Assignment 3 Sample Solution

There is no model answer in software engineering. The following document serves only as a sample solution to help students to understand the main concepts and practice in object-oriented design. It should not be interpreted as the only possible specification or the best specification.

Part 1: Before the Requirement Change

Step 1: Transforming Use Case Documentation

Author: Jolly Cheng Date: 1/11/2016

Use Case Name:	Register	
Actor(s):	Member	
Description:	This use case describes the membership registration process	
Reference:	TSE-D1	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Initiate the use case when a customer wants to register as a member. Click the <Register> button in main window.	**Step 2**: Display the registration window and ask for the personal information including name, address, email address, mobile phone number, date of birth (optional field) and gender (optional field).
	Step 3: Input the personal information and clicks <Submit>.	**Step 4**: Check the input data. **Step 5**: Display a dialog window to show the status of the storing process. Assign a unique membership number and a password and store the personal information. **Step 6:** When the storing process is finished, display a successful message with the membership number and password to the customer in the dialog window. Conclude the use case when the membership number and password together with a membership card is given to the customer.
Alternative Course(s):	**Alt-Step 3:** The customer can click <Cancel> button during the input of personal information and conclude the use case. **Alt-Step 4A:** If an input item is missing, open a pop-up window that displays an error message "[item] is missing", where [item] is the name of the missing item such as "phone number". If an input item is incorrect, open a pop-up window that displays an error message "[item] is incorrect", where [item] is the name of the missing item such as "email address". If more than one input item is missing or incorrect, consolidate all the error messages into one pop-up window only. **Alt-Step 5:** The customer can press the <Quit> button during the storing process and display the cancelled message. Conclude the use case.	
Precondition:	None.	
Postcondition:	None.	
Assumptions:	The customer is not registered before.	

Step 2: Modeling Object Interactions and Behaviors

a. Identify and Categorize Design Class

Boundary Class	Control Class	Entity Class
Main window	Registration Processor	Member
Registration window		
Pop-up window		
Dialog window		

b. Identify (additional) Attributes

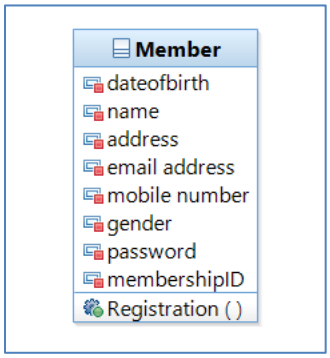
Class	Attribute
Main window	<Register> button
Registration window	Name, address, email address, mobile phone number, date of birth, gender, <Submit> button, <Cancel> button
Pop-up window	Error messages
Dialog window	<Quit> button, successful message, storing status

c. Identify Behaviors and Responsibilities

Class behaviors	Automated/Manual	Type of Class
Wants to register	Manual	
Click <Register> button	Manual	
Display the registration window and ask for the personal information including name, email address, mobile phone number, date of birth (optional field) and gender (optional field).	Automated	Boundary
Input the personal information and clicks <Submit>	Manual	
Check the input data	Automated	Entity

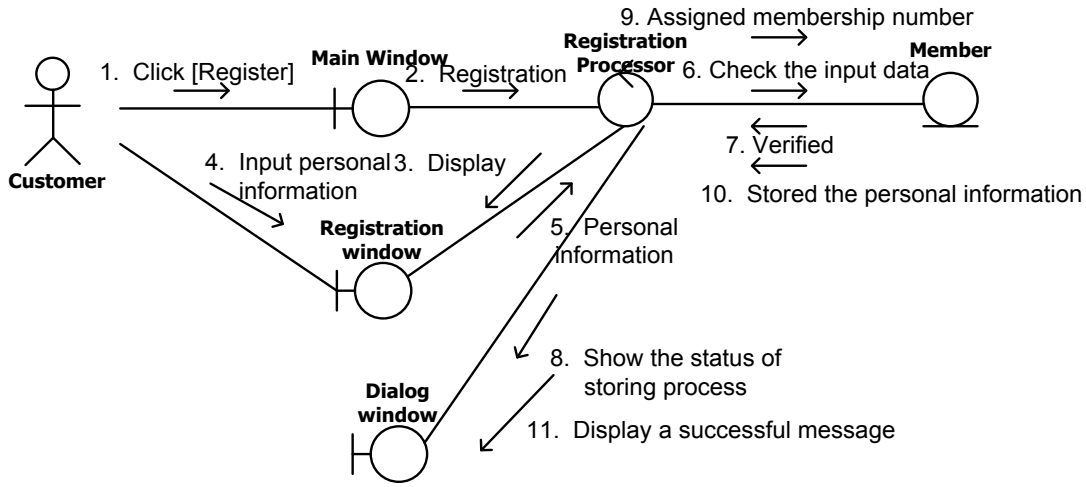
Display a dialog window to show the status of the storing process.	Automated	Boundary
Assign a unique membership number	Automated	Control
Store the personal information.	Automated	Entity
Display a successful message with membership number and password	Automated	Boundary
The membership number and password together with a membership card is given to the customer	Manual	
Click <Cancel> button	Manual	
Open a pop-up window that displays an error message “[item] is missing”	Automated	Boundary
Open a pop-up window that displays an error message “[item] is incorrect”	Automated	Boundary
Consolidate all the error messages into one pop-up window	Automated	Boundary
Press the <Quit> button	Manual	
Display the cancelled message	Automated	Boundary

Step 3: Update the Class Diagram



Step 4: Sketch a High-level Object Interaction

Register



After changing requirement

Step 1: Transforming Use Case Documentation

Author: Jolly Cheng Date: 1/11/2016

Use Case Name:	Register	
Actor(s):	Member	
Description:	This use case describes the membership registration process	
Reference:	TSE-D2	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Initiate the use case when a customer wants to register as a member. Click the <Register> button in main window.	
		Step 2: Display the registration window and ask for the personal information including name, address, email address, mobile phone number, date of birth (optional field) and gender (optional field).
	Step 3: Input the personal information and clicks <Submit>.	
		Step 4: Check the input data.
		Step 5: Display a dialog window to show the status of the storing process. Assign a unique membership number and a password and store the personal information.
		Step 6: When the storing process is finished, display a successful message with the membership number and password to the customer in the dialog window. Conclude the use case when the electronic membership card is sent to the customer by email.
Alternative Course(s):	**Alt-Step 3:** The customer can click <Cancel> button during the input of personal information and conclude the use case. **Alt-Step 4A:** If an input item is missing, open a pop-up window that displays an error message "[item] is missing", where [item] is the name of the missing item such as "phone number". If an input item is incorrect, open a pop-up window that displays an error message "[item] is incorrect", where [item] is the name of the missing item such as "email address". If more than one input item is missing or incorrect, consolidate all the error messages into one pop-up window only. **Alt-Step 5:** The customer can press the <Quit> button during the storing process and display the cancelled message. Conclude the use case.	
Precondition:	None.	
Postcondition:	None.	
Assumptions:	The customer is not registered before.	

Step 2: Modeling Object Interactions and Behaviors

a. Identify and Categorize Design Class
b. Identify and Categorize Design Class

Boundary Class	Control Class	Entity Class
Main window	Registration Processor	Member
Registration window		
Pop-up window		
Dialog window		
Email sender		

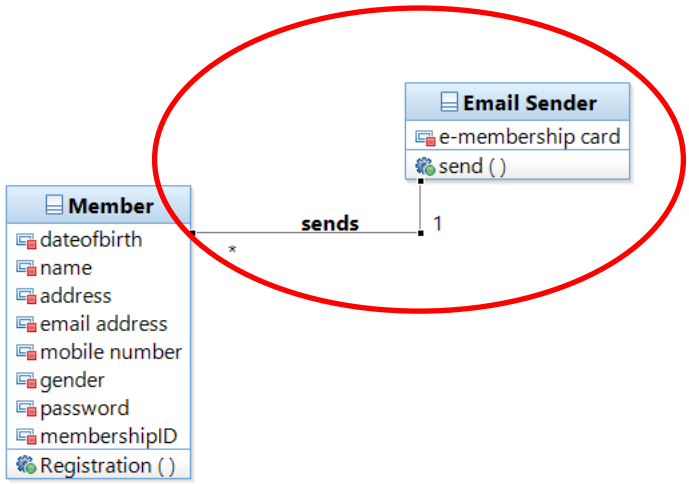
c. Identify (additional) Attributes

Class	Attribute
Email sender	Electronic membership card

d. Identify Behaviors and Responsibilities

Class behaviors	Automated/Manual	Type of Class
send the electronic membership card to the customer by email	Automated	Boundary

Step 3: Update the Class Diagram



Step 4: Sketch a High-level Object Interaction