Post Condition:** The specified voucher is modified by user specification.

Sequence Diagram : Modify Voucher



Post Vouchers

- **Priority:** High
- **Primary Actor:** Company Administrator, Privileged users.
- **Secondary Actor:** Checking person
- **Description:** When a voucher is finalized, it is posted.
- **Precondition:** The user needs to have privilege to post vouchers.

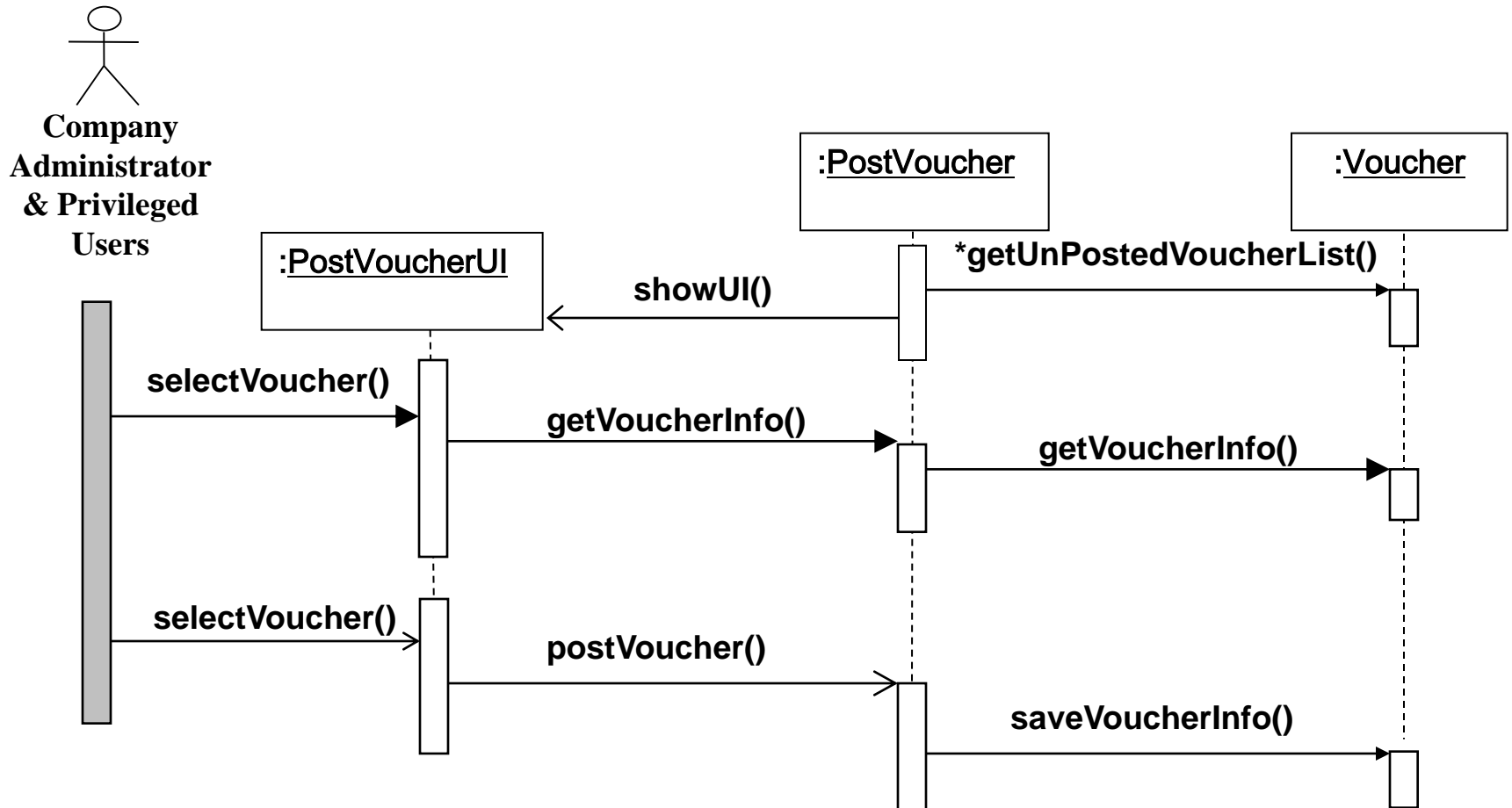
Post Vouchers (cont...)

- Typical Course of events:
 1. The user wants to see the list of vouchers that is not posted.
 2. User posts the appropriate vouchers
 3. The user saves all these information.
 4. The system responds by setting the posted bit of the voucher table.
- Alternate Course of events:
 - 1. The user may not have privilege to reconcile vouchers. System alerts him.
 - 4. The user may cancel the process at any time before saving.

Post Vouchers (cont...)

- **Post Condition:** The posted bit of the vouchers are set true.
- **Business Rules:**
 - A BR/BP Voucher needs to be reconciled before being posted.
 - After being posted, voucher information cannot be modified.

Sequence Diagram : Post Voucher



Reconcile BP Vouchers

- **Priority:** Normal
- **Primary Actor:** Company Administrator, Privileged users.
- **Description:** After each month, the bank issues a list of cheque drawn from that account. That is the actual amount of money that has been removed from the company account. BP voucher are reconciled according to this
- **Precondition:** The user needs to have privilege to reconcile BP vouchers.

Reconcile BP Vouchers (cont...)

- Typical Course of events:

1. The user wants to see the list of BP vouchers that is not reconciled.
2. User reconciles the appropriate vouchers seeing the bank statement.
3. The user saves all these information.
4. The system responds by setting the reconciled bit of the voucher table.

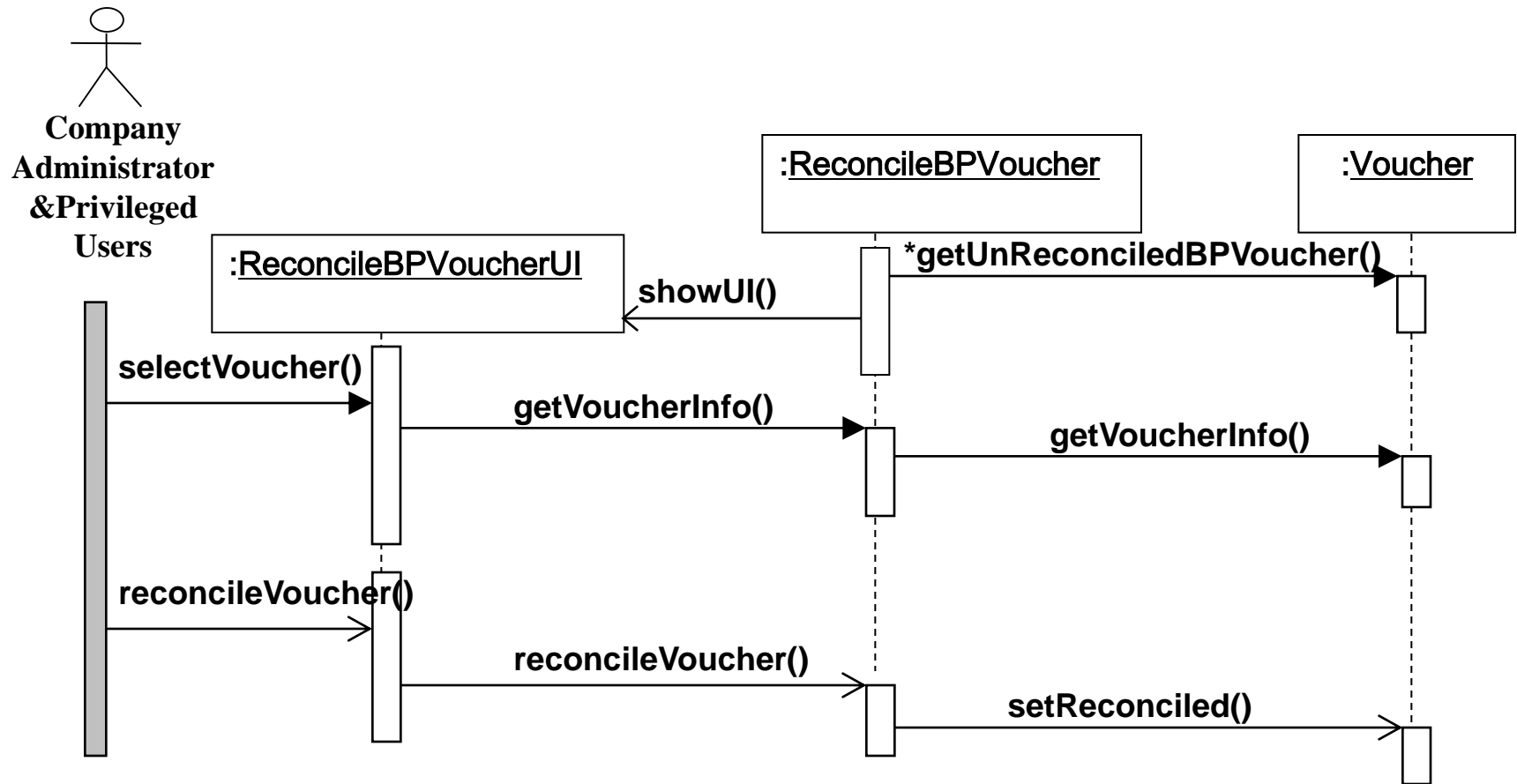
- Typical Course of events:

1. The user may not have privilege to reconcile vouchers. System alerts him.
2. The user may cancel the process at any time before saving.

Reconcile BP Vouchers (cont...)

- **Post Condition:** The reconciled bit of the BP vouchers are set true.
- **Business Rules:**
 - A BP Voucher needs to be reconciled before being posted.
 - After reconciliation, BP voucher information cannot be modified.

Sequence Diagram: Reconcile BP Voucher



StateChart Diagram

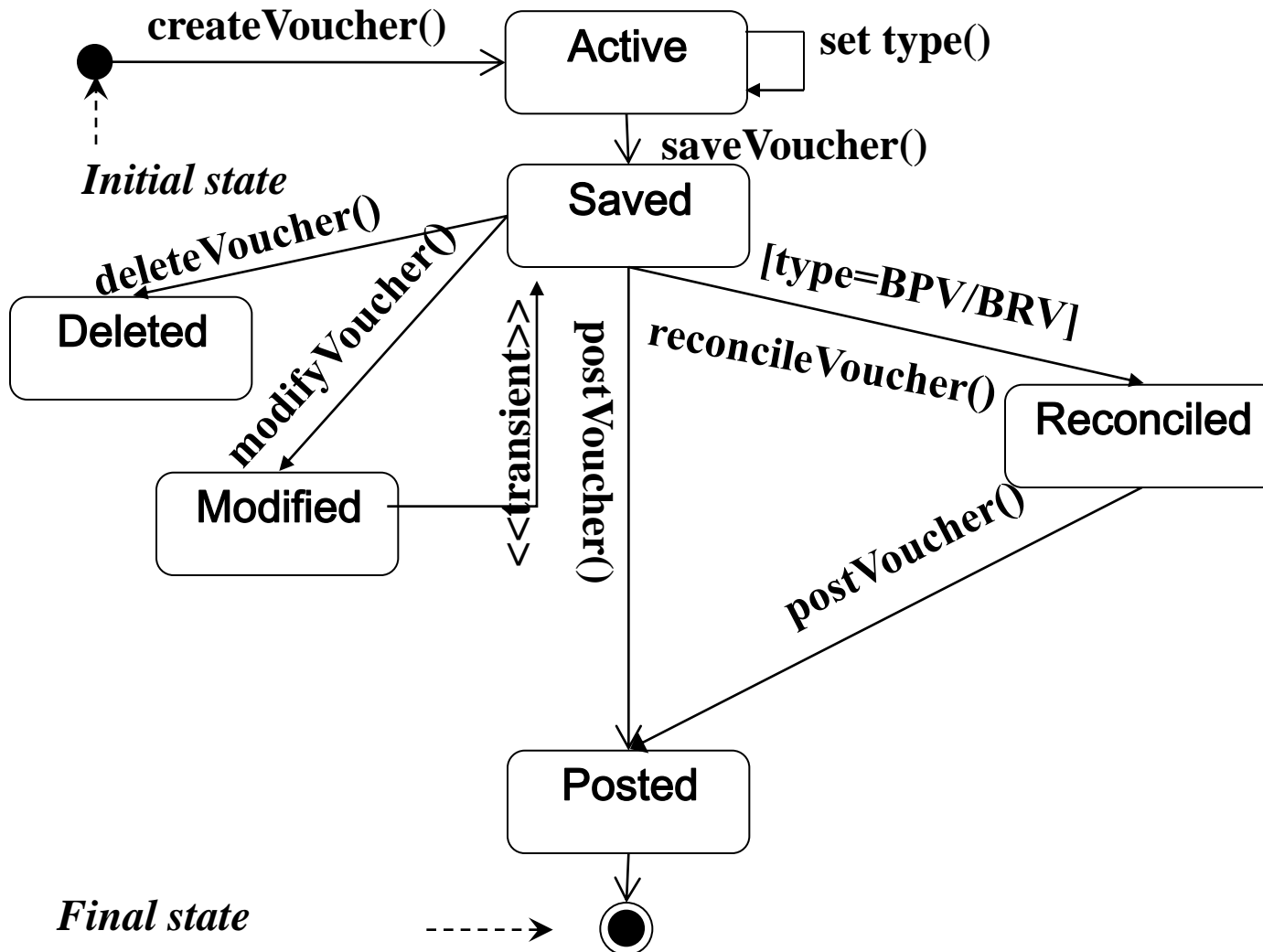
State : Voucher

- Initial State
- Active
- Saved
- Modified/deleted
- Posted
- Reconciled

Use Case : Voucher

- Create Bank Payment Voucher
- Modify Voucher
- Delete Voucher
- Post Voucher
- Reconcile BP Voucher

Statechart for Voucher



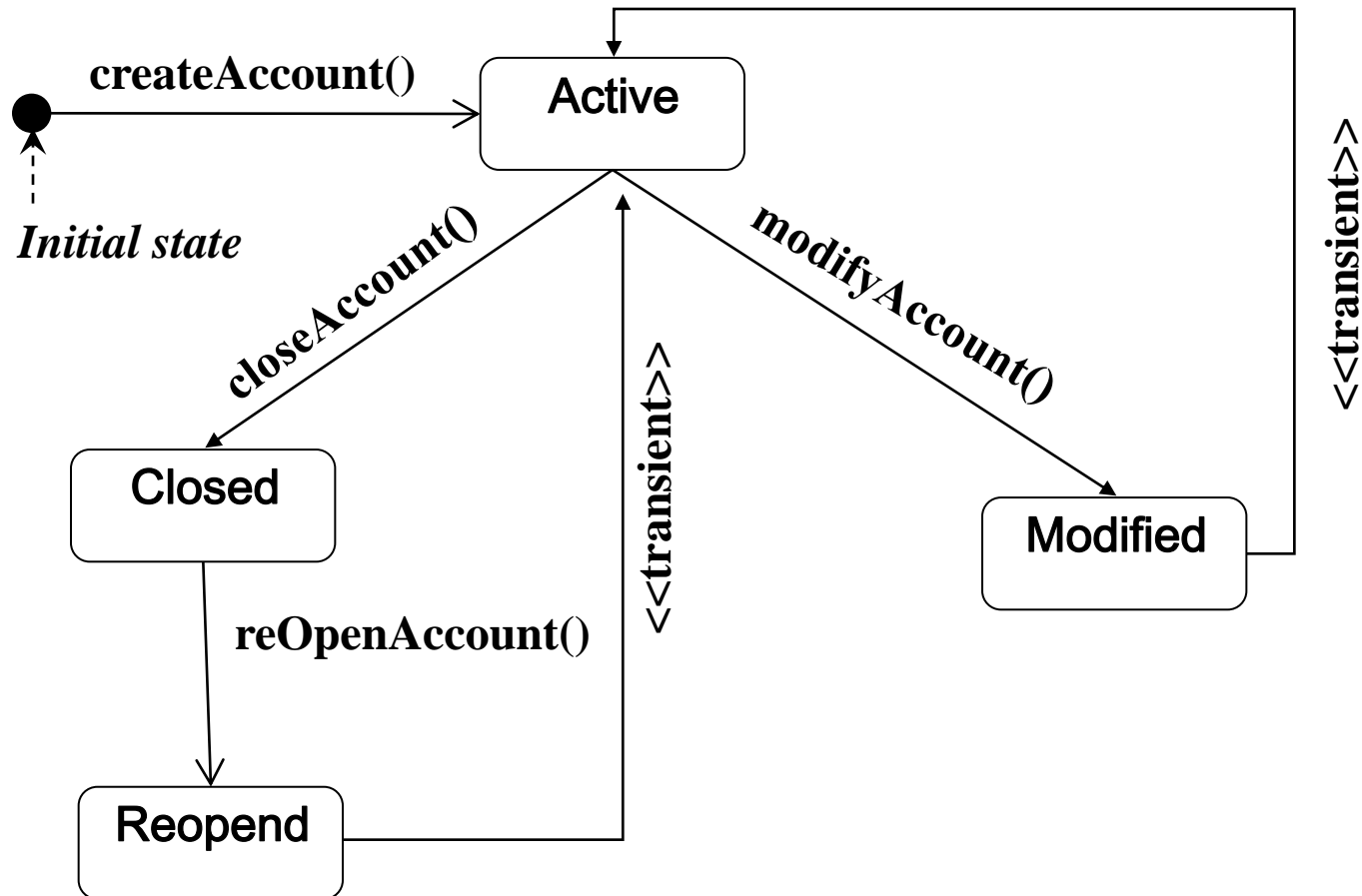
State : Account

- Initialize
- Active
- Modified
- Closed
- Reopened

Use Case : Account

- Create Account
- Modify Account
- Close Account
- Reopen Account

Statechart for Account



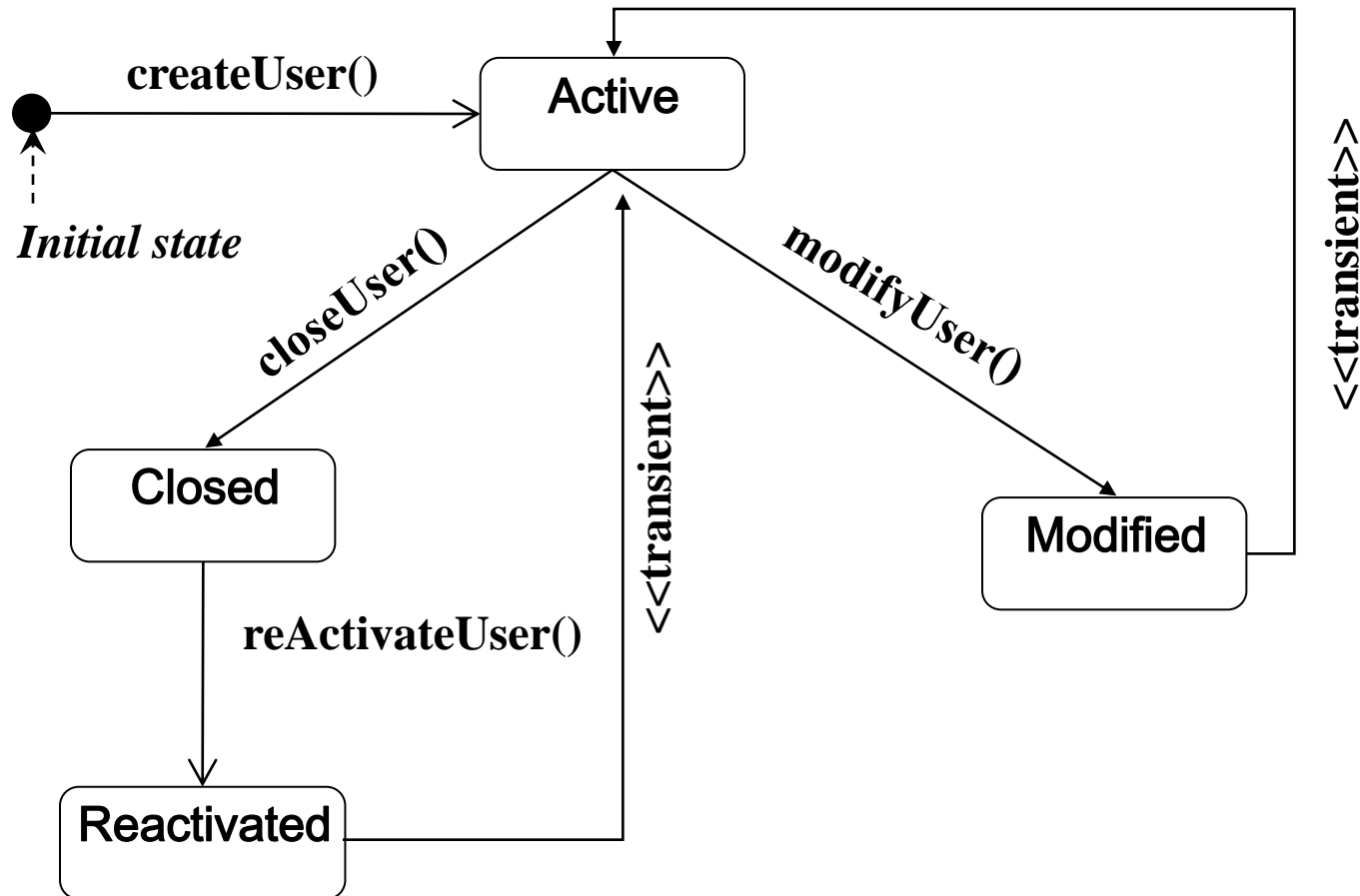
State : User

- Initialized
- Active
- Inactive
- Reactivated

Use Case : User

- Create User
- Modify User
- Delete User
- Reactivate User

Statechart for User



• ***Thanks to All***