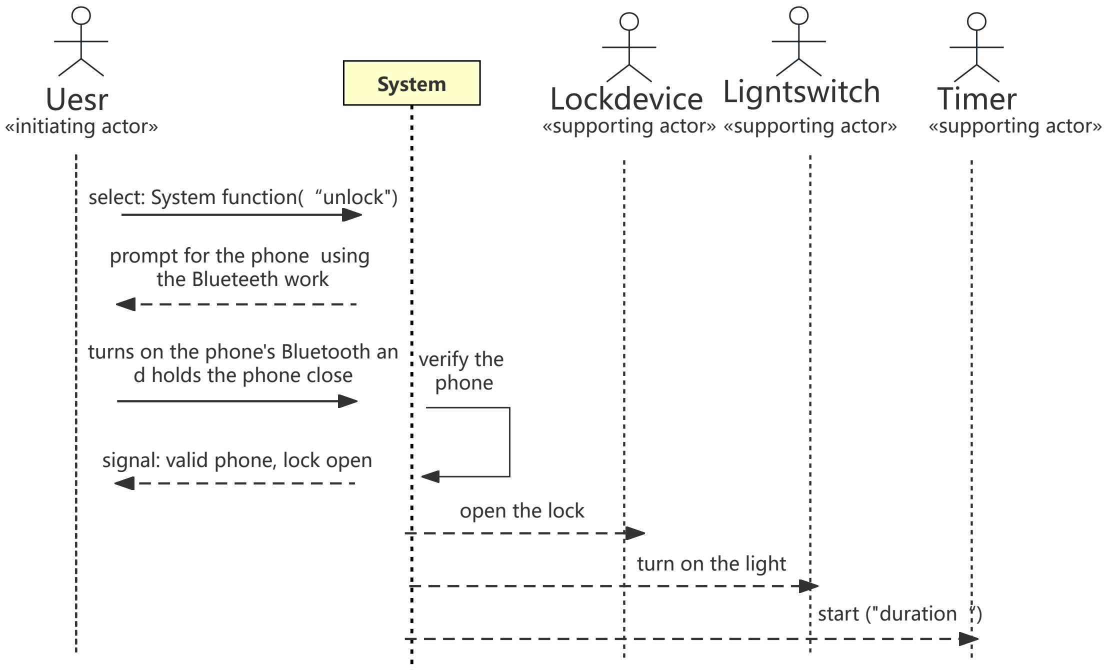
system sequence diagram

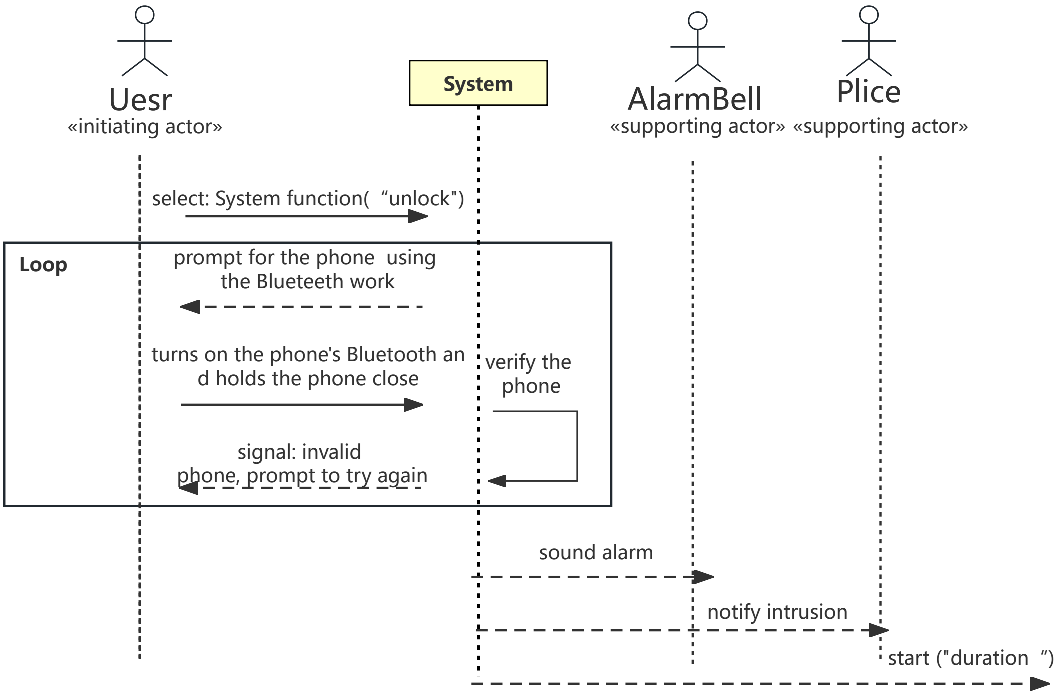
UC-1(Unlock)

Use case: Unlock



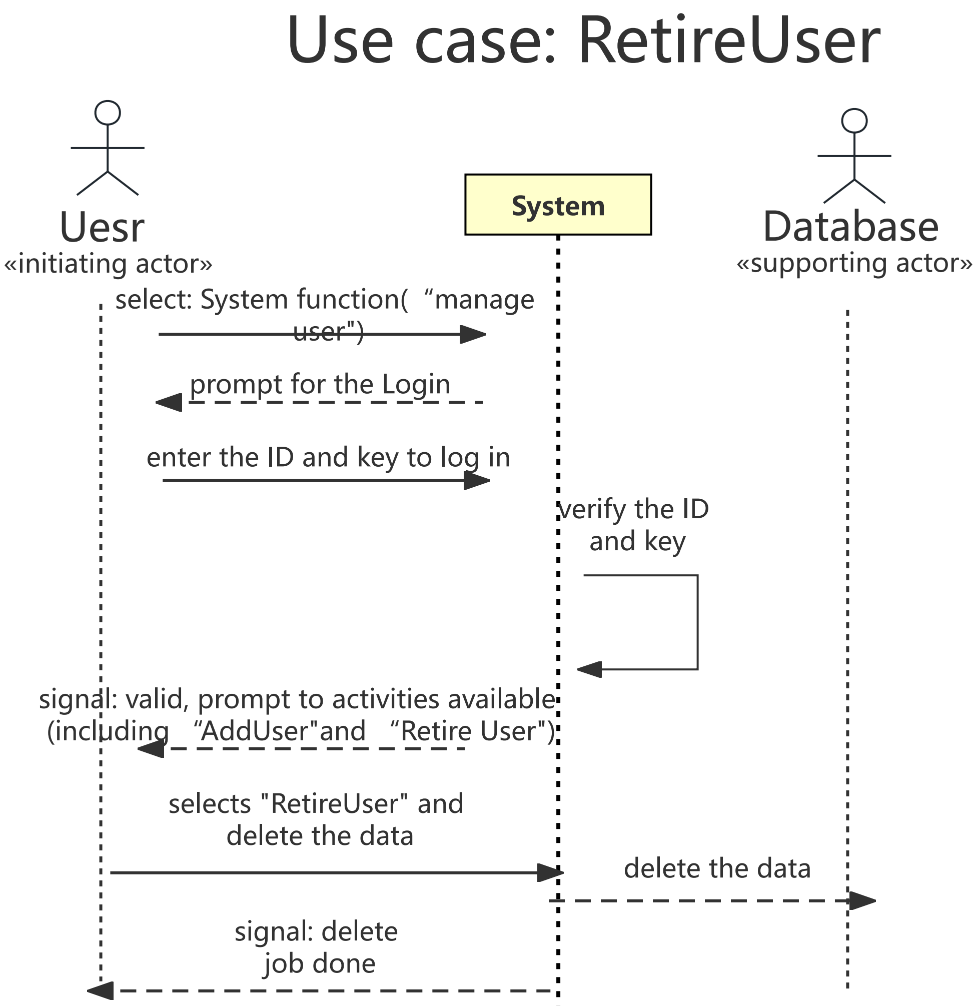
Main success scenario

Use case: Unlock


Uesr
«initiating actor»
System
AlarmBell
«supporting actor»
Plice
«supporting actor»
select: System function(“unlock")
Loop
prompt for the phone using the Blueteeth work
turns on the phone's Bluetooth and holds the phone close
verify the phone
signal: invalid phone, prompt to try again
sound alarm
notify intrusion
start ("duration ”)

Alternate scenario (burglary attempt)

UC-4(RetireUser)

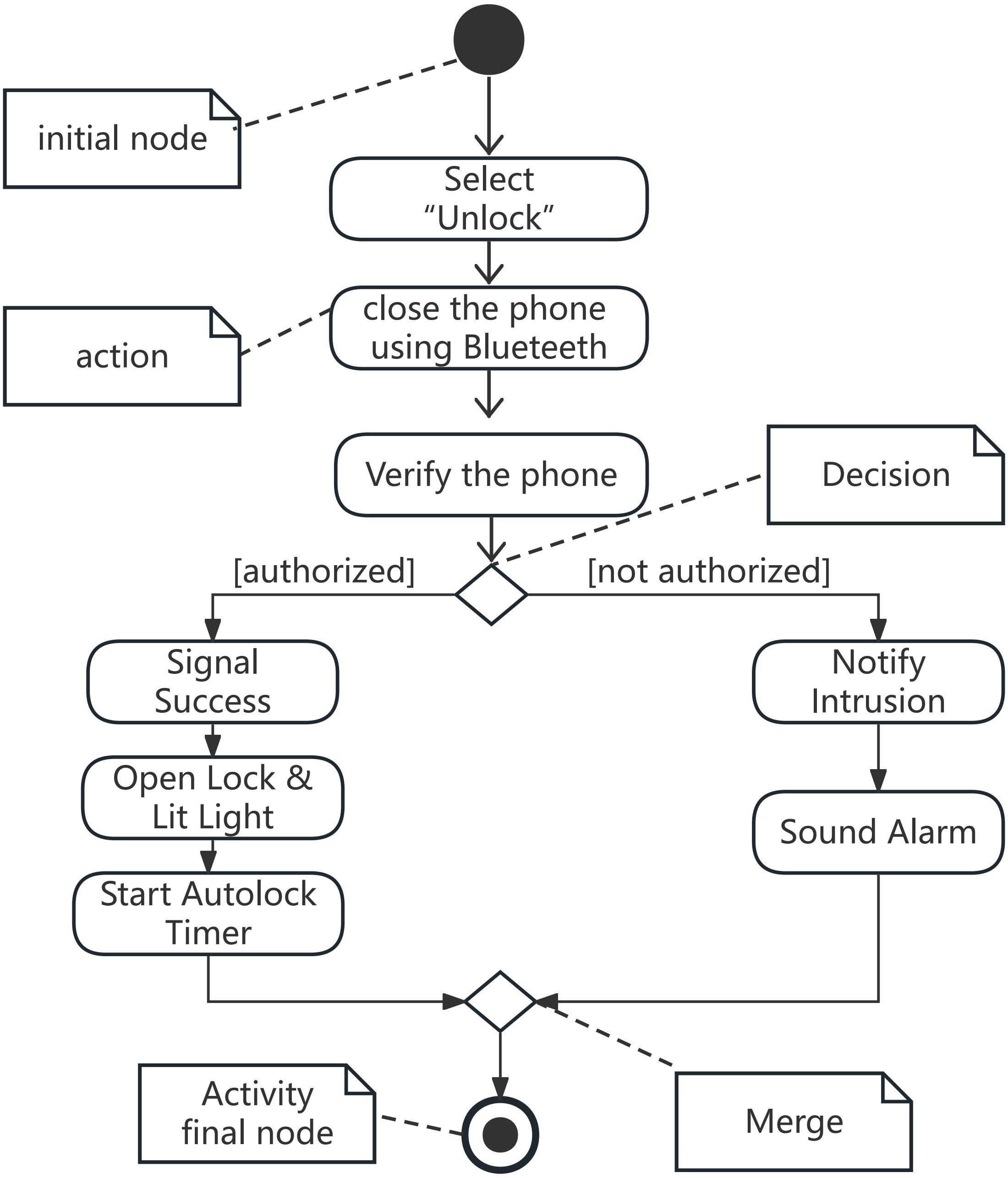


Use case: RetireUser
Uesr
«initiating actor»
System
Database
«supporting actor»
select: System function(“manage user")
prompt for the Login
enter the ID and key to log in
verify the ID and key
signal: valid, prompt to activities available (including “AddUser"and “Retire User")
selects "RetireUser" and delete the data
delete the data
signal: delete job done

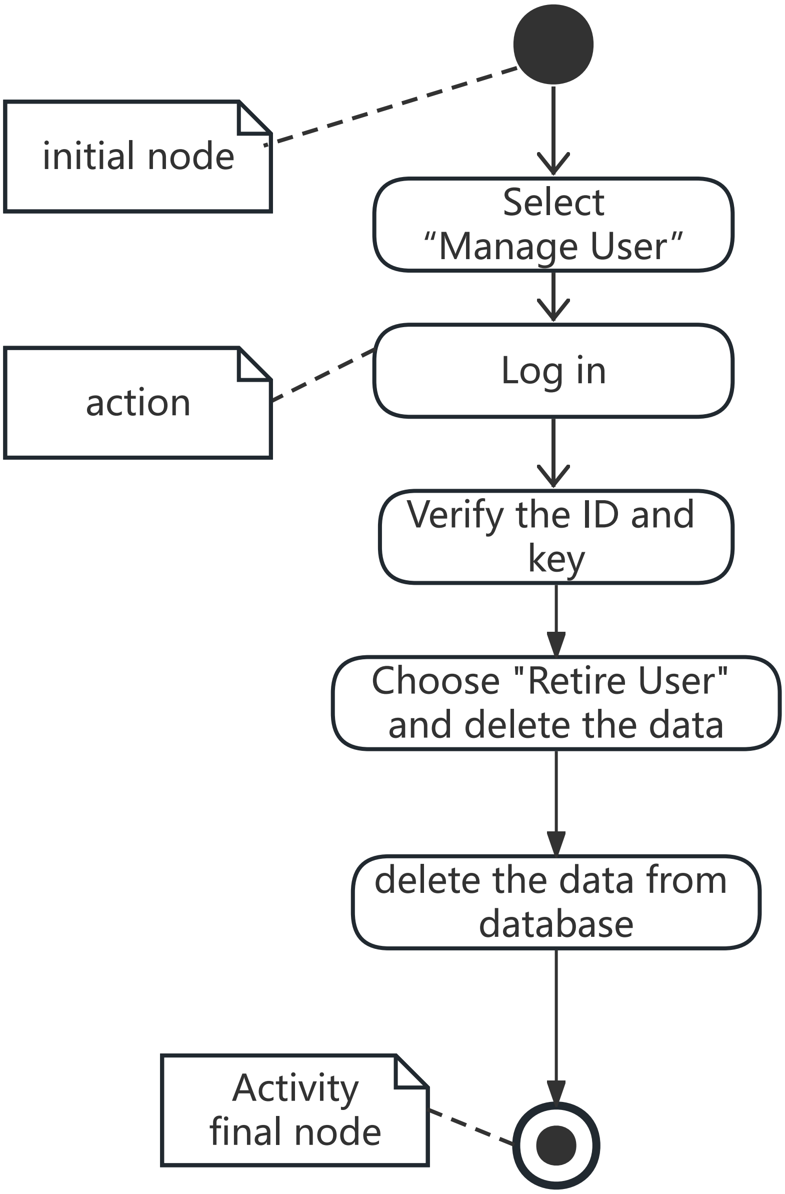
UC-4

Activity Diagram

UC-1(Unlock)

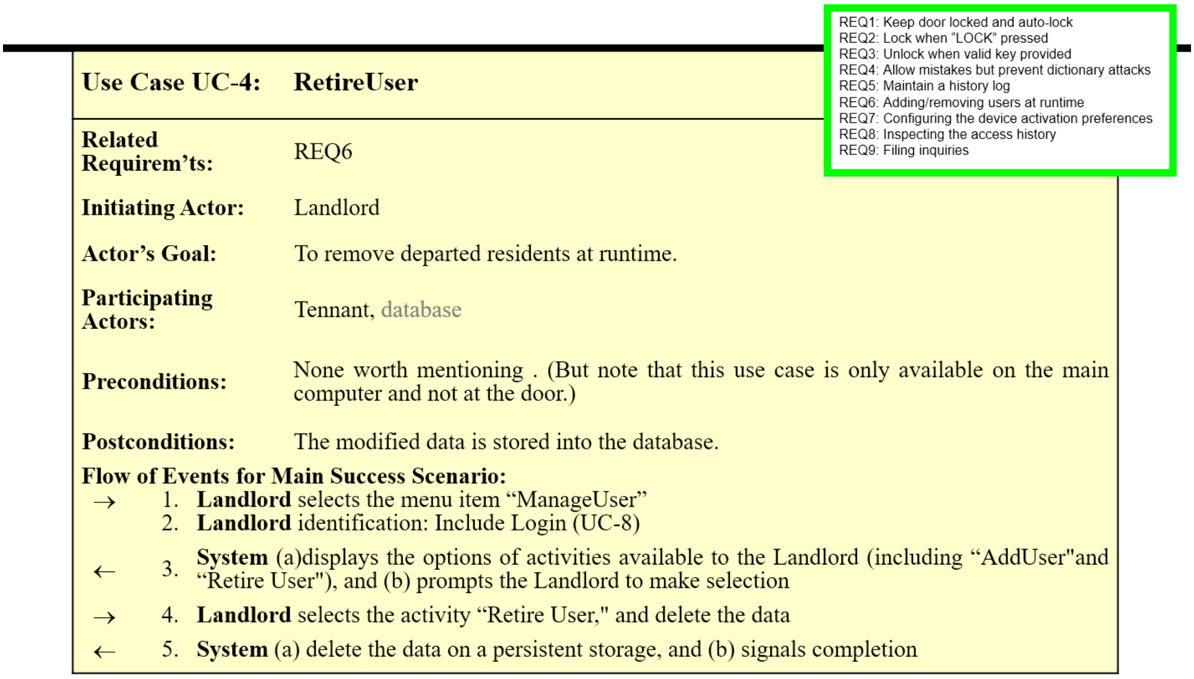


UC-4(RetireUser)



Domain Model

Use Case 4: RetireUser



REQ1: Keep door locked and auto-lock
REQ2: Lock when "LOCK" pressed
REQ3: Unlock when valid key provided
REQ4: Allow mistakes but prevent dictionary attacks
REQ5: Maintain a history log
REQ6: Adding/removing users at runtime
REQ7: Configuring the device activation preferences
REQ8: Inspecting the access history
REQ9: Filing inquiries

Use Case UC-4:	**RetireUser**
Related Requirem'ts:	REQ6
Initiating Actor:	Landlord
Actor's Goal:	To remove departed residents at runtime.
Participating Actors:	Tennant, database
Preconditions:	None worth mentioning . (But note that this use case is only available on the main computer and not at the door.)
Postconditions:	The modified data is stored into the database.

Flow of Events for Main Success Scenario:

→ 1. **Landlord** selects the menu item "ManageUser"
2. **Landlord** identification: Include Login (UC-8)

← 3. **System** (a)displays the options of activities available to the Landlord (including "AddUser"and "Retire User"), and (b) prompts the Landlord to make selection

→ 4. **Landlord** selects the activity "Retire User," and delete the data

← 5. **System** (a) delete the data on a persistent storage, and (b) signals completion

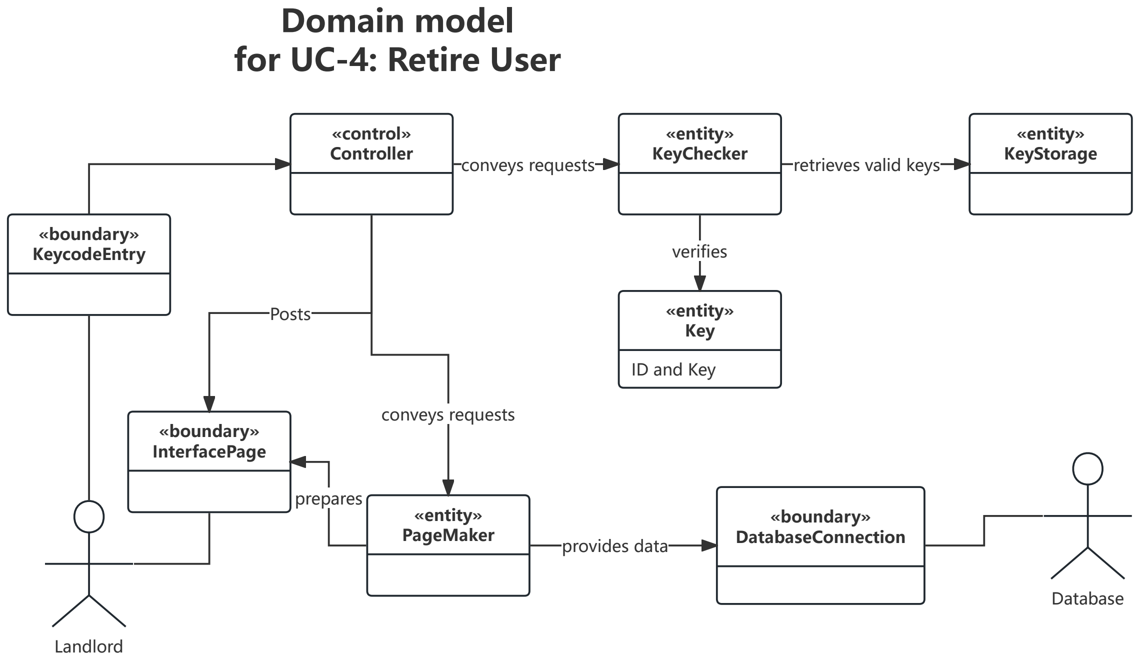
Extracting the Responsibilities

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Container for user's authentication data, such as pass-code, timestamp, etc.	K	Key
Rs3. Verify whether or not the key-code entered by the user is valid.	D	KeyChecker
Rs4. Container for the collection of valid keys associated with Landlord.	K	KeyStorage
Rs5. Render the retrieved records into an HTML document for sending to actor's Web browser for display.	D	Page Maker
Rs6. HTML document that shows the actor the current context, what actions can be done, and outcomes of the previous actions.	K	Interface Page
Rs7. Prepare a database query and retrieve the records from the database (from UC-4, Step 4).	D	Database Connection

Extracting the Associations

Concept pair	Association description	Association name
Controller <-> Page Maker	Controller passes requests to Page Maker and receives back pages prepared for displaying	conveys requests
Page Maker <-> Database Connection	Database Connection passes the retrieved data to Page Maker to render them for display	provides data
Page Maker <-> Interface Page	Page Maker prepares the Interface Page	prepares
KeyChecker <-> Keystorage	KeyChecker retrieves valid keys from Keystorage	retrieves valid keys
KeyChecker <-> Key	judge whether the key is valid.	verifies
Controller <-> KeyCheceker	judge the num of attempts whether is above the maxnum of attempts	conveys requests

Domain model



Domain model for UC-4: Retire User