



CHAPTER V

EVALUATION OF SYSTEM DESIGN

This chapter will perform evaluation of overall outputs – feasibility study, analysed information, improved primary design specification – from chapter 3.

5.1 Implementation of Primary System Design Specification

The system analysts will first analyse overall company background that will be gathered from system planning phase of SDLC – especially feasibility study – to list suitable criteria for technology selection and score them based on actual situation.

Then, the analysts will transfer primary design specification and provide brief training about company workflow to programmers. They will use design specification to design database and prototype based on the selected technology platform.

5.1.1 Technology Selection

The suitable technology is the first step of development phase because all of the following design and implementation must be based on the selected technology. The suitable technology is not the best one in but good enough one that enable features that the company required.

In this case, the prioritising approach is used for selecting suitable technology. The prioritising is a very simple method that is performed by ranking '1' to the most match technology, and '2' for the second match sequentially. After summarising the score of each technology, the lowest one – the most selected for most-match technology – will win the selection and is used as the information system platform.

The conditions that are used for ranking are divided to 3 parts.

- **Executives;** focus on big picture about budget, acquisition, and utilisation.
- **IT;** focus on technical issues, flexibility, availability, and performance.
- **User;** focus on learning curve, response time, and ease of usage.

For technology, the analyst recommends some popular database systems and application platforms for ranking which can be summarised their features as below.

Table 5.1: Features of each database system

Database System
<ul style="list-style-type: none"> • <u>Microsoft Access</u> <ul style="list-style-type: none"> • Microsoft-based database system • Most people are familiar with • License cost about 10,000 bath (include MS access form) • Mainly used for small – medium level of data access • Easy to manage and use • Low resources consuming • Just enough features (that the company requires) • <u>Microsoft SQL Server</u> <ul style="list-style-type: none"> • Microsoft-based database system • Most people are familiar with • License cost >> 10,000 bath • Mainly used for medium-enterprise level of data access • Easy to manage / backup / restore • Easy to use • High stability • Medium to High resources consuming • Complete features (may be too much) • <u>Oracle</u> <ul style="list-style-type: none"> • Platform independent database system • License cost very expensive • Mainly used for enterprise level of data access • Very high stability • High resources consuming • Complete features (may be too much) • <u>Mv SOL</u> <ul style="list-style-type: none"> • Web-based database system • Free of charge • Mainly used for small – medium level of data access • Easy to manage / backup / restore • Easy to use • Low resources consuming • Just enough features (that the company requires)

Table 5.2: Features of each application platform

Application Platforms	
•	<p><u>Microsoft Access Form</u></p> <ul style="list-style-type: none"> • Microsoft-based software • Most people are familiar with • License cost about 10,000 bath • Mainly used for small – medium level company • Easy to develop and use • Low resources consuming • Just enough features (that the company requires)
•	<p><u>Visual Basic</u></p> <ul style="list-style-type: none"> • Microsoft-based software • Most people are familiar with • License cost >> 10,000 bath • Mainly used for medium-enterprise level company • Require programming skills to develop • Easy to use • Medium resources consuming • Complete features (may be too much)
•	<p><u>Java Application</u></p> <ul style="list-style-type: none"> • Operation Systems independent • Free of charge • Mainly used for low-medium level company • Require high programming skills to develop • Medium resources consuming • Complete features (may be too much)
•	<p><u>Java Server Pages</u></p> <ul style="list-style-type: none"> • Web-based application • Most people are familiar with • Free of charge • Mainly used for low-medium level company • Require programming skills to develop • Easy to use • Low resources consuming • Just enough features (that the company requires)

Technologies are ranked by executives, IT people, and users as table below.

Table 5.3: Scoring table of technology selection [Database Management System]

<i>Database Management System</i>	MS Access	MS SQL Server	Oracle	My SQL
<i>Acquisition (ranked by executives)</i>				
Software license cost	2	3	4	1
Ease of acquisition	2	3	4	1
Completeness of required functions	2	1	3	4
<i>Total score of Executives</i>	<i>6</i>	<i>7</i>	<i>11</i>	<i>6</i>
<i>Development and Maintenance (rated by IT)</i>				
Learning curve of development	1	2	4	3
Look and feel	2	1	4	3
Ease of manage data	1	3	4	2
Flexibility	3	2	4	1
Memory / CPU Usage	1	3	4	2
Upgrade availability	1	2	4	3
Difficulty of maintenance	1	2	4	3
<i>Total score of IT</i>	<i>10</i>	<i>15</i>	<i>22</i>	<i>17</i>
<i>Usage (rated by users)</i>				
Learning curve of usage	1	2	4	3
Easy-to-use User Interface	1	2	4	3
Response Speed	1	3	4	2
Database crash time	2	1	3	4
Memory / CPU Usage	1	3	4	2
<i>Total score of Users</i>	<i>6</i>	<i>11</i>	<i>19</i>	<i>14</i>
<i>Summary Score</i>	<i>22</i>	<i>33</i>	<i>58</i>	<i>37</i>

Table 5.4: Scoring table of technology selection [application platform]

<i>Application Language</i>	MS Access form	Visual Basic	Java application	Java Server Pages (JSP)
<i>Acquisition (ranked by executives)</i>				
Software license cost	3	4	1	2
Ease of acquisition	1	2	4	3
Completeness of required functions	2	1	3	4
<i>Total score of Executives</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<i>Development and Maintenance (rated by IT)</i>				
Learning curve of development	3	4	2	1
Look and feel	2	1	4	3
Ease of developing user interfaces	1	2	4	3
Flexibility	4	3	2	1
Memory / CPU Usage	1	4	3	2
Upgrade availability	1	2	3	4
Ease of maintenance	1	3	4	2
Matchable with database	1	2	3	4
<i>Total score of IT</i>	<i>14</i>	<i>21</i>	<i>25</i>	<i>20</i>
<i>Usage (rated by users)</i>				
Learning curve of usage	1	2	3	4

Easy-to-use User Interface	1	2	3	4
Response Speed	2	1	4	3
System failure time	1	2	3	4
Memory / CPU Usage	1	2	4	3
<i>Total score of Users</i>	<i>6</i>	<i>9</i>	<i>17</i>	<i>18</i>
Summary Score	26	37	50	47

Reasons for selection in each condition are listed below.

5.1.1.1 Acquisition

- **Rated by executives**
- **Software license cost** = My SQL / Java Application

Though the current computer system is PC-IBM based which most users are familiar with, open-sources software which is free of charge is most attractive for executives who pay for the technology.

- **Ease of acquisition** = My SQL / Microsoft Access Form

Though Java application and JSP can be downloaded from website like My SQL, to acquire complete features java needs many libraries to be plugged-in which not convenient. So Microsoft Access is the easiest way.

- **Completeness of required functions** = Microsoft SQL Server / Visual Basic

These technologies provide complete features but the license prices are very expensive.

5.1.1.2 Development and Maintenance

- **Rated by IT**
- **Learning curve of development** = Microsoft Access / JSP

JSP is the winner in lowest learning curve because it is a web-based technology. And MS Access is the winner for its drag-and drop behaviour.

- **Look and feel** = MS SQL Server / Visual Basic

Their themes are windows-based which most familiar for users.

- **Ease of developing / manage** = Microsoft Access / Microsoft Access Form

Drag and Drop techniques is what IT staffs prefer.

- **Flexibility** = My SQL / JSP

Web-based technologies are the winner in this condition.

- **Memory / CPU Usage** = Microsoft Access / Microsoft Access Form
Microsoft access is a small-scale software which can run standalone.
- **Upgrade availability** = Microsoft Access / Microsoft Access Form
Licensed software always provides upgrade features.
- **Ease of maintenance** = Microsoft Access / Microsoft Access Form
Microsoft software is easy to install, backup, restore, etc.

5.1.1.3 Usage

- **Rated by user**
- **Learning curve of usage** = Microsoft Access / Microsoft Access Form
Users of the company are already familiar with Microsoft software.
- **Easy-to-use User Interface** = Microsoft Access / Microsoft Access Form
Microsoft software is very easy to use.

- **Response Speed** = Microsoft Access / Visual Basic
Visual basic with good design may response quicker than Microsoft Access Form but Form is better matched with Microsoft Access database.

- **System failure / crash** = MS SQL Server / Microsoft Access Form
MS SQL server is more stable than Microsoft Access but it consumes more resources and is more difficult to implement.

- **Memory / CPU Usage** = Microsoft Access / Microsoft Access Form
Microsoft Access is a small-scale software that uses low resources.

The winner of this ranking is the one that gain lowest scores. So the selected technology is 'Microsoft Access Form and Database' because it is easy to develop, users are already familiar with its Windows-based user interfaces, small and quick response with low level of resources consumption, especially it can provide the required features that new system needs such as status tracking, to do list.

5.1.2 Sample Database Design

This sample database is designed based on functions of selected technology – Microsoft Access. Programmers start their design process by referencing all data from data access diagram, defining their meanings, and providing sample data.

This document is called 'Data Definition' which available in Appendix C. These fields are rearranged follow database normalisation standard and actual company behaviors. The list below are some points to concern in database design.

- Long text of properties detail should be split into many fields in order to easier access and perform some automation calculation.
- One job may have many set of properties, so property detail should be separated from job detail.
- All constant values – such as property type, document type, valuation purpose, valuation approach, etc. – in the system must be managed in admin database, not embedded in programming code to handle policy changes from organisation which controls valuation standard; VAT, SEC.

Figure below is a sample design database of the valuation company.

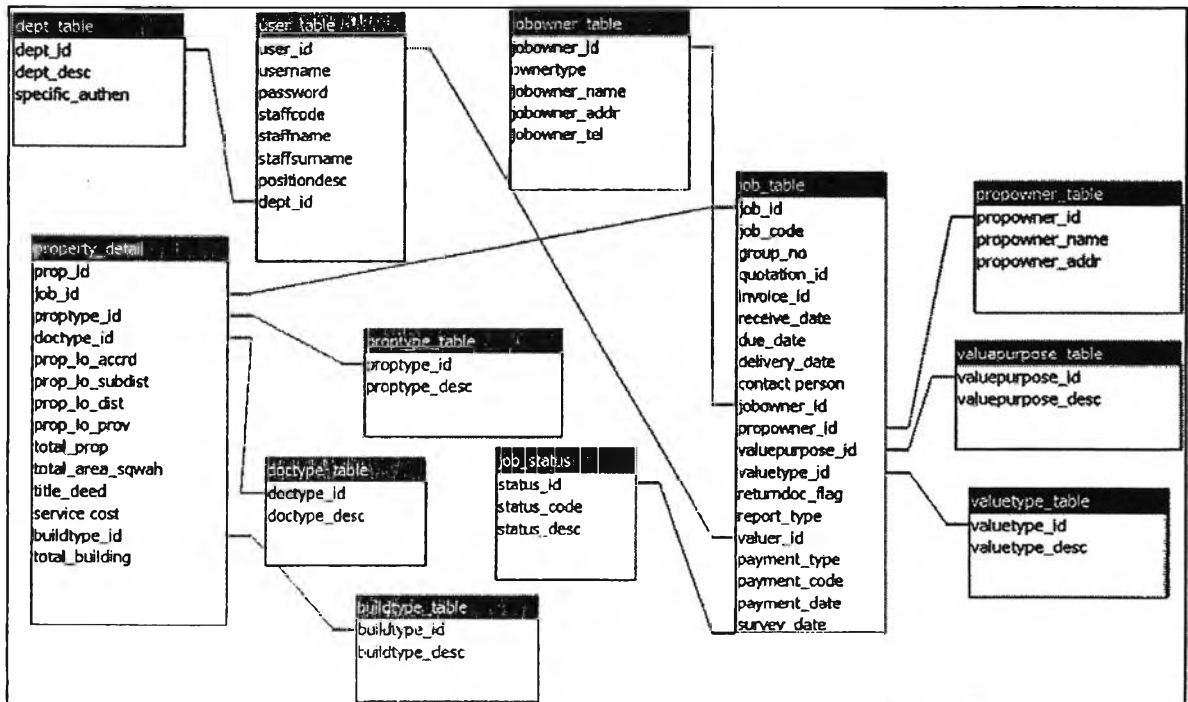


Figure 5.1: Sample Database Design based on primary design specification

5.1.3 Sample Prototype

Sample prototypes below are designed based on look and feel of the selected technology – Microsoft Access Form. The structure of user interfaces are arranged follows sketched screen that users confirm to use.

- Insert / Update screens are based on internal documents.
- Data retrieval screens required field that displays in search results are dependent on objectives of each page and users’ opinions.

Figure below are sample users interfaces that users commit to use.

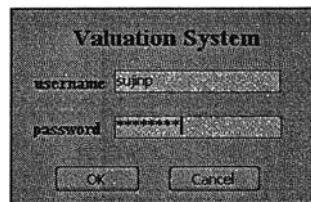


Figure 5.2: Login screen

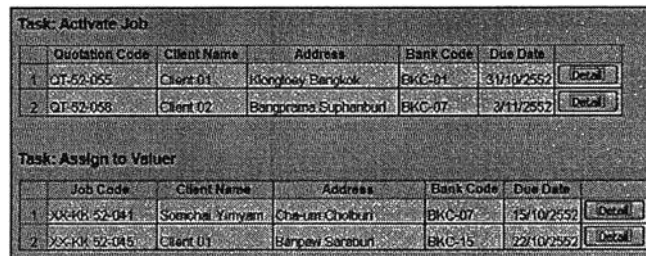


Figure 5.3: To do list screen

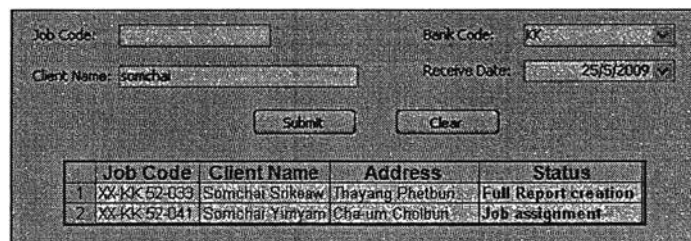


Figure 5.4: Status tracking screen

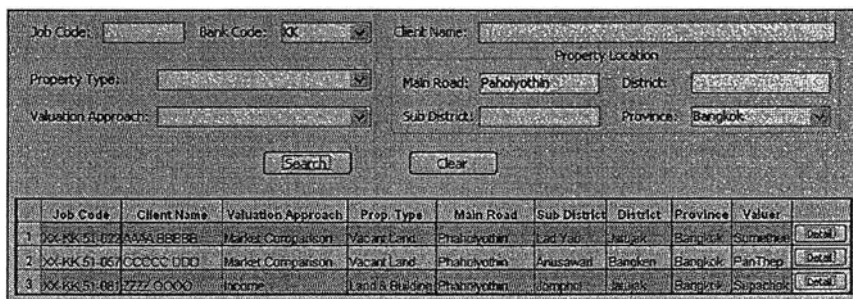


Figure 5.5: Job Search screen

5.2 Feedbacks from Programmers' about the design

Programmers will first receive the 'who / what / where / when / how and why' document and improved basic flowchart to begin studying business logic of the company. The system analyst also provides brief training about the company structure and culture, working styles, standard documents that are currently used, and guidelines for design and development of satisfied information system.

Then, the analyst transfers all analysed information and diagrams to programmers and continuously monitors their progress.

5.2.1 Completeness of design specification and analysed information

The 'who / what / where / when / how and why' document and basic flowchart are very effective starting point that allow programmers understand overall processes of the company. Then, programmers can study activity diagrams and sequence diagrams to get deeper ideas about steps of work. IDEFØ and DFD allow them to get complete logic within each step of work.

Feasibility studies are good guidelines for programmers to know the company's culture, people attitudes and it very useful when they need more detail of work. They will know who or where and how to get required information.

However, the different diagrams are focusing on different points of view. So each diagram may not fully be used while design and develop phase. It is better for them to use all diagrams together to fulfil understanding about complete business processes of the company and lead to a system that fully satisfy all users.

5.2.2 Usability of diagrams in design and development phase

Standard diagrams that illustrate company's business processes can benefit for programmers in design and development phase as summarised in table below.

Table 5.5: Benefits of standard diagrams in design and development phase

Standard Diagram	Benefits
Basic flowchart	<ul style="list-style-type: none"> ▪ Used as fundamental flow because it is easy to understand ▪ Provide overall tasks and step of work
Use Case Diagram	<ul style="list-style-type: none"> ▪ The most important resources while design and develop authentication module ▪ It summarised all activities in the system and specify which roles can or can not do.
State Chart Diagram	<ul style="list-style-type: none"> ▪ A powerful resources for design and develop status tracking feature. ▪ It lists all states in the system and provides input / output criteria.
Sequence Diagram	<ul style="list-style-type: none"> ▪ Encourage programmers to correctly control work flows
Activity Diagram	<ul style="list-style-type: none"> ▪ Standard UML version of basic flowchart ▪ Clarify some ambiguous points in basic flowchart by standard UML diagram which easily understandable by IT-related people
IDEFØ and DFD	<ul style="list-style-type: none"> ▪ Provide specific details in depth into logical flow level ▪ Explain both function – by IDEFØ – and data – DFD
Data Definition	<ul style="list-style-type: none"> ▪ Used for analyse data and normalisation in database design and access phase. ▪ Easy to understand and transform to ‘data dictionary’ when the project is finished.

5.3 Sample test cases

Definition of test case from IEEE Standard 610 (1990) is

“A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance” (Kaner, 2003: 1)

There are many templates of test case available in website and books but the selected template – from Carleton University – mainly describes test scenario of various situation of business processes that is suitable for this research. In this sample test, the analyst uses template of test script detail from below.

Table 5.6: Test Script Detail

Test Script #	Scenario	Expected Results	Actual Results – if different - and/or Comments
1	Build a Full TA over one term (Fall) for one year	Error generated.	

(Carleton University, 2007: online)

The analyst customises the template a bit for more matches with the company.

Table 5.7: The customised test script

Test Case	Scenario	Actual Results (Old system – manual processes)	Expected Results (New system -- Information System implemented)
1	Scenerio#1		.
2			

The customised template will be used to compare the old processes with the new information system that being developed.

5.3.1 Normal case

The normal cases that the analyst selects to test are 2 problems that force the company to acquire information system – slow status response, and duplicate data.

5.3.1.1 Status tracking

Table 5.8: Test case of status tracking

Test Case	Scenario	Actual Results (Old system – manual processes)	Expected Results (New system -- Information System implemented)
1	Client asks for progress of their jobs Using Job Code.	<ul style="list-style-type: none"> ▪ Look up in Job List document to find detail of the selected Job Code ▪ Examine that which valuer is responsible for this job ▪ Go or Call him and ask about Job Progressive ▪ Call back to client to answer job status 	<ul style="list-style-type: none"> ▪ Open the information system ▪ Access 'Status Tracking' module ▪ Query job by Job Code ▪ View status of job in result pane ▪ Answer status of selected job to client.
2	Client asks for progress, of their jobs Using other detail of jobs	<ul style="list-style-type: none"> ▪ Ask for more detail about client's name ▪ Look up in Job List to find matched name ▪ If duplicate name, ask for more detail from client ▪ Ask valuer who response to the selected Job List about status of job ▪ Call back to client to answer job status 	<ul style="list-style-type: none"> ▪ Open the information system ▪ Access 'Status Tracking' module ▪ Ask client about other information of job ▪ Query job from this information ▪ Found matched job ▪ Answer status of selected job to client.

5.3.1.2 Key-in data

Table 5.9: Test case of key-in data

Test Case	Scenario	Actual Results (Old system – manual processes)	Expected Results (New system -- Information System implemented)
1	New job received	<ul style="list-style-type: none"> ▪ Marketing key-in data into Job List when payment processes of job are approved ▪ Marketing update valuer into Job List when job is assigned ▪ HR key-in all data into Job Summary when job is closed. 	<ul style="list-style-type: none"> ▪ Marketing key-in data into Job Database when receiving request from client ▪ Marketing update payment detail ▪ Marketing update valuer ▪ Valuer update status of job ▪ HR updates valuation detail when job is closed.

5.3.2 Unexpected case

Unexpected cases that the analyst selects are 2 high probability issues that may occur – input incorrect data, and add more properties to appraise.

5.3.2.1 Incorrect data input

Table 5.10: Test case of incorrect client data

Test Case	Scenario	Actual Results (Old system – manual processes)	Expected Results (New system -- Information System implemented)
1	Marketing key-in incorrect clients' data	<ul style="list-style-type: none"> ▪ Marketing correct data in Job List ▪ Valuer and other departments may use the wrong data if marketing do not notify all staff. 	<ul style="list-style-type: none"> ▪ Marketing correct data in Job Database ▪ Valuer and other departments can automatically access the correct data

5.3.2.2 New properties after job opened

Table 5.11: Test case of new properties after job opened

Test Case	Scenario	Actual Results (Old system – manual processes)	Expected Results (New system -- Information System implemented)
1	New properties are added to job after job opened.	<ul style="list-style-type: none"> ▪ Marketing add properties to Job List ▪ Valuer and other departments may use the wrong data if marketing do not notify all staff. 	<ul style="list-style-type: none"> ▪ Marketing add new properties into Job database ▪ Valuer and other departments can automatically access the updated data