Activity 2 - Use case documentation

Sec 1 (Ben Kweon)

1. Log in the system
    - Use case number: 1
    - Use case name: Log in the system
    - Actor: The landlord
    - Precondition: The user needs to have a valid account
    - Summary: Authenticate user before starting the program so that unauthenticated users cannot use the program.
    - Scenario:
        1. Ask input
            - Input username
            - Input password
        2. Validation
            - Does username exist
            - Does the stored password and the given password match
    - Exception:
        - Scenario 2 => When user credential does not match
    - NonFunctional:
        - Interaction with database is needed to save userdata
        - Passwords should be saved and encrypted.
        - Input validation needed for both username and password(no whitespace, … etc)
2. Start the program
    - Use case number: 2
    - Use case name: Start the program
    - Actor: Log in the system (usecase)
    - Summary: Ask user to choose options between insert, display, and quit
    - Scenario:
        1. Display the menu
        2. Ask user input
        3. Actor would choose either insert, display or quit
    - Exception
        - Scenario 3 => If the userinput is neither ‘i’, ‘d’, or ‘q’, ask input again
    - Non Functional
        - Input validation needs to cause no error (otherwise the program will crash)
        - Use one variable to store the current userinput

Sec 2 (Yuwei Du)

3. Add new tenant

- Use case number: 3
- Use case name: Add new tenant
- Summary: Adding a new tenant to the system
- Actor: The landlord
- Precondition: The tenant is new here and the apt is empty.
- Scenario:
    1. Enter input data surface
    2. Enter to add tenant surface
    3. Enter tenant’s name
    4. Enter tenant’s apt number
    5. Back to input data surface
- Exception

  4 -The tenant has existed in the system or apt is occupied.

- Postcondition: New tenant added to the tenant list
- NonFunctional: interact with input surface

4. Input a rental payment

- Use case number: 4
- Use case name: Input a rental payment
- Summary: Enter the rental payment amount
- Actor: The landlord
- Precondition: The tenant is in the system, and is currently living in the apt
- Scenario:
    1. Enter ‘r’ to enter record rent payment page
    2. Enter tenant’s name to enter his/her page
    3. Enter amount paid
    4. Enter the month for the payment
    5. Back to main page
- Exception:

  The tenant is not in the system, cannot exceed the payment it should be paid.

- Postcondition: input rental payment data to record
- NonFunctional: interact with paymentRecord

Sec 3

5. Display the Tenant List

- Use case number: 5
- Use case name: Display the Tenant List
- Actor: The landlord
- Summary: display tenant’s list, name and apt number
- Scenario:
    1. Enter ‘t’ to display tenant’s list

2. Access to tenant Record
3. Display tenant’s list

- Exception:

  3. tenantList is empty

- Postcondition: Display the tenantList
- NonFunctional: interact with tenant record

6. Display the Rent Record

- Use case number: 6
- Use case name: Display the Rent Record
- Actor: The Landlord
- Summary: Ask user to choose options between insert, display, and quit
- Scenario:
  1. Enter ‘r’ to display rent record
  2. Display each RentRow stored in rent record
- Exception:
  - Scenario 2 => If the rent record does not contain any RentRow, print out a warning message and go back to the ask user input stage.
- NonFunctional:
  - We implement a caching system for display so that we don’t have to use the for loop everytime we call the display function.

Sec 4 (Hoang-Uyen Tran)

7. Display the Expense Record

- Use case number: 7
- Use case name: Display Expense Record
- Summary: Display all the finances fees according to the tenants
- Actor: The landlord
- Precondition: Need to specify the budget category to which payments to be display
- Scenario:
  1. Display general information
  2. budget category with according date, month
- NonFunctional:
  - ○ tenant’s name should be documentation

8. Display the Annual Summary

- Use case number: 8
- Use case name: Display Annual Summary
- Summary: Record annual expenses
- Actor: annual report

- Precondition: need datas from rent income and expense record
- Scenario:
    1. Display table
        - display the amount of summed rents
    2. Display budget
        a. Total of expenses for each budget category
    3. Display balance
        a. Display profit/ loss for the year to date