

# **Digital Policy Office**

## **EFFECTIVE SYSTEMS ANALYSIS AND DESIGN GUIDE**

### **APPENDIX A**

#### **TOOLS**

##### **[G61a]**

Version: 1.2

**July 2024**

© The Government of the Hong Kong Special Administrative Region  
of the People's Republic of China

The contents of this document remain the property of, and may not be reproduced in whole or in part without the expressed permission of the Government of the Hong Kong Special Administrative Region of the People's Republic of China

Amendment History				
Change Number	Revision Description	Section Affected	Rev. Number	Date
1	As detailed in 1.01 to 1.03		1.1	December 2016
1.01	Rename “Current Business Model” to “Current Business Process	3.2, 4.2, 5.2		
1.02	Add the State Diagram in “Alternative / Complementary Tools” column of relevant sections of “Area of Use”  Add new tool section “State Diagram”	3.2, 4.2, 5.2, 6.2  27 ( <i>New</i> )		
1.03	Add the “Use Case Specification” in the section of “Use Case Diagram”	4.1(g), 4.2.2 <i>(New)</i>		
2	Update Documents for the Establishment of Digital Policy Office (DPO)	Cover page	1.2	July 2024

## Table of Contents

<b>OVERVIEW .....</b>	<b>1</b>
<b>TOOLS.....</b>	<b>2</b>
<b>1 TOOL 1 – INFORMATION COLLECTION TECHNIQUES.....</b>	<b>2</b>
1.1 Purpose & Description .....	2
1.2 Areas of Use.....	3
<b>2 TOOL 2 – PROTOTYPING TECHNIQUES.....</b>	<b>4</b>
2.1 Purpose & Description .....	4
2.2 Areas of Use.....	4
2.2.1 Example (1): Wireframe.....	5
2.2.2 Example (2): Screen mock-up.....	6
2.2.3 Example (3): Prototype .....	6
<b>3 TOOL 3 – FLOWCHART DIAGRAM .....</b>	<b>7</b>
3.1 Purpose & Description .....	7
3.2 Areas of Use.....	7
3.2.1 Example (1): Flowchart Diagram .....	9
3.2.2 Example (2): Flowchart Diagram .....	9
<b>4 TOOL 4 – USE CASE DIAGRAM .....</b>	<b>10</b>
4.1 Purpose & Description .....	10
4.2 Areas of Use.....	11
4.2.1 Example: Use case diagram .....	11
4.2.2 Example: Use Case specification .....	12
<b>5 TOOL 5 – ACTIVITY DIAGRAM.....</b>	<b>14</b>
5.1 Purpose & Description .....	14
5.2 Areas of Use.....	14
5.2.1 Example: Activity diagram .....	16
<b>6 TOOL 6 – SEQUENCE DIAGRAM.....</b>	<b>17</b>
6.1 Purpose & Description .....	17
6.2 Areas of Use.....	17
6.2.1 Example (1): Business-Level Sequence Diagram.....	18
6.2.2 Example (2): System-Level Sequence Diagram .....	18
<b>7 TOOL 7 – USER STORY.....</b>	<b>19</b>
7.1 Purpose & Description .....	19
7.2 Areas of Use.....	19
7.2.1 Example: Borrow Book Story.....	19
<b>8 TOOL 8 – DECISION TABLE.....</b>	<b>20</b>
8.1 Purpose & Description .....	20
8.2 Areas of Use.....	20
8.2.1 Example: Decision Table .....	20
<b>9 TOOL 9 – DECISION TREE .....</b>	<b>21</b>
9.1 Purpose & Description .....	21

---

9.2 Areas of Use.....	21
9.2.1 Example: Decision Tree .....	22
<b>10 TOOL 10 – ACTION CHECKLIST .....</b>	<b>23</b>
10.1 Purpose & Description .....	23
10.2 Areas of Use .....	23
10.2.1 Example: Screenshot of the action checklist .....	23
<b>11 TOOL 11 – REQUIREMENTS TRACEABILITY MATRIX .....</b>	<b>24</b>
11.1 Purpose & Description .....	24
11.2 Areas of Use .....	24
11.2.1 Example: Requirements Traceability Matrix.....	24
<b>12 TOOL 12 – CONCEPTUAL DATA MODEL DIAGRAM.....</b>	<b>26</b>
12.1 Purpose & Description .....	26
12.2 Areas of Use .....	26
12.2.1 Example (1): Conceptual ER-diagram.....	27
12.2.2 Example (2): Conceptual class diagram .....	27
<b>13 TOOL 13 – DATA DISSEMINATION DIAGRAM.....</b>	<b>28</b>
13.1 Purpose & Description .....	28
13.2 Areas of Use .....	28
13.2.1 Example: Data dissemination diagram .....	29
<b>14 TOOL 14 – DATA MIGRATION DIAGRAM .....</b>	<b>30</b>
14.1 Purpose & Description .....	30
14.2 Areas of Use .....	30
14.2.1 Example (1): Data migration diagram .....	31
14.2.2 Example (2): Data migration diagram .....	31
<b>15 TOOL 15 – BLOCK DIAGRAM .....</b>	<b>32</b>
15.1 Purpose & Description .....	32
15.2 Areas of Use .....	32
15.2.1 Example (1): Block diagram (Application).....	33
15.2.2 Example (2): Block diagram (Integration) .....	33
<b>16 TOOL 16 – DEPLOYMENT DIAGRAM .....</b>	<b>34</b>
16.1 Purpose & Description .....	34
16.2 Areas of Use .....	34
16.2.1 Example: Deployment diagram .....	34
<b>17 TOOL 17 – APPLICATION COMMUNICATION DIAGRAM.....</b>	<b>35</b>
17.1 Purpose & Description .....	35
17.2 Areas of Use .....	35
17.2.1 Example: Application Communication Diagram.....	35
<b>18 TOOL 18 – ENTITY RELATIONSHIP DIAGRAM.....</b>	<b>36</b>
18.1 Purpose & Description .....	36
18.2 Areas of Use .....	36
18.2.1 Example (1): Logical ER diagram.....	36
18.2.2 Example (2): Physical ER diagram.....	36
<b>19 TOOL 19 – CLASS DIAGRAM.....</b>	<b>38</b>

---

19.1 Purpose & Description .....	38
19.1.1 Areas of Use .....	38
19.1.2 Example: Class diagram .....	39
<b>20 TOOL 20 – SECURITY CONTROLS IMPLEMENTATION OVERVIEW DIAGRAM .....</b>	<b>40</b>
20.1 Purpose & Description .....	40
20.2 Areas of Use .....	40
20.2.1 Example: Security controls implementation overview diagram.....	41
<b>21 TOOL 21 – NETWORK DIAGRAM.....</b>	<b>42</b>
21.1 Purpose & Description .....	42
21.2 Areas of Use .....	42
21.2.1 Example: Network diagram .....	42
<b>22 TOOL 22 – DATA STORAGE DIAGRAM.....</b>	<b>43</b>
22.1 Purpose & Description .....	43
22.2 Areas of Use .....	43
22.2.1 Example (1): Data storage diagram .....	44
22.2.2 Example (2): Data storage diagram .....	44
<b>23 TOOL 23 – SERVER DIAGRAM.....</b>	<b>45</b>
23.1 Purpose & Description .....	45
23.2 Areas of Use .....	45
23.2.1 Example: Server diagram .....	46
<b>24 TOOL 24 – COST / BENEFIT EVALUATION TECHNIQUE.....</b>	<b>47</b>
24.1 Purpose & Description .....	47
24.2 Areas of Use .....	47
24.2.1 Example: Cost / benefit evaluation.....	48
<b>25 TOOL 25 – GANTT CHART .....</b>	<b>49</b>
25.1 Purpose & Description .....	49
25.2 Areas of Use .....	49
25.2.1 Example: Gantt chart .....	50
<b>26 TOOL 26 – DECISION CHECKLIST .....</b>	<b>51</b>
26.1 Purpose & Description .....	51
26.2 Areas of Use .....	51
26.2.1 Example: Decision checklist.....	51
<b>27 TOOL 27 – STATE DIAGRAM .....</b>	<b>52</b>
27.1 Purpose & Description .....	52
27.2 Areas of Use .....	52
27.2.1 Example: State Diagram for Membership Application .....	53

## OVERVIEW

- (a) This Appendix provides supplementary materials, including tools, techniques, checklists and diagrams that are referred to in the Guide. It aims to facilitate readers in effectively performing SA&D activities and developing the SA&D Report.
- (b) A brief description will be provided for each tool, followed by purpose and when to use. Where appropriate, examples will be provided to better illustrate how individual tools can be used.
- (c) The Appendix will guide the readers on which tools and techniques should be applied in different situations, and is not intended for textbook-level of details on the syntax and notation of each tool.
- (d) Project teams should flexibly select the right tools in accordance with their needs, skills and experiences.

# TOOLS

## 1 TOOL 1 – INFORMATION COLLECTION TECHNIQUES

### 1.1 PURPOSE & DESCRIPTION

In addition to conducting interviews, workshops and surveys, three other approaches are described below which can be used to help identify issues in the current environment, identify business process flows and elicit requirements.

#### i) Focus Group

- This technique is suitable for use when the project team has little information about the situation and would like to get insights from a large group of audience. It helps to acquire stakeholders' input and requirements on a system or an aspect of the system. Members of the focus group can exchange their ideas with each other, while the conversation is directed by a moderator.

#### ii) Brainstorming

- Brainstorming is best used when the project team needs to generate as many ideas as possible for the situation. It involves a creative thinking process among a group of people brought together to solve a given problem. It works by focusing on one problem at a time, and then deliberately coming up with as many solutions as possible and pushing the ideas as far as possible.

#### iii) Storyboarding

- This technique is suitable for use when the project team wants to illustrate or capture complex process flows. A storyboard is a set of sketches depicting user activities that occur in an existing or envisioned system. It is a kind of paper prototyping where the project team and the stakeholders start by drawing pictures of the screens, dialogs, toolbars, and other elements they expect that the system should provide. The approach with storyboarding can continue to evolve until requirements and details are worked out and aligned.

## 1.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Current Environment Description</b> <ul style="list-style-type: none"> <li>• Information Gathering</li> </ul>	i) The information collection techniques can be used to help obtain better understanding of the current environment and identify current issues from stakeholders.	<ul style="list-style-type: none"> <li>• Nil</li> </ul>
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>• Define Future Business Process</li> </ul>	i) The information collection techniques can be used to define future business processes and can be used to capture the functional requirements, business rules, and reassure project scope, etc.	<ul style="list-style-type: none"> <li>• Nil</li> </ul>
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>• Elicit Functional Requirements</li> </ul>		<ul style="list-style-type: none"> <li>• Prototyping Techniques</li> </ul>



## 2 TOOL 2 – PROTOTYPING TECHNIQUES

### 2.1 PURPOSE & DESCRIPTION

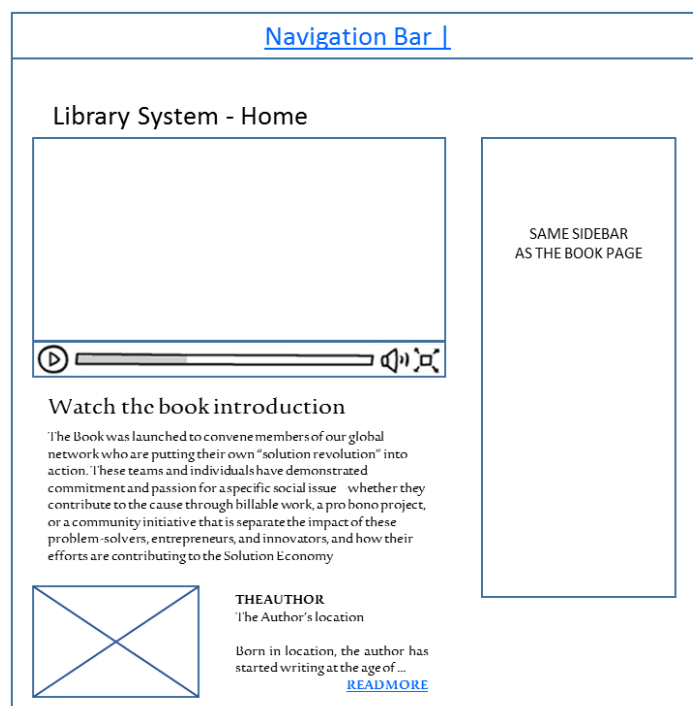
- (a) Prototyping techniques are used to facilitate requirements elicitation. A prototype can demonstrate which aspects of the required system are effectively useful and applicable, and those that are not; it can also demonstrate the look and feel of the user interface (colours, layout, graphics, controls), and the information architecture (navigation structure) of the required system.
- (b) At the beginning, the user should focus on broad requirements and workflow issues without being distracted by the precise appearance of screen elements such as their position, fonts, and colours etc. Specifics of user interface design will be explored after requirements are clarified and general structure of the interface is determined in the Functional Specification.

### 2.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Requirements specification</b> User Requirements Document <ul style="list-style-type: none"> <li>Elicit Functional Requirements</li> </ul>	i) Prototyping can be used to facilitate business users to visualise their needs and any acceptance criteria identified.	<ul style="list-style-type: none"> <li>Information Collection Techniques</li> </ul>
<b>System Specification</b> Functional Specification <ul style="list-style-type: none"> <li>Facilitate Design Sessions</li> </ul>	i) Prototyping can be used to facilitate design sessions/workshops with key business users and project team. Wireframes, screen mock-ups, as well as behavioural prototypes are developed to illustrate how business requirements/rules/flows will be implemented in the systems. ii) Wireframes act as a visual guide that represents the skeletal framework of a website. iii) Screen mock-ups will look like the final design, but will not perform actual function beyond what the user sees. iv) A prototype will also look like the final design, and in addition, simulate user interface	<ul style="list-style-type: none"> <li>Nil</li> </ul>

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
	interactions, allowing the user to experience the work flow and the main interactions.	
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>• Design User Experience</li> </ul>	i) Prototyping can be used to help finalise visual designs, by gathering feedback from stakeholders and updating the screen/report/form mock-ups accordingly.	<ul style="list-style-type: none"> <li>• Nil</li> </ul>

### 2.2.1 Example (1): Wireframe



### 2.2.2 Example (2): Screen mock-up

**Library Booking System – Member Portal**

Search ▾

- ▶ [Service Request](#)
- ▶ [Service Request](#)
- ▶ [Service Request](#)

**Contacts**

Contact Details

*Last Name	Date of Birth	Country	Address Line 1
<input type="text" value="Chan"/>	<input type="text" value="11/04/1966"/>	<input type="text" value="Hong Kong"/>	<input type="text" value="12/F"/>
First Name	HK ID	Phone Number	Address Line 2
<input type="text" value="James"/>	<input type="text" value="A871432(1)"/>	<input type="text" value="9621 7601"/>	<input type="text" value="Taikoo Shing"/>
Mr/Ms			Address Line 3
<input type="text" value="Mr."/>			<input type="text" value="Hong Kong"/>
			Address Line 4
			<input type="text"/>

**Contacts**

Date and Time	Type (Initial Registration/Login)
01/01/2012 11:25:04 PM	Initial Registration
10/09/2011 12:42:00 AM	Login
10/11/2013 09:54:01 PM	Login

Previous | Next

### 2.2.3 Example (3): Prototype

6

### 3 TOOL 3 – FLOWCHART DIAGRAM

#### 3.1 PURPOSE & DESCRIPTION

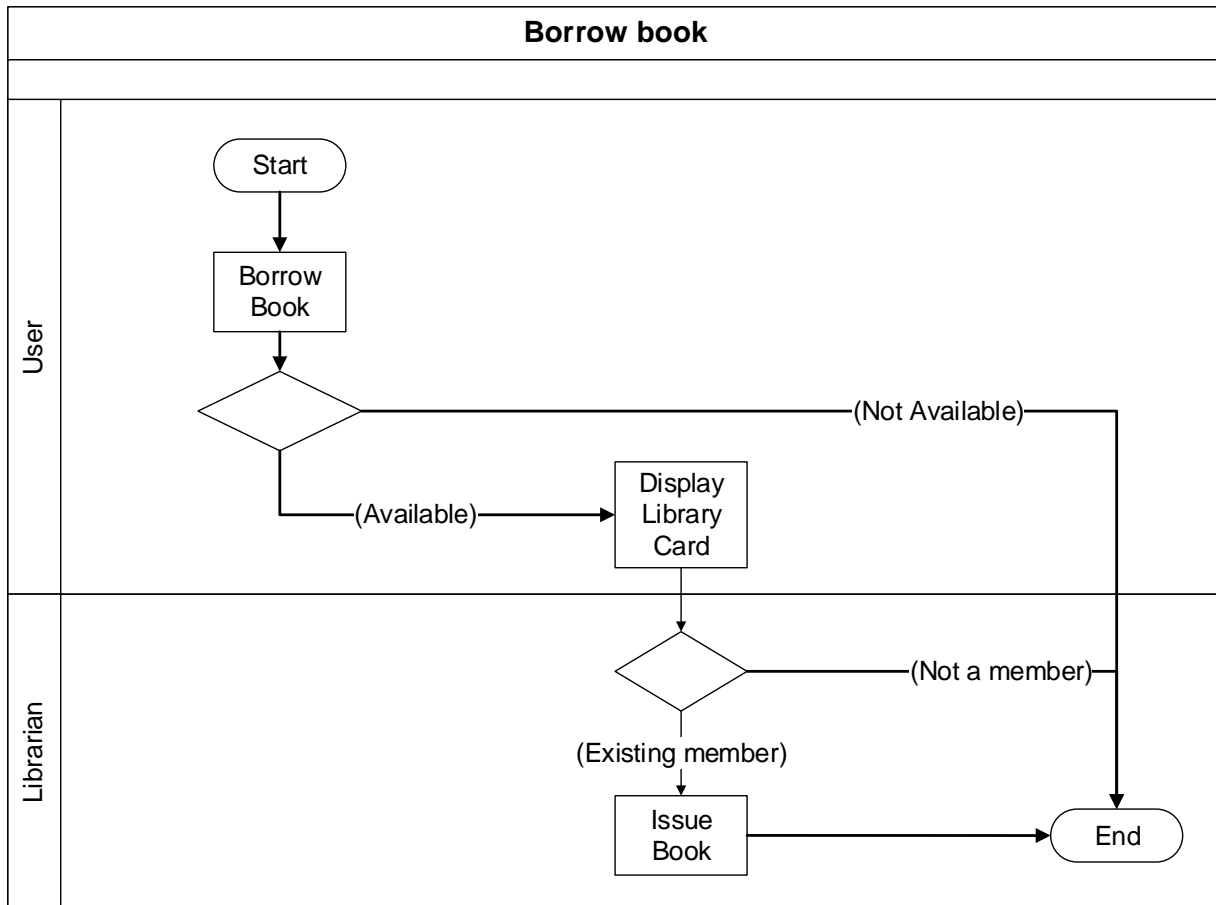
- (a) A flowchart diagram helps to present processes visually. It can be used at both the requirements phase (CED and requirements specification) and analysis and design phase (application architecture).
- (b) Flowchart diagram is used to map the business process flow, and help spot processing gaps and inefficiencies, especially the high risk interconnections between business units/functions (e.g. bottlenecks, redundancies and other causes of inefficiency). It involves mapping processes linearly as a series of tasks across a page. It contains horizontal rows across the page, allowing the audience to quickly plot and trace processes and the interconnections between processes and business units/functions. Flowchart diagram can be used during architecture design, for example, to illustrate the flow of data across data stores or the interfaces between high level components.

#### 3.2 AREAS OF USE

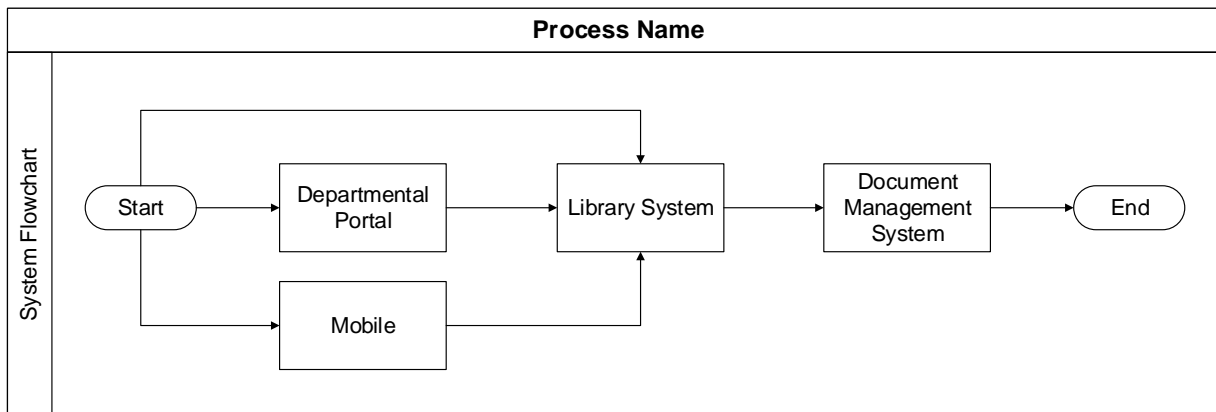
Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Current Environment Description</b> Produce Current System Description <ul style="list-style-type: none"> <li>Produce Current Business Process</li> </ul>	i) Flowchart diagram can highlight and model the major current business processes and help the project team understand how the organisation functions. ii) The Current Business Process can be broken down by major business process, and modelled using Flowchart Diagram.	<ul style="list-style-type: none"> <li>Use Case Diagram <i>and/or</i> Activity Diagram</li> </ul>
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>Define Future Business Process</li> </ul>	i) Flowchart diagram can be used to help model high-level or low-level business processes/workflow and visualise the future processes and user interactions in the system.	<ul style="list-style-type: none"> <li>Use Case Diagram <i>and/or</i> Activity Diagram <i>and/or</i> Sequence Diagram <i>and/or</i> State Diagram</li> </ul>

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>Requirements Specification</b></p> <p>User Requirements Document</p> <ul style="list-style-type: none"> <li>Elicit Functional Requirements</li> </ul>		<ul style="list-style-type: none"> <li>Use Case Diagram <i>and/or</i> Activity Diagram <i>and/or</i> User Story</li> </ul>
<p><b>System Specification</b></p> <p>Architecture Design</p> <ul style="list-style-type: none"> <li>Develop Application Architecture</li> </ul>	<p>i) Flowchart diagram can be used to illustrate the different layers in the application and their relationships or information flow between systems/components.</p>	<ul style="list-style-type: none"> <li>Block Diagram <i>and/or</i> Deployment Diagram</li> </ul>

### 3.2.1 Example (1): Flowchart Diagram



### 3.2.2 Example (2): Flowchart Diagram



## 4 TOOL 4 – USE CASE DIAGRAM

### 4.1 PURPOSE & DESCRIPTION

- (a) Use case diagrams in the Unified Modeling Language (UML)<sup>1</sup> are used when textual description of requirements may not be sufficient in illustrating the interactions between the future system and users, as well as other systems. Often, corporations use them as the standard way of documenting all types of system requirements. Use case diagrams can be used at the requirements phase.
- (b) It shows how the system will be used by someone external to the system. That someone may be a person, an organisation, a different system and even a more abstract concept such as time. One guideline is that the system cannot start its own use case, and that it must always be externally initiated.
- (c) General statements of user goals or business tasks that users need to perform are use cases; a single specific path through a use case is a usage scenario. Use case diagrams summarise all use cases together in one picture and shows the associations between actors and use cases and boundary of the system.
- (d) Project team need to work with the user to generalise specific scenarios into more abstract use cases, and also gather use cases by asking users to describe their business workflow. Alternatively, the project team may ask users to state the goals they have in mind when they sit down to work with the system. A user who says, I need to "do something" is probably describing a use case.
- (e) Use case diagram can also be leveraged for developing application architecture. It displays the relationships between consumers and providers of application services. It provides additional information on application functionality by illustrating how and when that functionality is used. As the architecture progresses, the use case diagram can evolve from functional information to include system design details.
- (f) It shows the different views of information from activity diagram and flowchart diagram. It presents the information of smaller work in bulk within the larger activity but not the flow of activities.
- (g) Each use case in the use case diagram could be further developed to form use case specification which describes the use case in further details in Functional Requirements of Requirements Specification and Function Definition of System Specification.

---

<sup>1</sup> The Unified Modeling Language (UML) is a standard adopted by the Object Management Group (OMG) and has been accepted by the International Organization for Standardization (ISO) as an approved ISO standard. More information can be found at [www.uml.org](http://www.uml.org)

## 4.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>Current Environment Description</b></p> <p>Produce Current System Description</p> <ul style="list-style-type: none"> <li>● Produce Current Business Process</li> </ul>	<p>i) Use case diagram can be used to explain the system's objective(s) and major functions.</p> <p>ii) Use case diagram can also be used as a starting point to identify stakeholders and existing systems context.</p>	<ul style="list-style-type: none"> <li>● Flowchart Diagram <i>and/or</i> Activity Diagram</li> </ul>
<p><b>Requirements Specification</b></p> <p>User Requirements Document</p> <ul style="list-style-type: none"> <li>● Define Future Business Process</li> <li>● Elicit Functional Requirements</li> </ul>	<p>i) Use case diagram can be used to model the initial set of functional requirements and boundary of the systems. It can also be used as a starting point to identify key major business processes.</p>	<ul style="list-style-type: none"> <li>● Flowchart Diagram <i>and/or</i> Activity Diagram <i>and/or</i> Sequence Diagram <i>and/or</i> State Diagram</li> </ul>

### 4.2.1 Example: Use case diagram

Initial system requirements:

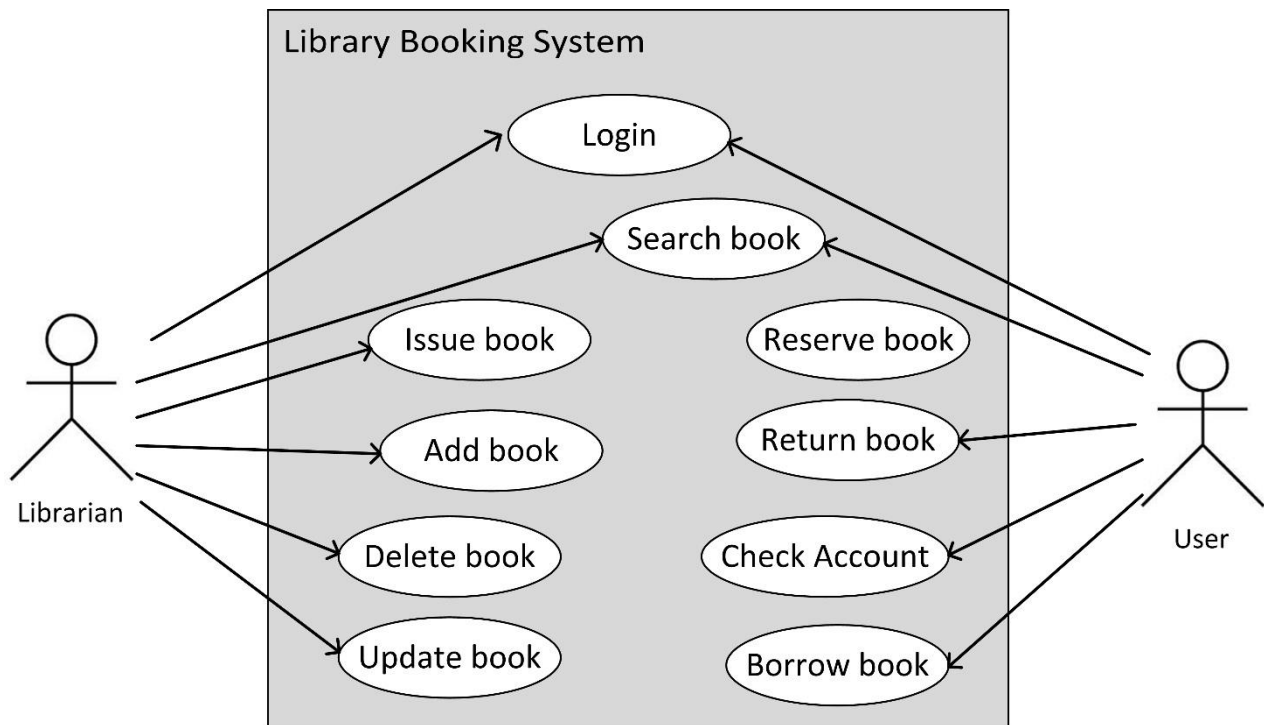
“Users need to login into the Library Booking System.”

“Librarian needs to check and update the database by adding a book”

“Users need to provide feedback such as comments or rating for the book”

“Users need to search, reserve and borrow book from the system”





### 4.2.2 Example: Use Case specification

<b>Use Case ID:</b>	<i>UC01</i>		
<b>Use Case Name:</b>	<i>Login</i>	<b>Version No:</b>	<i>01.002</i>
<b>Purpose:</b>	<i>User/Librarian login the system</i>		
<b>Last Update by:</b>	<i>Hamilton</i>	<b>On (date):</b>	<i>04 April 2016</i>
<b>Approved by:</b>	<i>Vettel</i>	<b>On (date):</b>	<i>05 April 2016</i>
<b>User/Actor:</b>	<i>User/Librarian</i>		
<b>Business Owner Name:</b>	<i>Alonso</i>		
<b>Trigger:</b>	<i>User or Librarian access the system protected login page</i>		
<b>References:</b>	<i>N/A</i>		
<b>Frequency of Use:</b>	<i>Daily</i>		
<b>Volume of Use</b>	<i>N/A</i>		
<b>Preconditions:</b>	<i>N/A</i>		
<b>Post Conditions:</b>	<i>N/A</i>		
<b>Non-functional Requirements</b>	<i>N/A</i>		
<b>Assumptions:</b>	<i>N/A</i>		
<b>Business Rules:</b>	<i>N/A</i>		
<b>Main Flow 1</b>			
<b>Step</b>	<b>Actor</b>	<b>Description</b>	<b>Alternate/Exception Flow</b>
<b>1</b>	<i>User/Librarian</i>	<i>enters the ID and password</i>	

2		<i>Process the authentication: if the ID and password are correct, the system authenticates the user and displays the relevant functions page. The use case ends</i>	
3		<i>if any or both of ID and password is/are invalid, the system directs user/librarian to alternate flow #1.1</i>	#1.1
<b>Alternate/Exception Flow</b>	<i>Alternate Flow#1.1</i>		
<b>Step</b>	<b>Actor</b>	<b>Description</b>	<b>Alternate /Exception Flow</b>
1		<i>The system prompts the error message. The use case ends</i>	

## 5 TOOL 5 – ACTIVITY DIAGRAM

### 5.1 PURPOSE & DESCRIPTION

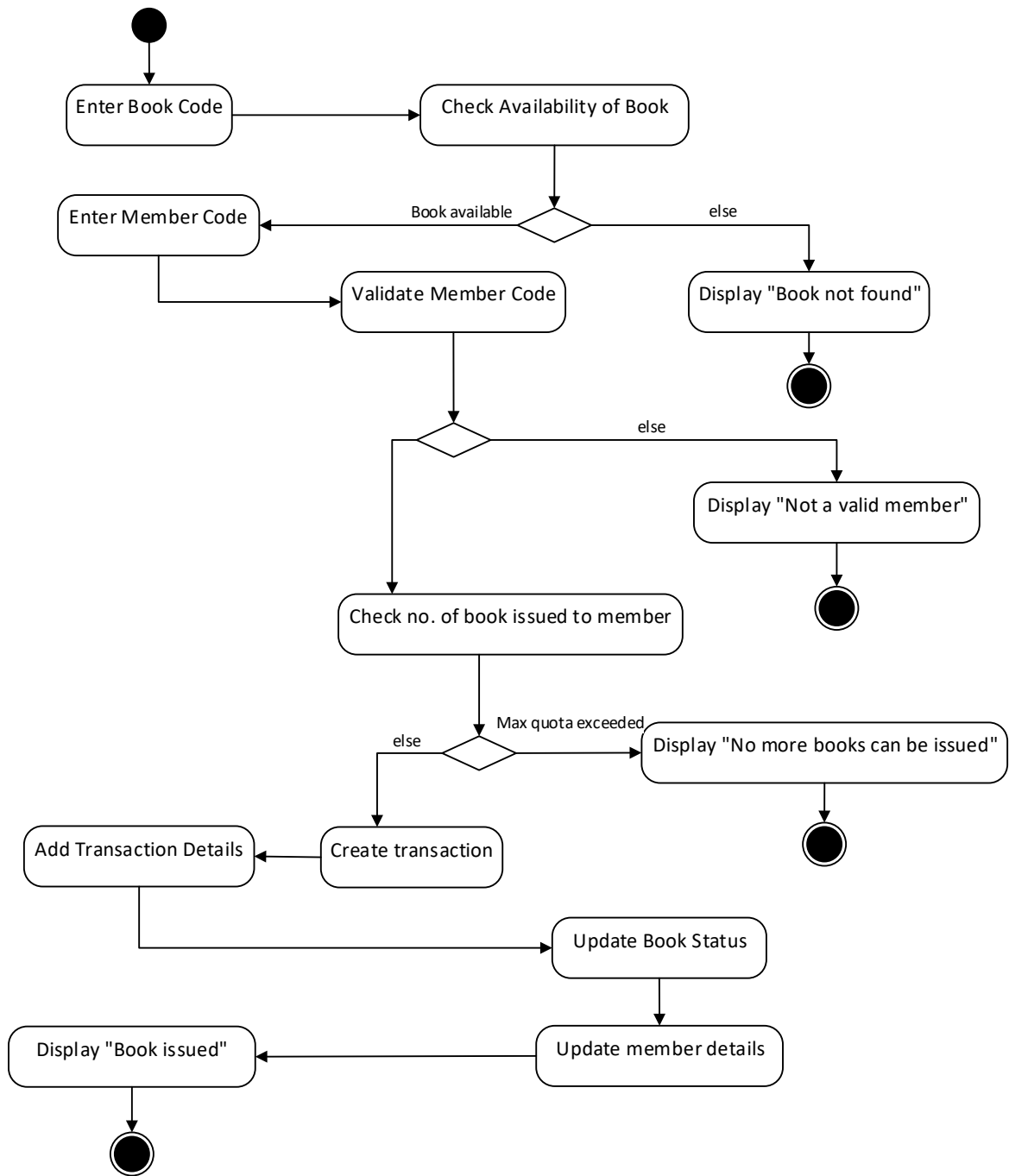
- (a) An activity diagram in the UML as an alternative to the flowchart diagram, is used to capture the dynamic behaviour of the system. Activity diagrams are often recommended for use if the project team wants to illustrate and capture operational flow or complex interactions between users and components of the future system.
- (b) The principal difference between activity diagrams and flowchart diagrams is that activity diagram supports parallel behaviour, as they can be used to detail situations where parallel processing may occur in the execution of some activities.

### 5.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Current Environment Description</b> Produce Current System Description <ul style="list-style-type: none"> <li>Produce Current Business Process</li> </ul>	i) Activity diagram can help describe key business logic of the current environment. ii) Activity diagram can also be used to model the current business process. The current business process can be broken down by major business process, and modelled using activity diagrams.	<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i> Use Case Diagram</li> </ul>
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>Define Future Business Process</li> </ul>	i) Activity diagram allows business users to get a better understanding of the business rules / logic in which a system will run, and model the business states.	<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i> Use Case Diagram <i>and/or</i> Sequence Diagram <i>and/or</i> State Diagram</li> </ul>
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>Elicit Functional Requirements</li> </ul>		<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i></li> </ul>

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
		Use Case Diagram <i>And/or</i> User Story
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>● Design Application</li> </ul>	i) Activity diagram can help to model key business and / or systems logic of a systems and /or components.	<ul style="list-style-type: none"> <li>● Sequence Diagram <i>and/or</i> State Diagram</li> </ul>

### 5.2.1 Example: Activity diagram



## 6 TOOL 6 – SEQUENCE DIAGRAM

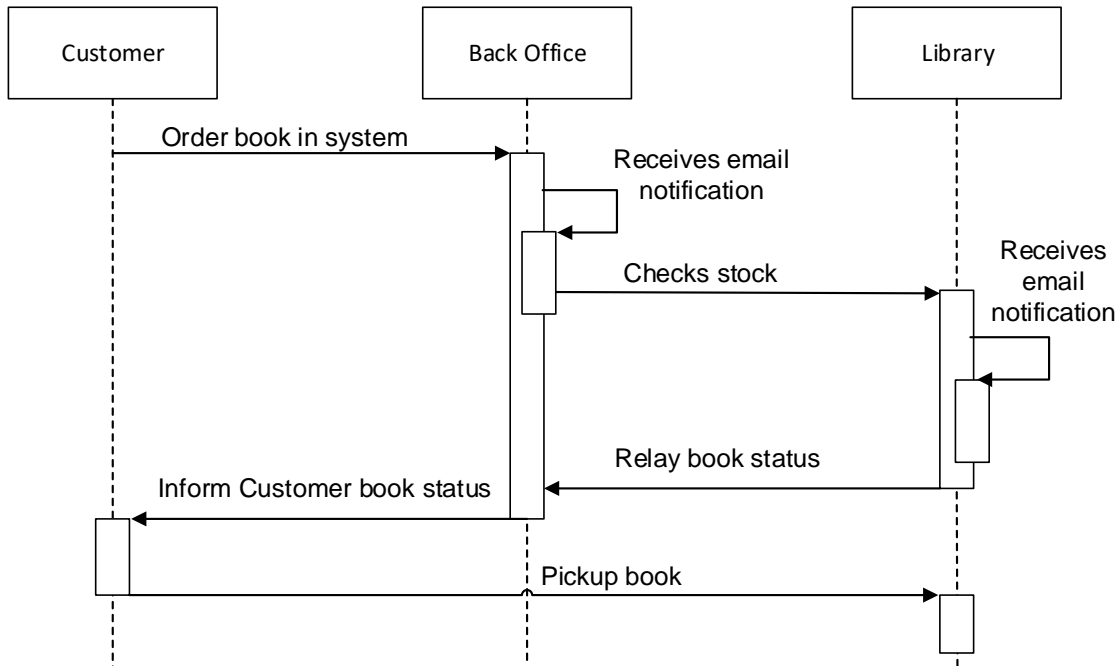
### 6.1 PURPOSE & DESCRIPTION

- (a) A sequence diagram in the UML is used to show how processes interact with one another and in time sequence. One advantage of sequence diagrams is that it is easy to see what messages come in/out of the system.
- (b) Other than communicating how the business currently works by showing interactions among various business components, it documents how the new system should behave in terms of the new business processes. Use case diagrams can often be refined into one or more sequence diagrams.
- (c) The use of sequence diagram is more important when interactions between processes need to be arranged in order of time sequence. In addition, simultaneous processes can be shown in parallel to each other; whether the messages are synchronous or asynchronous is also revealed.

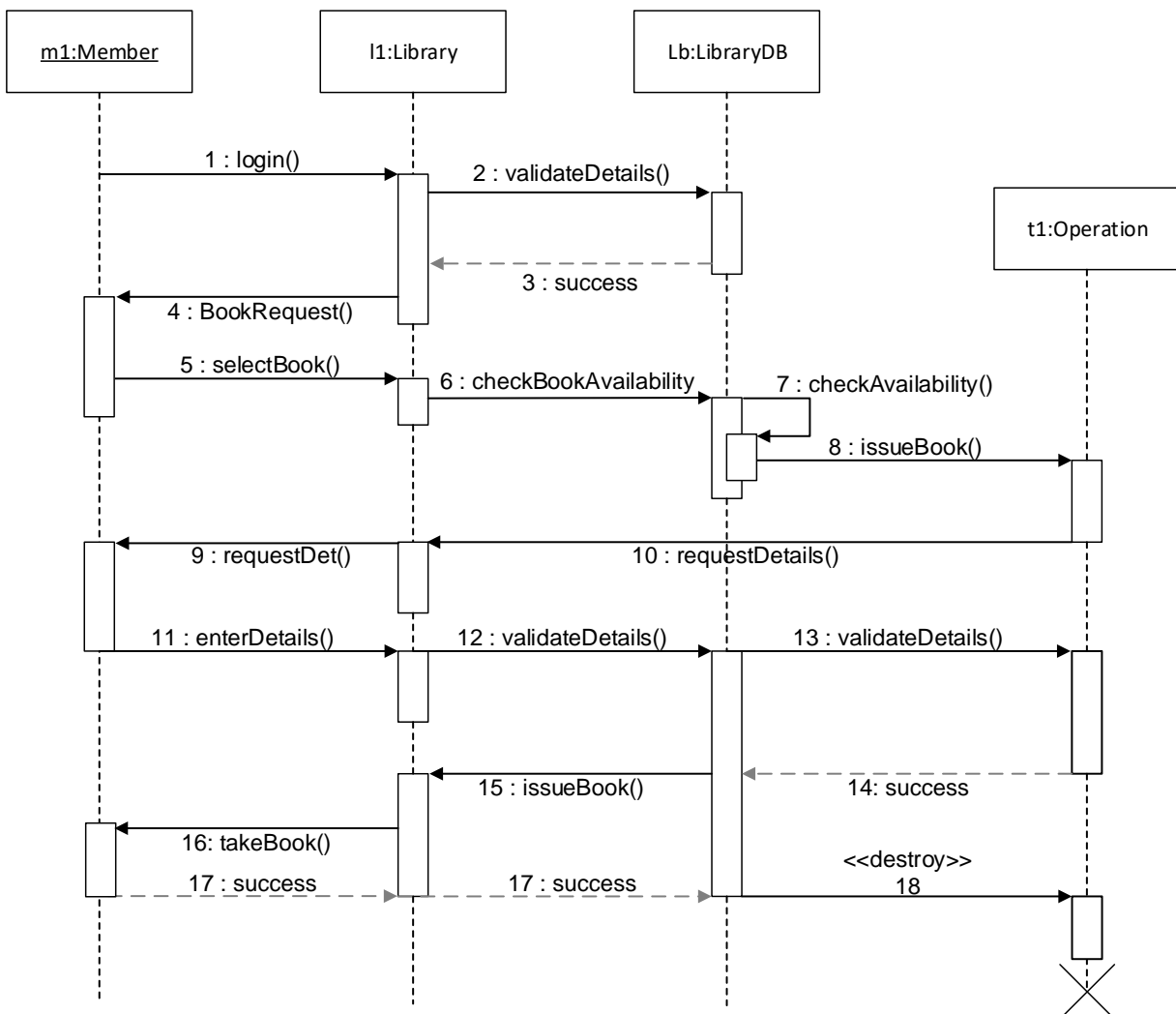
### 6.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>• Define Future Business Process</li> </ul>	i) As an alternative to flowchart diagram, sequence diagram can communicate how the business currently works by showing how various business components interact. A business-level sequence diagram can communicate requirements for future system implementation.	<ul style="list-style-type: none"> <li>• Flowchart Diagram <i>and/or</i> Use Case Diagram <i>and/or</i> Activity Diagram <i>and/or</i> State Diagram</li> </ul>
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>• Design Application</li> </ul>	i) Sequence diagram can be used to depict the sequence of actions and interactions among different components within a system. It can be used to depict the logic behind from a systems perspective, modelling the relationships between data and systems, and the flow of information.	<ul style="list-style-type: none"> <li>• Activity Diagram <i>and/or</i> State Diagram</li> </ul>
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>• Design Integration</li> </ul>		<ul style="list-style-type: none"> <li>• Class Diagram <i>and/or</i> State Diagram</li> </ul>

### 6.2.1 Example (1): Business-Level Sequence Diagram



### 6.2.2 Example (2): System-Level Sequence Diagram



## 7 TOOL 7 – USER STORY

### 7.1 PURPOSE & DESCRIPTION

User story is a common approach to capture high level user requirements in agile methodology. A typical format of a user story is shown below:

"As a/an <user role/who>, I want <goal/what> so that <benefit/why>"

User story may be written in free format to suit the project needs or user preference. It is intended not to contain any technical terms in the user story. Details requirements or technical requirements should be provided in the format of acceptance criteria. Complete acceptance criteria must be provided before the implementation of the story.

### 7.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>Elicit Functional Requirements</li> </ul>	i) User story can be used to collect and document user requirement as well as its acceptance criteria of this requirement. A group of user stories are usually grouped under a prioritised requirements list.	<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i> Use Case Diagram <i>and/or</i> Activity Diagram</li> </ul>

#### 7.2.1 Example: Borrow Book Story

**User Story Title:** Online Book Reservation

**User Story:**

As a customer, I want to reserve a book online so that I can make sure the book is available.

**Acceptance Criteria:**

- Each customer cannot reserve more than 3 books
- Reservation will be cancelled if customer did not pick up on specified date
- Customer must specify pick up date and location

**Submitted By:** Mr. Hamilton, User of Division XXX

**Submitted On:** 04.05.2016

**Last Modified:** 10.06.2016



## 8 TOOL 8 – DECISION TABLE

### 8.1 PURPOSE & DESCRIPTION

- (a) A decision table is used to illustrate business rules when multiple requirements interact. A system is often governed by complex logic such as business rules, with various combinations of conditions leading to different system behaviours. For complex business logic and decision which may be difficult to illustrate by text, Decision Tables can be used to represent what the system should do under different conditions.
- (b) In a decision table, there are two parts: conditions and actions displayed at the beginning of each row. Conditions are factors to consider when making a business decision. Actions are possible choices to take when certain business decisions are made.
- (c) Each column in the table shows a rule in the business logic that describes the unique combination of condition(s) and action(s) that form the business decision. Conditions can be expressed as possible values (i.e. True (T) / False (F) or Yes (Y) / No (N).

### 8.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>• Elicit Functional Requirements</li> </ul>	i) Decision table can be created to help project team understand and formulate requirements of each process that involves complex business logic. It also helps consider all possible combinations of each condition.	<ul style="list-style-type: none"> <li>• Decision Trees</li> </ul>

#### 8.2.1 Example: Decision Table

		Rules					
Conditions	Member has placed booking request	Y	Y	Y	Y	N	N
	Book is available	Y	Y	Y	N	N	N
	Book is restricted	N	N	Y	Y	N	Y
	Member has VIP membership	Y	N	Y	Y	Y	N
Actions	Check if Member has outstanding loans	X	X	X	X	X	X
	Check if Member is blacklisted	X	X	X	X		
	Accept booking request	X	X				
	Reject booking request			X	X		

## 9 TOOL 9 – DECISION TREE

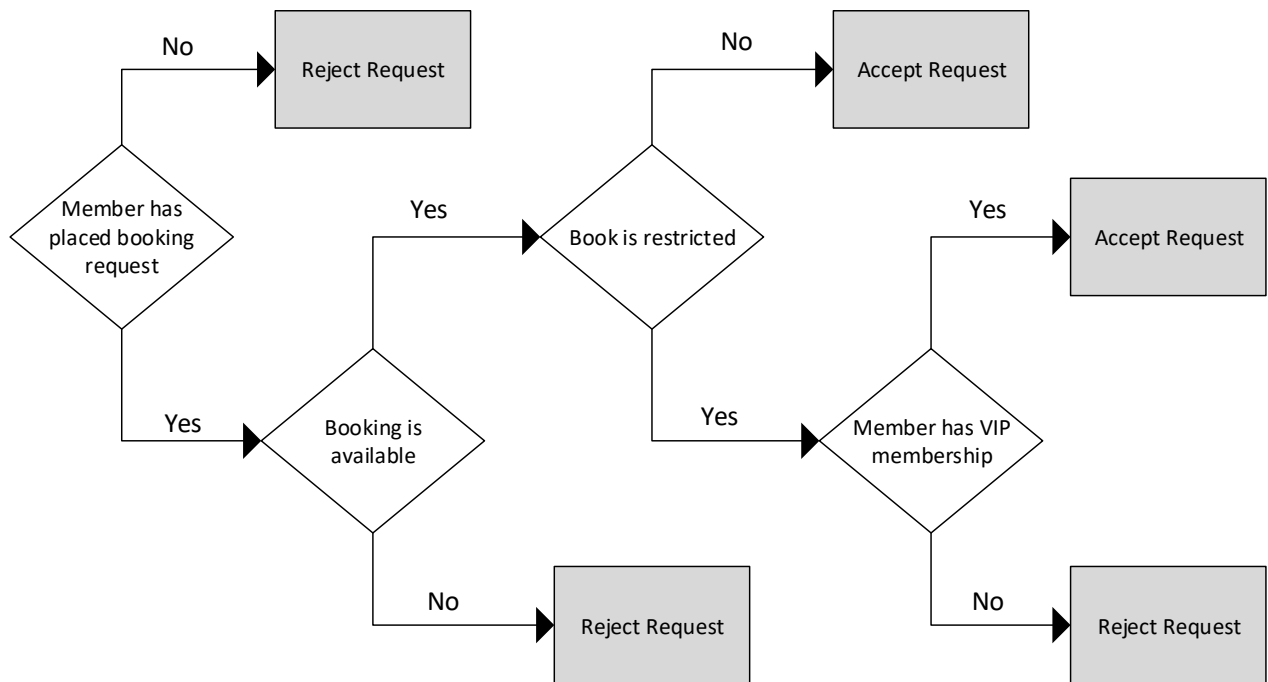
### 9.1 PURPOSE & DESCRIPTION

- (a) Decision tree is an alternative technique to decision table in presenting complex business logic as shown above. A system is often governed by complex logic such as business rules, with various combinations of conditions leading to different system behaviours. For complex business logic and decision which may be difficult to illustrate by text, decision trees can be used to illustrate options and investigate possible outcomes if specific options are chosen.
- (b) Similar to a flowchart diagram, a decision tree contains two main types of shapes: rectangles and diamonds. A rectangle denotes an action, while a diamond denotes decision points/ questions to be made.
- (c) The decision tree can also be read not only at the start of the left hand side, but can be traced from the right hand side. This allows the project team to analyse fully the possible consequences of a decision.

### 9.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>● Elicit Functional Requirements</li> </ul>	i) Decision tree can be used to visually aid communication with business users when considering complex business logic. It can also explicitly show all factors of considerations.	<ul style="list-style-type: none"> <li>● Decision Tree</li> </ul>

### 9.2.1 Example: Decision Tree



## 10 TOOL 10 – ACTION CHECKLIST

### 10.1 PURPOSE & DESCRIPTION

- (a) Action Checklist helps the project team to execute a self-check exercise to identify any gaps or mistakes prior to a team review. It consists of items for requirement traceability, deliverables sign-off and action items related to deliverables of CED, Requirements Specification, System Specification, and TSO.
- (b) Project team should check against each action item on the action checklist at a regular basis and update each item’s status where applicable.

### 10.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
The Action Checklist is used across all processes within the Guide.	i) Based on the action items stated in the action checklist, project team can conduct a self-check exercise towards the end of each deliverable to ensure all required action items are covered.	<ul style="list-style-type: none"> <li>• Nil</li> </ul>

#### 10.2.1 Example: Screenshot of the action checklist

Checklists			
<b>Project Name:</b>	<Enter the project name>	<b>Reviewer:</b>	<Enter the reviewer's name>
<b>Project ID:</b>	<Enter the project ID, if any>	<b>Date:</b>	<Enter the date, dd-mmm-yy>
<b>Deliverable Titles:</b>	<Enter the titles of the deliverables reviewed>	<b>Version:</b>	<Enter the version number>
#	Item/Description	Status	
	<b>Current Environment</b>		
1	Information on the current environment has been obtained	Incomplete	
2	The system's objectives and major functions have been documented	Incomplete	
3	The system's hardware specifications, software specifications and network infrastructure/configurations have been stated	Incomplete	
4	The transaction volumes and frequencies of each system module are documented	Incomplete	
5	All the current system's interfaces have been highlighted	Incomplete	
6	The business process diagram is documented in a high level manner, capturing only the key user interactions, system activities and system process flow	Incomplete	
7	The current pain points and issues have been captured	Incomplete	

## 11 TOOL 11 – REQUIREMENTS TRACEABILITY MATRIX

### 11.1 PURPOSE & DESCRIPTION

- (a) Requirements Traceability Matrix (RTM) is used to show the impacts of a change in requirements and make sure there are no orphan requirements that do not have development and test efforts associated with them.
- (b) Traceability needs to be considered at the beginning of requirements gathering, and maintained throughout the project lifecycle. As a best practice, traceability should begin at business requirements gathering stage and link to the subsequent design, development, and testing phases. Requirements traceability can be recorded/updated in MS Excel file.

### 11.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>The Requirements Traceability Matrix is used across all processes within the Guide.</b></p>	<p>i) Based on the requirements recorded in the Requirements Traceability Matrix, project teams can update the Matrix at the end of each sub-deliverable/deliverable to establish and maintain the traceability of each requirement or element as it evolves through the project lifecycle.</p>	<ul style="list-style-type: none"> <li>• Nil</li> </ul>

#### 11.2.1 Example: Requirements Traceability Matrix

- (a) After the completion of Requirements Specification, the columns “*Business Process Model*”, “*Requirements ID*”, “*Use Case Name/ID*” and/or “*User Story ID*” should be filled in.
- (b) Completion of System Specification should allow columns “*Design Document Name*” and “*Component Name*” to be filled in.
- (c) After the Technical System Option has been selected, the columns “*Unit Test Case ID*”, “*System Test Case ID*” and “*User Acceptance Test Case ID*” can be filled and used for testing and ensuring all identified requirements have been fulfilled.

Business Requirement	Business Process Model	Req. ID/ Use Case ID/User Story ID	Function ID	Related System	Architecture & System Design section	Unit Test Case ID	System Test Case ID	User Acceptance Test Case ID
<i>e.g. R1</i>	<i>Figure 1</i>	<i>UseCase-1</i>	<i>System Specification</i>	<i>Document Management System</i>	<i>Design Application Section 3.3.1</i>	<i>UnitTest-1</i>	<i>ST-1</i>	<i>UAT-1</i>

## 12 TOOL 12 – CONCEPTUAL DATA MODEL DIAGRAM

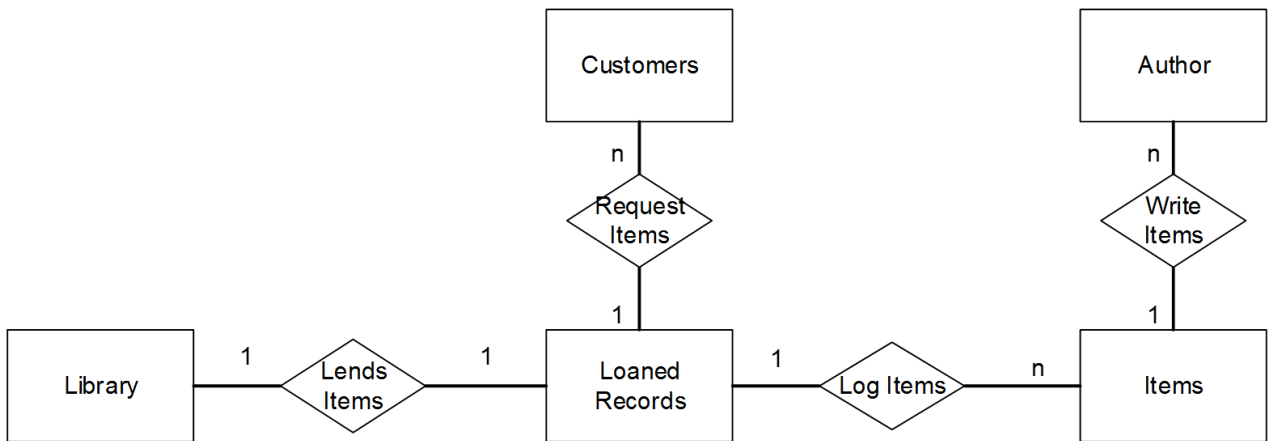
### 12.1 PURPOSE & DESCRIPTION

- (a) Conceptual data model diagram provides a high-level description of a business's informational needs. It typically includes only the main concepts and the main relationships among core data entities (or classes). It shows how the business users see organisational information as well as enables IT users to gain a business view and business users to gain a high-level data view.
- (b) There are two types of diagrams available for use: Conceptual Entity-Relationship diagram (conceptual ER-diagram) and conceptual class diagram. They are the same except that their notations are different. Consideration of use depends on the skill and preference of the stakeholders.

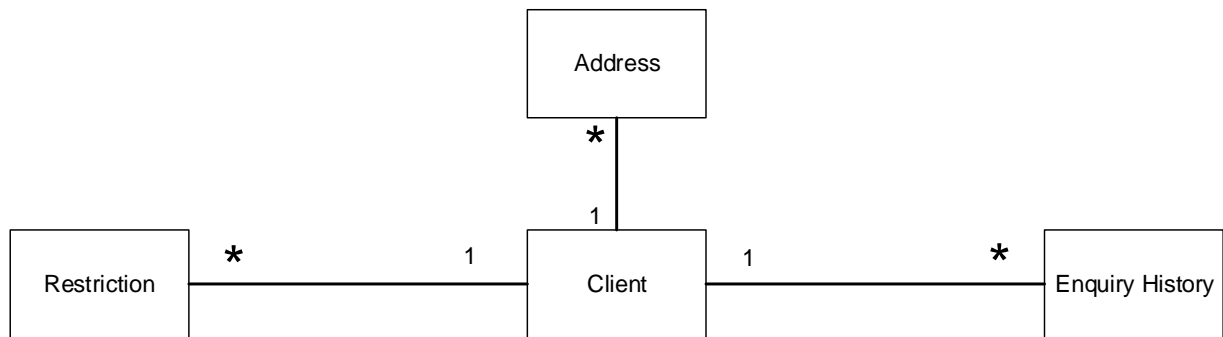
### 12.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>System Specification</b></p> <p>Architecture Design</p> <ul style="list-style-type: none"> <li>• Develop Data Store Architecture</li> </ul>	<p>i) Conceptual data model diagram helps to engage business end-users in new system development or changes to the existing system. It should be understandable to business users, using only high-level data view to illustrate entity relationships. It should not include any attributes, procedures, interface definitions and platform-specific information, such as data types.</p>	<ul style="list-style-type: none"> <li>• Data Dissemination Diagram</li> </ul>

### 12.2.1 Example (1): Conceptual ER-diagram



### 12.2.2 Example (2): Conceptual class diagram





## 13 TOOL 13 – DATA DISSEMINATION DIAGRAM

### 13.1 PURPOSE & DESCRIPTION

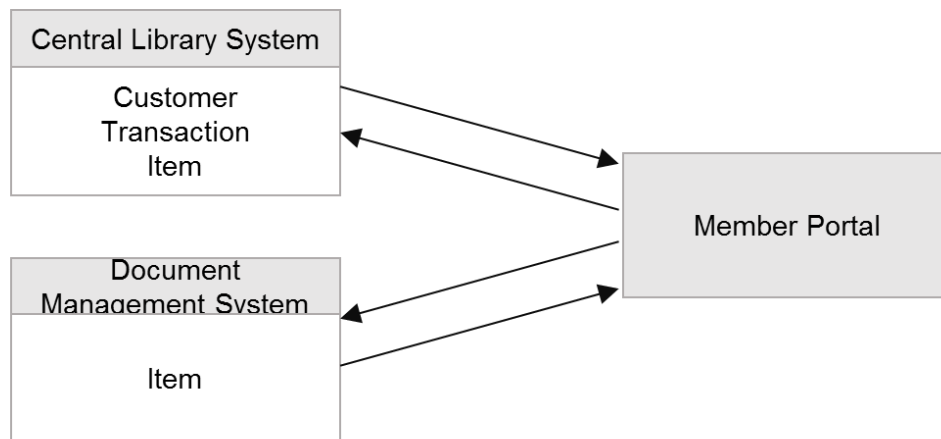
Data dissemination diagram, recognised as a supporting tool for enterprise architecture by The Open Group Architecture Framework (TOGAF)<sup>2</sup>, shows the relationship between logical data entities, business services and application components. The diagram shows how the logical data entities are to be physically managed and handled by application components. Additionally, the diagram may show data replication and system ownership of the master reference for data. It can show copies and the master-copy relationship. It can also include business services in association with an application, which either encapsulate data or access data encapsulated within the application.

### 13.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>System Specification</b></p> <p>Architecture Design</p> <ul style="list-style-type: none"> <li>Develop Data Store Architecture</li> </ul>	<p>i) Data Dissemination Diagram is used to show the flow of data entities across different data stores of the application or across systems. Additionally, the diagram may show data replication and system ownership of the master reference for data.</p>	<ul style="list-style-type: none"> <li>Conceptual Data Model Diagram</li> </ul>

<sup>2</sup> The Open Group Architecture Framework (TOGAF) is a framework for developing enterprise architecture. More information for data dissemination diagrams can be found at [www.togaf-modeling.org/models/data-architecture-menu/data-dissemination-diagrams-menu](http://www.togaf-modeling.org/models/data-architecture-menu/data-dissemination-diagrams-menu)

### 13.2.1 Example: Data dissemination diagram



Business service	Data entities	Application
	Item	<ul style="list-style-type: none"> <li>Central Library System (Master)</li> <li>Document Management System (Master)</li> <li>Member Portal (Copy)</li> </ul>
Member Portal	Customer	<ul style="list-style-type: none"> <li>Central Library System (Master)</li> <li>Member Portal (Copy)</li> </ul>
	Transaction	<ul style="list-style-type: none"> <li>Central Library System (Master)</li> <li>Member Portal (Copy)</li> </ul>

## 14 TOOL 14 – DATA MIGRATION DIAGRAM

### 14.1 PURPOSE & DESCRIPTION

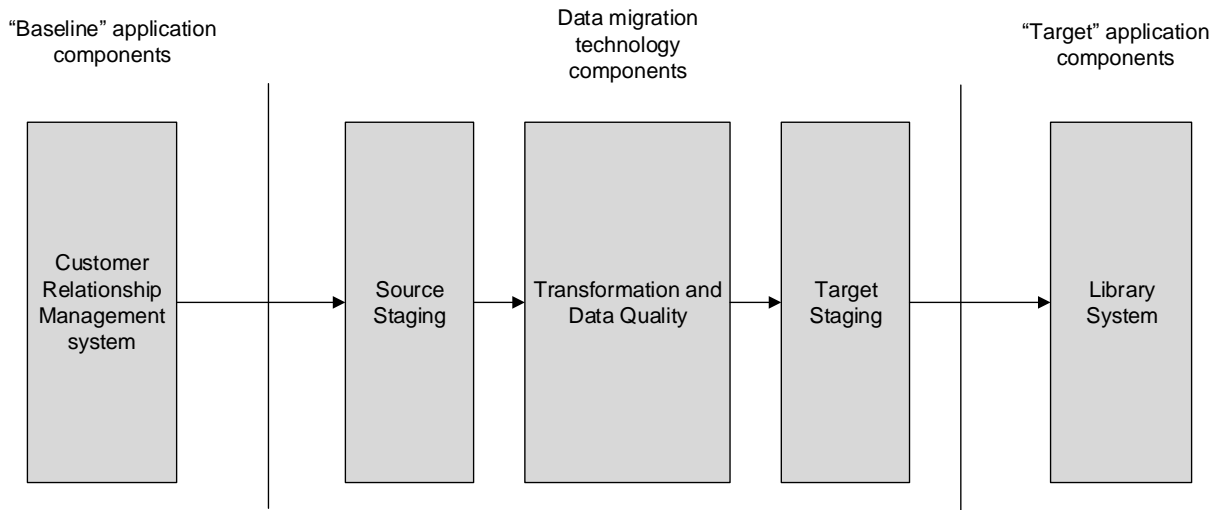
- (a) The purpose of data migration diagram under TOGAF<sup>3</sup> is to show the data flow and flow type between the source system and new system. It provides a visual presentation of the differences between the source system and new system, and is used to carry out data audits to check that no information has been lost.
- (b) Data migration diagram can be developed and elaborated as necessary. For example, the diagram can show a high-level overview of the data and processes to migrate or can go into greater detail, showing all individual data elements and types of flow.

### 14.2 AREAS OF USE

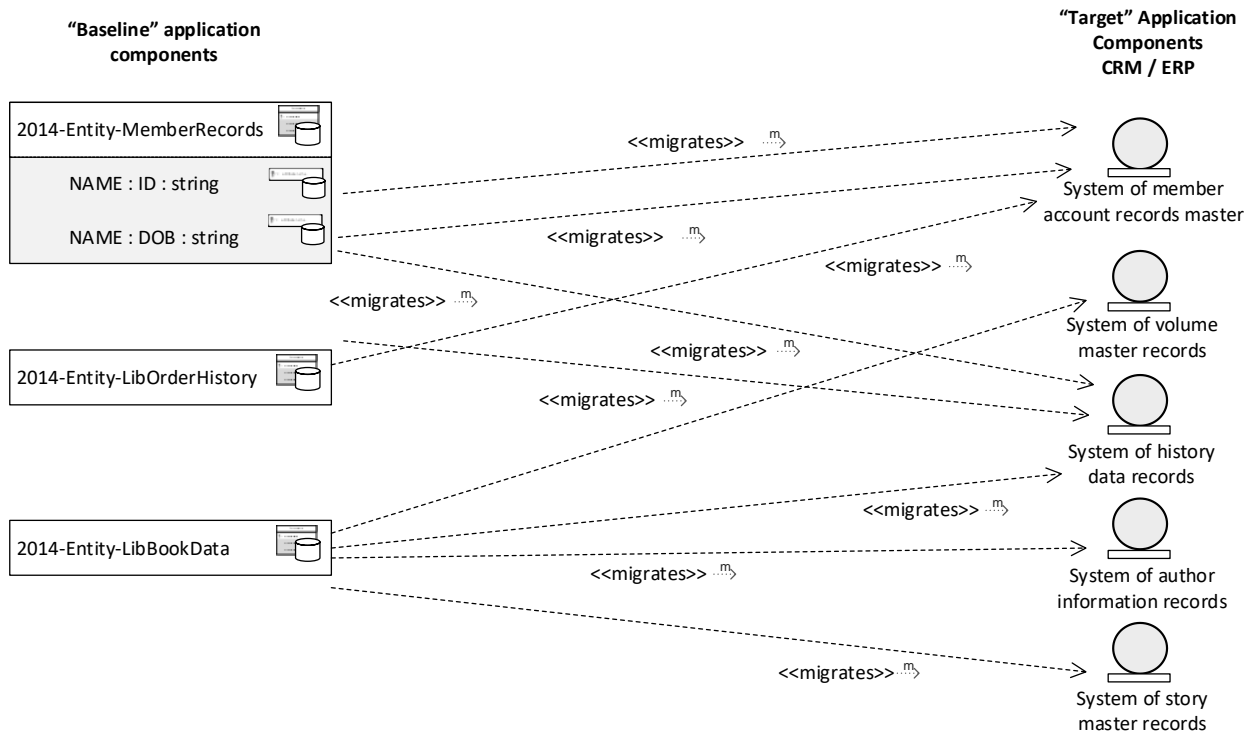
Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>System Specification</b></p> <p>Architecture Design</p> <ul style="list-style-type: none"> <li>Design Data Conversion Architecture</li> </ul>	<p>i) Data migration diagram can show the flow of data from the source to the new applications and contain a high-level view of migration processes or contain the details of individual system data elements. It can also help in modelling integration between systems.</p>	<ul style="list-style-type: none"> <li>Nil</li> </ul>

<sup>3</sup> More information for data migration diagrams can be found at [www.togaf-modeling.org/models/data-architecture-menu/data-migration-diagrams-menu](http://www.togaf-modeling.org/models/data-architecture-menu/data-migration-diagrams-menu)

### 14.2.1 Example (1): Data migration diagram



### 14.2.2 Example (2): Data migration diagram



## 15 TOOL 15 – BLOCK DIAGRAM

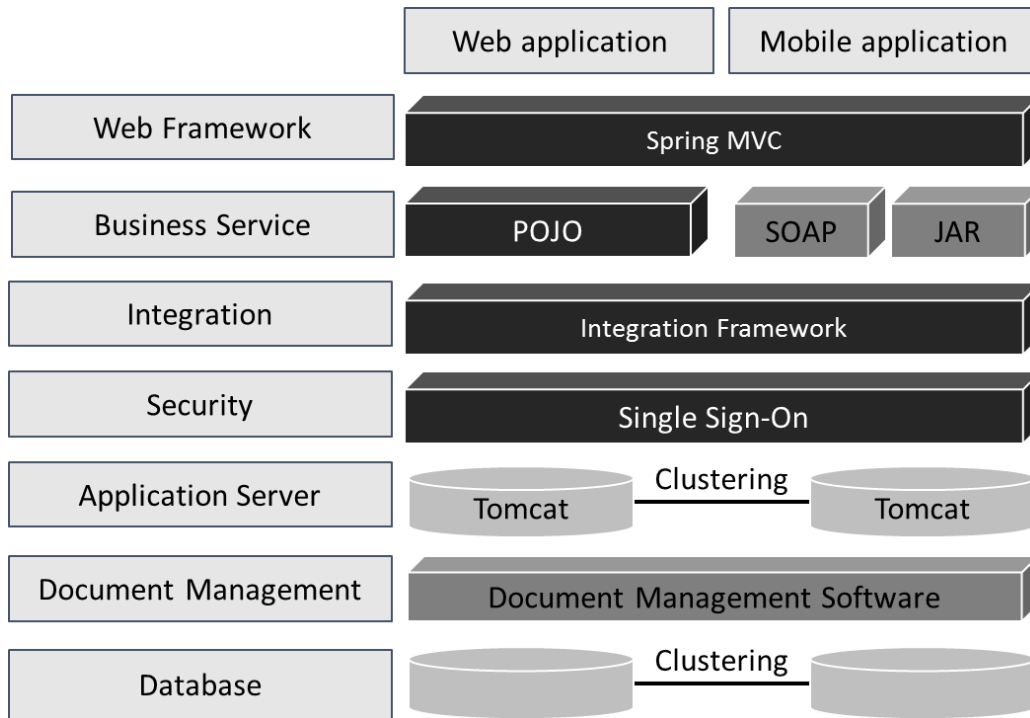
### 15.1 PURPOSE & DESCRIPTION

Block diagram is typically used to show high level information of the overall architecture of the system. It is useful in explaining the system without the need to understand individual components within the system.

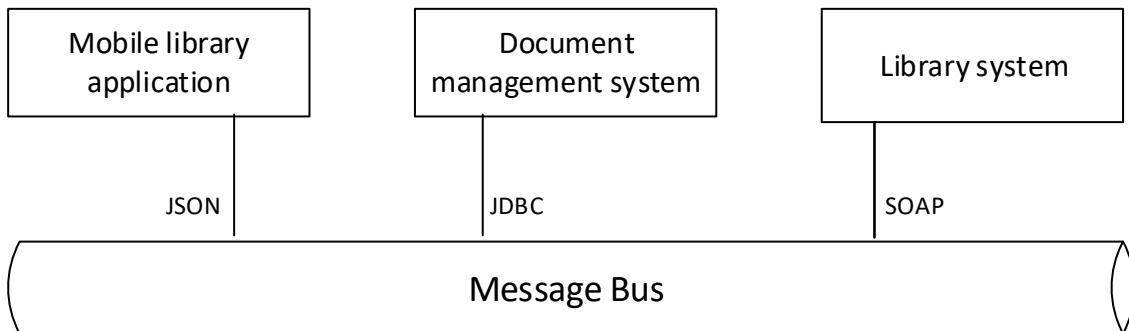
### 15.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>System Specification</b> Architecture Design <ul style="list-style-type: none"> <li>Develop Application Architecture</li> </ul>	i) Block diagram can show the architecture of the system as deployment of software components to deployment targets. The functions or components are represented by blocks and the blocks may be connected by lines that show the relationships between the blocks.	<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i> Deployment Diagram</li> </ul>
<b>System Specification</b> Architecture Design <ul style="list-style-type: none"> <li>Develop Integration Architecture</li> </ul>		<ul style="list-style-type: none"> <li>Application Communication Diagram</li> </ul>
<b>Technical System Option</b> Technical System Architecture <ul style="list-style-type: none"> <li>Design Technical System Architecture</li> </ul>		<ul style="list-style-type: none"> <li>Network Diagram <i>and/or</i> Data Storage Diagram <i>and/or</i> Server Diagram</li> </ul>

### 15.2.1 Example (1): Block diagram (Application)



### 15.2.2 Example (2): Block diagram (Integration)



## 16 TOOL 16 – DEPLOYMENT DIAGRAM

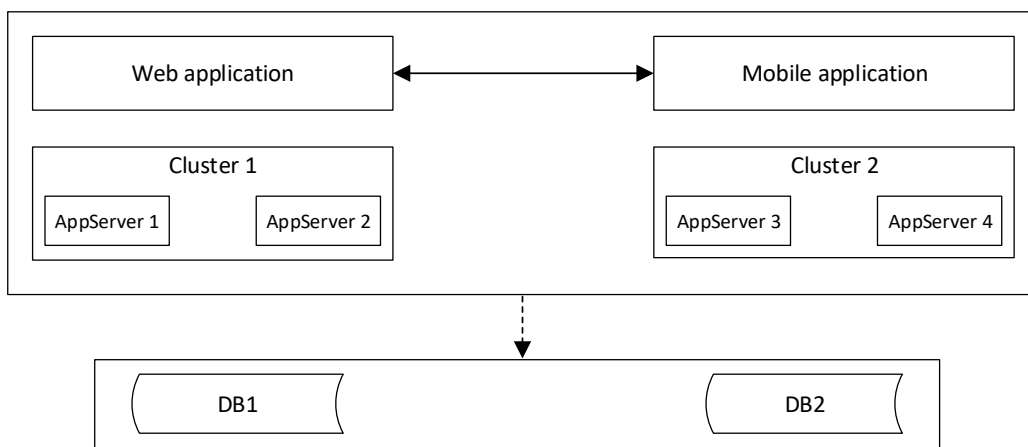
### 16.1 PURPOSE & DESCRIPTION

Deployment diagram in the UML<sup>4</sup> is used to visualise the distribution between software and hardware elements in the system application architecture. A node is either a hardware or software element both of which are both shown as boxes. A line that connects two nodes represents the communication path between them.

### 16.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>System Specification</b></p> <p>Architecture Design</p> <ul style="list-style-type: none"> <li>Develop Application Architecture</li> </ul>	<p>i) Deployment diagram is developed to model the architecture of the system as deployment of software components to deployment targets. A node can contain other elements, such as artefacts. An artefact is an element of the software development process. Examples of artefacts include scripts, executable files in a motherboard, table in a database system, user manuals, mail message etc. Artefact is denoted by a rectangle showing the artefact name e.g. &lt;&lt;component&gt;&gt;.</p>	<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i> Block Diagram</li> </ul>

#### 16.2.1 Example: Deployment diagram



<sup>4</sup> More information about UML tools can be found at [www.uml.org](http://www.uml.org)

## 17 TOOL 17 – APPLICATION COMMUNICATION DIAGRAM

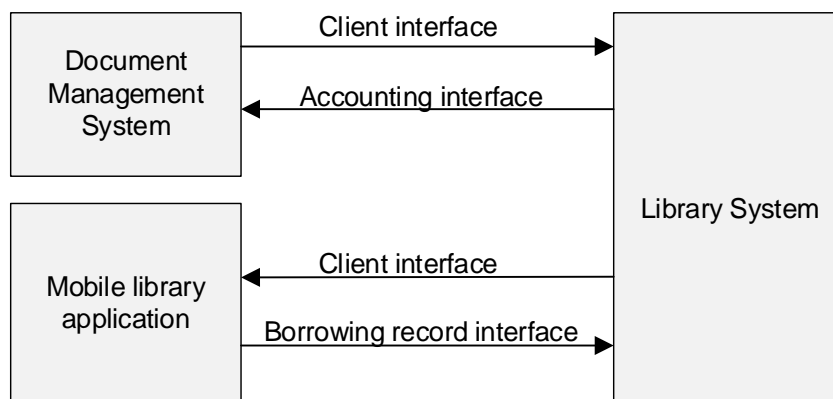
### 17.1 PURPOSE & DESCRIPTION

- (a) Application communication diagram under TOGAF<sup>5</sup> is used to depict application components and interfaces between components. Communication should be logical and should only show intermediary technology where appropriate. The application communication diagram helps emphasise the linkages between applications.
- (b) Typically, a system interface can be one of following types:
  - i) Web services
  - ii) File transfer
  - iii) Database
  - iv) Messaging

### 17.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>System Specification</b> Architecture Design <ul style="list-style-type: none"> <li>• Develop Integration Architecture</li> </ul>	i) Application communication diagram shows application components and interfaces between components. Interfaces may be associated with data entities and applications may be associated with business processes where appropriate.	<ul style="list-style-type: none"> <li>• Block Diagram</li> </ul>

#### 17.2.1 Example: Application Communication Diagram



<sup>5</sup> More information for data migration diagrams can be found at [www.togaf-modeling.org/models/application-architecture-menu/application-communication-diagrams-menu](http://www.togaf-modeling.org/models/application-architecture-menu/application-communication-diagrams-menu)



## 18 TOOL 18 – ENTITY RELATIONSHIP DIAGRAM

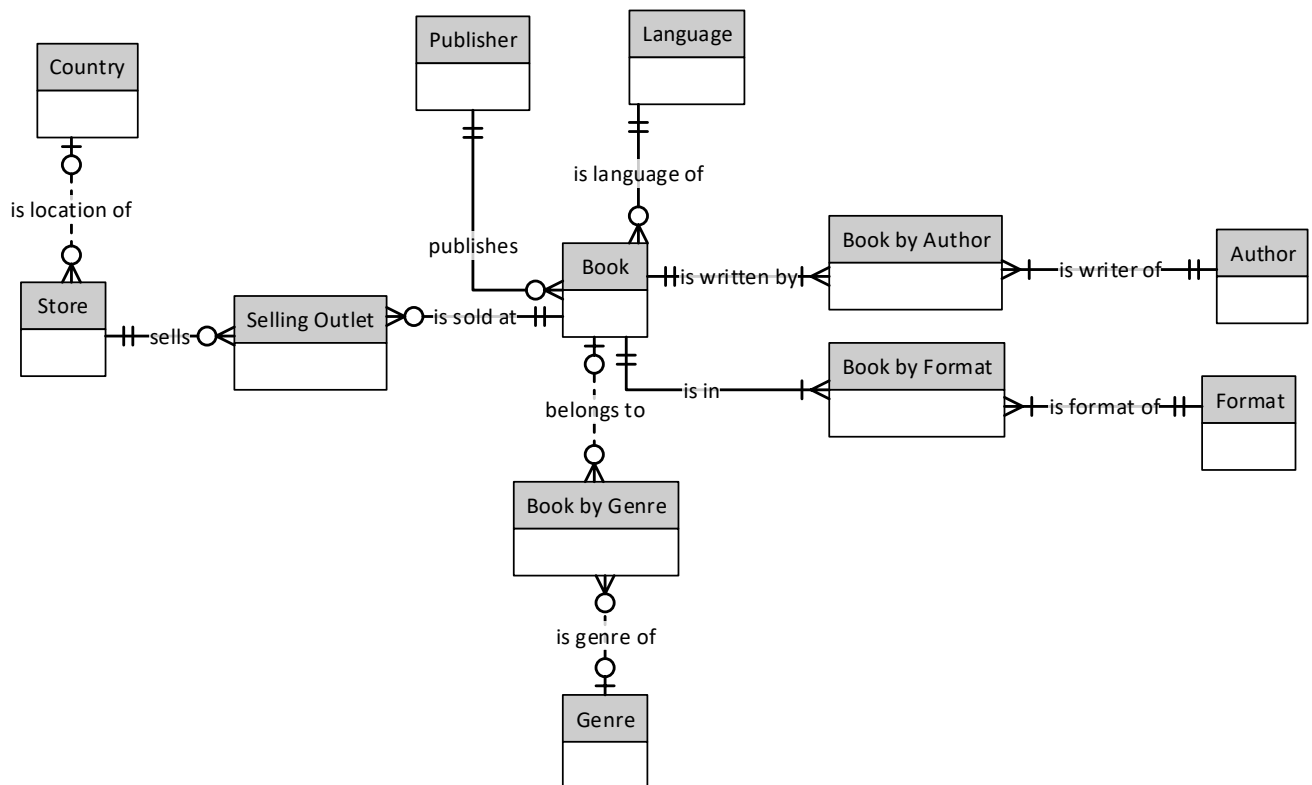
### 18.1 PURPOSE & DESCRIPTION

Entity Relationship (ER) diagram is used to illustrate the relationships between entities in a data model. The ER diagram can be used to model both logical and physical data model. Entities are linked with each other by relationships that express the dependencies and requirements between them.

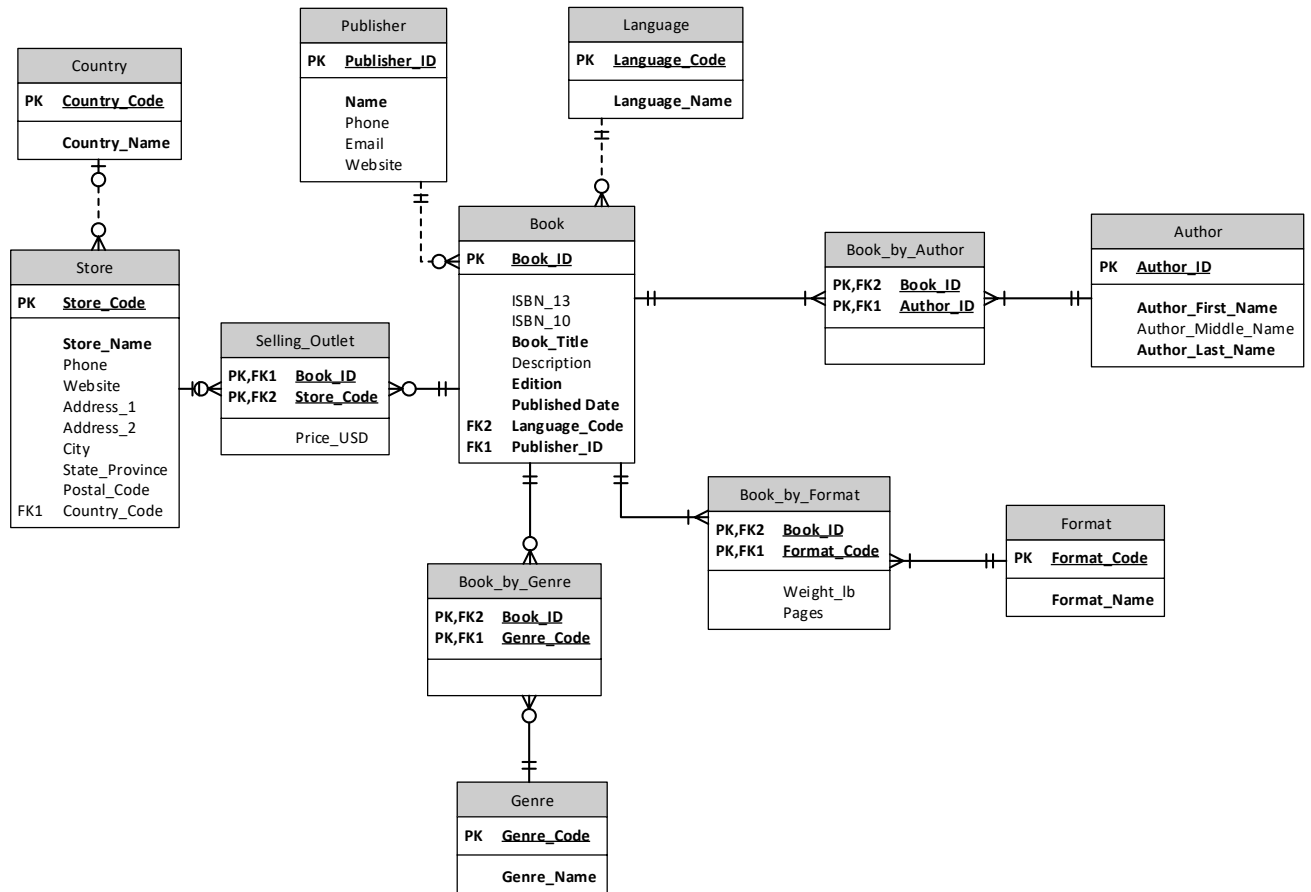
### 18.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>• Design Logical Data Model</li> <li>• Design Physical Data Model</li> </ul>	i) Logical ER diagram shows the logical data entities and their relationships in a logical data model.  ii) Physical ER diagram shows the details of entities, such as keys and attributes, in a physical data model.	<ul style="list-style-type: none"> <li>• Class Diagram</li> </ul>

#### 18.2.1 Example (1): Logical ER diagram



#### 18.2.2 Example (2): Physical ER diagram



## 19 TOOL 19 – CLASS DIAGRAM

### 19.1 PURPOSE & DESCRIPTION

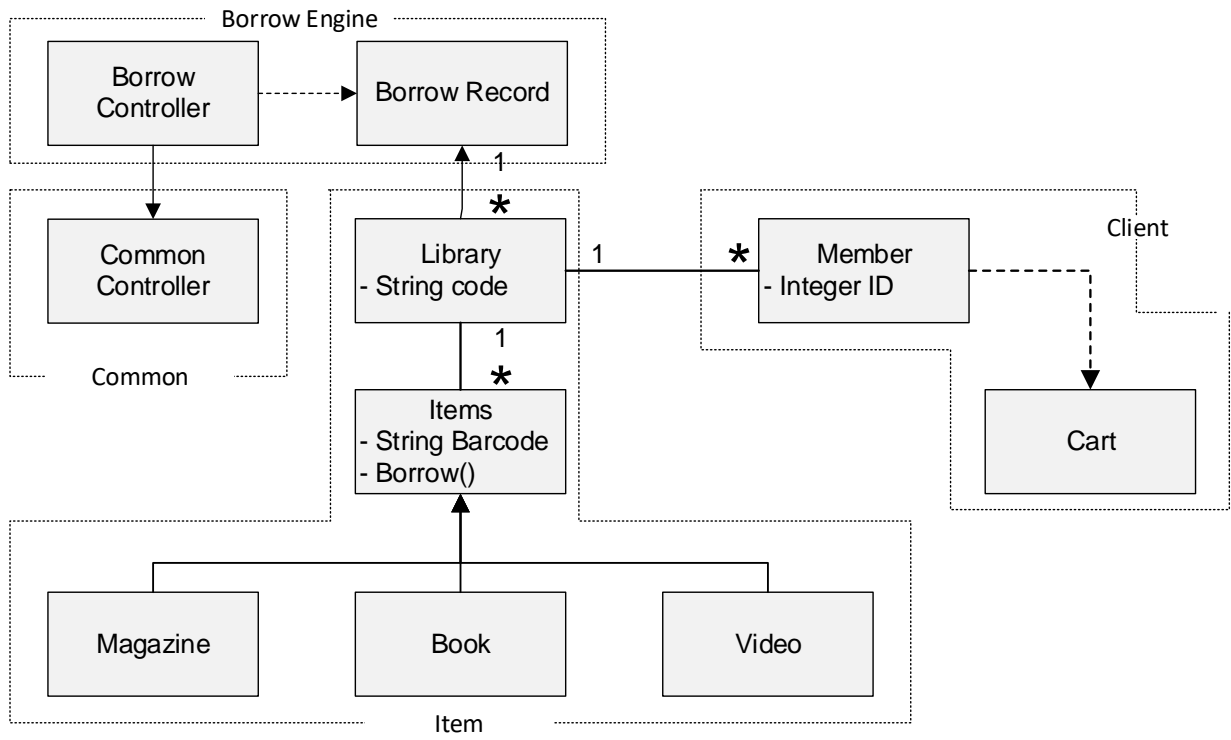
- (a) Class diagram in the UML<sup>6</sup> is a static structure diagram that can be used for both logical and physical data model (like Entity Relationship diagram). In addition, it can be used to describe the structure of a system by showing the system's classes, their attributes, operations or methods, and the relationships among components.
- (b) Class diagrams are generally developed iteratively over the lifecycle of the analysis and design phase. The main concepts in class diagram are class, attribute, association, multiplicity and generalisation.
- i) A class is the descriptor for a set of components with similar structure, behaviour, and relationships.
  - ii) An attribute is a single, named fragment of the persistent state.
  - iii) An association exists when some number of components are connected (linked) in some significant manner.
  - iv) Multiplicity describes the rules about the number of components in each class that may or may not be linked.

#### 19.1.1 Areas of Use

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>System Specification</b> System Design • Design Application	i) Class diagram shows the logical data entities and their relationships, the details of entities such as keys and attributes, modelling of interfaces between major components, and modelling of system structure, including attributes, methods and relationships between components of subsystems.	• Nil
<b>System Specification</b> System Design • Design Integration		• Sequence Diagram
<b>System Specification</b> System Design • Design Logical Data Model • Design Physical Data Model		• Entity Relationship Diagram

<sup>6</sup> More information about UML tools can be found at [www.uml.org](http://www.uml.org)

### 19.1.2 Example: Class diagram



## 20 TOOL 20 – SECURITY CONTROLS IMPLEMENTATION OVERVIEW DIAGRAM

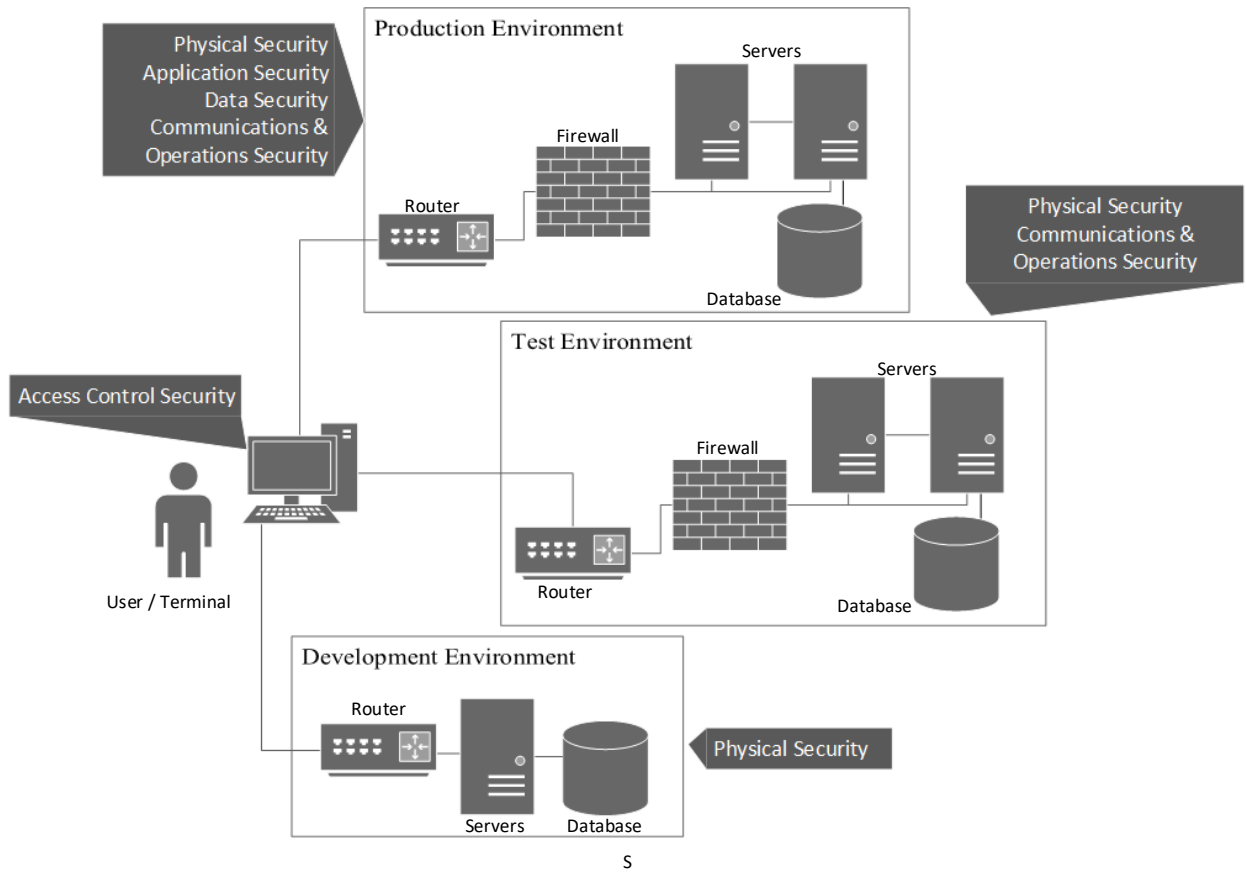
### 20.1 PURPOSE & DESCRIPTION

Security controls implementation overview diagram is used to demonstrate where security controls will be implemented based on a user’s view and logical system architecture. This will help the user and system owners understand where controls are placed and how data will be protected from a high level perspective.

### 20.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>System Specification</b></p> <p>Architecture Design</p> <ul style="list-style-type: none"> <li>• Develop Security Architecture</li> </ul> <p>System Design</p> <ul style="list-style-type: none"> <li>• Design Security</li> </ul>	<p>i) Security controls implementation overview diagram is developed during security design to demonstrate the security controls. It shows where the security controls will be implemented to protect the system components (servers, desktops, and supporting infrastructure).</p>	<ul style="list-style-type: none"> <li>• Nil</li> </ul>

### 20.2.1 Example: Security controls implementation overview diagram



## 21 TOOL 21 – NETWORK DIAGRAM

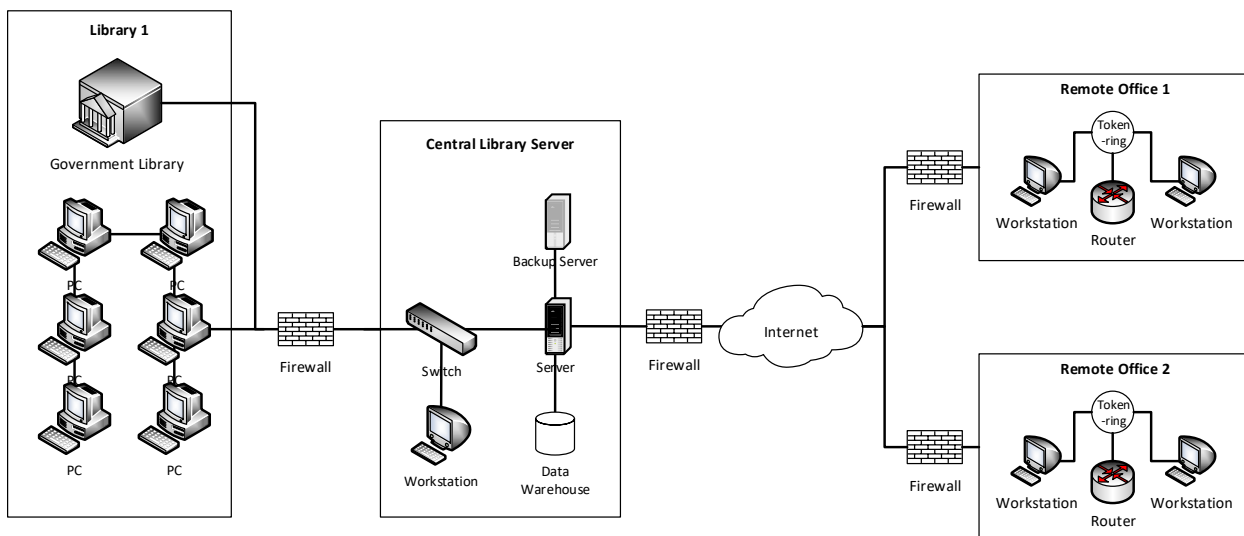
### 21.1 PURPOSE & DESCRIPTION

Network diagram can be used to illustrate network architecture. It usually consists of a network that is made up of interconnected groups or systems. The scope of a network diagram can be confined to a single application, a family of applications or even an entire enterprise. Network diagram can also cover wireless networks, networking wirings, LAN networks, network topology, activity networks and network cabling.

### 21.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Technical System Option</b> Technical System Architecture <ul style="list-style-type: none"> <li>Design Technical System Architecture</li> </ul>	i) Network diagram can be used to illustrate the network architecture incorporating the network, storage and platform architecture components.	<ul style="list-style-type: none"> <li>Data Storage Diagram <i>and/or</i> Server Diagram</li> </ul>

#### 21.2.1 Example: Network diagram



## 22 TOOL 22 – DATA STORAGE DIAGRAM

### 22.1 PURPOSE & DESCRIPTION

Data storage diagram can be used to demonstrate storage architecture. It is to illustrate how the data is stored and what types of storage solution are used. Data storage diagram does not have a standard format.

### 22.2 AREAS OF USE

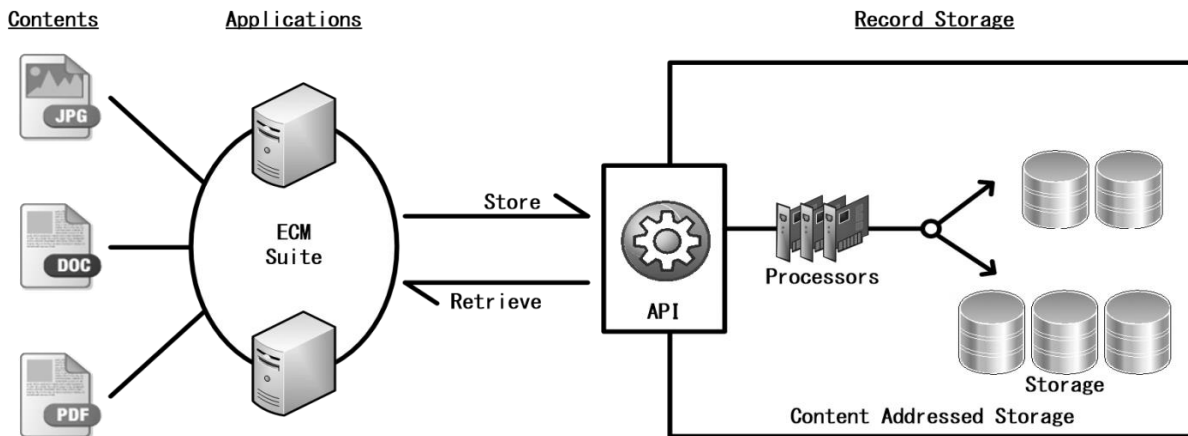
Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>Technical System Option</b></p> <p>Technical System Architecture</p> <ul style="list-style-type: none"> <li>Identify Technical System Architecture Components</li> </ul>	<p>i) Data storage diagram helps in identifying technical system architecture component, in particular the storage component. It can be used to illustrate how data is stored and accessed from architectural perspective.</p>	<ul style="list-style-type: none"> <li>Network Diagram <i>and/or</i> Server Diagram</li> </ul>
<p><b>Technical System Option</b></p> <p>Technical System Architecture</p> <ul style="list-style-type: none"> <li>Design Technical System Architecture</li> </ul>		<ul style="list-style-type: none"> <li>Block Diagram <i>and/or</i> Network Diagram <i>and/or</i> Server Diagram</li> </ul>



### 22.2.1 Example (1): Data storage diagram



### 22.2.2 Example (2): Data storage diagram



## 23 TOOL 23 – SERVER DIAGRAM

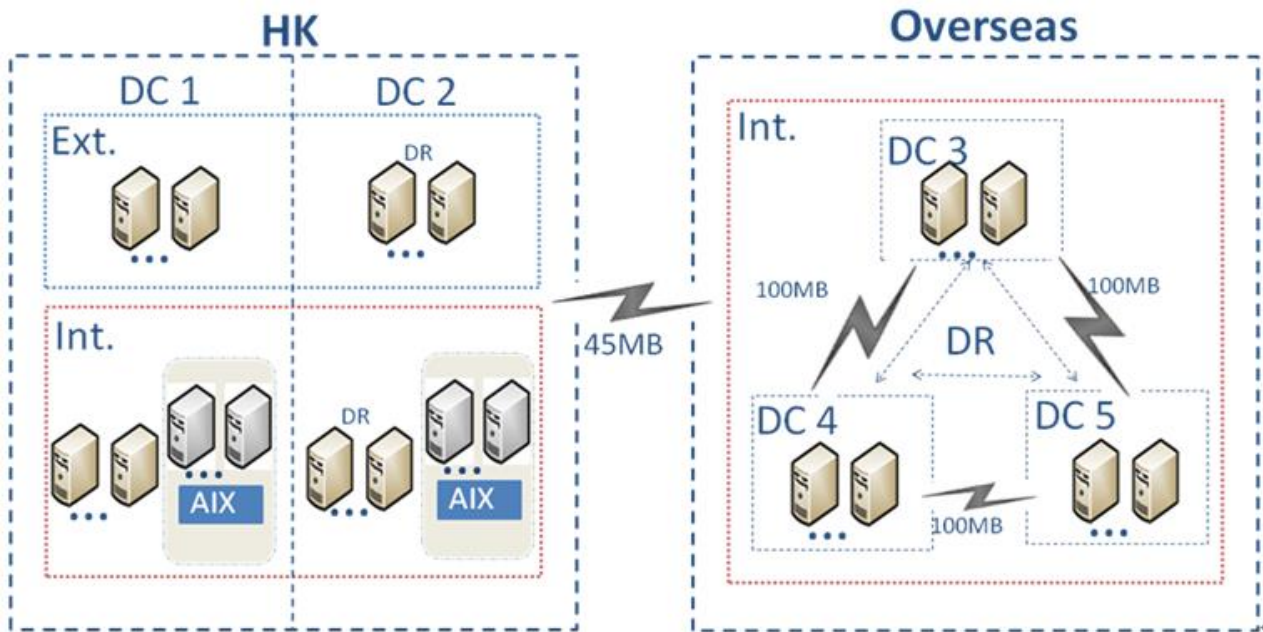
### 23.1 PURPOSE & DESCRIPTION

Server diagram is used to present the platform architecture, which shows how the servers in different physical/logical locations are connected with one another. The diagram also describes the bandwidth, types of servers, and their usage (e.g. use as internal, use as DR, etc.). Server diagram can be represented by a network or block diagram.

### 23.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>Technical System Option</b></p> <p>Technical System Architecture</p> <ul style="list-style-type: none"> <li>Identify Technical System Architecture Components</li> </ul>	<p>i) Server diagram helps in identifying the platform architecture component. It can be used to illustrate how the servers in different locations are connected with one another.</p>	<ul style="list-style-type: none"> <li>Network Diagram <i>and/or</i> Data Storage Diagram</li> </ul>
<p><b>Technical System Option</b></p> <p>Technical System Architecture</p> <ul style="list-style-type: none"> <li>Design Technical System Architecture</li> </ul>		<ul style="list-style-type: none"> <li>Block Diagram <i>and/or</i> Network Diagram <i>and/or</i> Data Storage Diagram</li> </ul>

### 23.2.1 Example: Server diagram



## 24 TOOL 24 – COST / BENEFIT EVALUATION TECHNIQUE

### 24.1 PURPOSE & DESCRIPTION

Cost / benefit evaluation for TSO provides a systematic approach for project team to leverage when choosing a technical system option. Costs refer to both non-recurrent and recurrent costs of the proposed system, while benefits refer to both the tangible and intangible benefits of the proposed option.

### 24.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Technical System Option</b> Cost / Benefit Evaluation <ul style="list-style-type: none"> <li>Perform Cost / Benefit Evaluation</li> </ul>	i) After identifying gaps and producing estimation of tangible and intangible costs/benefits, cost / benefit evaluation tool can be used in making comparisons between technical system options.	<ul style="list-style-type: none"> <li>Nil</li> </ul>

### 24.2.1 Example: Cost / benefit evaluation

Values are in HKD dollars in thousands ('000) unless otherwise indicated  
Legend:

Costs and Savings Descriptions	5 year costs and savings projections						
	Beginning of:	Year 1.0	Year 2.0	Year 3.0	Year 4.0	Year 5.0	5-Year Total
<b>Costs</b>							
<b>Non-recurrent Expenditure</b>							
(a) Hardware	6,360.7	12,819.5	16,251.5	21,436.1	6,533.1	63,400.9	
(b) Software	12,702.0	32,322.0	22,190.9	12,073.8	72.4	79,361.2	
(c) Implementation Services	19,189.3	44,850.7	51,139.1	52,120.2	21,737.1	189,036.4	
(d) Contract Staff	0.0	0.0	0.0	0.0	0.0	0.0	
(e) Site Preparation	6,133.3	2,143.9	11,690.7	21,444.7	8,031.2	49,443.9	
(f) Communication Lines	3,251.0	0.0	0.0	0.0	0.0	3,251.0	
(g) Training Cost	166.7	2,270.8	2,600.0	2,537.5	600.0	8,175.0	
(h) Contingency	4,780.3	9,440.7	10,387.2	10,961.2	3,697.4	39,266.8	
Sub-total	52,583.3	103,847.6	114,259.6	120,573.6	40,671.2	431,935.3	
<b>Non-recurrent Staff</b>							
Sub-total	16,265.6	19,073.9	22,110.0	19,601.4	6,529.6	83,580.4	
<b>Total non-recurrent cost</b>	<b>68,848.9</b>	<b>122,921.5</b>	<b>136,369.6</b>	<b>140,175.0</b>	<b>47,200.8</b>	<b>515,515.7</b>	
<b>Recurrent Expenditure</b>							
(a) Hardware and Software Maintenance	0.0	0.0	0.0	4,574.2	17,693.1	22,267.3	Annual Recurring 20,088.1
(b) On-going System Support Services	167.0	648.8	1,257.3	3,964.6	7,061.9	13,099.6	7,061.9
(c) Communication Lines	0.0	11,736.0	11,736.0	11,736.0	11,736.0	46,944.0	11,736.0
(d) Consumable	10.0	35.0	1,455.0	2,470.0	2,090.0	6,060.0	2,090.0
Sub-total	177.0	12,419.8	14,448.3	22,744.8	38,581.0	88,370.9	40,976.0
<b>Recurrent Staff</b>							
Sub-total	5,840.2	7,291.4	9,123.9	19,220.4	30,263.5	71,739.4	30,263.5
<b>Total recurrent cost</b>	<b>6,017.2</b>	<b>19,711.2</b>	<b>23,572.2</b>	<b>41,965.2</b>	<b>68,844.5</b>	<b>160,110.3</b>	<b>71,239.6</b>
<b>Total cost</b>	<b>74,866.1</b>	<b>142,632.7</b>	<b>159,941.7</b>	<b>182,140.2</b>	<b>116,045.3</b>	<b>675,626.0</b>	
<b>Saving</b>							
<b>Non-recurrent</b>							
Cost avoidance	4,432.5	4,432.5	0.0	0.0	0.0	8,865.0	
Sub-total	4,432.5	4,432.5	0.0	0.0	0.0	8,865.0	
<b>Recurrent</b>							
Realisable savings	0.0	0.0	5,057.9	7,309.2	10,622.3	22,989.4	10,896.3
Notional savings	0.0	1,539.6	3,832.6	11,621.7	46,758.7	63,752.5	61,105.7
Cost avoidance	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sub-total	0.0	1,539.6	8,890.5	18,930.9	57,381.0	86,741.9	72,002.0
<b>Total saving</b>	<b>4,432.5</b>	<b>5,972.1</b>	<b>8,890.5</b>	<b>18,930.9</b>	<b>57,381.0</b>	<b>95,606.9</b>	
<b>Net saving</b>	<b>(70,433.6)</b>	<b>(136,660.6)</b>	<b>(151,051.3)</b>	<b>(163,209.3)</b>	<b>(58,664.2)</b>	<b>(580,019.0)</b>	<b>762.4</b>
<b>Net cumulative saving</b>	<b>(70,433.6)</b>	<b>(207,094.2)</b>	<b>(358,145.5)</b>	<b>(521,354.8)</b>	<b>(580,019.0)</b>		

## 25 TOOL 25 – GANTT CHART

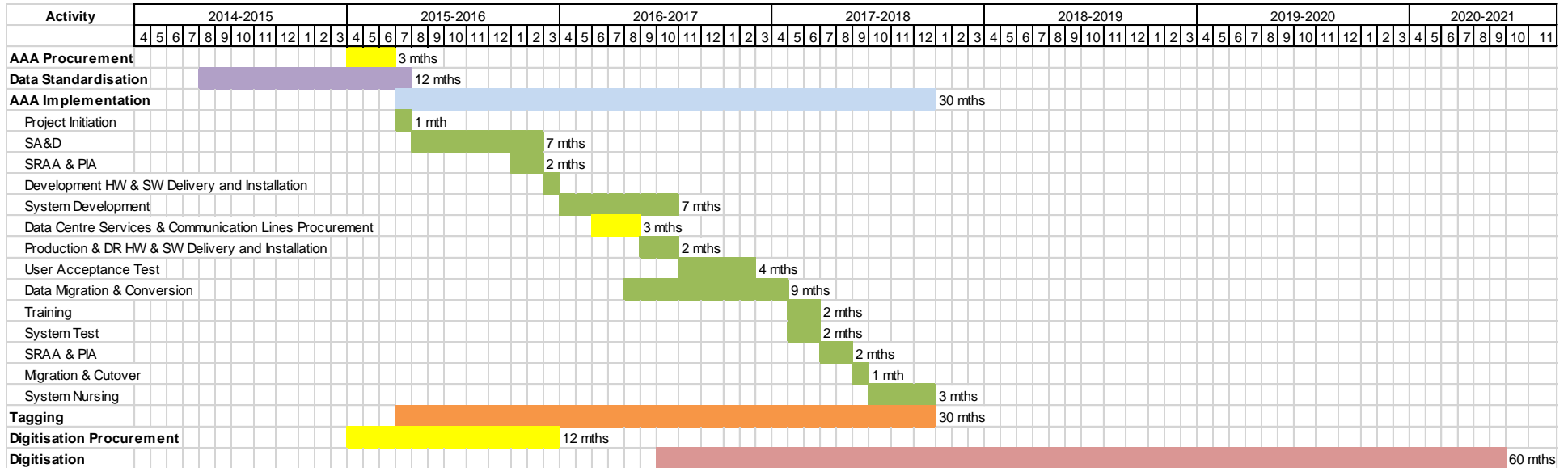
### 25.1 PURPOSE & DESCRIPTION

A Gantt chart, recognised as a project management tool, is commonly used to visualise the project schedule in a timescale. It shows the start date and end date of activities, which lays out the order in which activities need to be carried out. It may also show the dependencies between activities or be used to monitor current project status by adding a vertical 'Today' line.

### 25.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<p><b>Technical System Option</b>                      Implementation Plan</p> <ul style="list-style-type: none"> <li>• Workout Implementation Plan</li> </ul>	<p>i) Having consulted business users for their input on the Implementation Strategy and begun devising preliminary implementation milestones/ plan, a Gantt chart can be drawn up to present the proposed timeframe to stakeholders.</p>	<ul style="list-style-type: none"> <li>• Nil</li> </ul>

### 25.2.1 Example: Gantt chart



#### Key

AAA	System Name
SRAA	Security Risk Assessment and Audit
PIA	Privacy Impact Assessment
HW	Hardware
SW	Software
R	Data Recovery

## 26 TOOL 26 – DECISION CHECKLIST

### 26.1 PURPOSE & DESCRIPTION

Decision Checklist helps the project team to conduct a discussion exercise to identify any disagreements to the selected criteria chosen for the consideration factors. It helps document the selected criteria for the consideration factors and the outcomes in a structured manner.

### 26.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
The Decision Checklist is used, where applicable (i.e. Flexible adoption of processes includes Decision Tree)	i) Based on the consideration factors shown in the decision checklist, the project team can document the chosen criteria and outcomes.	<ul style="list-style-type: none"> <li>• Nil</li> </ul>

#### 26.2.1 Example: Decision checklist

Consideration Factor	
<b>Project Scale and Complexity</b>	<input type="checkbox"/> Small Project <input type="checkbox"/> Large Project
<b>Project Type</b>	Is project building a new system from scratch? <input type="checkbox"/> Yes <input type="checkbox"/> No
	Is project enhancing existing functionality in current system? <input type="checkbox"/> Yes <input type="checkbox"/> No
	Is project replacing the current system? <input type="checkbox"/> Yes <input type="checkbox"/> No



## 27 TOOL 27 – STATE DIAGRAM

### 27.1 PURPOSE & DESCRIPTION

- (a) State diagram in the UML is used to represent objects of a single class and track different states that object passes through the system during its life in response to events, along with its responses and actions.
- (b) State diagram is created to help simplify the design of algorithms for their methods. It helps to understand the dynamic aspects of a single class and how its instances evolve over time. It should not be used to show the behaviour or sequence of actions of several classes in a single or several use cases.

### 27.2 AREAS OF USE

Process / Task Where Applicable	How to Use This Tool	Alternative / Complementary Tools
<b>Requirements Specification</b> User Requirements Document <ul style="list-style-type: none"> <li>Define Future Business Process</li> </ul>	i) State diagram is used to describe the behavioral state of business components/classes across several use cases, especially for the business components/classes which are very dynamic and complex, requiring a good understanding of their states over time and events triggering changes.  ii) State diagram shows the change in state, not the processes that create those changes. For the processes which are executed in the system and change the state of objects, they can be illustrated by the diagrams such as Flowchart Diagram, Use Case Diagram, and Activity Diagram.	<ul style="list-style-type: none"> <li>Flowchart Diagram <i>and/or</i> Use Case Diagram <i>and/or</i> Activity Diagram <i>and/or</i> Sequence Diagram</li> </ul>
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>Design Application</li> </ul>		<ul style="list-style-type: none"> <li>Activity Diagram <i>and/or</i> Sequence Diagram</li> </ul>
<b>System Specification</b> System Design <ul style="list-style-type: none"> <li>Design Integration</li> </ul>		<ul style="list-style-type: none"> <li>Class Diagram <i>and/or</i> Sequence Diagram</li> </ul>

### 27.2.1 Example: State Diagram for Membership Application

