CHAPTER 2

THEORETICAL FOUNDATION

In this thesis, there are many technical terms which will be used over and over all the way through it. This theoretical foundation will assist the reader to be more familiarized to the computer-related vocabulary.

2.1 Theoretical Foundation

2.1.1 Organization definition

Organization is a structure designed to coordinate the efforts of many to accomplish shared goals [2]. We are using organizational chart to illustrate the reporting lines and official communications network within the organization.

2.1.2 Data definition

Data are raw facts, or a brief images of things, occurrences, behaviors, and transactions that are captured, recorded, stored, and grouped, but not organized to convey any specific meaning [4].

Furthermore, if you can manage the data properly, it could transform into information. Information is a collection of facts (data) organized in some manner so that they are meaningful to recipient [4].

Moreover, useful information, if being handled properly, would result in knowledge. Meanwhile, “data”, in general meaning is a form of information storage which needs to be interpreted in order to produce information. The data itself does not

have any meaning, but if being related and interpreted in right ways, it can form knowledge [13].

2.1.3 System Definition

A system, in an ordinarily, can be defined as a relationship of interrelated and joined components [7].

Meanwhile; a system is a group of interrelated elements that works together to achieve some designated goals. Even when each component is well-designed, efficient, and simple, the system will malfunction if the components do not work together. Further, a change in one component may affect other components [5].

2.1.3.1 System Development Process

System development process is a collection of many elements, such as activities, methods, best practices, deliverables, and automated tools to develop and maintain information systems and software [7]. The process of system development typically includes the following steps:

1. System initiation. The objective of this step is to define initial project scope, goals, schedule, and budget.
2. System analysis. The objective of this step is identify and analyze problems of the existing system, identify requirements of new system, analyze solution candidates, and to select the best solution to be implemented. There are five approaches in system analysis:

- Model Driven Analysis approach. This approach pictures the system models into document, validate it, and then propose the new system using blueprint to design and construct it.
- Accelerated Systems Analysis approach. This approach uses prototypes to identify the new requirements of the new system.
- Requirements Discovery Methods. This approach identifies the problem of the system and process the solution requirements derived from what the user requires.
- Business Process Redesign Method. This approach dramatically changes the existing fundamental business process.
- Agile Method. This approach is the integration of all the other approaches.

3. System design. The objective of this step is to build the computer based solution based on the specification resulted from the previous step.
4. System implementation. The objective of this step is to implement the system's physical design using selected technology, to install the newly build system, test and deliver the new system.

2.1.3.1.1 Types of System Development Process

There are two main system development approaches, as follows:

- Sequential. This approach comprise of a series of simple processes that need to be implemented sequentially. For instance, a waterfall development method, as described in figure 2.1. However, some project sequence relationships cannot always be specified in the planning stage, but are

conditional on the outcome of previous activities. Thus, there comes the new approach, namely Iterative [25].

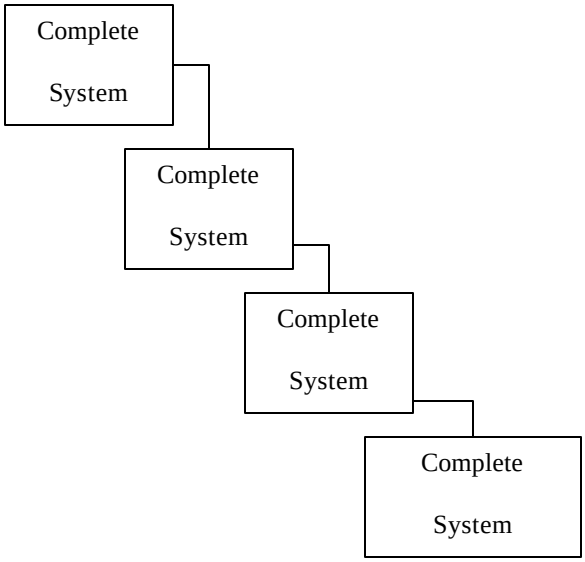

Figure 2.1 Sequential strategy[8]

- Iterative or incremental. This approach requires that analysis, design, and implementation need to be finished up partially up to a certain level before the next iteration will be implemented until there are no more additional steps required.

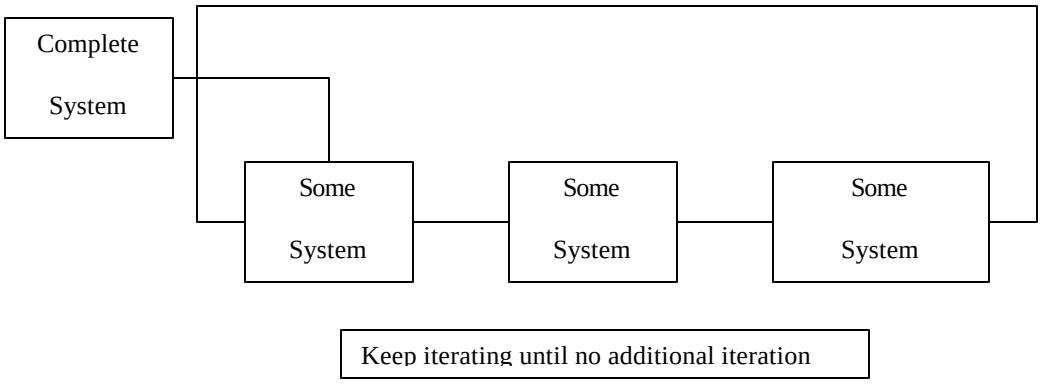

Figure 2.2 Iterative or Incre mental strategies [8]

2.1.4 Business Process definition

Business is an organizational entity that set up the resources to supply customers with their preferred products or services[10].

Subsequently, a process is[11]:

1. A natural phenomenon marked by gradual changes that lead to a particular result
2. A natural continuing activity or function
3. A series of actions or operations conducing to an end

Thus, a business process is collaboration of interrelated works and buffers with a pre defined boundaries and prerequisite relationships which convert the inputs into outputs to satisfy the customer requirements by employing all of the resources.

2.1.5 Information Management

In order to have useful information, information has to be managed wisely, so that it could help people in various aspects, such as business, science, and so on. On the other hand, there is an argument which stated that Information System (IS)[7] is a collection of processes, stores, analyzes, and disseminates information for a specific purpose. While, if the system is combined with technology, it would result in Information Technology (IT) - collection of the individual technology components that are typically organized into computer-based information system. IT and IS are very well related, it can not be too far separated from each other. Furthermore, IS [4] is the collection of information technology, procedures, and people responsible for the capture, movement, management, and distribution of data and information. As with any system, any crucial

that the components of an IS work together. That is, the components must be consistent, minimally redundant, complete, and well connected with each other.

In order to be able to compete successfully in the modern business environment, organizations expect their information systems to have many powerful capabilities:

- Provide fast and accurate transaction processing.
- Provide large capacity, fast-access storage
- Provide fast communications (machine to machine, human to human)
- Reduce information overload
- Span boundaries
- Provide support for decision making
- Provide a competitive weapon

Information system can be grouped into several classifications. Information system can be grouped by its services, which are namely[7]:

- Front-office Information System is an information system that supports business functions that extend out to the organization's customer. For example, handles the registration for an event.
- Back-office Information System is an information system that supports integral business operations of an organization, as well as reaches out to suppliers. For instance, generating reports to analyze, then forecast the upcoming business needs.

Meanwhile, information system can be differentiated by its users that are namely [2]:

- Single user system is a system in which at most one user can access the database at any given time.
- Multi-user system is a system in which many users can access the database at the same time.

Based on its classification, information system can be grouped into four, they are namely[7]:

- Transaction Processing System,
- Management Information System
- Decision Support System
- Executive Information System.

2.1.6 Unified Modeling Language(UML)

Unified Modeling Language (UML) is an approach, collection of techniques, and notations to support the software development process which combines some procedures and philosophies[18]. There are nine diagrams involved in the UML:

- Use-case Diagram
- Class Diagram
- Object Diagram
- Sequence Diagram
- Collaboration Diagram
- Statechart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

2.1.7 Database Theory

Database is a collection of data items related to some enterprise [9]. There is also an argument which stated that database is a collection of related files, and where those related files are located can greatly affect user accessibility, response times, data entry, security, and cost[3]. With database approach, all the data are typically stored in the same storage location, rather than spread in many places across the organization.

In general, the data in the database will be both integrated and shared [3].

- Integrated the database can be thought of as unification of several otherwise distinct files, with any redundancy among those files.
- Shared means that individual piece of data in the database can be shared among different users, in the sense of those users can have access to the same piece of data, possibly for different purposes.

2.1.7.1 Database types

In general, database can be categorized into two types [3]:

- Centralized database; all the related files in one physical location. With it, files can generally be made consistent with each other when they are physically kept in one location because file changes can be made in a supervised and orderly fashion. Also, recovery from disasters can be more easily accomplished at a central location. However, centralized database are vulnerable to a single point of failure. This kind of database is appropriate for small companies, as pictured in figure 2.3.

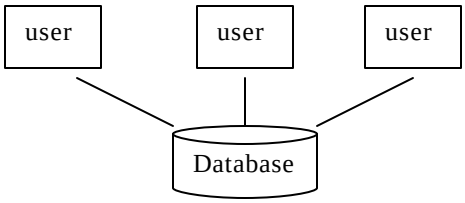

Figure 2.3 Centralized database

- Distributed database; complete copies of a database, or portions of a database, in more than one location, which is usually close to the user. This kind of database is great for a large companies or organizations with geographical difficulty, as pictured in figure 2.4.

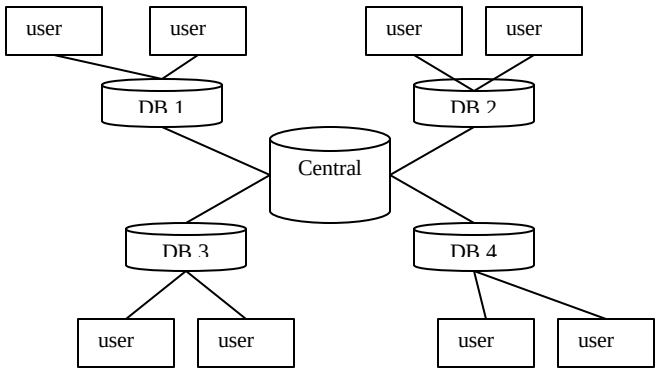

Figure 2.4 Distributed database

2.1.7.2 Database access

Database access is classified into two categories[9]:

- Query is a request to retrieve data.
- Update is a request to insert, delete, or modify existing data items.

2.1.7.3 Database Model

Considering the characteristics of the data and how the data will be accessed; there are three types of database model[2]:

- Hierarchical Database Model. It structures the data into a shape like a “tree”, just like an organizational chart, with the parents and child key, or main and subordinate key.
- Network Database Model. Organizes the data into linked-list, on which a record can be linked to more than one data element, like a record can own many members.
- Relational Database Model. A combination of tables related and organized to simply connect the records in a way that have not been predefined before (opposite to Hierarchical and Network Database Model).

2.1.8 Web Applications

Web is a system with common principles to store, retrieve, format, and display all types of digital information, including text, hypermedia, graphs, and sounds [4]. Moreover, it uses graphical user interfaces, so it is very easy to use.

The World Wide Web is an ever changing, kaleidoscopic collection of hundreds of millions of documents, all of which reside someplace on the Internet and are written in some form of HTML [23]

Application program is a machine based language which can only be read by itself, to define what a software process must do or how to do its job[7].

There are four analysis activities in modeling a web application[15]:

1. Content analysis: recognize the content supplied by the web application, including text, graphics and images, and video and audio data
2. Interaction analysis: describes the user interaction with the web application
3. Functional analysis: identify the operations to be applied by the web application and other processing functions essential to end user.
4. Configuration analysis: describes the environment and infrastructure in which the web application exists in.

Based on its architecture, Web application can be separated into two categories:

1. Two tier architecture. This is the type where the application logic resides in the client or personal computer, with the first tier as the client, and the second tier as the database.
2. Three tier architecture. This is the type where the application logic is separated from the client or personal computer. In this architecture, the first tier is the browser, the second tier is the application logic, and the third tier is the database.

2.1.9 Local Area Network (LAN)

LAN is a communications network that allows information exchange among those devices. The scope is small, typically a single building or a cluster of buildings[19]. Moreover, it is owned by a same organization that owns the attached devices.

2.2 Theoretical Framework

2.2.1 Transaction Processing System (TPS)

Transactions is occurrences that happen real time by the program when there is real-world event take place. For example, when a customer withdraws money in an ATM, the credit in his/her account will be directly deducted[9].

TPS is a system which monitors, collects, stores, process, and distributes the organizations' general business transaction. In the real world, when the number of processing increases, manual processing will no longer be able to handle them all[7]. TPS assists to replace the work of manual processing.

2.2.2 Iterative Approach

In comparison with the sequential approach, the iterative approach is more suitable since it allows the developer to develop many versions to be selected by users. On the other hand, the sequential approach, such as the waterfall method, does not accommodate the change easily. This is why this thesis will use iterative method in developing the system.

The main processes that need to be implemented during initial analysis of the incremental approach are scoping the problem and identifying major requirements[16]. The purpose of the requirement identifications is to assist client in selecting the main focus of system development during the incremental

2.2.3 PIECES Framework

James Wetherbe had invented a useful framework called PIECES to organize problems[7]. Each of the letters stands for:

- P is for the need to correct or improve performance, such as the system response time, and throughput.
- I is for the need to correct or improve information and data, this includes output, input, and data storage (check whether it is redundant or unorganized)
- E is for the need to correct or improve economics, control costs, or increase profits.
- C is for the need to correct or improve control or security, whether it is too much or too little
- E is for the need to correct or improve efficiency of people and processes, to people and to machines.
- S is for the need to correct or improve service to customers, suppliers, partners, employees, and so on, to provide consistency, reliable and compatible information.

2.2.4 System Development Life Cycle (SDLC)

SDLC is an ordered framework that consists of sequential processes by which information systems are developed[3]. The steps of processes are:

1. System Investigation is the process of understanding business problem, technical options for systems, and problems that are likely to occur during development phase.
2. System Analysis is the assessment of the business problem that planned to be solved with an information system.
3. System Design describes how the system will accomplish this task. It encompasses two major aspects of the new system:

- Logical systems design states what the system will do
- Physical systems design states how the system will perform its functions.

4. Programming is the conversion of the design specification into computer code.
5. Testing is designed to track errors in the computer code.
6. Implementation is the process of converting form the old system to the new system.
7. Operation and maintenance is the process to determine if it is being used correctly, and it continuity throughout the life of the system.

2.2.5 Unified Modeling Language (UML) Diagrams

This thesis will use four out of nine UML diagrams, which are:

2.2.5.1 Use-case diagram

Use case is a model diagram which depict the interactions between the system and external systems and users[7].

The purpose of developing use cases is as follows[15]:

1. To help the developer to understand how users perceive their interaction with the web application
2. To provide the detail necessary to create an effective analysis model
3. To help compartmentalize the work
4. To provide important guidance for those who must test the web application

2.2.5.1.1 Use case diagrams' notations

- Actors; symbolize the roles that people, other systems, communicates with the system.

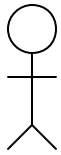

Figure 2.5 Actors

- Use case; defines what the system can do.

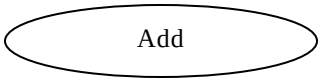

Figure 2.6 Use case

- Communication association; defines the interaction between user and system.

Figure 2.7 Communication association

- System or Subsystem boundary; defines the limitation of working area of a system.

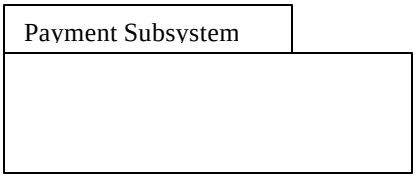

Figure 2.8 Subsystem

2.2.5.2 Activity diagrams

Activity diagram is a diagram that can be used to shows the business process flow, the use case's steps, or object's behavior.

2.2.5.2.1 Activity diagrams' notations

- Solid dot; represents the start of the process

Figure 2.9 Solid dot

- Rounded-corner rectangle; defines an activity or task that needs to be performed

Figure 2.10 Rounded-corner rectangle

- Arrows; shows triggers that initiate activities

Figure 2.11 Arrows

- Solid black bar; represent activities that can occur in parallel

Figure 2.12 Solid Black Bar

- Text; shows a trigger as result of a decision activity

- Diamond; depicts decision activities

Figure 2.13 Diamond

- Solid dot inside a hollow circle; represents the termination of the process

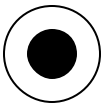

Figure 2.14 Solid dot with hollow circle

2.2.6 Model Driven Approach

A model is a symbol of reality. Thus, system model is a picture of a system that represents reality[7].

2.2.6.1 Model Driven Analysis

Model driven analysis is a problem solving approach that emphasizes the drawing of pictorial system models to document and validate existing and / or proposed systems[7]. Ultimately, the system model becomes the blueprint for designing and construction an improved system. Examples of model driven analysis:

- Structured – a process centered techniques used to either analyze an existing system, define business requirements for a new system, or both.
- Information Engineering and Data Modeling is a model driven and data centered, but process sensitive, technique for planning, analyzing, and designing information systems.

- Object Oriented - a model driven technique that integrates data and process concerns into constructs called objects.

This thesis will put emphasize on object oriented approach.

2.2.6.1.1 Object Oriented Approach

In 1990 once stated that object is an abstraction of something in a problem domain, reflecting the capabilities of the system to keep information about it, interact with it, or both. This may not immediately appear to help very much, as parts of the definition are themselves a little obscure and raise further questions[13]. Thus, in 1997, it was then expanded into concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand[8]. Object serves two purposes:

- Promote understanding of the real world.
- Provide a practical basis for computer implementation.

Object oriented approach provides a mechanism for mapping from real-world problems to abstractions from which software can be developed effectively[18].

Later then comes the Object-oriented programming (OOP) languages; the idea of taking a small amount of data and the instructions about what to do with that data, and putting both of them together into what is called an object [3].

There are two advantages using object oriented method:

- Objects are reusable. It can be reused in multiple information systems and applications. Reusable means that classes created for one purpose can be used in a different object-oriented program if desired.
- Objects are extensible, they can be changed or expanded easily without adversely impacting any previous applications that used them.

2.2.7 Requirements Discovery Method

Requirements discovery identifies the user requirements from the user community[7]. The most common requirements discovery approach is the fact finding techniques, which includes:

- Observations on the current system
- Spreads the questionnaires to the user community
- Interviews the users and staffs

2.2.8 Database Management System

Database Management System is software designed to assist in maintaining and utilizing large collections of data [14].

Using a DBMS to manage data has many advantages:

1. Data independence: application programs should not, ideally, be exposed to details of data representation and storage. The DBMS provides an abstract view of the data that hides such details. There are two types of data independence:
    - Logical data independence: where users can be shielded from changes in the logical structure of the data, or changes in the choice of relations to be stored. It hides details, such as how the data is actually laid out on disks; file structure, and the choice of indexes.
    - Physical data independence: the conceptual schema insulates users from changes in physical storage details.

2. Efficient data access: DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This is especially important if the data is stored on external storage devices.
3. Data integrity and security: if data is always accessed through the DBMS, the DBMS can enforce integrity constraints.
4. Data administration: when several users share the data, centralizing the administration of data can offer significant improvements.
5. Concurrent access and crash recovery: DBMS schedules concurrent access to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users form the effects of system failures
6. Reduced Application Development Time: Clearly, the DBMS supports important functions that are common to many applications accessing data in the DBMS. This, in conjunction with the high-level interface to the data, facilitates quick application development. DBMS applications are also likely to be more robust than similar stand-alone applications because many important tasks are handled by the DBMS.

However, it also has some drawbacks, such as:

1. DBMS is a complex piece of software.
2. DBMS is optimized for certain kinds of workloads.
3. Its performance may not be adequate for certain specialized applications. For example: application with tight real-time constraints.
4. Some applications may need to manipulate the data in ways not supported by the query language.

DBMS provides facilities such as follows [17]:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Access Control
- Security System, which prohibits unauthenticated user to access the database
- Integrity System, to preserve the consistency
- Concurrency Control System, to allow the use and access of multiple users
- Recovery Control System, which permits the database to do the data restoration when some errors occurs on the software or hardware.
- User Accessible Catalog, which has all the description of the data stored in the database.

2.2.8.1 Data Definition Language (DDL)

DDL is the language that allows the database administrator or users to describe and name the entities, attributes, and relationships that is needed by the application [18].

2.2.8.2 Data Manipulation Language (DML)

DML is the language that permits the data operation and supports simple data manipulation on the database [18]. The operation on DML includes:

- inserting new data into the database
- modifying the existing data in the database
- retrieving the existing data in the database
- deleting the data from database

2.2.8.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram allows us to describe the data involved in a real-world enterprise in terms of objects and their relationships and is widely used to develop initial database design [14]. It provides useful concepts that allow us to move from an informal description of what users want from their database to a more detailed, precise description that can be implemented in a DBMS

2.2.8.3.1 ERD Notations

- Entity; a class of persons, places, objects, events, or concepts about which we need to capture and store data

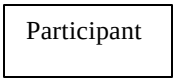

Figure 2.15 Entity

- Attributes; descriptive property of an entity

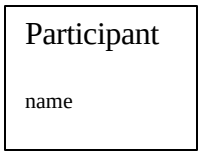

Figure 2.16 Attributes

- Cardinality; number of occurrences of one entity, there are five variance of cardinality

- Exactly one

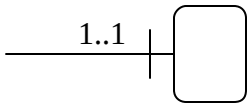

Figure 2.17 Exactly one cardinality

- Zero or one

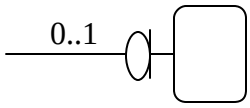

Figure 2.18 Zero or one cardinality

- One or more

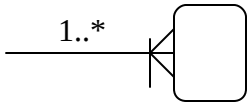

Figure 2.19 One or more cardinality

- Zero, one, or more

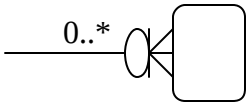

Figure 2.20 Zero, one, or more cardinality

- More than one

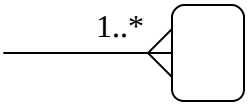

Figure 2.21 More than one cardinality

2.2.8.4 Stored Procedure

Stored procedures are trusted application code that is registered with the DBMS and can be called from SQL statements [14]. There is also an argument which define stored procedures as a subprogram that can take parameters and be invoked. It can be encapsulated as a function [18]. The difference between a procedure and a function is that a function will always return a single value, while stored procedure could return many values.

2.2.8.5 Trigger

Trigger is a program embedded within a table that will automatically updates the other tables when it is executed [7]. There is also exist an argument which defines trigger as an action that databases do when there is an event occurs in the application [18].

2.2.8.6 Relational Database Management System(RDBMS)

Data structure in the RDBMS consists of:

- Relationship; is the name of a two dimensional tables [6].
- Domain; is a value set which can be used by one or more attributes [18].
- Tuple; is the row in a relation [18]
- Degree; is the total attributes within the table [18]
- Relational Datbase; is the collection of normalized relation with different names of relations.
- Key; is a minimal set of attributes whose values uniquely identify an entity in the set [14].

- Primary key: unique key to identify which tables does it belong to [14].

2.2.8.7 Relationship

Relationship is a table with rows and columns within it [18]. There are three types of relationships:

- One-to-one relationship (1:1); is a relationship where each existing entity can only have the maximum number of one relation to the other entities.

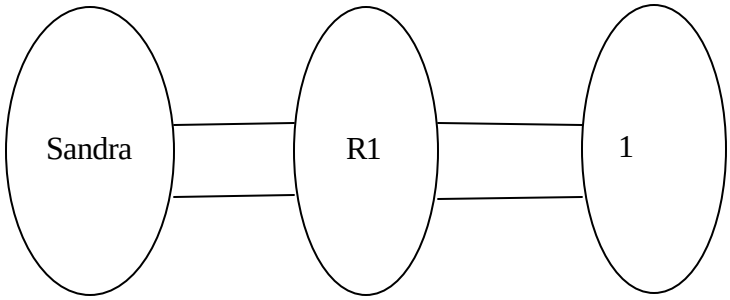

Figure 2.22 One-to-one relationship

- One-to-many relationship(1:*); is a relation where each existing entity can have one or many relationships to the other entity.

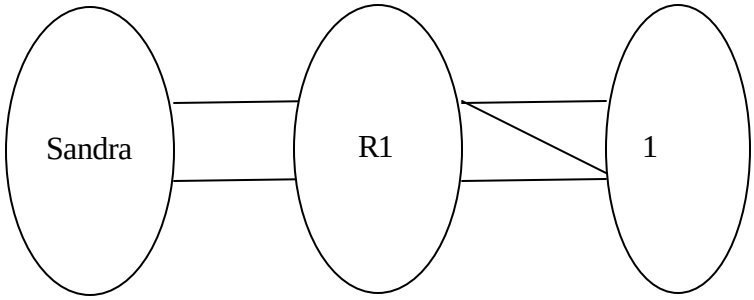

Figure 2.23 One-to-many relationship

- Many-to-many relationship (*:*); is a relation where each existing entity can have one or many relationships to the other entity.

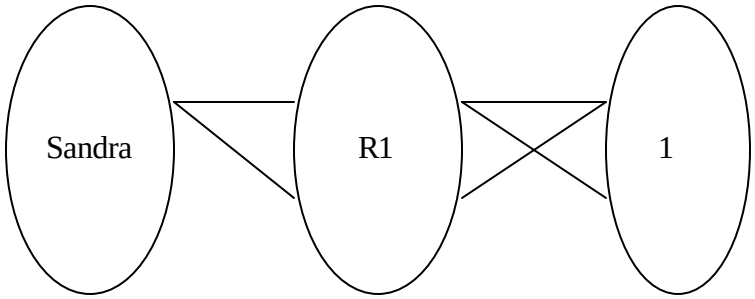

Figure 2.24 Many-to-many relationship

2.2.8.8 Cardinality and participation

Cardinality defines the maximum occurrence relationships exist in an entity that participates in a relation [18]. Meanwhile, participation determines whether only some or all occurrences are participating in a relationship.

2.2.8.9 Normalization

Normalization is techniques that produce a collection of relationship with the desired property, with the given data [18].

The following is the normalization processes:

1. Unormalized Normal Form(UNF); is a table with one or many repeating groups.

2. First Normal Form(1NF); is a table where the repeating groups or derived attributes has been eliminated. Thus, in a 1NF table, each row and column only have one value.
3. Second Normal Form(2NF); is a table in 1NF where each non primary key's attribute is fully functional dependent on the primary key.
4. Third Normal Form(3NF); is a table in 1NF and 2NF where each non primary key's attribute is transitively dependent on the primary key.

2.2.9 Database Life Cycle

Database application life cycle is the most important component in database system, since it is related to the existing system information. The steps of the database life cycle are as follows:

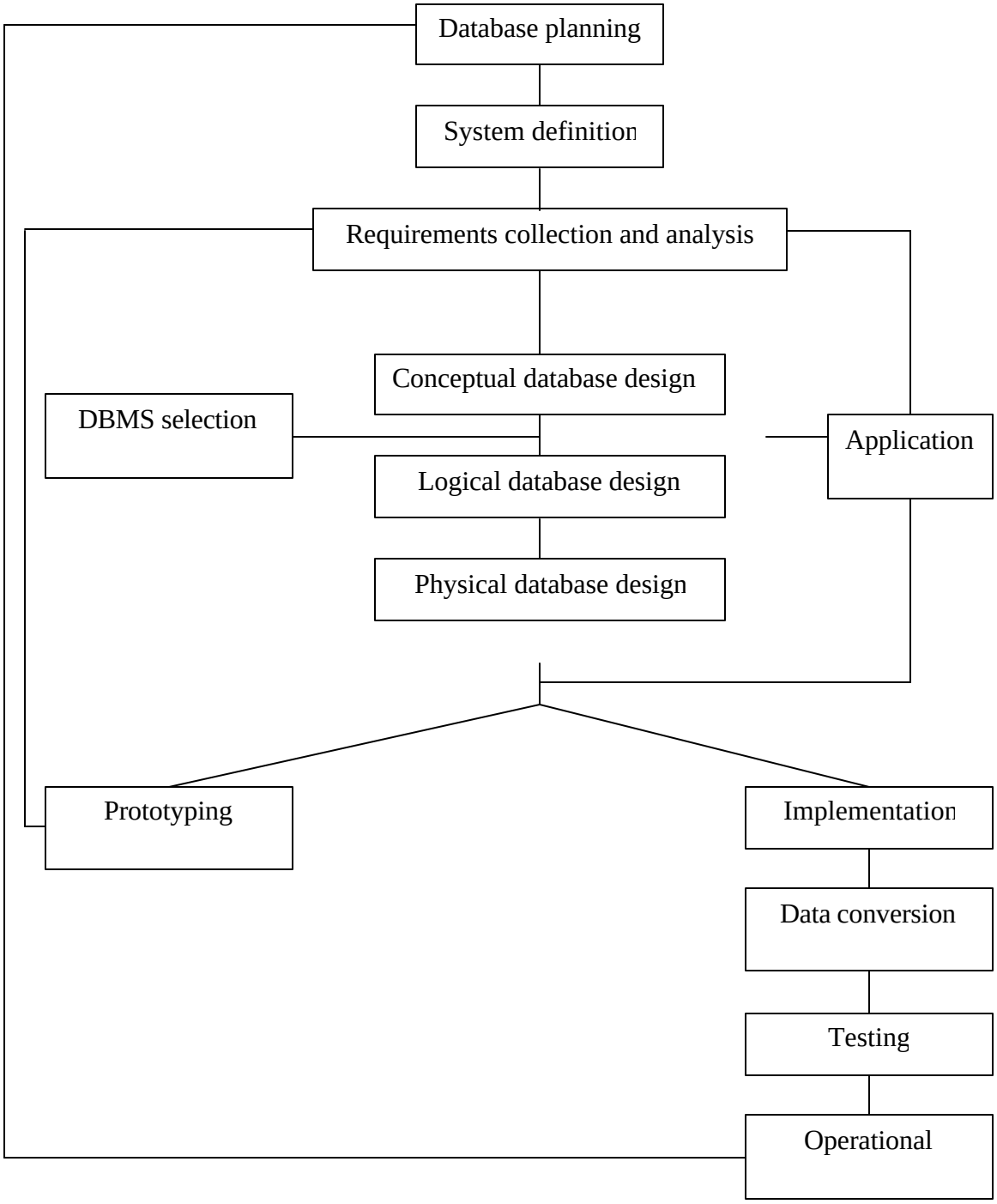

Figure 2.25 Database Application Life Cycle, Connolly and Begg, [18]

The explanation about the database life cycle above is as follows:

1. Database Planning; are management activities which plan the database life cycle so that it can be brought into reality efficiently and effectively. There are three important things in formulating information system and database planning:
    - Identify the plan and objectives of the organization with the information system needed
    - Evaluate the existing information system, the strengths and weaknesses
    - Assess the opportunity of information technology which allows the competitive benefit.
2. System Definition; defines the scope and boundaries of database application and view of the primary user's view.
3. Requirement Collection and Analysis; are the requirements gathering and information analysis on part of the organization that will be supported by the database application and use this information to identify the needs of the new system's user.

The information gathered from each user's view is:

- Description on how the data will be used or made
- Details on how the data will be used or made
- Additional requirements for the new database application

4. Database design; is the process of designing the database that will support the organization. There are two approaches that are normally being used:

- Top-down approach: starts with the data model development that contain several high level entities, and then identify the low level entity, relationship, and related attributes.
- Bottom-up approach: starts with the basic attributes, in which will then be grouped into relationships which represents the entity's type and the relation between entities.

5. DBMS selection; is the process of selection on which DBMS that will support the database application. There are several factors that could take into considerations:
   - Ease of use
   - Reliability
   - Compatibility
   - Security
   - Vendor stability
   - Open for future development
   - Appropriate system requirements
6. Application Design: is the design of user view in using and processing data. Application design will then be break down into two general aspects:
   - Transaction Design; that consists of three primary transactions:
   - Retrieval action; is to retrieve the existing data to be displayed.
   - Update action; is to insert, delete, and change the existing record.
   - Mixed transaction; is the combination of retrieval and update action. For instance, retrieve a data, and then update its value.

- Interface Design; the most outer look of the application, which will interact with user.

7. Prototyping: is the process in building a working model of database application. The purpose of building a prototype is as follows:
   - Identify the system is working properly
   - Give additional features on the system
   - Clarify the user's need
   - Evaluate the feasibility of special system design

   Despite the advantages, the disadvantages of using prototype is that there are often found that prototyped systems had less efficient performance [24].
8. Implementation; is the physical realization on database and its application design.
9. Data conversion and loading; is transferring all the data from the old system into the new database to be applied in the new application.
10. Testing; is the process of executing the application program with the intention of finding the errors.
11. Operational Maintenance: is the process of monitoring and maintaining system after the installation phase. This step requires:
    - Monitoring the system performance; when it is at the low level, then system tuning must be made.
    - Upgrading, when necessary, new requirements could be inserted into database application by re-do the previous life cycle steps.

2.2.10 Conceptual, Logical, and Physical Database Design Methodology

A structured approach which is using procedures, techniques, tools and documentations in order to support and facilitate every design process is considered as design methodology [18].

There are three primary phases in designing a database [18]:

a) Database Conceptual Design

Database conceptual design is the process in building an information model which will be used by a company, apart from every physical consideration [17]. The steps are as follows:

Step 1: Construct a Database Conceptual Model for every view. The aim is to build a local conceptual model on that company for every view.

1.1 Identifying entity type

The aim of this step is to identify the primary entity type which is needed for every view. The method to identify primary entities is by checking the references needed by users [18].

1.2 Identifying relationship type

The aim of this step is to identify the relationships on the identified entities. Normally, relationship could be obtained from a verb that relates the two attributes, for example: Employee *works in* Department.

1.3 Identifying and relating attributes with entity or relationship type

In this part, we relate the attributes with the entity, with the correct relationship type. Attribute could be identified by a non which is a part of identity, and characteristic of an entity.

After the attributes has been identified, a document is needed for every existing attributes, as followed:

- Attribute names and descriptions
- Data type and length

1.4 Determining the attribute domain

Domain is a value container, where one or more attributes contained in it. For example is the HomePostcode domain, it will be numbers, with the length of 5.

1.5 Determining the Candidate Keys and Primary Key in attributes

Candidate key is a collection of minimum attributes on an entity which uniquely identify every occurrence on that particular entity. This key could be more that one.

Meanwhile, a Primary Key usually is chosen from one of the existing candidate keys.

b) Logical Database Design

A database logical design is a process in building information needed in a company based on one specific data model, apart from DBMS or other physical considerations.

Eliminates the incompatible features

This step is an adaptation from the conceptual model so that it can be easily used by the system: eliminates many-to-many relationship.

Logical design relationship is copying the primary key in a particular entity to be used as foreign key to another entity

c) Physical Database Design

Physical database design consists of base relations, file organizations, and indexing to get the efficient access to the data, which includes:

- Designing the base relations
- Designing the representation of derived attributes
- Transaction analysis
- Deciding the implemented file organization
- Indexing

2.2.11 Three Tier Architecture

Three-tier architecture separates the application logic from the clients [15]. There are three layers in three tier architecture:

1. Presentation layer: users require a natural interface to make requests, provide input, and to see results. The widespread use of the Internet has made Web based interfaces increasingly popular.
2. Middle tier: the application logic executes here. An enterprise class application reflects complex business processes, and is coded in a general purpose language, such as C++ or Java
3. Data Management tier: data-intensive web applications involve DBMSs.

2.2.12 Platform

2.2.12.1 .NET framework

Microsoft's .NET is a platform for XML web services which basically integrates the web sites and programs to deliver application[3]. The .NET framework allows unrelated Web sites to communicate with each other and with programs that run on

personal computers. .NET means that one click could set off a cascade of actions without requiring the user to open new programs or visit additional web sites.

2.2.12.2 C#

C# (pronounced C-sharp) is Microsoft's object-oriented language that enables programmers to quickly build a wide range of applications for the .NET platform.

2.2.12.3 SQL Server

SQL server is a Database Management System which ensures the security and reliability for enterprises, and can simultaneously run several databases