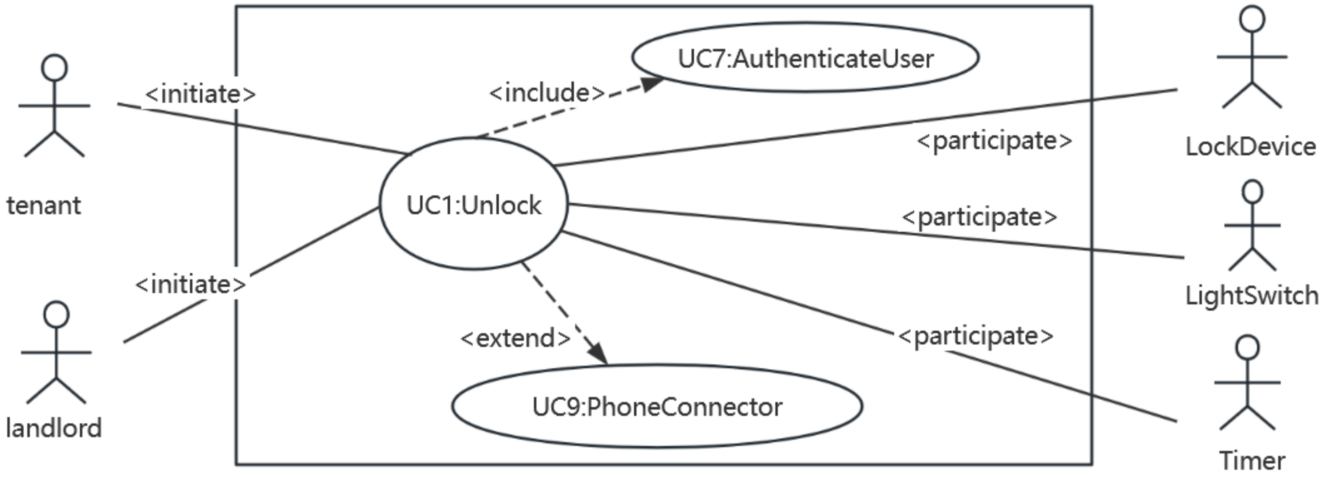
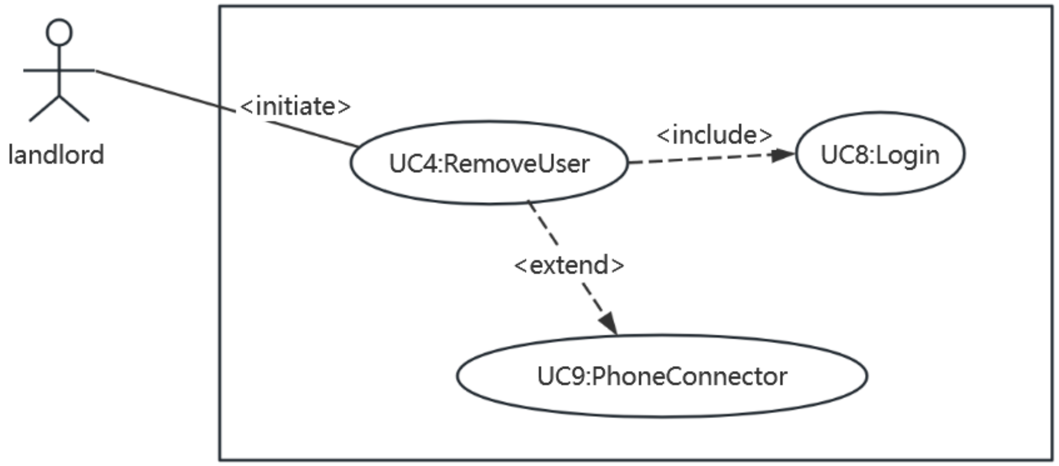
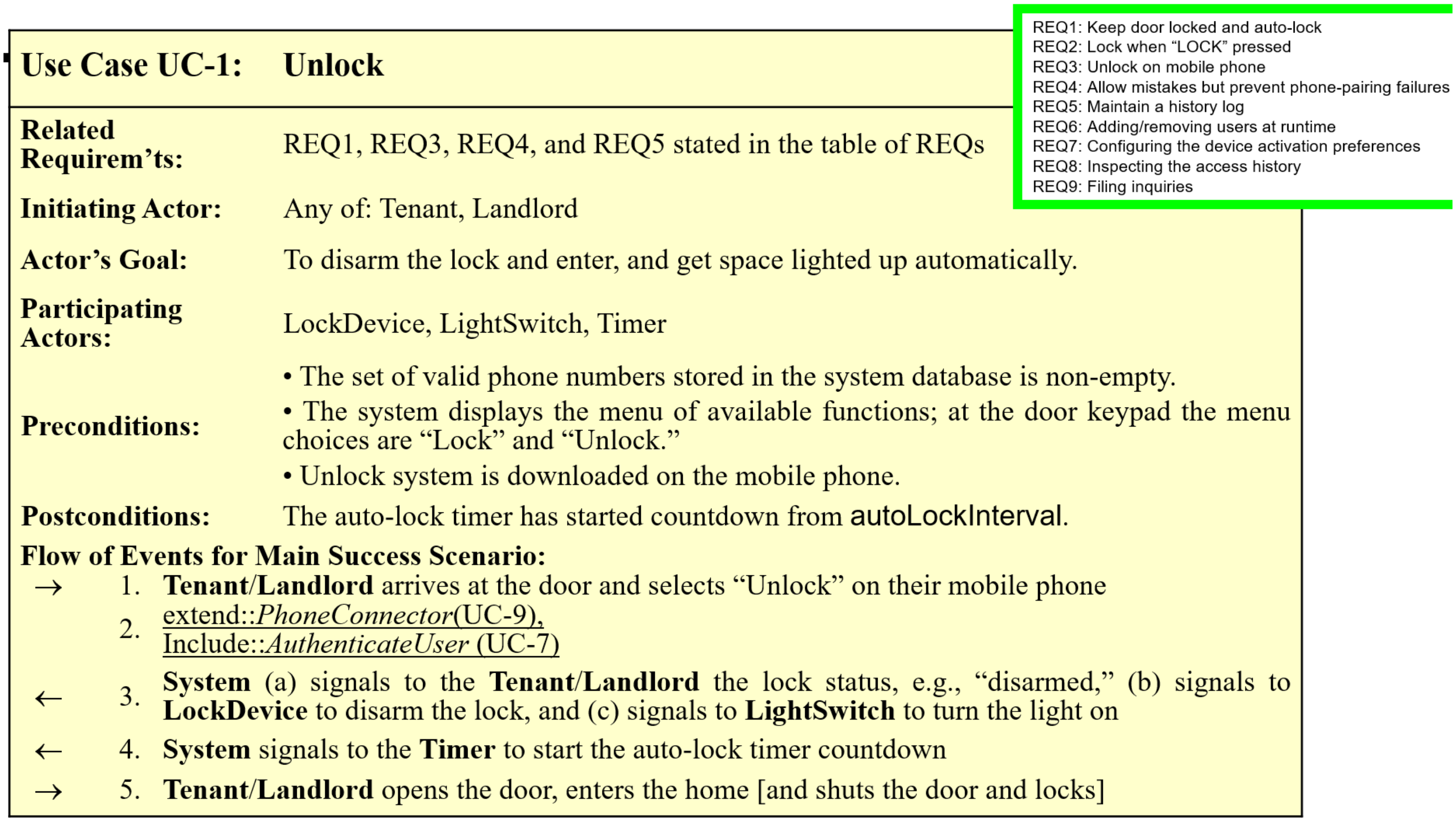
assignment_1

1.





2.



Use Case UC-1:	Unlock
Related Requirem'ts:	REQ1, REQ3, REQ4, and REQ5 stated in the table of REQs
Initiating Actor:	Any of: Tenant, Landlord
Actor's Goal:	To disarm the lock and enter, and get space lighted up automatically.
Participating Actors:	LockDevice, LightSwitch, Timer
Preconditions:	• The set of valid phone numbers stored in the system database is non-empty. • The system displays the menu of available functions; at the door keypad the menu choices are "Lock" and "Unlock." • Unlock system is downloaded on the mobile phone.
Postconditions:	The auto-lock timer has started countdown from autoLockInterval.

Flow of Events for Main Success Scenario:

→ 1. **Tenant/Landlord** arrives at the door and selects "Unlock" on their mobile phone

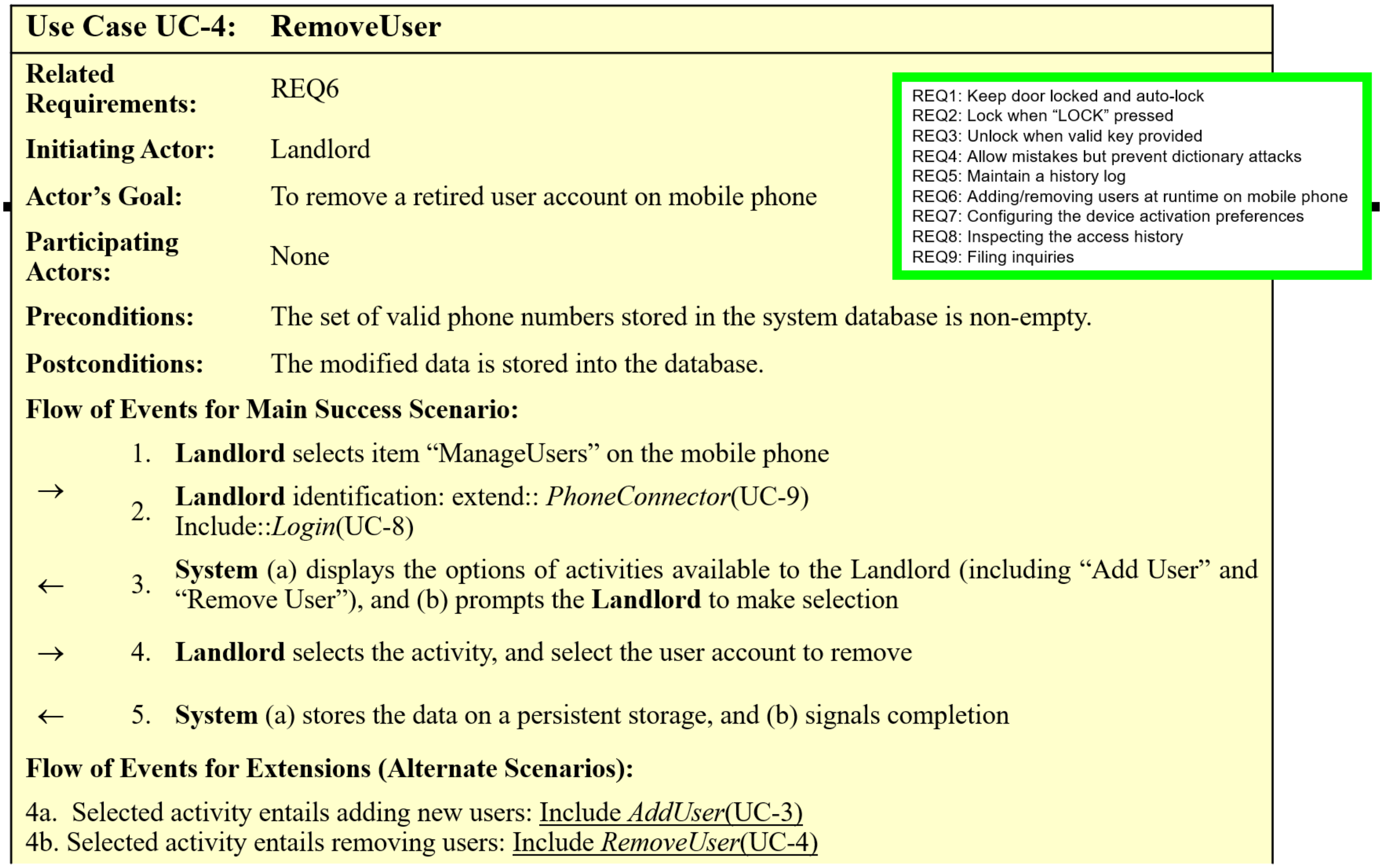
2. extend::*PhoneConnector*(UC-9),
Include::*AuthenticateUser* (UC-7)

← 3. **System** (a) signals to the **Tenant/Landlord** the lock status, e.g., "disarmed," (b) signals to **LockDevice** to disarm the lock, and (c) signals to **LightSwitch** to turn the light on

← 4. **System** signals to the **Timer** to start the auto-lock timer countdown

→ 5. **Tenant/Landlord** opens the door, enters the home [and shuts the door and locks]

REQ1: Keep door locked and auto-lock
REQ2: Lock when "LOCK" pressed
REQ3: Unlock on mobile phone
REQ4: Allow mistakes but prevent phone-pairing failures
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:	RemoveUser
Related Requirements:	REQ6
Initiating Actor:	Landlord
Actor's Goal:	To remove a retired user account on mobile phone
Participating Actors:	None
Preconditions:	The set of valid phone numbers stored in the system database is non-empty.
Postconditions:	The modified data is stored into the database.

Flow of Events for Main Success Scenario:

1. **Landlord** selects item "ManageUsers" on the mobile phone

→ 2. **Landlord** identification: extend:: *PhoneConnector*(UC-9)
Include::*Login*(UC-8)

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

→ 4. **Landlord** selects the activity, and select the user account to remove

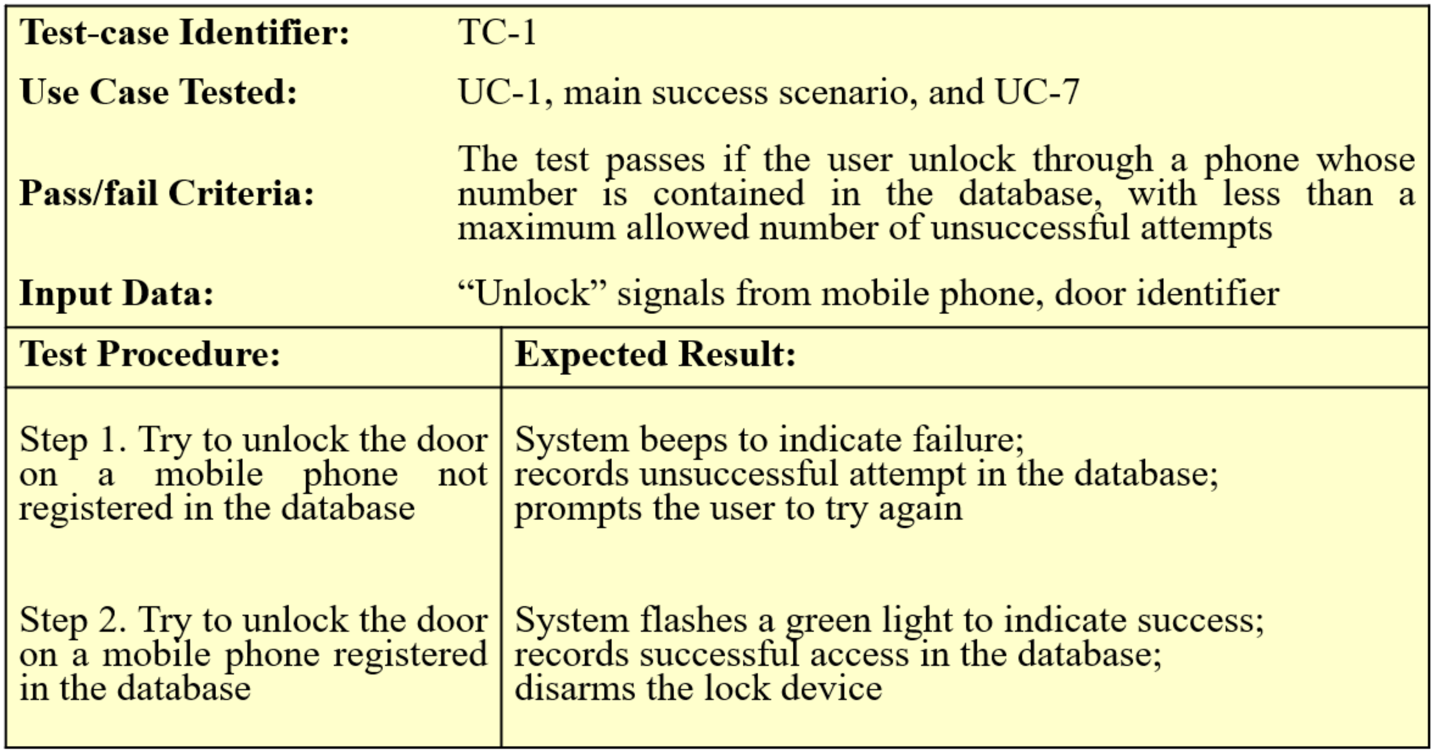
← 5. **System** (a) stores the data on a persistent storage, and (b) signals completion

Flow of Events for Extensions (Alternate Scenarios):

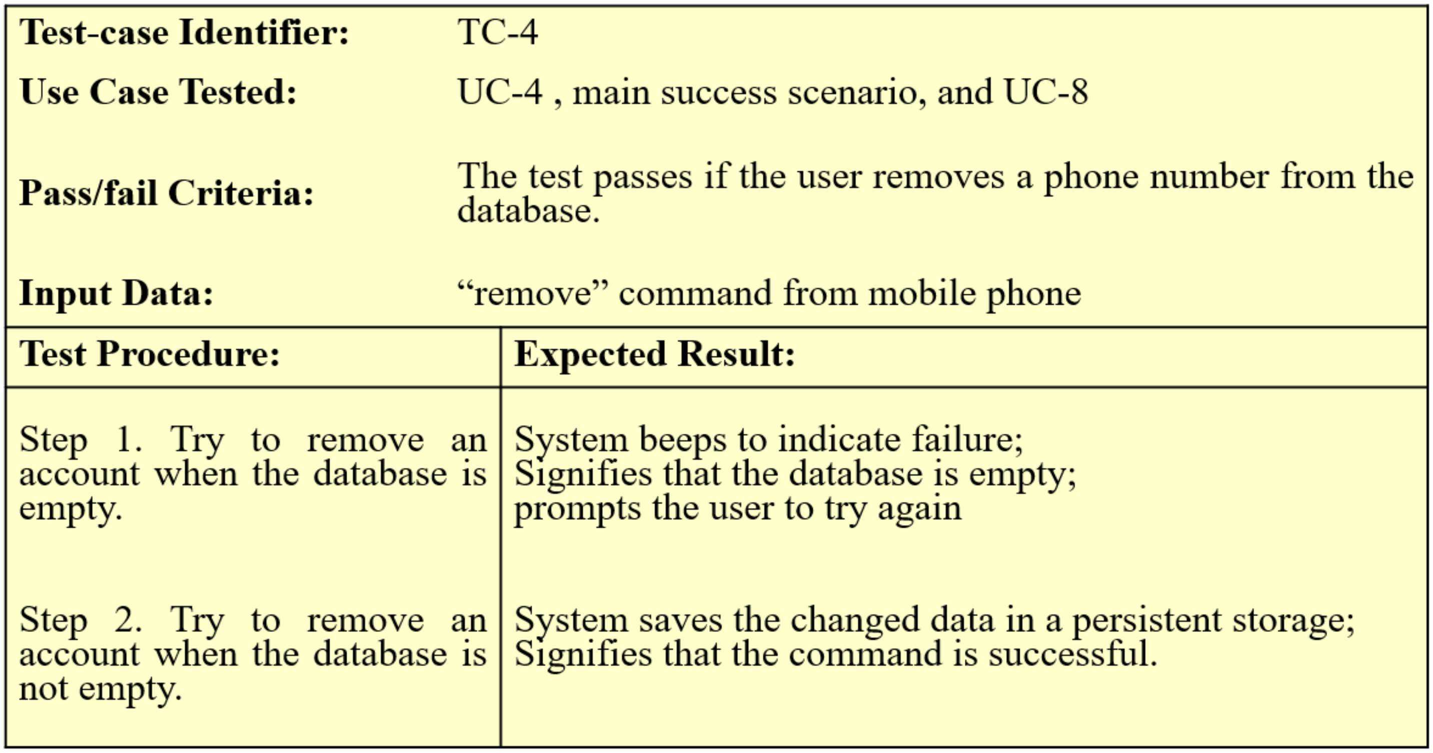
4a. Selected activity entails adding new users: Include *AddUser*(UC-3)
4b. Selected activity entails removing users: Include *RemoveUser*(UC-4)

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 on mobile phone
REQ7: Configuring the device activation preferences
REQ8: Inspecting the access history
REQ9: Filing inquiries

3.



Test-case Identifier:	TC-1
Use Case Tested:	UC-1, main success scenario, and UC-7
Pass/fail Criteria:	The test passes if the user unlock through a phone whose number is contained in the database, with less than a maximum allowed number of unsuccessful attempts
Input Data:	“Unlock” signals from mobile phone, door identifier
Test Procedure:	**Expected Result:**
Step 1. Try to unlock the door on a mobile phone not registered in the database	System beeps to indicate failure; records unsuccessful attempt in the database; prompts the user to try again
Step 2. Try to unlock the door on a mobile phone registered in the database	System flashes a green light to indicate success; records successful access in the database; disarms the lock device



Test-case Identifier:	TC-4
Use Case Tested:	UC-4 , main success scenario, and UC-8
Pass/fail Criteria:	The test passes if the user removes a phone number from the database.
Input Data:	“remove” command from mobile phone
Test Procedure:	**Expected Result:**
Step 1. Try to remove an account when the database is empty.	System beeps to indicate failure; Signifies that the database is empty; prompts the user to try again
Step 2. Try to remove an account when the database is not empty.	System saves the changed data in a persistent storage; Signifies that the command is successful.