

Project-2 Design - Dargo Wang

1. Analysis:

Use case	Log-in
Description	Idle state. For anyone to login or quit.
Actors	Any user
Assumptions	The system will check user's password and recognize its identification.
steps	<ol style="list-style-type: none">1. Prompt the user to choose login or quit.2. If choosing login, prompt user to input its user name and password.3. Check if the password is correct. If correct, login. If incorrect, re prompt input.4. Check if the user is an administrator or staff, and jump to that use case.

Use case	Administration management
Description	After a user logged in as an administrator, the system will jump into this use case.
Actors	Administrator
Assumptions	The system will prompt the administrator to choose an option of what it wants to manage.
steps	<ol style="list-style-type: none">1. Print out the options for administrator to choose from2. Prompt administrator to choose.3. Jump to that case.

Use case	Branch staff management
Description	After a user logged in as a branch staff, the system will jump into this use case.
Actors	Branch staff
Assumptions	The system will prompt the staff if he wants to change password or manage client accounts.
steps	<ol style="list-style-type: none">1. Print out the options for staff to choose from2. Prompt staff to choose.3. Jump to that case.

Use case	Client account management
----------	---------------------------

Description	After a branch staff choosing to manage client accounts, the system will just into this case.
Actors	Branch staff
Assumptions	The system will prompt the staff to choose an option of what it wants to manage about clients.
steps	<ol style="list-style-type: none"> 1. Print out the options for staff to choose from 2. Prompt staff to choose. 3. Jump to that case.

2. Design:

Class name	Login
Purpose	Idle state. For uses login or quit.
Member variables	String name; String password; int option;
functions	welcome(); menu(); login(); quit();
Depends on	/
uses	Administrator Staff

Class name	Administrator
Purpose	Active state. For administrator to manage.
Member variables	int option;
functions	adMenu(); addStaff(); deleteStaff(); displayStaff(); changePassword(); exit();
Depends on	Login
uses	ClientManagement

Class name	Staff
Purpose	Active state. For staff to choose.
Member variables	int option;
functions	staffMenu(); changePassword(); clientManage(); exit();
Depends on	Login
uses	ClientManagement

Class name	ClientManagement
Purpose	Active state. For staff to manage clients.
Member variables	int option;
functions	clientMenu(); addClient(); editClient(); manageClient(); save(); exit();
Depends on	Staff
uses	/

3. Test

1. Password test: 2 tests. Test if a user enters his name and password correctly or incorrectly, the system will give the expected response.
2. addStaff() test: 1 test. Test if the system can successfully add a staff by administrator.
3. deleteStaff() test: 1 test. Test if the system can successfully delete a staff by administrator.
4. displayStaff() test: 1 test. Test if the system can successfully display a staff information.

5. `changePassword()` test: : 2 tests. Test if the system can successfully change password by an administrator and by a staff.
6. `addClient()` test: 1 test. Test if the system can successfully add a client by staff.
7. `editClient()` test: 1 test. Test if the system can successfully edit a client information by staff.
8. `manageClient()` test: 2 tests. Test if the system can successfully manage a client account if the account exists, or give proper response if the account does not exists.
9. `save()` test: 1 test. Test if the system can successfully save a client information by staff.
10. `quit()` test: 4 tests. Test if the system quit properly in 4 classes.