

# Automatic Generating UML Use Case Diagram and Test Cases Based on Classification Tree Method

Wassana Naiyapo  
Computer Science Department Faculty of Science  
Chiang Mai University  
Chiangmai Thailand  
wassana.n@cmu.ac.th, wassanacs@gmail.com

Atichat Sangtong  
Computer Science Department Faculty of Science  
Chiang Mai University  
Chiangmai Thailand  
Atichat.sangtong@gmail.com

**Abstract**—The processes in software development by Object Oriented methodology have many stages, and those take time and high cost. The inconceivable error in system analysis process will affect the design and the implementation process. The unexpected output is the reason why we need to revise the previous process. The more rollback of each process takes more expense and delayed time. The implemented software, which is efficient and reliable and meets the user's requirement, comes from the good test process. Unified Modelling Language (UML) is the tool which uses symbols to describe the work process in Object Oriented Analysis (OOA). This paper presents the approach for the automatically generated UML use case diagram and test cases. The UML use case diagram is generated from the event table, and the test cases are generated from use case specifications and Graphic User Interfaces (GUI). Test cases are derived from the Classification Tree Method (CTM) that classifies data to a node present in the hierarchy structure. Moreover, this paper refers to the program that generates the use case diagram and test cases. As a result, it can reduce work time and increase efficiency work.

**Keywords**—Classification Tree Method, Test Case, UML Use Case Diagram, Use Case Specification

## I. INTRODUCTION

Software development has many phases. It starts with acquiring user requirement, analysis phase, design phase, implement phase, integrate phase, testing phase and maintenance phase. All of these processes take time and cost, especially testing phases are very important [1]. Testing phase spends 30%-40% of the system's cost [2]. The better testing processes the more efficient system. The objective of software testing is the process intent of finding errors. The Object-Oriented methodology broadens the testing view that includes the review of both analysis and design model [3].

Object Oriented Analysis (OOA) has the object considered in the real-life system. UML is used to describe in OOA, with international standard symbol and format, then it makes more understandable system.

UML is divided into Structure diagram and Behavior diagram [4]. UML use case diagram is classified in Behavior diagram and it is used in requirement phase to elicit user requirement. User perspective can explain what-who occurred in the system. Use case diagram is described user's work, user's service, and communicated with stakeholder [5].

The use case is activities or events in the system. All use cases consist of the use case diagram. Each use case description is called the use case specification. The use case specification explains workflow event detail. There is no standard definition for the use case specification but can be present in the template [6].

Test case derives from the use case specification. Basic flow and alternative flow in use case specification have been considered to generate test cases. The more test cases make the more system efficiency.

The test case can present in the format like a tree structure. Input data are classified to a node in the tree. For this concept, the test case or result can be simply recognized in Classification Tree Method (CTM) [7][8].

An additional test case can be generated from Graphic User Interface (GUI). The use case is the activities' workflow of the system, then most of the events are related to GUI.

This paper presents an automatic generate use case diagram and test cases. The use case diagram derives from the event table. Test cases generate from use case specification and GUI, these are represented in the test case table and tree format based on CTM.

## II. RELATED WORKS

There are several discussions about UML and test case as follows. One of the approaches is to derive the use case diagrams from an event table as in [9] by Muhairat and Al-qutaish. This research introduces an approach to derive use case diagram from an event table. This paper explains the event table, extended event table, and step for create use case diagram. Car rental web page is a case study. However, this approach will completely depend on the availability of a comprehensive event table which is built on the availability requirements.

Another one is the UML Diagrams Generator: A New CASE Tool to Construct the Use-Case and Class Diagrams from an Event Table as in [10] by Muhairat et al. This research represents the CASE tool to automate the proposed approach will be introduced, that is, the UML diagrams generator (UMLdg). The CASE tool generates UML use case diagram, class diagram from the event table. UML use case diagram is generated by mapping event and actor from event table. UML class diagram is generated by mapping the event table to attribute, method, and association with class. The result from approach is the CASE tool that generates UML use case diagram and class diagram. This research clarifies step, and the library system is the example.

UML Behavioral Model Based Test Case Generation: A Survey by Shirole and Kumar as in [11]. This research describes survey for test case generation from the behavioral diagrams with UML sequence, state chart and activity diagrams. Test case generation technique approaches are broadly classified into four major categories: specification based testing, graph theoretic testing, heuristic testing, and direct UML specification processing. This research surveys the UML diagram to generate the test case and present on-

going research in the field of UML transition sequence based testing.

A Survey on Test Case Generation Techniques Using UML Diagrams by Rathee and Chillar as in [12]. This paper provides an overview of various test case generation techniques using single UML diagrams, combinational UML diagrams, and prioritization of the generated test cases. The paper presents technique that UML diagram uses technique for generating the test case and an example in each case for explaining the work process. This paper can help researchers to identify what work has been done in their respective fields.

Literature Review of Test Case Generation Techniques for Object Oriented System by Khurana and Chillar as in [13]. This paper presents a review of various test case generation techniques. The techniques are generated using different UML Diagrams and using different algorithms. This paper describes topics that help the researcher to get a variety of perspectives.

Test Case Generation from UML Models by Wang and Zheng as in [14]. This paper presents a methodology to generate the test case from a design level class diagram and an interaction diagram. This paper generates the test cases by using a case study from a class diagram in part association between class. UML collaboration diagram that UML2.5 is a communication diagram in part return car to generate the test cases. The results are that the test cases can satisfy all messages. The results showed that the test case generated from model elements directly are able to satisfy all the required graph coverage criteria discussed in those approaches.

Kansomkeat et al. present research in the topic “Generating Test Cases from UML Activity Diagrams using the Condition CTM” as in [15]. This paper proposes the Condition-CTM for generating test cases from activity diagrams. The test cases, from UML activity diagram, are generated by using Condition CTM. The sale system and result test case are the example work process. Class Payment and class CarPaking represent mutation analysis for experimental data show that tests generated by the Condition CTM and method have strong ability to detect faults.

### III. PROPOSED APPROACH

The software development starts with acquired user requirement and system behaviour studies. The methodology that creates use case diagram and test cases will process with the following steps, overview approach is shown in Fig. 1.

#### A. Event Table Process

User inputs data to the event table. The event table which is added more details is called extended event table. The details of extended event table are Event, General source, Special source, Action, Include Action, Extends Action, Specializes Action and Destination. Each datum in the extended event table is the event that occurs in the system. The user can manage (add, delete, modify) extended event table, as shown in Fig. 2.

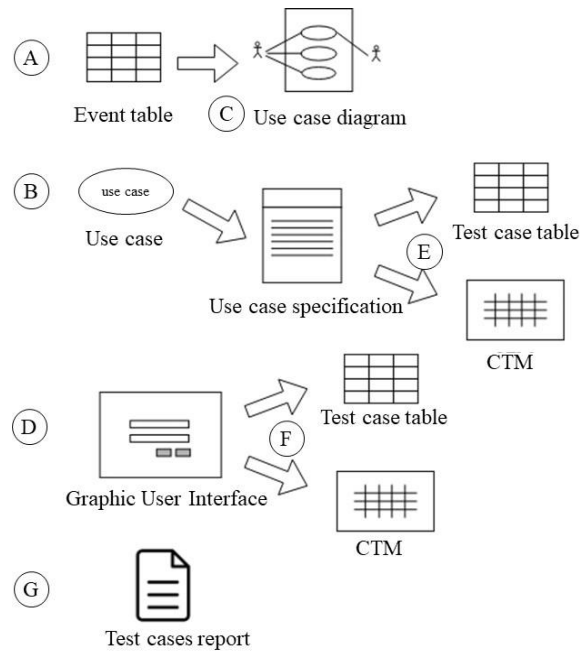


Fig. 1 Overview of the approach

Event	General Source	Special Source	Action	Object	Includes "Action"	Extends "Action"	Specializes "Action"	Destination

Fig. 2 Extended event table

#### B. Use Case Specification Process

The user can manage use case specification details which are Use Case ID, Use Case Name, Actor, Description, Pre Condition, Basic Flow of Event, Alternative Flow and Post Condition. The use case specification is shown in Fig. 3.

Use case ID	UC01
Use Case Name	xxxxx
Actor	xxxxxxxx
Description	xxxxxxxxxx
Pre Condition	xxxxxxxxxx
Basic Flow of Event	1.xxxxxxxxxx
	2.xxxxxxxxxx
	3.xxxxxxxxxx
Alternative Flow	3a. xxxxxxxxxxxx
Pos Condition	xxxxxxxxxx

Fig. 3 Use case specification template

#### C. Use Case Diagram Generating

The program will retrieve data from the extended event table then generates the use case diagram.

#### D. GUI and Detail Process

GUI are screens that are shown the interaction between user and system. In this process, each object tool is defined its detail and priority. The user can manage GUI details (GUI id, GUI name, GUI description, GUI use case, Type of tool, Priority, and Action).

#### E. Generating Test Cases from Use Case Specification

The program generates test cases from the flow of event and alternative flow in use case specification. The results of

this process are represented in the test case table and test case based on CTM

**F. Generating Test Cases from GUIs**

The program generates test cases from GUIs. The results of this process are represented in the test case table and test case based on CTM.

**G. Result**

The output is shown in all the system test case tables which are related to the use cases in use case diagram.

**IV. CASE STUDY AND TEST RESULTS**

The case study of the evaluated program is dormitory management system. The use case diagram and test cases are generated by the following steps:

**A. Event Table Process**

The dormitory management system event data are prepared in the extended event table. The example of an extended event table of dormitory management system is shown in Table I.

The next step, the user inputs the extended event data into the program are shown in Fig. 4.

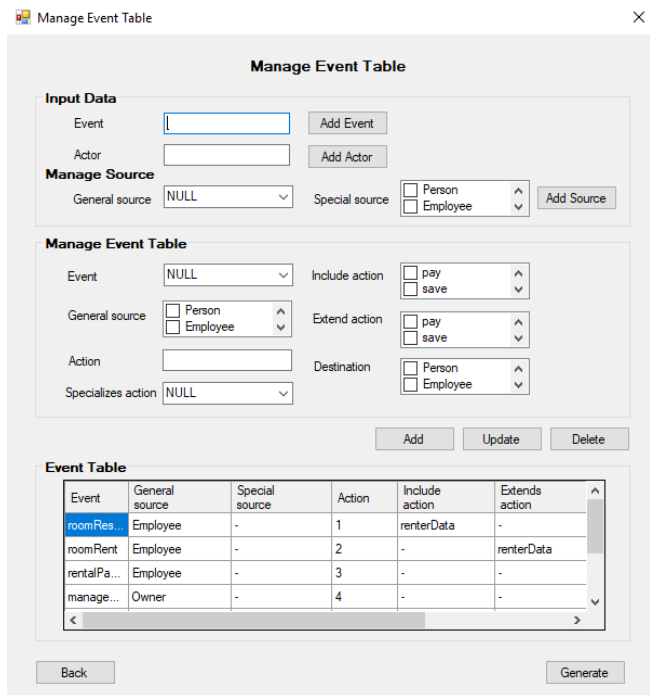


Fig. 4 The extended event data input screen

**B. Use Case Specification Process**

The dormitory management system use case is prepared to use case specification form, as shown in Fig. 5.

Each event from the use case specification will be input to the program. The use case specification data input screen is shown in Fig. 6.

Use case ID	UC01
Use Case Name	login
Actor	Employee
Description	verify username & password
Pre Condition	-
Basic Flow of Event	1.display login user interface (GUI name Rent01UI) 2.enter username & password 3.verify username & password 4.display Frontend user interface (GUI name Rent02UI)
Alternative Flow	3a. username & password not valid 3a.1 display error message

Use case ID	UC10
Use Case Name	roomReserve
Actor	Customer, Employee
Description	Employee reserve room for customer
Pre Condition	login valid
Basic Flow of Event	1.display reserve room user interface (GUI name Rent05UI) 2.enter room number 3.enter customer information 4.enter date/month/year for room rental
Alternative Flow	2a. room number not found 2a.1 display error message 4a. date not valid 4a.1 display error message

Fig. 5 Use case specification form

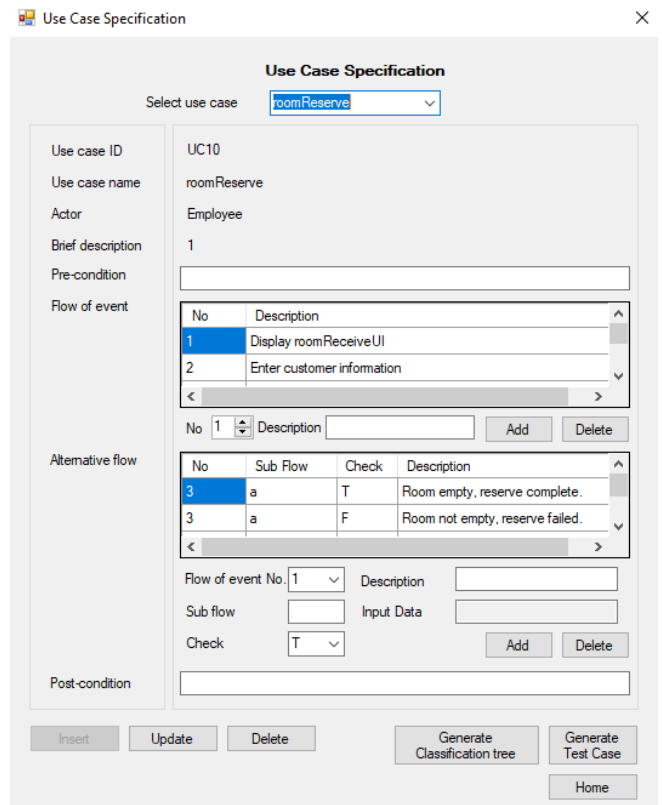


Fig. 6 Use case specification data input screen

### C. Use Case Diagram Generating

From the dormitory management system's extended event table, the program could generate a use case diagram, as shown in Fig. 7, and the example of the dormitory management system use case diagram snapshot is shown in Fig. 8.

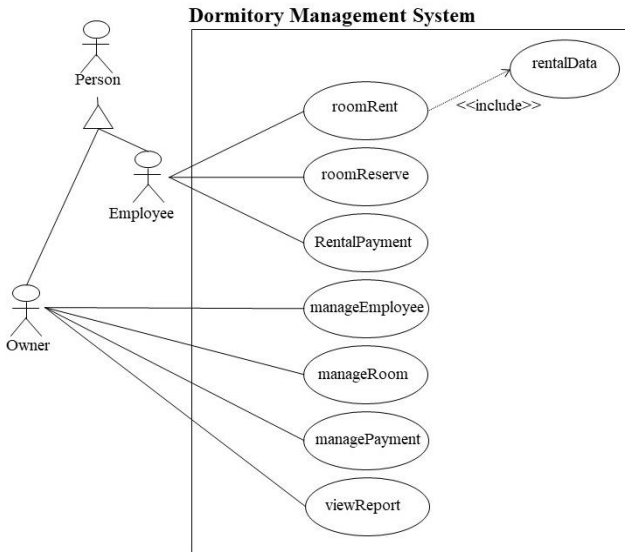


Fig. 7 Dormitory management system use case diagram

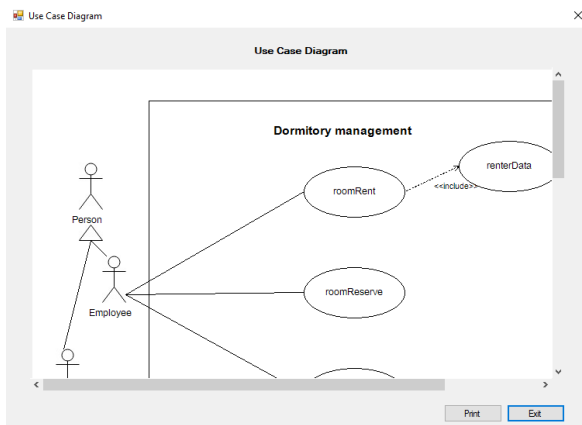


Fig. 8 Dormitory management system use case diagram snapshot

### D. GUI and Detail Process

The dormitory management system use case is related to GUIs. The GUI will upload to the program. The program will be ready to input each tool's priority and GUI details. The example of GUI object tools and details data input is shown in Fig. 9.

### E. Generating Test Cases from Use Case Specification

The program will generate test cases from use case specification. The results of this process are represented in the test case table and test case based on CTM. The test case table snapshot and report are shown in Fig. 10 and Fig. 11. The test case based on the CTM snapshot and report are shown in Fig. 12 and Fig. 13.

### F. Generating Test Cases from GUI

The program will generate test cases from GUI. The results of this process are represented in the test case table and test case based on CTM. The test case table snapshot

and printed report are shown in Fig. 14 and Fig. 15. The test case based on the CTM snapshot and printed report are shown in Fig. 16 and Fig. 17.

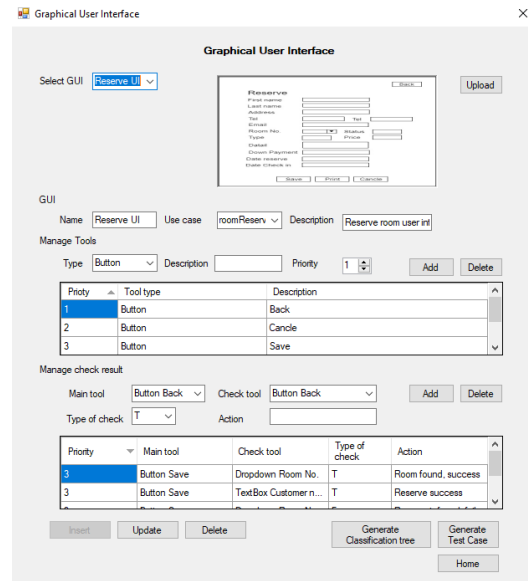


Fig. 9 GUI object tools and details data input

Test Case Table : roomReserve

Test Case No	Input Data	Result check			System action
		Room No	Check in date		
TC01	Input Room No and Check in date	valid	valid		Room empty, reserve complete.
TC02	Input Room No and Check in date	valid	invalid		Room not empty, reserve failed.
TC03	Input Room No and Check in date	invalid	invalid		Room not empty, reserve failed.

Fig. 10 The test case table snapshot

Use case name : roomReserve  
Description : Employee reserve room for customer  
- Test case table : roomReserve use case

Test Case No	User action	Room no	Date Rental	System action
TC01	Input room number and date rental	valid	valid	Reservation success
TC02	Input room number and date rental	valid	invalid	Date invalid
TC03	Input room number and date rental	invalid		Room not available

Fig. 11 The test case table printed report

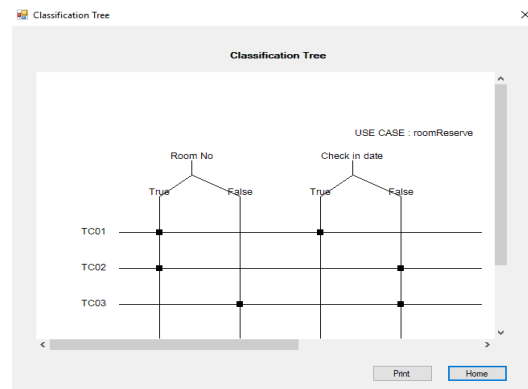


Fig. 12 The test case based on CTM snapshot

TABLE I EXTENDED EVENT TABLE OF DORMITORY MANAGEMENT SYSTEM

Event	General source	Special source	Action	Object	Include action	Extended action	Special action	Destination
room Rent	Employee		Employee process a room rental	Room item	rental data			Employee
room Reserve	Employee		Employee process a room reservation	Reservation item				Employee
rental Payment	Employee		Employee process rental payment	Rental payment item				Employee
manage Payment	Owner		Owner manage payment information					Owner
manage Room	Owner		Owner manage room information					Owner
manage Employee	Owner		Owner manage employee information					Owner
view Report	Owner		Owner view report					Owner

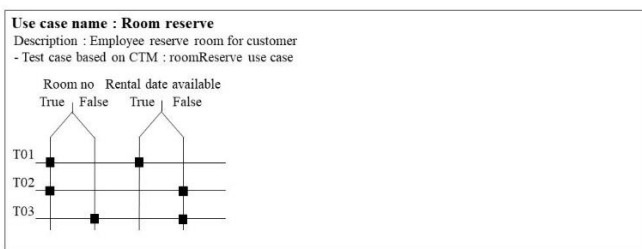


Fig. 13 The test case based on CTM printed report

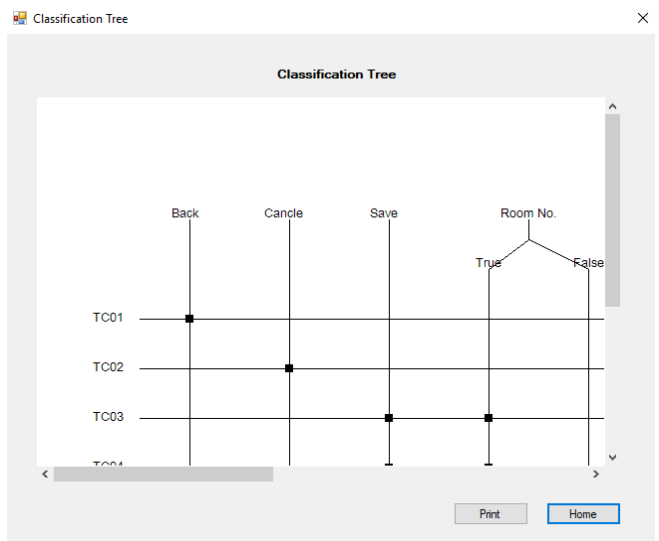


Fig. 16 The test cases based on CTM from GUI snapshot

Test Case Table

**Test Case Table : Reserve UI**

Test Case No	User action	Input data		System action
		Room No.	Customer name	
TC01	Click Back			Back
TC02	Click Cancel			Cancel
TC03	Click Save	valid	valid	Reserve success
TC04	Click Save	valid	invalid	Reserve failed
TC05	Click Save	invalid	invalid	Reserve failed

Fig. 14 The test case table from GUI snapshot

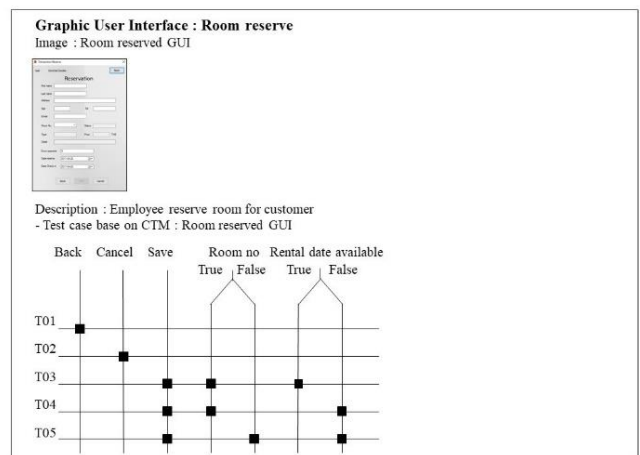


Fig. 17 The test cases based on CTM from GUI printed report

**Use case name : roomReserve**  
 Description : Employee reserve room for customer  
 - Test case table : Room reserve GUI

Test Case No	User action	Room no	Rental date	System action
TC01	Click Back			Back
TC02	Click Cancel			Cancel
TC03	Click Save	valid	valid	Reservation success
TC04	Click Save	invalid	valid	Room not found, message error
TC05	Click Save	invalid	invalid	Room not found, message error

Fig. 15 The test case table from GUI printed report

**G. Result: List for All Test Cases**

All test cases from the dormitory management system are shown in Fig. 18.

## V. CONCLUSION

This paper presents a new approach that generates use case diagram and test cases. The UML use case diagram is created from the event table. Test cases are created from use case specification and GUI, those have results shown in test case tables and test case based on CTM. This approach can reduce work time in the software development process. The program can help to create test cases in the testing process.

The program is applied in the dormitory management system case study, the program can create use case diagram and test cases. In addition, the program has 28 students test in various application. They satisfy with the result. The results from program are the UML use case diagram, and test cases can help them understand in the system and workflow in OOD.

In the future work, the developing of UML diagram generate tools can create UML diagram such as class diagram, sequence diagram, state machine diagram. The tools can apply to all processes in the system that supports UML diagram and test cases.

## REFERENCES

- [1] Ian Sommerville, "Software Process," in Software Engineering, 9th ed. Pearson Education, 2011, pp. 27-55.
- [2] R. S. Pressman, "Software Testing Technique," in Software Engineering: A Practitioner's Approach, 5th ed. New York: McGraw-Hill, 2001, pp. 437-476.
- [3] R. S. Pressman, "Object Oriented Testing," in Software Engineering: A Practitioner's Approach, 5th ed. New York: McGraw-Hill, 2001, pp. 631-651.
- [4] UML 2.2 Diagrams Overview. Available at: <https://www.uml-diagrams.org/uml-25-diagrams.html>. Accessed May 20, 2018.
- [5] Simon Bennet, John Skelton, and Ken Lunn, "Use Case" in Schaum's Outline UML, 2nd ed. McGraw-Hill, 2005, pp. 20-41.
- [6] Ambler S. Technical library, "Web services programming tips and tricks: Documenting a use case", Available at: <https://www.ibm.com/developerworks/library>. Accessed May 20, 2018.
- [7] M. Grochtmann and K. Grimm, "Classification Trees for Partition Testing," Software Testing, Verification & Reliability, vol. 3, Jun. 1993, pp. 63-82,
- [8] M. Grochtmann, J. Wegener, and K. Grimm, "Test case design using classification trees and The Classification Tree Editor CTE".8th international Software Quality Week,1995, pp. 1-11.
- [9] Mohammad I. Muhairat and Rafa E. Al-Qutaish. "An Approach to Derive the Use Case Diagrams from an Event Table", in Proceeding of the 8th WSEAS int. Conference on Software Engineering, Parallel and Distributed Systems, ISBN:978-960-474-052-9,pp.33-38.
- [10] Mohammad I. Muhairat, Rafa E. Al-Qutaish and Akram A. Abdelqader, "UML Diagrams Generator: A New CASE Tool to Construct the Use-Case and Class Diagrams from an Event Table", Journal of Computer Science,2010,pp.253-260.
- [11] Mahesh Shirole and Rajeev Kumar, "UML Behavioral Model Based Test Case Generation: A Survey.", Journal of ACM SIGSOFT Software Engineering Notes,Vol.38,No.4,July 2013.
- [12] Nisha Rathee and Rajender Singh Chhillar, "A Survey on Test Case Generation Techniques Using UML Diagrams", Journal of Software, Vol.12, No.8,August 2017.
- [13] Namita Khurana and R.S Chillar, "Literature Review of Test Case Generation Techniques for Object Oriented System", International Journal of Computer Applications,Vol.105,No.15,November 2014.
- [14] Yiwen Wang and Mao Zheng, "Test Case Generation from UML Models." 45th Annual Midwest Instruction and Computing Symposium, Cedar Falls, Iowa. 2012.
- [15] Supaporn Kansomkeat, Phachayanee Thiket and Jeff Offutt, Generating Test Cases from UML Activity Diagrams using the Condition Classification Tree Method, 2nd International Conference on Software Technology and Engineering (ICSTE),2010.

Test Cases Report					
<b>Use case name : roomReserve</b>					
Description : Employee reserve room for customer					
- Test case table : roomReserve use case					
Test case no	Input data	Room no	Rental date	System action	
TC01	Input room number and date rental	valid	valid	Reservation success	
TC02	Input room number and date rental	valid	invalid	Date invalid	
TC03	Input room number and date rental	invalid		Room not available	
- Test case table : Room reserve GUI					
Test case no	User action	Room no	Rental date	System action	
TC01	Click Back			Back to menu	
TC02	Click Cancel			Cancel	
TC03	Click Save	valid	valid	Reservation success	
TC04	Click Save	invalid	valid	Room not found, message error	
TC05	Click Save	invalid	invalid	Room not found, message error	

Test Cases Report					
<b>Use case name : roomRent</b>					
Description : Employee rent room for customer					
- Test case table : roomRent use case					
Test case no	Input data	Room no	Customer name	Customer Telephone	System action
TC01	Input room number and customer inf	valid	valid	valid	Rental success
TC02	Input room number and customer inf	valid	valid	invalid	Error message
TC03	Input room number and customer inf	valid	invalid	invalid	Error message
TC04	Input room number and customer inf	invalid	invalid	invalid	Error message
- Test case table : Room rent GUI					
Test case no	User action	Room no	Customer name	Customer Telephone	System action
TC01	Click Back				Back to menu
TC02	Click Cancel				Cancel
TC03	Click Delete				Delete rental inf
TC04	Click Save	valid	valid	valid	Rental success
TC05	Click Save	valid	valid	invalid	Error message
TC06	Click Save	valid	invalid	invalid	Error message
TC07	Click Save	invalid	invalid	invalid	Error message

Test Cases Report			
<b>Use case name : rentalPayment</b>			
Description : Customer room payment			
- Test case table : rentalPayment use case			
Test case no	Input data	Bill no	System action
TC01	Input rental information	valid	Payment success
TC02	Input rental information	invalid	Error message
- Test case table : Rental payment GUI			
Test case no	User action	Bill no	System action
TC01	Click Back		Back to menu
TC02	Click Cancel		Cancel
TC03	Click Save	valid	Payment success
TC04	Click Save	invalid	Error message

Test Cases Report				
<b>Use case name : manageRoom</b>				
Description : Owner manages room information				
- Test case table : manageRoom use case				
Test case no	Input data	Room no	Room type	System action
TC01	Input room number and room type	valid	valid	Search found, display room information
TC02	Input room number and room type	valid	invalid	Search not found, display error message
TC03	Input room number and room type	invalid	invalid	Search not found, display error message
- Test case table : Room rent GUI				
Test case no	User action	Room no	Room type	System action
TC01	Click Back			Back to menu
TC02	Click Cancel			Cancel
TC03	Click Delete			Delete room inf.
TC04	Click Save	valid	valid	Manage room success
TC05	Click Save	valid	invalid	Error message
TC06	Click Save	invalid	invalid	Error message

Fig. 18 All test cases from the dormitory management system

The program has 28 students in Object Oriented Designed (OOD) class test and there are 17 students like the program because it can help to understand the work on OOD. Nevertheless, it still takes more time to understand how the program works. The students said that the program should be more friendly and easier to use.