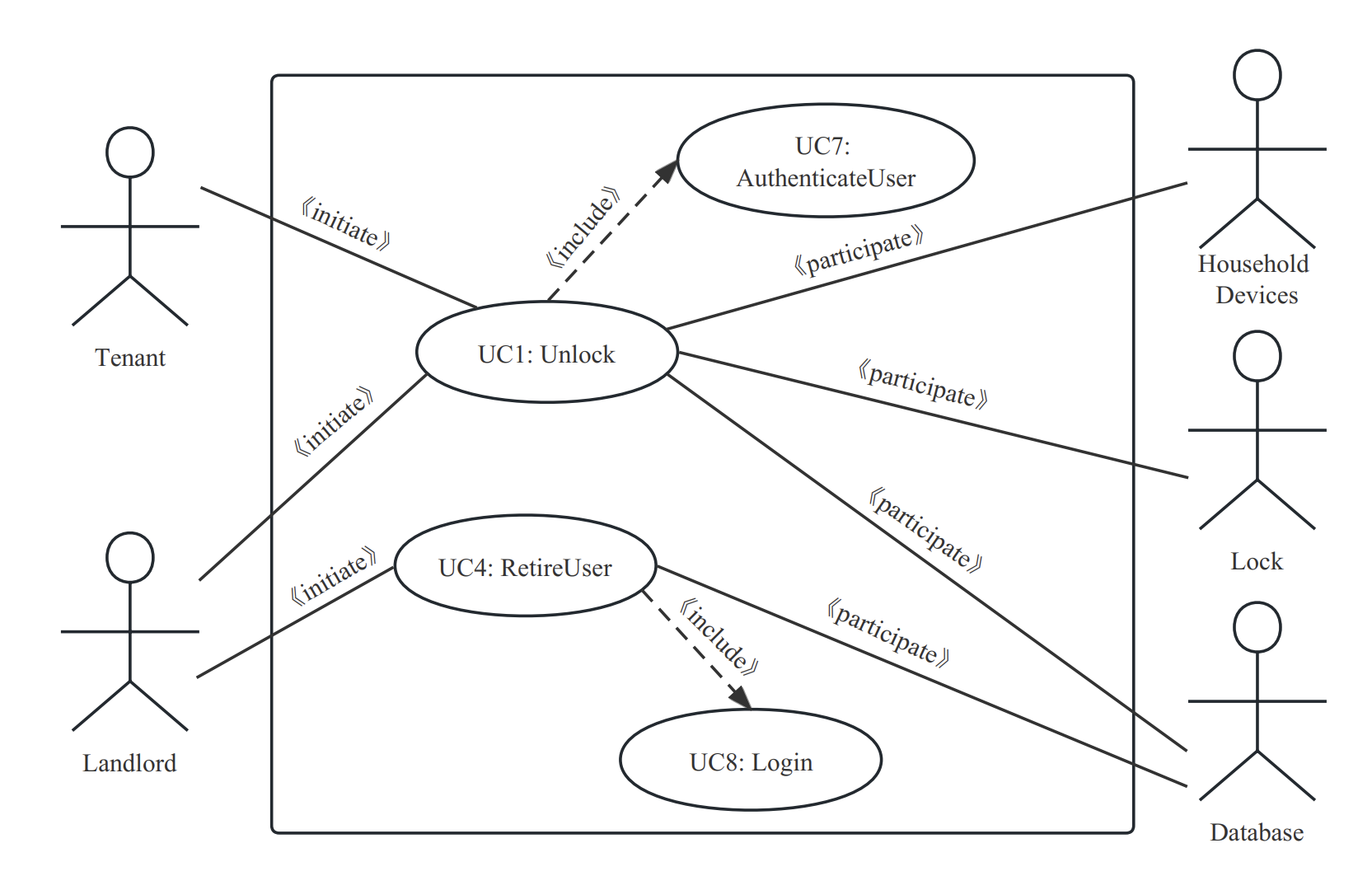
**Course Assignment (Mini Project I-1)**

█ 谭诚 10215101460

1. Draw use case diagram for UC-1 (Unlock) and UC-4 (RetireUser)



1. Write the use case schemas of UC-1 and UC-4

|  |  |  |
| --- | --- | --- |
| **Use Case UC-1: Unlock** | | |
| **Related Requirem’ts:** | | REQ1, REQ2, 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. The door will be locked after unlocking in few minutes. |
| **Participating Actors:** | | Lock, Household Devices, Database |
| **Preconditions:** | | • The set of valid keys 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.” |
| **Postconditions:** | | • If the user is valid in authentication, the door will be unlocked. The auto-lock timer has started countdown from autoLockInterval.  • Otherwise, the door will NOT be unlocked. If the user failed too many times in authentication, block his phone in Bluetooth. |
| **Flow of Events for Main Success Scenario:** | | |
| → | 1. **Tenant/Landlord** arrives at the door and selects the menu item “Unlock” 2. include::AuthenticateUser (UC-7) | |
| ← | 1. **System** authenticates the **Tenant/Landlord**’s verification. If it’s valid, signal the **Lock** to be disarmed, and open the **Household Devices**. Otherwise, notify the failure information. Moreover, if the user tried too many times, ban him. | |
| ← | 1. **System** signals to the **Timer** to start the auto-lock timer countdown. | |
| → | 1. **Tenant/Landlord** opens the door, enters [and shuts the door and locks]. | |

\*The table of REQs is recognized as page 3 in file *lab\_3 User cases.pdf*.

|  |  |  |
| --- | --- | --- |
| **Use Case UC-4: RetireUser** | | |
| **Related Requirem’ts:** | | REQ2, REQ3 stated in the table of REQs\* |
| **Initiating Actor:** | | Landlord |
| **Actor’s Goal:** | | Retire a user’s access authentication. For example, if a tenant’s contract is expired, his access should be revoked. |
| **Participating Actors:** | | Database |
| **Preconditions:** | | • The set of valid keys stored in the system database is non-empty.  • The target user is existed in database. |
| **Postconditions:** | | • Target user’s access vanished. |
| **Flow of Events for Main Success Scenario:** | | |
| → | 1. The **Landlord** wants to revoke a **Tenant**’s access, enters the **System**. | |
| ← | 1. The **System** requires login to make sure it is the **Landlord** own. | |
| ← | 1. If the **Landlord** logins correctly, **System** will revoke the target **Tenant**’s access. Otherwise, **System** will require another valid login. | |
| → | 1. The **Landlord** ensures the target **Tenant**’s access is revoked and gets satisfied. | |

\*The table of REQs is recognized as page 3 in file *lab\_3 User cases.pdf*.

1. Write the acceptance tests for UC-1 and UC-4

|  |  |  |
| --- | --- | --- |
| **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 enters a key that is contained in the database, with less than a maximum allowed number of unsuccessful attempts, and any key that is not contained will NOT open the door, and the user will be blocked if he tried too many times.  • If any statement above is broken, then the test fails. | |
| **Input Data:** | Password, door identifier | |
| **Test Procedure:** | | **Expected Result:** |
| Step 1. Type in the correct keycode and door identifier  Step 2. Type in an incorrect keycode and door identifier  Step 3. Type in many incorrect keycodes and door identifiers | | • System indicates success; records successful access in the database; disarms the lock device.  • System beeps to indicate failure; records unsuccessful attempt in the database; prompts the user to try again.  • System beeps to indicate failure; records offensive access record in the database; block the user. |

|  |  |  |
| --- | --- | --- |
| **Test-case Identifier:** | TC-4 | |
| **Use Case Tested:** | UC-4, main success scenario, and UC-8 | |
| **Pass/fail Criteria:** | • The test passes if: The landlord needs to login correctly, and then try to retire an existed tenant. After that, the target tenant will no longer be able to unlock the lock.  • This is a strict serial process. If the user could login the System with wrong info, or the user retired could still unlock a lock, the test fails. | |
| **Input Data:** | System login info, target tenant | |
| **Test Procedure:** | | **Expected Result:** |
| Step 1. Type in the correct login info to enter the System  Step 2. Try to retire an existed user  Step 3. Try to retire a not existed user  Step 4. Type in the incorrect login info once | | • The System indicates success and record the access, showing the user panel to the landlord.  • The target user will no longer be able to unlock, and get deleted in DB.  • The System will tell landlord it’s not existed  • The System indicates fail and record the access. |