

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

# Software Requirement Specification

Version 1.0

<Project Name>

Subject: <Name of subject>

<Group Number>

<List of participants>

*Hanoi, <month, year>*

*<All notations inside the angle bracket are not part of this document, for its purpose is for extra instruction. When using this document, please erase all these notations and/or replace them with corresponding content as instructed. >*

*<This document, written by Asst. Prof. NGUYEN Thi Thu Trang, is used as a case study for student with related courses. Any modifications and/or utilization without the consent of the author is strictly forbidden>*

## Table of contents

Table of contents.....	1
1 Introduction.....	3
1.1 Objective.....	3
1.2 Scope.....	3
1.3 Glossary.....	3
1.4 References.....	4
2 Overall Description.....	5
2.1 Actors.....	5
2.2 Overall use case diagram.....	5
2.3 Decomposition of Use cases.....	6
2.3.1 Use case diagram for “Administer users”.....	6
2.3.2 Use case diagram for “Manage group of user”.....	7
2.3.3 Use case diagram for “Grant permission to using functions”.....	8
2.4 Business processes.....	8
2.4.1 Process of software utilization.....	8
2.4.2 Process of group management.....	9
2.4.3 Process of user management.....	10
2.4.4 Process of function management.....	11
3 Specific requirements.....	13
3.1 Specification of Use case UC001 - “Log in”.....	13
3.2 Specification of Use case UC002 - “Create menu”.....	14
3.3 Specification of Use case UC003 - “Registration”.....	14
3.4 Specification of Use case UC004 - “Search for user”.....	16
3.5 Specification of Use case UC005 - “CRUD function”.....	17
4 Supplementary specification.....	20
4.1 Functionality.....	20

4.2	Usability.....	20
4.3	Other requirements.....	20

# 1 Introduction

## 1.1 Objective

This document presents the detailed description for User management subsystem, user group and their usable function at run time. This document also describes the objectives and features of the system, interfaces and constraints of the system in response to external action.

This document is for stakeholders and related software developers.

## 1.2 Scope

In reality, any software needs to have the function of managing users or group of users, and dynamically granting permissions for usage of certain functions in the system.

The software's goal includes creating users right managements, role and function available to users at runtime. Users can register accounts then use it to log in for access to system functions. Logging in is done via either using registered account in the system or Facebook account. Any users can update their personal info in the systems. Should the user forget the password, he/she can request the system to reconfigure the password using a link attached with token to be sent to his/her registered email.

The administrators can disable an account, or request the user to change their password at a certain time or periodically. Administrator is able to grant certain role to a user. A user may possess many roles in the software. Each user's role is configured to be used with some functions. Each function may be utilised by many roles of users. If a new function is added, Administrator needs to provide information for this function in order to manage it using the software. After a successful log in, depends on the user's role, the software will automatically create a menu according to the granted action of such user. Each time a function on the menu is selected, the corresponding interface will be presented.

## 1.3 Glossary

*<Listing and explaining the terms appearing in the software's profession and this documents. Any assumption of the reader's prior knowledge or experience on the subject is ill advised>*

## **1.4 References**

*<Listing the referenced material used in this documents, including the one related to the project>*

## 2 Overall Description

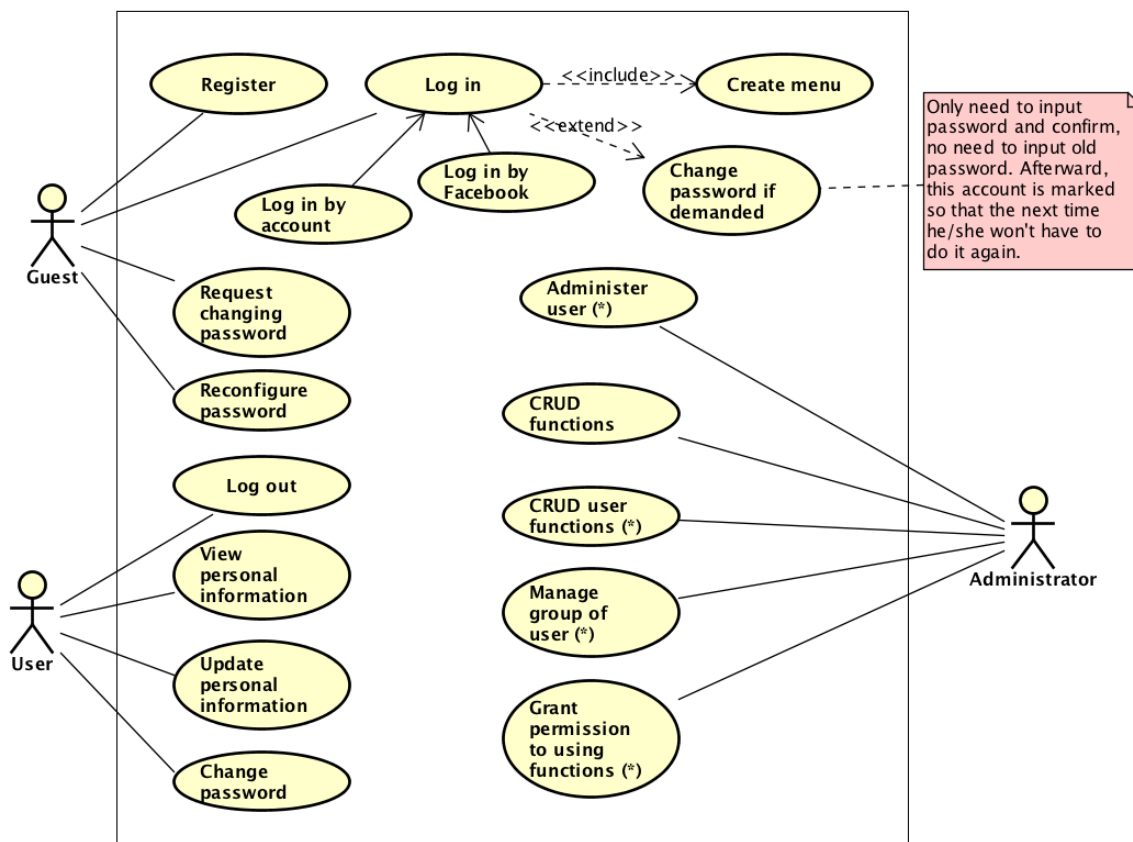
### 2.1 Actors

There are 3 main actors in the system: Guest, Users, and Administrator. Guest is the role when not logged in. User is the role of a regular user after successfully logging into the system.

### 2.2 Overall use case diagram

Before log in, a guest can create a new account, log in, request a password change if forgotten, and reconfigure the password after receiving instruction from the sent email. After receiving a successful log in, the system will present a menu with functions corresponding to the user's group. When a guest requests a password change, the system creates a token and sends the instruction for it using email.

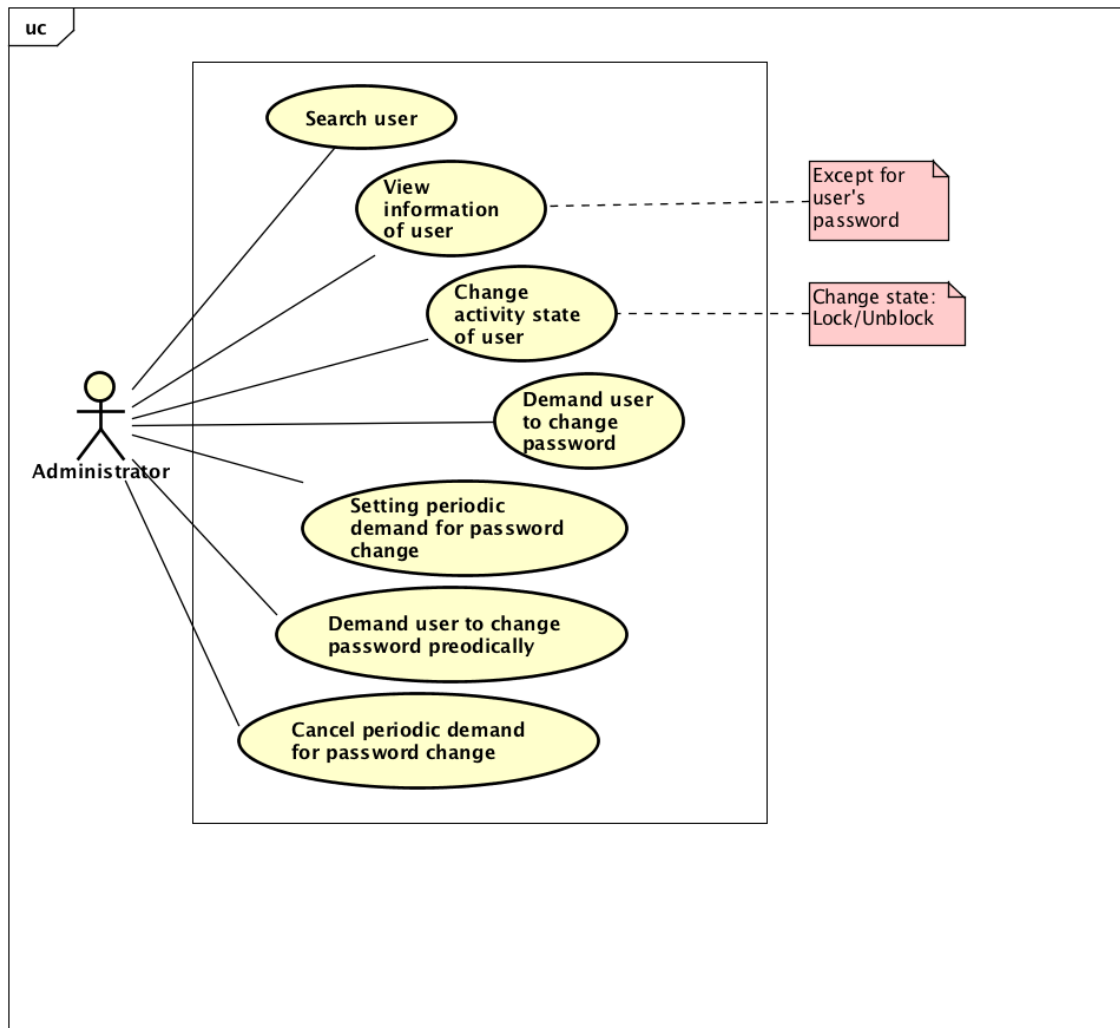
After logging in, a user has the abilities to see and update personal information, and change password.



Administrator can manage users and group of users, function and type of function in the system. Concurrently, he/she can also arrange users into groups, and granting rights to use functions for user groups. The use cases of Administrator in the overall diagram are a compound use case composed of groups of use cases. The details of these compound use cases are given in the Use case decomposition diagram in the next section.

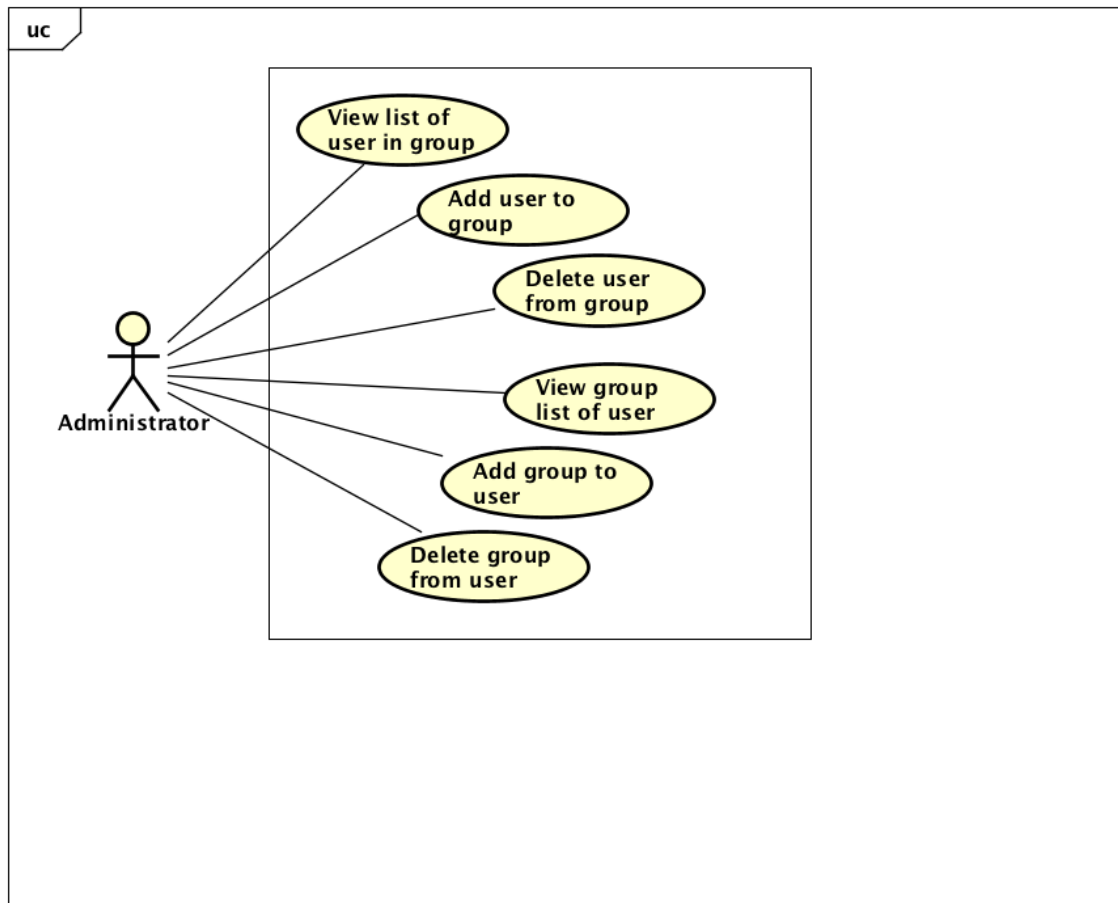
## 2.3 Decomposition of Use cases

### 2.3.1 Use case diagram for “Administer users”

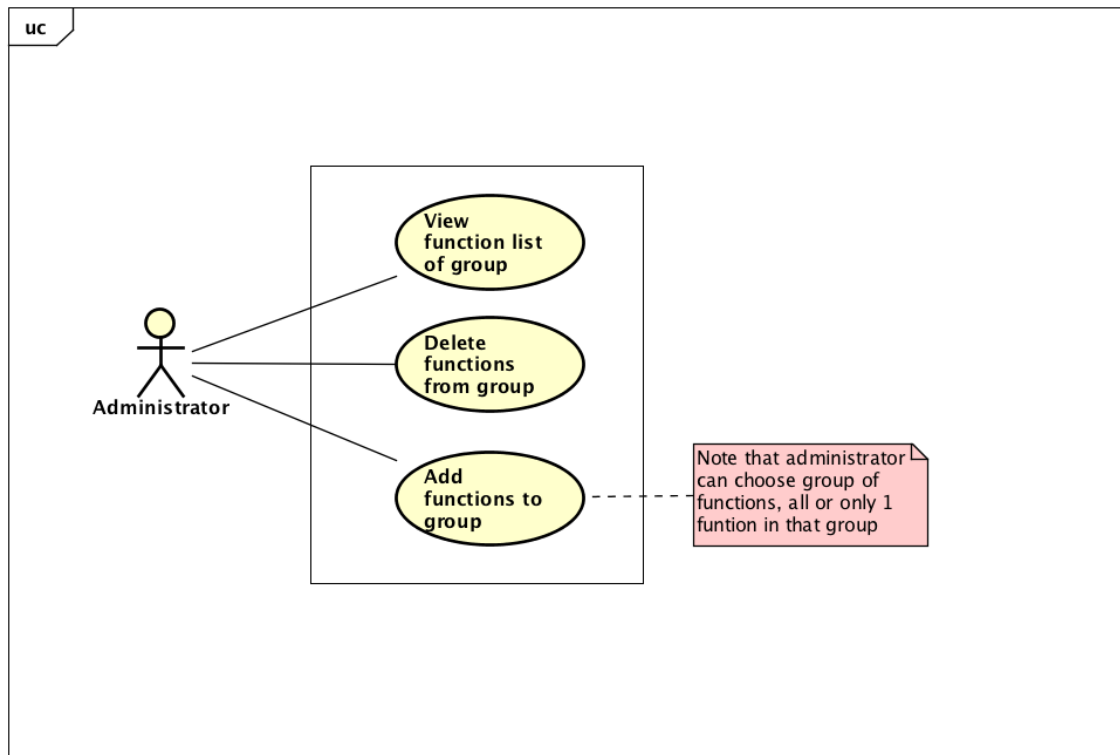




### 2.3.2 Use case diagram for “Manage group of user”



### 2.3.3 Use case diagram for “Grant permission to using functions”



## 2.4 Business processes

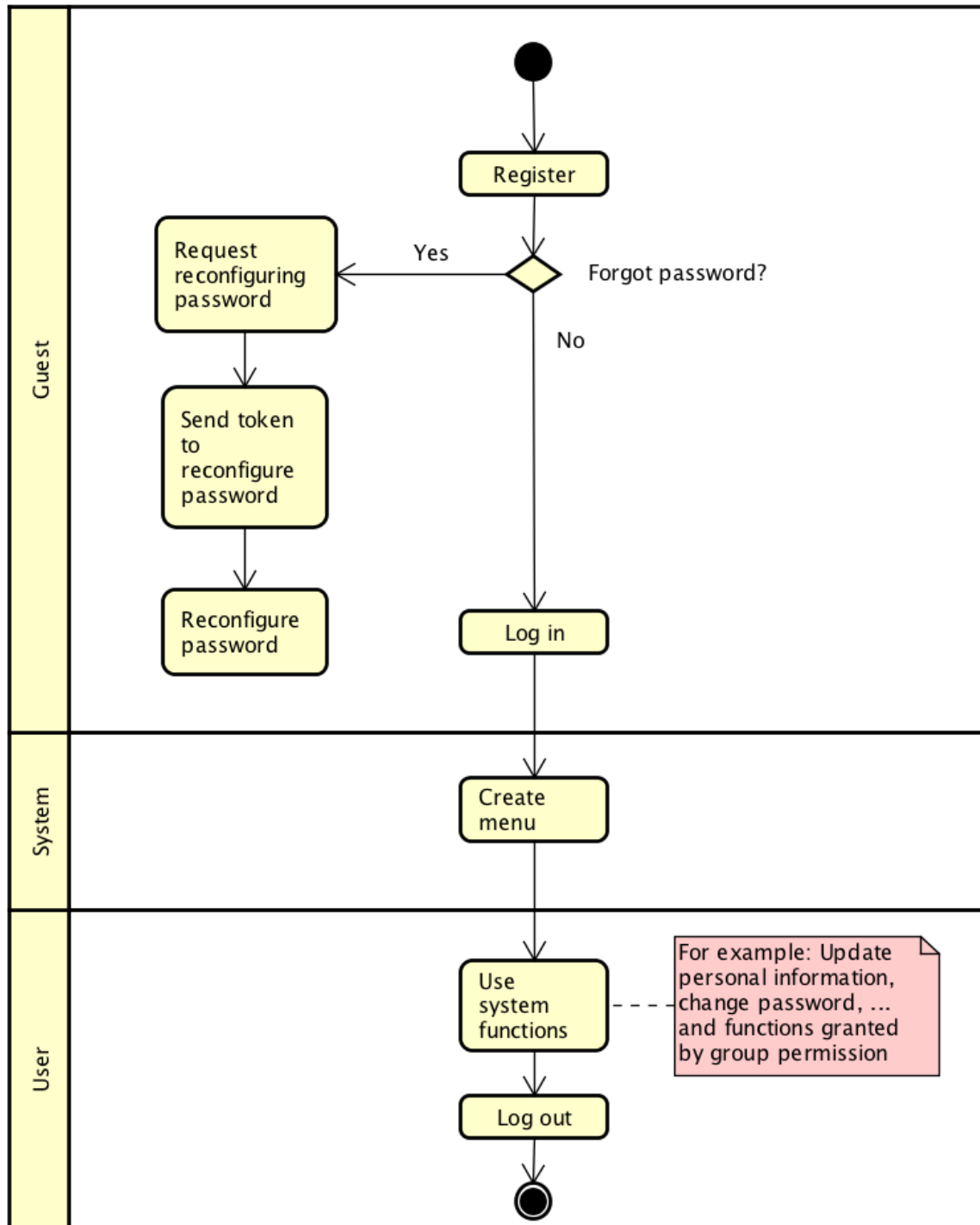
In this subsystem, there are 4 main professional processes: Guest and User’s Software utilization process, User in group management process, Group of user management process and Administrator’s group functions management process.

The details of these processes are modeled in the sub-section of each process.

### 2.4.1 Process of software utilization

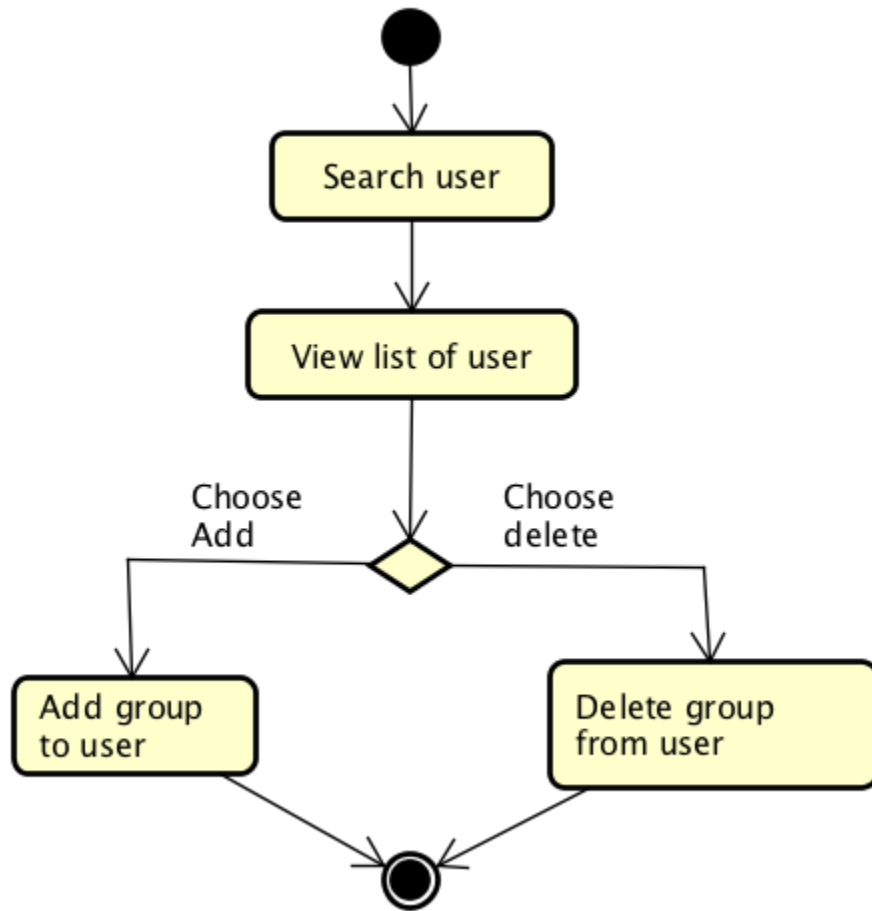
Guest can register to create his/her own account and log in to use the functions of the software afterward. If guest forgot password, he/she is able to request the system for retrieving the password. At this moment, the system will send an email attached with a token to the guest’s registered email address. The guest may access that link to reconfigure the password.

After successfully logging into the system, the guest is able to use functions such as view, update his/her information, change the password, and other functions that were granted for the user group that he/she is in.



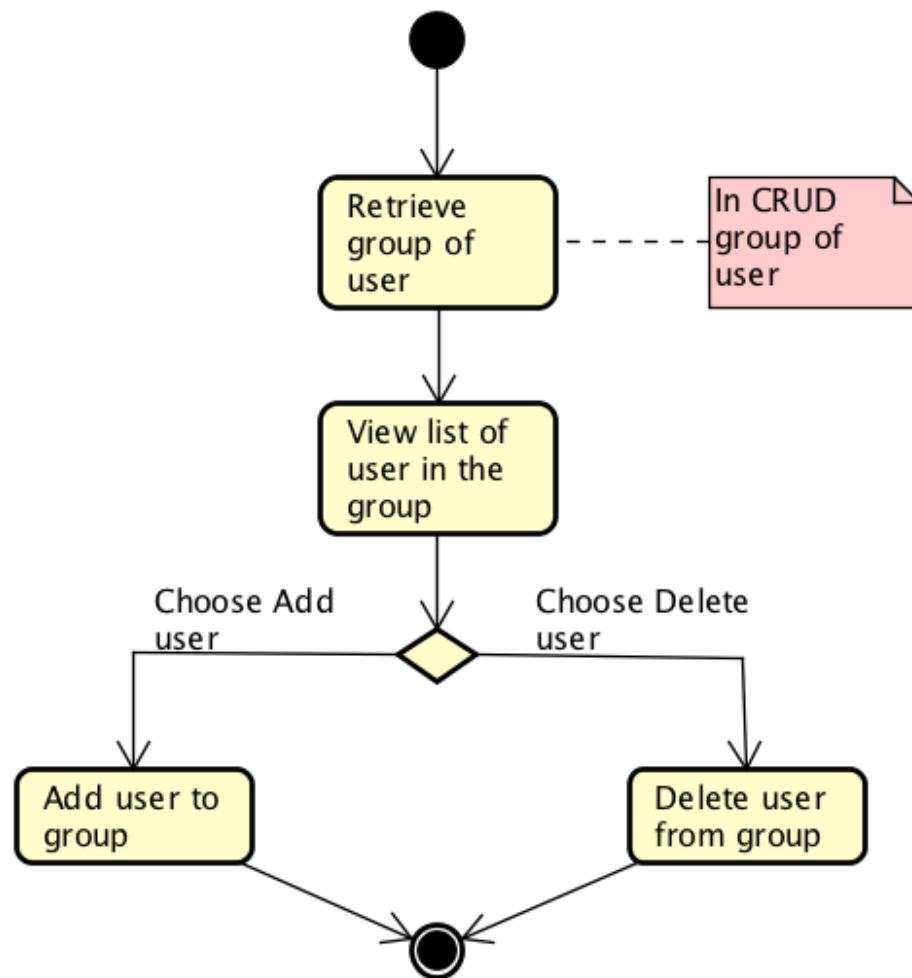
#### 2.4.2 Process of group management

Administrator has the ability to manage group of users base on the following process: The administrators find the user then select that user and view that user's group. Afterward, the administrators may add or delete the user's groups.



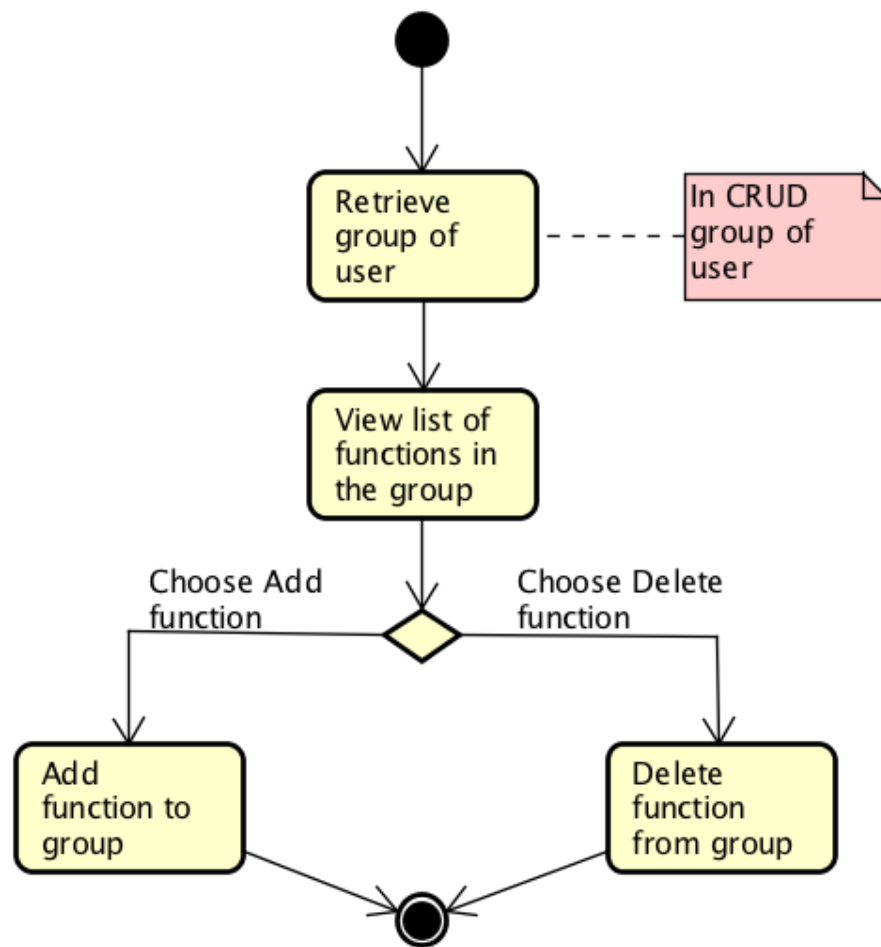
### 2.4.3 Process of user management

Administrator can manage user in a group according to the following process: Administrators select the user group to work with then choose to view the user list of that group. Afterward, the administrator may add user or remove an user of that group.



#### 2.4.4 Process of function management

Administrator manages the group functions in the following process: Administrator select the desired user group then choose to view the function list of such group. Afterward, administrator may add or delete functions of that group.



### 3 Specific requirements

Details of the use cases given in following sections are specified below.

#### 3.1 Specification of Use case UC001 - “Log in”

<b>Use case code</b>	UC001	<b>Use case name</b>	Log in
<b>Actor</b>	Guest		
<b>Pre-condition</b>	None		
<b>Main flow of events (Success)</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	1.	Guest	Select Log in function
	2.	System	Display log in interface
	3.	Guest	Input email and passwords (detailed descriptions are given below *)
	4.	Guest	request log in
	5.	System	check whether guest had input the required field
	6.	System	check whether the inputted email and password exist in the system
	7.	System	call use case “Create menu” including the inputted email
<b>Alternative flow of events</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	6a.	System	output error to guest: need to input into the missing required field if hadn’t done so
	7a.	System	output error: Invalid email and/or password if neither were found
	7b.	System	output error: the account is locked, if the email and password is valid for the locked account by admin
	7c1.	System	call use case “Change password as requested” if email and password are valid; but if only the user is tagged with needing to change password
	7c2.	System	call use case “Create menu” including the email of the logging in guest
<b>Post-condition</b>	None		

\* Input data of personal information include these data fields:

No	Data fields	Description	Mandatory	Valid condition	Example
----	-------------	-------------	-----------	-----------------	---------

1.	Email		Yes		h.anh@gmail.com
2.	Password		Yes		ToiLa12#\$

### 3.2 Specification of Use case UC002 - “Create menu”

<b>Use case code</b>	UC002	<b>Use case Name</b>	Create menu
<b>Actors</b>	System		
<b>Pre-condition</b>	Guest successfully logged in		
<b>Main flow of events (Success)</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	1.	System	call use case “create menu” including the input email of logged in guest
	2.	System	retrieve the list of user groups that the logged in guest belong to
	3.	System	Retrieve the list of fuctions according to each group access rights
	4.	System	display menu containings retrieved function (submenu) sorted in group (menu)
<b>Alternative flow of eventss</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	3a.	System	only show 3 functions: Display, Update personal information and Change password if users doesn't belong to any groups
	4a.	System	only show 3 functions: Display, Update personal info and Change password if all group that the user belong to don't have access to any functions
<b>Post-condition</b>	None		

### 3.3 Specification of Use case UC003 - “Registration”

<b>Use case Code</b>	UC003	<b>Use case Name</b>	Registration
<b>Actor</b>	Guest		
<b>Pre-condition</b>	None		
<b>Main flow of events (Success)</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	1.	Guest	choose Registration function
	2.	System	display registration interface
	3.	Guest	input personal information (detailed description are given below *)
	4.	Guest	request registration
	5.	System	check whether the guest had input all the required fields



	<b>No</b>	<b>Actor</b>	<b>Action</b>
	6.	System	check guest's inputted email validity
	7.	System	check whether the inputted password and confirmed password are identical
	8.	System	check for adequacy of password strength
	9.	System	save the account information and inform Successful registration
<b>Alternative flow of events</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	6a.	System	output error to guest: need to input into the missing required field if hadn't done so
	7a.	System	output error: invalid email address if email address is invalid
	8a.	System	output error: confirmed password are not identical to password if they are not the same
	9a.	System	output error: password need to have adequate strength if it did not meet the requirement of the system (at least 8 characters, including lowercase and uppercase letters, at least 1 number and 1 special characters).
<b>Post-conditions</b>	None		

\*Input data of personal information includes these data fields:

No	Data Fields	Description	Mandatory	Valid condition	Example
1.	First Name		Yes		Hoàng Anh
2.	Last Name		Yes		Nguyễn
3.	Email		Yes	Valid email	h.anh@gmail.com
4.	Password		Yes	At least 8 characters, including lowercase and uppercase letters, at least 1 number and 1 special characters	ToiLa12#\$
5.	Confirmed password		Yes	Identical to password	ToiLa12#\$
6.	Date of birth	Only choose from Date picker	Yes	Valid date	12/12/1991
7.	Sex	Male or female	Yes		Female
8.	Phone Number		No	Valid phone number includes number, space, period or hyphen	0913.123.321

### 3.4 Specification of Use case UC004 - “Search for user”

<b>Use case Code</b>	UC004	<b>Use case Name</b>	Search for user
<b>Actor</b>	Administrator		
<b>Pre-condition</b>	Guest successfully logged in as administrator		
<b>Main flow of event (Success)</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	1.	Administrator	select function Search for user
	2.	System	retrieve list of users role and display interface of “Search for user”
	3.	Administrator	input first name, last name, email or role of the desired user (detailed description are given below *)
	4.	Administrator	request search
	5.	System	check if guest has input at least one search information
	6.	System	search and retrieve the information of users that satisfy the search criteria
	7.	System	display the list of users satisfying the search criteria (detailed description below **) if at least one user is found
<b>Alternative flow of events</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	6a.	System	output: need to input at least one search criteria if users did not do so
	7a.	System	output: Could not find any user satisfying the search criteria if the search return an empty list
<b>Post-condition</b>	None		

\* Input data when search:

No	Data Fields	Description	Mandatory	Valid condition	Example
1.	First Name		No		only need to input first name
2.	Last Name		No		only need to input last name
3.	Email		No		only need to input email
4.	Group	Chosen from a list of user groups	No		not selected or only select 1 group from list

\*\* Output data when displaying search result:

No	Data Fields	Description	Display format	Example
1.	First Name			
2.	Last Name			
3.	Email			
4.	Group	List of groups that the user belongs to	Listed using number	1. Administrator 2. Manager
5.	Date of birth		dd/mm/yyyy	02/12/1987
6.	Phone number		Separate 2 numbers using period	09.12.13.23.12

### 3.5 Specification of Use case UC005 - “CRUD function”

<Note: To avoid spending too much time on CRUD function, only 1 CRUD should be specified, other CRUD can be written as similar to Use case UC005 for example and only need to have input/output data specification>

Use case code	UC005	Use case Name	CRUD function
Actor	Administrator		
Pre-condition	Guest successfully logged in as administrator		
Read:			
Main flow of events	No	Actor	Action
	2.	Administrator	request “Seeing list of function”
	4.	System	retrieve list of functions groups
	6.	System	retrieve list of functions
	8.	System	Display interface for “Seeing list of function” including: <ul style="list-style-type: none"><li>- Upper: List the function groups so that the user can filter the function by group</li><li>- Center: List of functions (Detailed description below *)</li></ul>
Alternative flow of events	5a1	Administrator	choose one of the function group
	5a2	System	retrieve the functions in the selected function group, display in the interface
Update:			
Main flow of events	No	Actor	Action
	1.	Administrator	choose one function and request to update it
	3.	System	retrieve detailed information of the function and display the old information of the function in the “update

			function” interface
	5.	User	update the function information (detailed description below **) and request to update it
	7.	System	checking the required data fields
	8.	System	checking for the data fields that need to be unique
	9.	System	updating the changed information and inform “update success”
<b>Alternative flow of event</b>	5a	System	output error: “all the required fields have not been filled yet” if administrator did so
	6a	System	output error: specify fields that are not unique if administrator repeat them

#### **Delete:**

<b>Main flow of events</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	1.	Administrator	choose one function and request deleting
	3.	System	display the notification that request the user to confirm deleting
	5.	User	confirm deletion
	7.	System	delete the function and inform “Deletion success”
<b>Alternative flow of events</b>	4a	System	end the use case if the user denied the deletion

#### **Create:**

<b>Main flow of events</b>	<b>No</b>	<b>Actor</b>	<b>Action</b>
	1.	Administrator	request adding new function
	3.	System	retrieve the list of function groups and display the interface for adding function
	5.	User	input the information for the function (detailed description below **) and request modification
	7.	System	check for the required data fields
	8.	System	check for the data fields that need to be unique
	9.	System	update the changed information and inform “Update success”
<b>Alternative flow of event</b>	5a	System	output error: “all the required fields have not been filled yet” if administrator did so
	6a	System	output error: specify fields that are not unique if administrator repeat them

<b>Post-condition</b>	None
-----------------------	------

\* Output data for displaying list of functions:

No	Data fields	Description	Display format	Example
1.	No	Ordinal number		1
2.	Function name			Find user
3.	Function group	Name of the function group that the function belongs to		User
4.	Boundary Class	Name of the boundary class that interacts with that function's actor		SearchUserForm

\*\*Input data when adding/updating features:

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Function name		Yes	Unique	Change the activity state of user
2.	Function group	Choose one of the function groups in the list	Yes		User
3.	Boundary class		Yes	Unique	SearchUserForm

## 4 Supplementary specification

*<Presenting other requirements if necessary, including non-functional requirements such as performance, reliability, usability, and supportability; or other technical requirements such as database system, used technology...>*

### 4.1 Functionality

- Among the flow of events of use case, in all the steps that interacts with the database system, if there are errors in the connection or operation processes, there need to be a corresponding error notifications so that the actor knows that the error is related to the database system rather than the user.
- Use cases by Administrator and User must require the Guest to log in with the corresponding role.
- The display format is of following:
  - o Right justification number
  - o Left justification text
  - o Font: Arial 14, black
  - o White background

### 4.2 Usability

Functions need to be designed with ease of use in mind. There need to be detailed instructions about the errors of the users for locating, specifying and fixing the errors.

### 4.3 Other requirements

*<Descriptions of other requirements are located here; this section may add other qualification properties like Efficiency/Performance, Reliability, Maintainability, and Portability>*