

SS Apartment System

By  
Mr. Surakiat Somboonphokkaphan

Submitted in Partial Fulfillment of  
the Requirements for the Degree of  
Master of Science  
in Information Technology  
Assumption University

May, 2002

# SS Apartment System

By

**Mr.SURAKIAT SOMBOONPHOKKAPHAN**



**Submitted in Partial Fulfillment of the  
Requirements for the Degree of  
Master of Science  
in Information Technology  
Assumption University**

06,2002

# The Faculty of Science and Technology

## Project Approval

Project Title            SS Apartment System


By                        Mr. Surakiat Somboonphokkaphan  
Project Advisor        Asst.Prof.Dr. Thotsapon Sortrakul

Academic Year        1/2002


---

The Department of Information Technology , Faculty of Science and Technology of Assumption University has approved this final report of the **three** credits course. **IT6900 Master Project**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology .


Approval Committee:



(Asst.Prof.Dr. Thotsapon Sortrakul)  
Advisor & Program Director



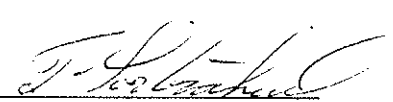
(Dr. Jirapun Daengdej)  
Committee Member



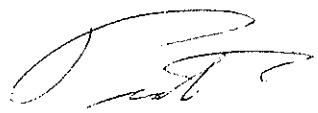
(Dr. Thiraphong Charoenkhunwiwat)  
Committee Member

---

Faculty Approval:



(Asst.Prof.Dr. Thotsapon Sortrakul)  
Program Director



(Asst.Prof.Dr. Pratit Santiprabhob)  
Dean

June/ 2002

## ACKNOWLEDGEMENTS

I would like to specially thank Asst. Prof. Dr.Thotsaphon Sortrakul, my advisor, for his valuable guidance and kind suggestions during the period of this project.

I would also like to thank all instructors of MS(IT) program for the knowledge they have provide. Finally, I would like to express my deepest gratitude for my family for giving their inspiration throughout the course of this project

Surakiat Somboonphokkaphan

Bangkok, March 2002



## ABSTRACT

Nowadays more expansion for room rental business making hectic competition; however the system used is still being a manual system. This make troubles for the tenants who need to know an up-to-date rental information and it is difficult for the room owner to supply information and transaction date. It is very easy to make mistakes and it also takes more time.

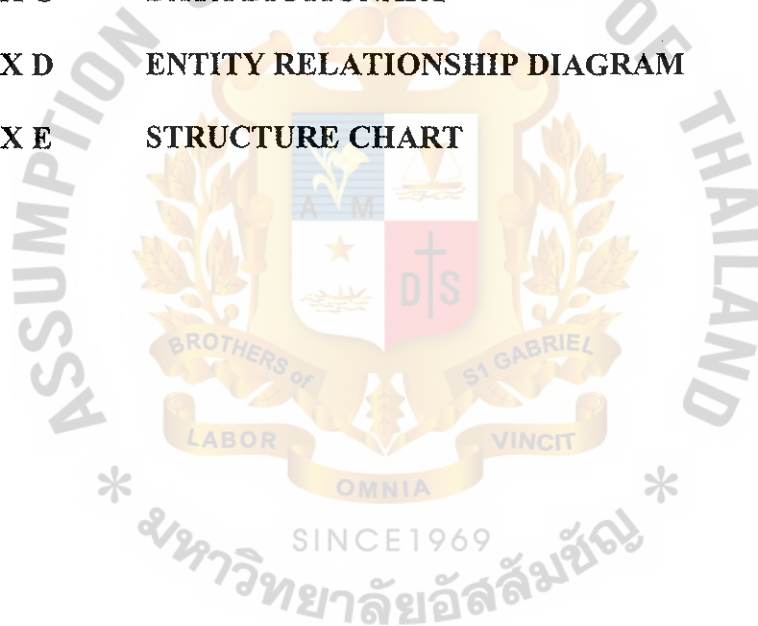
The proposed system has been designed to help improve such manual operations. The tools of structured analysis such as Data Flow Diagram, Structure Chart, and Data dictionary are used for analysis. The detailed design is carried out through file design, interface design, and report design. Computer system can assist in storing, adding, editing, searching and retrieving of data, which as a result will reduce the mentioned problems. It will also enhance this business to be more standardization and will certainly save more time in rental fee calculation. Thus, this system can help the owner to provide a better and faster service in order to gain a competitive advantage and be the leader of this business.

St. Gabriel's Library, An

## TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENT	I
ABSTRACT	II
TABLE OF CONTENTS	III
LIST OF FIGURES	V
LIST OF TABLES	IV
CHAPTER 1 INTRODUCTION	
1.1 Overviews	1
1.2 Objectives of the project	2
1.3 Scopes of the project	2
1.4 Outlines of the project	3
CHAPTER 2 THE EXISTING SYSTEM	
2.1 Background of the organization	4
2.2 Existing system	5
2.3 Current problems	5
CHAPTER 3 THE PROPOSED SYSTEM	
3.1 Methodology	7
3.2 User's requirement	8
3.3 System design	8
3.4 Cost vs. Benefit Analysis	9
CHAPTER 4 PROJECT IMPLEMENTATION & EVALUATION	
4.1 Project Implementation	14
4.2 Screen & Report Layout	15

	PAGE
4.3 Project Evaluation	39
CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS	
5.1 Conclusions	40
5.2 Recommendations	41
BIBLIOGRAPHY	42
APPENDIX A DATAFLOW DIAGRAM	
APPENDIX B PROCESS DESCRIPTION	
APPENDIX C DATA DICTIONARY	
APPENDIX D ENTITY RELATIONSHIP DIAGRAM	
APPENDIX E STRUCTURE CHART	



## LIST OF FIGURES

	<b>PAGE</b>
Figure 1.1      Project Outline of the SS Apartment System	3
Figure 2.1      Location of SS Apartment	4
Figure 3.1      Break Even Analysis Graph of SS Apartment System	13
Figure 4.1      Screen Layout of the User Login	15
Figure 4.2      Screen Layout of User Change Password	15
Figure 4.3      Screen Layout of Main Menu	16
Figure 4.4      Screen Layout of Add Data	16
Figure 4.5      Screen Layout of Data Deleting	17
Figure 4.6      Screen Layout of Data Editing	17
Figure 4.7      Screen Layout of Backup Data	18
Figure 4.8      Screen Layout of Restore Data	18
Figure 4.9      Screen Layout of Making Lease Agreement	19
Figure 4.10     Screen Layout of Reserving Room	19
Figure 4.11     Screen Layout of Cancelling Lease Agreement	20
Figure 4.12     Screen Layout of Adding Utility Data	20
Figure 4.13     Screen Layout of Editing Utility Data	21
Figure 4.14     Screen Layout of Printing Receipt	21
Figure 4.15     Screen Layout of Printing Memo	22
Figure 4.16     Screen Layout of Printing Report	22
Figure 4.17     Screen Layout of Printing Analysis Report	23
Figure 4.18     Screen Layout of Searching Data	23
Figure 4.19     Sign Agreement Report Layout	24

	<b>PAGE</b>
Figure 4.20 Reserve Room Report Layout	25
Figure 4.21 Utility Memo Report Layout	26
Figure 4.22 Monthly Summary Report Layout	27
Figure 4.23 Summary Utility Report Layout	28
Figure 4.24 Building List Report Layout	29
Figure 4.25 Receipt List Report Layout	29
Figure 4.26 Reserve List Report Layout	30
Figure 4.27 Utility List Report Layout	30
Figure 4.28 Room List Report Layout	31
Figure 4.29 Tenant List Report Layout	32
Figure 4.30 Receipt Report Layout	33
Figure 4.31 Annually Summary Report Layout	34
Figure 4.32 Annually Electric Used Report Layout	34
Figure 4.33 Annually Telephone Used Report Layout	35
Figure 4.34 Annually Water Used Report Layout	35
Figure 4.35 Annually Summary Graph Report Layout	36
Figure 4.36 Annually Electric Unit Graph Report Layout	36
Figure 4.37 Annually Telephone Graph Report Layout	37
Figure 4.38 Annually Water Graph Report Layout	37
Figure 4.39 Room Rent Graph Report Layout	38
Figure A.1 Context Diagram	A-1
Figure A.2 Data Flow Diagram Level 0 of SS Apartment System	A-2
Figure A.3 Data Flow Diagram Level 1 of Checking Requirement	A-3

	<b>PAGE</b>
Figure A.4 Data Flow Diagram Level 1 of Room Service	A-4
Figure A.5 Data Flow Diagram Level 1 of Searching Information	A-5
Figure A.6 Data Flow Diagram Level 1 of Maintain Information	A-6
Figure A.7 Data Flow Diagram Level 1 of Room Payment	A-7
Figure A.8 Data Flow Diagram Level 1 of Generate Report	A-8
Figure A.9 Data Flow Diagram Level 2 of Contraction Process	A-9
Figure A.10 Data Flow Diagram Level 2 of Cancel Contraction Process	A-9
Figure A.11 Data Flow Diagram Level 2 of Reservation Process	A-10
Figure A.12 Data Flow Diagram Level 2 of Search Room Information	A-10
Figure A.13 Data Flow Diagram Level 2 of Search Tenant Information	A-11
Figure A.14 Data Flow Diagram Level 2 of Calculate Water Cost	A-11
Figure A.15 Data Flow Diagram Level 2 of Calculate Electric Cost	A-12
Figure A.16 Data Flow Diagram Level 2 of Calculate Telephone Cost	A-12
Figure A.17 Data Flow Diagram Level 2 of Generate Payment Receipt	A-13
Figure A.18 Data Flow Diagram Level 2 of Generate Monthly Report	A-13
Figure A.19 Data Flow Diagram Level 2 of Generate Annual Report	A-13
Figure C.1 Entity Relationship Diagram of SS Apartment System	C-1
Figure E.1 Structure Chart of SS Apartment System	E-1
Figure E.2 Structure Chart of Checking Requirement	E-2
Figure E.3 Structure Chart of Room Service	E-3
Figure E.4 Structure Chart of Searching Information	E-4
Figure E.5 Structure Chart of Maintain Information	E-5
Figure E.6 Structure Chart of Room Payment	E-6

Figure E.7      Structure Chart of Generate Report



LIST OF TABLES

	PAGE
Table 4.1      Comparison of each process between existing system and proposed system	39



# 1. INTRODUCTION

## 1.1 Overviews

SS Apartment is a family business apartment, which is located at Soi. Samsen 19, Dusit area in Bangkok, Thailand. It provides 103 rooms for tenants. Most of the tenants are students and worker age groups.

Traditionally, SS Apartment has used the manual system to operate and to control all the processes such as leasing agreement, check-in, receipt management, filing, utility fee and so on. The manual system had caused many problems, for instance, information of tenant is outdated, customer service delayed, too many documents need to be collected, receipt is inaccurate, documents are lost. This is very difficult to control and to create a standard for all tenant rules. Nowadays in this competitive world; fast service, accurate information, and updated environment are the added value to obtain the competitive advantage beside monthly rental fee and location. The owner has realized this fact and has decided to implement a new computerized system to apply for the business. This new system is called SS Apartment System (SSAS). This application will be implemented on Personal Computer stand alone PC to keep all details and information for utilization.

## 1.2 Objectives of the project

The objectives of this project are as following:

1. To accurately store tenant's and room data
2. To easily enhance the efficiency of adding, editing, searching of tenant's and room data
3. To save time in checking the status for available or vacant room
4. To save time in calculating monthly rental fee and utility fee of each room
5. To increase the system reliability
6. To increase the effectiveness of issuing the receipt
7. To automatically issue transactional report for reference, for prediction and for analyzing the trend of the business
8. To reduce paper work by storing all information in hard disk

## 1.3 Scopes of the Project

The scopes of developing the new system to cover all the tasks and activities are listed as below:

1. The system can help the owner to easily maintain and update the information
2. The system can generate reports to help in owner's decision making
3. The timely status report of the organization. It can generate monthly as well as yearly report
4. The simple and user friendly design of the screen layout for the new system.

## 1.4 Outlines of the Project

The project outline is represented in term of Gantt Chart as shown below.

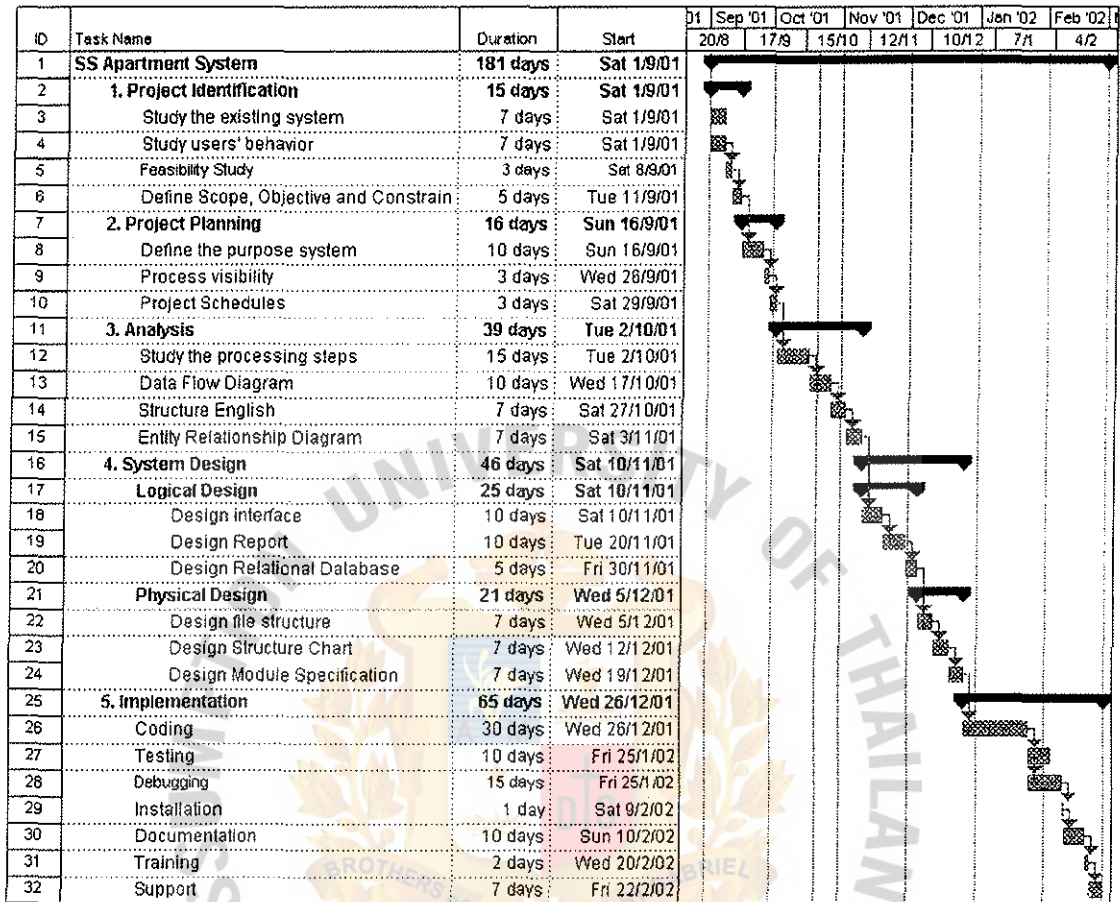


Figure 1.1 Project Outline of the SS Apartment System

## 2. THE EXISTING SYSTEM

### 2.1 Background of the organization

There are altogether 4 buildings with a total of 103 rooms: Building A has 22 rooms, Building B has 18 rooms, Building C has 31 rooms, and Building D has 32 rooms. The building layout is shown in Figure 1.1. This apartment is located near to an education institute and it is very convenient for transportation. Most tenants are students and salary man.

SS Apartment all the while has been using a manual system, which resulted in a repeated job operation. This caused the owner to spend more unnecessary time on their services and operations. It also requires a lot of paper work which, resulted, in process, less of documents, and complicated filing. Therefore, the computerized system is one of the methods to help solve their daily problems and to improve their business to be more standardized

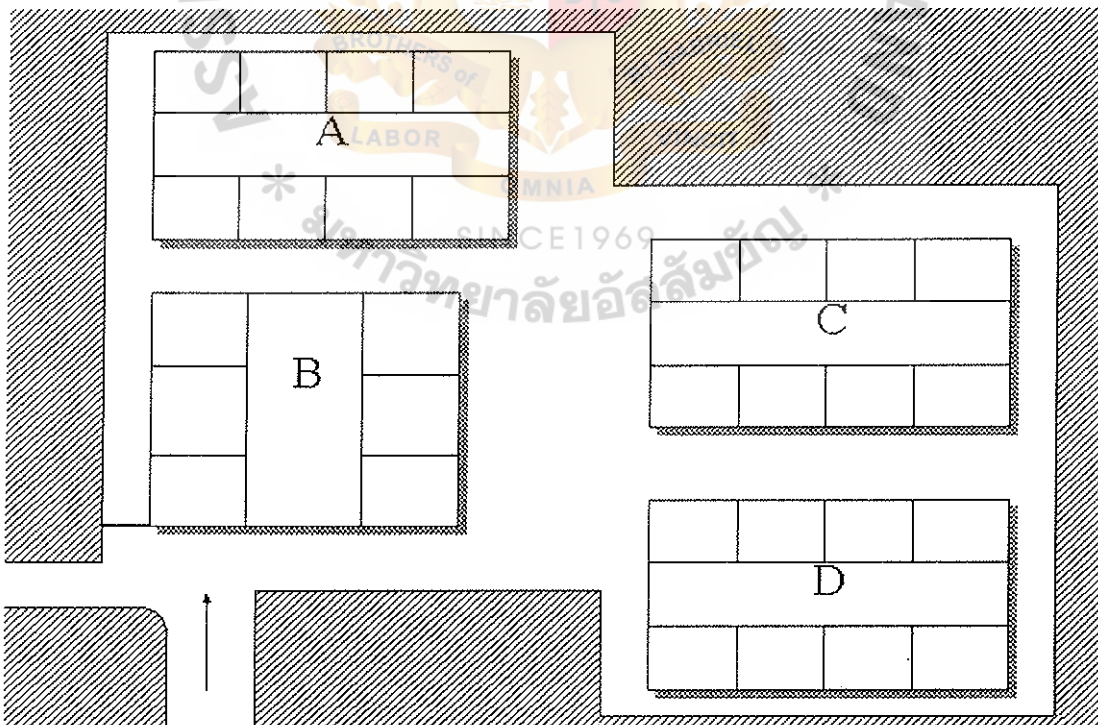


Figure 2.1 Location of SS Apartment

## 2.2 The Existing System

The functions of the existing system are manual and can be classified as below:

1. Apartment owner cannot accurately provide their new customers with the available or vacant room because he must remember the list himself.
2. When doing a leasing, the customer must read the contract and sign the agreement form, which resulting in a slow process.
3. The apartment owner must calculate a monthly utility fee for each room and issue receipt for both the utility fees and the rental fees to the tenants. Unfortunately, they always do the wrong calculation.
4. The apartment owner must do the monthly and yearly reports, rental income, utility fee cost and utility paid by the tenant's and it is extremely difficult for him to do the yearly report by himself.
5. When a tenant moves out, the apartment owner must calculate all the expenses and return the bond to home. He also has to record the status of room as available or vacant, when the customers ask for a room reservation.
6. Unfortunately, when a customer reserve the room, the apartment owner just gives an oral promise instead of written document.

## 2.3 The Current Problems

### 2.3.1 Current Problem

The existing system can cause many problems as follow:

1. Due to the manual system, the apartment owner uses only his own memory and calculator to manage room status and lease agreement form and calculate the monthly rental fees, the utility fees and the bonding fees for all 103 rooms. That is very time consuming. However, the new system can resolve all the above problems

2. There are a lot of paper works each year and it requires a lot of space to keep all the documents. And there is always a risk of document loss or damaged.
3. Lack of standardization. There is no record of room status (Available/Vacancy). They only remember which room is available for rental.
4. There is a miscalculation of all fees, which result in lost of customers confident in paying their bills.
5. Discontinue of workflow. If the apartment owner is not free or has bad handwriting, no one will know and understand the detailed job description well.

All of the above mentioned are the major problems for SS Apartment business. Therefore, we decided to design a proposed system to solve the problems. We will later describe in deep detail.

### *2.3.2 Area of Improvement*

The area of improvement consists of the following:

1. To improve service quality. The existing system promises customer of all apartment service by verbally. But the new system provides on the paper.
2. To reduce work redundancy, for instance, the apartment owner has to generate receipt room by room.
3. To provide the report system. The existing system is very hard to generate because it stores the data room by room.
4. To reduce time consuming for all services. See Project Evaluation.
5. To improve data security. In the existing system, if the data is stolen or lost, the apartment owner cannot generate the receipt.

### 3. THE PROPOSED SYSTEM

#### 3.1 Methodology

This project is implemented by SDLC. It is composed of many sub-processes and each of them could not be implemented separately because they are related to one another. In this case to develop the system for SS Apartment, it requires a clear understanding of the whole processes, so that all of the requirements could be delivered.

The waterfall model is used in this implementation because it is best suited for SS Apartment system. The advantage of this model is it can easily manage the components, each of which has a well defined derivable. This model will also provide a high visibility in term of documentation and is easy to track and verify progress.

This project will cover all steps of the SDLC methodology. There are 5 steps of SDLC as follow:

1. Project Identification Phase studies the existing system, user's behavior, feasibility and also define scope, and objective.
2. Project Planning Phase. This phase defines the purposed system and project schedules.
3. Analysis & Design Phase studies the processing steps, creates Data Flow Diagram, and Entity Relationship Diagram and designs input/output Layout.
4. Implementation Phase starts to do the coding, testing and debugging, installation and training user to use the proposed system.
5. Maintenance Phase. The developer has to support the user when the proposed system has errors and fix the errors.

### 3.2 User's Requirement

1. The system should create apartment image standardization.
2. The system should reduce time consuming work process.
3. The system must provide an accurate and reliable calculation.
4. The system should help the apartment owner to facilitate in updating tenant's data and building data.
5. The system will save storage of data in computer system instead of paper document.
6. The system should be user friendly and easy for anyone to learn or continue working
7. The system can provide periodic report and also predict trend of the business.
8. The system can print out all report and receipt and also can provide all information for user.
9. The system also provides security, it can identify user by using password.

### 3.3 System design

The objective of the system design is to convert the requirements into a computer solution to improve operations, services and controls. The system design describes the data input, calculation, or storage. Data items and calculation procedures are written in detail. Designers select file structures and storage devices such as magnetic disk, or even paper file. The procedure they write tells how to process the data and procedure it to the output. The detailed design information (Charts, Tables and special Symbols) is passed on to software development.

3.3.1 *Data Flow Diagram*. See Appendix A.

3.3.2 *Process Specification*. See Appendix B.

3.3.3 *Entity Relationship Diagram*. See Appendix C.

3.3.4 *Data Dictionary* . See Appendix D.

3.3.5 *Structure Chart*. See Appendix E.

### 3.4 Cost vs. Benefit Analysis

The proposed system requires the following hardware components:

1. Personal Computer 1 Set
  - CPU Celeron 900 MHz
  - SD RAM 128 MB
  - Hard disk 20 GB
  - Floppy disk 1.44 MB
  - Monitor 15"
  - CD-ROM 52x
  - Keyboard and Mouse
2. Printer 2 Sets
  - Laser Printer (HP LaserJet 1200)
  - Dot Matrix (Epson 2080i)
3. Scanner (HP ScanJet 2200C) 1 Set
4. UPS (Syndome 500VA) 1 Set

The Software requirements can be summarized as the followings:

1. Microsoft Windows 98 Second Edition
2. Microsoft Visual Studio 6
3. Microsoft Office 2000 (Professional)

### 3.4.1 Cost Analysis

For the cost of SS Apartment system development, there are two categories that must be taken into consideration when developing the proposed system. These two categories are:

#### 3.4.1.1 Estimated One Time Cost

##### 1. Hardware

• 1 set of Personal Computer	35,000฿	
• 1 Laser Printer	18,000฿	
• 1 Dot Matrix Printer	22,000฿	
• 1 Scanner	4,500฿	
• 1 UPS	3,500฿	83,000฿

##### 2. Software

• Ms Window 98 Second Edition	3,500฿	
• Microsoft Visual Studio 6	18,000฿	
• Ms Office 2000 (Professional)	28,500฿	50,000฿

##### 3. Development

• Software Development		
(100 Days X 1500 Baht/Day)	150,000฿	
• Training (2 Days X 1500 Baht/Day)	3,000฿	153,000฿

**Total One Time Cost** **286,000฿**

1. Maintenance (per year)	20,000฿	20,000฿
---------------------------	---------	---------

Repairing and upgrade all equipment,  
hardware and software

2. IS Support (per year)	24,000฿	24,000฿
--------------------------	---------	---------

(12 Times X 2000 Baht/Time)

<b>Total Recurring Cost</b>		<b>44,000฿</b>
-----------------------------	--	----------------

### 3.4.2 Benefit Analysis

The benefit of SS Apartment system is not only for increasing the efficiency of the room service, it also provides many other benefits mostly intangible. However, the benefits from the proposed system can probably be projected in both tangible and intangible as follows:

#### 3.4.2.1 Tangible Benefits

1. Estimate to increase the leasing room amount by 5%

#### 3.4.2.2 Intangible Benefits

1. Retrieve information faster.
2. Better provide various kinds of reports.
3. Improve decision making of the apartment owner.
4. Reduce risk of human errors, which can be made by human.
5. Reduce the volume of paper work and time consumption.
6. Increase good will of the apartment.
7. Increase customer satisfaction.

# SS Apartment

## Cost and Benefit Analysis

	Year0	Year1	Year2	Year3	Year4	Year5	Total
Net economic benefit	0	132,000	264,000	396,000	396,000	396,000	
Discount Rate (7.25%)	1.0000	0.9324	0.8693	0.8106	0.7558	0.7047	
PV of benefit	0	123,077	229,495	320,998	299,297	279,061	
NPV of all Benefits	0	123,077	352,572	673,570	972,866	1,251,928	1,251,928

Onetime Cost -286,000

Recurring Costs	0	-44,000	-44,000	-44,000	-44,000	-44,000	
Discount Rate (7.25%)	1.0000	0.9324	0.8693	0.8106	0.7558	0.7047	
PV of Recurring Cost	0	-41,025	-38,250	-35,666	-33,255	-31,007	
NPV of all Costs	-286,000	-327,025	-365,275	-400,941	-434,196	-465,203	-465,203

Overall NPV 786,724

Overall ROI - (Overall NPV / NPV of all Costs) 1.69

### Break Even Analysis

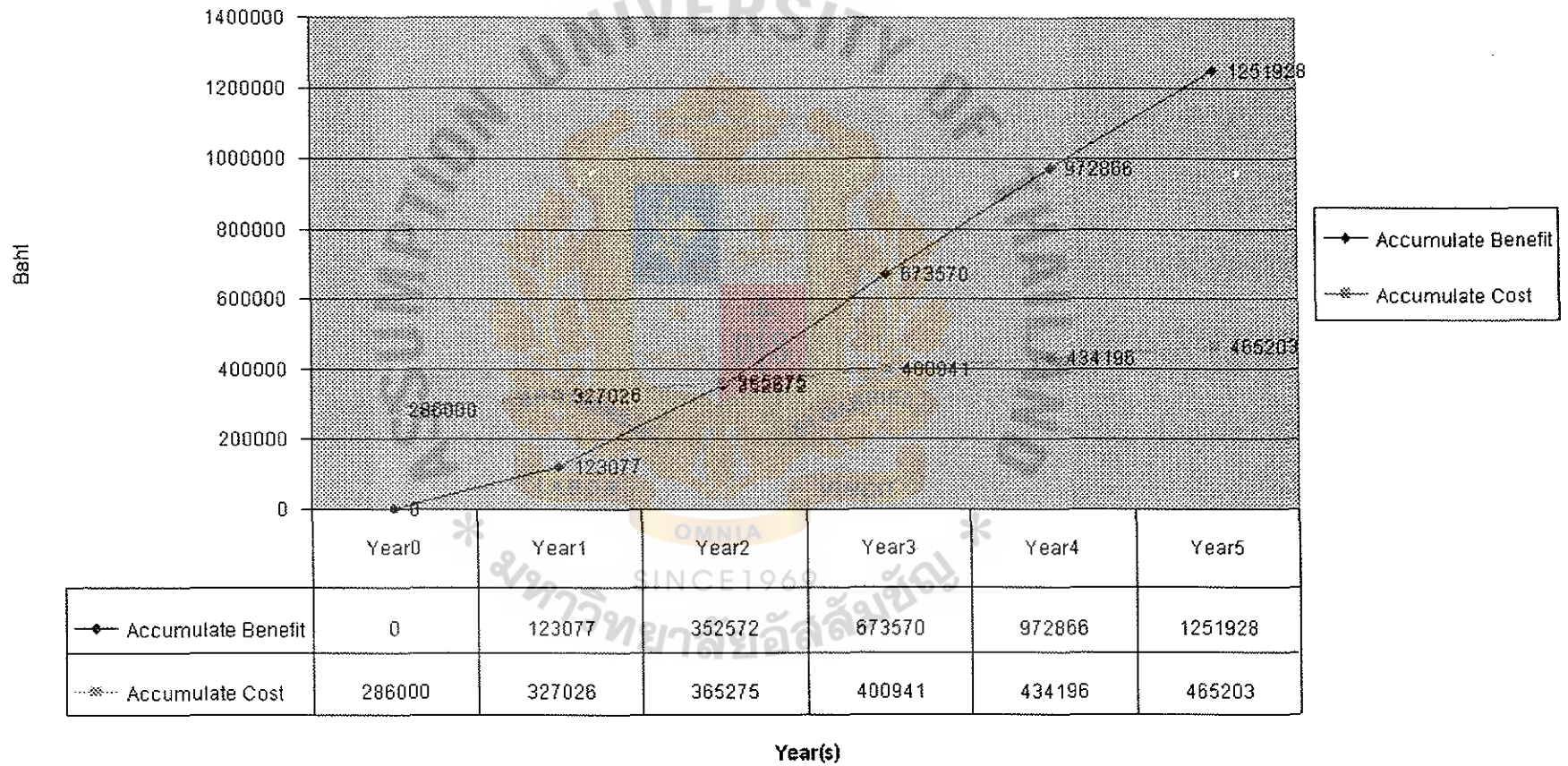
Yearly NPV Cash Flow	-286,000	82,051	191,246	285,331	266,042	248,054
Overall NPV Cash Flow	-286,000	-203,949	-12,703	272,628	538,670	786,724

Project break-even occurs between year2 and year3

Use first year of positive cash flow calculate break-even fraction = (1,682,080 – 1,396,050) / 1,682,080 0.04

Actual break-even occurred at 2.04 Year

### Break Even Analysis of SS Apartment



## 4. PROJECT IMPLEMENTATION & EVALUATION

### 4.1 Project Implementation

The system implementation is one of the phases in the system development that is important for overall success. For this system, the conversion plan can replace the existing system by the new system.

There are three activities to perform the new system as follows:

1. *Collect the existing data*

The apartment owner must collect the existing data and prepare to store into the database of the new system

2. *Install the new system*

The developer must send the SS Apartment System Software Package and train the user to install the new system.

3. *Train the system user*

After the new system has been installed, the developer should train the users to use all of the functions of the SS Apartment System such as backing up/restoring the database, generating all forms of report, and making the leasing agreement.

4.2 Screen & Report Layout

4.2.1 Screen Layout

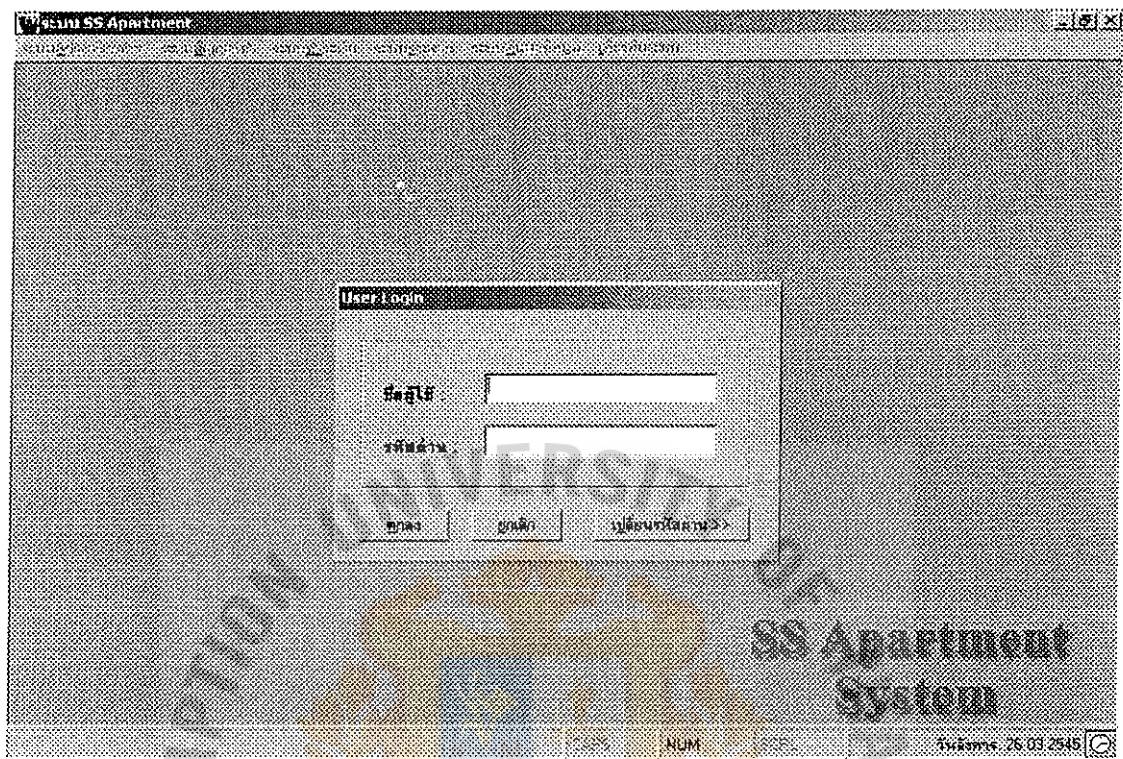


Figure 4.1 Screen Layout of the User Login

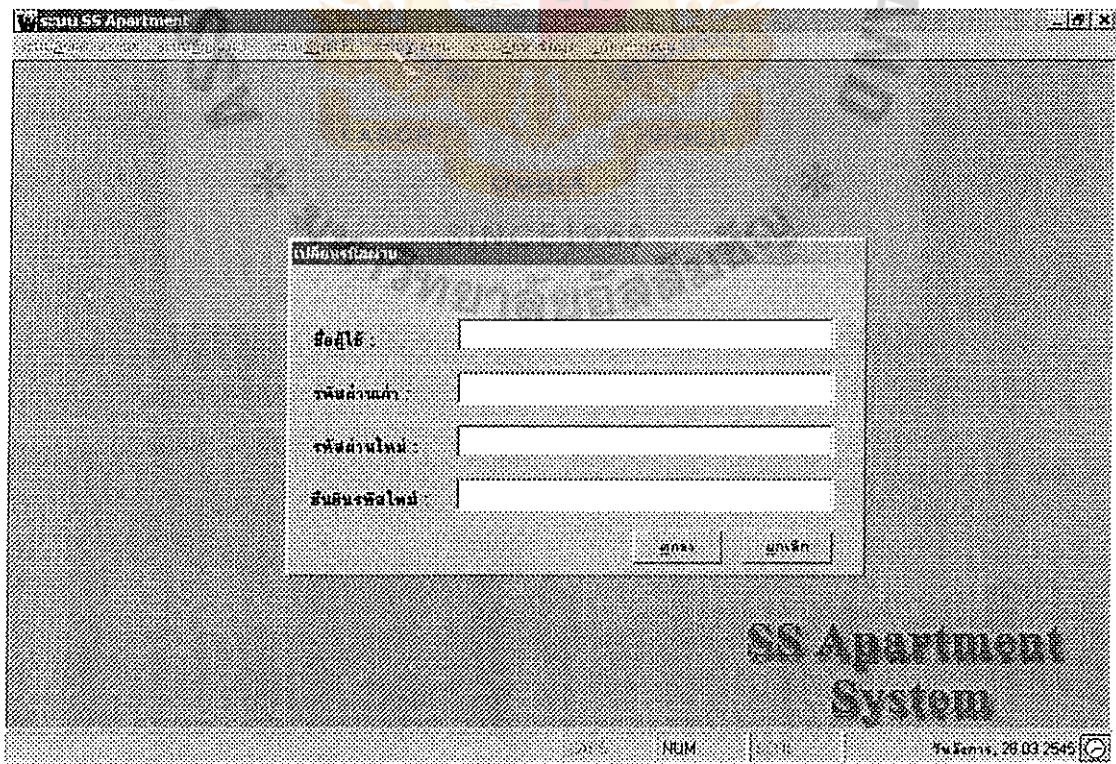


Figure 4.2 Screen Layout of User Change Password

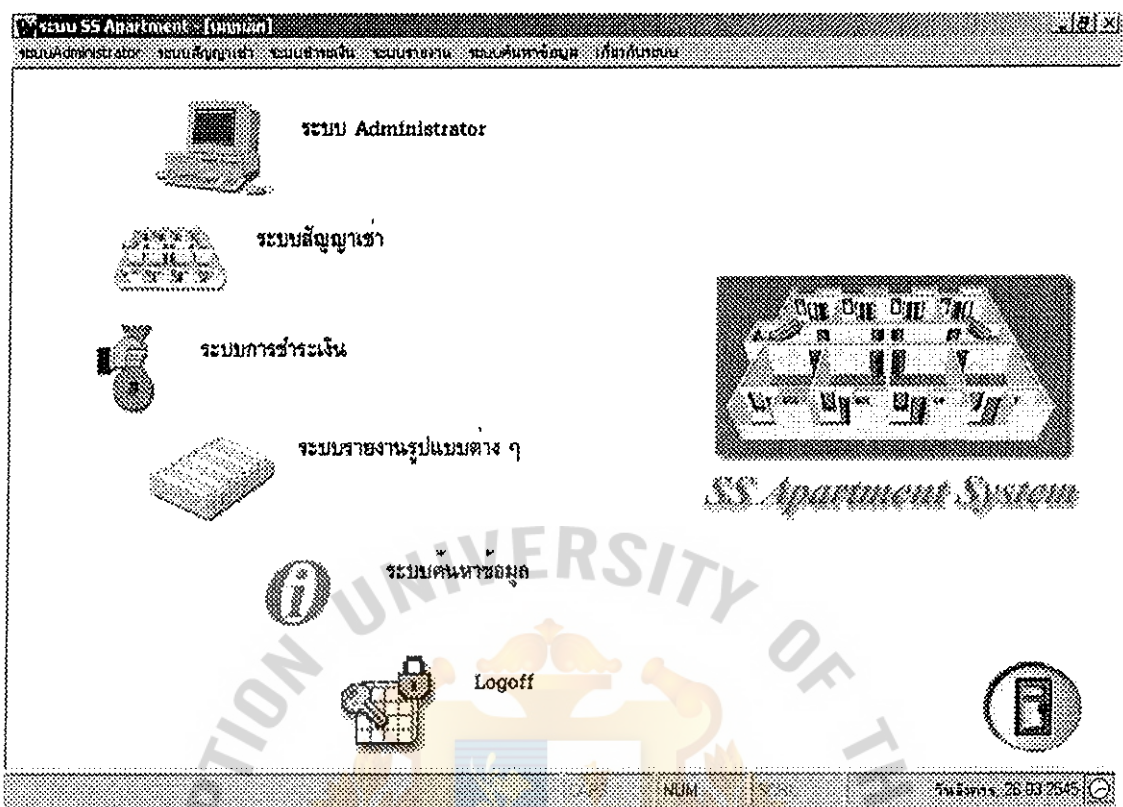


Figure 4.3 Screen Layout of Main Menu

Figure 4.4 Screen Layout of Add Data

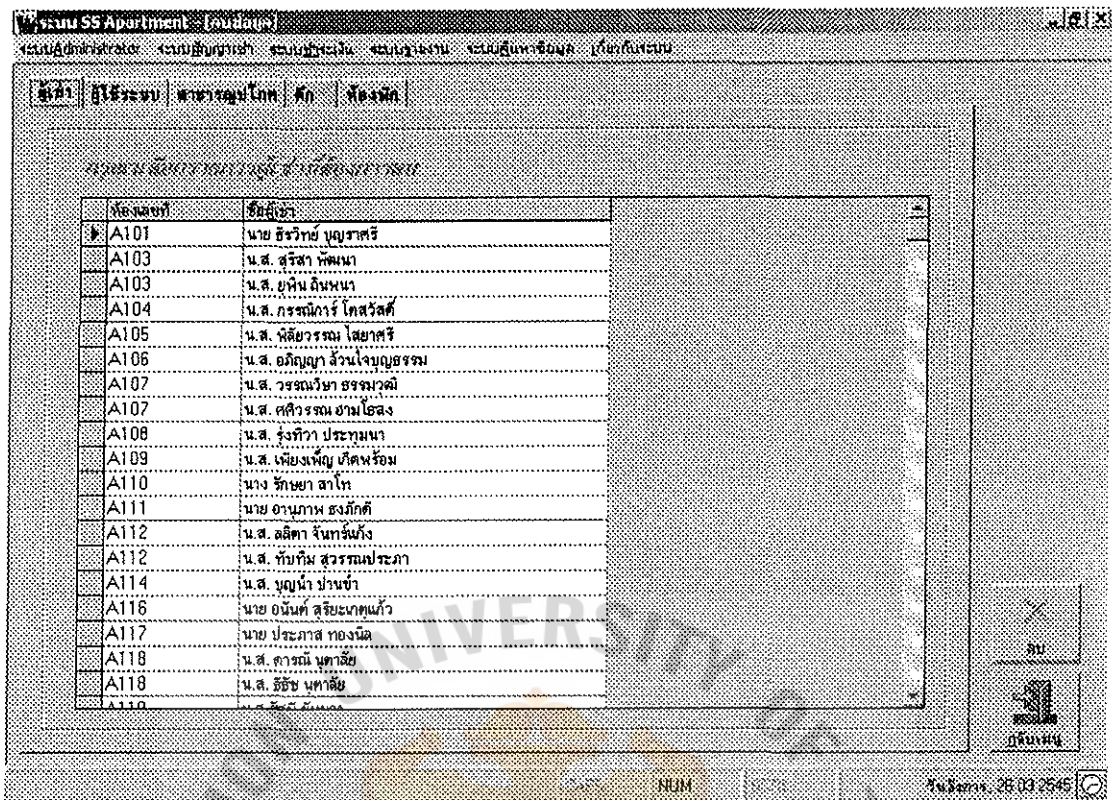


Figure 4.5 Screen Layout of Data Deleting

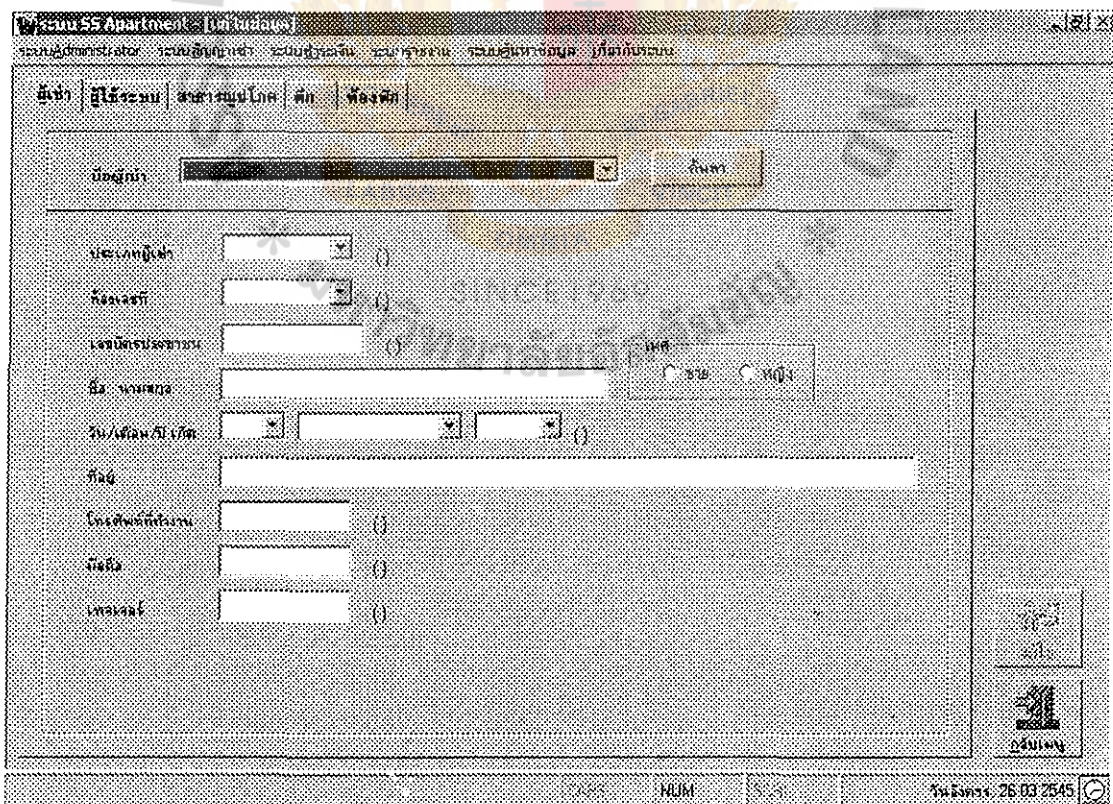


Figure 4.6 Screen Layout of Data Editing

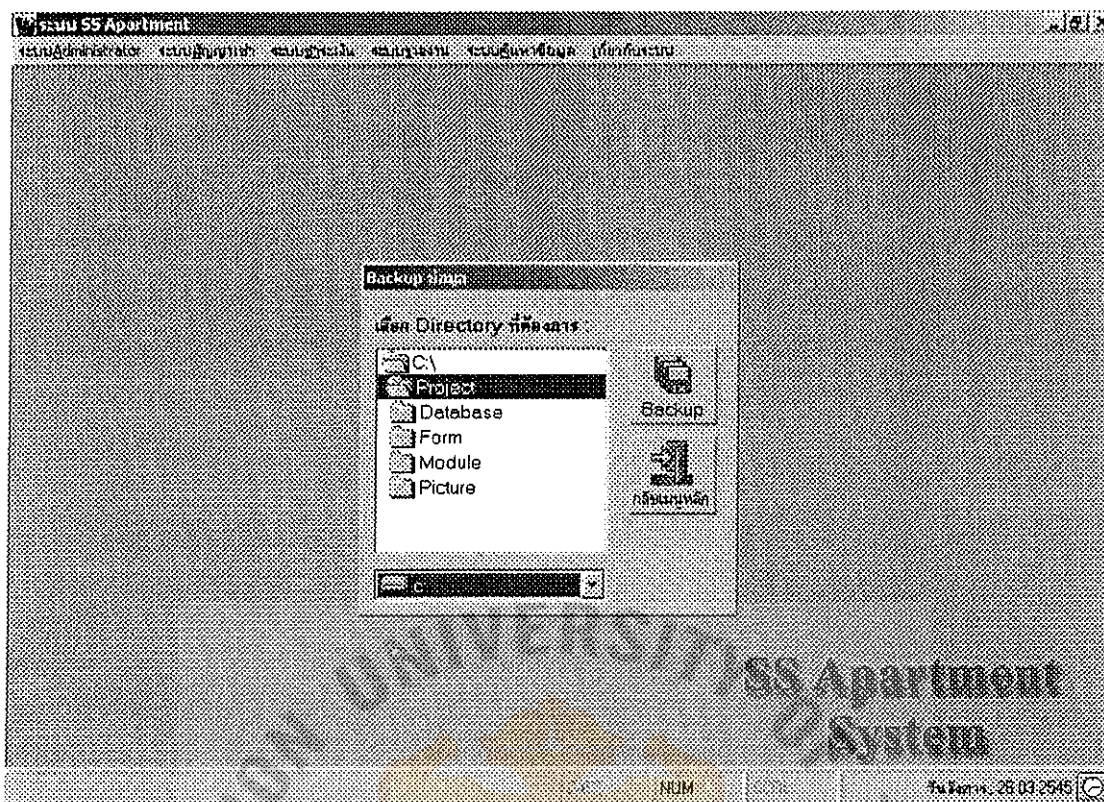


Figure 4.7 Screen Layout of Backup Data

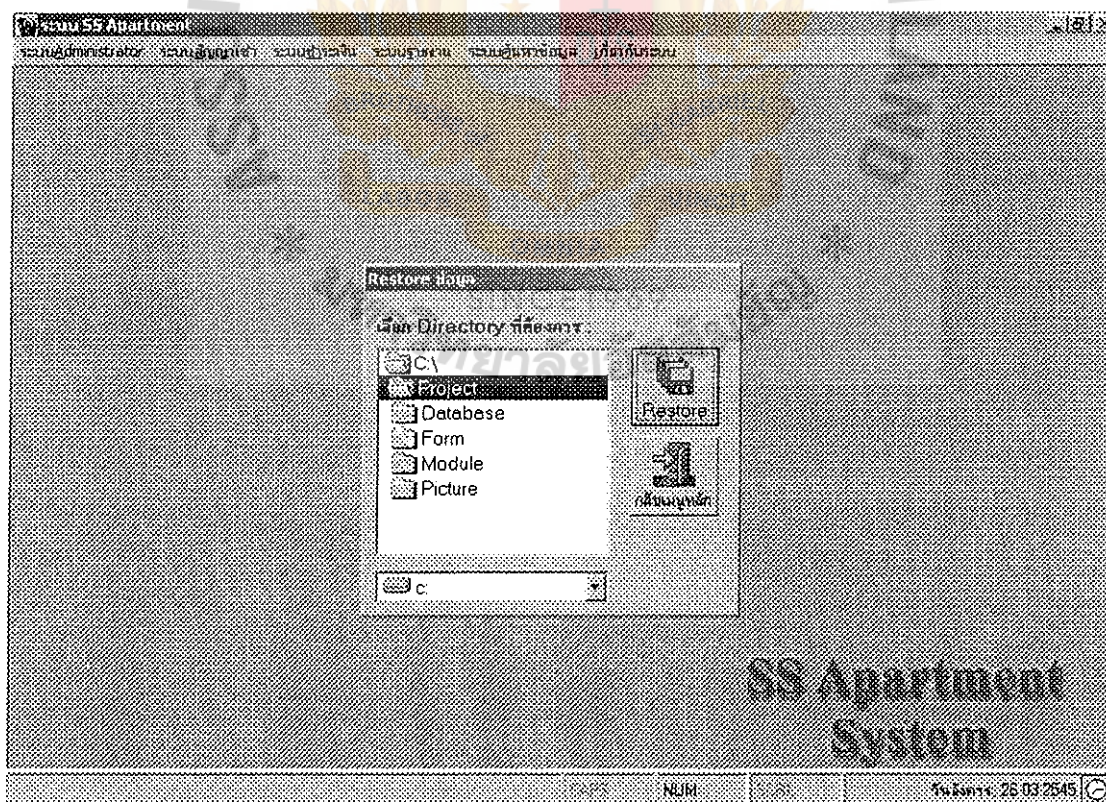


Figure 4.8 Screen Layout of Restore Data

ระบบ 55 Apartment - [ข้อมูลส่วนตัว]

ระบบผู้ดูแลระบบ ระบบผู้เช่าระบบ ระบบผู้เช่าระบบ ระบบผู้เช่าระบบ ระบบผู้เช่าระบบ

ผู้เช่า | ผู้เช่าที่ 1 | ผู้เช่าที่ 2

ข้อมูลห้องพัก

ห้องเลขที่  ส่วนต่อเติม  บาท

เงินส่วนต่อเติม  จำนวน  บาท

ข้อมูลผู้เช่า

เลขบัตรประชาชน

ชื่อ - นามสกุล

วัน/เดือน/ปีเกิด

เพศ ☐ ชาย ☐ หญิง

บ้านเลขที่  หมู่ที่  ซอย/ตรอก

ถนน  ตำบล/แขวง

เขต/อำเภอ  จังหวัด

โทรศัพท์มือถือ  อีเมล  เพจเฟซบุ๊ก

วันที่  เวลา

NUM

วันที่ 26 03 2545

Figure 4.9 Screen Layout of Making Lease Agreement

ระบบ 55 Apartment - [ข้อมูลส่วนตัว]

ระบบผู้ดูแลระบบ ระบบผู้เช่าระบบ ระบบผู้เช่าระบบ ระบบผู้เช่าระบบ ระบบผู้เช่าระบบ

ข้อมูลห้องพัก

ห้องเลขที่  ส่วนต่อเติม  บาท

คิดค่าส่วนต่อเติม

ข้อมูลผู้จอง

เลขบัตรประชาชน

ชื่อ - นามสกุล

เงินมัดจำค่าเช่าห้อง  บาท

บ้านเลขที่  หมู่ที่  ซอย/ตรอก

ถนน  ตำบล/แขวง

เขต/อำเภอ  จังหวัด

โทรศัพท์มือถือ

วันที่  เวลา

NUM

วันที่ 26 03 2545

Figure 4.10 Screen Layout of Reserving Room

Figure 4.11 Screen Layout of Cancelling Lease Agreement

Figure 4.12 Screen Layout of Adding Utility Data

โปรแกรม SS Apartment (โปรแกรมสำหรับอาคารชุด)

ระบบAdministrator ระบบผู้เช่าระบบผู้ดูแลระบบฐานข้อมูลระบบค้นหาข้อมูลเกี่ยวกับระบบ

ใบเสร็จเลขที่  วันที่

รายละเอียดใบเสร็จ

ค่าน้ำ

ค่าไฟฟ้า

เดือนก่อนหน้า : เดือนก่อน  เดือนนี้  ()

เดือนก่อนหน้า : เดือนก่อน  เดือนนี้  ()

ค่าโทรศัพท์ :  ()

ปุ่ม: ค้นหา, แก้ไข, ลบ, เพิ่ม, พิมพ์, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, Enter, Esc, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Print, Ctrl, Alt, Shift, Tab, Space, Backspace, Delete, Insert, Home, End, Page Up, Page Down, Arrow Keys, Num Lock, Caps Lock, Esc, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Print, Ctrl, Alt, Shift, Tab, Space, Backspace, Delete, Insert, Home, End, Page Up, Page Down, Arrow Keys, Num Lock, Caps Lock

วันที่พิมพ์: 26-03-2545

Figure 4.13 Screen Layout of Editing Utility Data

โปรแกรม SS Apartment (โปรแกรมสำหรับอาคารชุด)

ระบบAdministrator ระบบผู้เช่าระบบผู้ดูแลระบบฐานข้อมูลระบบค้นหาข้อมูลเกี่ยวกับระบบ

เลือกข้อมูลที่จะพิมพ์

☐ พิมพ์รวมทั้งหมด

☐ พิมพ์รวมค่า

☐ พิมพ์ทั้งหมด

ข้อมูลที่จะพิมพ์

Field:

การตั้งค่าเลือกเดือนปีที่จะบันทึก

ปุ่ม: ค้นหา, แก้ไข, ลบ, เพิ่ม, พิมพ์, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, Enter, Esc, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Print, Ctrl, Alt, Shift, Tab, Space, Backspace, Delete, Insert, Home, End, Page Up, Page Down, Arrow Keys, Num Lock, Caps Lock

วันที่พิมพ์: 26-03-2545

Figure 4.14 Screen Layout of Printing Receipt

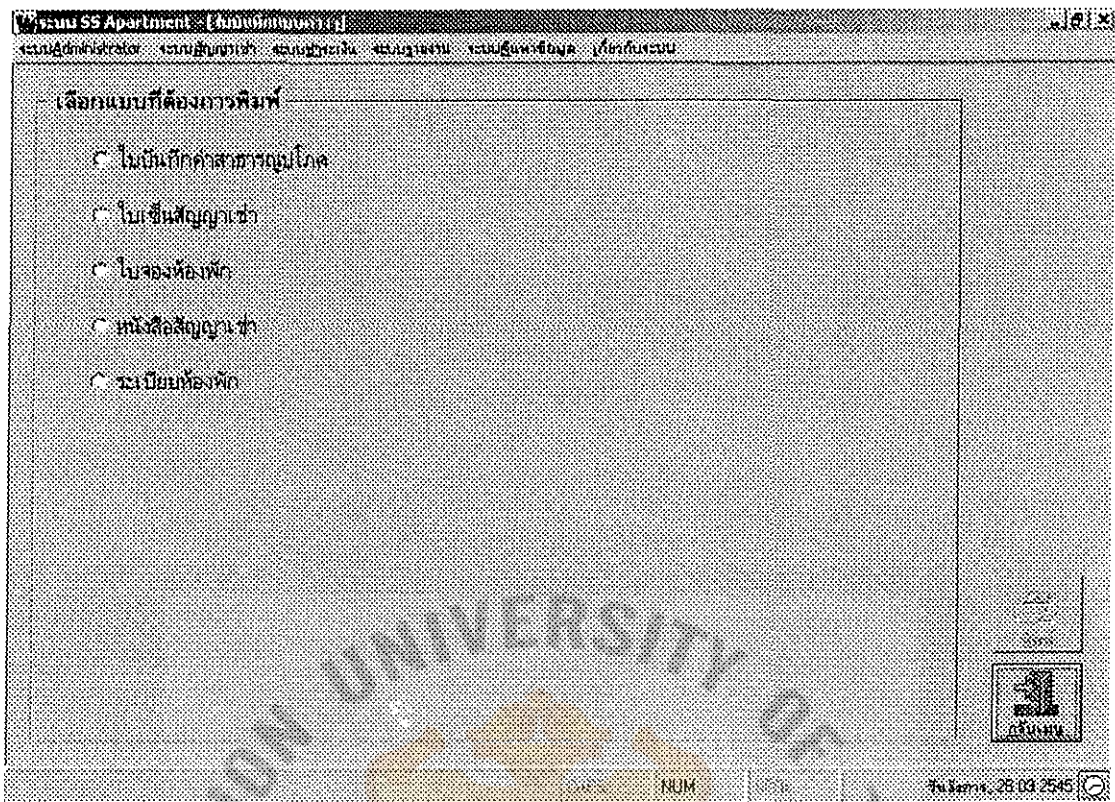


Figure 4.15 Screen Layout of Printing Memo

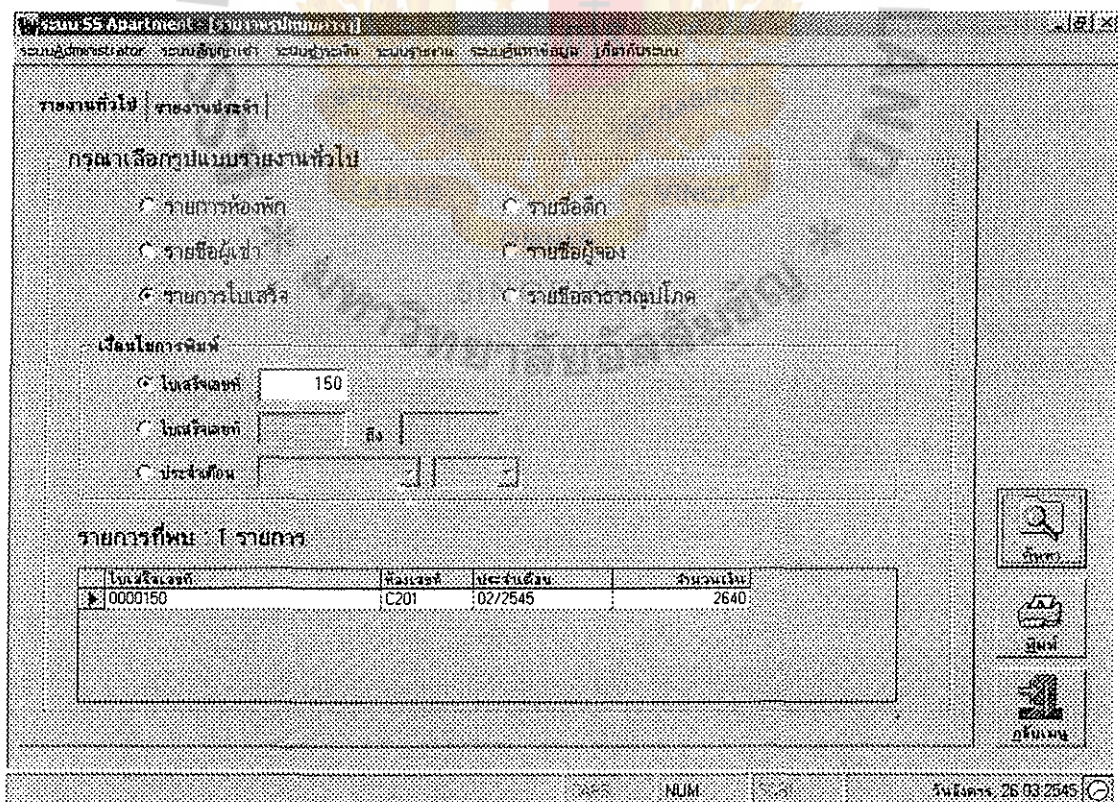


Figure 4.16 Screen Layout of Printing Report



#### 4.2.2 Report Layout

### ใบสัญญาเช่าห้องพัก

ทำที่ บ้านเลขที่ 179 ซอยสามเสน 19

ถนนนครไชยศรี แขวงถนนนครไชยศรี

ตลิ่ง กรุงเทพมหานคร

วันที่ \_\_\_\_\_

สัญญาเช่าทำขึ้นระหว่าง \_\_\_\_\_ ซึ่งต่อไปในสัญญานี้จะเรียกว่า

"ผู้ให้เช่า" ฝ่ายหนึ่งกับ \_\_\_\_\_ ถือบัตรประชาชนเลขที่ \_\_\_\_\_

อยู่บ้านเลขที่ \_\_\_\_\_

ซึ่งต่อไปในสัญญานี้จะเรียกว่า "ผู้เช่า" ซึ่งฝ่ายหนึ่งโดยทั้งสองฝ่ายตกลงทำสัญญาดังกล่าวกันมีรายละเอียดดังนี้

ข้อ 1. ผู้เช่าตกลงเช่าห้องพักอาศัยของผู้ให้เช่า \_\_\_\_\_ เพื่อใช้อยู่อาศัยในอัตรา

ค่าเช่าห้องเดือนละ \_\_\_\_\_ บาท ในรวมค่าไฟฟ้า ค่าน้ำประปา และค่าโทรศัพท์นับตั้งแต่วันที่ \_\_\_\_\_

ข้อ 2. ผู้เช่าได้วางเงินประกันความเสียหายที่อาจจะเกิดขึ้นเนื่องจากการเช่าห้องพักนี้ไว้กับผู้ให้เช่า

สัญญานี้เป็นเงิน \_\_\_\_\_

ข้อ 3. คู่สัญญาทั้งสองฝ่ายได้อ่านข้อความและเข้าใจหนังสือสัญญาเช่าและระเบียบการเช่าห้องพักแล้ว

จึงลงลายมือชื่อไว้เป็นหลักฐาน

ลงชื่อ \_\_\_\_\_ ผู้ให้เช่า  
(.....)

ลงชื่อ \_\_\_\_\_ ผู้เช่า  
(.....)

ลงชื่อ \_\_\_\_\_ พยาน  
(.....)

Figure 4.19 Sign Agreement Report Layout

## ใบของห้อง

ข้าพเจ้า \_\_\_\_\_ กิ่ขอบัตรประชาชนเลขที่ \_\_\_\_\_  
อยู่บ้านเลขที่ \_\_\_\_\_  
ได้จองห้องพักเลขที่ \_\_\_\_\_ กับผู้ให้เช่าโดยได้วางเงินมัดจำค่าจองเป็นจำนวนเงิน \_\_\_\_\_ บาท  
นับตั้งแต่วันที่ \_\_\_\_\_ ถึงวันที่ \_\_\_\_\_ โดยผู้จองไม่มีสิทธิ์เรียกเงินค่ามัดจำ  
คืนวันแต่ได้เซ็นสัญญาเช่าห้องพักกับผู้ให้เช่าเรียบร้อยแล้ว

ผู้จอง \_\_\_\_\_  
(.....)

ผู้ให้เช่า \_\_\_\_\_  
(.....)

Figure 4.20 Reserve Room Report Layout

๔. ๓๓

[illegible]

26

## รายงานสรุปยอดประจำเดือน

เงื่อนไขการพิมพ์ ติก 8

ประจำเดือน มกราคม/2545

ลำดับที่	ห้องเลขที่	ใบเสร็จเลขที่	จำนวนเงิน	หมายเหตุ
1	B201	0000086	2,506	
2	B202	0000086	2,422	
3	B203	0000087	2,376	
4	B204	0000088	2,504	
5	B205	0000089	2,449	
6	B206	0000090	2,560	
7	B301	0000091	3,076	
8	B302	0000092	2,865	
9	B303	0000093	2,800	
10	B304	0000094	2,555	
11	B305	0000095	2,348	
12	B306	0000096	2,376	
13	B401	0000097	2,572	
14	B402	0000098	2,655	
15	B403	0000099	2,328	
16	B404	0000100	2,422	
17	B405	0000101	2,648	
18	B406	0000102	2,996	
ยอดรวม			46,437	

Figure 4.22 Monthly Summary Report Layout

## สรุปการใช้สาธารณูปโภคประจำเดือน

เงื่อนไขการพิมพ์: ติ๊ก Bประจำเดือน: มกราคม/2546

ห้องเลขที่	หน่วยไฟฟ้า	หน่วยประปา	ค่าไฟฟ้า	ค่าประปา	ค่าบริการโทรศัพท์	ค่าโทรศัพท์
B201	34	8	136	120	50	0
B202	28	4	112	60	50	0
B203	24	2	96	30	50	0
B204	41	6	164	90	50	0
B205	31	5	124	75	50	0
B206	60	4	240	60	50	0
B301	169	10	676	150	50	0
B302	110	11	440	165	50	0
B303	100	10	400	150	50	0
B304	65	3	260	45	50	0
B305	17	2	68	30	50	0
B306	24	2	96	30	50	0
B401	58	6	232	90	50	0
B402	75	7	300	105	50	0
B403	12	2	48	30	50	0
B404	28	4	112	60	50	0
B405	62	10	248	150	50	0
B406	100	23	400	345	50	0
ยอดรวม			4,152	1,785	900	0

Figure 4.23 Summary Utility Report Layout

ใบรายชื่อเด็ก

พิมพ์ ณ วันที่ 03/เมษายน/2546

ลำดับที่	ชื่อเด็ก	ที่อยู่	จำนวนห้อง
1	A	179 ซ.สามเสน 19 ถนนนครไชยศรี แขวงถนนนครไชยศรี เขตดุสิต กรุงเทพฯ 10300	22
2	B	179 ซ.สามเสน 19 ถนนนครไชยศรี แขวงถนนนครไชยศรี เขตดุสิต กรุงเทพฯ 10300	18
3	C	177 ซ.สามเสน 19 ถนนนครไชยศรี แขวงถนนนครไชยศรี เขตดุสิต กรุงเทพฯ 10300	31
4	D	177 ซ.สามเสน 19 ถนนนครไชยศรี แขวงถนนนครไชยศรี เขตดุสิต กรุงเทพฯ 10300	32

Figure 4.24 Building List Report Layout

ใบรายการใบเสร็จรับเงิน

พิมพ์ ณ วันที่ 03/เมษายน/2546

ลำดับที่	ใบเสร็จ	ห้องเลขที่	ประจำตัว	ค่าไฟฟ้า	ค่าประปา	ค่าโทรศัพท์	ค่าห้องพัก	ยอดรวม
1	0000005	C105	01/2545	244	105	139	2,200	2,688
2	0000006	C106	01/2545	140	60	50	2,200	2,450
3	0000007	C107	01/2545	156	120	83	2,200	2,559
4	0000008	C201	01/2545	204	90	50	2,200	2,544
5	0000009	C202	01/2545	0	0	100	2,200	2,300
6	0000010	C203	01/2545	112	120	99	2,200	2,531
								15,072

Figure 4.25 Receipt List Report Layout

ใบรายชื่อผู้จอง

พิมพ์ ณ วันที่ 03/มกราคม/2545

ลำดับที่	หมายเลขจอง	รหัสบัตรประชาชน	ชื่อผู้จอง	เลขที่ห้องจ	เงินมัดจำ	ภายในวันที่	เบอร์โทรศัพท์
1	0000001	11111111111111	111111111111111111	A105	1,000	26/08/2545	1111111111

Figure 4.26 Reserve List Report Layout

ใบรายชื่อสาธารณูปโภค

พิมพ์ ณ วันที่ 03/มกราคม/2545

ลำดับที่	ชื่อสาธารณูปโภค	ราคาต่อหน่วย
1	ไฟฟ้า	4
2	ประปา	15
3	โทรศัพท์	50

หมายเหตุ : \_\_\_\_\_

Figure 4.27 Utility List Report Layout

## ใบรายการห้องพัก

พิมพ์ ณ วันที่ 03/เมษายน/2545

ลำดับที่	ห้องเลขที่	สถานะห้องพัก	ชั้นที่	ราคาต่อเดือน	เงินมัดจำ	วันที่ย้ายเข้า	ชื่อเด็ก
1	B201	O	2	2,200	6,600	29/01/2545	B
2	B202	E	2	2,200	0		B
3	B203	O	2	2,200	6,600	29/01/2545	B
4	B204	O	2	2,200	6,600	29/01/2545	B
5	B205	O	2	2,200	6,600	29/01/2545	B
6	B206	E	2	2,200	0		B
7	B301	E	3	2,200	0		B
8	B302	E	3	2,200	0		B
9	B303	E	3	2,200	0		B
10	B304	E	3	2,200	0		B
11	B305	O	3	2,200	6,600	29/01/2545	B
12	B306	O	3	2,200	6,600	29/01/2545	B
13	B401	O	4	2,200	6,600	29/01/2545	B
14	B402	E	4	2,200	0		B
15	B403	E	4	2,200	0		B
16	B404	O	4	2,200	6,600	29/01/2545	B
17	B405	E	4	2,200	0		B
18	B406	E	4	2,200	0		B

Figure 4.28 Room List Report Layout

ใบรายชื่อเช่า

เงื่อนไขการพิมพ์

พิมพ์ ณ วันที่ 03 เมษายน 2545

ลำดับที่	ชื่อ - นามสกุล	สถานะผู้เช่า	ที่อยู่	เบอร์โทรศัพท์ติดต่อ	ห้องเลขที่
1	น.ส. ศุภลักษณ์ ภัทรวดี	1	265 หมู่ที่ 17 ต.หนองตะเภาใหญ่ อ.ด่านช้าง สุพรรณบุรี		8201
2	น.ส. วิไลลักษณ์ อารักษ์รัตน์	2	48/1 หมู่ที่ 2 ต.ปากน้ำ อ.เดิมบางนางบวช สุพรรณบุรี		8201
3	น.ส. ศายศุภชัย พงษ์หอม	1	66 หมู่ที่ 4 ต.โคกหม้อ อ.เมืองราชบุรี		8203
4	น.ส. ศายพิณ ปัญญาพล	1	13 หมู่ที่ 3 ต.หนองสระเตา อ.สามชุก สุพรรณบุรี		8204
5	น.ส. จำนงค์ คำมี	2	208 หมู่ที่ 4 ต.ด่านช้าง อ.ด่านช้าง สุพรรณบุรี		8204
6	น.ส. นิภาพร โสมโน้	1	628/1 หมู่ที่ 8 ต.เมืองเก่า อ.เมือง สุโขทัย		8205
7	น.ส. ธัญญาพร คุ้มศรีจำรัสชัย	1	103/2 หมู่ที่ 1 ต.นางบวช อ.เดิมบางนางบวช สุพรรณบุรี		8305
8	น.ส. ไพจิตร พงษ์พันธ์ุ	1	262 หมู่ที่ 2 ต.สามชุก อ.สามชุก สุพรรณบุรี		8306
9	น.ส. จักร์ สุธา	1	19/1 หมู่ที่ 12 ต.สระทราย อ.โคกสำโรง ลพบุรี		8401
10	นาย กมลศักดิ์ สมศรี	1	39 หมู่ที่ 12 ต.หนองใหญ่ อ.เมืองจันทบุรี จ.จันทบุรี		8404

หมายเหตุ : 1 = ผู้เช่า, 2 = ผู้ถืออาศัยคนที่ 1, 3 = ผู้ถืออาศัยคนที่ 2

1

Figure 4.29 Tenant List Report Layout

ห้องเลขที่ <b>B201</b>		เลขที่ใบเสร็จ 0000085			
		ประจำเดือน 01/2545			
	เลขเดือนก่อน	เลขเดือนนี้	หน่วยที่ใช้	จากค่าหน่วย	จำนวนเงิน
ห้องพัก	-	-	-	2,200	2,200
ประปา	1974	1982	8	15	120
ไฟฟ้า	6114	6148	34	4	136
ค่าบริการโทรศัพท์	-	-	-	-	50
ค่าโทรศัพท์	-	-	-	-	0
ยอดรวม					2,506

ห้องเลขที่ <b>B202</b>		เลขที่ใบเสร็จ 0000086			
		ประจำเดือน 01/2545			
	เลขเดือนก่อน	เลขเดือนนี้	หน่วยที่ใช้	จากค่าหน่วย	จำนวนเงิน
ห้องพัก	-	-	-	2,200	2,200
ประปา	549	553	4	15	60
ไฟฟ้า	3068	3096	28	4	112
ค่าบริการโทรศัพท์	-	-	-	-	50
ค่าโทรศัพท์	-	-	-	-	0
ยอดรวม					2,422

Figure 4.30 Receipt Report Layout

สรุปยอดประจำปี

3A/2545

เงินในการพิมพ์ ปีที่ ๒

ประจำปี : 2545

ห้องเลขที่	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน	พฤษภาคม	มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	ยอดรวม
A101	2,454	2,420	2,440	0	0	0	0	0	0	0	0	0	7,314
A102	0	2,270	2,269	0	0	0	0	0	0	0	0	0	4,539
A103	2,430	2,410	4,257	0	0	0	0	0	0	0	0	0	9,097
A104	2,408	2,423	2,456	0	0	0	0	0	0	0	0	0	7,287
A105	2,560	2,628	3,099	0	0	0	0	0	0	0	0	0	8,287
A106	2,319	2,432	2,484	0	0	0	0	0	0	0	0	0	7,235
A107	2,448	2,472	2,683	0	0	0	0	0	0	0	0	0	7,603
A108	2,498	2,636	2,624	0	0	0	0	0	0	0	0	0	7,758
A109	2,376	2,376	2,620	0	0	0	0	0	0	0	0	0	7,372
A110	2,805	2,673	2,785	0	0	0	0	0	0	0	0	0	8,263
A111	2,833	2,367	2,466	0	0	0	0	0	0	0	0	0	7,666
A112	2,628	2,536	2,652	0	0	0	0	0	0	0	0	0	7,816
A113	2,406	3,179	2,470	0	0	0	0	0	0	0	0	0	8,055
A114	2,403	2,453	2,792	0	0	0	0	0	0	0	0	0	7,648
A115	2,541	2,631	2,746	0	0	0	0	0	0	0	0	0	7,918
A116	2,388	2,335	2,365	0	0	0	0	0	0	0	0	0	7,088
A117	2,341	2,250	2,250	0	0	0	0	0	0	0	0	0	6,841

Figure 4.31 Annually Summary Report Layout

สรุปการใช้ไฟฟ้าประจำปี

3A/2545

เงินในการพิมพ์ ปีที่ ๒

ประจำปี : 2545

ห้องเลขที่	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน	พฤษภาคม	มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	ยอดรวม
B201	2,506	2,537	2,823	0	0	0	0	0	0	0	0	0	7,866
B202	2,422	2,634	2,465	0	0	0	0	0	0	0	0	0	7,521
B203	2,376	2,479	2,347	0	0	0	0	0	0	0	0	0	7,202
B204	2,504	2,662	2,671	0	0	0	0	0	0	0	0	0	7,837
B205	2,449	2,434	2,453	0	0	0	0	0	0	0	0	0	7,336
B206	2,550	2,675	2,690	0	0	0	0	0	0	0	0	0	7,915
B301	3,076	2,738	2,868	0	0	0	0	0	0	0	0	0	8,482
B302	2,855	2,898	3,907	0	0	0	0	0	0	0	0	0	9,660
B303	2,800	2,678	2,721	0	0	0	0	0	0	0	0	0	8,199
B304	2,555	2,481	2,515	0	0	0	0	0	0	0	0	0	7,551
B305	2,348	2,336	2,463	0	0	0	0	0	0	0	0	0	7,147
B306	2,376	2,486	2,357	0	0	0	0	0	0	0	0	0	7,219
B401	2,572	2,628	2,751	0	0	0	0	0	0	0	0	0	7,951
B402	2,655	2,629	2,676	0	0	0	0	0	0	0	0	0	7,960
B403	2,328	2,503	2,422	0	0	0	0	0	0	0	0	0	7,253
B404	2,422	2,414	2,417	0	0	0	0	0	0	0	0	0	7,253
B405	2,648	2,650	2,824	0	0	0	0	0	0	0	0	0	8,122

Figure 4.32 Annually Electric Used Report Layout

# สรุปการใช้โทรศัพท์ประจำปี

3/4/2545

เงื่อนไขการพิมพ์ **ตึก B**

ประจำปี : 2545

ห้องเลขที่	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน	พฤษภาคม	มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	ยอดรวม
B201	120	135	150	0	0	0	0	0	0	0	0	0	405
B202	60	60	60	0	0	0	0	0	0	0	0	0	180
B203	30	120	45	0	0	0	0	0	0	0	0	0	195
B204	90	120	105	0	0	0	0	0	0	0	0	0	315
B205	75	60	75	0	0	0	0	0	0	0	0	0	210
B206	60	45	45	0	0	0	0	0	0	0	0	0	150
B301	150	75	90	0	0	0	0	0	0	0	0	0	315
B302	165	120	120	0	0	0	0	0	0	0	0	0	405
B303	150	75	75	0	0	0	0	0	0	0	0	0	300
B304	45	30	30	0	0	0	0	0	0	0	0	0	105
B305	30	30	30	0	0	0	0	0	0	0	0	0	90
B306	30	30	30	0	0	0	0	0	0	0	0	0	90
B401	90	90	105	0	0	0	0	0	0	0	0	0	285
B402	105	75	90	0	0	0	0	0	0	0	0	0	270
B403	30	30	15	0	0	0	0	0	0	0	0	0	75
B404	60	60	60	0	0	0	0	0	0	0	0	0	180
B405	150	30	150	0	0	0	0	0	0	0	0	0	330

Figure 4.33 Annually Telephone Used Report Layout

# สรุปการใช้ประปาประจำปี

3/4/2545

เงื่อนไขการพิมพ์ **ตึก B**

ประจำปี : 2545

ห้องเลขที่	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน	พฤษภาคม	มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	ยอดรวม
B201	136	152	96	0	0	0	0	0	0	0	0	0	384
B202	112	140	116	0	0	0	0	0	0	0	0	0	368
B203	96	104	62	0	0	0	0	0	0	0	0	0	262
B204	164	236	288	0	0	0	0	0	0	0	0	0	688
B205	124	124	128	0	0	0	0	0	0	0	0	0	376
B206	240	212	240	0	0	0	0	0	0	0	0	0	692
B301	676	288	280	0	0	0	0	0	0	0	0	0	1,244
B302	440	528	384	0	0	0	0	0	0	0	0	0	1,352
B303	400	348	396	0	0	0	0	0	0	0	0	0	1,144
B304	260	196	220	0	0	0	0	0	0	0	0	0	676
B305	68	56	80	0	0	0	0	0	0	0	0	0	204
B306	96	160	72	0	0	0	0	0	0	0	0	0	328
B401	232	288	396	0	0	0	0	0	0	0	0	0	916
B402	300	304	320	0	0	0	0	0	0	0	0	0	924
B403	48	44	76	0	0	0	0	0	0	0	0	0	168
B404	112	104	92	0	0	0	0	0	0	0	0	0	308
B405	248	272	308	0	0	0	0	0	0	0	0	0	828

Figure 4.34 Annually Water Used Report Layout

กราฟแสดงยอดเงินในแต่ละเดือน

เงื่อนไขการพิมพ์: พิมพ์ตามตึก B

ประจำปี : 2545

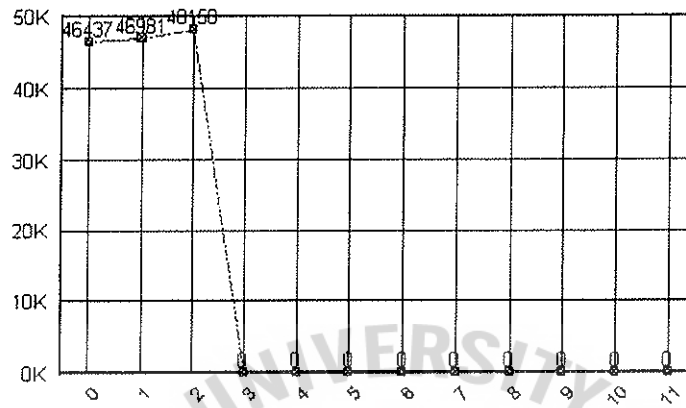


Figure 4.35 Annually Summary Graph Report Layout

3/4/2545

กราฟแสดงจำนวนหน่วยไฟฟ้าที่ใช้

เงื่อนไขการพิมพ์: พิมพ์ตามตึก B

ประจำปี : 2545

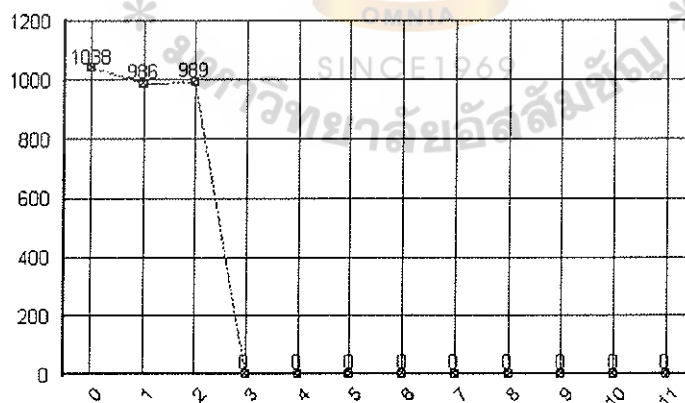


Figure 4.36 Annually Electric Unit Graph Report Layout

กราฟแสดงจำนวนหน่วยโทรศัพท์ที่ใช้

เงื่อนไขกราฟ: พิมพ์ตามตึก B

ประจำปี : 2545

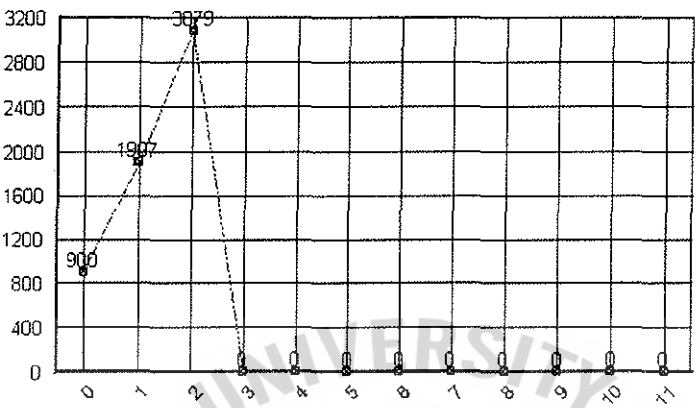


Figure 4.37 Annually Telephone Graph Report Layout

กราฟแสดงจำนวนหน่วยประปาที่ใช้

เงื่อนไขกราฟ: พิมพ์ตามตึก B

ประจำปี : 2545

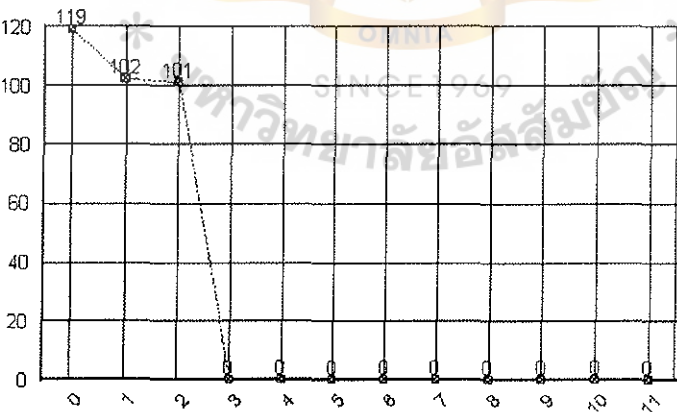


Figure 4.38 Annually Water Graph Report Layout

กราฟแสดงจำนวนห้องเช่าในแต่ละเดือน

เงื่อนไขกราฟ: พิมพ์ตามตัว B

ประจำปี : 2545

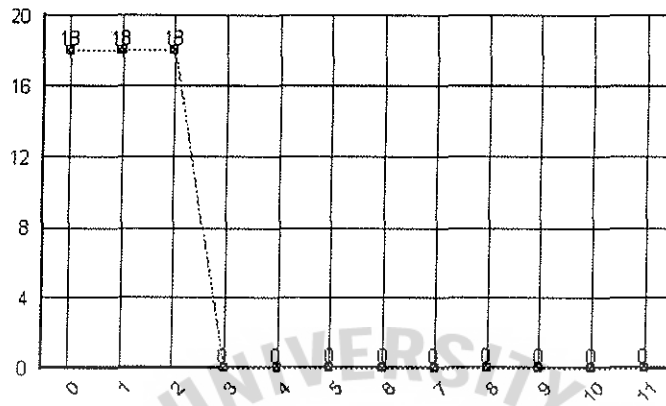


Figure 4.39 Room Rent Graph Report Layout



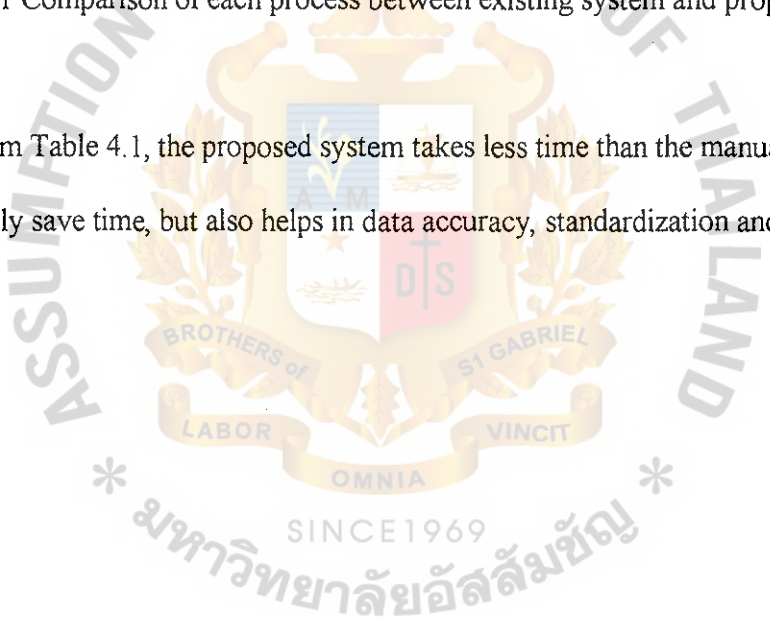
4.3 Project Evaluation

After the purpose system is implemented to SS Apartment, it is tested by the users. Table 4.1 shows the time spent on each process of the proposed system compared with the existing system This way we can explain that the proposed system is more efficient and effective than the existing system.

Process	Existing System	Proposed System
Record data process	15 minutes.	5 minutes.
Retrieve data process	10 minutes.	2 minutes.
Monthly receipt process	10 minutes.	2 minutes.
Update data process	10 minutes.	3 minutes.
Generate report process	1 hour	3 minutes.
Total	1 hour 45 minutes	15 minutes.

Table 4.1 Comparison of each process between existing system and proposed system

From Table 4.1, the proposed system takes less time than the manual system. It does not only save time, but also helps in data accuracy, standardization and integrity.



## 5. CONCLUSIONS AND RECOMMENDATIONS

### 5.1 Conclusions

The SS Apartment system is implemented to store, manage, and analyze the data of SS Apartment. The required functions include storing tenant, room, reserver, building and utility records and reporting. The development of computer information system is costly. It takes much time and many factors in order to accomplish the objective. Therefore, user and apartment owner have to participate in almost every phase including analysis, design and implementing the system.

For the existing system, it is a manual system. There is data redundancy and it is time consuming to do the routine task such as finding the tenant information, and generating the receipt. Moreover, it is hard to do further development.

For the purposed system, the system can provide more efficient service to tenant and help the apartment owner to do the routine task conveniently and quickly. Moreover, it can provide accurate, timely and up-to-date information in form of report.

### 5.2 Recommendations

There are three important success factors to the implementation of SS Apartment System:

1. Developer should assist the users while the system is in the implementation stage.
2. The company should not allow the tenant to access the information individually.
3. The apartment owner has to understand the system.

In the future, this system can be expanded and furtherly developed to fit the users requirements. The further development for this system would be:

1. To allow the tenant and customer to access the information.
2. To develop a mailing system. The tenant can check the mail from the apartment system instead of finding it in the mail box, which sometimes can end up with a loss of mail.
3. To improve the system security. When the apartment system allows the tenant to access the system, the system has to improve the system security to provide the right to the users.
4. To improve the database system in order to support high volume of data.



## BIBLIOGRAPHY

- [1] Jeffrey, A. Hoffer, Joey F. George and Joseph S. Valacich. Modern System Analysis and Design. The Benjamin/Cummings Publishing Company, Inc. 1995.
- [2] Perry, Edwards. System Analysis & Design. McGraw-Hill International Editions. 1993.
- [3] Gary B. Shelly, Thomas J. Cashman, Judy Adamski, and Joseph J. Adamski. Systems Analysis and Design, 2th Edition. Boyd & Fuaser Publishing Company. 1995
- [4] พ.อ. เจนวิทย์ เหลืองอร่าม, และ ปิยวิทย์ เหลืองอร่าม. การเขียนโปรแกรมสำหรับ Applications ด้วย Visual Basic 6. บริษัท ธรรมสาร จำกัด., 2543
- [5] กิตติ ภักดีวัฒนะกุล, และ จำลอง คุรุอดสาหะ. Visual Basic6 ฉบับงานข้อมูล. พิมพ์ครั้งที่ 2. หจก. ไททจริญการพิมพ์., 2543



# **APPENDIX A**

## **DATAFLOW DIAGRAM**

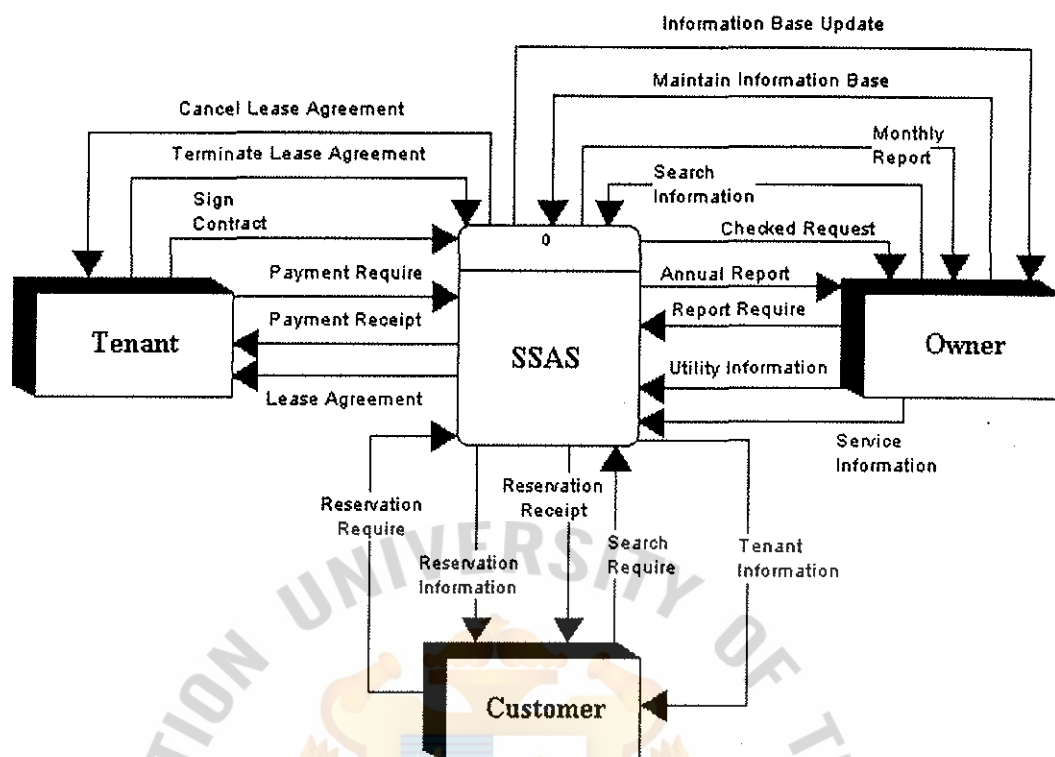


Figure A.1 Context Diagram

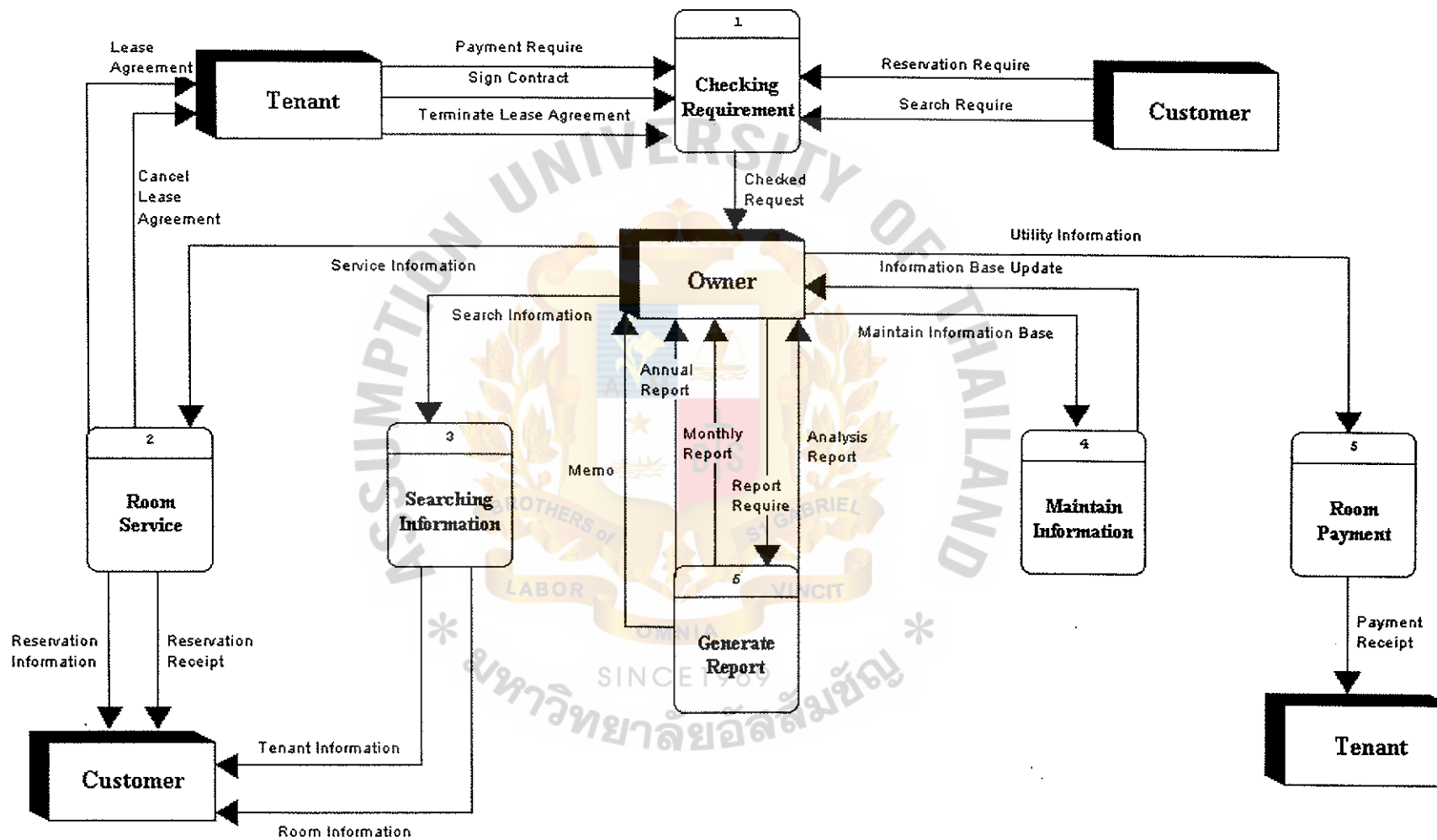


Figure A.2 Data Flow Diagram Level 0 of SS Apartment System

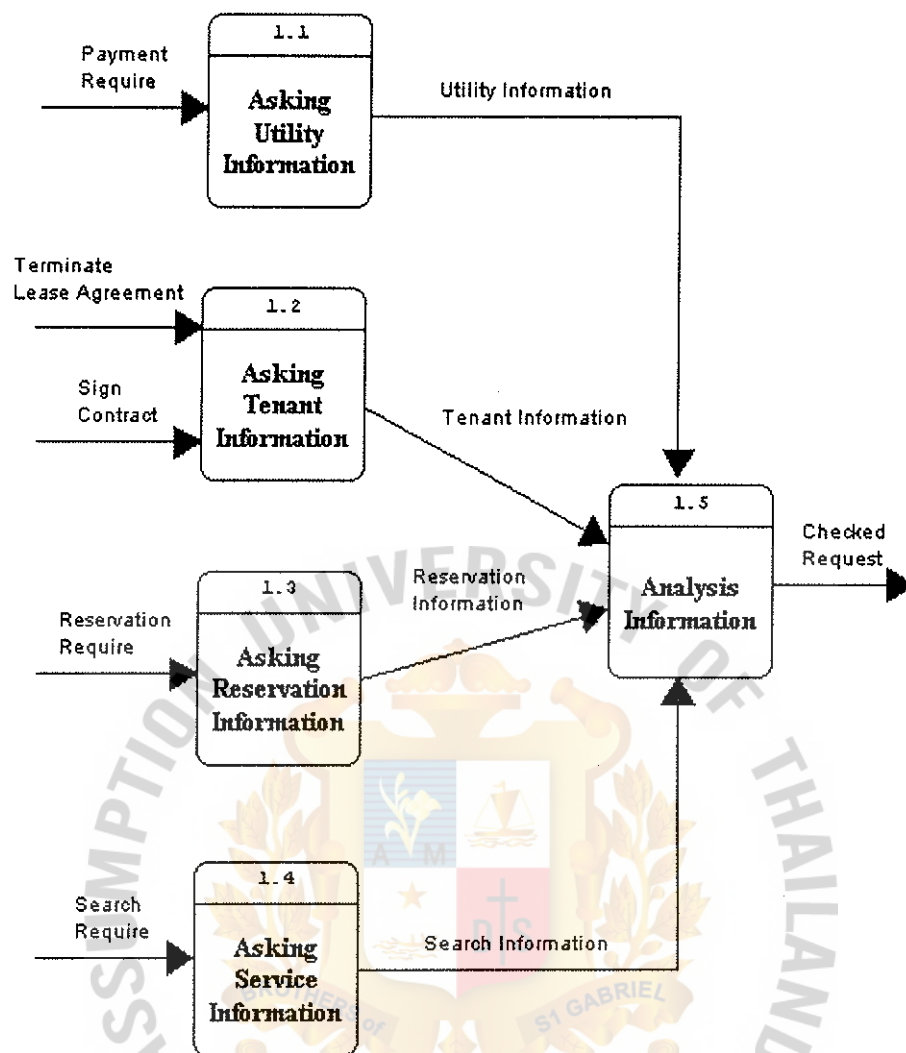


Figure A.3 Data Flow Diagram Level 1 of Checking Requirement

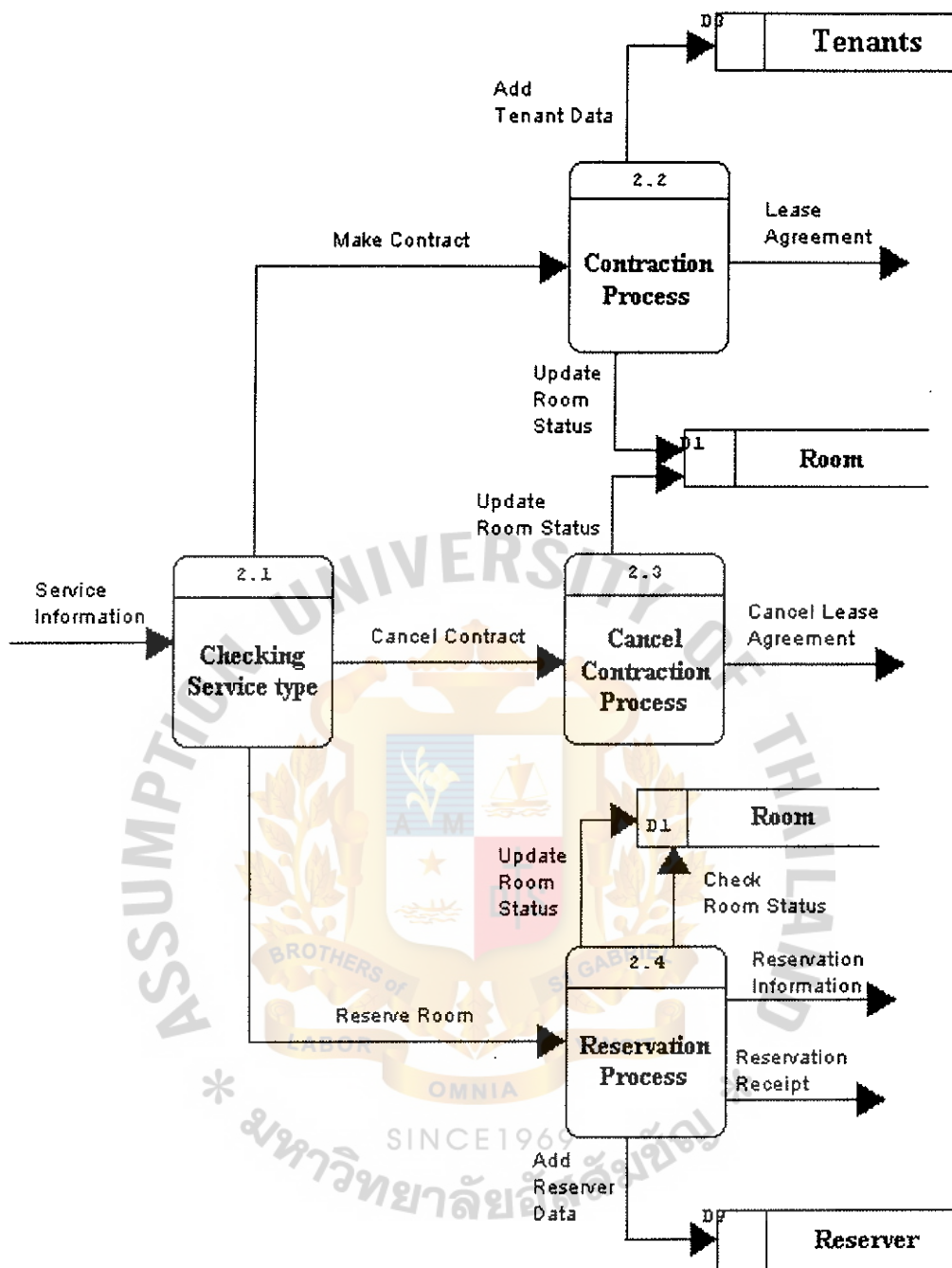


Figure A.4 Data Flow Diagram Level 1 of Room Service

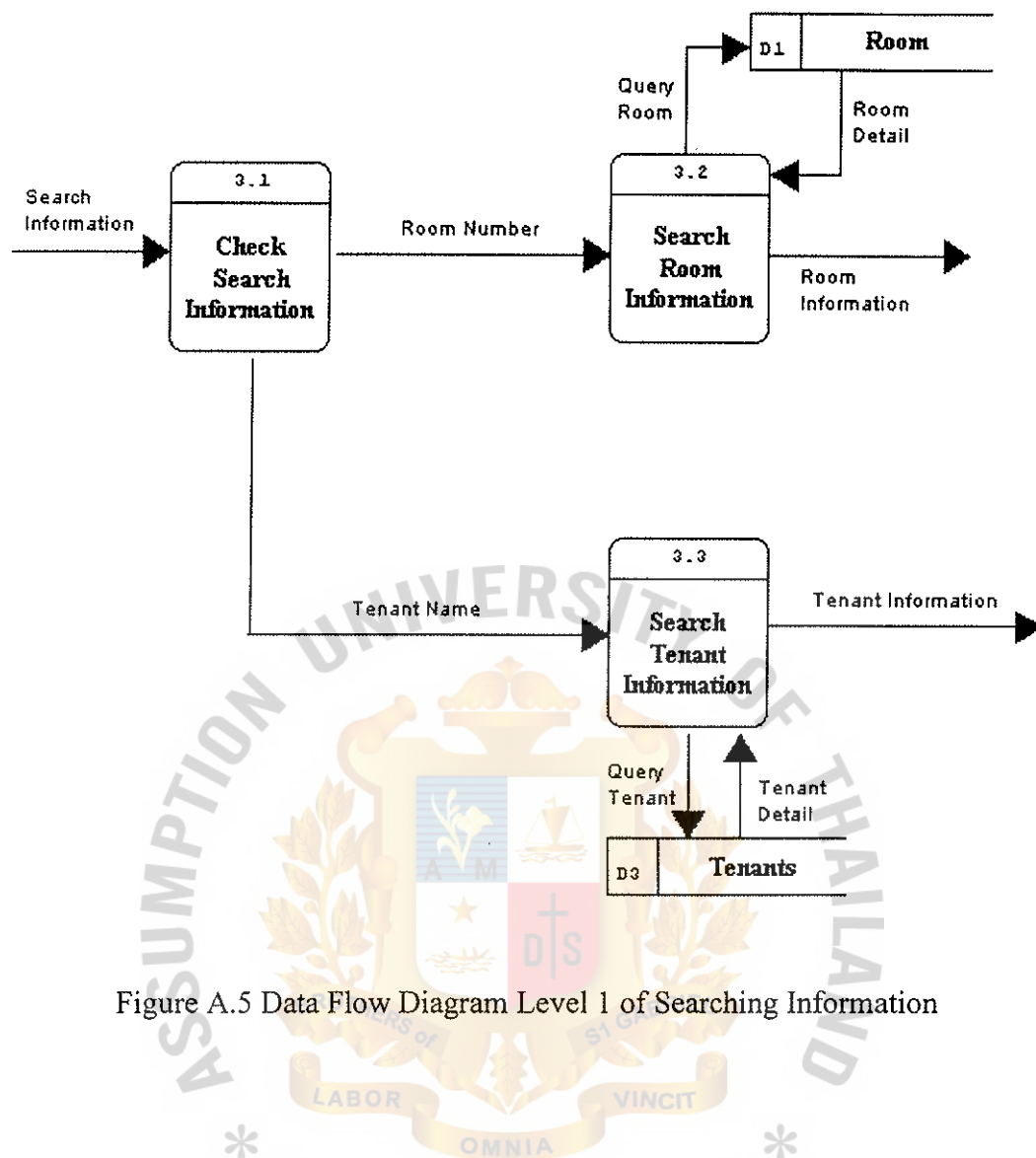


Figure A.5 Data Flow Diagram Level 1 of Searching Information

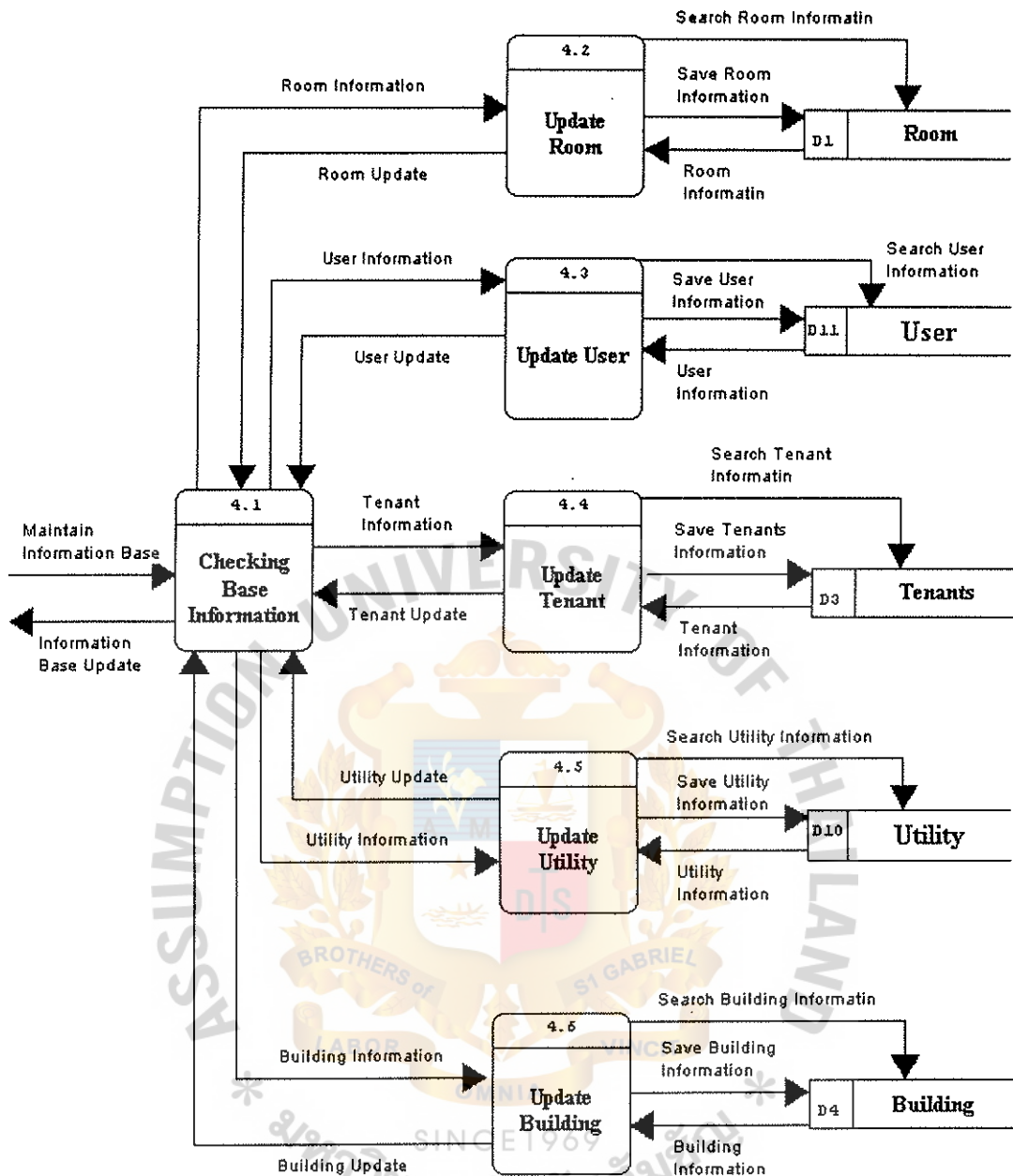


Figure A.6 Data Flow Diagram Level 1 of Maintain Information

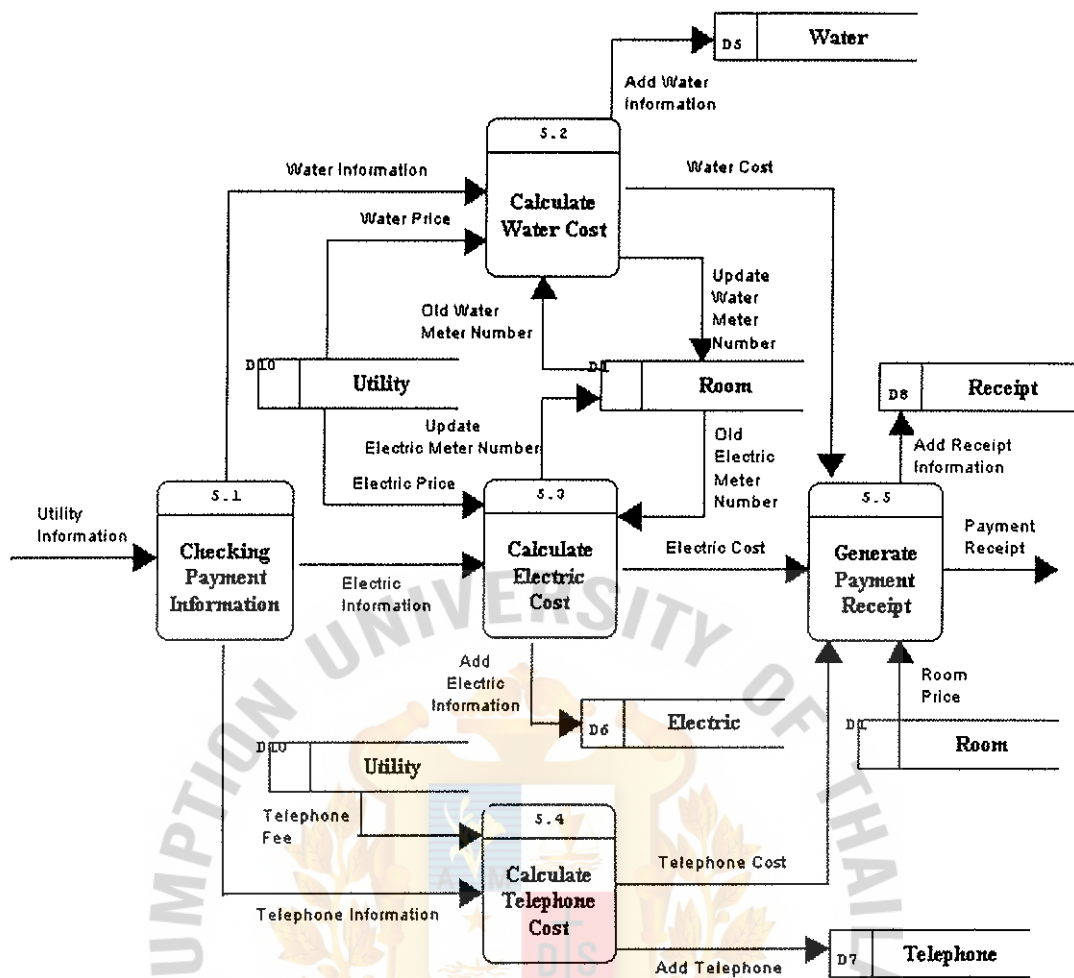


Figure A.7 Data Flow Diagram Level 1 of Room Payment

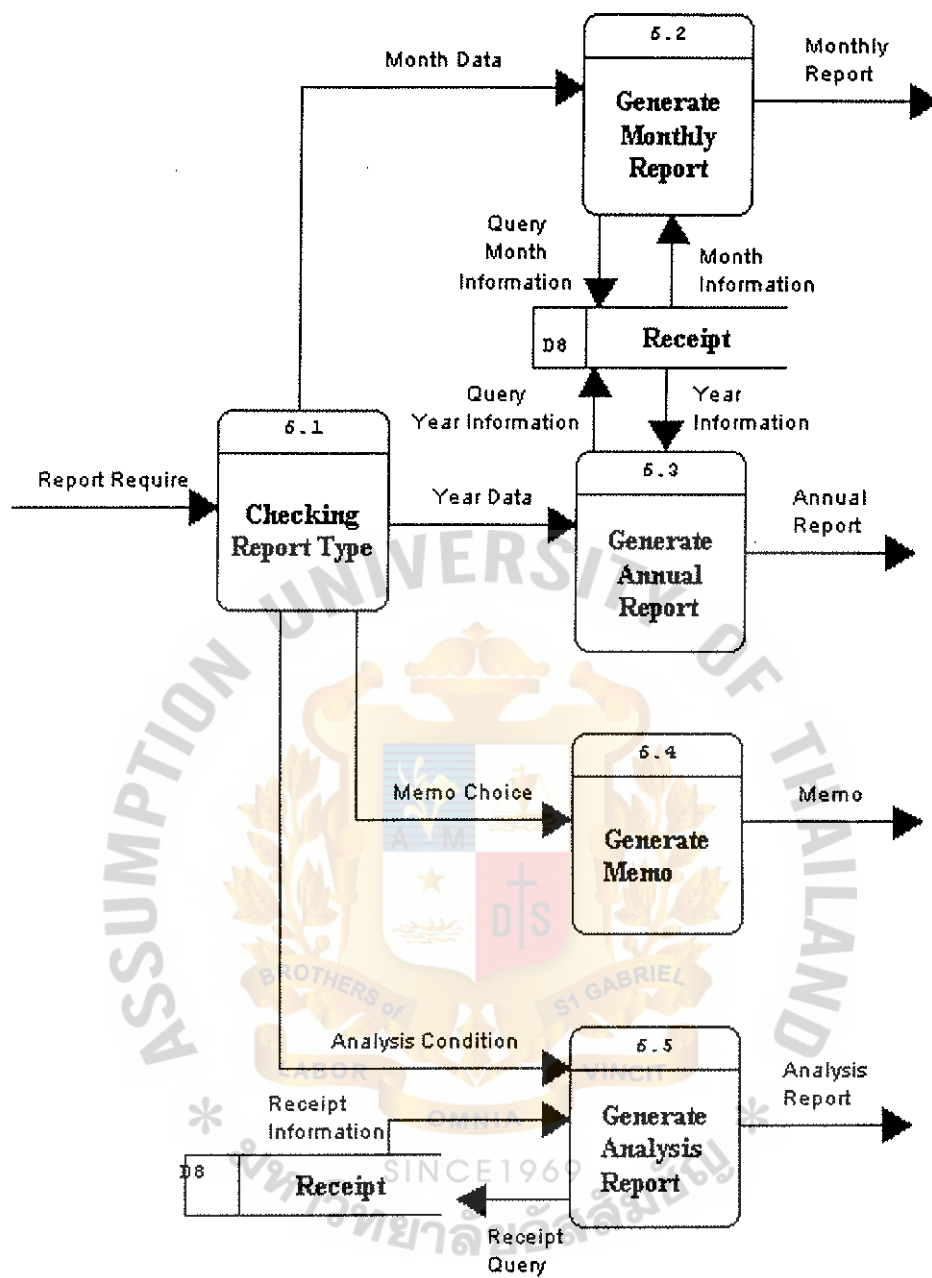


Figure A.8 Data Flow Diagram Level 1 of Generate Report

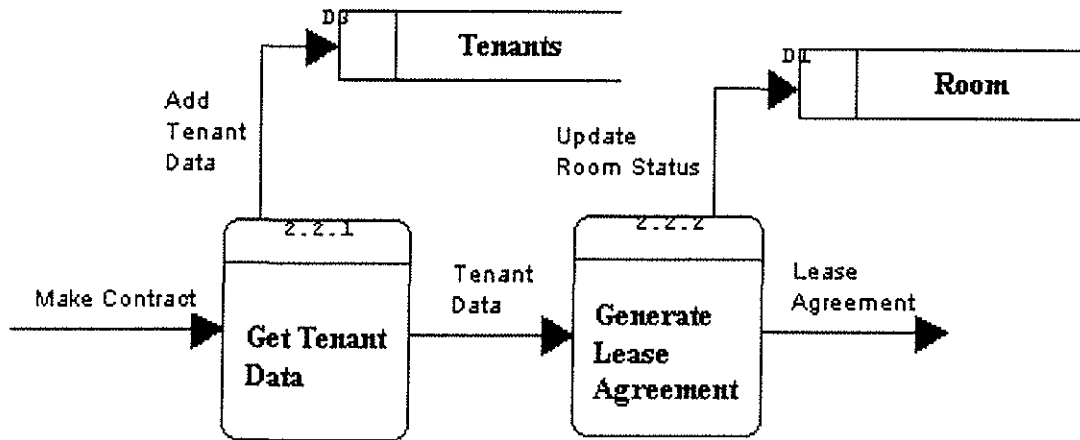


Figure A.9 Data Flow Diagram Level 2 of Contraction Process

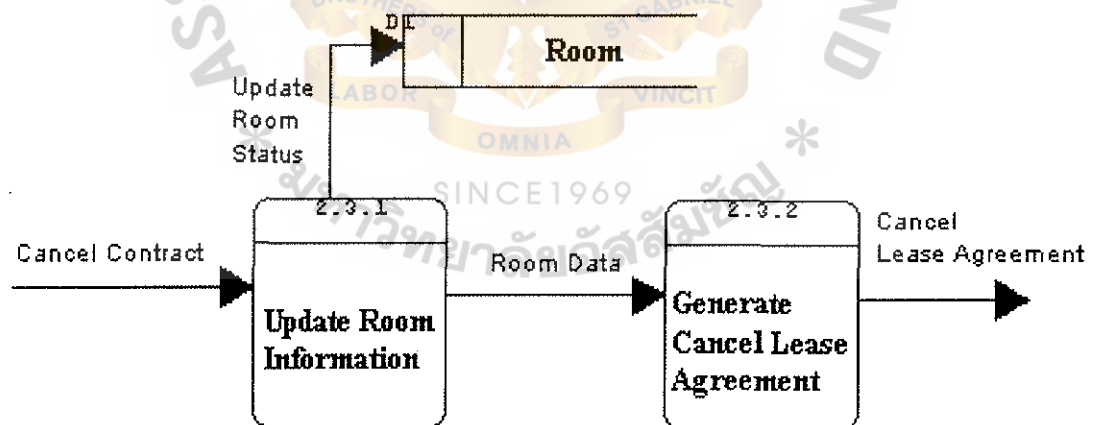


Figure A.10 Data Flow Diagram Level 2 of Cancel Contraction Process

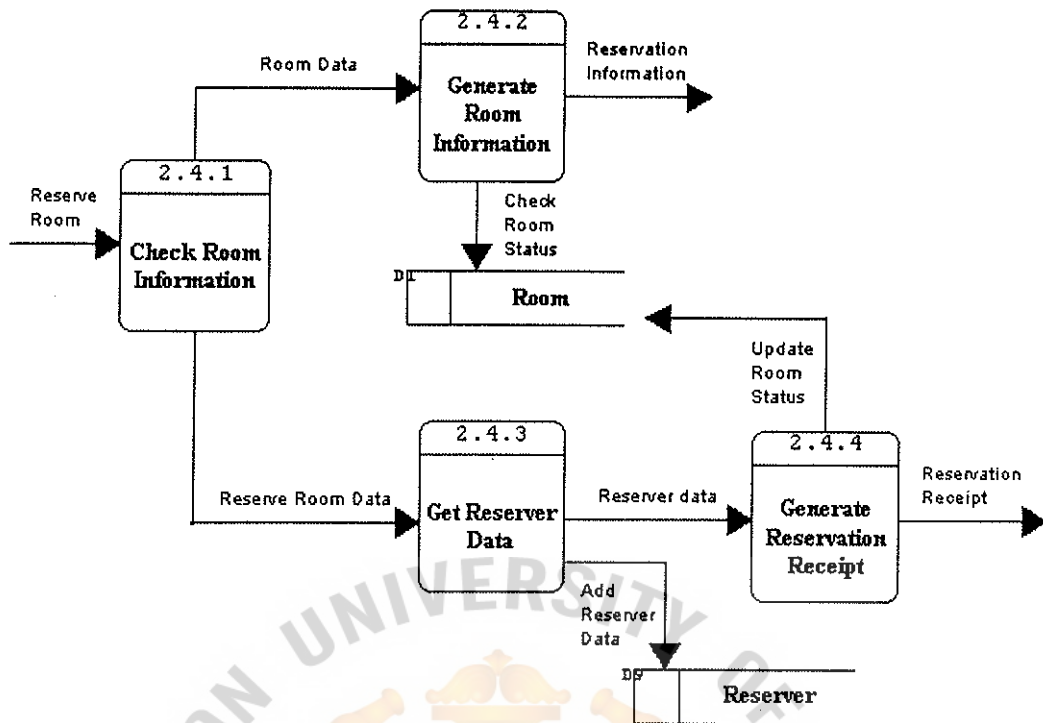


Figure A.11 Data Flow Diagram Level 2 of Reservation Process

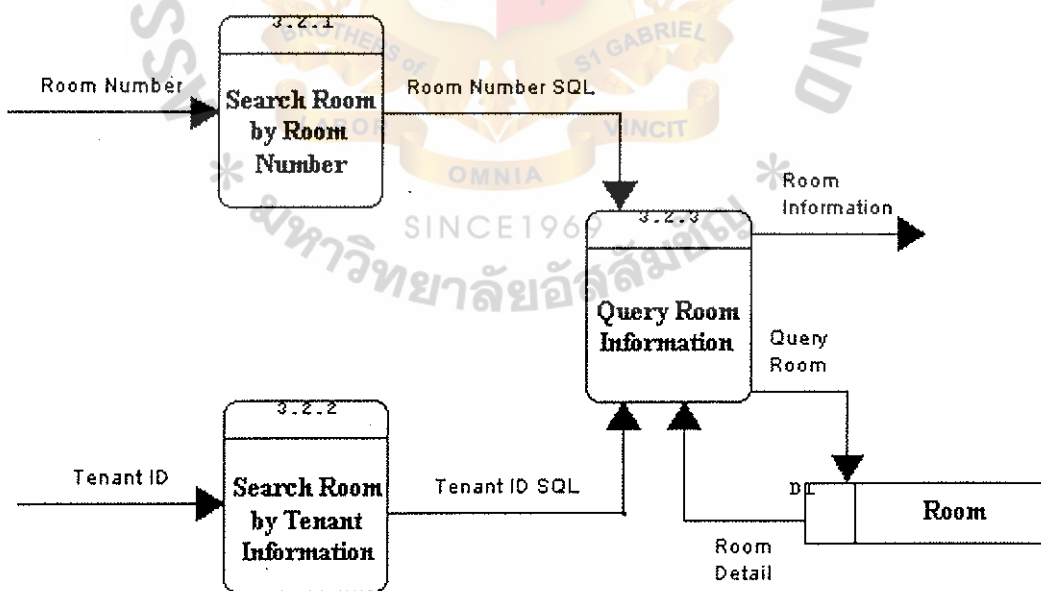


Figure A.12 Data Flow Diagram Level 2 of Search Room Information

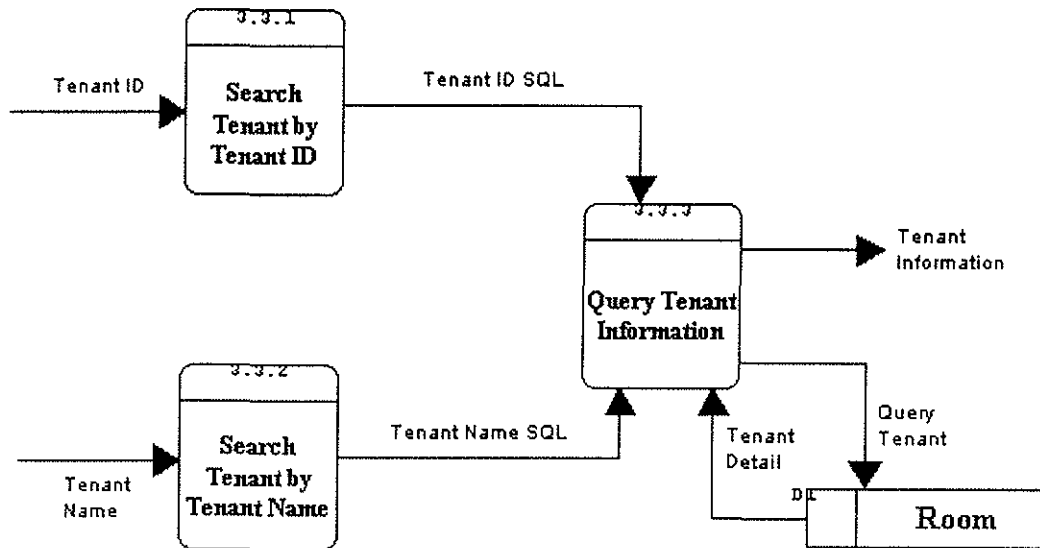


Figure A.13 Data Flow Diagram Level 2 of Search Tenant Information

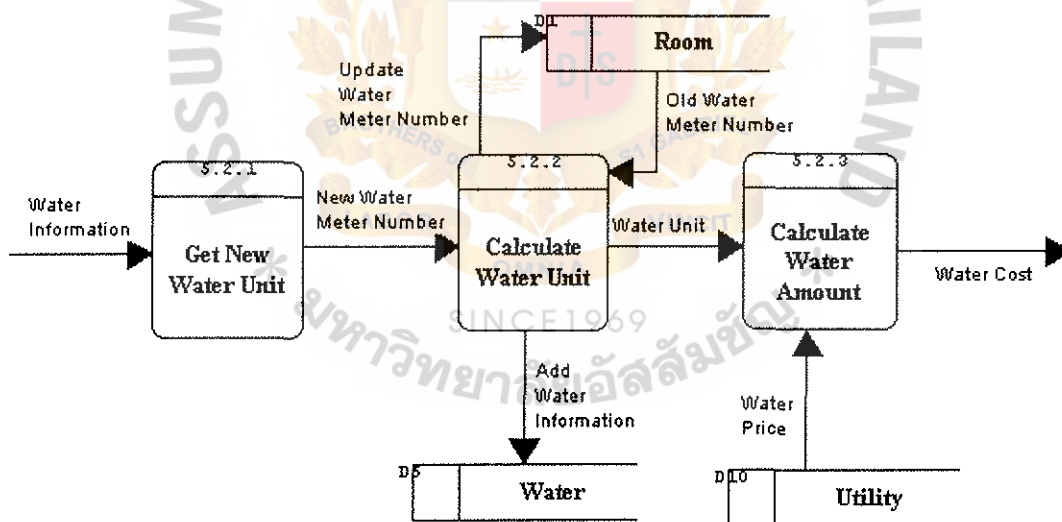


Figure A.14 Data Flow Diagram Level 2 of Calculate Water Cost

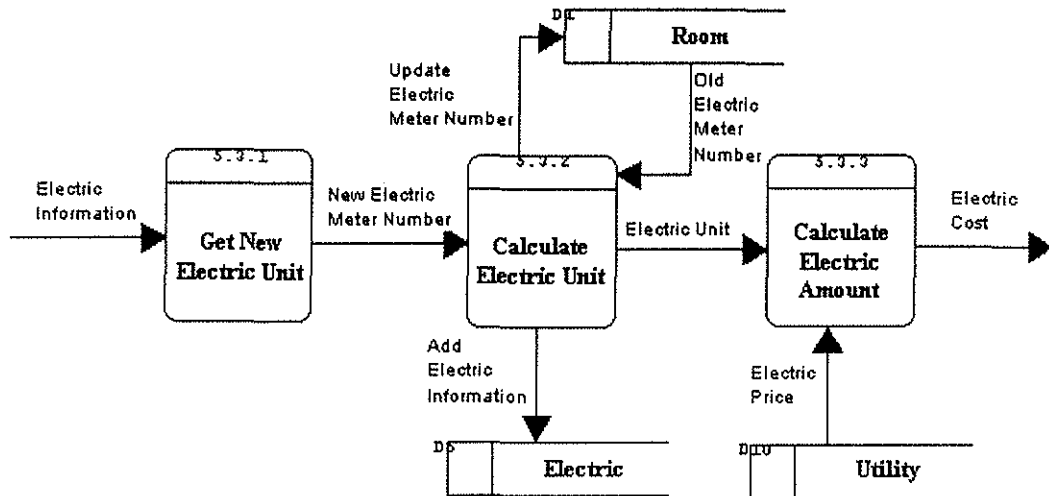


Figure A.15 Data Flow Diagram Level 2 of Calculate Electric Cost

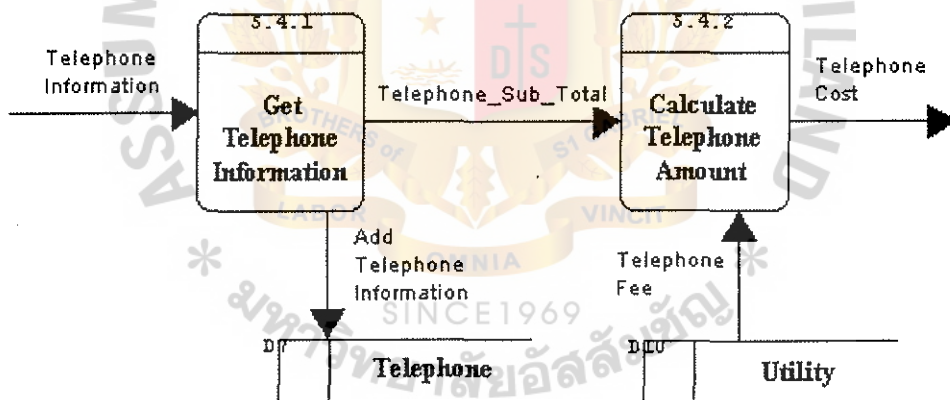


Figure A.16 Data Flow Diagram Level 2 of Calculate Telephone Cost

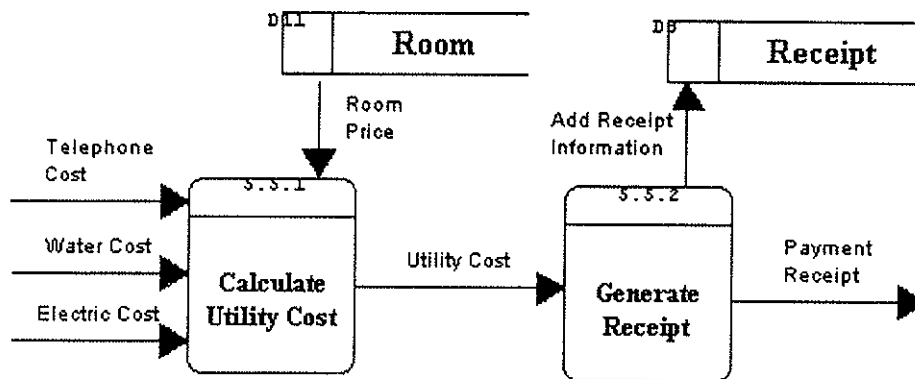


Figure A.17 Data Flow Diagram Level 2 of Generate Payment Receipt

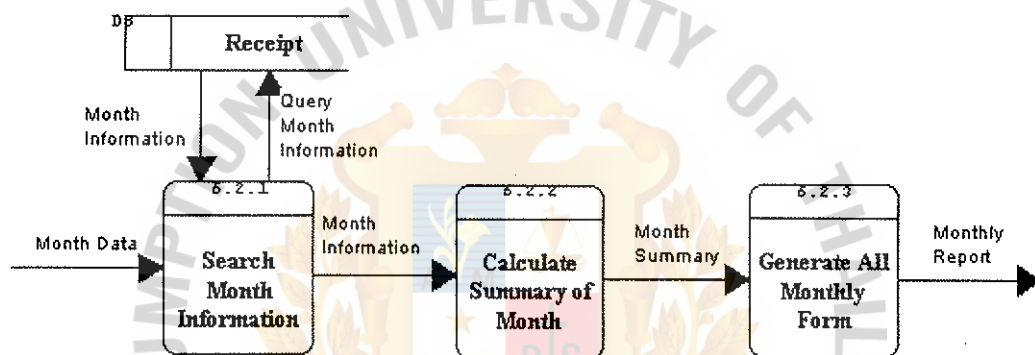


Figure A.18 Data Flow Diagram Level 2 of Generate Monthly Report

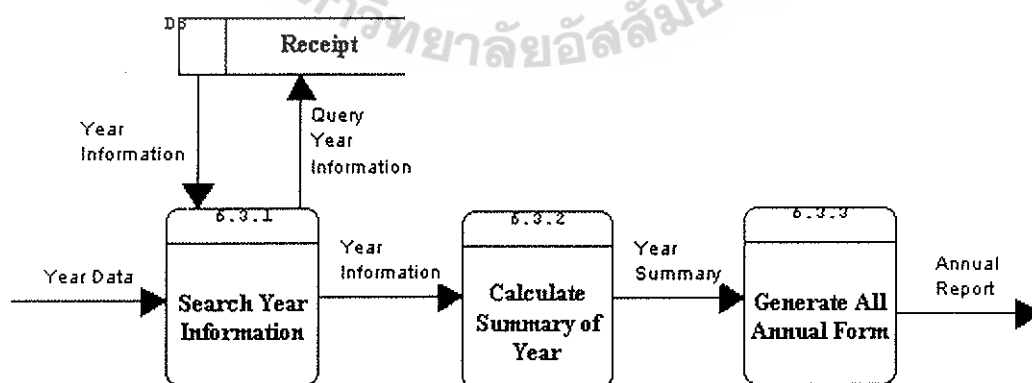


Figure A.19 Data Flow Diagram Level 2 of Generate Annual Report



**APPENDIX B**  
**PROCESS DESCRIPTION**

## Analysis Information

## Process

### Description:

The process analyzes the information and chooses the proper menu to handle the information.

**Process #:** 1.5

### Location:

Checking Requirement (1)

Input Flows: Tenant Information

Reservation Information

Search Information

Utility Information

Output Flows: Checked Request

---

## Asking Reservation Information

## Process

### Description:

The process receives reserver information such as reserver's name, and sends information to the apartment owner to generate the Reservation Receipt.

**Process #:** 1.3

### Location:

Checking Requirement (1)

Input Flows: Reservation Require

Output Flows: Reservation Information

---

## Asking Service Information

## Process

### Description:

The process receives service information such as room number, tenant name, and sends information to the apartment owner to provide the service.

Process #: 1.4

### Location:

Checking Requirement (1)

Input Flows: Search Require

Output Flows: Search Information

---

## Asking Tenant Information

## Process

### Description:

The process receives tenant information such as tenant's name, tenant's ID, and sends information to the apartment owner to generate the Lease Agreement.

Process #: 1.2

### Location:

Checking Requirement (1)

Input Flows: Sign Contract

Terminate Lease Agreement

Output Flows: Tenant Information

---

## Asking Utility Information

## Process

### Description:

The process receives utility information such as water meter number, and

sends information to the apartment owner to generate the receipt.

**Process #:** 1.1

**Location:**

Checking Requirement ( 1 )

Input Flows: Payment Require

Output Flows: Utility Information

---

### Calculate Electric Amount

**Process**

**Process #:** 5.3.3

**Location:**

5.3.x Calculate Electric Cost ( 5.3 )

Input Flows: Electric Price

Electric Unit

Output Flows: Electric Cost

---

### Calculate Electric Cost

**Process**

**Description:**

This process receives new electric meter number to calculate the electric cost.

**Process #:** 5.3

**Location:**

Room Payment ( 5 )

Input Flows: Electric Information

Electric Price

Old Electric Meter Number

Output Flows:            Electric Cost  
                                 Add Electric Information  
                                 Update Electric Meter Number

---

**Calculate Electric Unit**

**Process**

**Process #:**     5.3.2

**Location:**

5.3.x Calculate Electric Cost ( 5.3 )

Input Flows:            Old Electric Meter Number  
                                 New Electric Meter Number  
Output Flows:            Add Electric Information  
                                 Update Electric Meter Number  
                                 Electric Unit

---

**Calculate Summary of Month**

**Process**

**Process #:**     6.2.2

**Location:**

6.2.x Generate Monthly Report ( 6.2 )

Input Flows:            Month Information  
Output Flows:            Summary

---

**Calculate Summary of Year**

**Process**

**Process #:**     6.3.2

**Location:**

6.3.x Generate Annual Report ( 6.3 )

Input Flows: Month Information  
Output Flows: Year Summary

---

### Calculate Telephone Amount

### Process

Process #: 5.4.2

Location:

5.4.x Calculate Telephone Cost ( 5.4 )

Input Flows: Telephone Fee  
Telephone\_Sub\_Total  
Output Flows: Telephone Cost

---

### Calculate Telephone Cost

### Process

Description:

This process receives telephone charge to calculate with telephone fee.

Process #: 5.4

Location:

Room Payment ( 5 )

Input Flows: Telephone Information  
Telephone Fee

Output Flows: Telephone Cost  
Add Telephone Information

---

### Calculate Utility Cost

### Process

Process #: 5.5.1

Location:

### 5.5.x Generate Payment Receipt ( 5.5 )

Input Flows:	Water Cost
	Telephone Cost
	Electric Cost
	Room Price
Output Flows:	Utility Cost

---

#### Calculate Water Amount

Process

Process #: 5.2.3

Location:

5.2.x Calculate Water Cost ( 5.2 )

Input Flows:	Water Price
	Water Unit
Output Flows:	Water Cost

---

#### Calculate Water Cost

Process

Description:

This process receives new water meter number to calculate the water cost.

Process #: 5.2

Location:

Room Payment ( 5 )

Input Flows:	Water Information
	Old Water Meter Number
	Water Price
Output Flows:	Water Cost

Add Water Information

Update Water Meter Number

---

**Calculate Water Unit**

**Process**

**Process #:** 5.2.2

**Location:**

5.2.x Calculate Water Cost ( 5.2 )

Input Flows: Old Water Meter Number

New Water Meter Number

Output Flows: Add Water Information

Update Water Meter Number

Water Unit

---

**Cancel Contraction Process**

**Process**

**Description:**

This process receives the tenant information to inform movement, cancel lease agreement.

**Process #:** 2.3

**Location:**

Room Service ( 2 )

Input Flows: Cancel Contract

Output Flows: Update Room Status

Cancel Lease Agreement

---

**Check Room Information**

**Process**

**Process #:** 2.4.1

**Location:**

2.4.x Reservation Process ( 2.4 )

Input Flows: Reserve Room

Output Flows: Room Data

Reserve Room Data

---

### Check Search Information

**Process**

**Description:**

The process checks type of search information such as room, and tenant, and gets the information from User.

**Process #:** 3.1

**Location:**

Searching Information ( 3 )

Input Flows: Search Information

Output Flows: Tenant Name

Room Number

---

### Checking Base Information

**Process**

**Description:**

This process checks the user's requirement to update which type of information such as room, user, tenant, building, and utility data.

**Process #:** 4.1

**Location:**

Maintain Base Information ( 4 )

St. Gabriel's Library, Au

Input Flows:	Maintain Information Base
	Room Update
	Tenant Update
	Building Update
	User Update
	Utility Update

Output Flows:	Information Base Update
	Tenant Information
	Building Information
	Room Information
	User Information
	Utility Information

### Checking Payment Information

## Process

**Description:**

This process checks the utility information from tenant and calculate utility cost.

**Process #:** 5.1

**Location:**

Room Payment ( 5 )

Input Flows:	Utility Information
Output Flows:	Electric Information
	Water Information
	Telephone Information

## Checking Report Type

## Process

### Description:

The process gets the information and sends them to generate the report that user requires.

Process #: 6.1

### Location:

Generate Report ( 6 )

Input Flows: Report Require

Output Flows: Month Data

Year Data

Memo Choice

Analysis Condition

## Checking Requirement

## Process

### Description:

The Checking Requirement process is to inquire and keep the requirement from Customer and Tenant to use in the SSA System.

Process #: 1

### Location:

Level0 ( 0 )

Input Flows: Payment Require

Terminate Lease Agreement

Reservation Require

Search Require

Sign Contract

Output Flows:            Checked Request

---

**Checking Service type**

**Process**

**Description:**

The process checks the service requirement and collects the service information.

**Process #:**     2.1

**Location:**

Room Service        ( 2 )

Input Flows:            Service Information

Output Flows:            Make Contract

Cancel Contract

Reserve Room

---

**Contraction Process**

**Process**

**Description:**

This process receives the tenant information such as tenant name, address and so on and generates the lease agreement.

**Process #:**     2.2

**Location:**

Room Service        ( 2 )

Input Flows:            Make Contract

Output Flows:            Update Room Status

Lease Agreement

Add Tenant Data

---

**Generate All Annual Form****Process****Process #:** 6.3.3**Location:**

6.3.x Generate Annual Report ( 6.3 )

Input Flows: Year Summary

Output Flows: Annual Report

---

**Generate All Monthly Form****Process****Process #:** 6.2.3**Location:**

6.2.x Generate Monthly Report ( 6.2 )

Input Flows: A M Summary

Output Flows: Monthly Report

---

**Generate Analysis Report****Process****Description:**

The process gets analysis condition to generate SQL command and prints out the analysis report.

**Process #:** 6.5**Location:**

Generate Report ( 6 )

Input Flows: Analysis Condition

Receipt Information

Output Flows: Receipt Query

Generate Annual Report

Process

Description:

The process get annual data to generate SQL command and print out the annual report.

Process #: 6.3

Location:

Generate Report ( 6 )

Input Flows:

Year Data

Year Information

Output Flows:

Annual Report

Query Year Information

Generate Cancel Lease Agreement

Process

Process #: 2.3.2

Location:

2.3.x Cancel Contraction Process ( 2.3 )

Input Flows:

Room Data

Output Flows:

Cancel Lease Agreement

Generate Lease Agreement

Process

Process #: 2.2.2

Location:

2.2.x Contraction Process ( 2.2 )

Input Flows:            Tenant Data

Output Flows:        Update Room Status

                         Lease Agreement

---

## Generate Memo

## Process

### Description:

This process generates the memo that user requires.

Process #:        6.4

### Location:

Generate Report ( 6 )

Input Flows:        Memo Choice

Output Flows:      Memo

---

## Generate Monthly Report

## Process

### Description:

The process gets month data to generate SQL command and prints out the monthly report.

Process #:        6.2

### Location:

Generate Report ( 6 )

Input Flows:        Month Data

Month Information

Output Flows:      Monthly Report

Query Month Information

## Generate Payment Receipt

## Process

### Description:

The process receives the utility cost and generates the receipt.

Process #: 5.5

### Location:

Room Payment ( 5 )

Input Flows: Water Cost  
Telephone Cost  
Electric Cost  
Room Price

Output Flows: Payment Receipt  
Add Receipt Information

---

## Generate Receipt

## Process

Process #: 5.5.2

### Location:

5.5.x Generate Payment Receipt ( 5.5 )

Input Flows: Utility Cost

Output Flows: Payment Receipt

Add Receipt Information

---

## Generate Report

## Process

### Description:

The Generate Report process is to provide the general report such as monthly

report, annual report and so on.

**Process #:** 6

**Location:**

Level0 (0)

Input Flows: Report Require

Output Flows: Annual Report

Monthly Report

Memo

Analysis Report

---

**Generate Reservation Receipt**

**Process**

**Process #:** 2.4.4

**Location:**

2.4.x Reservation Process (2.4)

Input Flows: Reserver data

Output Flows: Reservation Receipt

Update Room Status

---

**Generate Room Information**

**Process**

**Process #:** 2.4.2

**Location:**

2.4.x Reservation Process (2.4)

Input Flows: Room Data

Output Flows: Reservation Information

Check Room Status

---

### Get New Electric Unit

### Process

**Process #:** 5.3.1

**Location:**

5.3.x Calculate Electric Cost ( 5.3 )

Input Flows: Electric Information

Output Flows: New Electric Meter Number

---

### Get New Water Unit

### Process

**Process #:** 5.2.1

**Location:**

5.2.x Calculate Water Cost ( 5.2 )

Input Flows: Water Information

Output Flows: New Water Meter Number

---

### Get Reserver Data

### Process

**Process #:** 2.4.3

**Location:**

2.4.x Reservation Process ( 2.4 )

Input Flows: Reserve Room Data

Output Flows: Reserver data

Add Reserver Data

---

## Get Telephone Information

## Process

**Process #:** 5.4.1

### Location:

5.4.x Calculate Telephone Cost ( 5.4 )

Input Flows: Telephone Information

Output Flows: Add Telephone Information

Telephone\_Sub\_Total

---

## Get Tenant Data

## Process

**Process #:** 2.2.1

### Location:

2.2.x Contraction Process ( 2.2 )

Input Flows: Make Contract

Output Flows: Add Tenant Data

Tenant Data

---

## Maintain Information

## Process

### Description:

The Maintain Base Information process is provided to you to manage the database's information such as Room cost.

**Process #:** 4

### Location:

Level0 ( 0 )

Input Flows: Maintain Information Base

Output Flows: Information Base Update

---

## Query Room Information

## Process

**Process #:** 3.2.3

**Location:**

3.2.x Search Room Information ( 3.2 )

Input Flows: Room Detail  
Tenant ID SQL  
Room Number SQL

Output Flows: Query Room  
Room Information

---

## Query Tenant Information

## Process

**Process #:** 3.3.3

**Location:**

3.3.x Search Tenant Information ( 3.3 )

Input Flows: Tenant Detail  
Tenant Name SQL  
Tenant ID SQL

Output Flows: Tenant Information  
Query Tenant

---

## Reservation Process

## Process

**Description:**

This process receives the reserver information such as reserver name, address and so on and generates reservation receipt.

**Process #:** 2.4

**Location:**

Room Service (2)

Input Flows: Reserve Room

Output Flows: Reservation Information

Reservation Receipt

Update Room Status

Check Room Status

Add Reserver Data

---

## Room Payment

## Process

### Description:

The Room Payment process calculates the utility cost, room cost, telephone cost and generates billing to the tenant.

**Process #:** 5

**Location:**

Level0 (0)

Input Flows: Utility Information

Output Flows: Payment Receipt

---

## Room Service

## Process

### Description:

The Room Service is the process that manages the information about Reservation, Sign Agreement, Cancel Agreement.

**Process #:** 2

**Location:**

Level0 ( 0 )

Input Flows: Service Information

Output Flows: Lease Agreement

Reservation Information

Reservation Receipt

Cancel Lease Agreement

---

**Search Month Information**

**Process**

**Process #:** 6.2.1

**Location:**

6.2.x Generate Monthly Report ( 6.2 )

Input Flows: Month Data

Month Information

Output Flows: Query Month Information

Month Information

---

**Search Room by Room Number**

**Process**

**Process #:** 3.2.1

**Location:**

3.2.x Search Room Information ( 3.2 )

Input Flows: Room Number

Output Flows: Room Number SQL

Search Room by Tenant Information	Process
<b>Process #:</b> 3.2.2  <b>Location:</b>  3.2.x Search Room Information ( 3.2 )  <div> Input Flows: Tenant ID Output Flows: Tenant ID SQL </div>	

Search Room Information	Process
<b>Description:</b>  Get room information and generate the SQL command to search in database.  <b>Process #:</b> 3.2  <b>Location:</b>  Searching Information ( 3 )  <div> Input Flows: Room Number Room Detail Output Flows: Query Room Room Information </div>	

Search Tenant by Tenant ID	Process
<b>Process #:</b> 3.3.1  <b>Location:</b>  3.3.x Search Tenant Information ( 3.3 )  <div> Input Flows: Tenant ID </div>	

Output Flows:            Tenant ID SQL

---

### Search Tenant by Tenant Name

### Process

Process #:        3.3.2

#### Location:

3.3.x Search Tenant Information ( 3.3 )

Input Flows:            Tenant Name

Output Flows:           Tenant Name SQL

---

### Search Tenant Information

### Process

#### Description:

Get tenant information and generate the SQL command to search in database.

Process #:        3.3

#### Location:

Searching Information        ( 3 )

Input Flows:            Tenant Name

Tenant Detail

Output Flows:           Tenant Information

Query Tenant

---

### Search Year Information

### Process

Process #:        6.3.1

#### Location:

6.3.x Generate Annual Report ( 6.3 )

Input Flows:	Year Data
	Year Information
Output Flows:	Query Year Information
	Month Information

## Searching Information

## Process

### Description:

The Searching Information process is to query the information of the tenant from database and to show it to the user.

Process #: 3

### Location:

Level0 ( 0 )

Input Flows: Search Information

Output Flows: Tenant Information

Room Information

## SSAS

## Process

### Description:

The SS Apartment System (SSAS) is used to control all the processes of the SS apartment such as Billing, Check-In, Payment, Reservation and so on.

Process #: 0

### Location:

context diagram ( CONTEXT )

Input Flows: Payment Require

Sign Contract

Terminate Lease Agreement

Report Require

Search Information

Reservation Require

Search Require

Maintain Information Base

Utility Information

Service Information

Output Flows: Payment Receipt

Lease Agreement

Annual Report

Reservation Information

Reservation Receipt

Cancel Lease Agreement

Monthly Report

Tenant Information

Checked Request

Information Base Update

---

## Update Building

## Process

### Description:

The process receives building information and updates building information such as building name, location.

**Process #:** 4.6

**Location:**

Maintain Base Information ( 4 )

Input Flows:	Building Information
	Building Information
Output Flows:	Building Update
	Save Building Information
	Search Building Informatin

Update Room Process

Description:

The process receives room information and updates room information such as room rate, water meter number.

Process #: 4.2

Location:

Maintain Base Information ( 4 )

Input Flows:	Room Information
	Room Informatin
Output Flows:	Room Update
	Save Room Information
	Search Room Informatin

Update Room Information Process

Process #: 2.3.1

Location:

2.3.x Cancel Contraction Process ( 2.3 )

Input Flows:

Cancel Contract

Output Flows:

Update Room Status

Room Data

Update Tenant

Process

Description:

The process receives tenant information and updates tenant information such as tenant name, address, telephone number.

Process #: 4.4

Location:

Maintain Base Information ( 4 )

Input Flows:

Tenant Information

Tenant Information

Output Flows:

Tenant Update

Save Tenants Information

Search Tenant Informatin

Update User

Process

Description:

The process receives user information and updates user information such as user name, password.

Process #: 4.3

Location:

Maintain Base Information ( 4 )

Input Flows:

User Information

	User Information
Output Flows:	User Update
	Save User Information
	Search User Information

Update Utility

Process

Description:

The process receives utility information and updates utility information such as utility price, utility name.

Process #: 4.5

Location:

Maintain Base Information ( 4 )

Input Flows:	Utility Information
	Utility Information
Output Flows:	Utility Update
	Save Utility Information
	Search Utility Information



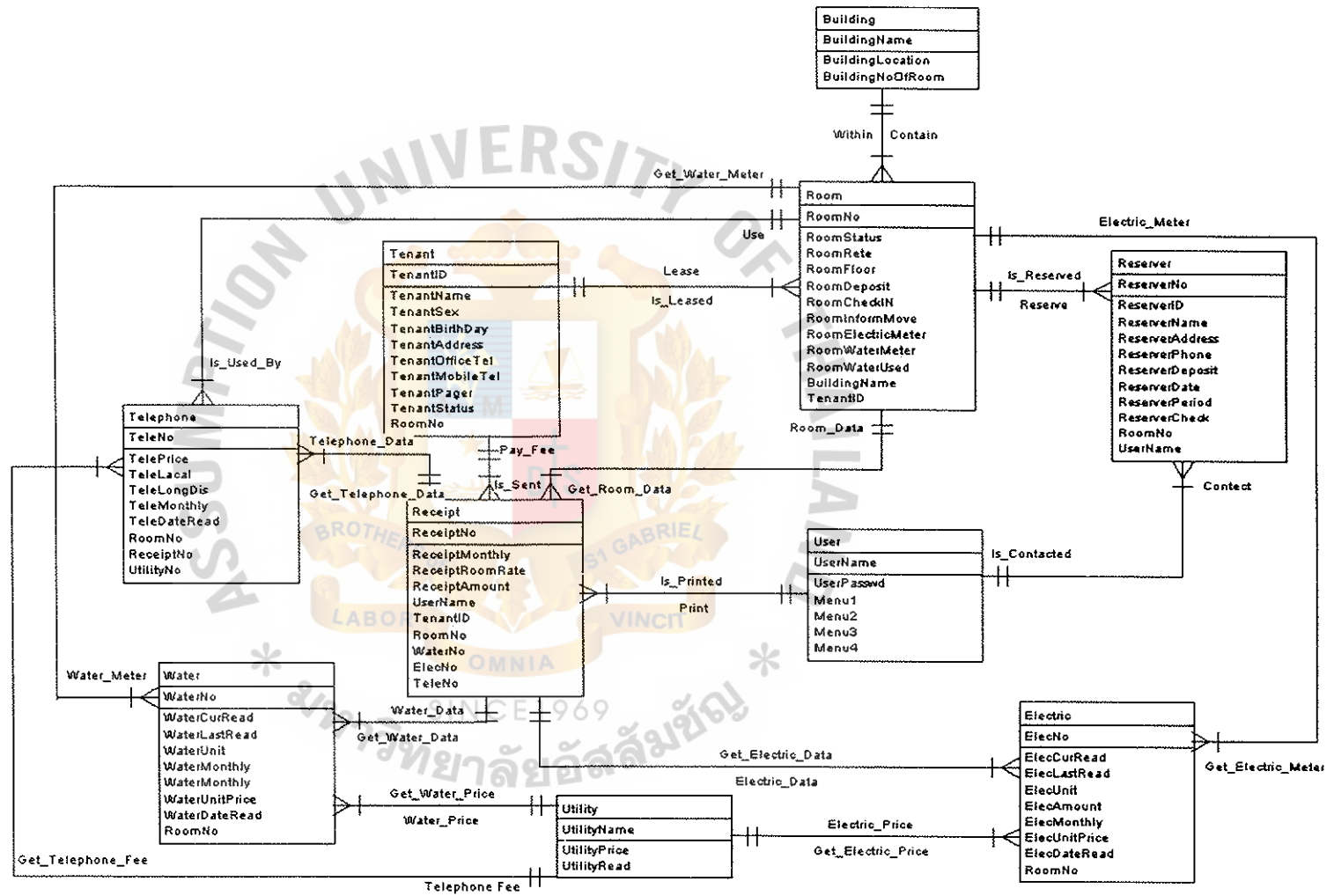


Figure C.1 Entity Relationship Diagram of SS Apartment System



## **APPENDIX D**

### **Data Dictionary**

## Building

## Entity

### Composition:

BuildingName[1] : String

BuildingLocation[100] : String

BuildingNoOfRoom : Integer

### Primary Key:

Column(s): BuildingName

### Location:

#### ERD

Attached relationships on ERD:

Contain

MIN: 1 MAX: many

Room

## Electric

## Entity

### Composition:

ElecNo : Long

ElecCurRead : Long

ElecLastRead : Integer

ElecUnit : Integer

ElecAmount : Long

ElecMonthly[10] : String

ElecUnitPrice : Integer

ElecDateRead[2] : String

RoomNo[4] : String

**Primary Key:**

Column(s): ElecNo

**Foreign Key(s):**

Room 'Electric\_MeterElectricic

Utility 'Electric\_PriceElectricic

Water 'Electric\_PriceElectricic

Receipt 'Get\_Electric\_DataElectricic

**Location:**

ERD

Attached relationships on ERD:

Get\_Electric\_Price MIN: 1 MAX: 1

Utility

Electric\_Data MIN: 1 MAX: 1

Receipt

Get\_Electric\_Meter MIN: 1 MAX: 1

Room

Receipt

Entity

**Composition:**

ReceiptNo[7] : String

ReceiptMonthly[10] : String

ReceiptRoomRate : Integer

ReceiptAmount : Long

UserName[15] : String

TenantID[13] : String

RoomNo[4] : String

WaterNo : Long

ElecNo : Long

TeleNo : Long

**Primary Key:**

Column(s):        ReceiptNo

**Foreign Key(s):**

Room 'Room\_Data' Receipt

Tenant 'Pay\_Fee' Receipt

User 'Print' Receipt

**Location:**

ERD

*Attached relationships on ERD:*

Get\_Room\_Data                      MIN: 1    MAX: 1

Room

Is\_Printed                          MIN: 1    MAX: 1

User

Is\_Sent                              MIN: 1    MAX: 1

Tenant

Get\_Water\_Data                      MIN: 1    MAX: many

Water

Get\_Telephone\_Data                      MIN: 1    MAX: many

Telephone

Get\_Electric\_Data                      MIN: 1    MAX: many

Electric

## Reserver

## Entity

### *Composition:*

ReserverNo[7] : String

ReserverID[13] : String

ReserverName[50] : String

ReserverAddress[100] : String

ReserverPhone[10] : String

ReserverDeposit : Integer

ReserverDate[10] : String

ReserverPeriod[10] : String

ReserverCheck : Boolean

RoomNo[4] : String

UserName[15] : String

### *Primary Key:*

Column(s): ReserverNo

### *Foreign Key(s):*

Room 'Is\_Reserved' Reserver

User 'Is\_Contacted' Reserver

### *Location:*

ERD

*Attached relationships on ERD:*

Reserve MIN: 1 MAX: 1

Room

Contect MIN: 1 MAX: 1

User

## Room

## Entity

### Composition:

RoomNo[4] : String

RoomStatus[1] : String

RoomRete : Integer

RoomFloor[1] : String

RoomDeposit : Integer

RoomCheckIN[12] : String

RoomInformMove[10] : String

RoomElectricMeter[4] : String

RoomWaterMeter[4] : String

RoomWaterUsed : Integer

BuildingName[1] : String

TenantID[13] : String

### Primary Key:

Column(s): RoomNo

### Foreign Key(s):

Tenant 'Lease' Room

Building 'Contain' Room

### Location:

ERD

Attached relationships on ERD:

Is\_Reserved

MIN: 1 MAX: many

Reserver

Within

MIN: 1 MAX: 1

Building

Room\_Data MIN: 1 MAX: many

Receipt

Is\_Leased MIN: 1 MAX: 1

Tenant

Electric\_Meter MIN: 1 MAX: many

Electric

Water\_Meter MIN: 1 MAX: many

Water

Use MIN: 1 MAX: many

Telephone

**Telephone**

**Entity**

**Composition:**

TeleNo[6] : String

TelePrice : Integer

TeleLacal : Integer

TeleLongDis : Integer

TeleMonthly[7] : String

TeleDateRead : Date

RoomNo[4] : String

ReceiptNo[7] : String

UtilityNo[2] : String

**Primary Key:**

Column(s): TeleNo

**Foreign Key(s):**

Receipt 'Get\_Telephone\_Data' Telephone

Utility 'Telephone\_Fee' Telephone

Room 'Use' Telephone

**Location:**

ERD

*Attached relationships on ERD:*

Telephone\_Data MIN: 1 MAX: 1

Receipt

Is\_Used\_By MIN: 1 MAX: 1

Room

Get\_Telephone\_Fee MIN: 1 MAX: 1

Utility

**Tenant**

**Entity**

**Composition:**

TenantID[13] : String

TenantName[50] : String

TenantSex[1] : String

TenantBirthDay[10] : String

TenantAddress[150] : String

TenantOfficeTel[10] : String

TenantMobileTel[10] : String

TenantPager[11] : String

TenantStatus[1] : String



Reserver

Print

MIN: 1 MAX: many

Receipt

---

**Utility**

**Entity**

***Composition:***

UtilityName[30] : String

UtilityPrice : Integer

UtilityRead[2] : String

***Primary Key:***

Column(s): UtilityName

***Location:***

ERD

*Attached relationships on ERD:*

Electric\_Price MIN: 1 MAX: many

Electric

Water\_Price MIN: 1 MAX: many

Water

Telephone Fee MIN: 1 MAX: many

Telephone

---

**Water**

**Entity**

***Composition:***

WaterNo : Long

WaterCurRead : Integer

WaterLastRead : Integer

WaterUnit : Integer

WaterMonthly : String

WaterMonthly[10] : String

WaterUnitPrice : Integer

WaterDateRead[2] : String

RoomNo[4] : String

**Primary Key:**

Column(s): WaterNo

**Foreign Key(s):**

Room 'Water\_MeterWaterer

Utility 'Water\_PriceWaterer

Receipt 'Get\_Water\_DataWaterer

**Location:**

ERD

Attached relationships on ERD:

Water\_Data MIN: 1 MAX: 1

Receipt

Get\_Water\_Price MIN: 1 MAX: 1

Utility

Get\_Water\_Meter MIN: 1 MAX: 1

Room

---

Boolean		Data Element	
Location:	Data Element -->	Reserver::ReserverCheck	
	Data Element -->	Reserver::ReserverCheck	
<hr/>			
BuildingLocation		Data Element	
Building::BuildingLocation			
Data Type:	String		
Location:	Entity -->	Building	
<hr/>			
BuildingLocation		Data Element	
Building::BuildingLocation			
Data Type:	String		
Location:	Data Store -->	Building	
<hr/>			
BuildingName		Data Element	
Building::BuildingName			
Data Type:	String		
Location:	Entity -->	Building	
<hr/>			
BuildingName		Data Element	
Building::BuildingName			
Data Type:	String		
Location:	Data Store -->	Building	

<b>BuildingName</b>	<b>Data Element</b>	
Room::BuildingName		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Room
<hr/>		
<b>BuildingName</b>	<b>Data Element</b>	
Room::BuildingName		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Room
<hr/>		
<b>BuildingNoOfRoom</b>	<b>Data Element</b>	
Building::BuildingNoOfRoom		
<b>Data Type:</b>	Integer M	
<b>Location:</b>	Entity -->	Building
<hr/>		
<b>BuildingNoOfRoom</b>	<b>Data Element</b>	
Building::BuildingNoOfRoom		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Data Store -->	Building
<hr/>		
<b>Date</b>	<b>Data Element</b>	
<b>Location:</b>	Data Element -->	Telephone::TeleDateRead
<hr/>		

[illegible][illegible][illegible]

**Table S6.**

\_\_\_\_\_

**ElecDateRead** **Data Element**

Electric::ElecDateRead

**Data Type:** String  
**Location:** Data Store --> Electric

**ElecLastRead** **Data Element**

Electric::ElecLastRead

**Data Type:** Integer  
**Location:** Entity --> Electric

**ElecLastRead** **Data Element**

Electric::ElecLastRead

**Data Type:** Integer  
**Location:** Data Store --> Electric

**ElecMonthly** **Data Element**

Electric::ElecMonthly

**Data Type:** String  
**Location:** Entity --> Electric

**ElecMonthly** **Data Element**

Electric::ElecMonthly

**Data Type:** String  
**Location:** Data Store --> Electric

<b>ElecNo</b>	<b>Data Element</b>	
Electric::ElecNo		
<b>Data Type:</b>	Long	
<b>Location:</b>	Entity -->	Electric

<b>ElecNo</b>	<b>Data Element</b>	
Electric::ElecNo		
<b>Data Type:</b>	Long	
<b>Location:</b>	Data Store -->	Electric

<b>ElecNo</b>	<b>Data Element</b>	
Receipt::ElecNo		
<b>Data Type:</b>	Long	
<b>Location:</b>	Entity -->	Receipt

<b>ElecNo</b>	<b>Data Element</b>	
Receipt::ElecNo		
<b>Data Type:</b>	Long	
<b>Location:</b>	Data Store -->	Receipt

<b>ElecUnit</b>	<b>Data Element</b>	
Electric::ElecUnit		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Electric

<b>ElecUnit</b>		<b>Data Element</b>
Electric::ElecUnit		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Data Store -->	Electric
<b>ElecUnitPrice</b>		<b>Data Element</b>
Electric::ElecUnitPrice		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Data Store -->	Electric
<b>ElecUnitPrice</b>		<b>Data Element</b>
Electric::ElecUnitPrice		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Electric
<b>Integer</b>		<b>Data Element</b>
<b>Location:</b>		
Data Element -->	Room::RoomRete	
Data Element -->	Room::RoomDeposit	
Data Element -->	Reserver::ReserverDeposit	
Data Element -->	Electric::ElecLastRead	
Data Element -->	Electric::ElecUnit	
Data Element -->	Water::WaterUnit	
Data Element -->	Telephone::TelePrice	

Data Element -->	Telephone::TeleLacal
Data Element -->	Telephone::TeleLongDis
Data Element -->	Room::RoomRate
Data Element -->	Room::RoomDeposit
Data Element -->	Room::RoomWaterUsed
Data Element -->	Receipt::ReceiptRoomRate
Data Element -->	Electric::ElecLastRead
Data Element -->	Electric::ElecUnit
Data Element -->	Electric::ElecUnitPrice
Data Element -->	Utility::UtilityPrice
Data Element -->	Water::WaterCurRead
Data Element -->	Water::WaterLastRead
Data Element -->	Water::WaterUnit
Data Element -->	Water::WaterUnitPrice
Data Element -->	Telephone::TeleCharge
Data Element -->	Reserver::ReserverDeposit
Data Element -->	Building::BuildingNoOfRoom
Data Element -->	Building::BuildingNoOfRoom
Data Element -->	Room::RoomWaterUsed
Data Element -->	Electric::ElecUnitPrice
Data Element -->	Utility::UtilityPrice
Data Element -->	Receipt::ReceiptRoomRate
Data Element -->	Water::WaterCurRead
Data Element -->	Water::WaterLastRead
Data Element -->	Water::WaterUnitPrice

---

**Long****Data Element****Location:**

Data Element -->	Receipt::ReceiptAmount
Data Element -->	Receipt::WaterNo
Data Element -->	Receipt::ElecNo
Data Element -->	Receipt::TeleNo
Data Element -->	Electric::ElecNo
Data Element -->	Electric::ElecCurRead
Data Element -->	Electric::ElecAmount
Data Element -->	Water::WaterNo
Data Element -->	Water::WaterAmount
Data Element -->	Telephone::TeleNo
Data Element -->	Telephone::TelePrice
Data Element -->	Electric::ElecNo
Data Element -->	Electric::ElecCurRead
Data Element -->	Electric::ElecAmount
Data Element -->	Receipt::ReceiptAmount
Data Element -->	Receipt::WaterNo
Data Element -->	Receipt::ElecNo
Data Element -->	Receipt::TeleNo
Data Element -->	Water::WaterNo

---

**Menu1****Data Element**

User::Menu1

**Data Type:** String

**Location:**

Entity --> User

---

**Menu2** **Data Element**

User::Menu2

**Data Type:** String

**Location:**

Entity --> User

---

**Menu3** **Data Element**

User::Menu3

**Data Type:** String

**Location:**

Entity --> User

---

**Menu4** **Data Element**

User::Menu4

**Data Type:** String

**Location:** Entity --> User

---

**ReceiptAmount** **Data Element**

Receipt::ReceiptAmount

**Data Type:** Long

**Location:** Entity --> Receipt

**ReceiptAmount**

**Data Element**

Receipt::ReceiptAmount

**Data Type:** Long

**Location:** Data Store --> Receipt

**ReceiptMonthly**

**Data Element**

Receipt::ReceiptMonthly

**Data Type:** String

**Location:** Entity --> Receipt

**ReceiptMonthly**

**Data Element**

Receipt::ReceiptMonthly

**Data Type:** String

**Location:** Data Store --> Receipt

**ReceiptNo**

**Data Element**

Receipt::ReceiptNo

**Data Type:** String

**Location:** Entity --> Receipt

**ReceiptNo**

**Data Element**

Receipt::ReceiptNo

**Data Type:** String

**Location:** Data Store --> Receipt

---

<b>ReceiptNo</b>	<b>Data Element</b>
Telephone::ReceiptNo	
<b>Data Type:</b>	String
<b>Location:</b>	Entity --> Telephone

---

<b>ReceiptRoomRate</b>	<b>Data Element</b>
Receipt::ReceiptRoomRate	
<b>Data Type:</b>	Integer
<b>Location:</b>	Entity --> Receipt

---

<b>ReceiptRoomRate</b>	<b>Data Element</b>
Receipt::ReceiptRoomRate	
<b>Data Type:</b>	Integer
<b>Location:</b>	Data Store --> Receipt

---

<b>ReserverAddress</b>	<b>Data Element</b>
Reserver::ReserverAddress	
<b>Data Type:</b>	String
<b>Location:</b>	Entity --> Reserver

---

<b>ReserverAddress</b>	<b>Data Element</b>
Reserver::ReserverAddress	
<b>Data Type:</b>	String
<b>Location:</b>	Data Store --> Reserver

<b>ReserverCheck</b>	<b>Data Element</b>	
Reserver::ReserverCheck		
<b>Data Type:</b>	Boolean	
<b>Location:</b>	Data Store -->	Reserver
<b>ReserverCheck</b>	<b>Data Element</b>	
Reserver::ReserverCheck		
<b>Data Type:</b>	Boolean	
<b>Location:</b>	Entity -->	Reserver
<b>ReserverDate</b>	<b>Data Element</b>	
Reserver::ReserverDate		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Reserver
<b>ReserverDate</b>	<b>Data Element</b>	
Reserver::ReserverDate		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Reserver
<b>ReserverDeposit</b>	<b>Data Element</b>	
Reserver::ReserverDeposit		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Reserver

---

**ReserverDeposit****Data Element**

Reserver::ReserverDeposit

**Data Type:** Integer**Location:** Data Store --> Reserver

---

**ReserverID****Data Element**

Reserver::ReserverID

**Data Type:** String**Location:** Entity --> Reserver

---

**ReserverID****Data Element**

Reserver::ReserverID

**Data Type:** String**Location:** Data Store --> Reserver

---

**ReserverName****Data Element**

Reserver::ReserverName

**Data Type:** String**Location:** Entity --> Reserver

---

**ReserverName****Data Element**

Reserver::ReserverName

**Data Type:** String**Location:** Data Store --> Reserver

---

ReserverNo	Data Element
------------	--------------

Reserver::ReserverNo

**Data Type:** String

**Location:** Entity --> Reserver

---

ReserverNo	Data Element
------------	--------------

Reserver::ReserverNo

**Data Type:** String

**Location:** Data Store --> Reserver

---

ReserverPeriod	Data Element
----------------	--------------

Reserver::ReserverPeriod

**Data Type:** String

**Location:** Entity --> Reserver

---

ReserverPeriod	Data Element
----------------	--------------

Reserver::ReserverPeriod

**Data Type:** String

**Location:** Data Store --> Reserver

---

ReserverPhone	Data Element
---------------	--------------

Reserver::ReserverPhone

**Data Type:** String

**Location:** Entity --> Reserver

---

**ReserverPhone****Data Element**

Reserver::ReserverPhone

**Data Type:** String**Location:** Data Store --> Reserver

---

**RoomCheckIN****Data Element**

Room::RoomCheckIN

**Data Type:** String**Location:** Entity --> Room

---

**RoomCheckIn****Data Element**

Room::RoomCheckIn

**Data Type:** String**Location:** Data Store --> Room

---

**RoomDeposit****Data Element**

Room::RoomDeposit

**Data Type:** Integer**Location:** Entity --> Room

---

**RoomDeposit****Data Element**

Room::RoomDeposit

**Data Type:** Integer**Location:** Data Store --> Room

<b>RoomElectricMeter</b>		<b>Data Element</b>
Room::RoomElectricMeter		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Room
<hr/>		
<b>RoomElectricMeter</b>		<b>Data Element</b>
Room::RoomElectricMeter		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Room
<hr/>		
<b>RoomFloor</b>		<b>Data Element</b>
Room::RoomFloor		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Room
<hr/>		
<b>RoomFloor*</b>		<b>Data Element</b>
Room::RoomFloor		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Room
<hr/>		
<b>RoomInformMove</b>		<b>Data Element</b>
Room::RoomInformMove		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Room

<b>RoomInformMove</b>		<b>Data Element</b>
Room::RoomInformMove		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Room
<b>RoomNo</b>		<b>Data Element</b>
Electric::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Electric
<b>RoomNo</b>		<b>Data Element</b>
Electric::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Electric
<b>RoomNo</b>		<b>Data Element</b>
Receipt::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Receipt
<b>RoomNo</b>		<b>Data Element</b>
Receipt::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Receipt

**RoomNo** **Data Element**

Reserver::RoomNo

**Data Type:** String

**Location:** Entity --> Reserver

**RoomNo** **Data Element**

Reserver::RoomNo

**Data Type:** String

**Location:** Data Store --> Reserver

**RoomNo** **Data Element**

Room::RoomNo

**Data Type:** String

**Location:** Entity --> Room

**RoomNo** **Data Element**

Room::RoomNo

**Data Type:** String

**Location:** Data Store --> Room

**RoomNo** **Data Element**

Telephone::RoomNo

**Data Type:** String

**Location:** Entity --> Telephone

<hr/>		
<b>RoomNo</b>	<b>Data Element</b>	
Telephone::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Telephone
<hr/>		
<b>RoomNo</b>	<b>Data Element</b>	
Tenant::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Tenant
<hr/>		
<b>RoomNo</b>	<b>Data Element</b>	
Tenants::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Tenants
<hr/>		
<b>RoomNo</b>	<b>Data Element</b>	
Water::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Water
<hr/>		
<b>RoomNo</b>	<b>Data Element</b>	
Water::RoomNo		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Water

RoomRate		Data Element
Room::RoomRate		
Data Type:	Integer	
Location:	Data Store -->	Room

RoomRete		Data Element
Room::RoomRete		
Data Type:	Integer	
Location:	Entity -->	Room

RoomStatus		Data Element
Room::RoomStatus		
Data Type:	String	
Location:	Entity -->	Room

RoomStatus		Data Element
Room::RoomStatus		
Data Type:	String	
Location:	Data Store -->	Room

RoomWaterMeter		Data Element
Room::RoomWaterMeter		
Data Type:	String	
Location:	Entity -->	Room

<b>RoomWaterMeter</b>		<b>Data Element</b>
Room::RoomWaterMeter		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Room
<hr/>		
<b>RoomWaterUsed</b>		<b>Data Element</b>
Room::RoomWaterUsed		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Data Store -->	Room
<hr/>		
<b>RoomWaterUsed</b>		<b>Data Element</b>
Room::RoomWaterUsed		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Room
<hr/>		
<b>String</b>	<b>Data Element</b>	
<b>Location:</b>		
Data Element -->	Room::RoomNo	
Data Element -->	Room::RoomStatus	
Data Element -->	Room::RoomFloor	
Data Element -->	Building::BuildingLocation	
Data Element -->	Reserver::ReserverNo	
Data Element -->	Reserver::ReserverID	
Data Element -->	Reserver::ReserverName	

Data Element -->	Reserver::ReserverAddress
Data Element -->	Reserver::ReserverPhone
Data Element -->	Reserver::RoomNo
Data Element -->	Reserver::UserName
Data Element -->	User::UserName
Data Element -->	User::UserPasswd
Data Element -->	Electric::ElecMonthly
Data Element -->	Electric::RoomNo
Data Element -->	Water::RoomNo
Data Element -->	Receipt::ReceiptNo
Data Element -->	Receipt::ReceiptMonthly
Data Element -->	Receipt::UserName
Data Element -->	Receipt::TenantID
Data Element -->	Receipt::RoomNo
Data Element -->	Telephone::TeleNo
Data Element -->	Telephone::TeleMonthly
Data Element -->	Telephone::RoomNo
Data Element -->	Tenant::TenantID
Data Element -->	Tenant::TenantName
Data Element -->	Tenant::TenantAddress
Data Element -->	Tenant::TenantOfficeTel
Data Element -->	Tenant::TenantMobileTel
Data Element -->	Tenant::TenantPager
Data Element -->	Tenant::TenantStatus
Data Element -->	Telephone::ReceiptNo

Data Element -->	Telephone::UtilityNo
Data Element -->	Utility::UtilityName
Data Element -->	Room::RoomNo
Data Element -->	Room::RoomStatus
Data Element -->	Room::RoomFloor
Data Element -->	Room::RoomCheckIn
Data Element -->	Room::RoomInformMove
Data Element -->	Room::RoomElectricMeter
Data Element -->	Room::RoomWaterMeter
Data Element -->	Room::BuildingName
Data Element -->	Tenants::TenantID
Data Element -->	Tenants::TenantName
Data Element -->	Tenants::TenantSex
Data Element -->	Tenants::TenantBirthDay
Data Element -->	Tenants::TenantAddress
Data Element -->	Tenants::TenantOfficeTel
Data Element -->	Tenants::TenantMobileTel
Data Element -->	Tenants::TenantPager
Data Element -->	Tenants::TenantStatus
Data Element -->	Tenants::RoomNo
Data Element -->	Receipt::ReceiptNo
Data Element -->	Receipt::ReceiptMonthly
Data Element -->	Receipt::UserName
Data Element -->	Receipt::TenantID
Data Element -->	Receipt::RoomNo

Data Element -->	Electric::ElecMonthly
Data Element -->	Electric::ElecDateRead
Data Element -->	Utility::UtilityName
Data Element -->	Utility::UtilityRead
Data Element -->	Water::WaterMonthly
Data Element -->	Water::WaterDateRead
Data Element -->	Water::RoomNo
Data Element -->	Telephone::TeleMonthly
Data Element -->	Telephone::TeleDateRead
Data Element -->	Telephone::RoomNo
Data Element -->	Reserver::ReserverNo
Data Element -->	Reserver::ReserverID
Data Element -->	Reserver::ReserverName
Data Element -->	Reserver::ReserverAddress
Data Element -->	Reserver::ReserverPhone
Data Element -->	Reserver::ReserverDate
Data Element -->	Reserver::ReserverPeriod
Data Element -->	Reserver::RoomNo
Data Element -->	Reserver::UserName
Data Element -->	Building::BuildingName
Data Element -->	Building::BuildingLocation
Data Element -->	Building::BuildingName
Data Element -->	Room::RoomCheckIN
Data Element -->	Room::RoomInformMove
Data Element -->	Room::RoomElectricMeter

Data Element -->	Room::RoomWaterMeter
Data Element -->	Room::BuildingName
Data Element -->	Reserver::ReserverDate
Data Element -->	Reserver::ReserverPeriod
Data Element -->	User::Menu1
Data Element -->	User::Menu2
Data Element -->	User::Menu3
Data Element -->	User::Menu4
Data Element -->	Electric::ElecDateRead
Data Element -->	Utility::UtilityRead
Data Element -->	Water::WaterMonthly
Data Element -->	Water::WaterDateRead
Data Element -->	Tenant::TenantSex
Data Element -->	Tenant::TenantBirthDay
Data Element -->	Tenant::RoomNo

String

Data Element

Location:

Data Element -->	Electric::RoomNo
Data Element -->	Room::TenantID

TeleCharge

Data Element

Telephone::TeleCharge

Data Type: Integer

Location: Data Store --> Telephone

<b>TeleDateRead</b>	<b>Data Element</b>	
Telephone::TeleDateRead		
<b>Data Type:</b>	Date	
<b>Location:</b>	Entity -->	Telephone

---

<b>TeleDateRead</b>	<b>Data Element</b>	
Telephone::TeleDateRead		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Telephone

---

<b>TeleLacal</b>	<b>Data Element</b>	
Telephone::TeleLacal		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Telephone

---

<b>TeleLongDis</b>	<b>Data Element</b>	
Telephone::TeleLongDis		
<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Telephone

---

<b>TeleMonthly</b>	<b>Data Element</b>	
Telephone::TeleMonthly		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Telephone

**TeleMonthly**

**Data Element**

Telephone::TeleMonthly

**Data Type:** String

**Location:** Data Store --> Telephone

**TeleNo**

**Data Element**

Receipt::TeleNo

**Data Type:** Long

**Location:** Data Store --> Receipt

**TeleNo**

**Data Element**

Receipt::TeleNo

**Data Type:** Long

**Location:** Entity --> Receipt

**TeleNo**

**Data Element**

Telephone::TeleNo

**Data Type:** String

**Location:** Entity --> Telephone

**TeleNo**

**Data Element**

Telephone::TeleNo

**Data Type:** Long

**Location:** Data Store --> Telephone

TelePrice		Data Element
Telephone::TelePrice		
Data Type:	Integer	
Location:	Entity -->	Telephone

TelePrice		Data Element
Telephone::TelePrice		
Data Type:	Long	
Location:	Data Store -->	Telephone

TenantAddress		Data Element
Tenant::TenantAddress		
Data Type:	String	
Location:	Entity -->	Tenant

TenantAddress		* Data Element
Tenants::TenantAddress		
Data Type:	String	
Location:	Data Store -->	Tenants

TenantBirthDay		Data Element
Tenant::TenantBirthDay		
Data Type:	String	
Location:	Entity -->	Tenant

<hr/>		
TenantBirthDay	Data Element	
Tenants::TenantBirthDay		
Data Type:	String	
Location:	Data Store -->	Tenants
<hr/>		
TenantID	Data Element	
Receipt::TenantID		
Data Type:	String	
Location:	Entity -->	Receipt
<hr/>		
TenantID	Data Element	
Receipt::TenantID		
Data Type:	String	
Location:	Data Store -->	Receipt
<hr/>		
TenantID *	Data Element	
Room::TenantID		
Data Type:	String	
Location:	Entity -->	Room
<hr/>		
TenantID	Data Element	
Tenant::TenantID		
Data Type:	String	
Location:	Entity -->	Tenant

TenantID		Data Element
Tenants::TenantID		
Data Type:	String	
Location:	Data Store -->	Tenants

TenantMobileTel		Data Element
Tenant::TenantMobileTel		
Data Type:	String	
Location:	Entity -->	Tenant

TenantMobileTel		Data Element
Tenants::TenantMobileTel		
Data Type:	String	
Location:	Data Store -->	Tenants

TenantName		Data Element
Tenant::TenantName		
Data Type:	String	
Location:	Entity -->	Tenant

TenantName		Data Element
Tenants::TenantName		
Data Type:	String	
Location:	Data Store -->	Tenants

TenantOfficeTel		Data Element
Tenant::TenantOfficeTel		
Data Type:	String	
Location:	Entity -->	Tenant

TenantOfficeTel		Data Element
Tenants::TenantOfficeTel		
Data Type:	String	
Location:	Data Store -->	Tenants

TenantPager		Data Element
Tenant::TenantPager		
Data Type:	String	
Location:	Entity -->	Tenant

TenantPager		Data Element
Tenants::TenantPager		
Data Type:	String	
Location:	Data Store -->	Tenants

TenantSex		Data Element
Tenant::TenantSex		
Data Type:	String	
Location:	Entity -->	Tenant

<hr/>		
<b>TenantSex</b>	<b>Data Element</b>	
Tenants::TenantSex		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Tenants
<hr/>		
<b>TenantStatus</b>	<b>Data Element</b>	
Tenant::TenantStatus		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Tenant
<hr/>		
<b>TenantStatus</b>	<b>Data Element</b>	
Tenants::TenantStatus		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Tenants
<hr/>		
<b>UserName</b>	<b>Data Element</b>	
Receipt::UserName		
<b>Data Type:</b>	String	
<b>Location:</b>	Entity -->	Receipt
<hr/>		
<b>UserName</b>	<b>Data Element</b>	
Receipt::UserName		
<b>Data Type:</b>	String	
<b>Location:</b>	Data Store -->	Receipt

---

**UserName****Data Element**

Reserver::UserName

**Data Type:** String**Location:** Entity --> Reserver

---

**UserName****Data Element**

Reserver::UserName

**Data Type:** String**Location:** Data Store --> Reserver

---

**UserName****Data Element**

User::UserName

**Data Type:** String**Location:** Entity --> User

---

**UserPasswd****Data Element**

User::UserPasswd

**Data Type:** String**Location:** Entity --> User

---

**UtilityName****Data Element**

Utility::UtilityName

**Data Type:** String**Location:** Entity --> Utility

UtilityName	Data Element	
Utility::UtilityName		
Data Type:	String	
Location:	Data Store -->	Utility
UtilityNo	Data Element	
Telephone::UtilityNo		
Data Type:	String	
Location:	Entity -->	Telephone
UtilityPrice	Data Element	
Utility::UtilityPrice		
Data Type:	Integer	
Location:	Entity -->	Utility
UtilityPrice*	*Data Element	
Utility::UtilityPrice		
Data Type:	Integer	
Location:	Data Store -->	Utility
UtilityRead	Data Element	
Utility::UtilityRead		
Data Type:	String	
Location:	Entity -->	Utility

UtilityRead		Data Element
Utility::UtilityRead		
Data Type:	String	
Location:	Data Store -->	Utility

WaterAmount		Data Element
Water::WaterAmount		
Data Type:	Long	
Location:	Data Store -->	Water

WaterCurRead		Data Element
Water::WaterCurRead		
Data Type:	Integer	
Location:	Entity -->	Water

WaterCurRead		Data Element
Water::WaterCurRead		
Data Type:	Integer	
Location:	Data Store -->	Water

WaterDateRead		Data Element
Water::WaterDateRead		
Data Type:	String	
Location:	Entity -->	Water

WaterDateRead		Data Element
Water::WaterDateRead		
Data Type:	String	
Location:	Data Store -->	Water
WaterLastRead		Data Element
Water::WaterLastRead		
Data Type:	Integer	
Location:	Entity -->	Water
WaterLastRead		Data Element
Water::WaterLastRead		
Data Type:	Integer	
Location:	Data Store -->	Water
WaterMonthly		*Data Element
Water::WaterMonthly		
Data Type:	String	
Location:	Entity -->	Water
	Entity -->	Water
WaterMonthly		Data Element
Water::WaterMonthly		
Data Type:	String	

Location	Data Store -->	Water
<hr/>		
WaterNo		Data Element
Receipt::WaterNo		
Data Type:	Long	
Location:	Entity -->	Receipt
<hr/>		
WaterNo		Data Element
Receipt::WaterNo		
Data Type:	Long	
Location:		
Data Store -->	Receipt	
<hr/>		
WaterNo		Data Element
Water::WaterNo		
Data Type:	Long	
Location:	Entity -->	* Water
<hr/>		
WaterNo		Data Element
Water::WaterNo		
Data Type:	Long	
Location:	Data Store -->	Water
<hr/>		
WaterUnit		Data Element
Water::WaterUnit		

<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Water

---

<b>WaterUnit</b>	<b>Data Element</b>
------------------	---------------------

Water::WaterUnit

<b>Data Type:</b>	Integer	
<b>Location:</b>	Data Store -->	Water

---

<b>WaterUnitPrice</b>	<b>Data Element</b>
-----------------------	---------------------

Water::WaterUnitPrice

<b>Data Type:</b>	Integer	
<b>Location:</b>	Entity -->	Water

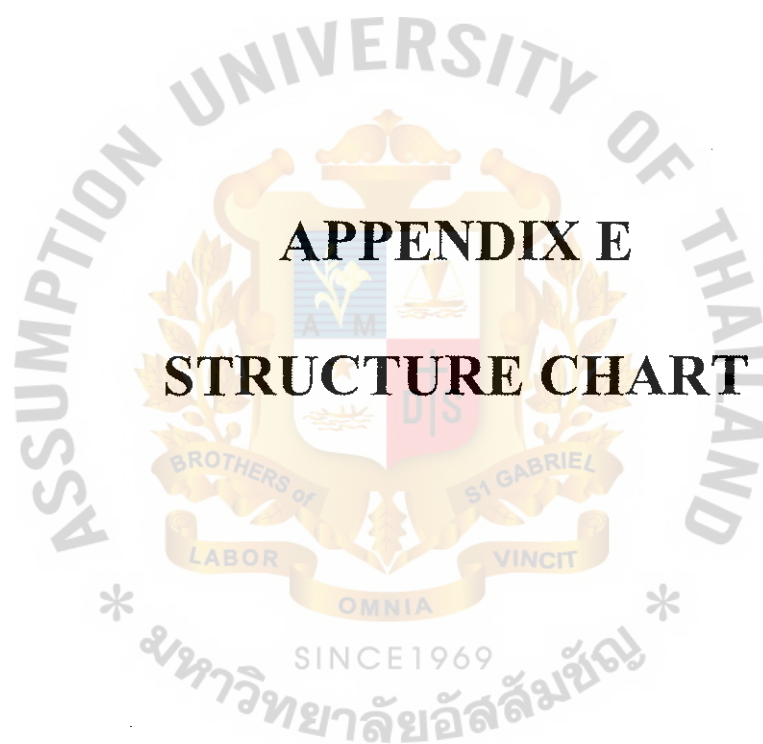
---

<b>WaterUnitPrice</b>	<b>Data Element</b>
-----------------------	---------------------

Water::WaterUnitPrice

<b>Data Type:</b>	Integer	
<b>Location:</b>	Data Store -->	Water

---



# **APPENDIX E**

## **STRUCTURE CHART**

Figure E.1 : Structure Chart of SS Apartment System.

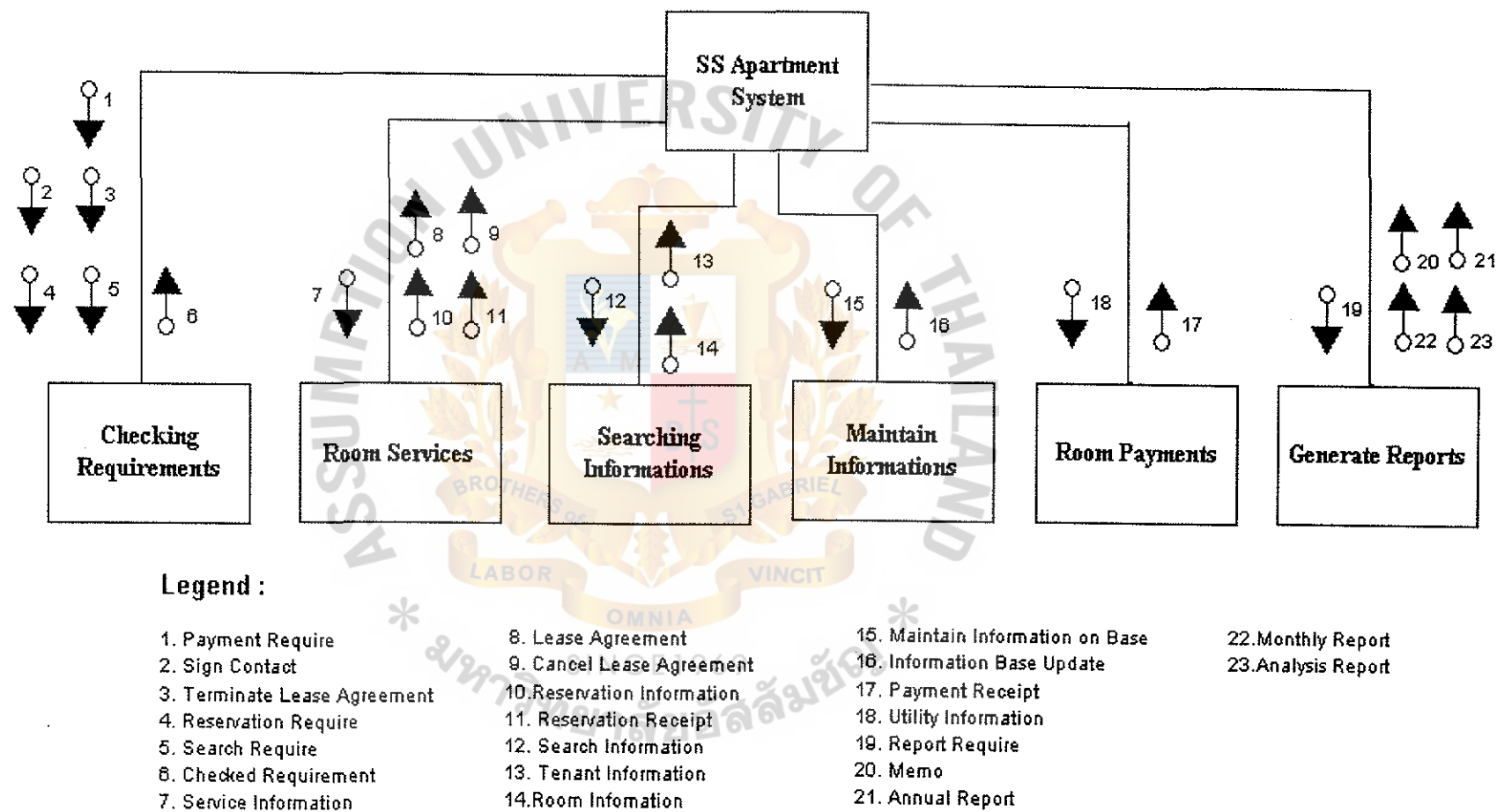


Figure E.2: Structure Chart of Checking Requirement

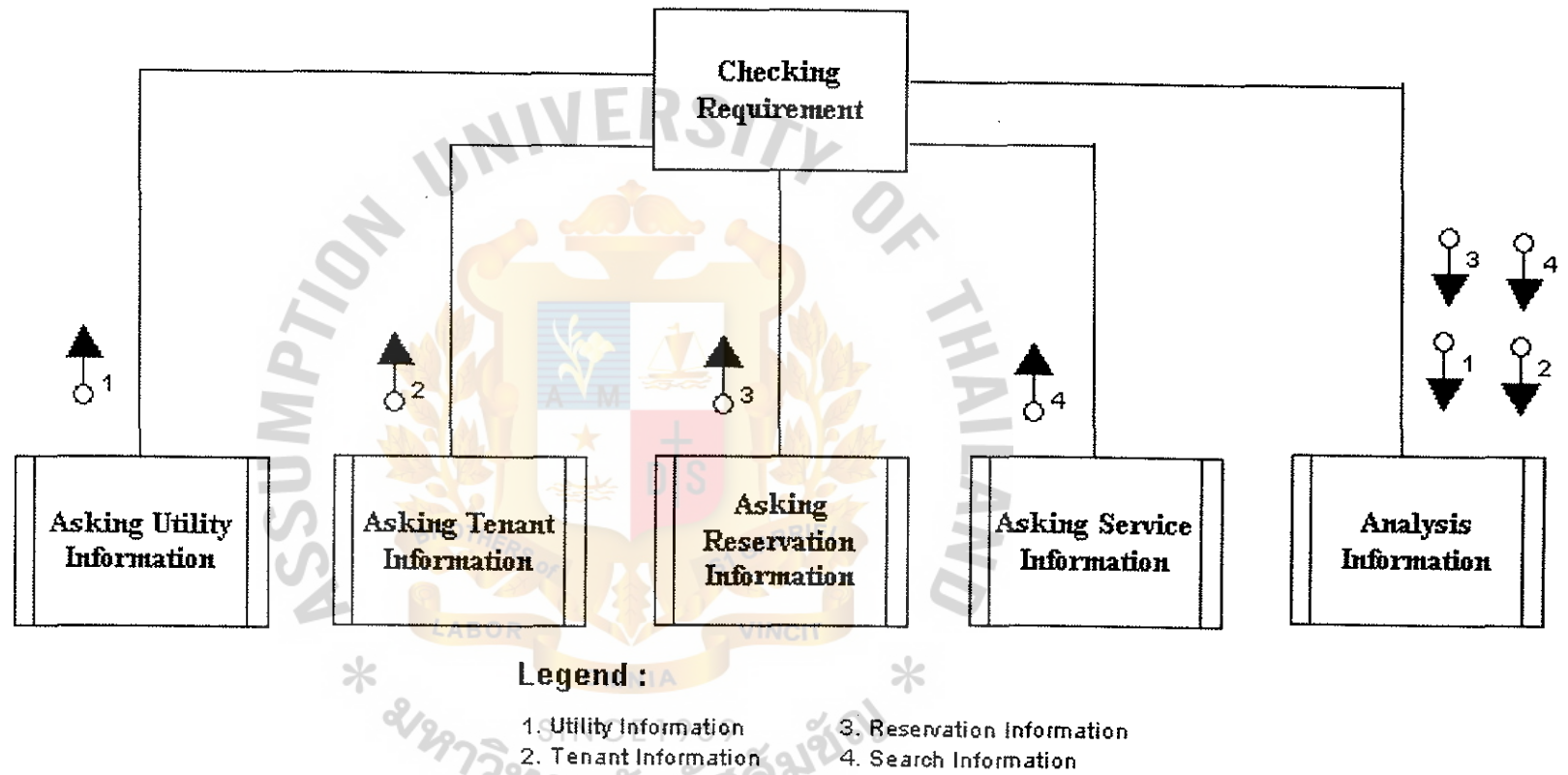


Figure E.3: Structure Chart of Room Service

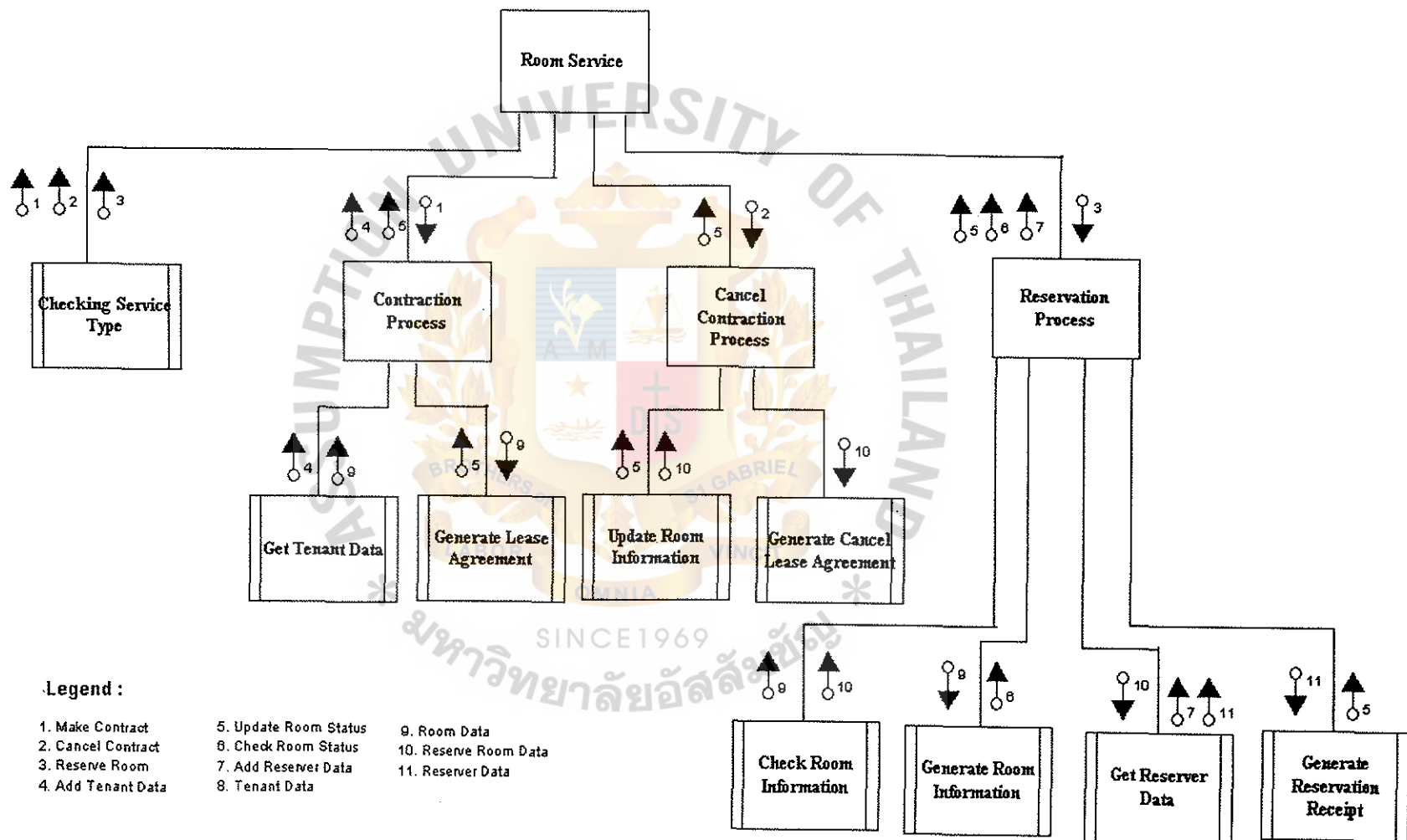


Figure E.4: Structure Chart of Searching Information

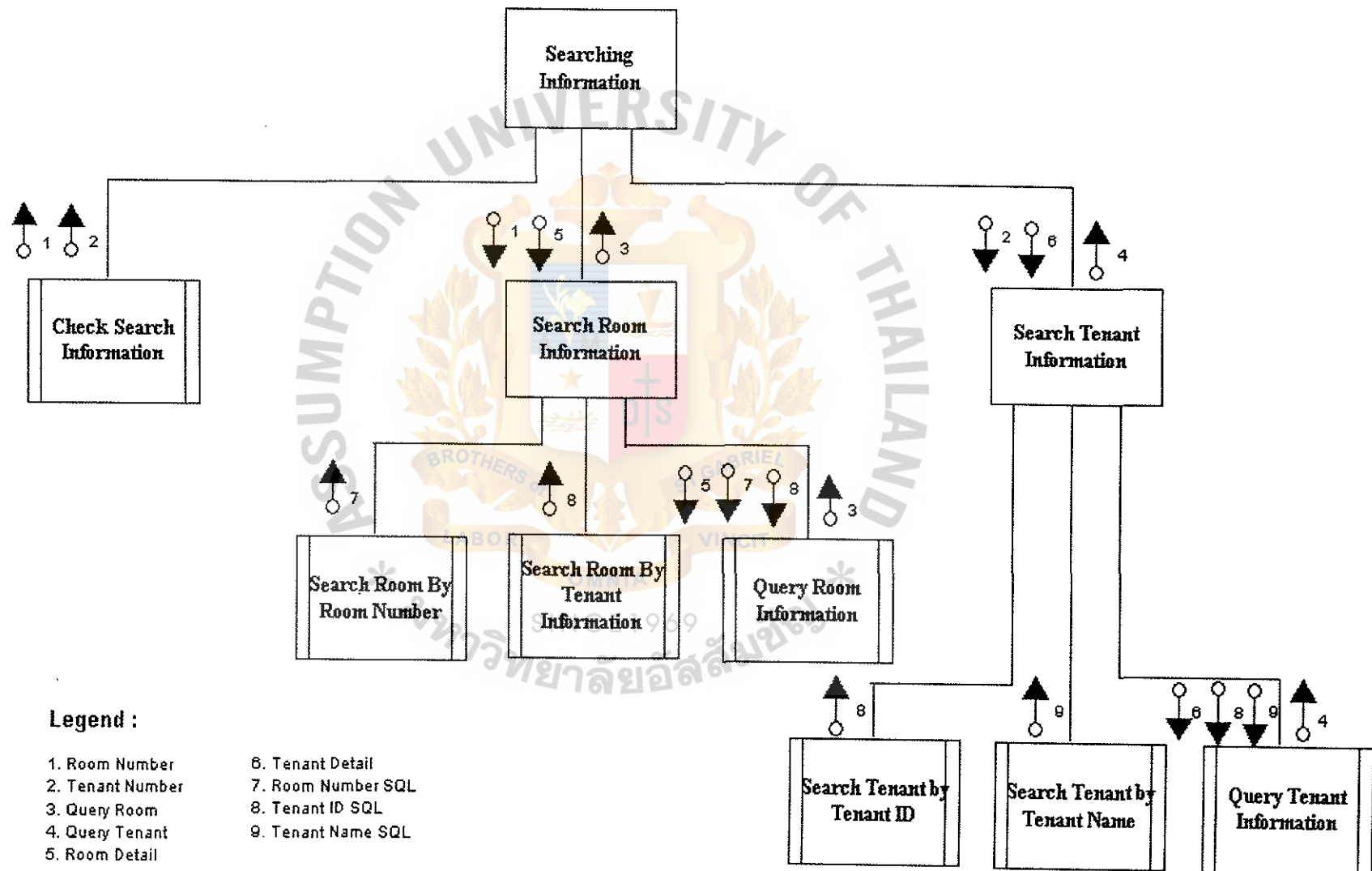
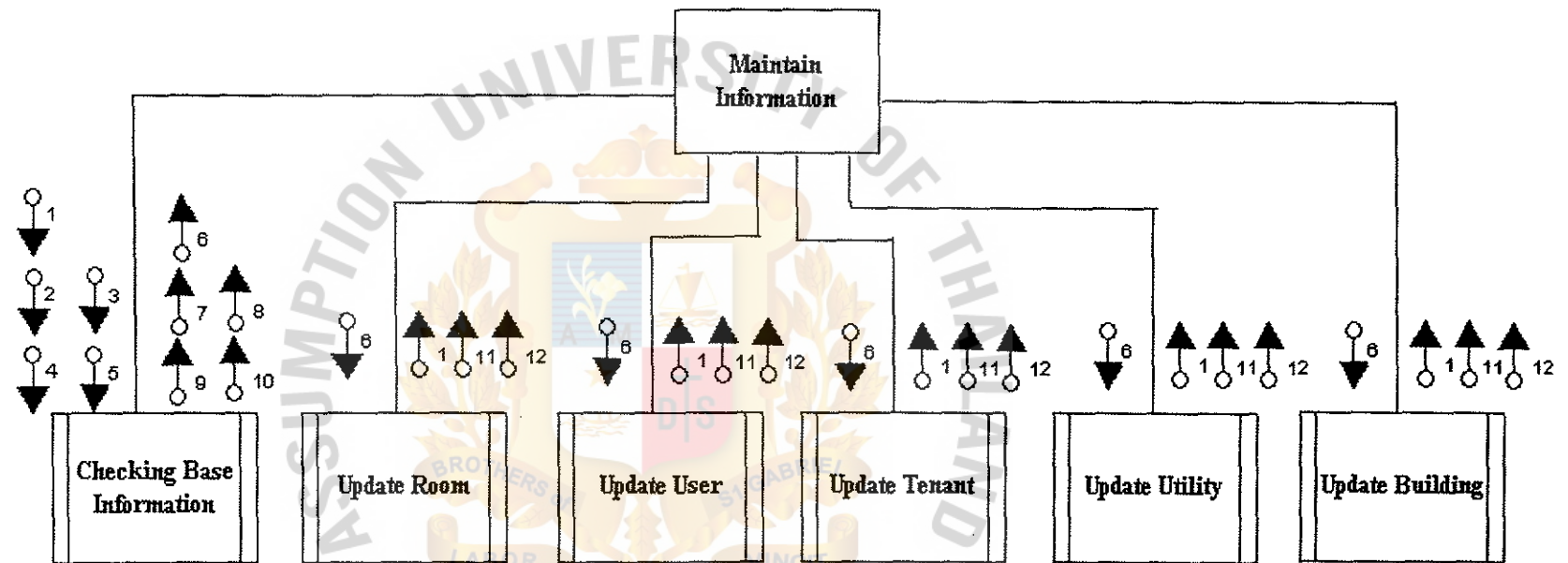


Figure E.5: Structure Chart of Maintain Information



**Legend :**

1. Room Update
2. User Update
3. Tenant Update
4. Utility Update
5. Building Update

6. Room Information
7. User Information
8. Tenant Information
9. Utility Information
10. Building Information

11. Search Room Information
12. Save Room Information
13. Search User Information
14. Save User Information
15. Search Tenant Information

16. Save Tenant Information
17. Search Utility Information
18. Save Utility Information
19. Search Building Information
20. Save Building Information

Figure E.6: Structure Chart of Room Payment

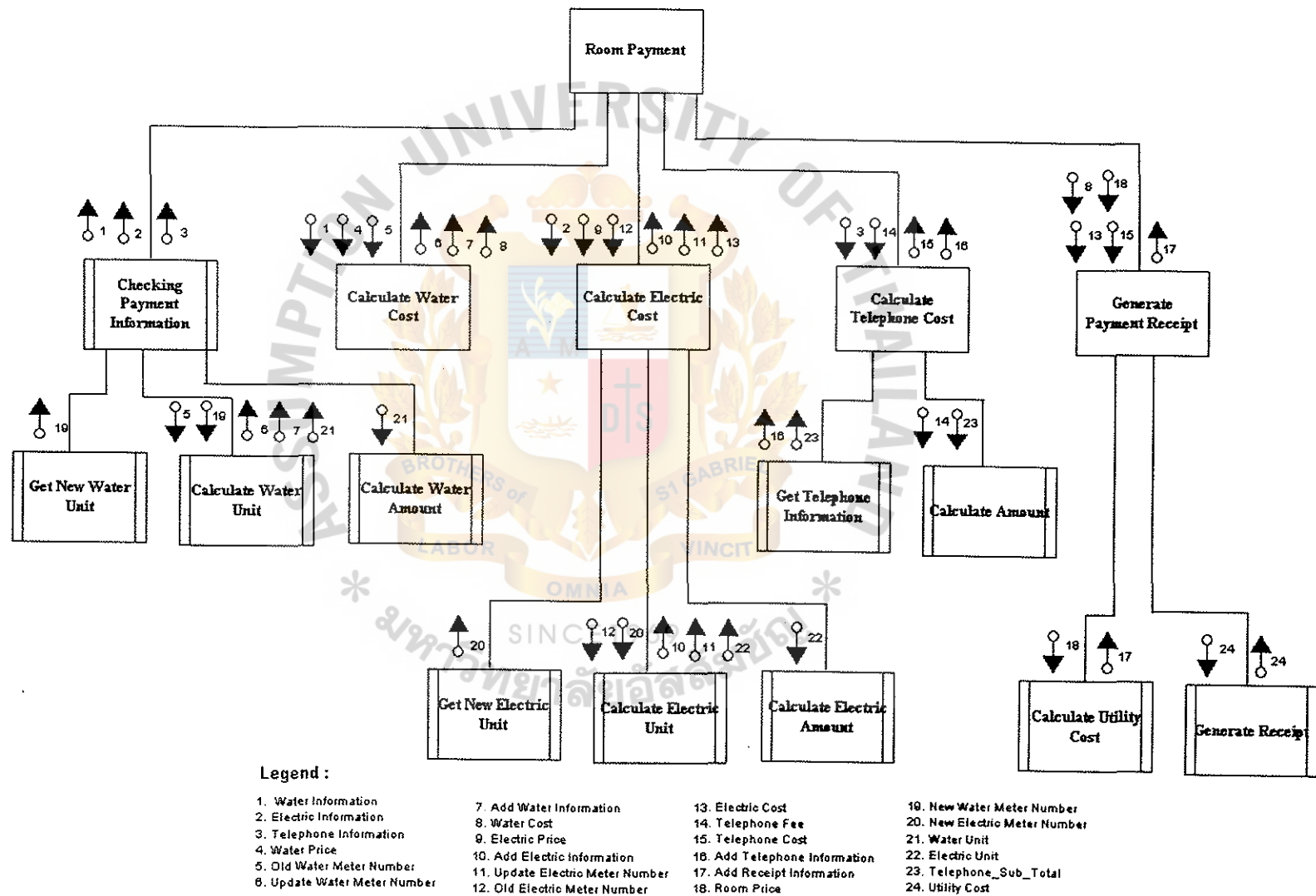


Figure E.7: Structure Chart of Generate Report

