

# **YARANG HOSPITAL KEY PERFORMANCE INDICATOR SYSTEM**

**RAIYAN JAPAKEEYA**

**FATONI UNIVERSITY**

**1440/2019**



## CHAPTER ONE

### INTRODUCTION

#### 1.1 PROJECT OVERVIEW

Healthy is a valuable asset and necessary to agility conduct of daily life. Human attempt to seek for treatment their health if it had the deterioration. The place for the best choice to get treatment is hospital. There are many hospitals in Thailand which difference types and classes. Whatever the private hospital, the government hospital, large or small hospital and etc. The most important of all hospitals are the efficiency and quality of maintenance to patients.

**Yarang Hospital** is a government hospital as district level of Yarang district, Pattani, Thailand. Yarang Hospital provides 24-hour service to patients, so perpetually patients every day. Yarang Hospital has been accepted and trusted from community. To ensure that Yarang Hospital is efficiency and quality by “**Key Performance Indicator**” or “**KPI**”. KPI is a measurable value that demonstrates how effectively a company is achieving key business objectives. Formerly, Yarang Hospital are required to assess KPI in every year with undergo trouble. All values have been collected by admin from several resources such many branches and teams around hospital. Then, admin types all values in Microsoft Excel to calculate and make a graph. After that, they would have a meeting to discuss and analyze the graph. Causing most of the responsibilities to become only admin with many of numeric values from many branches and teams. If so, errors will occur very easily. It would be great if let a member

of each branches and teams can type numeric values by themselves as a spread responsibility. Therefore, **KPI System** will be replace the old process that can access to every computer through network in hospital.

**Yarang Hospital KPI System** is web application online available for Yarang Hospital or may work on another hospital as well. To make an assess the KPI to ensure the efficiency and quality of maintenance to patients easily and accurately because users just type numeric values then the system will calculate and shows the graph. In addition, the system has comment box under the graph to provide users make an analyze the graph and consider to develop hospital to be better soon.

## 1.2 PROBLEM STATEMENT

From an assess the KPI of Yarang Hospital efficiency and quality as the old way that already exist, be able to identify the problem as follows:

1. Frequent typo data because of only one responsible man with many numeric values.
2. Take much time to collect the data as numeric values from several resources around hospital.
3. The time difference to conduct a meeting for analyzing and considering the graph of KPI.

### **1.3 PROJECT OBJECTIVE**

The main purpose of this project is to develop Yarang Hospital KPI System can assess the KPI of Yarang Hospital easier and save many troubles to achieve the following objectives:

1. To develop Yarang Hospital KPI System.
2. To design and develop a centralized database system of Yarang Hospital KPI.

### **1.4 SCOPE OF STUDY**

Yarang Hospital KPI System will develop for admin and staff to assess the KPI of Yarang Hospital efficiency and quality. The system being process by using ID and password to limit access only approved individuals. Admin can manage all staff and all indicator with graph and comment in the hospital. Staff can only manage especially indicator with graph and comment of own branch. The project also will develop a centralized database system to storage all KPI data and it will develop into Thai language.

## **1.5 SIGNIFICANT OF STUDY**

Yarang Hospital KPI System developed to organize an assessing the KPI of Yarang Hospital. Signification of this system hopefully to gain the benefit both for developer and user. Developer will get more knowledge and more practice about develop system and centralized database system. User will get the new system as upgrade the whole processing with easily and accurately.

## **1.6 CHAPTER ORGANIZE**

### **Chapter One: Introduction**

This chapter introduce about overview of the project, problem statement, objectives to develop the Yarang Hospital KPI System, scope of study and software and hardware requirement.

### **Chapter Two: literature review**

This chapter discusses about the theories from many sources, articles, book or case study that related to this project in order to be benefits and guideline to develop the Yarang Hospital KPI System.

### **Chapter Three: Methodology**

This chapter will discuss about the methodology used in the Yarang Hospital KPI System. It is the most important part because it will describe in each phases of project process in detail. For this project the method that used is Waterfall.

## **Chapter Four: Implementation**

This chapter shoe the coding and testing the system. If user found any error in this application, developer will fix bugs and solve the problem.

## **Chapter Five: Conclusion**

The conclusion of this study is described how the system can be benefits for the users and recommendation for the future work.

## CHAPTER TWO

### LITERATURE REVIEW

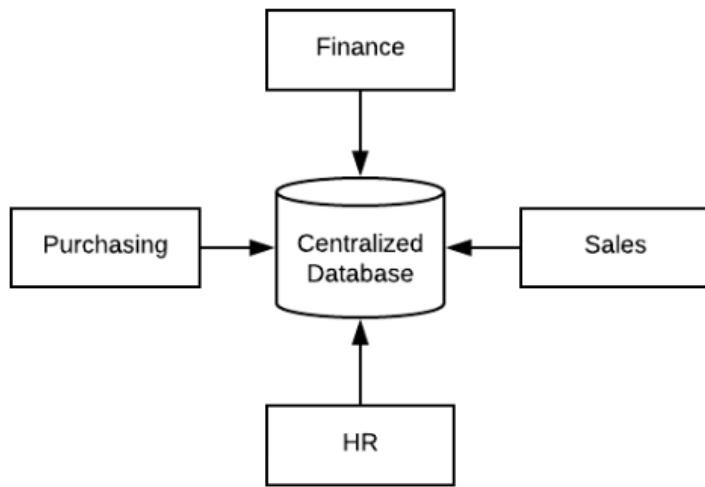
#### 2.1 DEFINITION

##### 2.1.1 Key Performance Indicator (KPI)

A Key Performance Indicator (KPI) is a metric that's used to quantify progress towards important business objectives. High-level KPIs measure the overall performance of a company, while low-level KPIs focus on measuring the impact of tasks and projects led by individual teams like marketing, sales, customer service, or IT. KPIs cascade down from high-level goals. For example, your company sets an overall goal to grow revenue by 20% this year. Individual department leaders then define the specific components of that goal that their teams directly control and establish KPIs that will quantify their teams' contributions to the goal. (Kononenko, 2019)

##### 2.1.2 Centralized Database Management System

A centralized database is stored at a single location such as a mainframe computer. It is maintained and modified from that location only and usually accessed using an internet connection such as a LAN or WAN. The centralized database is used by organizations such as colleges, companies, banks etc. (Onsman, 2018)



*Figure 2.1: Centralized Database*

(Retrieved December 17, 2019, from <https://www.tutorialspoint.com/Centralized-Database-Management-System>)

As can be seen from the above diagram, all the information for the organization is stored in a single database. This database is known as the centralized database.

### **Advantages**

Some advantages of Centralized Database Management System are:

1. The data integrity is maximized as the whole database is stored at a single physical location. This means that it is easier to coordinate the data and it is as accurate and consistent as possible.

2. The data redundancy is minimal in the centralized database. All the data is stored together and not scattered across different locations. So, it is easier to make sure there is no redundant data available.
3. Since all the data is in one place, there can be stronger security measures around it. So, the centralized database is much more secure.
4. Data is easily portable because it is stored at the same place.
5. The centralized database is cheaper than other types of databases as it requires less power and maintenance.
6. All the information in the centralized database can be easily accessed from the same location and at the same time. (Onsman, 2018)

### **Disadvantages**

Some disadvantages of Centralized Database Management System are:

1. Since all the data is at one location, it takes more time to search and access it. If the network is slow, this process takes even more time.
2. There is a lot of data access traffic for the centralized database. This may create a bottleneck situation.
3. Since all the data is at the same location, if multiple users try to access it simultaneously it creates a problem. This may reduce the efficiency of the system.
4. If there are no database recovery measures in place and a system failure occurs, then all the data in the database will be destroyed. (Onsman, 2018)

## **2.2 RELATED WORK**

### **2.2.1 Key Performance Indicators to Benchmark Hospital Information Systems – A Delphi Study**

In the context of IT governance, IT benchmarking is gaining importance in the healthcare area. The found indicators reflect the view of health care IT professionals and researchers. Research is needed to further validate and operationalize key performance indicators, to provide an IT benchmarking framework, and to provide open repositories for a comparison of the HIS benchmarks of different hospitals. (Hübner-Bloder & AmmenwerthKevin, 2018)

### **2.2.2 A KPI framework for process-based benchmarking of hospital information systems.**

Benchmarking is a major topic for monitoring, directing and elucidating the performance of hospital information systems (HIS). Current approaches neglect the outcome of the processes that are supported by the HIS and their contribution to the hospital's strategic goals. We suggest to benchmark HIS based on clinical documentation processes and their outcome. A framework consisting of a general process model and outcome criteria for clinical documentation processes is introduced. (Jahn & Winter, 2011)

### **2.2.3 Virtopsy – The concept of a centralized database in forensic medicine for analysis and comparison of radiological and autopsy data**

Recent developments in clinical radiology have resulted in additional developments in the field of forensic radiology. After implementation of cross-sectional radiology and optical surface documentation in forensic medicine, difficulties in the validation and analysis of the acquired data was experienced. To address this problem and for the comparison of autopsy and radiological data a centralized database with internet technology for forensic cases was created. (Aghayev & et al, 2007)

## CHAPTER THREE

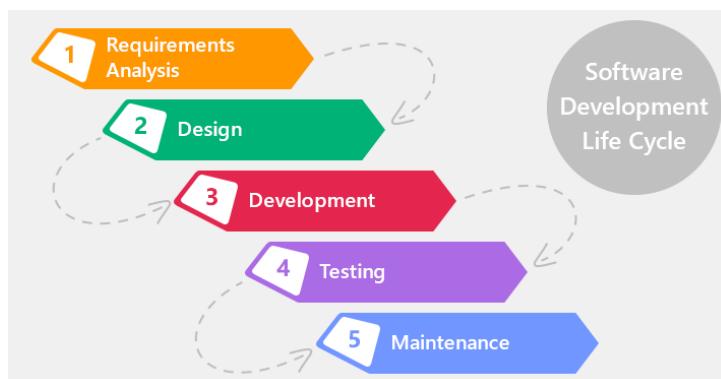
### METHODOLOGY

This chapter focuses on methodology, analysis and design of Yarang Hospital KPI system that provide of idea and logic of how the system was built and how about system workflow. It also describes each processes of system development life cycle in detail. For this project the method that used is **Waterfall model**.

#### **3.0 INTRODUCTION**

##### **3.0.1 Definition of Waterfall model**

Waterfall model is a linear (sequential) development life cycle model that describes development as a chain of successive steps. No phase can be started before or simultaneously with the previous or current one. Let's consider the Waterfall model's main phases as they go. (Sarycheva, 2019)



*Figure 3.1: Waterfall Model's Main Phases*

*(Retrieved December 23, 2019, from <https://xbsoftware.com/blog/software-development-life-cycle-waterfall-model/>)*

### 3.1 REQUIREMENTS AND ANALYSIS

The first step of requirements and analysis phase is developer make an interview with the leader of computer center at Yarang hospital who required this system for gathering information and requirements to identify the problem occurring during use old system and any information related with the system until the need of new system. Developers have to analyst system architecture overview, system structure chart and any process analysis and design.

The screenshot shows a Microsoft Excel spreadsheet titled "ผู้ช่วยเบิกบานเดือน2561" (Assistant Budgeting Month 2561). The data is organized into several sections:

- Section 1:** รายการค่าใช้จ่ายพยาบาลต่อจำนวนเรียบเทียบตัวต่อหัวพยาบาลโรงพยาบาล จำนวน 47 ตัว ปีงบประมาณ 2561 (ผู้ช่วยเบิกบานเดือนที่ 15 ของเดือนไป)
- Section 2:** รายการค่าใช้จ่ายของผู้ป่วย AMI (จำนวน 1 ราย)
- Section 3:** รายการค่าใช้จ่ายของผู้ป่วย AMI ตามเกณฑ์ จำนวน 1 ราย
- Section 4:** รายการค่าใช้จ่ายของผู้ป่วย AMI ตามเกณฑ์ จำนวน 1 ราย
- Section 5:** รายการค่าใช้จ่ายของผู้ป่วย AMI ตามเกณฑ์ จำนวน 1 ราย
- Section 6:** รายการค่าใช้จ่ายของผู้ป่วย AMI ตามเกณฑ์ จำนวน 1 ราย
- Section 7:** รายการค่าใช้จ่ายของผู้ป่วย AMI ตามเกณฑ์ จำนวน 1 ราย
- Section 8:** รายการค่าใช้จ่ายของผู้ป่วย AMI ตามเกณฑ์ จำนวน 1 ราย
- Section 9:** รายการค่าใช้จ่ายของผู้ป่วย AF ให้รับยา Warfarin มีระดับความเสี่ยงทางการรักษา จำนวน 3 ราย
- Section 10:** รายการค่าใช้จ่ายของผู้ป่วย AF ให้รับยา Warfarin มีระดับความเสี่ยงทางการรักษา จำนวน 1 ราย
- Section 11:** รายการค่าใช้จ่ายของผู้ป่วย Atrial Fibrillation ที่มีระดับความเสี่ยงทางการรักษาในระดับสูง จำนวน 1 ราย
- Section 12:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 28 ราย
- Section 13:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 1 ราย
- Section 14:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 1 ราย
- Section 15:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 1 ราย
- Section 16:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 1 ราย
- Section 17:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 1 ราย
- Section 18:** รายการค่าใช้จ่ายของผู้ป่วย Stroke จำนวน 1 ราย

Figure 3.2: The old processing of KPI in Yarang Hospital

According to figure 2.2 all values have been collected by admin from several resources such many branches and teams around hospital. Then, admin types all values in Microsoft Excel to calculate and make a graph.

There are requirements from requirer about providing the distribution of levels and teams as the tables below.

ระดับโรงพยาบาล		
ลำดับ	เรื่อง	รหัส
1	โรงพยาบาลยรัง	YRH
2	ศูนย์คุณภาพ	QSC
3	ทีมนำทางคลินิก	PCT
4	ทีมพัฒนาทรัพยากรบุคคลการ	HRD
5	ทีมบริหารความเสี่ยง	RMS
6	ทีมสิ่งแวดล้อมและความปลอดภัย	ENV
7	ทีมเครื่องมือทางการแพทย์	EQU
8	คณะกรรมการสิทธิผู้ป่วยและพฤติกรรมบริการ	ETH
9	ทีมระบบยา	PTC
10	ทีมเฝ้าระวังและป้องกันการติดเชื้อ	ICS
11	ทีมสารสนเทศและระบบเวชระบะบิน	IMS
12	ทีมสุขศึกษาและพัฒนาพฤติกรรมสุขภาพ	HEA
13	องค์กรพยาบาล	NSO
14	องค์กรแพทย์	MSO

Table 3.1: List of the name and code of hospital level teams in Yarang Hospital

ระดับหน่วยงาน		
ลำดับ	เรื่อง	รหัส
1	งานการเจ้าหน้าที่	ADM
2	งานการเงินและบัญชี	FIN
3	งานพัสดุ	CAT
4	ฝ่ายทันตกรรม	DEN
5	ฝ่ายเภสัชกรรม	PHA

ระดับหน่วยงาน		
ลำดับ	เรื่อง	รหัส
6	งานชันสูตร	LAB
7	งานรังสี	RAD
8	งานผู้ป่วยใน	IPD
9	งานผู้ป่วยนอก	OPD
10	งานโรคเรื้อรัง	NCD
11	งานอุบัติเหตุ-ฉุกเฉิน	EMR
12	งานสูติกรรม	OBS
13	งานกายภาพบำบัด	PTD
14	งานเวชระเบียน	MDR
15	ศูนย์คอมพิวเตอร์และเทคโนโลยี	CCT
16	งานประกันสุขภาพ	UCD
17	งานซักฟอก-จ่ายกล่อง	SUP
18	งานแพทย์แผนไทย	TTM
19	กลุ่มงานบริการปฐมภูมิและองค์รวม	PCH
20	งานสุขภาพจิต	PSY
21	งานโสตและประชาสัมพันธ์	PRC
22	งานโภชนาการ	NUT
23	งานเอดส์และยาเสพติด	ATD
24	งานyanพาหนะ	AUT
25	งานรักษาความปลอดภัย	SCD
26	งานซ่อมบำรุง	MAT
27	งานคุณภาพต่อเนื่อง	COC
28	งานวัณโรค	TBD
29	งานสวน	GDN
30	คลินิกฝากรรภ.	ANC

Table 3.2: List of the name and code of department level team in Yarang Hospital

In this analysis phase also shared the understanding of all the relevant information that have been collected to the analysis step to figure out the key users, the functionalities, and others as the papers: system architecture, structure chart, list of requirements, use case diagram, use case specification, sequence diagram and activity diagram.

### 3.2.1 System architecture overview

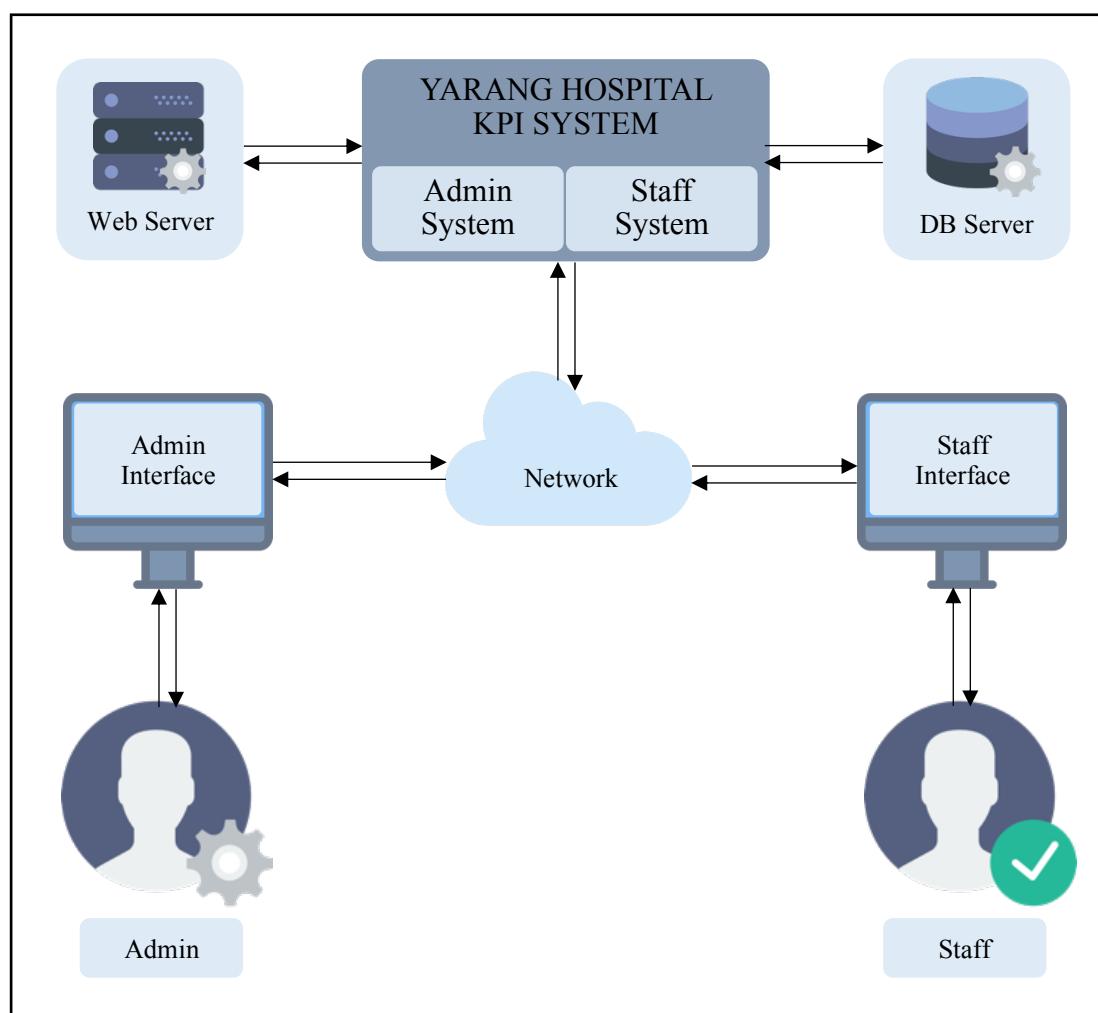


Figure 3.3: System architecture overview

### 3.2.2 System structure chart

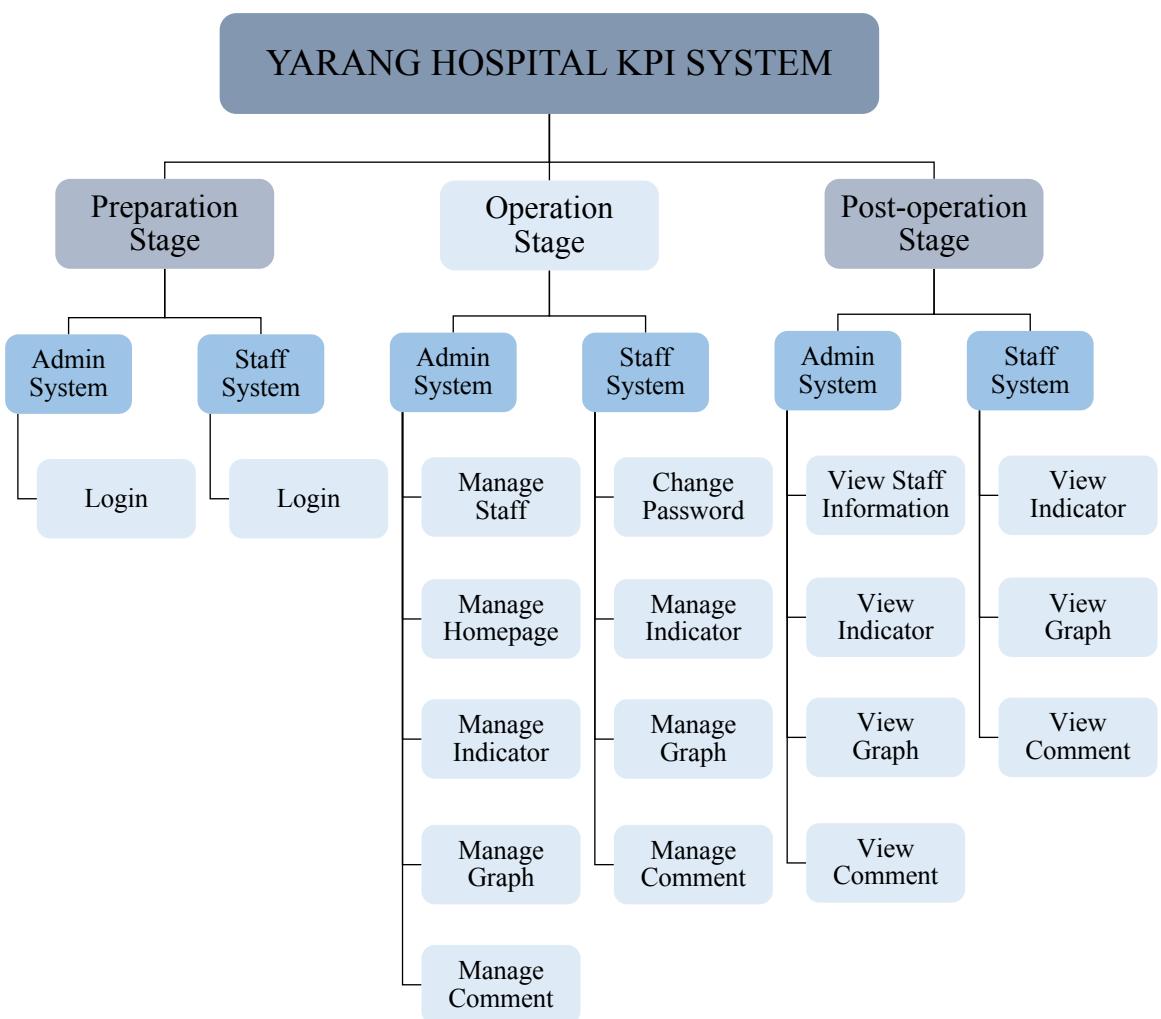


Figure 3.4: System structure chart

### 3.2.3 Process analysis and design

#### List of Requirements

F - Functional Requirement (The system must have)

N - Nonfunctional Requirement (The system nice to have)

No	Requirement ID	Requirement Description	Prior
	<b>REQ_01</b>	<b>Login</b>	
1	REQ_01.01	Admin can login	F
2	REQ_01.02	Staff can login	F
	<b>REQ_02</b>	<b>Manage Staff</b>	
3	REQ_02.01	Admin can add new staff	F
4	REQ_02.02	Admin can update staff	F
5	REQ_02.03	Admin can delete staff	F
	<b>REQ_03</b>	<b>Manage Homepage</b>	
6	REQ_03.01	Admin can select Indicator to show on Home page	F
	<b>REQ_04</b>	<b>Manage Indicator</b>	
7	REQ_04.01	Admin can add new Indicator	F
8	REQ_04.02	Admin can update all Indicator	F
9	REQ_04.03	Admin can delete all Indicator	F
10	REQ_04.04	Staff can add new Indicator	F
11	REQ_04.05	Staff can update own Indicator	F
12	REQ_04.06	Staff can delete own Indicator	F
	<b>REQ_05</b>	<b>Manage Graph</b>	
13	REQ_05.01	Admin can add new Graph	F
14	REQ_05.02	Admin can update all Graph	F
15	REQ_05.03	Admin can delete all Graph	F
16	REQ_05.04	Staff can add new Graph	F

No	Requirement ID	Requirement Description	Prior
17	REQ_05.05	Staff can update own Graph	F
18	REQ_05.06	Staff can delete own Graph	F
<b>REQ_06</b>		<b>Manage Comment</b>	
19	REQ_06.01	Admin can add new Comment	F
20	REQ_06.02	Admin can update all Comment	F
21	REQ_06.03	Admin can delete all Comment	F
22	REQ_06.04	Staff can add new Comment	F
23	REQ_06.05	Staff can update own Comment	F
24	REQ_06.06	Staff can delete own Comment	F
<b>REQ_07</b>		<b>View Staff Information</b>	
25	REQ_07.01	Admin can view staff information	F
<b>REQ_08</b>		<b>View Indicator</b>	
26	REQ_08.01	Admin can view all Indicator	F
27	REQ_08.02	Staff can view own Indicator	F
<b>REQ_09</b>		<b>View Graph</b>	
28	REQ_09.01	Admin can view all Graph	F
29	REQ_09.02	Staff can view own Graph	F
<b>REQ_10</b>		<b>View Comment</b>	
30	REQ_10.01	Admin can view all Comment	F
31	REQ_10.02	Staff can view own Comment	F
<b>REQ_11</b>		<b>Change Password</b>	
32	REQ_11.01	Staff can change password	F

Table 3.3: List of Requirements

### 3.2.4 Use case diagram

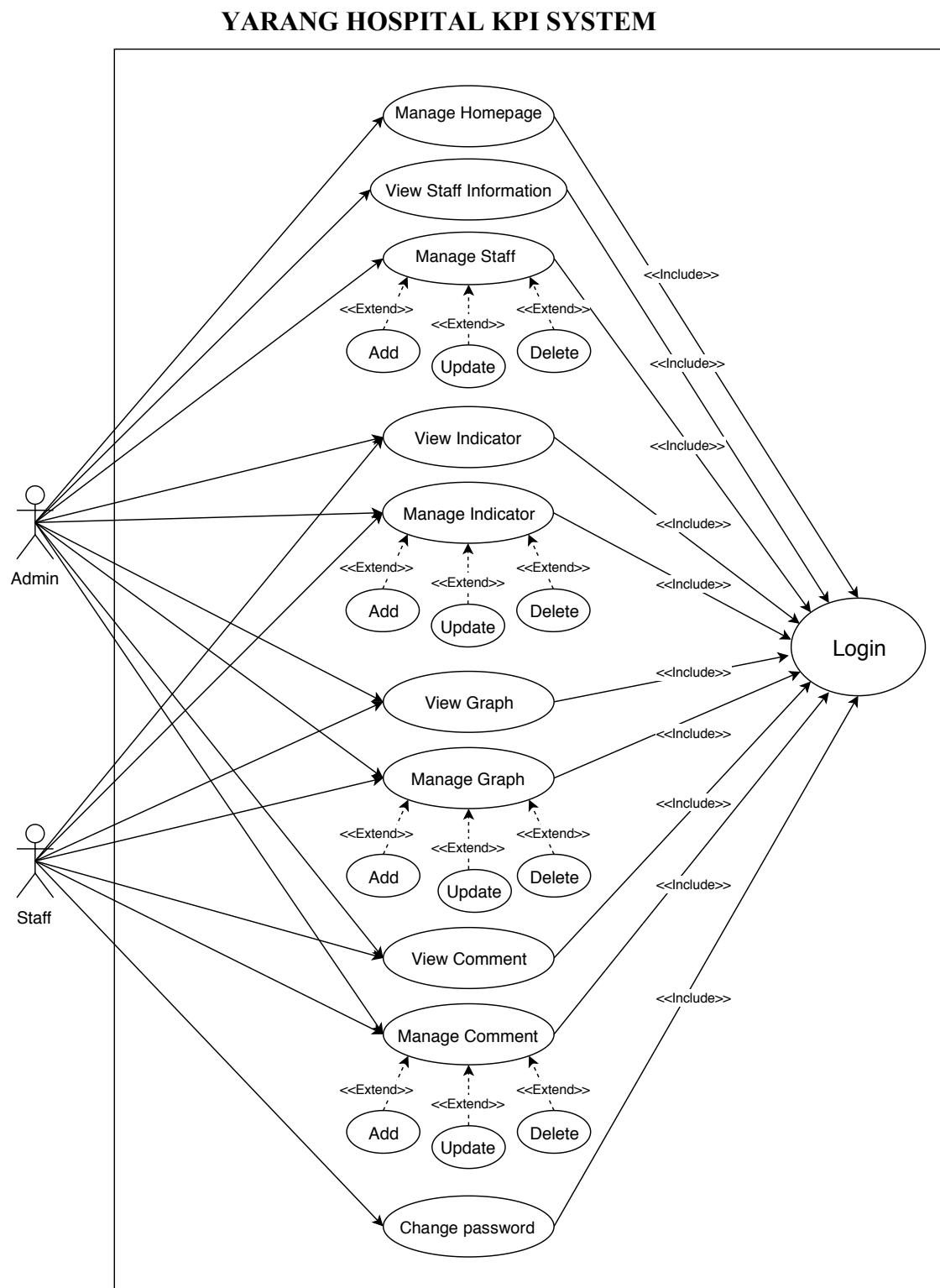


Figure 3.5: Use case diagram of Yarang hospital KPI system

### 3.2.5 Use case specification

#### REQ\_01: Login

<b>Use case name:</b> Login	<b>ID:</b> REQ_01	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin and Staff – want to login.		
<b>Brief Description:</b> This use case will be used to allow users (Admin and Staff) enter to the system.		
<b>Trigger:</b> Users need to enter to the system.		
<b>Type:</b> External		
<b>Relationships:</b> <ul style="list-style-type: none"> <li>Association: Admin and Staff</li> <li>Include: None</li> <li>Extend: Validate username or password</li> <li>Generalization: None</li> </ul>		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Users key-in username and password on the user login page</li> <li>2. Users press the login button</li> <li>3. System will validate username and password [E-1: Invalid username or password]</li> <li>4. System will display the admin home page for admin and the staff home page for staff.</li> </ol>		
<b>Alternative/Exception Flows:</b> <ul style="list-style-type: none"> <li>E-1: Invalid username or password           <ol style="list-style-type: none"> <li>1. The system will display error message.</li> <li>2. Users have to re-enter username or password.</li> </ol> </li> </ul>		

Table 3.4: Use case specification of Login

## **REQ\_02: Manage Staff**

<b>Use case name:</b> Manage Staff	<b>ID:</b> REQ_02	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin		
<b>Stakeholders and Interests:</b> Admin – want to manage staff		
<b>Brief Description:</b> This use case will be used to allow admin to manage staff such as add new staff, update staff's information and delete staff account from the system		
<b>Trigger:</b> Admin need to manage staff on the system		
<b>Type:</b> Internal		
<b>Relationships:</b> <ul style="list-style-type: none"> <li>Association: Admin</li> <li>Include: Login [REQ_01]</li> <li>Extend: Add [REQ_02.01], Update [REQ_02.02], Delete [REQ_02.03]</li> <li>Generalization: None</li> </ul>		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Admin select manage staff on the admin page</li> <li>2. The system displays manage staff form to allow admin to add new staff [A-1: Update], [A-2: Delete]</li> <li>3. Admin key-in all staff information on the add form</li> <li>4. Admin press the confirm button</li> <li>5. The system will validate the entering data [E-1: validate the entering data]</li> <li>6. System end when the system display acknowledge message that staff has been added</li> </ol>		
<b>Alternative/Exception Flows:</b> <ul style="list-style-type: none"> <li>A-1: Update           <ol style="list-style-type: none"> <li>1. The system will allow admin to update staff's information</li> </ol> </li> <li>A-2: Delete           <ol style="list-style-type: none"> <li>1. The system will allow admin to delete staff's account</li> </ol> </li> <li>E- 1: Invalid the entering data           <ol style="list-style-type: none"> <li>1. The system will display error message</li> <li>2. Admin has to re-enter the data</li> </ol> </li> </ul>		

*Table 3.5: Use case specification of Manage Staff*

### **REQ\_03: Manage Homepage**

<b>Use case name:</b> Manage Homepage	<b>ID:</b> REQ_03	<b>Important Level:</b> Low
<b>Primary Actor:</b> Admin		
<b>Stakeholders and Interests:</b> Admin – want to manage homepage		
<b>Brief Description:</b> This use case will be used to allow admin to select the high light indicators to show on homepage		
<b>Trigger:</b> Admin need to manage homepage on the system		
<b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin Include: Login [REQ_01] Extend: None Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Admin select manage homepage on the admin page</li> <li>2. The system displays manage homepage form to allow admin to select the high light indicators to show on homepage</li> <li>3. Admin click check box to select the high light indicators</li> <li>4. System end when the system display acknowledge massage that the high light indicators has been showed on homepage</li> </ol>		
<b>Alternative/Exception Flows:</b> None		

*Table 3.6: Use case specification of Manage Homepage*

### **REQ\_04: Manage Indicator**

<b>Use case name:</b> Manage Indicator	<b>ID:</b> REQ_04	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin – want to manage all indicator Staff – want to manage own indicator		
<b>Brief Description:</b> This use case will be used to allow admin to manage all indicator and staff to manage only indicator of own branch such as add new indicator, update indicator's information and delete indicator from the system		
<b>Trigger:</b> Admin and Staff need to manage indicator on the system <b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin and Staff Include: Login [REQ_01] Extend: Add [REQ_04.01, REQ_04.04], Update [REQ_04.02, REQ_04.05], Delete [REQ_04.03, REQ_04.06] Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Users select manage indicator</li> <li>2. The system displays manage indicator form to allow users to add new indicator [A-1: Update], [A-2: Delete]</li> <li>3. Users key-in all indicator information on the add form</li> <li>4. Users press the confirm button</li> <li>5. The system will validate the entering data [E-1: validate the entering data]</li> <li>6. System end when the system display acknowledge message that indicator has been added</li> </ol>		
<b>Alternative/Exception Flows:</b> A-1: Update <ol style="list-style-type: none"> <li>1. The system will allow users to update indicator's information</li> </ol> A-2: Delete <ol style="list-style-type: none"> <li>1. The system will allow users to delete indicator's account</li> </ol> E- 1: Invalid the entering data <ol style="list-style-type: none"> <li>1. The system will display error message</li> <li>2. Users have to re-enter the data</li> </ol>		

*Table 3.7: Use case specification of Manage Indicator*

### **REQ\_05: Manage Graph**

<b>Use case name:</b> Manage Graph	<b>ID:</b> REQ_05	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin – want to manage all graph Staff – want to manage own graph		
<b>Brief Description:</b> This use case will be used to allow admin to manage all graph and staff to manage only graph of own branch such as add new graph, update graph's information and delete graph from the system		
<b>Trigger:</b> Admin and Staff need to manage graph on the system <b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin and Staff Include: Login [REQ_01] Extend: Add [REQ_05.01, REQ_05.04], Update [REQ_05.02, REQ_05.05], Delete [REQ_05.03, REQ_05.06] Generalization: None		
<b>Normal Flow of Events:</b> 1. Users select manage graph 2. The system displays manage graph form to allow users to add new graph [A-1: Update], [A-2: Delete] 3. Users key-in all graph information on the add form 4. Users press the confirm button 5. The system will validate the entering data [E-1: validate the entering data] 6. System end when the system display acknowledge message that graph has been added		
<b>Alternative/Exception Flows:</b> A-1: Update 1. The system will allow users to update graph's information A-2: Delete 1. The system will allow users to delete graph's account E- 1: Invalid the entering data 1. The system will display error message 2. Users have to re-enter the data		

*Table 3.8: Use case specification of Manage Graph*

### **REQ\_06: Manage Comment**

<b>Use case name:</b> Manage Comment	<b>ID:</b> REQ_06	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin – want to manage all comment Staff – want to manage own comment		
<b>Brief Description:</b> This use case will be used to allow admin to manage all comment and staff to manage only comment of own branch such as add new comment, update comment's information and delete comment from the system		
<b>Trigger:</b> Admin and Staff need to manage comment on the system <b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin and Staff Include: Login [REQ_01] Extend: Add [REQ_06.01, REQ_06.04], Update [REQ_06.02, REQ_06.05], Delete [REQ_06.03, REQ_06.06] Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Users select manage comment</li> <li>2. The system displays manage comment form to allow users to add new comment [A-1: Update], [A-2: Delete]</li> <li>3. Users key-in all comment information on the add form</li> <li>4. Users press the confirm button</li> <li>5. The system will validate the entering data [E-1: validate the entering data]</li> <li>6. System end when the system display acknowledge message that comment has been added</li> </ol>		
<b>Alternative/Exception Flows:</b> A-1: Update <ol style="list-style-type: none"> <li>1. The system will allow users to update comment's information</li> </ol> A-2: Delete <ol style="list-style-type: none"> <li>1. The system will allow users to delete comment's account</li> </ol> E- 1: Invalid the entering data <ol style="list-style-type: none"> <li>1. The system will display error message</li> <li>2. Users have to re-enter the data</li> </ol>		

*Table 3.9: Use case specification of Manage Comment*

### **REQ\_07: View Staff Information**

<b>Use case name:</b> View Staff Information	<b>ID:</b> REQ_07	<b>Important Level:</b> Low
<b>Primary Actor:</b> Admin		
<b>Stakeholders and Interests:</b> Admin – want to view staff information		
<b>Brief Description:</b> This use case will be used to allow admin to view staff information		
<b>Trigger:</b> Admin need to view staff information on the system		
<b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin Include: Login [REQ_01] Extend: None Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"><li>1. Admin select manage staff on the admin page</li><li>2. The system displays manage staff to allow admin to view staff information</li><li>3. Admin press view button to see staff information</li><li>4. The system will display the result.</li></ol>		
<b>Alternative/Exception Flows:</b> None		

*Table 3.10: Use case specification of View Staff Information*

### **REQ\_08: View Indicator**

<b>Use case name:</b> View Indicator	<b>ID:</b> REQ_08	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin – want to view all indicator Staff – want to view own indicator		
<b>Brief Description:</b> This use case will be used to allow admin to view all indicator and staff to view own indicator of own branch		
<b>Trigger:</b> Admin and Staff need to view indicator information on the system <b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin and Staff Include: Login [REQ_01] Extend: None Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Users select manage indicator</li> <li>2. The system displays manage indicator to allow users to view indicator information</li> <li>3. Users press view button to see indicator information</li> <li>4. The system will display the result.</li> </ol>		
<b>Alternative/Exception Flows:</b> None		

*Table 3.11: Use case specification of View Indicator*

### **REQ\_09: View Graph**

<b>Use case name:</b> View Graph	<b>ID:</b> REQ_09	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin – want to view all graph Staff – want to view own graph		
<b>Brief Description:</b> This use case will be used to allow admin to view all graph and staff to view own graph of own branch		
<b>Trigger:</b> Admin and Staff need to view graph information on the system <b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin and Staff Include: Login [REQ_01] Extend: None Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Users select manage graph</li> <li>2. The system displays manage graph to allow users to view graph information</li> <li>3. Users press view button to see graph information</li> <li>4. The system will display the result.</li> </ol>		
<b>Alternative/Exception Flows:</b> None		

*Table 3.12: Use case specification of View Graph*

### **REQ\_10: View Comment**

<b>Use case name:</b> View Comment	<b>ID:</b> REQ_10	<b>Important Level:</b> High
<b>Primary Actor:</b> Admin and Staff		
<b>Stakeholders and Interests:</b> Admin – want to view all comment Staff – want to view own comment		
<b>Brief Description:</b> This use case will be used to allow admin to view all comment and staff to view own comment of own branch		
<b>Trigger:</b> Admin and Staff need to view comment information on the system <b>Type:</b> Internal		
<b>Relationships:</b> Association: Admin and Staff Include: Login [REQ_01] Extend: None Generalization: None		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Users select manage comment</li> <li>2. The system displays manage comment to allow users to view comment information</li> <li>3. Users press view button to see comment information</li> <li>4. The system will display the result.</li> </ol>		
<b>Alternative/Exception Flows:</b> None		

*Table 3.13: Use case specification of View Comment*

### **REQ\_11: Change Password**

<b>Use case name:</b> Change Password	<b>ID:</b> REQ_11	<b>Important Level:</b> Low
<b>Primary Actor:</b> Staff		
<b>Stakeholders and Interests:</b> Staff – want to change password		
<b>Brief Description:</b> This use case will be used to allow staff change password		
<b>Trigger:</b> Staff want to change password		
<b>Type:</b> Internal		
<b>Relationships:</b> <ul style="list-style-type: none"> <li>Association: Staff</li> <li>Include: Login [REQ_01]</li> <li>Extend: Validate old password</li> <li>Generalization: None</li> </ul>		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. Staff select change password on the staff page</li> <li>2. Staff key-in old password and new password</li> <li>3. Users press the confirm button</li> <li>4. System will validate password [E-1: Invalid password]</li> <li>5. System end when the system display acknowledge message that password has been changed</li> </ol>		
<b>Alternative/Exception Flows:</b> <ul style="list-style-type: none"> <li>E-1: Invalid password           <ol style="list-style-type: none"> <li>1. The system will display error message.</li> <li>2. Staff has to re-enter password.</li> </ol> </li> </ul>		

*Table 3.14: Use case specification of Change Password*

### 3.2.6 Sequence diagram

#### 1. Login

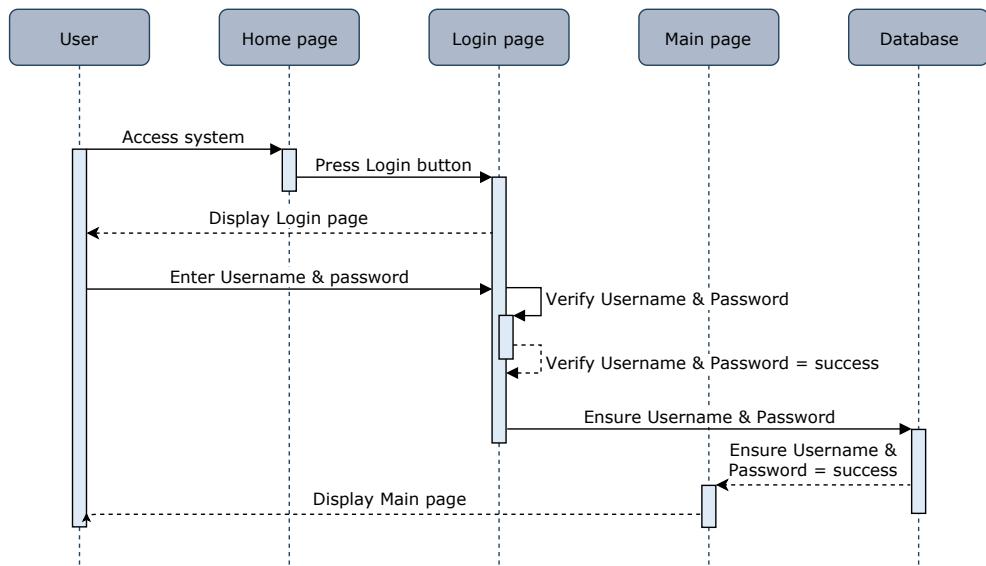


Figure 3.6: Sequence diagram of Login

#### 2. Manage Staff

##### 2.1 Add, Update and Delete Staff

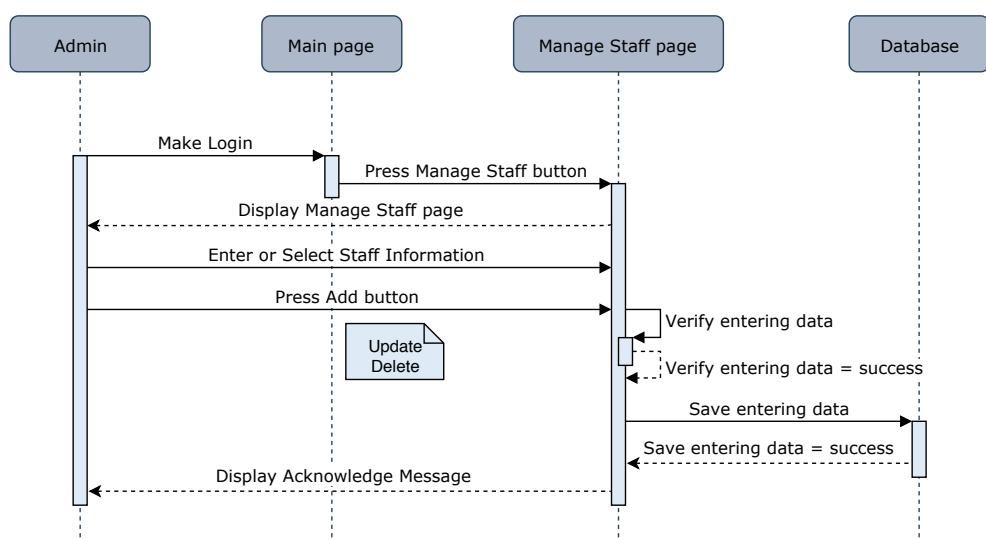


Figure 3.7: Sequence diagram of Manage Staff

### 3. Manage Homepage

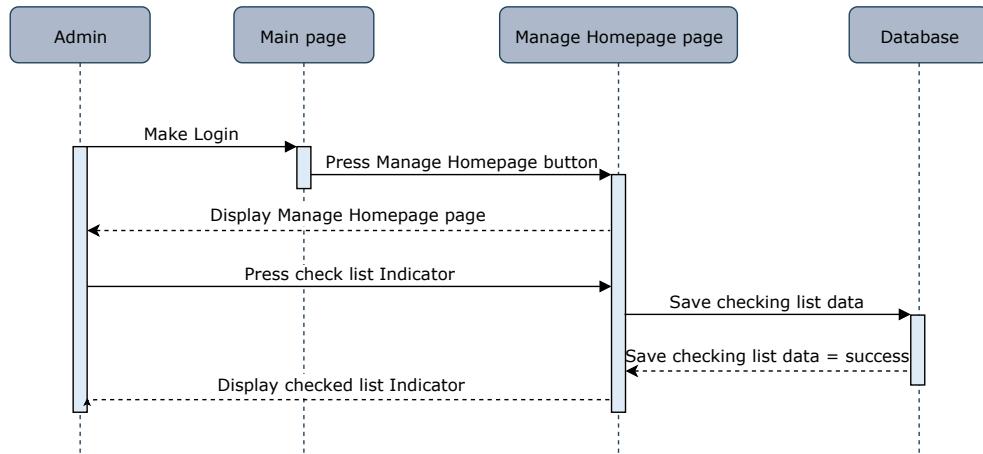


Figure 3.8: Sequence diagram of Manage Homepage

### 4. Manage Indicator

#### 4.1 Add, Update and Delete Indicator

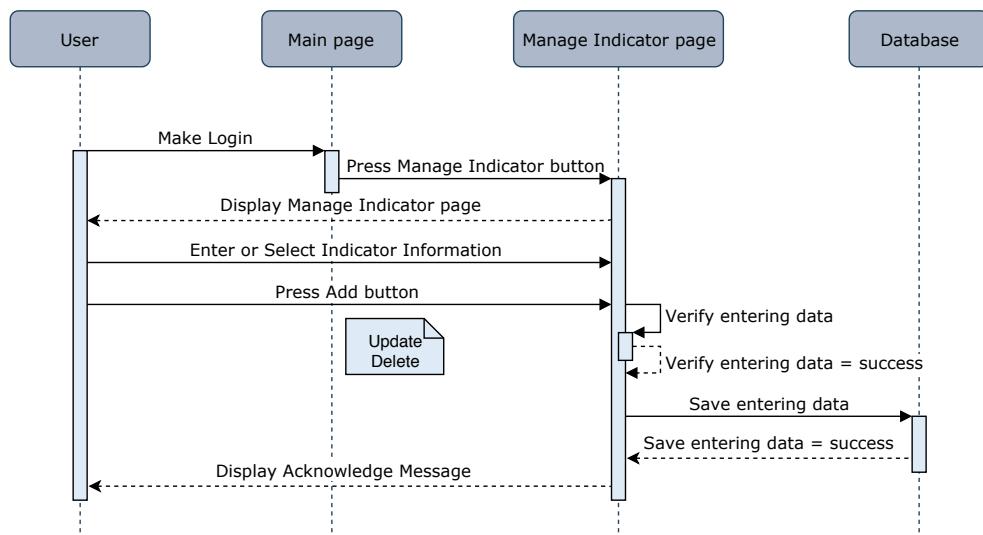


Figure 3.9: Sequence diagram of Manage Indicator

## 5. Manage Graph

### 5.1 Add Graph

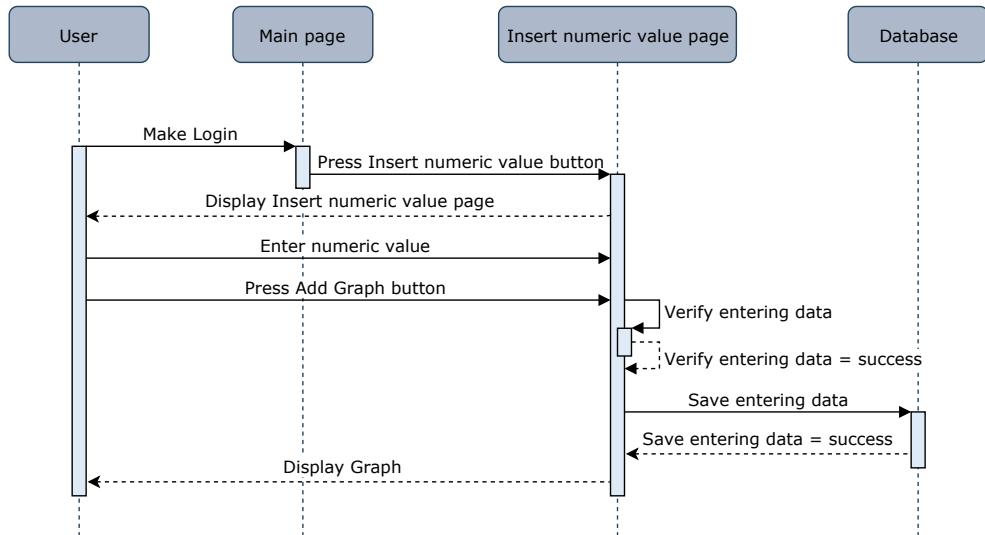


Figure 3.10: Sequence diagram of Add Graph

### 5.2 Update and delete Graph

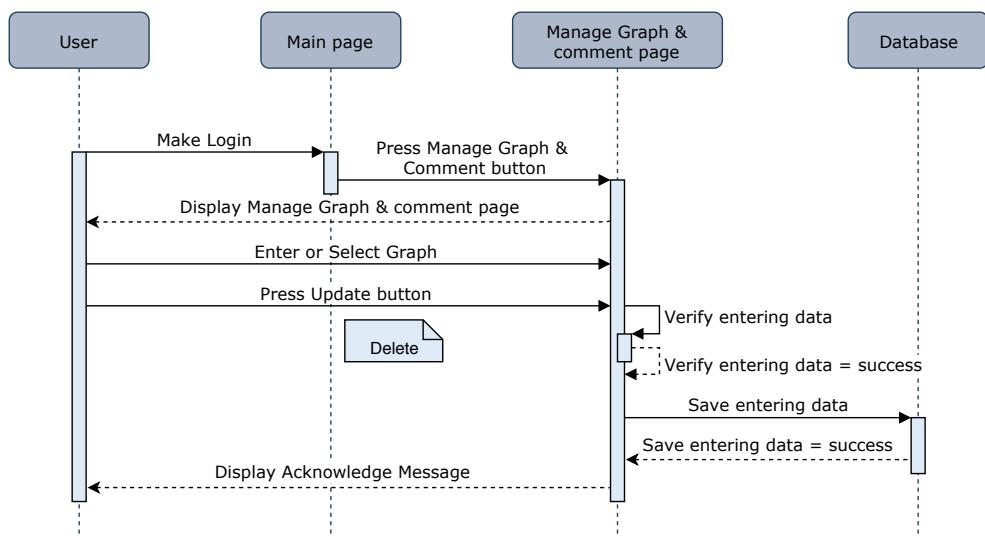


Figure 3.11: Sequence diagram of Update and delete Graph

## 6. Manage Comment

### 6.1 Add Comment

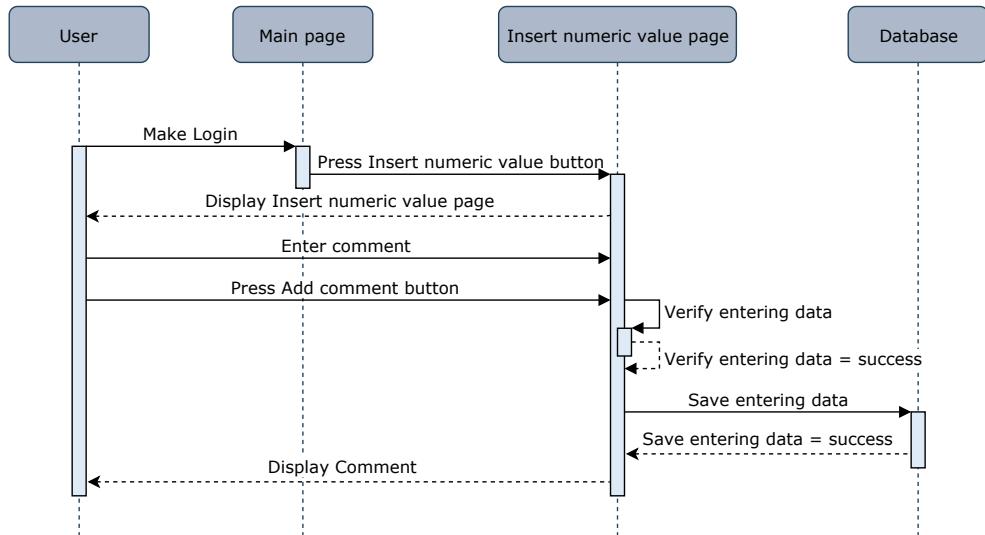


Figure 3.12: Sequence diagram of Add Comment

### 6.2 Update delete Comment

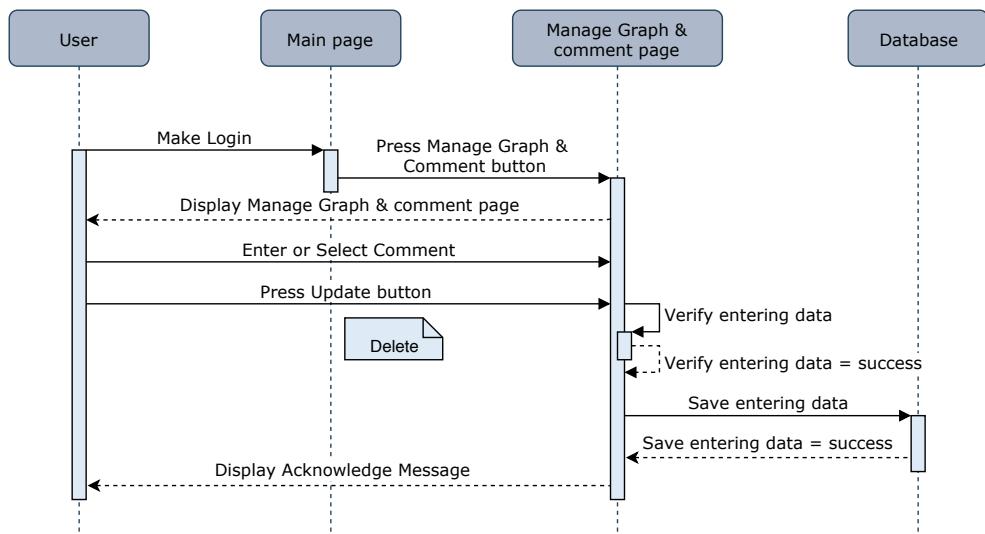
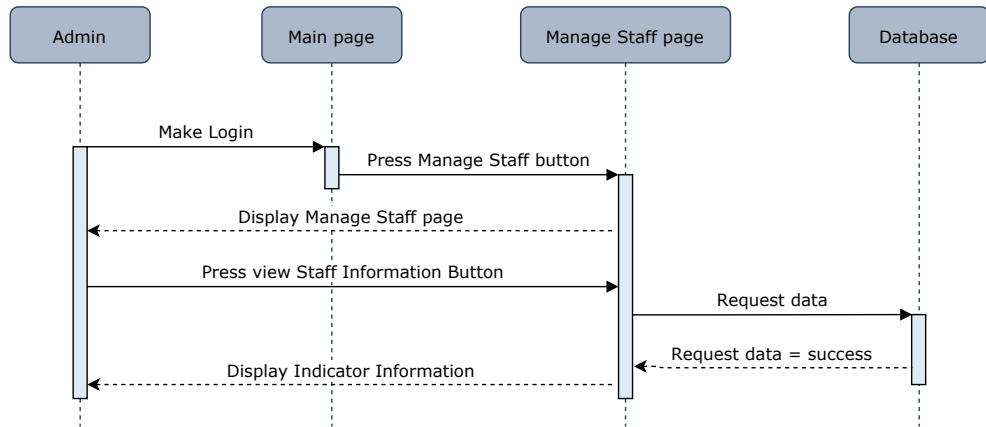


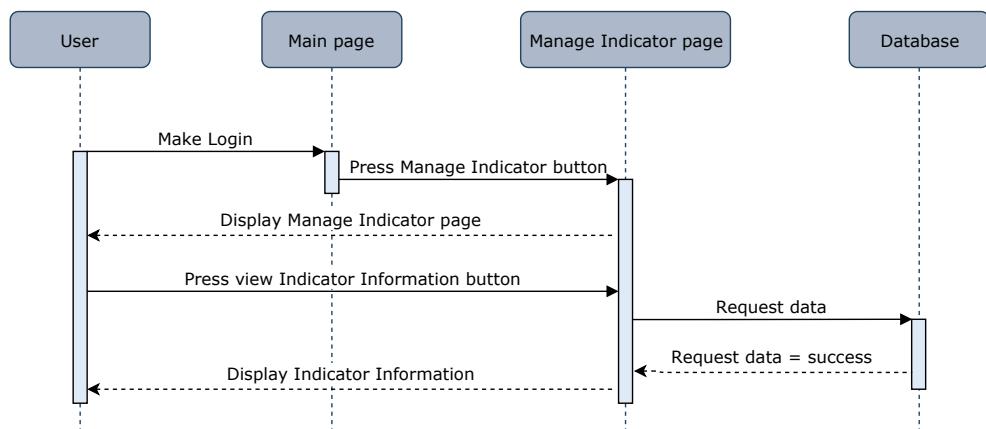
Figure 3.13: Sequence diagram of Update and delete Comment

## 7. View Staff Information



*Figure 3.14: Sequence diagram of View Staff Information*

## 8. View Indicator



*Figure 3.15: Sequence diagram of View Indicator*

## 9. View Graph

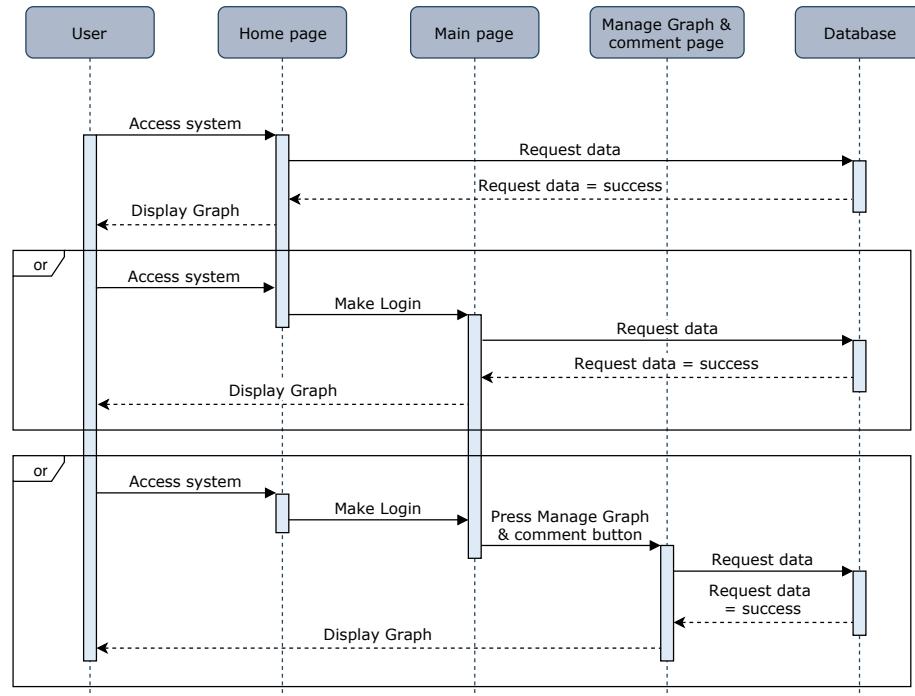


Figure 3.16: Sequence diagram of View Graph

## 10. View Comment

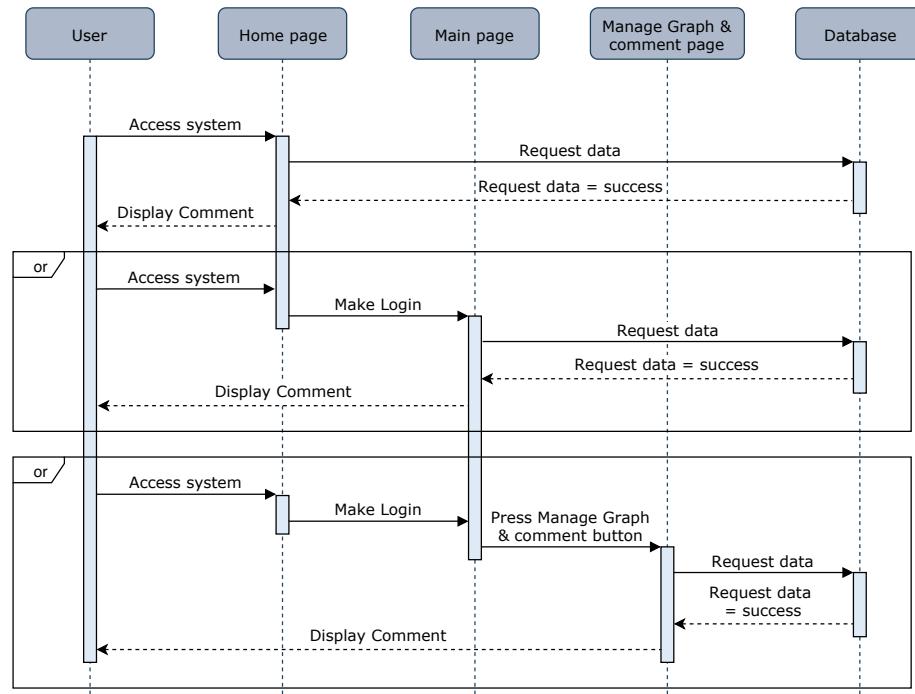


Figure 3.17: Sequence diagram of View Comment

## 11. Change Password

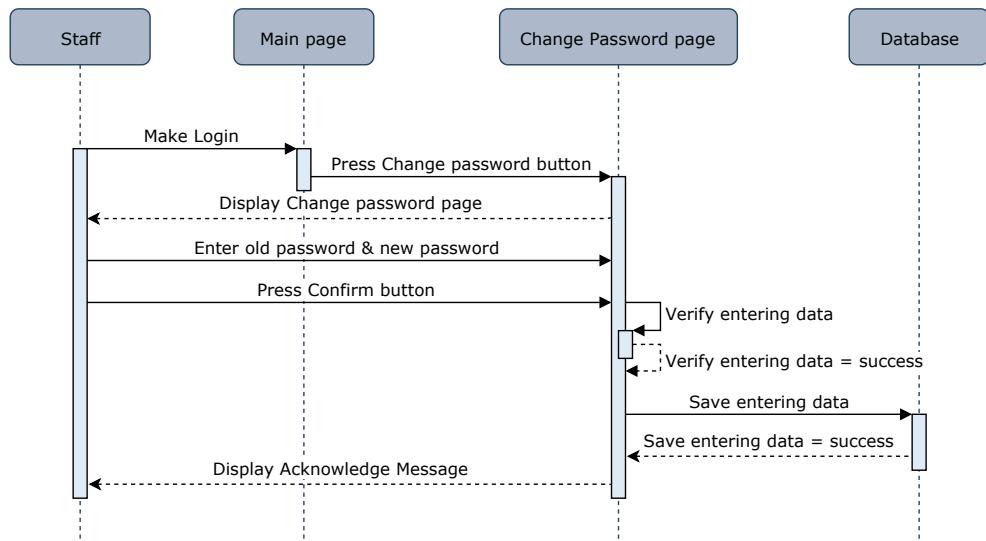


Figure 3.18: Sequence diagram of Change Password

### 3.2.7 Activity diagram

#### 1. Login

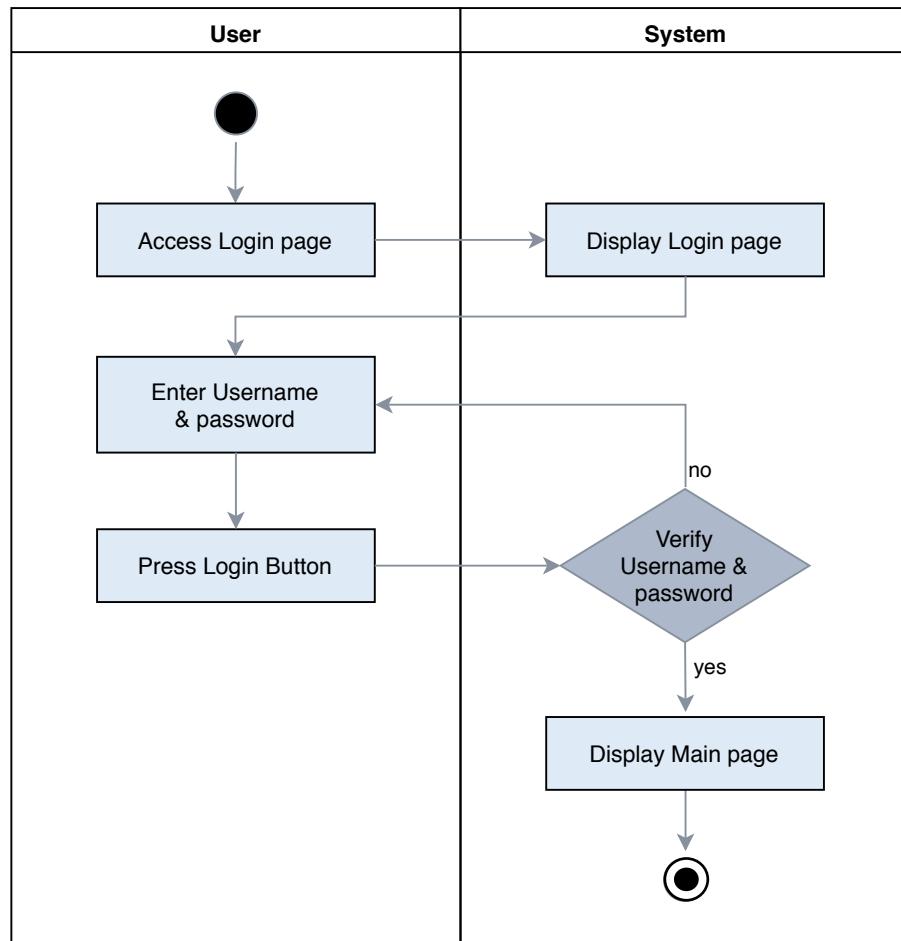


Figure 3.19: Activity diagram of Login

## 2. Manage Staff

### 2.1 Add, Update and Delete Staff

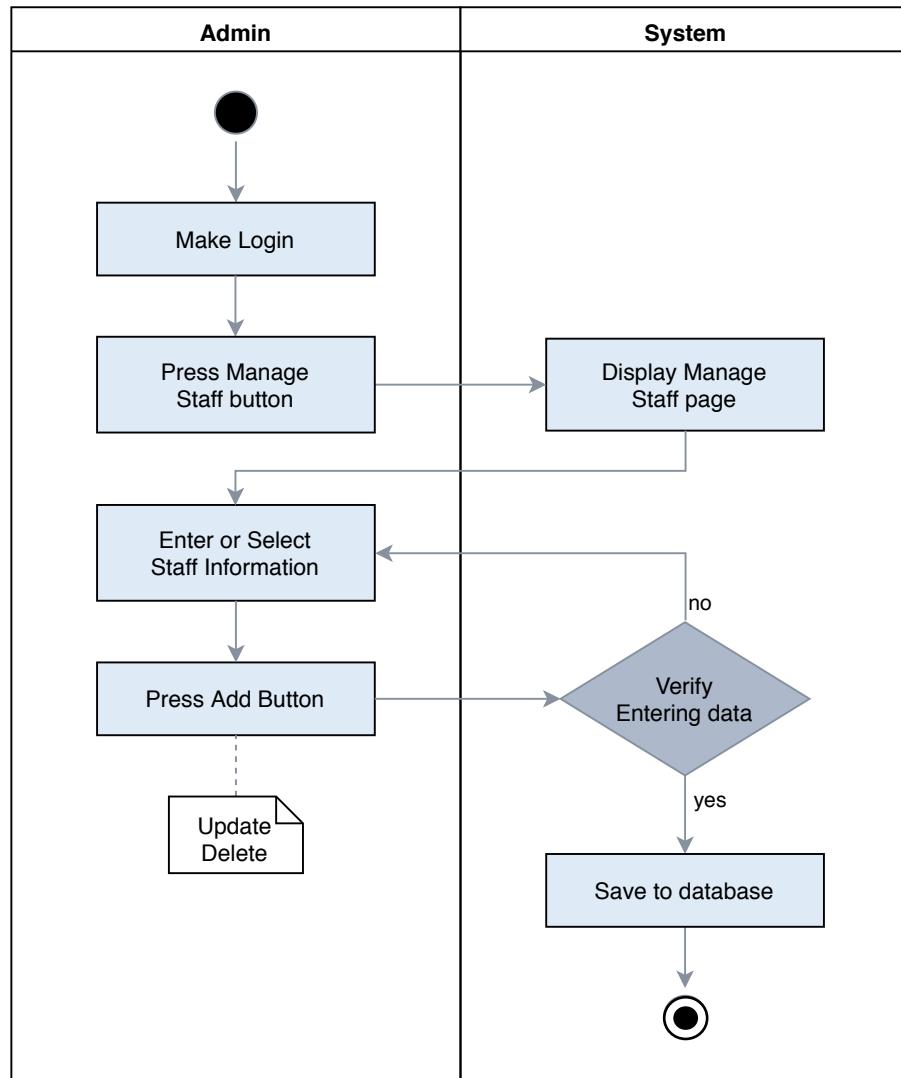


Figure 3.20: Activity diagram of Manage Staff

### 3. Manage Homepage

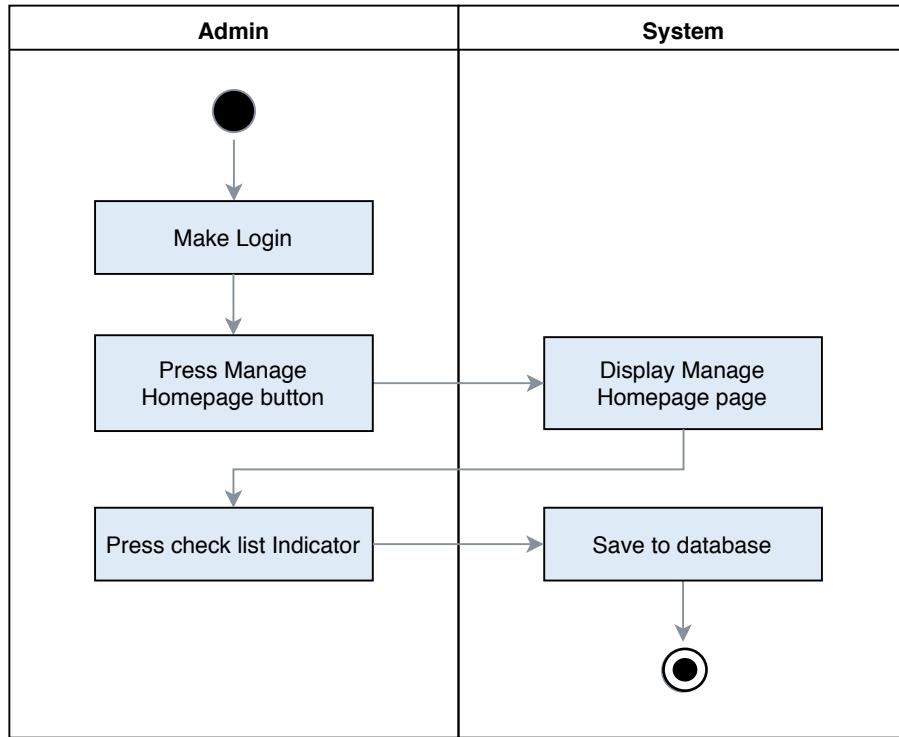


Figure 3.21: Activity diagram of Manage Homepage

## 4. Manage Indicator

### 4.1 Add, Update and Delete Indicator

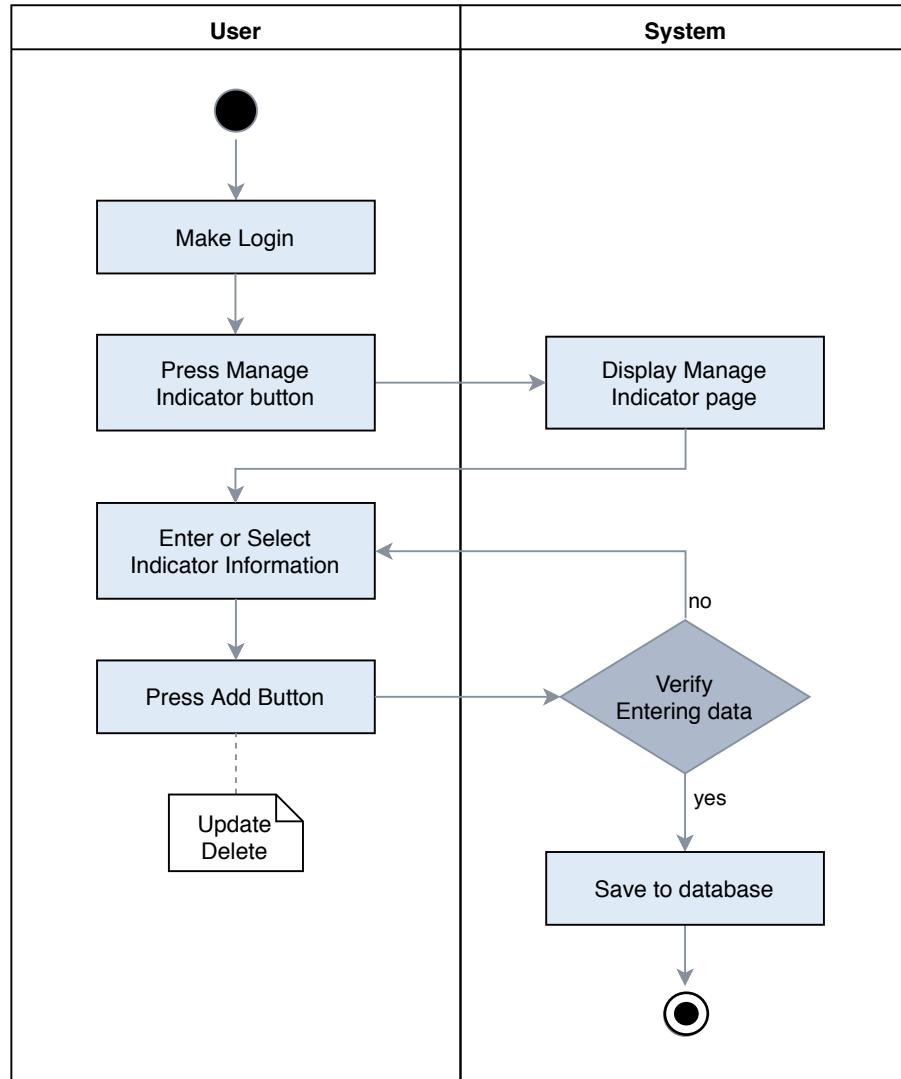


Figure 3.22: Activity diagram of Manage Indicator

## 5. Manage Graph

### 5.1 Add Graph

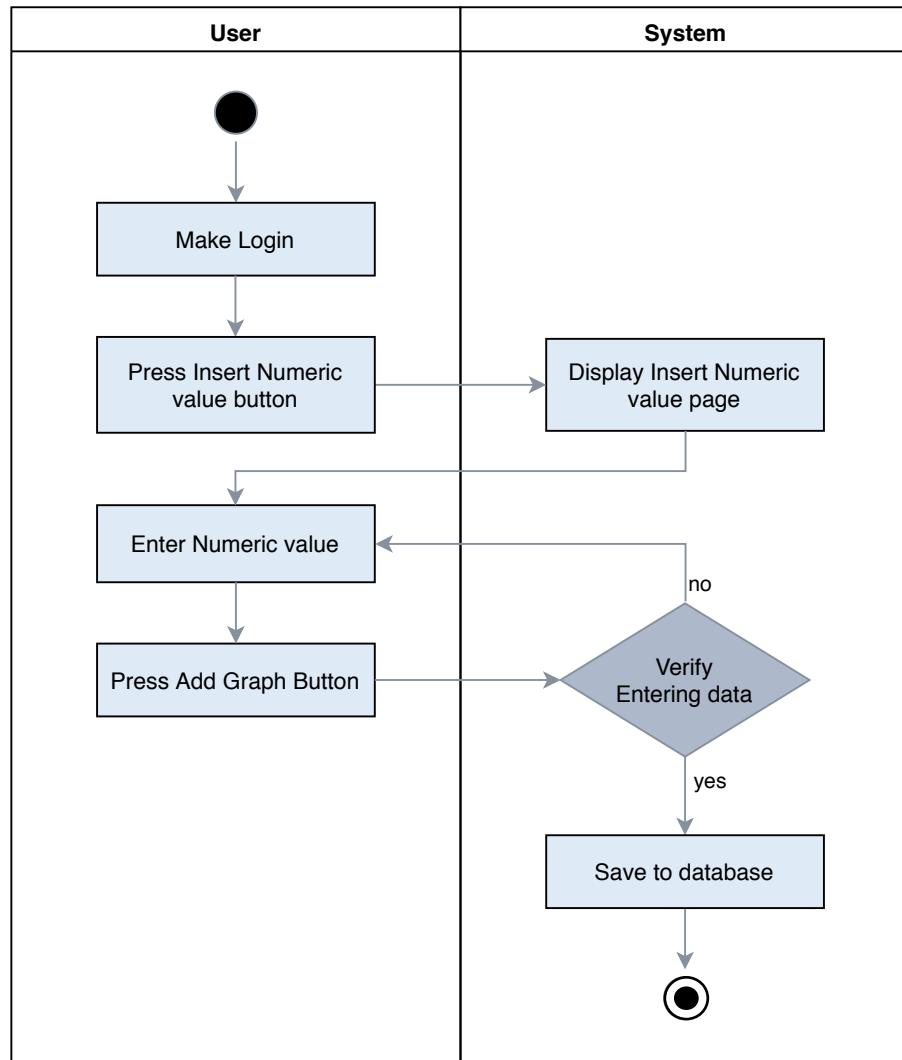


Figure 3.23: Activity diagram of Add Graph

## 5.2 Update and delete Graph

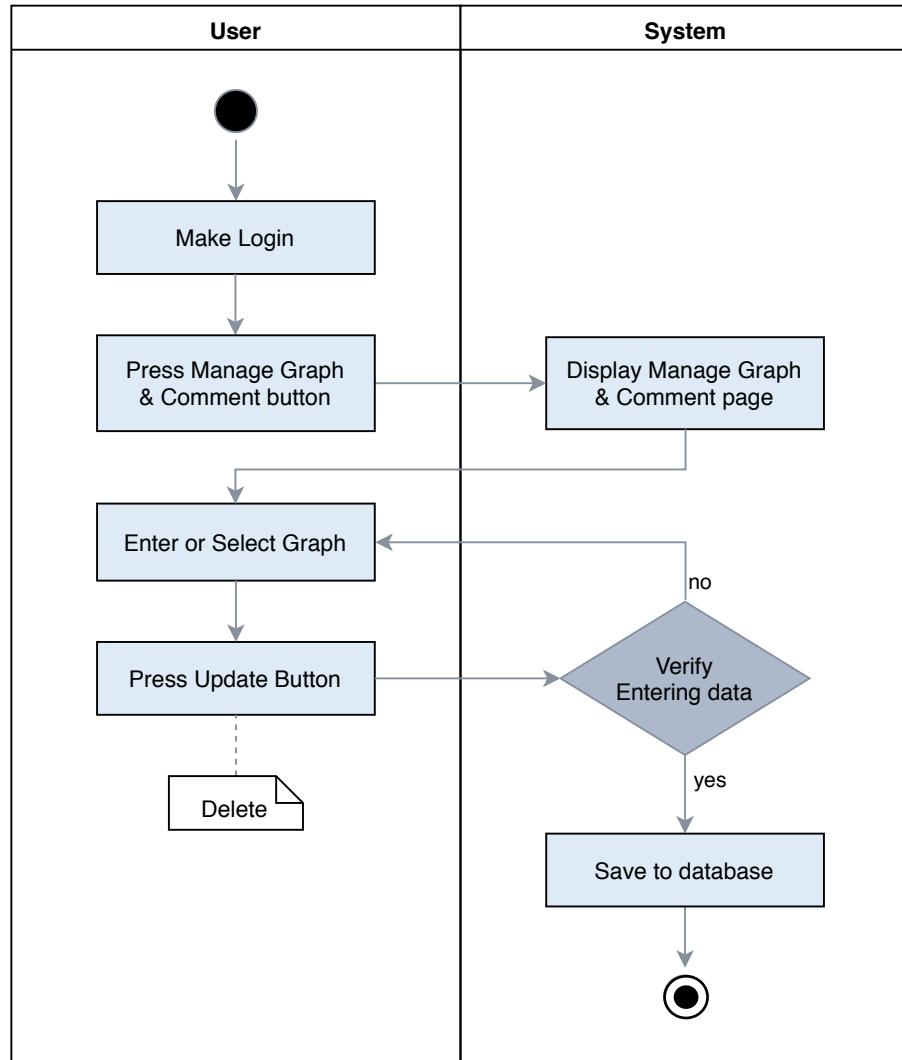


Figure 3.24: Activity diagram of Update and delete Graph

## 6. Manage Comment

### 6.1 Add Comment

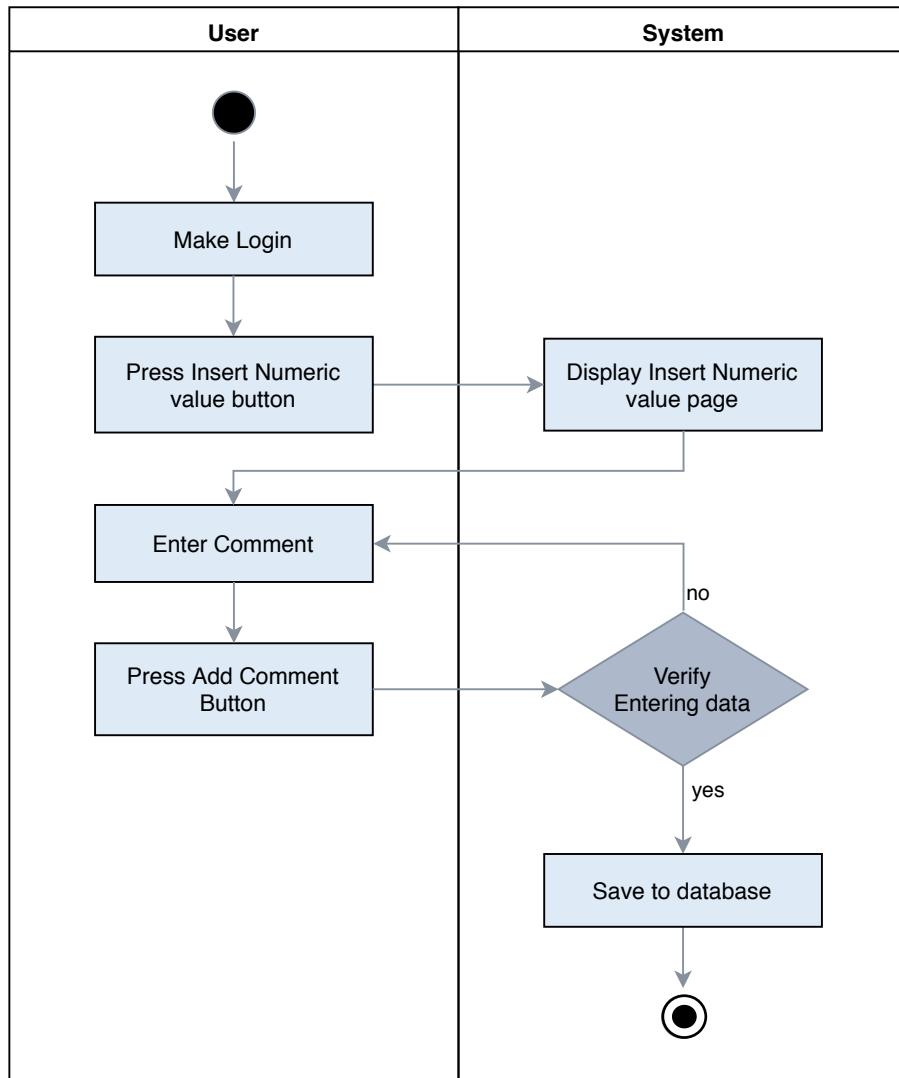


Figure 3.25: Activity diagram of Add Comment

## 6.2 Update delete Comment

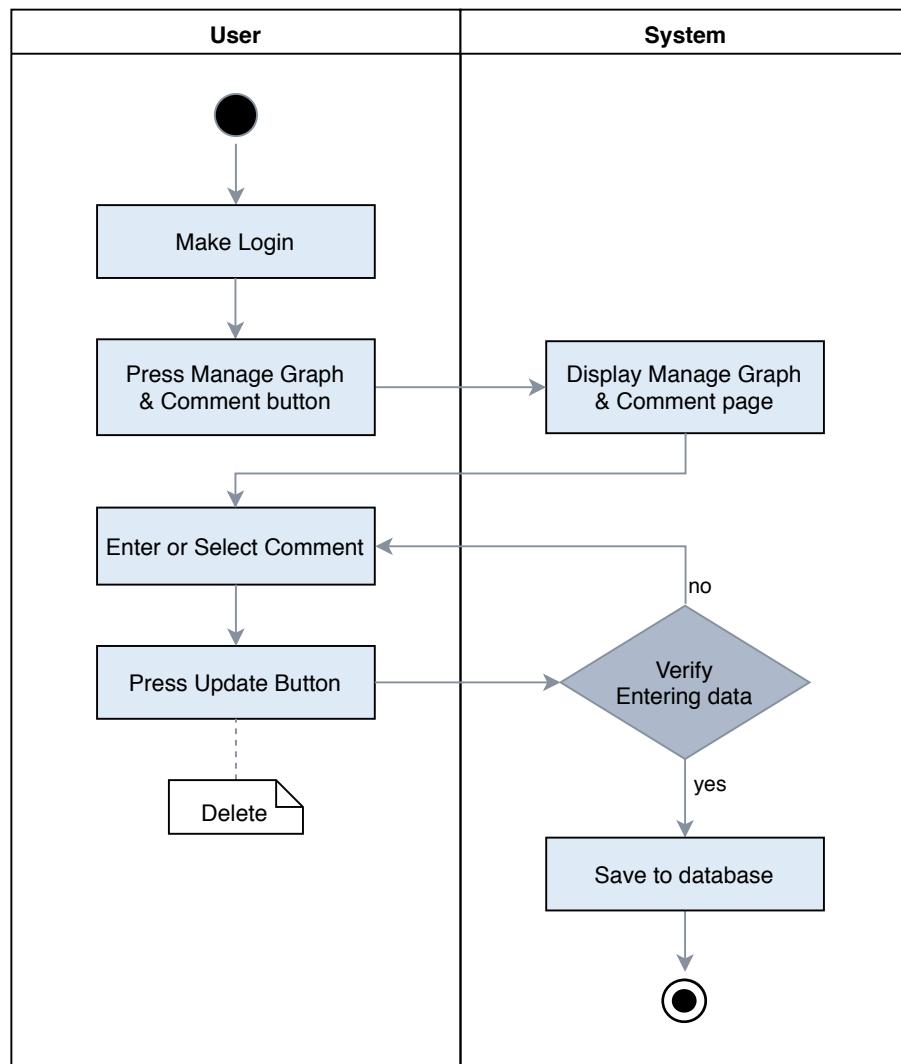


Figure 3.26: Activity diagram of Update and delete Comment

## 7. View Staff Information

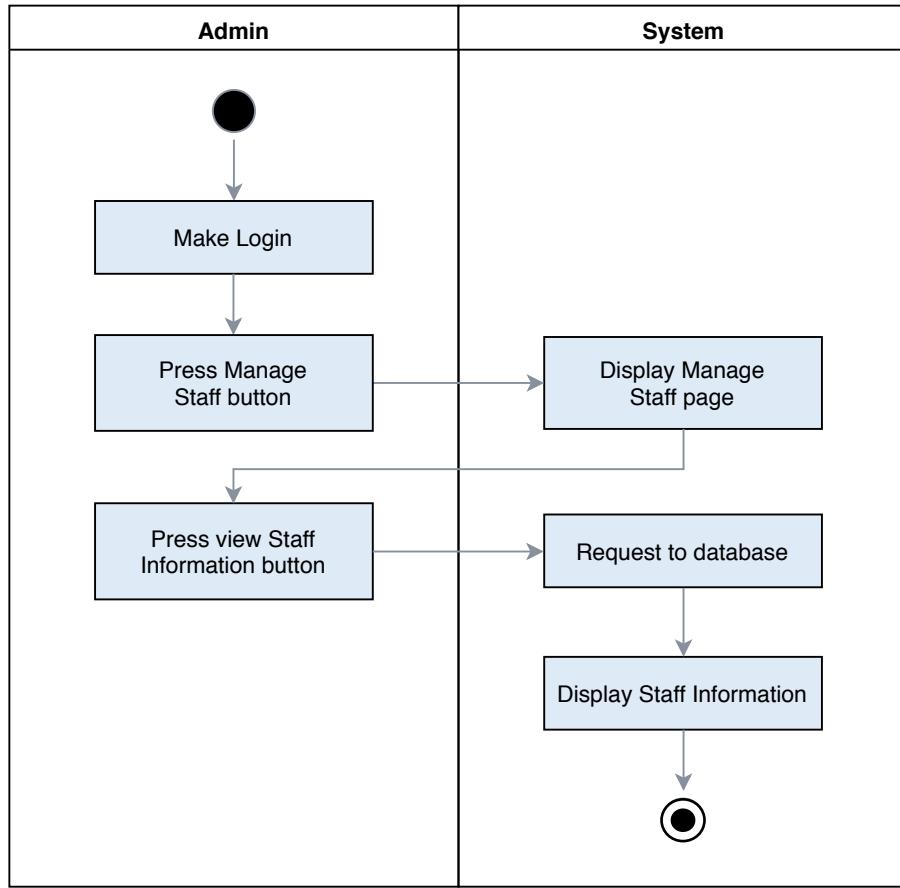


Figure 3.27: Activity diagram of View Staff Information

### 8. View Indicator

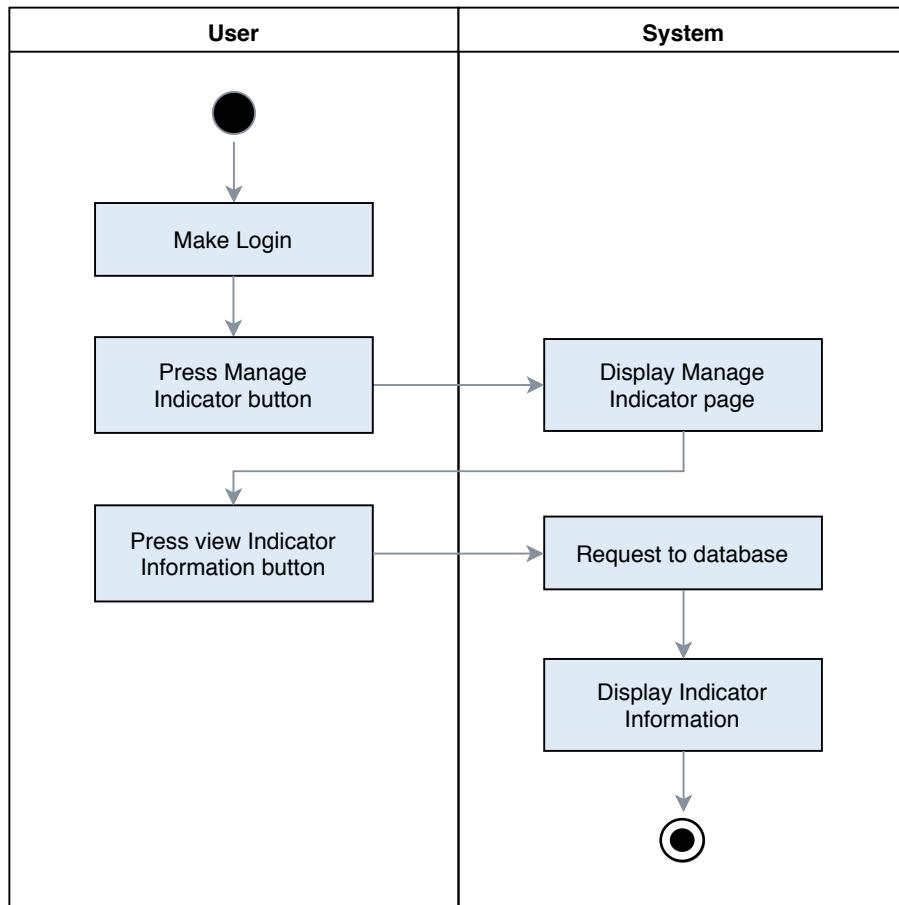


Figure 3.28: Activity diagram of View Indicator

## 9. View Graph

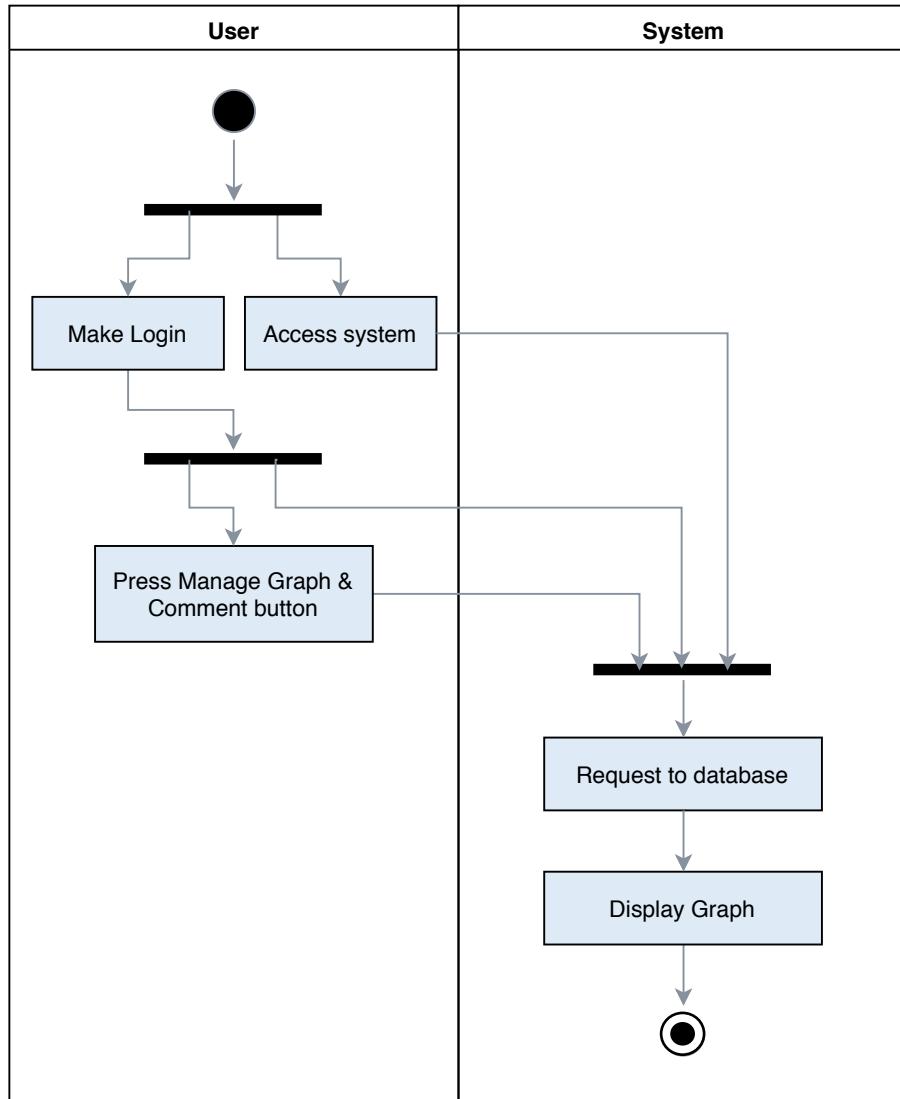


Figure 3.29: Activity diagram of View Graph

## 10. View Comment

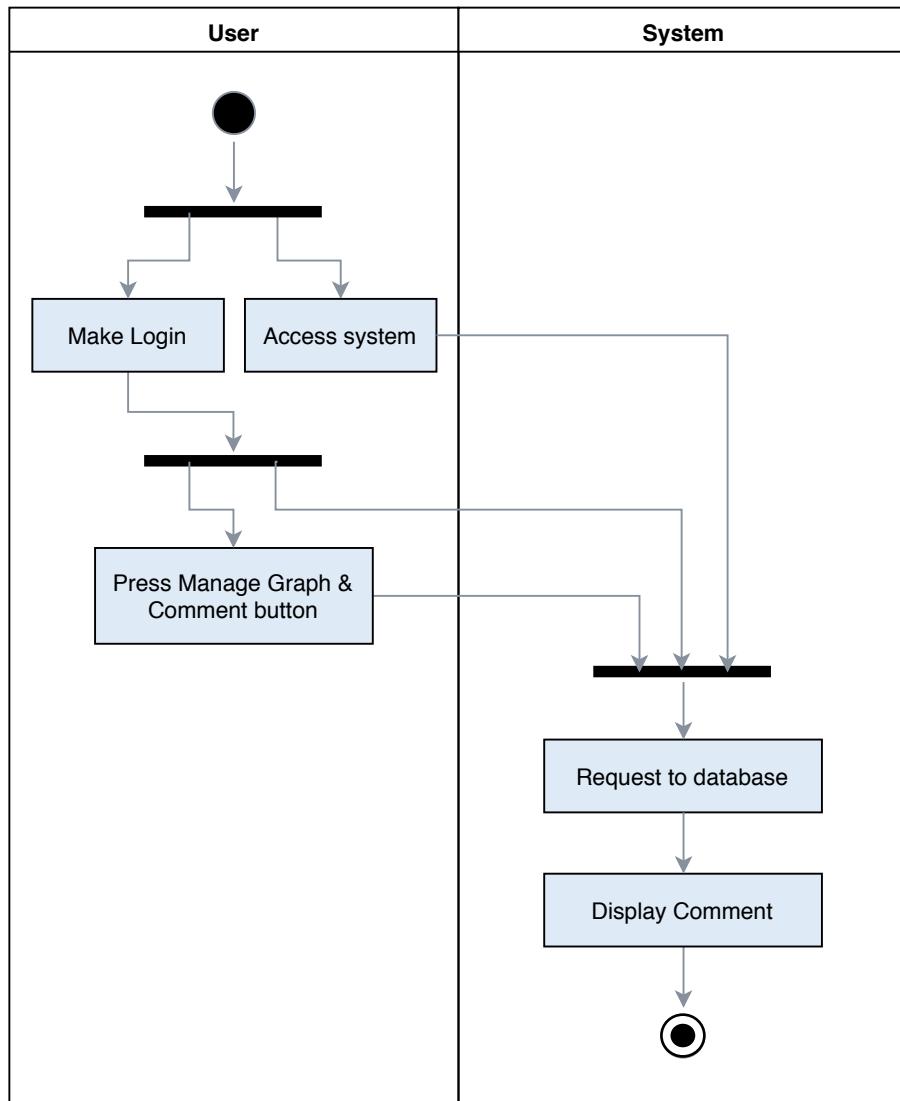


Figure 3.30: Activity diagram of View Comment

## 11. Change Password

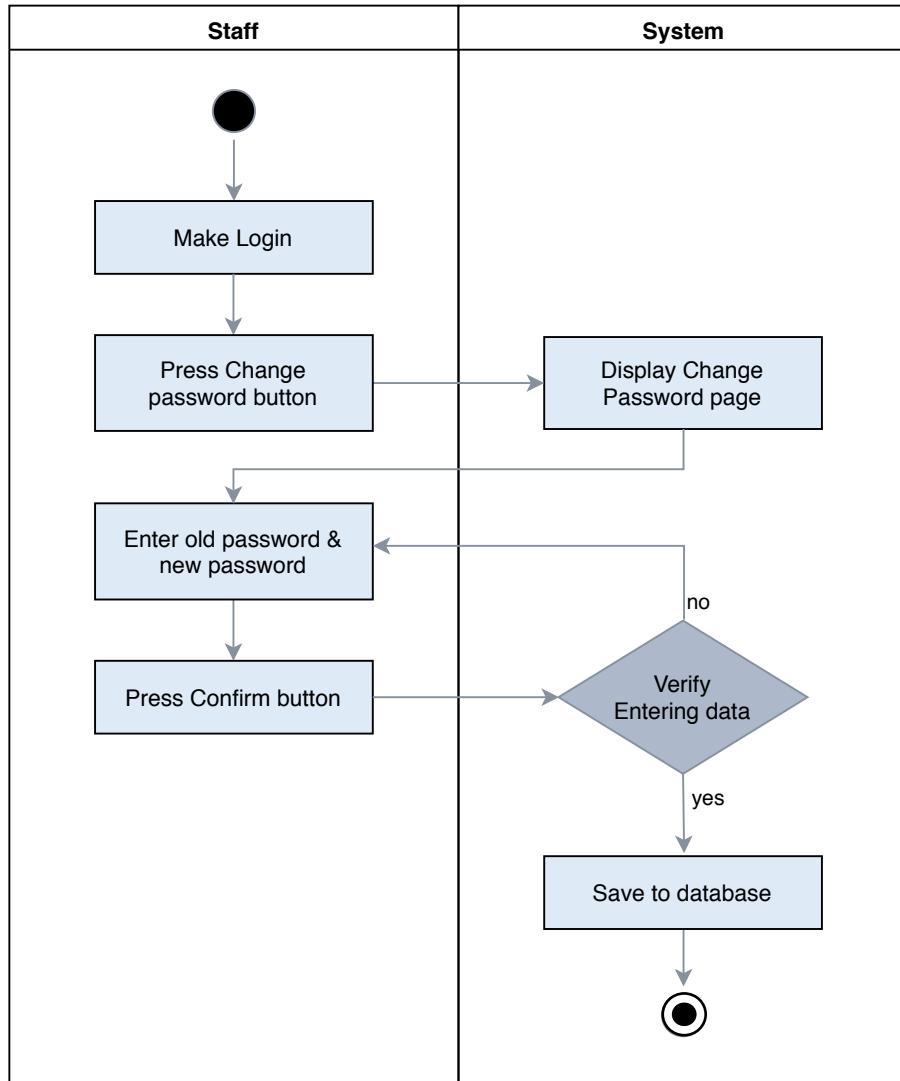


Figure 3.31: Activity diagram of Change Password

## 3.2 DESIGN

### 3.3.1 ER diagram

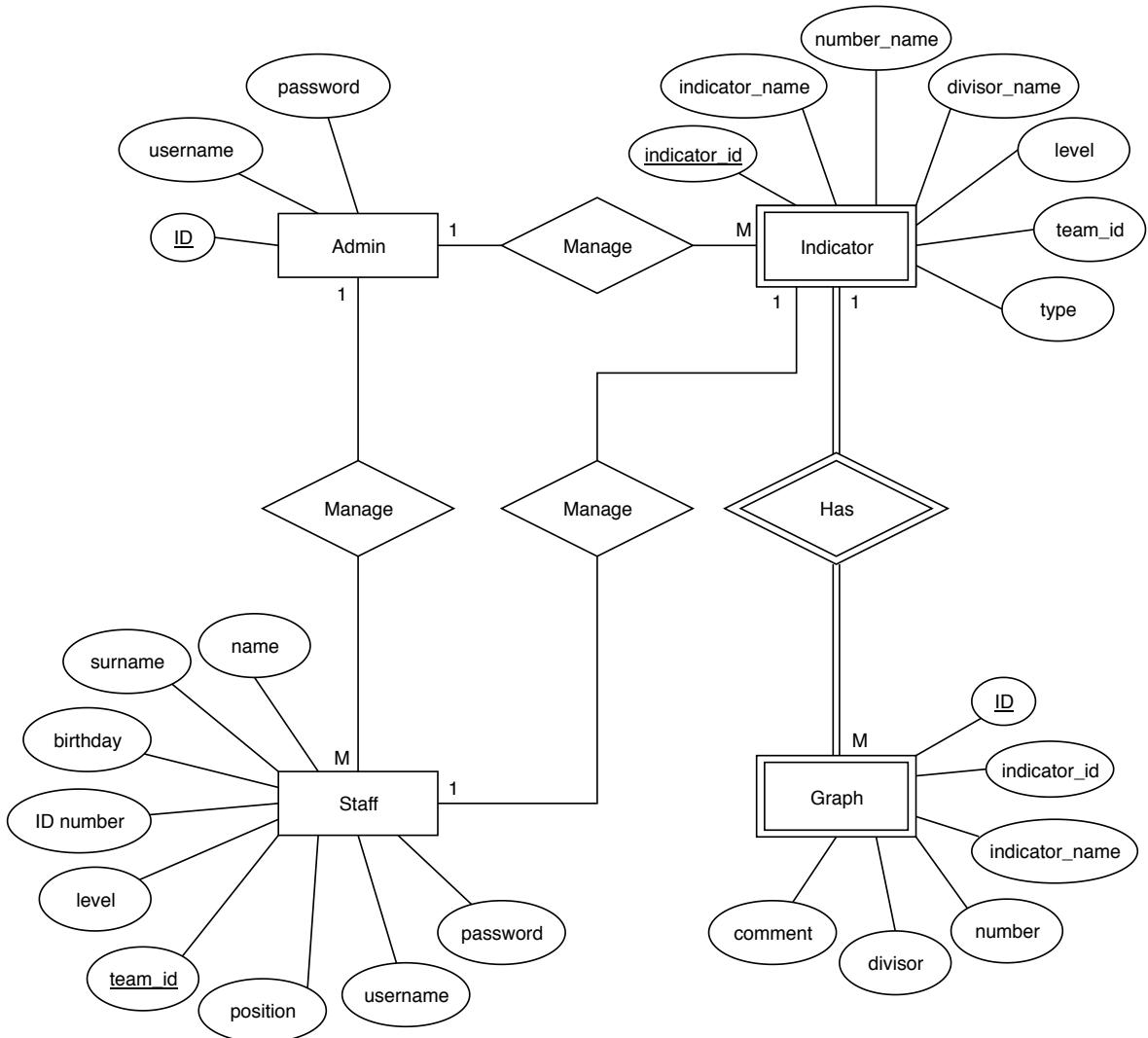


Figure 3.32: ER diagram of Yarang Hospital

### 3.3.2 File structure

<b>Table Name: Admin</b>							
<b>Description:</b> This table is used to keep the admin username and password							
Name	Type	Length	Description	Null	Key	Reference	
admin_id	int	10	Auto increment	no	PK	-	
admin_username	varchar	20	Username of admin	no	-	-	
admin_password	varchar	20	Password of admin	no	-	-	
<b>Total</b>		30					

*Table 3.15: List of Admin table*

<b>Table Name: Staff</b>							
<b>Description:</b> This table is used to keep the staff's information							
Name	Type	Length	Description	Null	Key	Reference	
staff_name	varchar	20	Name of staff	no	-	-	
staff_surname	varchar	20	Surname of staff	no	-	-	
staff_birthday	varchar	20	Birthday of staff	no	-	-	
staff_id_num	int	20	ID number of staff	no	-	-	
staff_level	varchar	20	Level of staff	no	-	-	
staff_team	varchar	20	Team of staff	no	PK	-	
staff_position	varchar	20	Position of staff	no	-	-	
staff_username	varchar	20	Username of staff	no	-	-	
staff_password	varchar	20	Password of staff	no	-	-	
<b>Total</b>		180					

*Table 3.16: List of Staff table*

<b>Table Name: Indicator</b>							
<b>Description:</b> This table is used to keep the indicator's information							
<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Description</b>	<b>Null</b>	<b>Key</b>	<b>Reference</b>	
indicator_id	varchar	20	ID of indicator	no	PK	-	
indicator_name	varchar	20	Name of indicator	no	-	-	
indicator_number_name	int	10	Numeric value name of indicator	no	-	-	
indicator_divisor_name	int	10	Divisor name of indicator	no	-	-	
indicator_level	varchar	20	Level of indicator	no	-	<a href="#">Staff</a>	
indicator_team	varchar	20	Team of indicator	no	FK	<a href="#">Staff</a>	
indicator_type	int	10	Type of indicator	no	-	-	
<b>Total</b>		110					

*Table 3.17: List of Indicator table*

<b>Table Name: Graph_year</b>							
<b>Description:</b> This table is used to keep the indicator's numeric value to perform the yearly graph and comment							
<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Description</b>	<b>Null</b>	<b>Key</b>	<b>Reference</b>	
g_year_id	int	10	Auto increment	no	PK	-	
indicator_id	varchar	20	ID of indicator	no	FK	<a href="#">Indicator</a>	
indicator_name	varchar	20	Name of indicator	no	-	<a href="#">Indicator</a>	
number1	int	10	Number value	no	-	-	
divisor1	int	10	Number divisor	no	-	-	
comment	varchar	1000	Comment of graph	no	-	-	
<b>Total</b>		1070					

*Table 3.18: List of Graph\_year table*

<b>Table Name: Graph_6month</b>						
<b>Description:</b> This table is used to keep the indicator's numeric value to perform the half yearly graph and comment						
<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Description</b>	<b>Null</b>	<b>Key</b>	<b>Reference</b>
g_6month_id	int	10	Auto increment	no	PK	-
indicator_id	varchar	20	ID of indicator	no	FK	<a href="#">Indicator</a>
indicator_name	varchar	20	Name of indicator	no	-	<a href="#">Indicator</a>
number1	int	10	Number value	no	-	-
divisor1	int	10	Number divisor	no	-	-
number2	int	10	Number value	no	-	-
divisor2	int	10	Number divisor	no	-	-
comment	varchar	1000	Comment of graph	no	-	-
<b>Total</b>		1090				

Table 3.19: List of Grapg\_6month table

<b>Table Name: Graph_3month</b>						
<b>Description:</b> This table is used to keep the indicator's numeric value to perform the quarterly graph and comment						
<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Description</b>	<b>Null</b>	<b>Key</b>	<b>Reference</b>
g_3month_id	int	10	Auto increment	no	PK	-
indicator_id	varchar	20	ID of indicator	no	FK	<a href="#">Indicator</a>
indicator_name	varchar	20	Name of indicator	no	-	<a href="#">Indicator</a>
number1	int	10	Number value	no	-	-
divisor1	int	10	Number divisor	no	-	-
number2	int	10	Number value	no	-	-
divisor2	int	10	Number divisor	no	-	-
number3	int	10	Number value	no	-	-
divisor3	int	10	Number divisor	no	-	-
number4	int	10	Number value	no	-	-
divisor4	int	10	Number divisor	no	-	-
comment	varchar	1000	Comment of graph	no	-	-
<b>Total</b>		1130				

Table 3.20: List of Graph\_3month table

<b>Table Name: Graph_month</b>						
<b>Description:</b> This table is used to keep the indicator's numeric value to perform the monthly graph and comment						
Name	Type	Length	Description	Null	Key	Reference
g_month_id	int	10	Auto increment	no	PK	-
indicator_id	varchar	20	ID of indicator	no	FK	<a href="#">Indicator</a>
indicator_name	varchar	20	Name of indicator	no	-	<a href="#">Indicator</a>
number1	int	10	Number value	no	-	-
divisor1	int	10	Number divisor	no	-	-
number2	int	10	Number value	no	-	-
divisor2	int	10	Number divisor	no	-	-
number3	int	10	Number value	no	-	-
divisor3	int	10	Number divisor	no	-	-
number4	int	10	Number value	no	-	-
divisor4	int	10	Number divisor	no	-	-
number5	int	10	Number value	no	-	-
divisor5	int	10	Number divisor	no	-	-
number6	int	10	Number value	no	-	-
divisor6	int	10	Number divisor	no	-	-
number7	int	10	Number value	no	-	-
divisor7	int	10	Number divisor	no	-	-
number8	int	10	Number value	no	-	-
divisor8	int	10	Number divisor	no	-	-
number9	int	10	Number value	no	-	-
divisor9	int	10	Number divisor	no	-	-
number10	int	10	Number value	no	-	-
divisor10	int	10	Number divisor	no	-	-
number11	int	10	Number value	no	-	-
divisor11	int	10	Number divisor	no	-	-
number12	int	10	Number value	no	-	-
divisor12	int	10	Number divisor	no	-	-
comment	varchar	1000	Comment of graph	no	-	-
<b>Total</b>		1290				

Table 3.21: List of Graph\_month table

### 3.3.3 Interface design

#### Home page

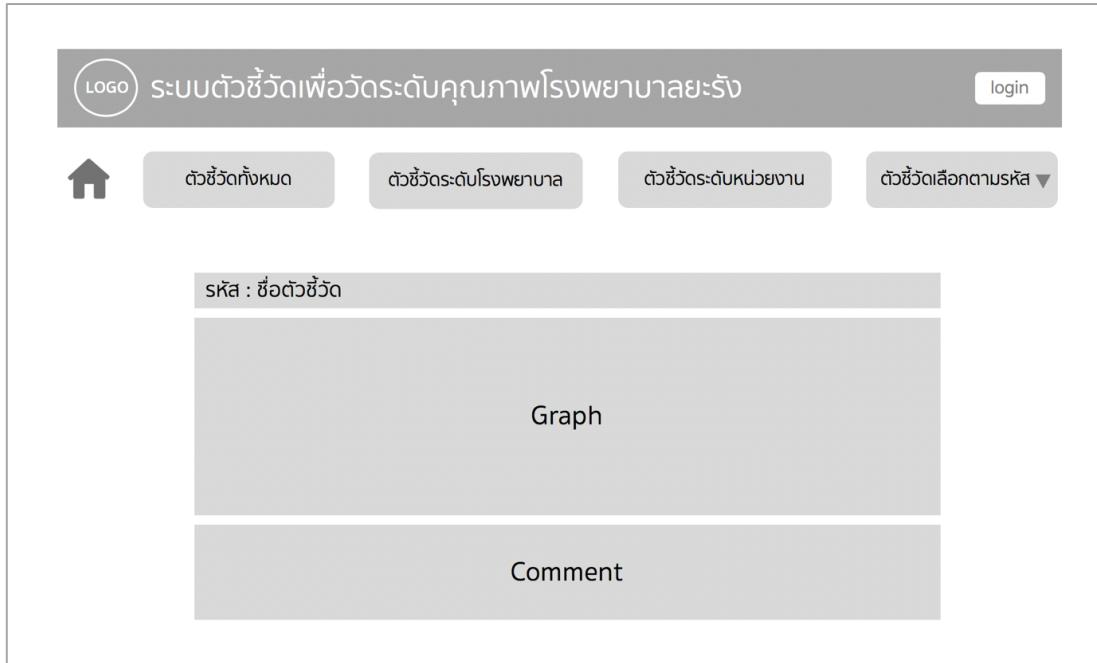


Figure 3.33: Home page

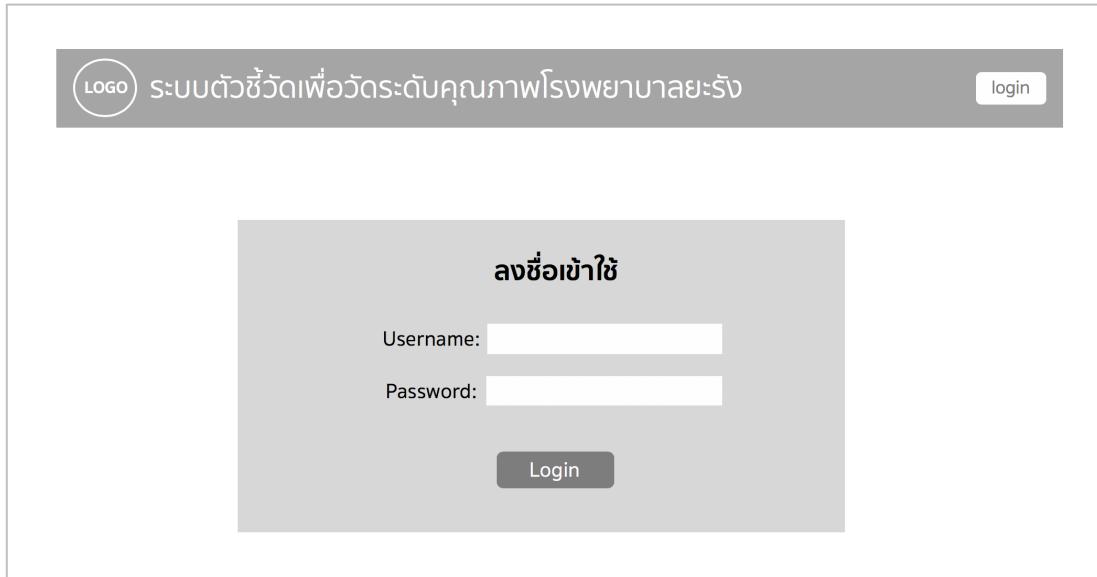


Figure 3.34: login page

## Admin page

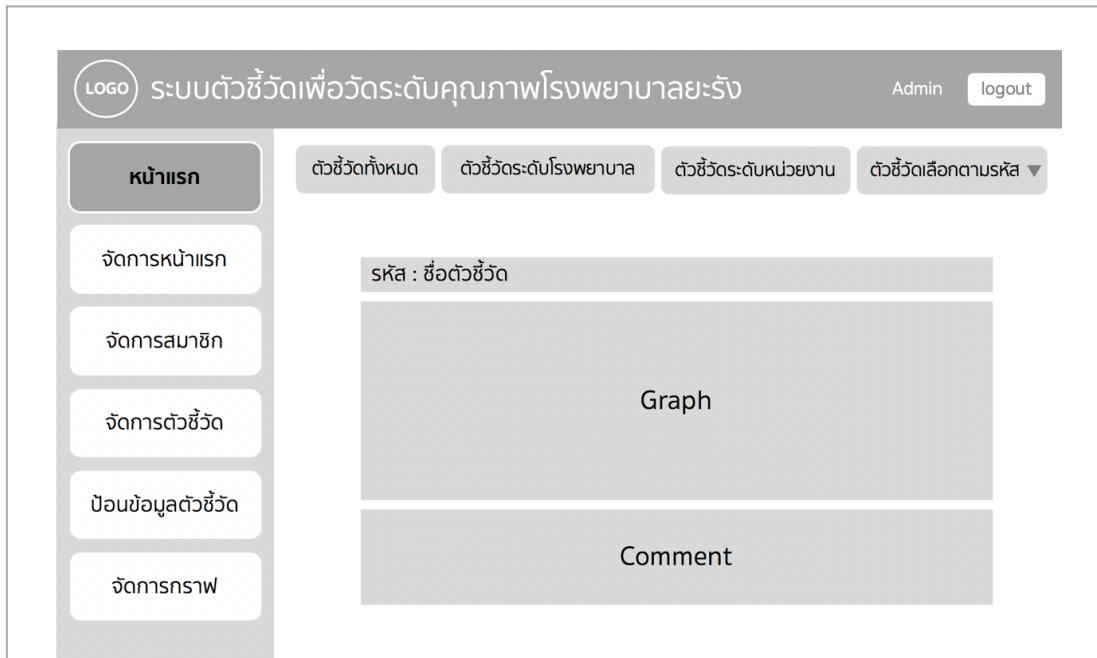


Figure 3.35: Main page for admin

The screenshot shows the "Manage homepage" page. The sidebar is identical to Figure 3.35. The main area displays a table of indicators:

รหัส	ชื่อตัวชี้วัด	ระดับ	กีบ	ดูกราฟ	แสดงหน้าแรก
YRH0001	ตัวชี้วัดโรงพยาบาล	โรงพยาบาล	โรงพยาบาล	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PTC0013	ตัวชี้วัดระบบยา	โรงพยาบาล	กีบระบบยา	<input checked="" type="checkbox"/>	<input type="radio"/>
ADM0007	ตัวชี้วัดงานการเจ้าหน้าที่	หน่วยงาน	งานการเจ้าหน้าที่	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3.36: Manage homepage page for admin

Username	ชื่อ-นามสกุล	ระดับ	ทีม	ดู	ลบ
Harwea-QSC	新加 姜松	โรงพยาบาล	ศูนย์คุณภาพ		

Figure 3.37: Manage staff page for admin

รหัส	ชื่อตัวชี้วัด	ความถี่	ทีม	ดู	ลบ
YRH0001	ตัวชี้วัดโรงพยาบาล	รายเดือน	ศูนย์คุณภาพ		
DH0101	อัตราการเสียชีวิต	รายเดือน	บริหารความเสี่ยง		

Figure 3.38: Manage indicator page for admin

The screenshot shows a web application interface. On the left, there is a vertical sidebar with rounded corners containing six buttons:

- หน้าแรก
- จัดการหน้าแรก
- จัดการสมาชิก
- จัดการตัวชี้วัด
- ป้อนข้อมูลตัวชี้วัด** (This button is highlighted in dark grey)
- จัดการกราฟ

The main content area has a title "ป้อนข้อมูลตัวชี้วัด". Below the title are two input fields: "รหัส" (Code) set to "โปรดเลือก" (Please select) with a dropdown arrow, and "ปี พ.ศ." (Year) set to "2560" with a "ยืนยัน" (Confirm) button next to it.

Figure 3.39: Insert numeric values page for admin (select indicator)

This screenshot shows the same application interface as Figure 3.39, but with more detailed search parameters. The "รหัส" field now contains "DH0101-อัตราการเสียเชื้อต" and the "ปี พ.ศ." field is still "2560". Below these fields, there are three dropdown menus labeled "ความถี่" (Frequency), "รายเดือน" (Monthly), "ตัวตั้ง" (Root), and "ตัวหาร" (Divisor). To the right of these dropdowns is a numeric keypad with a "แสดงกราฟ" (Show Graph) button at the bottom right.

Figure 3.40: Insert numeric values page for admin (insert numeric values)

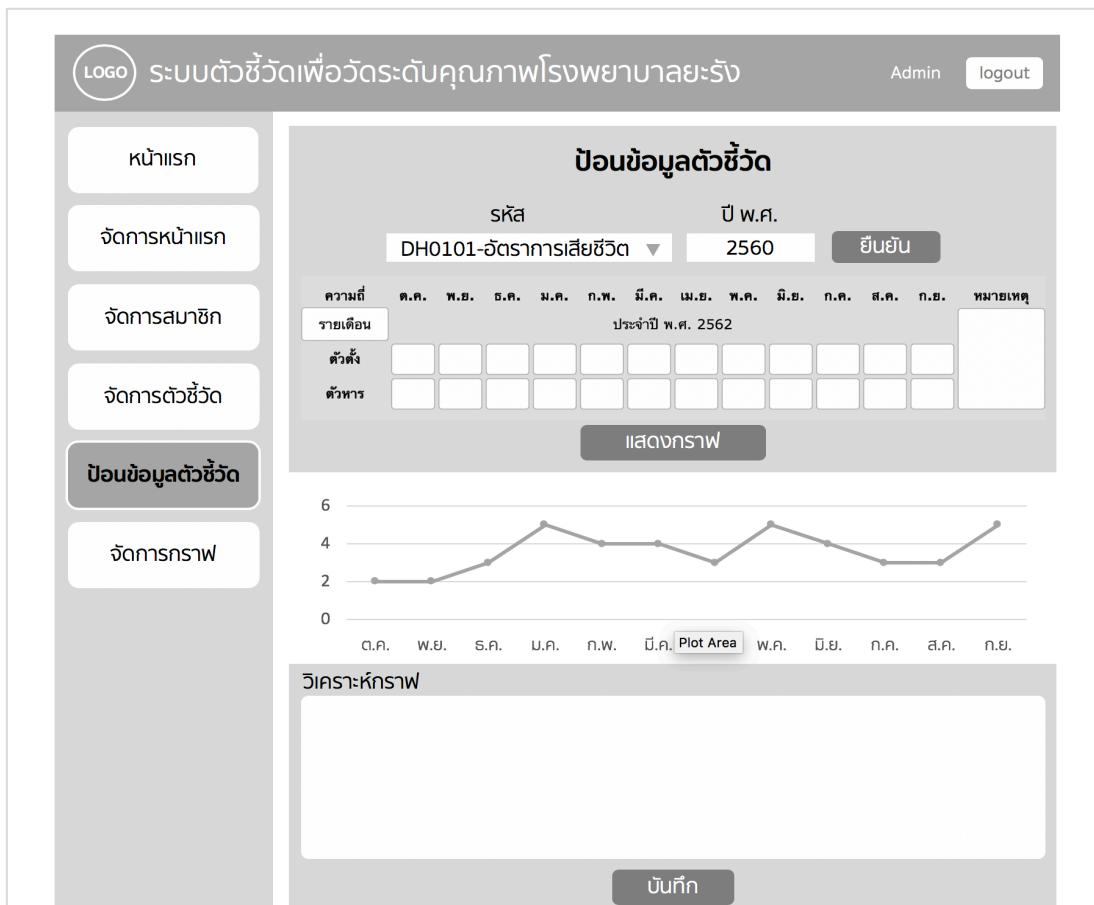


Figure 3.41: Insert numeric values page for admin (show graph and comment)

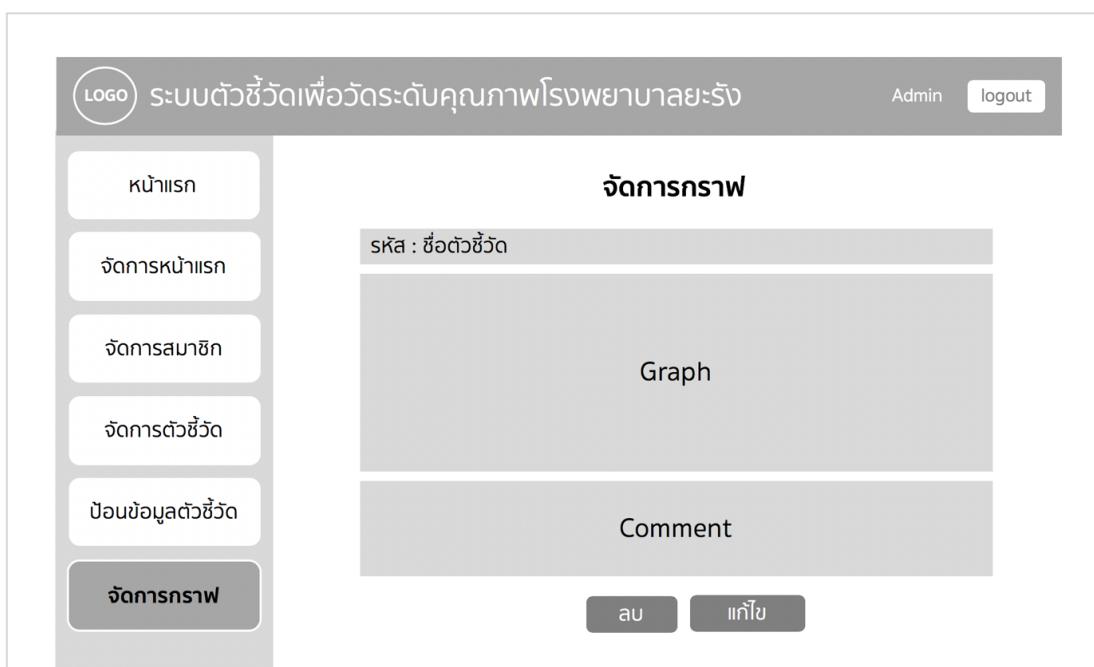


Figure 3.42: Manage graph page for admin

## Staff page

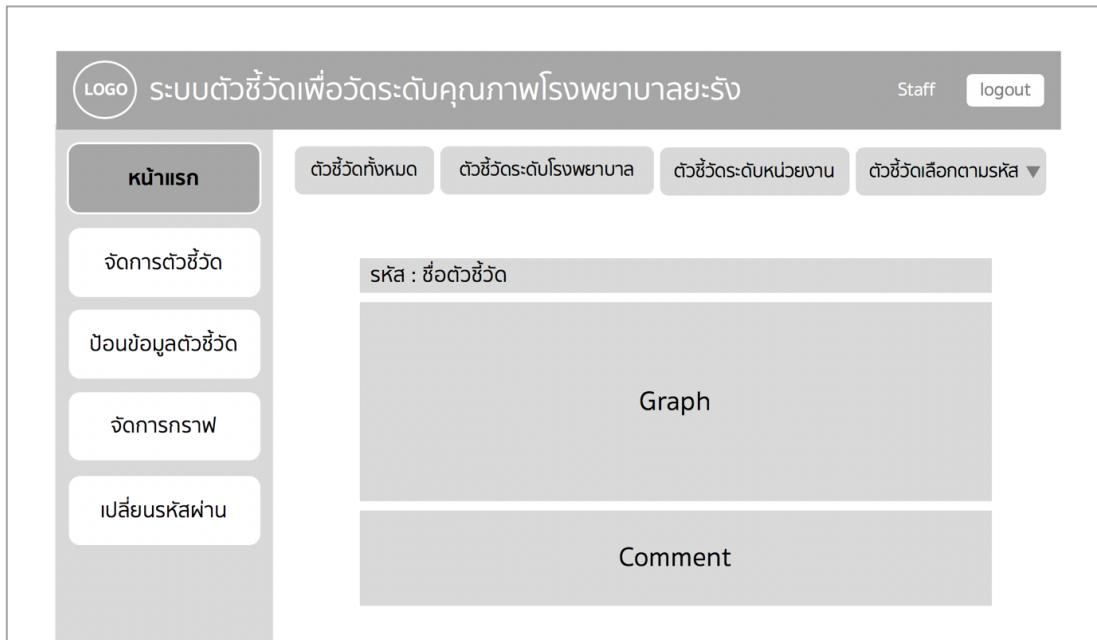


Figure 3.43: Main page for staff

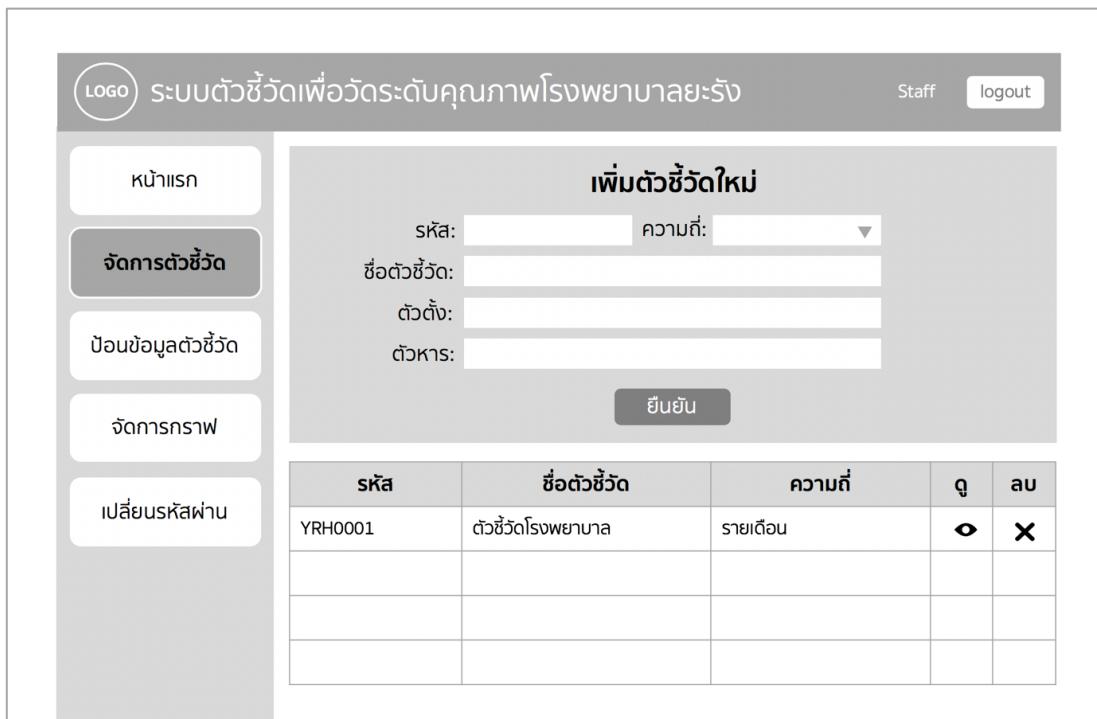


Figure 3.44: Manage indicator page for staff

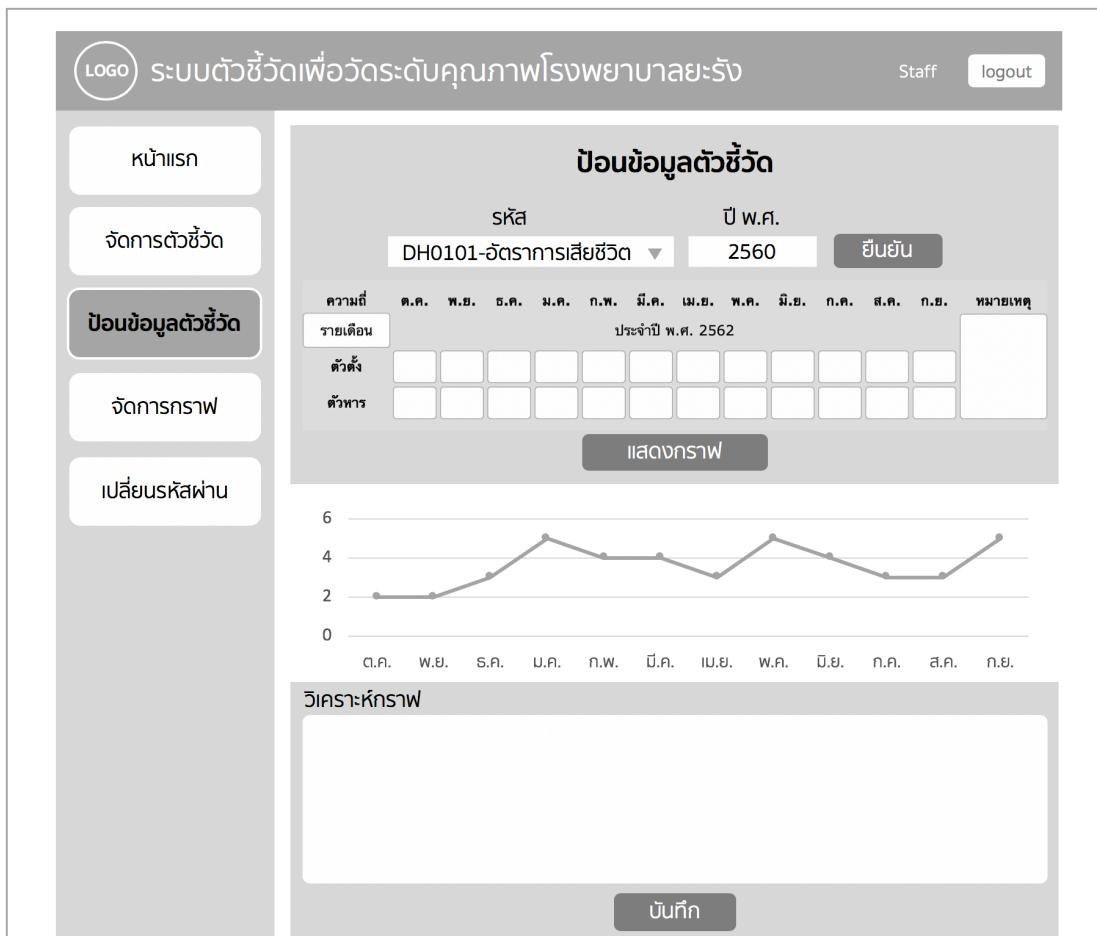


Figure 3.45: Insert numeric values page for staff



Figure 3.46: Manage graph page for staff



Figure 3.47: Change password page for staff

## REFERENCES

- Aghayev, E., Staub, L., Dirnhofer, R., Ambrose, T., Jackowski, c., Yen, K., Bolliger, S., Christe, A., Roeder, C., Aebi, M., & Thali, A. 2007. Virtopsy – The concept of a centralized database in forensic medicine for analysis and comparison of radiological and autopsy data. Institute of Forensic Medicine & Institute for Evaluative Research in Orthopaedic Surgery, University of Bern and Institute of Diagnostic Radiology, University Hospital of Bern, Switzerland.
- Hübner-Bloder, G. & AmmenwerthKevin, E. 2018. Key Performance Indicators to Benchmark Hospital Information Systems – A Delphi Study. Institute for health information systems, UMIT-University for Health Sciences, Medical Informatics and Technology, Hall in Tyrol, Austria.
- Jahn, F. & Winter, A. 2011. A KPI framework for process-based benchmarking of hospital information systems. University of Leipzig, Institute for Medical Informatics, Statistics and Epidemiology, Leipzig, Germany.
- Kononenko, K. 2019. Key Performance Indicator. Retrieved December 17, 2019, from <https://databox.com/what-is-a-kpi>
- Onsman, A. 2018. Centralized Database Management System. Retrieved December 17, 2019, from <https://www.tutorialspoint.com/Centralized-Database-Management-System>
- Sarycheva, Y. 2019. Waterfall model. Retrieved December 23, 2019, from <https://xbsoftware.com/blog/software-development-life-cycle-waterfall-model/>