Here’s a **use case diagram** in **PlantUML** format for managing bank shareholders. It includes use cases for registering shareholders, managing shares, distributing dividends, and generating reports.

plantuml

CopyEdit

@startuml

left to right direction

skinparam packageStyle rectangle

actor "Shareholder" as shareholder

actor "Bank Admin" as admin

actor "Auditor" as auditor

rectangle "Bank Shareholder Management System" {

usecase "Register Shareholder" as UC1

usecase "Update Shareholder Info" as UC2

usecase "Manage Shares" as UC3

usecase "Distribute Dividends" as UC4

usecase "Generate Financial Reports" as UC5

usecase "Audit Shareholder Records" as UC6

}

shareholder --> UC1

shareholder --> UC2

shareholder <-- UC4

admin --> UC1

admin --> UC2

admin --> UC3

admin --> UC4

admin --> UC5

auditor --> UC6

auditor --> UC5

@enduml

**Description of Use Cases**

1. **Register Shareholder**: Shareholders or admins can register new shareholders in the system.
2. **Update Shareholder Info**: Shareholders can update their personal details, and admins can oversee these updates.
3. **Manage Shares**: Admins manage the allocation, buying, or selling of shares.
4. **Distribute Dividends**: Admins calculate and distribute dividends to shareholders.
5. **Generate Financial Reports**: Admins generate reports for internal use, while auditors review them.
6. **Audit Shareholder Records**: Auditors access and review the shareholder data for compliance and accuracy.

Copy and paste this code into any **PlantUML editor** (like PlantUML's website or VS Code with the PlantUML extension) to generate the diagram. Let me know if you’d like further customization!

Here's a more **complex version** of the use case diagram for **bank shareholder management** in **PlantUML** format. It includes additional actors, interactions, and use cases such as notifications, share transfers, tax deductions, and compliance checks.

plantuml

CopyEdit

@startuml

left to right direction

skinparam packageStyle rectangle

actor "Shareholder" as shareholder

actor "Bank Admin" as admin

actor "Auditor" as auditor

actor "Tax Authority" as taxAuthority

actor "External System" as externalSystem

rectangle "Bank Shareholder Management System" {

usecase "Register Shareholder" as UC1

usecase "Update Shareholder Info" as UC2

usecase "Manage Shares" as UC3

usecase "Transfer Shares" as UC4

usecase "Distribute Dividends" as UC5

usecase "Generate Financial Reports" as UC6

usecase "Notify Shareholders" as UC7

usecase "Deduct Taxes from Dividends" as UC8

usecase "Perform Compliance Checks" as UC9

usecase "Integrate with External Systems" as UC10

usecase "Audit Shareholder Records" as UC11

}

shareholder --> UC1

shareholder --> UC2

shareholder --> UC4

shareholder <-- UC5

shareholder <-- UC7

admin --> UC1

admin --> UC2

admin --> UC3

admin --> UC4

admin --> UC5

admin --> UC6

admin --> UC7

admin --> UC8

admin --> UC9

admin --> UC10

auditor --> UC6

auditor --> UC9

auditor --> UC11

taxAuthority <-- UC8

externalSystem <-- UC10

@enduml

**Enhancements in Complexity**

1. **New Actors**:
   * **Tax Authority**: Interacts with dividend-related taxes.
   * **External System**: Integrates for data synchronization or regulatory purposes.
2. **Additional Use Cases**:
   * **Transfer Shares**: Allows shareholders to transfer shares between accounts.
   * **Notify Shareholders**: System sends notifications for important events like dividend distribution.
   * **Deduct Taxes from Dividends**: Ensures compliance with tax laws during dividend processing.
   * **Perform Compliance Checks**: Validates shareholder records against regulatory requirements.
   * **Integrate with External Systems**: Enables secure communication with external APIs or financial platforms.
3. **Relationships**:
   * Shareholders are more involved in direct actions like transferring shares and receiving notifications.
   * Admins oversee tax deductions, compliance checks, and integration with external systems.
   * Auditors focus on reports and compliance validation.

You can paste this code into a **PlantUML editor** to visualize the diagram. Let me know if you'd like further enhancements or explanations!