Chapter 2
Theoretical Foundation

2.1 General Theories

2.1.1 Management

Nowadays, all aspects of lives in the world have developed rapidly due to the robust development in the field of information technology and the effect of globalization. The role of human being is key to the revolution of information technology, hence increasing the quality of human resource is vital. To attain the quality human resources can be approached through adequate education or training. Education is required in order for human resources to keep up with the necessary skills and knowledge as well as for the human resource to be competitive. In an organization setting, competitive human resources are important in ensuring that the organization is managed properly so that the organization will be leading in the industry as compared with competitors. A well-managed organization is strategic in facing the competitive business world. it will be crucial to take advantage of good management practice as part of the process in managing an organization to achieve the goals.

Based on an English dictionary, management is the act, step, or manner of how to managing; handling, direction, or control [10]. This means that management represents a process involving human beings that are equipped with skill and knowledge to perform all the actions as mentioned in the definition.

Management element always stresses in the formulation of or definition of management business, as mentioned by some experts, for example Henry Fayol states as follows “To manage is to forecast and to plan, to organize, to command, to coordinate, and to control”. Similar statement is also stated by E.F.L Brench “Management is concerned with seeing that jobs gets done: its task are centered on planning and guiding the operations that are going on in the enterprise” [11]. Therefore, it is relevant that when managing a business organization, the concept of ‘management’ should be applied accordingly.

Furthermore, according to a business dictionary, management is organizing and coordinating a company's activities to achieve defined objectives [12]. All of the actions of coordinating and organizing the people, products, or market that has an impact on achieving to the goals of the company is also defined as managing a business.

Other expert defines management as a process of strategy, organizing, directing, and supervising that uses the potential resource to achieve the goal or objective that have been set through the resource nearby such as other people or worker [13].

2.1.1.1 Function of Management

Henri Fayol defines five management functions for the management component, which are still considered relevant for companies today. These five functions focus on the relationship between staff and their management, and provide reference points to solve the problem in a creative way [14]. The picture below is the flow of the function of management and also the explanation of it.

Figure 2.1 Five Functions of Management (Fayol) [14]

1. Planning

   According to Henri Fayol, drawing up and making a good plan is the hardest amongst the other five functions of management. This function needs an active participation in the organization. This will take time to implement and the plan must be linked to one and to another coordinated on different levels. Planning function must take the flexibility of the member into consideration as the guarantee continuity.

2. Organizing

   An organization can only go well if the management is well-organized which means to build a good organization structure and run the organization smoothly, it needs adequate capital, staff, and raw materials. The structure of the organization with a good division of functions and tasks is essential. When it comes to the growth of task, the organization will expand horizontally and vertically.

3. Commanding

When orders are given and clear working instructions are given, employees will know exactly what they need. Returns from all employees are optimized if specific instructions are given regarding the activities to be carried out by them. A good manager is capable to communicate, have integrity, and motivate the team to take initiative.

4. Coordinating

The organization will work better when all activities are harmonized. Positive behavior by influencing the employee is important in the organization. Coordination function aims at stimulating motivation and encouraging the discipline within the group dynamic. All of this needs a good leadership that can make clear communication to the member. With good behavior in the organization, the objective can be achieved.

5. Controlling

By checking whether everything is going according to the plan, the organization must control the activities inside the management. Controlling has four steps process:

- Establish standards for performance based on organizational objectives
- Real performance measurement and report
- Compare results with standards and performance
- Take corrective or preventive action as necessary

2.1.2 Fixed Asset

Fixed asset refers to tangible property that an organization have and are used to support an organization in running the business. But are not to be converted into cash in a short-term period [15]. This is clear that any organization running a business needs fixed assets beside other resources that are necessary for ensuring the business to be run according to whatever the mission, the goals and the vision that organization has. This also implies that without fixed asset, an organization will have difficulties in achieving the goals that have been set which is to gain profit.

2.1.2.1 Types of Fixed Asset

There are many types of fixed assets in a company that runs business either it is a small scale of business or a large scale of business. The following are the breakdown of fixed assets that usually used in large scale company [16]:

- Building
  This asset includes the cost of acquiring the building, the construction and all facilities owned by the entity.

- Computer Equipment
  This asset includes types of computer equipment such as computers, laptops, routers, server, and back-up generators.

- Software
  This asset includes only the most expensive types of software such as accounting software or enterprise resources planning software.

- Construction in progress
  This asset calculates the cost of the construction in progress, once it is finished it will move to fixed asset.

- Furniture and fixtures
  This asset includes storage warehouses, chairs, tables, desks, and filling cabinets.

- Intangible asses
  This asset includes all non-physical assets such as trademarks, patented technologies, broadcast rights, and copyrights.

- Land
  This asset includes the purchased of the land and the cost of land improvements.

- Leasehold improvements
  This asset includes the improvement of leased space such as air conditioning and telephone wiring.

- Machinery [17]
  This asset includes to a production machinery.

- Office equipment
  This asset includes equipment as copiers, video equipment, and printers.

- Vehicles
  This asset includes company cars, trucks and fork lifts.

Aside from the complete list of fixed assets items, fixed assets are also defined in a more general category and are usually used by a small to middle scale of company. Below are the lists of fixed assets [18]:

- Buildings
- Land
- Vehicles

- Machinery
- Furniture and fixture

The types of fixed assets owned by a company depends on the company's business fields, which means that not all companies or business have all the fixed assets in running the company. For example, a car rental company usually has significant number of fixed assets in the form of vehicles. On the other hand, a boarding house business has significant fixed assets in the form of building, land, and furniture and fixture. While a manufacturing company has machinery and equipment as the main fixed assets aside from the building and land. The number of fixed assets in a company depends on the scale of the company, the larger the scale of a company the more fixed assets it has and the smaller the scale of a company the fewer fixed assets it has.

2.1.2.2 Fixed Asset Management

Fixed assets management helps companies to manage fixed assets in a company by tracking and protecting the value of fixed assets [19]. To manage the fixed assets, it needs to keep track the key detail in the business. Below there is an image and explanation about the details of following items.

Figure 2.2 Fixed Assets Checklist [19]

- Location

  This is important to know where the fixed assets are located. The company needs to give tags of location for each item to make the control of the item easier.

- Quantity

  Keep track of how many does the company owned the fixed assets. This will help the company know when to buy more property and prevent the loss of fixed assets.

- Condition

  Give a note to the condition of the fixed assets. This will help the company with information whether the fixed assets are in good or bad condition. As a result, the company can arrange a schedule of maintenance and plan to purchase new fixed assets if necessary.

- Maintenance schedules

  The company needs to keep up with the maintenance schedule to ensure the fixed assets meet their expected economic life and to contribute service to the company.

- Depreciation status

  The company needs to keep track the depreciation of each fixed asset as part of cost allocation necessary for the purpose of profit and tax calculation for the company.

Things to be considered in controlling fixed assets, there are several procedures for the recordation of manufacturing assets as follows [20]:

1. Create a new record for the asset
2. Write a description of the asset
3. Enter tag number
4. Enter serial number of the asset
5. Note asset location
6. Assign responsibility for the asset
7. Record the acquisition date of the asset
8. Enter the cost of the asset
9. Assign the asset class
10. Enter the useful life
11. Approve the record
12. Store the record

2.1.3 Occupancy Rate

Occupancy rate is a ratio of rented space to the total amount of available space [21]. Occupancy rate is a term that is used in rental space business such as hotels, apartments, and boarding house to determine the percentage of available units. The occupancy rate can be calculated with the following formula:

Occupancy rate = Number of occupied rooms / the number of total rooms

Example: there are 200 room in total and 180 are occupied

180 / 200 = 0.9 = 90 percent occupied rate.

Occupancy rate is important for the business owner because they can identify of the success or the failure of the property [22]. Furthermore, it can be used to changes in pricing, marketing strategies, or determine the facility compared to competitors.

2.1.4 Maintenance Guest Data

Guest data is important especially in a business that involves guest as a client. One of the reasons is for security, in a business such as a boarding house the owner must know the identity of the tenants which will allow the owner to track the tenant if the tenants do bad or criminal activities. Besides that, with the guest data makes the business more organized because of the data could track whom is live in room a, b, or etc. This will also lead to the fixed assets management where the owner will know if the tenant broke the asset in their room or complaint about the asset in the room. To maintain the guest data the owner must have the guest data first, which will be collected when the potential tenant applies to the boarding house and then accepted by the owner. These are several data that need to have by the owner [23]:

- Guest name
- Address
- Email
- Phone number
- Date of birth
- Occupation
- Address of the university/office
- Picture of identity card

2.1.5 Computerized Record Keeping

The maintenance of accurate and up-to-date records is essential for the success of any company. The company must realize that records are one of the most important management tools it possesses and should therefore be given due importance. Many companies invest a lot of time in running the business without realizing the importance of maintaining good documentations [24].

In these few years, technologies have improved significantly and become more reliable, efficient, and easy to use. Many companies still use paper for the documentations especially for low to middle scale of business. The reason of that because there is significant amount of financial and operational barriers for the business in using computer generated document. Using computer for documentation has significant of benefits, there are identified as follows [25]:

- Reliable backups
- Increase security and control
- Facilitated collaboration
- Improved timeliness
- Lower archiving cost
- Improved version control
- Increase efficiency and file management
- Better searching capabilities
- More consistent content
- Improved task manager

2.2 Technologies

2.2.1 HTML

Hypertext Markup Language (HTML) is a markup language for formatting and structuring content on the website [26]. HTML uses ASCII markup tags to enrich the design of 20 websites with lists, paragraphs, option of fonts, images, and hypertext links to other Internet documents. The HTML5 is the latest generation of HTML, it has new features and APIs to help developers provide the user with dynamic interactions. To tell the browser how to structure the website, HTML uses markup tags. The tags mark parts of the page as headings, lists, paragraphs, etc.

2.2.2 CSS

Cascading Style Sheet (CSS) is a language that defines how the elements should be displayed to create the presentation for the HTML documents. CSS provides a great appearance for creating and designing websites [27]. Some of the features in CSS offers are manipulation of padding, border, margin, background, font, and many more.

2.2.3 React.js

React JavaScript is a user interface library. Facebook creates React in 2013. Reacting is declarative, effective, and flexible. React is the web application stack view only. The Model and the Controller are not available [28].

React is a library that is based on components. This means that each user interface created in React is a component format. A component can be reused and structured into 22 relationships between parent and child. Components communicate through "props," known as parameter to their child's component [29]. React uses JavaScript's full power to render view by using the component.

React also gives developers an option to create component using JSX. It's more readable and more natural because JSX contains HTML syntax [30].

2.2.4 Node.js

Node.js is a JavaScript runtime environment based on the Chrome V8 JavaScript engine, which uses JavaScript to process an event- driven, asynchronous (or non- blocking I / O) approach on the server side [31]. For the first time, JavaScript was known for a platform to incorporate JavaScript files. In HTML script, JavaScript is used to make a website to look more attractive by added animation, dynamic behavior, and special effects. Since Node.js exist, it uses its own module system to allow JavaScript file to stand on its own and import other JavaScript modules into it. Because of this, it promotes the separation of task performed by each module which it called as library and the library can be imported by using *require()* function in the script. This implementation approach also allows Node application to use third-party libraries that is provided by npm that are available for Node.

Npm is Node.js' package manager that is used for implementing external libraries. It was created as an open source project in 2009 to help developers in JavaScript easily share packaged code modules [32]. In their installations, libraries that need other dependencies can also be managed by npm. After that, the npm has the highest number of packages or libraries more than 700,000 surpassing maven who held the highest number before [33].

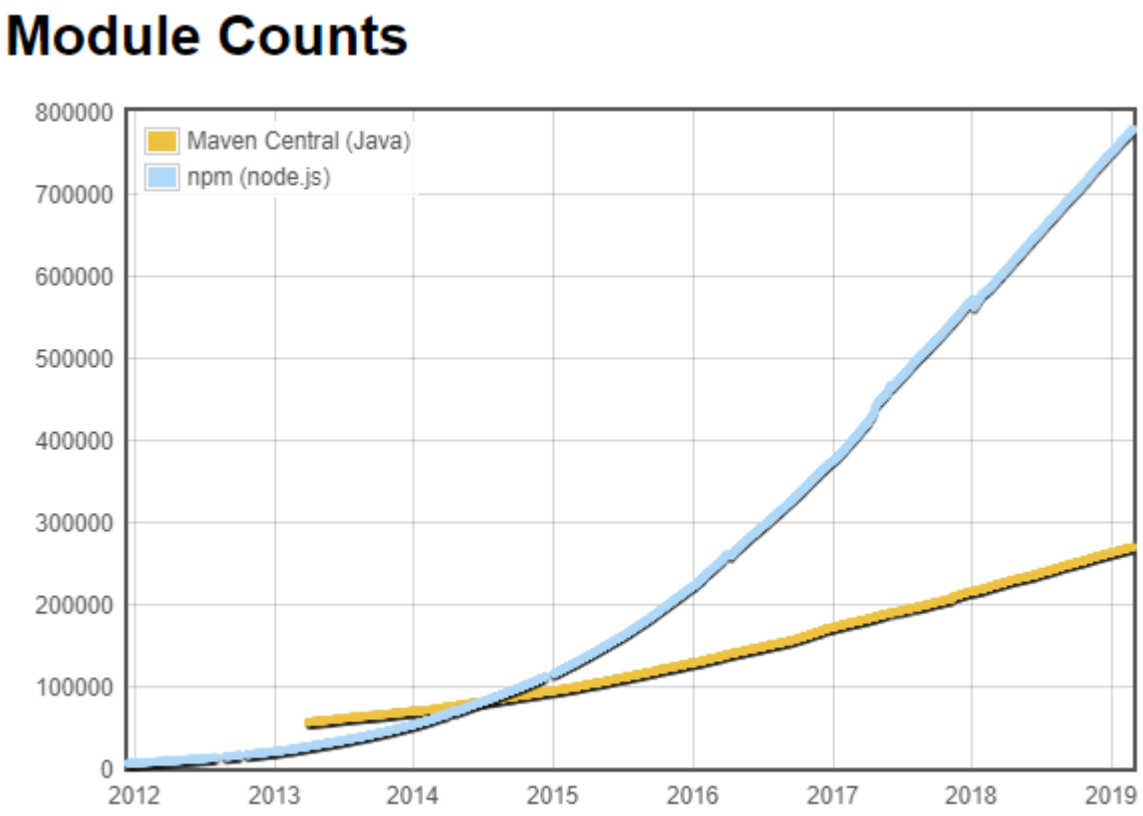

Figure 2.3 Comparison between NPM and Maven [28]

This makes npm a great environment for developers to develop the back end of their application, as the existing modules already solve or handle problems or features of the application.

The typical processing practice nowadays implements the threads of the operating system, using other available threads when a thread has to be blocked due to a certain event, such as file reading. The problem with this approach is that it is difficult to develop applications that process multiple threads, not to mention the likelihood of deadlocks if the application is not carefully designed in advance. Node.js, on the other hand, implements asynchronous processing without multiple threads. Defining whether a specific process has been completed depends on callbacks and can perform other processes that do not require waiting for another task to be processed which can improves the efficiency and output of the system [34]. Node has achieved this approach by implementing an event loop that takes requests from events and handles them in the appropriate processing, thus their called event-driven characteristic. Node can then multitask by running such a query in an asynchronous method and then return

to the event loop to handle other requests while waiting for the result of the query execution. The working flow of Node.js can be visualized in the figure below [35].

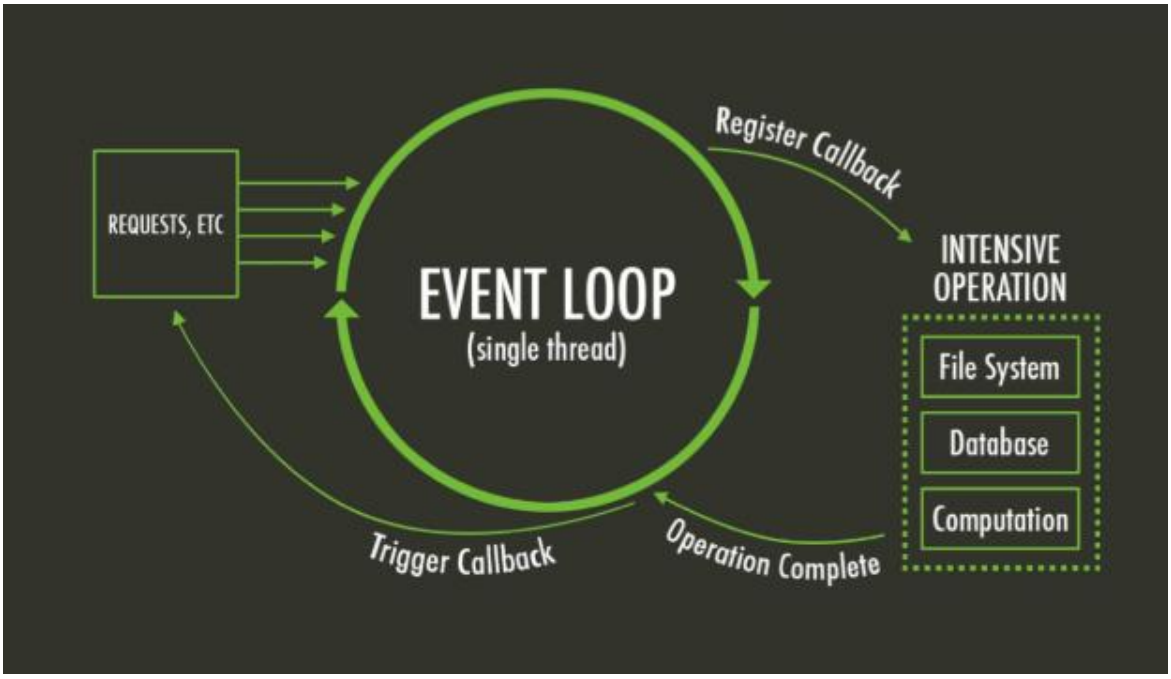

Figure 2.4 Illustration of the How the Event Loop Works [30]

2.2.5 Express.js

ExpressJS is a web application framework that allows you to create websites, web applications and back ends through the pattern in developing with express [36], and also the provided features such as routes definition. Express allows developers to determine what to do if a specific HTTP request, such as the GET or POST method, is received with the corresponding predefined route [37]. Responses can also be sent as a result of retrieval and processing of requests, such as sending processed data or sending custom headers. The Express middleware can also be used to implement additional processing methods so that some tasks carry out similar processing. In addition, middleware from third parties can also be installed from npm, so that developers do not need to define common tasks from scratch.

2.2.6 Database

A database is an organized collection of information that is accessible, managed and updated easily [38]. Data collection includes diagrams, tables, queries, news and reports. The data is usually organized to model aspects of reality in a way that supports information requiring processes. The database contains digitized information that is stored continuously. Many websites and applications rely on databases it is an important part of e-commerce, social media, banking systems, online video games and other software systems that handle persistent information. In addition, database systems offer a wide range of other properties that make them extremely useful and convenient such as data abstraction, efficiency, scalability, reliability, and high - level query languages. The database system refers to a collection of application software that communicates with both the database and the Database Management System (DBMS).

Database Management System (DBMS) is a software designed to help maintain and use large data collection and system requirements [39]. The characteristics of a DBMS are:

- Real-world entity
- Relation-based tables
- Isolation of data and application
- Less redundancy
- Consistency
- Query Language
- ACID Properties
- Multiple Views
- Security

The architecture of a DBMS can be attributed to its design. The design of DBMS can be centralized, decentralized or hierarchical. The architecture of a DBMS can be a single or multi-tier structure. The DBMS is the only entity in a 1-tier architecture where the user sits directly on the DBMS users. If the DBMS architecture is a 2-tier architecture, the application

to access the DBMS is required. A 3-tier architecture separates its levels based on the complexity of the users and the presence of the data in the database [40].

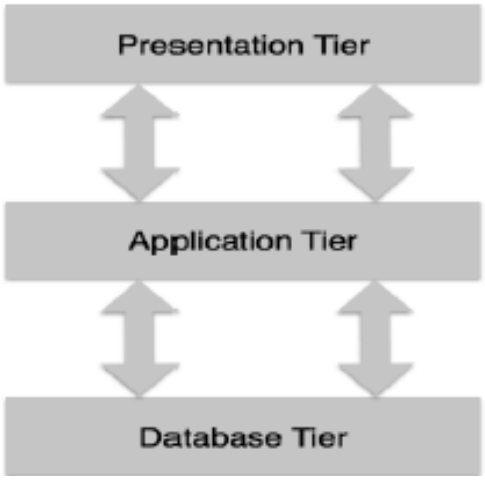

Figure 2.5 3-tier DBMS Architecture [40]

- Database (Data) Tier
  The database and its query processing language are located at this level.

- Application (Middle) Tier
  The application server and the program accessing the database are located at this level. This application tier provides an abstract view of the database for a user.

- User (Presentation) Tier
  End-users operate at this level and don't know about the existence of the database beyond this layer. The application can provide multiple views of the database at this layer.

A relationship database management system (RDBMS) is a program that enables you to create, update and manage a relationship database [41]. Most relational database management systems have access to the database using the SQL language There are many examples of Relational Database Management System (RDBMS), one of them is MySQL. SQL

(Structured Query Language) is a programming language used by developer to communicate with data stored in a Relational Database Management System (RDBMS). There advantages using SQL, include high speed, no coding needed, well defined standards, portability, interactive language, and multiple data view [42]. There are three types of SQL Commands [43]:

1. **Data Definition Language (DDL)**

   These SQL commands are used for creating, modifying, and dropping the structure of database objects. The commands that are commonly used are CREATE, ALTER, and DROP.

2. **Data Manipulation Language (DML)**

   These SQL commands are used for storing, retrieving, modifying, and deleting data. The commands that are commonly used are SELECT, UPDATE, DELETE, and DROP.

3. **Data Control Language (DCL)**

   These SQL commands are used for providing security to database objects. The commands that are commonly used GRANT and REVOKE

2.2.7 JSON

JSON, short for JavaScript Object Notation, is a format for the exchange of data between programming languages or systems [44]. Although it is named "JavaScript" and is based on JavaScript, JSON is not dependent on any programming language, since it is still a data format intended primarily for the exchange of data, so many programming languages or systems must accept it in order to be useful. Putting data in an object-based format such as JSON allows data to be represented more naturally, thus data processing in terms of objects with specific properties.

As its main formatting concept, JSON implemented name-value pairs. The name must be defined in the data type of a string, while the value can be an array, a number, a string, a Boolean, JSON object or null. An example of JSON can be seen in the figure below [45].

```
{
"name":"John",
"age":30,
"cars":[ "Ford", "BMW", "Fiat" ]
}
```

Figure 2.6 Example of JSON [45]

The figure above is a JSON example consists of name-value pairs, for example the variable or name called “age” is in string data type and the value, which is 30, is in number data type.

2.2.8 RESTful API

API, which is short for the Application Programming Interface, is a tool that allows the system to share its services with other systems by providing only the appropriate data and access. Developers can further define access from specific API services to read-only, add, or edit data. This access control allows the implementation of the API as a sharing of processing services without compromising the security of the system itself [46].

While REST stands for Representational State Transfer, Roy Fielding's web architectural style is used to design and manage distributed systems [47]. The overall concept of REST is to use the HTTP protocol as a way for a computer to access resources from another system, which affects the formatting of the URL to help represent the data model. In general, REST aims to increase the performance, reliability, simplicity, visibility, scalability, portability, and modifiability of distributed systems implementing this style.

An API that implements HTTP requests to create, read, update and delete data (CRUD) is a RESTful API [46]. The REST API accepts common HTTP verbs with the endpoint URL and processes if the HTTP verbs and the URL correspond to the definition set in advance. Using the REST API usually results in the data being processed as a response to the request. The HTTP verbs used in REST APIs are examples:

- GET for data retrieval.
- POST for data insertion.
- PUT for updating data.
- DELETE for deleting data.

The actions or processing carried out by the RESTful API must correspond to the defined verbs of that API, such as the data update cannot be carried out using the REST API using the GET method.

2.2.9 Encryption

Encryption is the process of converting originally readable or comprehensible data or information into an encrypted type which make the data become unreadable form [48]. Due to its unreadability, the original meaning of the data could not be obtained in its encrypted form. While decryption can be regarded as the opposite of encryption, which is the process of converting encrypted data into its original form in order to achieve the actual meaning. Encryption is considered to be two-way because encrypted data can be converted by decryption back to its original form. Encryption is carried out on data that needs to be converted back to get back its actual values, such as retrieve password in login form. In application, the password must be hashed because of the security reason but when the application submit login function the system must decrypt it first to obtain the original password to be able the application to login. There are two types of encryption methods [49]:

1. Symmetric encryption, in which the same key (secret key) is used for both encryption and decryption.

2. Asymmetric encryption, which separately assigns specific keys to encryption and decryption. The public key would be shared for encryption, but only the receiving side would own the secret key for decryption. Asymmetric encryption is found to be nearly a thousand times slower than symmetric encryption, as public key encryption itself requires high, intensive computer processing. The use of such a technique on low spec of devices would be affected.

2.2.10 Hashing and Salting

Hashing is a method of entering or generating variable - length data and generating fixed - length data with different values, typically a fixed - length integer value, that eliminates the original meaning of the data [50]. Hashing is used for security purposes, for example to secure sensitive data by failing to store its actual value in the database. While encryption and hashing share the same purpose of securing data by visually altering its value, the difference between the two methods is that hashing is referred to as one-way. The hashing method allows the encoding of a data into a hash value, but the resulting hash value cannot be decoded back, so the original value cannot be obtained from hashed data. The one-way characteristic makes security attacks more difficult, because it is impossible to try to get the value back from the same data this time. Which means, if someone try to crack it, it should either estimate the secret used in the hash or estimate the actual value of the data itself by means brute force approach [51].

The problem with the hashing approach is that the same data would produce the same hash value if hashing is the only method used to secure data. This allows the attackers to estimate by brute force approach the actual value of the password until the same hash value is found. The dictionary attack is an example of this attacking method. A dictionary attack attempts to locate the actual value of hashed data by comparing it by brute force with words from the dictionary or words commonly used as passwords. The attacker will crack the hash if the

word that he has, has the common string in the database. Salting is used to complement the hashing method to overcome the weakness that is found in the implementation of the hashing method only. Salt is a random data, typically of a significant length, used for the hash function as an additional input. It is used to ensure the uniqueness of any generated hash value [52]. By implementing both methods, although data have the exact same value, the resulting hash and salt values from these data would all be unique, thereby overcoming the problem that existed such as dictionary attack can be covered by this method.

2.3 Waterfall Software Development

Waterfall model is a process of software development that emphasizes a logical progression of steps to be taken throughout the life cycle of software development (SDLC), similar to cascading steps down an incremental waterfall [53]. While its popularity recently fell compared to its trendier counterpart, agile remained one of the most widely used methods of development and classic model for industries. One reason it is so often used is because it was the earliest SDLC approach to software development [54]. For some developers, the nature of its progression is also natural and probably more sensible.

Figure 2.7 Waterfall Method [55]

As explained above, the waterfall model has six linear stages of development where one must lead to another without any means of returning to previous stages. These are the following stages [53]:

- Requirements

  The initial phase of discussing and deciding an application's requirements. For the entire process, this will serve as development documentation that defines what the application should do without defining how it should do it.

- Analysis

  The system is thoroughly analyzed to generate appropriate business models and logics to be used in the application.

- Design

  Cover the requirements of technical design such as programming languages, data layers and services. It outlines how technical implementation of the business logic created in the Analysis phase.

- Coding

  Where the actual source code is written, implementing models, business logics, and services specified in previous stage.

- Testing

  The stage where applications bugs can be identified and fixed promptly. Quality Assurance and all other testers are responsible for identifying bugs and subsequently report to the engineer what bugs occur on what specific process. Ensuring that the final application is robust and bug-free is crucial so that business goals can be achieved.

- Operation

  The application is now ready to be deployed and live. This stage not only covered the details of the deployment itself, but also the functionality of the subsequent support and maintenance. It is best that continuous maintenance is committed to keeping the application up-to-date to be safe.

2.4 Data and Flow Modeling Technique

2.4.1 Unified Modeling Language

Unified Modeling Language (UML) is a standard set of diagramming techniques designed to provide a common vocabulary of object-oriented terms to adequately model any system development project from analysis through implementation [56]. Before UML starts, developers have different approaches to modeling an information system's object concept. Failure to standardize object concepts often leads to confusion and misunderstanding when a system diagram analysis is performed. The UML itself is proposed as a standardization

solution with a single approach in modeling system development project. Grady Booch, Ivar Jacobson, James Rumbaugh and several other system analysts designed and proposed it. A member of supersede 2.5 version, which includes fifteen diagramming techniques for modeling a system, is the latest current UML version. All diagrams are divided into two main groups: diagram of structure and diagram of behavior. Structure diagrams represent an approach in a system to represent data and static relationships, while behavioral diagrams represent dynamic relationships between system objects representing business flow. Although each group represents a total of fifteen combined diagramming techniques, the report focuses only on several diagrams to represent the underlying systems accurately. The diagrams are as follows:

- Use Case Diagram

  Using the case diagram can depict and understand the functionality of a high-level system. One of the main reasons for this is that the diagram sends the user a straightforward communication on what the system does. It is usually designed during the collection of functional system requirements and encourages users to add high-level functionality to the system. It then visualizes the interaction between the user and the system itself main functionality. The illustration of the diagram includes several Use Case elements:

  1. Actor

     Depicts the role the user play whilst interacting with the system.

  2. Use Case

     Represent core functionality of a system. Can extend or include to other use case.

  3. Subject Boundary

     Represent the scope of the subject or system.

4. Association/Include/Extend Relationship

   Association relationship depicts the interaction between users to a use case. Both include and extend depict relationship between use cases, with include meaning inclusion functionality of one use-case within another and extend meaning optional behavior.

- Class Diagram

  Class Diagram shows the overall interconnectivity and relationship between the installed business object model called class. A class is a general template used in the problem domain to create a particular business object. In general, two types, concrete class and abstract class can be considered as a class. A concrete class is a class of application domain and is used to create an actual object. However, an abstract class is only useful for the abstraction of a specific common information and is not directly instantiated. They help to abstract generalization identified between concrete classes. Classes contain method and attributes. Attributes are pieces of information that are useful to describe a specific class while methods describe the actions that can be performed by a certain class. Relationship between each classis can be classified into:

1. Generalization

   This is a relationship used to depict each class of children's parent classes or superclasses. The concept of superclasses is to generalize common subclass information found and refractured into one parent class. Semantically described as an expression of a kind.

2. Association

   Represents a general class relationship. Usually labeled with a verb to describe the relationship type semantically. Contains multiplicity that denotes the minimum and maximum instance number created by an object.

3. Aggregation

   Semantically speaking, the relationship of aggregation translates into a logical expression of a part or has-parts. It is used to portray a class belonging to a different class as in a part of wholes or assemblies. Aggregation is a form of association that is special.

4. Composition

   Semantically speaking, the relationship of composition translates into an expression of physical a - part or has - part. Depicts a class that belongs as part of wholes or assemblies to another class. Composition is a form of association that is special.

- Sequence Diagram

  This diagram illustrates objects as they are involved in use cases and the information convey through one use case. It emphasizes interaction based on the activity's time-based ordering, helping to describe complex use case based on real - time specifications. There is usually a set of sequence diagram instances in a system development design phase, each representing the system flow on a particular scenario. They are useful in representing a scenario's flow of control over time. There are actors and objects in a sequence diagram. Actor represents users who interact with the system while objects are merely a class instance that receives and sends messages to other objects during a system flow throughout the system. An intermediate message graph depicting the flow of information and occurring over a lifeline draws the entire interaction between actors and objects, a pointed line that extends downward representing the real - time specification.

- Activity Diagram

  Activity diagram is used to visualize an independent business process of actual technical objects. It creates a high - level model capable of documenting any business workflow. Activity diagram elements include actions and activities that represent behavior and actions that occur during a specific activity, respectively. The flow is

then modelled into any complex system flow with control flow, decision node, and fork node. The diagram starts with the initial node and the final node of activity.

- Entity Relationship Diagram

  Entity Relationship Diagram (or ERD) documents the entities in a relational database, supplemented by the attributes and relationship describing them [57]. In depicting ERD, there are several different methods known. The most well-known and most widely used methods at the moment are: Chen Model. In an ERD, elements include:

  1. Entities

     Entity is a model in a relational database that represents data storage. Most commonly referred to as table in the context of database management. Entity contains a label describing its nature of information as well as attribute tuples of data. It is mainly divided into weak entities dependent on other entities and strong entities independent of any kind of dependency.

  2. Domains

     Each database entity attribute has a domain. It is essentially a constraint that makes it possible to attribute permissible values to an entity. Normally, as an entity attribute standardization, the Database Management System (DBMS) enforces this constraint by default. Often domains restrict primitive values like integer, string, Boolean, and so on.

  3. Relationships

     Normally, ERD relationship is represented as a line connecting one entity to another. Especially in Chen Model, the line is presented as an arrow with detailed expression of multiplicity over it. The arrow direction from an entity to an entity on which it depends. There are several relationships to be represented in ERD such as one-to-one relationships, one-to-many relationships, many-to-one relationships, and some relationships.