KSMART TRACKING AND MANAGING DIETARY BEHAVIOUR OF CKD PATIENTS

การติดตาม และการให้คำแนะนำ พฤติกรรมการบริโภคของผู้ป่วยโรคไต

BY

MISS KULYARAT	**LEELATHUSANATHORN**	**6088052**
MISS NOPPARATH	**NGOWTRAKUL**	**6088098**
MISS SUPISARA	**YINGSOM**	**6088230**
MR. KRITSADA	**WITTAYAKITJAKUL**	**6088223**

ADVISOR
DR. PATTANASAK MONGKOLWAT

A Senior Project Submitted in Partial Fulfillment of the Requirements for

THE DEGREE OF BACHELOR OF SCIENCE (INFORMATION AND COMMUNICATION TECHNOLOGY)

Faculty of Information and Communication Technology
Mahidol University

2020

ACKNOWLEDGEMENTS

To finish this project successfully, we must put efforts for achieving the best final result that comes from many advice, guidance and assistance from many groups of related people.

Therefore, we would like to express our respect and appreciation to Prof. Dr. Pattanasak Mongkolwat, our advisor who gave a great opportunity to do this project. We appreciate his patience, motivation, and belief that he gave to us all the time.

Besides, we are thankful for the helps from Assoc. Prof. Anirut Pattaragarn and his team for providing medical information, CKD's knowledge, nutrition information and requirements to do the project.

Moreover, we are grateful to have our Co-advisor, Prof. Dr. Pagaporn Pengsart for the guidance about database, cloud storage and firebase cloud storage.

Certainly, we delighted to express our gratitude to our beloved families for their support with resources and advice throughout our four academic years.

Miss Kulyarat Leelathusanathorn
Miss Nopparath Ngowtrakul
Miss Supisara Yingsom
Mr. Kritsada Wittayakitjakul

KSMART: TRACKING AND MANAGING DIETARY BEHAVIOUR OF CKD PATIENTS

MISS KULYARAT LEELATHUSANATHORN 6088052 ITCS/B
MISS NOPPARATH NGOWTRAKUL 6088098 ITCS/B
MISS SUPISARA YINGSOM 6088230 ITCS/B
MR. KRITSADA WITTAYAKITJAKUL 6088223 ITCS/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: DR. PATTANASAK MONGKOLWAT

ABSTRACT

kSmart mobile is a tool for improving efficiency of CKD treatment and monitoring CKD patient food intake. It will be deployed on two platforms which are mobile and website. It is a meal and medicine recording application for CKD patients. kSmart app is available on both iOS and Android operating system. kSmart web application is a dashboard for doctors to track a patient's consumption behavior.

Developers expect that this project will give benefits to CKD patients. kSmart appl, improvs efficiency of CKD treatment to be easier by tracking patient’s behavior, and reduces the time that patient must be in a hospital. Initially, developers must gather requirements from the doctors, pharmacists, and nutritionists from Siriraj hospital to design the software architecture and workflow. They gave a consultant about the things that should be improved in CKD treatment and the necessary information, such as medicine information, phosphorus intake formula, and the list of foods and ingredients that have been declared the phosphorus value for building the application.

The functionalities that include in the mobile application are expected to help CKD patients and to motivate them to always record their food consumption. Besides, the dashboard on a web application is expected to help the doctors to track the patient’s behavior easily and collect the statistical data for the study purposes.

KEYWORDS: HEALTHCARE, CKD PATIENT, HOSPITAL, MEDICAL RECORD APPLICATION, FOOD RECORD APPLICATION, TRACKING CONSUMING BEHAVIOUR WEB APPLICATION, DIETARY

147 P.

การติดตาม และการให้คำแนะนำ พฤติกรรมการบริโภคของผู้ป่วยโรคไต

นางสาวกัลยรัตน์ ลีละทัศนาธร 6088052 ITCS/B
นางสาวนพรัตน์ โง้วตระกูล 6088098 ITCS/B
นางสาวศุภิสรา ยิ่งสม 6088230 ITCS/B
นายกฤษดา วิทยกิจจากุล 6088223 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ดร. พัฒนศักดิ์ มงคลวัฒน์

บทคัดย่อ

kSmart แอพลิเคชัน เป็นแอพลิเคชันบนที่มีส่วนช่วยในกระบวนการรักษาโรคไตเรื้อรัง โดยแบ่งเป็น 2 แพลตฟอรึมคือ แอพลิเคชันบนมือถือ และเว็บไซต์ โดยแอพลิเคชันบนมือถือรองรับทั้งระบบ Android และ iOS ใช้เพื่อดูข้อมูลฟอาฟอรัสในอาหาร บันทึกการรับประทานอาหารเพื่อคำนวนฟอสฟอรัส และบันทึกการรับประทานยาของคนไข้เป็นหลัก ในส่วนของเว็บไซต์จัดทำขึ้นเพื่อให้หมอสามารถดูข้อมูลของคนไข้ และข้อมูลอาหารที่คนไข้รับประทาน รวมทั้งข้อมูลฟอสฟอรัส เพื่อนำมาคำนวณทางสถิติเพื่อใช้ศึกษา และรักษาคนไข้โรคไตเรื้อรังต่อไปได้

ผู้พัฒนาหวังเป็นอย่างยิ่งว่าโปรเจคนี้จะเป็นประโยชน์ต่อการแพทย์ และช่วยเหลือในการปรับปรุงการรักษาคนไข้โรคไตเรื้อรัง เพื่อให้ง่ายต่อการติดตามพฤติกรรมของผู้ป่วย รวบรวมข้อมูลที่จำเป็นต่อการศึกษา และวินิจฉัย รวมทั้งตัดสินใจในการรักษา ในขั้นต้น ผู้พัฒนาได้พูดคุยกับหมอ นักโภชนาการ และเภสัชกร ของโรงพยาบาลศิริราช เพื่อรับข้อมูลที่จำเป็นในการพัฒนาแอพลิเคชัน เช่น รายกาอาหารพร้อม ข้อมูลฟอสฟอรัส และรายการยาที่ใช้รวมถึงรับปัญหา หรือความต้องการของหมอ เพื่อนำมาออกแบบพัฒนาแอพลิเคชัน

ในแอพลิเคชันประกอบด้วยฟังก์ชันที่ผู้พัฒนาคาดหวังว่าจะช่วยให้ผู้ป่วยสามารถใช้งานแอพลิเคชันอย่างง่ายดาย และสามารถบันทึกมื้ออาหารได้อย่างสม่ำเสมอ ซึ่งจะมีส่วนช่วยอย่างมากในการรักษาตัวผู้ป่วยเอง

CONTENTS

Page

CONTENTS (Cont.)

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1
INTRODUCTION

Chronic Kidney disease or CKD is a major health issue. CKD affects people around 17% of Thai population [1]. CKD has occurred when human kidneys have been damaged via the effects of other disease such as diabetes and high blood pressure [2] that patients have for a long time, months or years [3]. CKD can finally lead to end-stage kidney failure and kidneys to permanently deteriorate [3]. Nowadays, CKD has often occurred in Thai children because their heredity or inappropriate habits (e.g., less exercise, consuming foods or drinks that is too sweet or too fat) [2]. Therefore, CKD's treatment is the one of significant factors that needs to be managed carefully to maintain healthy kidney functions. According to online medical document [3], National Kidney Foundation divides CKD into five stages that will be described in Chapter 2. To cure the disease, doctors must decide to use suitable treatment plans and medications, depending on the CKD's phase of each patient [3].

Besides, dietary behavior modification is one of the factors to support CKD treatment too. Since the kidneys of CKD patients do not function well, they must carefully consume foods within a specific condition. For example, CKD patients must eat carbohydrates in every meal and avoid eating salty food, high phosphorus food, and high cholesterol food [4]. Several studies suggest that higher intakes of phosphorus are associated with an increased risk of cardiovascular disease. The more phosphorus level from your diet, the more calcium level in your body needs. The delicate balance between calcium and phosphorus is necessary for proper bone density and the prevention of osteoporosis [5].

Consequently, how do CKD patients know which dishes they can eat with safety? This project aims to develop an application that will help CKD patients to manage food consumption and level of phosphorus in their meal and record their meal to help in CKD treatment.

This chapter is an introduction to the kSmart project, which is the project to develop an application for tracking CKD's patient's consuming behavior. There are

six sections, which are ‘motivation’, ‘problem statement’, ‘objectives of the project’, ‘scope of the project’, expected benefit’, and ‘organization of the document’.

1.1 **Motivation**

At present, health is especially important for people because health is the one thing that people must properly maintain physical well-being and a healthy mind. If people are healthy, they can live a life without suffering from any aches, pain, or discomfort. Consequently, taking good care of one’s body and mind is becoming a new trend for better living. Everyone should focus on health more than before and the best way to be healthy is having well-balanced meals because we are what we eat. If a patient eats well-balanced diets, you can have good health. People can manage to have carbohydrates, protein, vegetable, and fat in each meal, but how can the patient manage the intake of “Phosphorus” and “Calcium”. Too much phosphorus level is a cause of CKD, and too much calcium can cause to other disease like stones in unary tract or brain disease. However, calcium is the main of phosphorus binder treatment (will be described in Chapter 2) and these minerals are balanced and work together to make bones and teeth strong. The human’s body has 85% of phosphorus is in bones and teeth. Phosphorus helps percolate out waste products in the kidneys. It also helps decrease muscle pain after an exercise. Phosphorus is also needed to help balance and use other vitamins and minerals, including vitamin D, iodine, magnesium, and zinc. If human body have unbalanced phosphorus and calcium value, it will be the cause of several diseases. Some medications and foods can decrease phosphorus. Symptoms of phosphorus deficiency include loss of appetite, anxiety, bone pain, fragile bones, stiff joints, fatigue, irregular breathing, irritability, numbness, weakness, and weight change. In children, decreased growth and poor bone and tooth development may occur. The most dangerous symptom is chronic kidney disease or CKD that means the kidneys are damaged permanently and they cannot filter blood as well as normal. CKD can grow up to be end stage renal disease, so that CKD in children is very important to be careful. If the disease occurred, parents must take care their kids very closely and observe the symptom. Because the disease is not completely curable, but the treatment can bring the waste out of kidneys and make patients feel better. National Kidney

Foundation [6] [7] has estimated and group patients by CKD stages and ranges of age. CKD in kids is divided by age in range 0-5 years old, and more than five years old. Mostly, we founded CKD in 0-5 years old because of two causes. The first is congenital abnormalities of the kidney structure, which is genetic disorders, was happened when kids in the mother's womb. The second is because improper care via give a medicine dosage to children without doctor's diagnosis. It is the cause of urinary tract infections. CKD mostly founded in more than five years old kids. The cause is they buy medicine by themself when they have sore throat or dermatitis. That influence immune system in their body. It will change and damage kidneys and it is the cause of chronic kidney disease. Therefore, our group creates a kSmart Application to help users record food consumption from each meal along with calculating and monitoring phosphorus level in each meal. Making kidney disease patient maintain a proper level of phosphorus and make everyone to be healthy [5].

1.2 Problem Statement

In the ideal world, chronic kidney disease (or CKD) patients must control the phosphorus intake from foods in each meal. Since the high phosphorus level in the blood causes unacceptable effects on the human body. For example, a high phosphorus level in the blood causes the human body to pull calcium out of bones to pair with phosphorus in term of chemical composition. It causes bones to become weak and brittle. CKD patients must control phosphorus levels in a range that their kidneys can receive because their kidneys cannot work well in phosphorus excretion. [8] Therefore, CKD patients' consumption behavior is one factor for treating CKD.

1.3 Objectives of the Project

The main objective of this project is to help doctors and CKD patients to control phosphorus level during treatment using kSmart Application.

- To record CKD patients' diet consumption behavior.
- To help doctors obtain a report of each patient's behavior between treatment.
- To calculate phosphorus value in each meal.
- To control phosphorus level of CKD patients.

- To help doctor communicate with patients easily.

1.4 Scope of the Project

kSmart Application is a web and mobile application to help CKD patients record each meal along with calculating and managing phosphorus in each meal. kSmart application will have two roles of the users that are doctor and patient. Each patient account will have the maximum phosphorus level that they can get in each day. Then the patient must record each meal by selecting from the list or add new food in the application, before choosing food to eat. The application will alert a patient to record each meal and notify a patient if the selected food has a high level of phosphorus too much for a patient. Each doctor's account will have a patient list and the doctor can retrieve and view food intake records of each meal. Besides, the doctor can set an appointment schedule for each patient and the patient can get the notification to remind about the appointment.

1.5 Expected Benefits

Benefits to Users

- Provide simple interface that is easy to use for general users.
- The users can use the application to calculate balance phosphorus in food.
- The users can know how much phosphorus they can take each day.
- The user can keep healthy by using the application.
- The doctor can communicate with patients easier.
- The doctor can track patients' consumption behavior.
- The doctor has more information to study CKD.

Benefits to Developers

- Learning to develop web-based application.
- Learning Flutter, Dart, and Android studio to develop cross platform application for iOS and Android smartphone.
- Utilizing knowledge of HTML, CSS, Java and Spring boot to develop web-based application.

- Learning to use firebase to be a database and cloud storage of the application.
- Learning to manage working as a team.

1.6 Organization of the Document

This document consists of six chapters including:

1. Introduction – This chapter includes six sections and it describe the rationales of the project, named motivation, objectives, scopes, expected benefits and the organization of the document.
2. Background – This chapter describes background of web-based application, mobile application and tools that will be used.
3. Analysis and Design – This chapter describes system architecture, system structure chart, process analysis, database design and interface design for the kSmart application. The process analysis will include data flow diagram, use case diagram, use case description, sequence diagram and class diagram with the explanation.
4. Implementation – This chapter contains topics about system environment and implementation guide.
5. Testing and Evaluation – This chapter is about application testing and evaluation including unit test, performance test and system integration test.
6. Conclusion – This chapter will discuss the final version of the application and the benefit to both developers and users. Also, discuss about problems and future work that can be extend from this project.

CHAPTER 2
BACKGROUND

This chapter provides background knowledge and the definition of CKD, blood test, how foods are significant to treat CKD, and describe a flutter framework (for building cross-platform applications between android and iOS). Therefore, base knowledge of CKD, food nutrition and Flutter framework together with technique for evaluation and literature review are declared and discussed. The information in this chapter gives the idea to propose the project and describes some methods in the design and development part. Besides, this chapter lists and explains the expected features of the application compared to related works too.

2.1 Chronic Kidney Disease

Chronic kidney disease damage kidneys and make them lose many abilities such as maintaining fluid balance in the body, controlling water-minerals in blood, eliminating waste like drugs and toxins from blood stream and body, and secretion of hormones into the blood streams [9].

The causes of chronic kidney disease usually are diabetes, hypertension, and obesity, and from other conditions such as nephritis, kidney cyst disease.

The National Kidney Foundation-Kidney Disease Outcomes Quality Initiative (NKF-K / DOQI) divides the kidney disease stages into five stages respectively of glomerular filtration rate (GFR) and evidence of kidney disease (National Kidney Foundation 2002). [3]

Stage	Detail	GFR
1	Kidneys are starting to deteriorate (there are proteins in urine), but in this stage GFR value is normal.	Greater than or equal 90
2	Kidneys are deteriorated and GFR value is decreased.	60 - 89

3	GFR values decrease more than 2nd stage.	30 - 59
4	GFR value dropped considerably.	15- 29
5	Kidney's failure	Less than 15

** GFR or Glomerular Filtration Rate is kidneys functions value used to determine the stage of CKD.

Table 2.1 *show the stages of Kidney disease* [9]

2.2 **Blood Test**

The blood testing is important for CKD's diagnostic to measure the creatine (serum creatinine; Scr). Creatinine is the waste from muscle that will be produced everyday with stable value. Creatinine will be excreted from our body via kidneys. Creatinine will be increase if the performance of kidney function is decreased. The creatinine value is used to calculate the glomerular filtration rate (GFR) following formula 2.1 and the normal creatinine value is shown on the table 2.2

If the glomerular filtration rate is high, it means the kidneys is function well. The normal value of the glomerular filtration rate. It will depend on age, gender, and weight.

In general, adults aged 18-30 years is estimated the glomerular filtration rate between 120-130 ml / min / 1.73 sq m, and the values will gradually decrease with increasing age. However, there are factors that affect creatinine result to increase or decrease. It is independent of the glomerular filtration rate such as muscle mass, protein foods, some drugs, etc. [11]

Formula 2.1 $\text{GFR} = \frac{0.413 \times height\ (cm.)}{level\ of\ creatine\ in\ blood\ (mg/dL)}$

Gender	Age	Normal creatinine value
Males	4-20 years	0.2-1.0 mg / dL
Females	4-20 years	0.2-1.0 mg / dL
Males	Adults	0.7-1.3 mg / dL
Females	Adults	0.6-1.1. mg / dL1

Table 2.2 *Show Normal Creatine value according to Age and Gender* [11]

2.3 Nutrition (or Foods)

Nutrition is an important factor to slow down chronic renal failure for the early stage, the middle stage that needs dialysis, and the end-stage of renal failure. Therefore, a responsible and good diet allow the kidneys to work optimally. Consuming food with high level of sodium values or high blood pressure, it will increase the rate of chronic kidney deterioration. Next nutrition is protein. When bodies derive protein, they are broken down into amino acids for absorption and excretion in urea form. If the bodies get the protein in excessive amounts, the kidneys will have to work hard. Finally, the kidneys are the main function of excreting potassium from the body. If patients eat a lot of potassium when the kidneys deteriorated, the kidneys work less causing potassium to remain in the body for too long. Be careful when buying foods that are high in potassium, such as bananas, oranges, or colored fruits, processed foods, milk, nut butters that need to be careful and reduce the amount. [12]

The most significant nutrition for CKD patients is Phosphorus or phosphate, which is the one mineral that contains in any food or drink that we have consumed. However, too high phosphorus value in the blood is the cause of fast kidneys deteriorated in CKD patients and drawback on the cardiovascular system, which is the significant cause of death in CKD patients.

In general, the phosphorus level is still in the normal range (according to table 2.3) in CKD stage 1-2. In contrast, the phosphorus value will be increasing very fast in phase 3-5. Because the performance of kidneys to filter waste in blood is decreasing.

Phosphorus level in blood	Values (mEq/L)	Effects
Normal	3.5 – 5.5	-
Low	Less than 3.5	aweary, anorexic, weak muscle
High	Greater than 5.5	itchy skin, hardened arteries, calcium clot catch with tissue, weak bones, enlarged parathyroid glands

***Note: mEq = Milli - Equivalence*

Table 2.3 *Show value of Phosphorus level in blood* [10]

Following the table 2.2, high phosphorus in food can cause bad impacts to CKD patients. There are the formulas, which are formula 2.2, 2.3 and 2.4, to calculate phosphorus value in the foods.

$$\frac{50}{100} \times (Phosphate\ in\ food\ (mg) - [\frac{RDA\ (mg)}{3}])$$

Formula 2.2 *Calculate phosphate in main meal for CKD stage 3*

$$\frac{50}{100} \times (Phosphate\ in\ food\ (mg) - [\frac{80}{100} \times \frac{RDA\ (mg)}{3}])$$

Formula 2.3 *Calculate phosphate in main meal for CKD stage 4-5*

$$\frac{50}{100} \times (Phosphate\ in\ food\ (mg))$$

Formula 2.4 *Calculate phosphate in snack*

2.4 **Medications**

In general, phosphorus will be filtered and excreted from our body via the kidneys like the other waste. If the kidneys were damaged, the kidney function cannot work well and cannot filter all waste out of our body. According to chemical bond, remain phosphorus will catch with calcium. Therefore, the more phosphorus level in bloodstream, the more calcium will be pulled out from our bones. This situation makes our bones become weak and finally brittle.

There are two ways to decrease high phosphorus level in blood that are consuming low phosphorus food and using phosphate binder.

However, the initial of treatment is always controlling phosphorus in food via using three formulas (2.2, 2.3 and 2.4) above to be in a range of RDA (Recommended Dietary Allowance).

To use medication for binding phosphorus in blood, Dr. Anirut Pattarakarn from Faculty of Medicine Siriraj Hospital mentioned that doctors usually use calcium medicine to bind phosphorus. But human body can receive limited calcium level. The maximum calcium level that human can take per day is determined to be not exceeding 2500 mg and under double of RDA for each age range. (see table 2.4)

Age	Need Calcium (RDA)	Max Calcium (RDA*2)
1-3 years old	500 mg	1000 mg
4-8 years old	800 mg	1600 mg
9-18 years old	1300 mg	2600 mg

Table 2.4 *Represent minimum and maximum calcium intake of each age range*

2.5 **Technology**

2.5.1 Flutter Framework

Flutter framework [13] is Google's UI toolkit that uses for developing cross-platform applications on mobile (android and iOS), web, and desktop from a single codebase. Flutter is developed by Dart language. Dart's syntax is clear and

uncomplicated; hence it is suitable for developing the application on any platform. Besides, dart's syntax is not difficult to learn because it is like C++, C#, and Java. [14] Google mentioned that Flutter can provide fast development with beautiful UI and have native performance. In the Flutter framework, many available widgets can work on both android and iOS. Flutter releases the stable version 10.22.3 since 30th October 2020. [13]

Flutter framework can be used through IDE [15] such as visual code, IntelliJ, Android Studio, and XCode by installing a plugin of flutter and dart. However, flutter use dart language to develop an application. Therefore, dart SDK is needed to add in environment path before starting to use flutter.

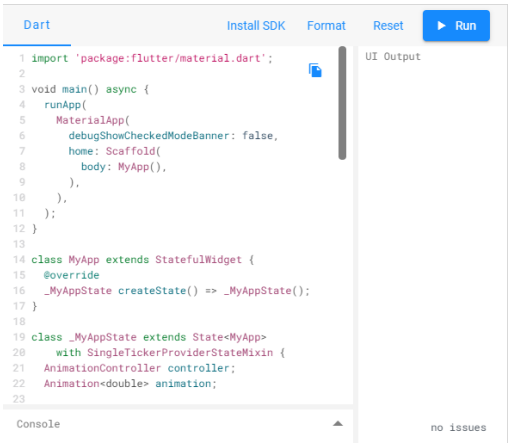


Figure 2.5.1 *image of dart SDK*

2.5.2 Tools - To develop flutter application, there are some IDE tools and the other tools that is important for designing and developing the application.

Android Studio [16] is an IDE (Integrated Development Environment) tool by Google to develop the Android app, based on IntelliJ IDEA. The purpose is to develop an IDE tool that can develop an application on Android more efficiently. Moreover, it can display the result immediately without run the application on the emulator. It also improves the speed of the emulator. Features in the android studio contain visual layout editor to create a complex layout for android application. Developers can preview the layout on any screen size by selecting one of the device configurations.

Currently, Android Studio is still in early access preview phase, but we can download it. To be used on various platforms, almost every OS such as Windows, Mac, and Linux. Android Studio still has the same design and implementation concept

as Eclipse, but it would be easier and easier to write apps on Android. It will have more capabilities and limitations on Eclipse [17].

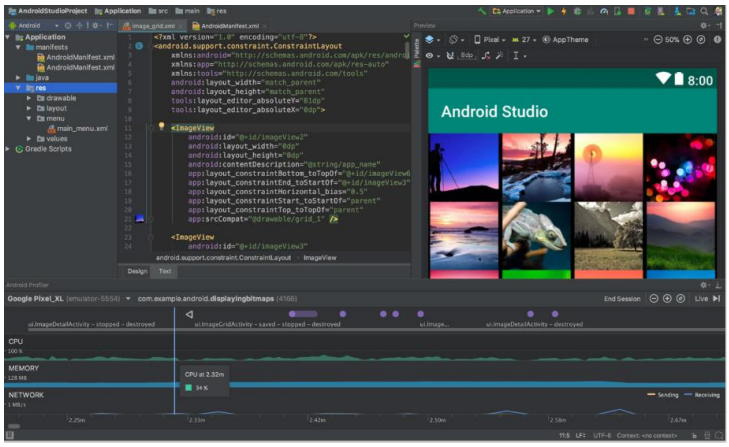


Figure 2.5.2 *image of Android Studios*

However, we can also build a cross platform application (web, iOS, and Android) via using flutter plug-in of Android studio.

IntelliJ IDEA [18] is an IDE (Integrated Development Environment) for Java language that aim to maximize developer productivity. It shortens the routine and repetitive tasks by offering clever code completion, static code analysis, and refactoring, so programmers can focus more on doing productive software development. IntelliJ supports multiple platforms, and it was designed as an IDE for Java languages that contain syntax such as Kotlin, Scala, and Groovy. However, various plugins of IntelliJ can extend an IDE to add a lot of experience. The plugins bundled support some most popular language e.g. Python, Ruby, SQL, PHP, Go, JavaScript, Thyme leaf, JSON, XML, HTML, Markdown, Dart, and the others. IntelliJ IDEA has three editions which are Community, Ultimate, and Edu. Furthermore, in term of usage, IntelliJ have shortcuts for everything while coding. Several features help the users to be convenient such as navigator and search for class, variable or method that was declared in the project's codes, file structure, recent files including location, coding assistance, and other feature.

Adobe XD [19] is a program to create a Prototype for application. It can use both website and mobile. It is very suitable for UX and designer. There are several benefits of Adobe XD. First, this program is to create website and UX/UI that make function and tools are very complete. Second, this program has many templates, and it

has Grid to adjust the size of the display. Third, it has Asset's function to collect components and styles. Fourth, the user can test the application so that make we can get feedback. The last one is Adobe XD has more than 100 plugins to make the application more convenient.

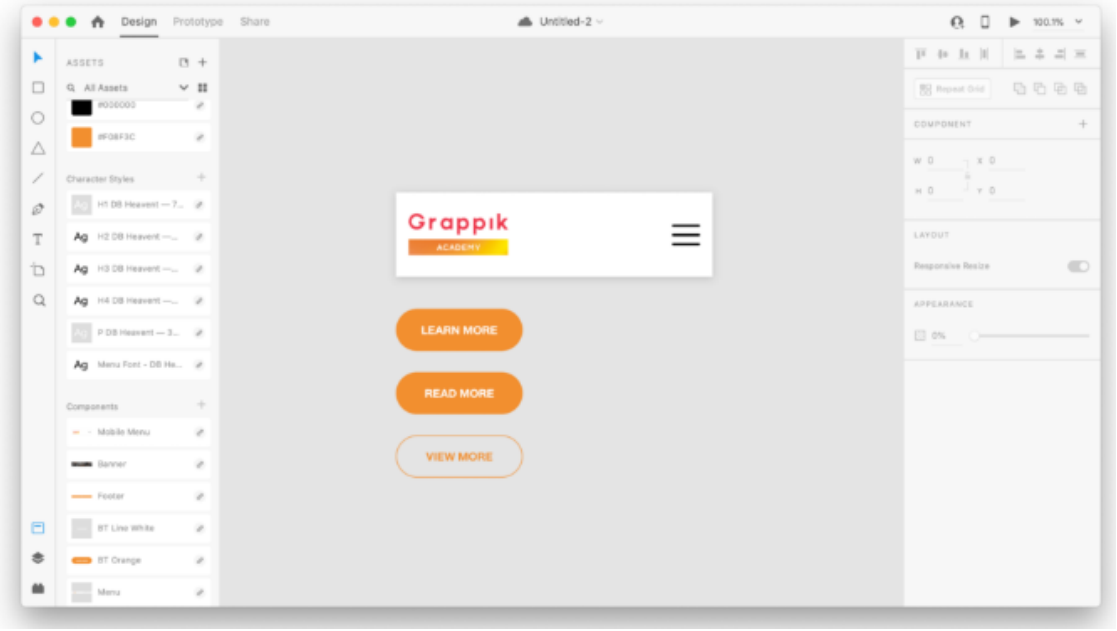


Figure 2.5.3 *image of AdobeXD*

Furthermore, Flutter allows to use native code like Kotlin, Java and Swift together with flutter SDK.

Java [20] is a programming language for general purposes. It can work simultaneously created from class and supports object-oriented programming. Designed to be ready for use. There are many methods and classes to facilitate. Java language is intended to make developers design and develop the program that writes once but can be used everywhere and every platform. The program that written in Java must be able to run on hardware and operating systems supported by the Java Runtime. This is accomplished by compiling the Java code into an intermediate rendering format known as Java bytecode instead of converting it directly to the machine language. Java bytecode commands are like machine language. Rather, it is run by a virtual machine (VM) written for each type of hardware. Typically, users will use the Java Runtime Environment (JRE) installed on their machines to run Standalone Java applications or in a web browser for Java applets. The Java language

also has a standard library in the Development of Graphics threading and networking applications.

HelloWorld.java

```java
// Hello World Program
public class HelloWorld {

    public static void main (String[] args) {
        System.out.println("Hello World!");
    }

}
```

Figure 2.5.4 *image of Java*

Kotlin [21] is one of the most popular developer's languages now. Kotlin is a programming language that developed from Java. The main feature of Kotlin is a 100% replacement for Java and a 40% reduction in coding. Kotlin is free and open source project of apache 2.0 license. It was developed and openly on GitHub by JetBrains team with contribution from Google and the other. Kotlin is used for developing android platform application.

```kotlin
/*
 Create a POJO with getters, setters, `equals()`, `hashCode()`, `toString()` and `copy()` in a single line:
*/

data class Customer(val name: String, val email: String, val company: String)

// Or filter a list using a lambda expression:

val positiveNumbers = list.filter { it > 0 }

// Want a singleton? Create an object:

object ThisIsASingleton {
    val companyName: String = "JetBrains"
}
```

Figure 2.5.5 *image of Kotlin*

Swift [22] is intuitive programming language for macOS, iOS and another Apple OS. Swift includes the combination of the latest programming languages research and decades of experience building Apple platforms. Swift does not need semi-colon at the end line of code. Swift support international languages and emoji. Moreover, string variable are Unicode correct and use a UTF-8 based. The syntax is quite similar to python.

```swift
struct Player {
    var name: String
    var highScore: Int = 0
    var history: [Int] = []

    init(_ name: String) {
        self.name = name
    }
}

var player = Player("Tomas")
```

```swift
extension Player {
    mutating func updateScore(_ newScore: Int) {
        history.append(newScore)
        if highScore < newScore {
            print("\(newScore)! A new high score for \(name)! 🎺")
            highScore = newScore
        }
    }
}

player.updateScore(50)
// Prints "50! A new high score for Tomas! 🎺"
// player.highScore == 50
```

Figure 2.5.6 *image of Swift*

2.6 **Features of the kSmart application**

2.5.1 **Meal log**

The kSmart application will record each meal for the users together with calculating approximate exceed phosphorus level. The application will take a user's information that is CKD phase, age-range, and food type (of the food that users eat in that meal) to calculate exceed phosphorus level. After that, the maximum phosphorus level that the user can each in a day will be deducted by phosphorus value from a meal.

2.5.2 **Personal Profile**

The kSmart application provides a register and login system so that users can have their account and keep their consuming information to help in CKD treatment. The profile will keep personal information like name and birth date, then it will keep information about the CKD phase and medical dosage. This information can help both doctors and patients to track and remind CKD treatment progress.

2.5.3 **User's Roles**

There are two roles in kSmart application which are doctor and patient. For the doctor's account, the application will keep the patient list and meeting schedule. The doctor also can retrieve patient information and their consuming record. For the patient, the application will keep the patient's information, meal record, phosphorus level, and calcium level record, and keep the meeting schedule.

2.5.4 **Recommend menu**

The application will randomly display the recommended menu for CKD patients every day. The menu will be shown on the profile page. Therefore, patients who do not know what to eat can review the menu for deciding.

2.5.5 **Calendar and Note**

The application contains a calendar that will show a meeting schedule that the user can set time to alert (For example, one day before) when meeting time is coming. If the patient touch on a day in the calendar it will show all meals records on that day. Moreover, the application will provide an additional note in each meal, so that user can note about food, drug, or any symptom during the day.

3.1 **Related Works**

This section will discuss the existing applications that are related to kidney dietary food application. To compare the main features of these applications.

2.6.1 **My Healthy Kidney**

The application gives customized nutritional recommendations based on keyed in basic information from doctors, patients and other users to help them make smart, healthy dietary choices in renal nutrition based on calculations and algorithms developed by renal specialists in the form of dietary plans, eliminating the frustration of patients and providing doctors with a reference guide for their patients.

Advantages: There are food's nutrients in detail of each dish. The application use symbols to represent the food that have appropriate phosphorus, almost high phosphorus, and too high phosphorus.

Disadvantages: There is no other features to help users in CKD treatment. The application cannot record each meal and cannot record favorite food.

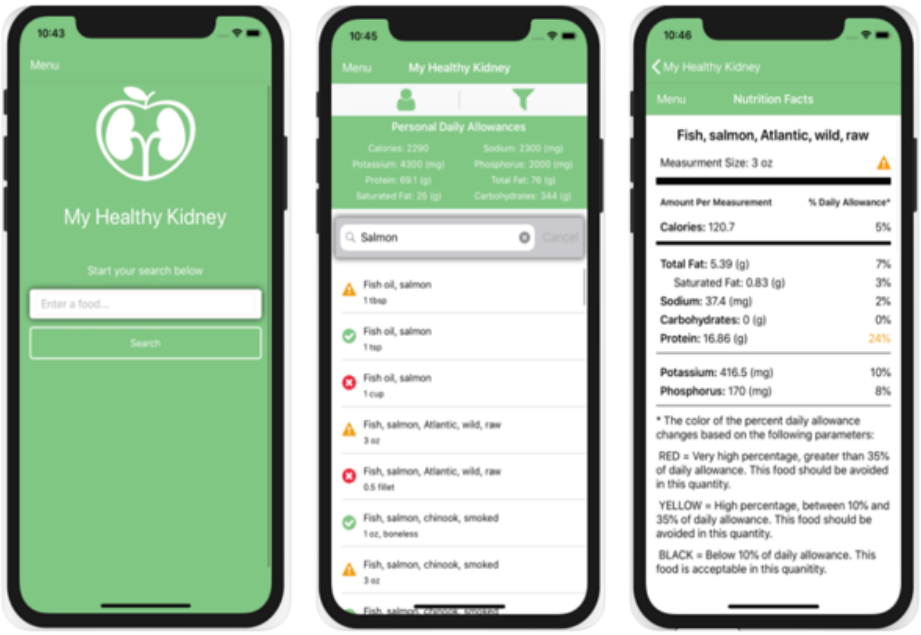


Figure 2.6.1 *images of My Healthy Kidney*

2.6.2 **GFR Calculator: Kidney Health & CKD Stage**

This application help users to calculate estimate Glomerular Filtration Rate (eGFR) in daily practice based on serum creatinine level and other parameters. The result of GFR will determine stage of chronic kidney disease (CKD).

Advantages: There is a feature to calculate eGFR that make users can track their current situation of their disease.

Disadvantages: This application has a feature that appropriate to professional users who know about eGFR and CKD stages. There is no Thai language.

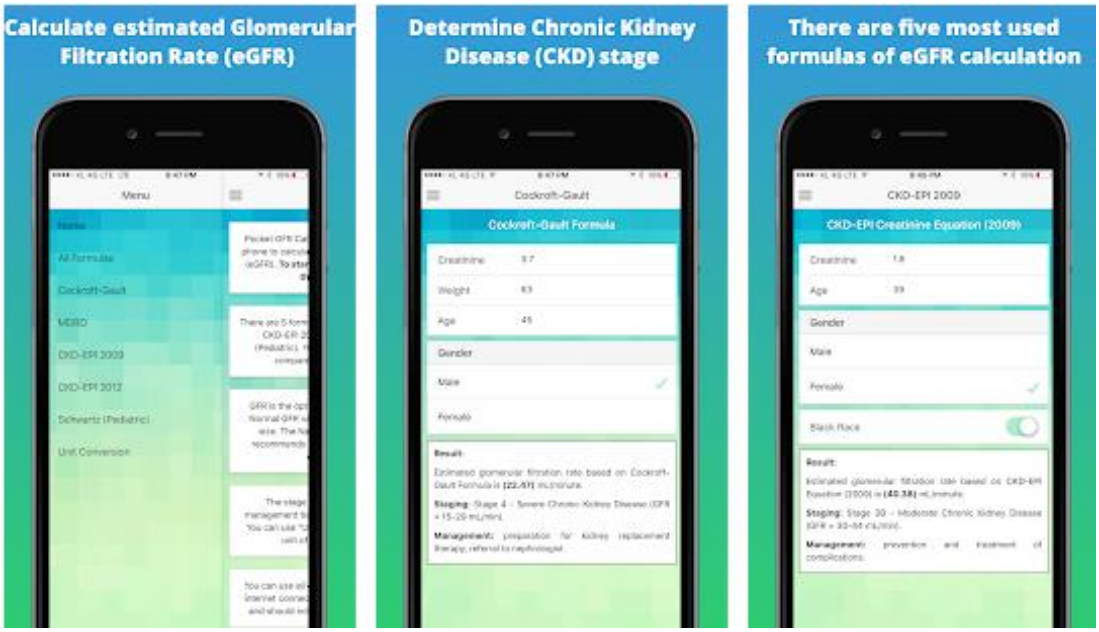


Figure 2.6.2 *images of GFR Calculator*

2.6.3 **CKD Care**

This application was developed by National Kidney Foundation (NKF). It was made for professionals to estimate kidney function via eGFR calculators and give guidelines to take care kidney.

Advantages: CKD Care provides features to calculate eGFR that use to determine stage of CKD, and the application also provide an information and guidelines to take care kidney.

Disadvantages: his application made for the one who is an expert in medical role. Normal user cannot understand the information clearly. There is no Thai language.

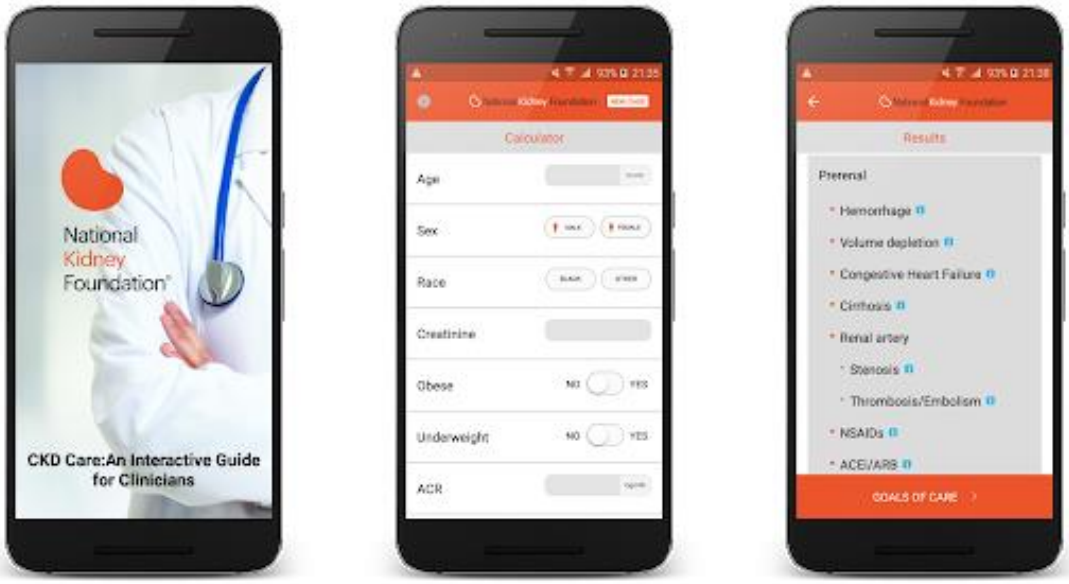


Figure 2.6.3 images of CKD Care

2.6.4 **KAREL - Kidney Health Tracker**

This application was developed by cooperation of kidney patients and medical experts. There are fully personalized and customizable CKD specific application. The application can help CKD patients at any stages track their symptom and get better control of their behavior.

Advantages: This application provides many features that can help CKD patients track their disease closely and make them can talk with the doctor easily. The application was designed from cooperation of patients and medical experts, so the user experiences is good for users more than other similar applications.

Disadvantages: There is no Thai language. There is only one option to register to application (cannot login via Face book or Google) There is no infographic to illustrate information that make some information difficult to understand.

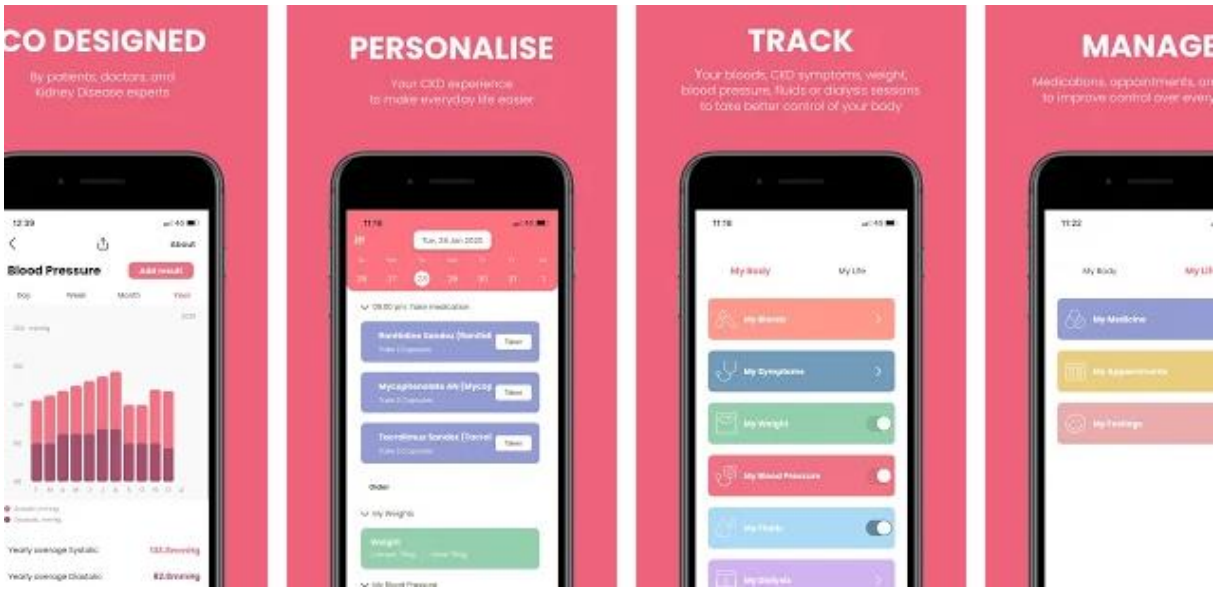


Figure 2.6.4 *images of KAREL*

2.6.5 **MyKidneyDiet - Phosphate Tracker**

This application provides several features and functions to help CKD patients track their situation and control their behavior to cure their disease. In the application, there are exercise guidelines, phosphate binder, phosphate calculator, meals record, blood test result to determine CKD stage and treatment records.

Advantages: There are fully features to help CKD patients track their symptom. The application also provides guidelines and features to help CKD patients control their behavior to treat their kidney.

Disadvantages: There is no Thai language support. There is only one option to register to application (cannot login via Face book or Google). The interface of the application quite difficult to use and font size is not friendly to some people.

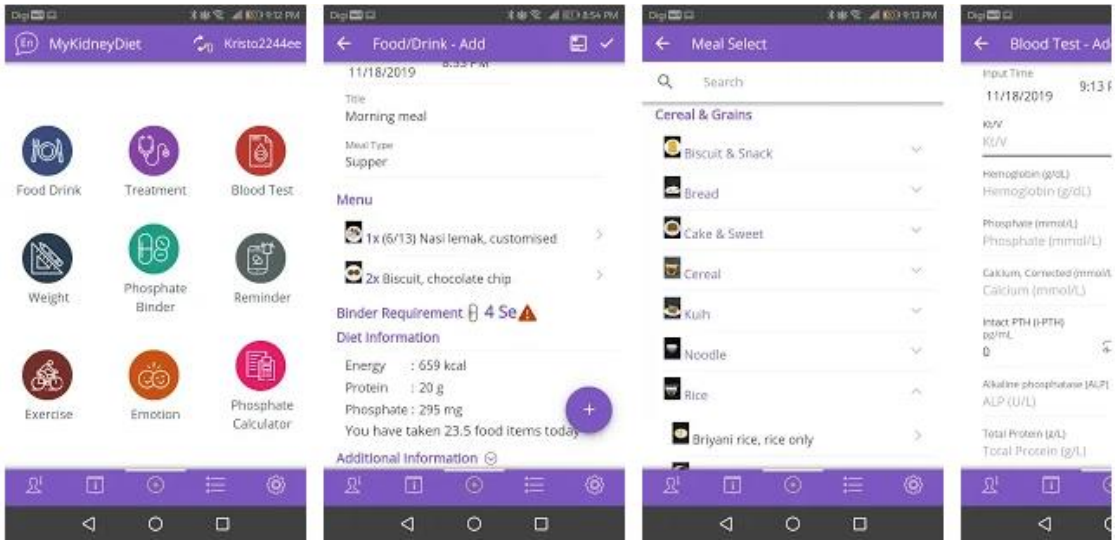


Figure 2.6.5 *images of MyKidneyDiet*

2.6.6 **Comparison Table**

Applications / Features	kSmart	My healthy Kidney	GFR Calculator	CKD Care	KAREL	MyKidneyDiet
Add meal	✓	✗	✗	✗	✗	✓
Personal Profile	✓	✗	✗	✗	✓	✓
Notification	✓	✗	✗	✗	✓	✗
Recommended Menu	✗	✓	✗	✗	✗	✗
Calendar	✓	✗	✗	✗	✓	✗
Favourite (save list)	✓	✗	✗	✗	✗	✗
Add Medicine	✓	✗	✗	✓	✓	✓

In summary, some of these five applications was developed for the medical expert users to calculate CKD stage with eGFR calculator. And some of them provide more usable features for CKD patient to track their diseases closely together with control their behavior. These features can help in CKD treatment. However, each application has some difficulties, and some application might be better if it combines with another application. Consequently, we designed our application ‘kSmart’ to be optimal solution to help CKD patients.

CHAPTER 3
ANALYSIS AND DESIGN

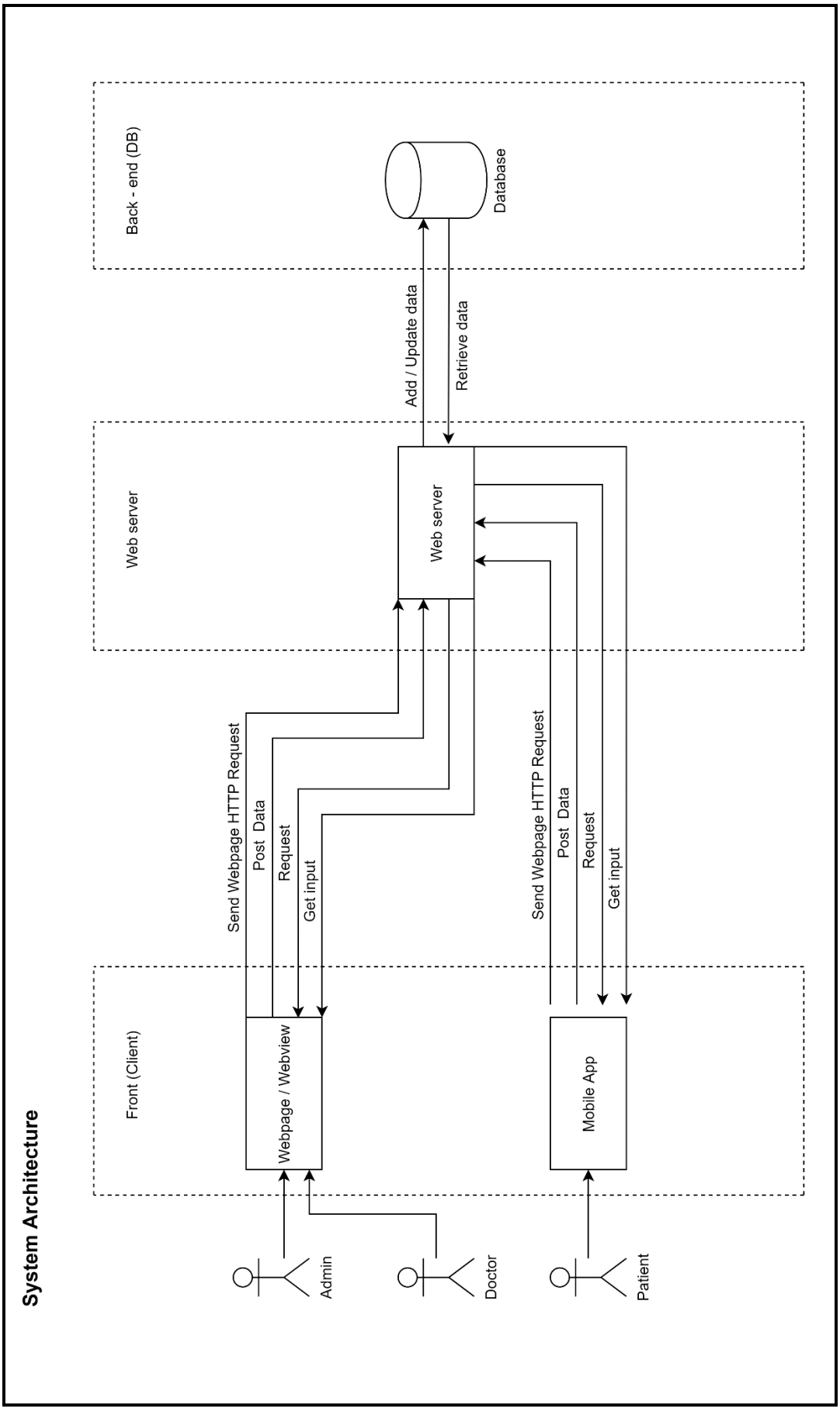
This chapter represents the analysis and design of kSmart application. There is the illustration of physical and logical view of this application including system architecture overview, system structure chart, process analysis and design, database analysis and design, and I/O design respectively

3.1 System Architecture Overview

To figure out accurately the analysis and design of this project, system architecture in Figure 3.1 is developed to demonstrate structure of kSmart application.

The application and system were separated into these following sessions:

- kSmart Application: This is the mobile application that will be available on both Android and iOS platform for the patient. Patient will use this application to record their information, their meals, and their medicine.
- kSmart Dashboard: This is the web application that represent via dashboard for the doctor to see patient's information and record.
- Database: This part will store the information from both mobile and web application.
- Web Server: This part is a bridge between the other parts that are kSmart Back-end, kSmart Application, kSmart Dashboard and Database.

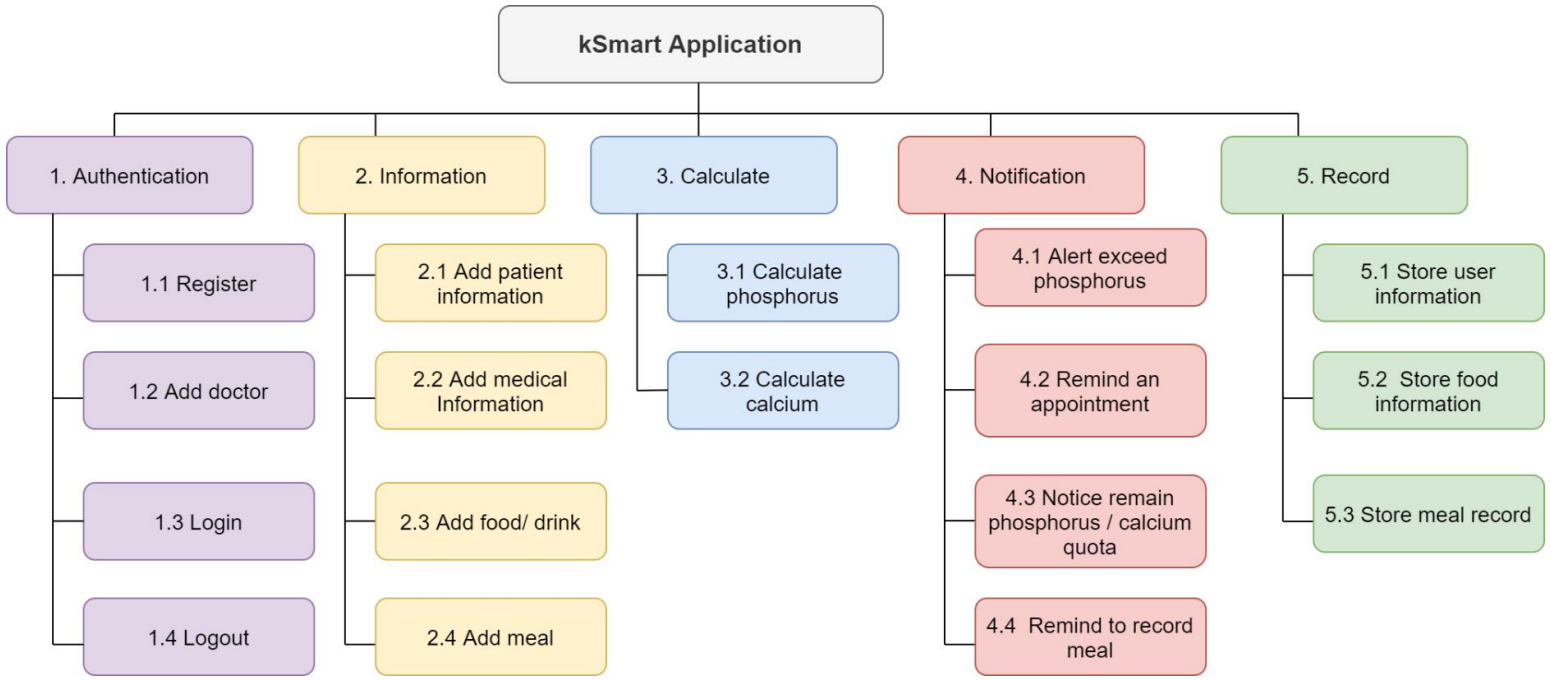

System Architecture
Back - end (DB)
Database
Add / Update data
Retrieve data
Web server
Web server
Send Webpage HTTP Request
Post Data
Request
Get input
Send Webpage HTTP Request
Post Data
Request
Get input
Front (Client)
Webpage / Webview
Mobile App
Admin
Doctor
Patient

Project: kSmart Application **System:** kSmart Application System	**Major Advisor:** Prof. Dr. Pattanasak Mongkolwat
Description: This system architecture demonstrates the structure of the kSmart application, which is built by our team and Siriraj doctors team's cooperation.	

Figure 3.1 kSmart's system architecture

3.2 System Structure Chart

To design kSmart application system, the system's boundary defined for mobile application and web application. There are five sub systems included in (Figure 3.2) that are Authentication, Information management, Calculating, Notification and recording system. The mobile application on Android and iOS provided to support the users who are CKD's patients (or their parents). The web application provided dashboard for doctors to see patient's behavior via their meal record and medicine.



Project: kSmart Application **System:** kSmart Application System	**Major Advisor:** Prof. Dr. Pattanasak Mongkolwat
Description: The structure chart of the kSmart application that refers to patient meal record, medical record, authentication, calculation and notification.	

Figure 3.2 The sub-system structure chart of kSmart application.

The detailed description of each subsystem is shown below:

(Note: In this session, 'the application' means kSmart application.)

1. Authentication – The application required authentication session that mean users must log in before using the application.

    1.1. Register – In case of new patient, the system will receive the required information from the user through mobile application. And the application will send the information to web server to create new account.

    1.2. Add Doctor – In case of registering a new doctor, the system will collect doctor information like email, username, and password from the admin from web application. And the application will send data to the web server to create new doctor account. This process is for the admin to add new doctor like registration. Doctors can reset the password in the first time they log in to the system.

    1.3. Login – When user needs to use the application (both web and mobile), users must log in via using username and password. The system receives them and send to web server. Web server will retrieve data from the database and validate the log in information with the information in database. If it is valid, the application will lead users to their first page.

    1.4. Logout – When users click log out button, system will ask to confirm their action, if they confirm to log out, the application will show the log in page and the system will set the account status as offline.

2. Information store and retrieve – In the application, there are many types of information that will be stored and retrieved.

    2.1. Add/ Update patient information – When patients register in the application their information will be recorded in the database. And users can also update their information later. Therefore, the application will receive the updated information and send to web server that will connect and update the information in the database.

2.2. Add medical information – The application provides feature to add medicine for patients. The application will receive the information of medicine like name and usage. Then record them into the system and set the notification to remind patient when the time is to take medicine.

2.3. Add food/ drink – Initially, we will add food and drink table with default quantity into the system. And the users can also add the new foods or drinks later if they are not existing.

2.4. Add meal – The application provides feature to record each meal that patient consumed. The application will get the list of food and drink that patient selected and use the portion and phosphorus table to calculate amount of phosphorus in that meal. After calculating, the system will store that meal with amount of phosphorus into database.

3. Calculation – The application must calculate amount of phosphorus in the food that patient selected in each meal. Besides, it must calculate the remain of phosphorus and calcium value that patient can receive in each day.

3.1. Calculate phosphorus – The application receives the food and drink that contain ingredient and portion. Then it retrieves phosphorus value from the table that developers collect from the nutritionists to calculate the amount of phosphate in the meal. Finally, the application deducts a phosphorus value of the meal from daily phosphorus allowance of a patient.

3.2. Calculate calcium – The application receives the medicine usage information of the patient and decrease the maximum calcium that patient can take by the amount of calcium in medicine.

4. Notification – The application must remind or alert the users for any activities.

4.1. Alerts exceed phosphorus – If the application check that the amount of phosphorus that patient take exceed the limited, it will alert the user to take medicine to bind phosphorus or stop eating more phosphorus.

4.2. Remind appointment – The application let users both patient and doctor set their appointment on the calendar in the application. Therefore, the application will remind them when time of appointment is coming.

4.3. Notice remains quota of phosphorus – The application will display the chart and number of remaining phosphorus value that patient can take on status bar.

4.4. Remind to record meal – The application will remind patient to record each meal by the default time of each meal. For example, the application will remind patient to record Breakfast at 7.00 am. However, patients can customize the time of their meal too.

5. Record – There is numerous amounts of information will be stored into database.

5.1. Store user information – After registering, the system will record user's information into database. When the users update their information on the application, the system will take them to update in the database too.

5.2. Store food information – The list of food and drink with default ingredient will be record in food table in the database. There is another ingredient table to store list of ingredients too. When the user adds new food, drink or ingredient, the system will take them and update in the database.

5.3. Store meal record – After the application get foods and drinks that user consumed and calculate the amount of phosphorus from the foods consumed, this information (foods, drinks, and phosphorus value) will be store in meal table in the database.

Process Analysis and Description

3.3 Data Flow Diagram (DFD)

3.3.1 DFD Level 0

DFD level 0 or the context level diagram of kSmart application illustrates the overview of the system. It includes the four external entities, which are patient, doctor, admin and anonymous user, send and receive data flow between four entities and the system (see in figure 3.3).

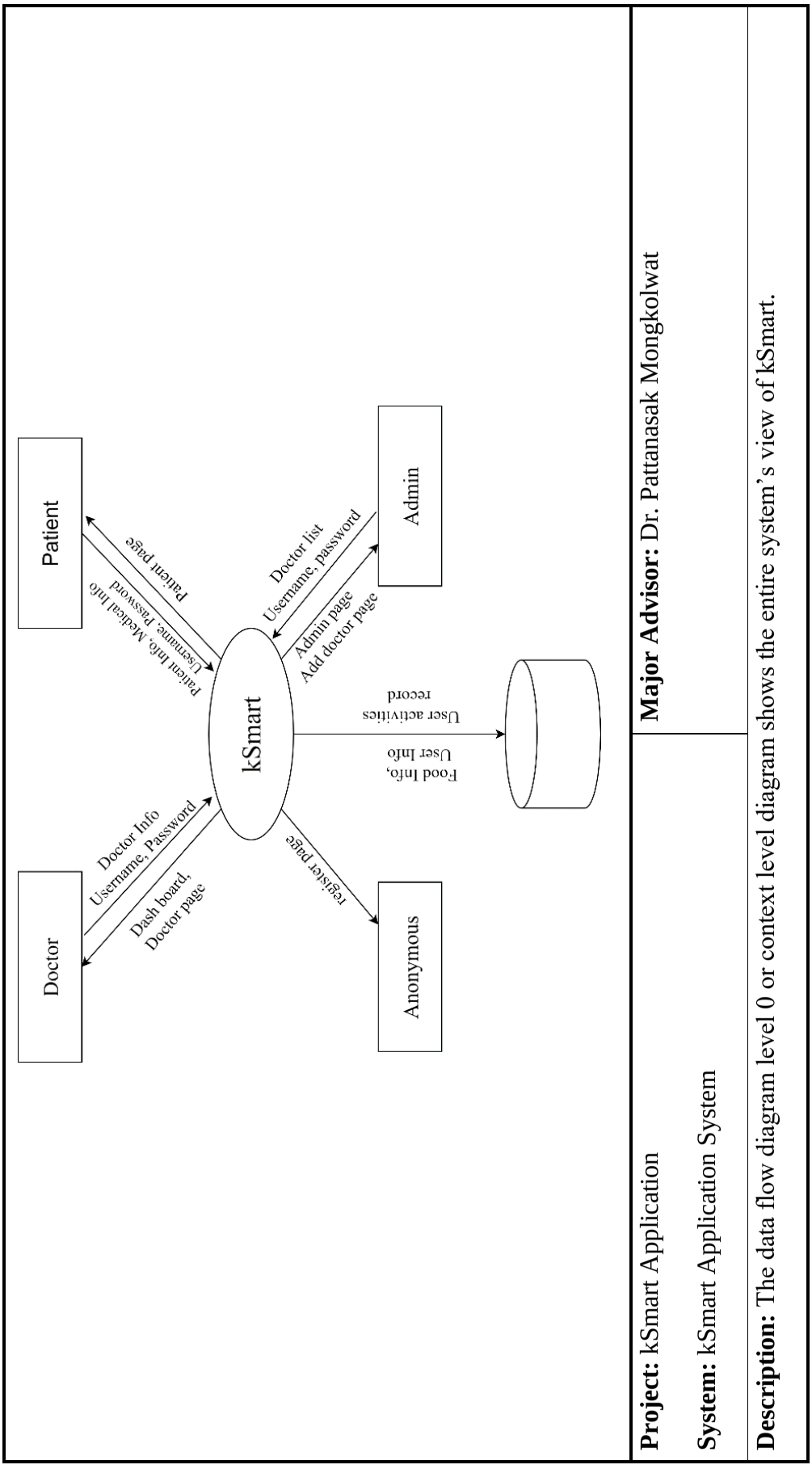
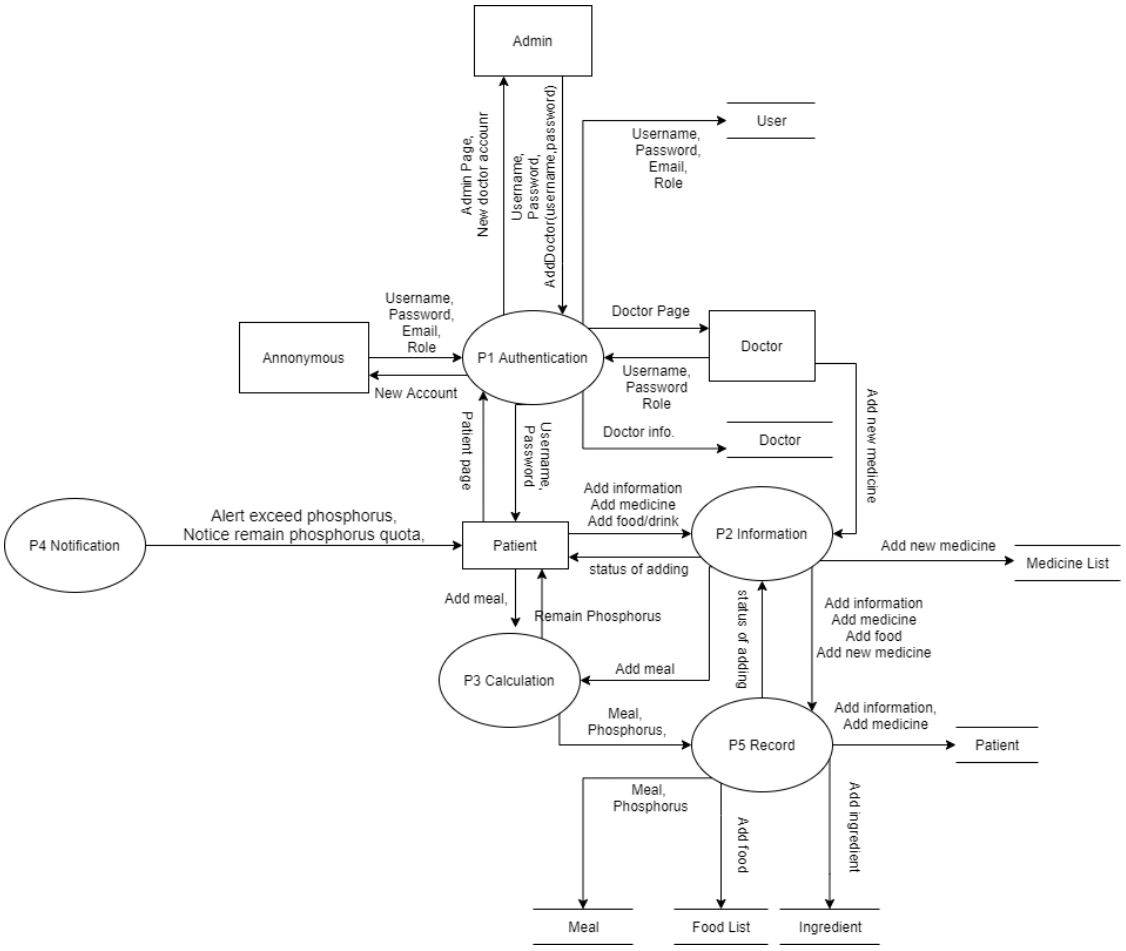


Figure 3.3: Data Flow Diagram Level 0

3.3.2 DFD Level 1

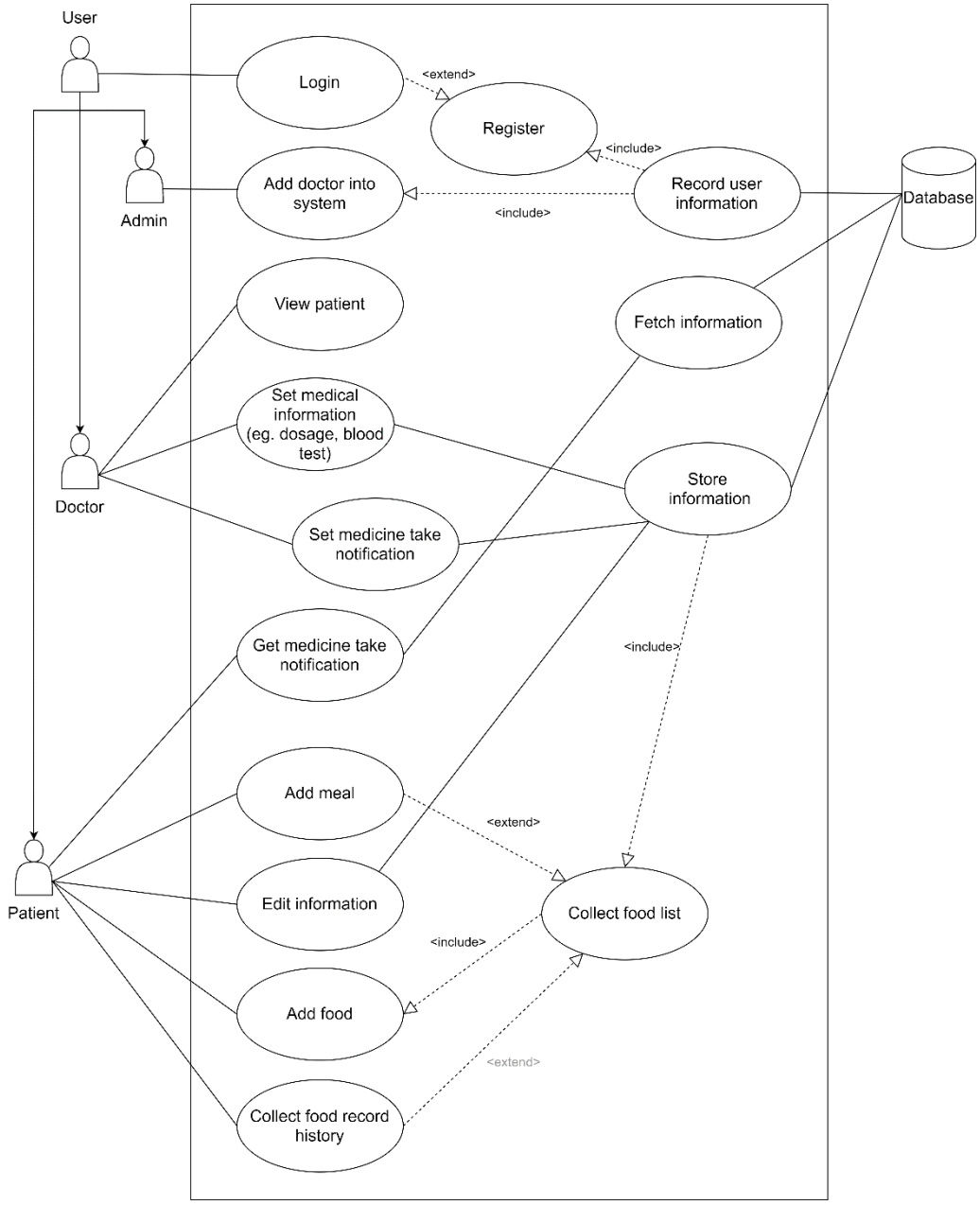
DFD Level 1 displays a minuscule portion extending from the context level diagram. This level comprises the five processes of kSmart including Authentication, Information, Calculation, Notification and Record. Besides, this level performs the data stores that use as databases to store both mobile and web application information.



Project: kSmart Application **System:** kSmart Application System	**Major Advisor:** Dr. Pattanasak Mongkolwat
Description: The data flow diagram level one of kSmart divided into five primary processes. And show how the data is recorded in data storage.	

Figure 3.4: Data Flow Diagram Level 1

3.4 Use case diagram



Project: kSmart Application **System:** kSmart Application System	**Major Advisor:** Dr. Pattanasak Mongkolwat
Description: The use case diagram of kSmart application represents the overall environmental illustration that both mobile and web application have been delivered to users.	

Figure 3.5: Use case diagram

3.5 Data Dictionary

A data dictionary is a way to document and describe Processes, Data Stores, and Data Elements (Data Flow) that occur in a Data Flow Diagram (DFD). It is composed of 3 parts as shown below.

- Process Descriptions
- Data Stores
- Data Elements

3.5.1 Process Description

This section will provide the detailed description of each process that exists in this system. It includes Inbound Data, Outbound Data, and Logic Summary.

Table 3.1: List of all Processes

No.	Process	Name	Description
1	P1	Authentication	This process will authorize users who can access to use the application.
2	P1.1	Register	This process is for a patient who never use kSmart application. Users must register before using the application.
3	P1.2	Add doctor	This process is for admin to add a new doctor into the system.
4	P1.3	Login	This process is for all users (patient, doctor, and admin) to log in their account to access the application.
5	P1.4	Logout	This process is for the user who need to sign out from the application.

No.	Process	Name	Description
6	P1.5	Reset Password	This process is for doctors to change password from the one that admin registered for them in the first time that they access the system. And for users who need to change password for some
7	P2	Information	This process is for adding or updating the information in the application.
8	P2.1	Add patient information	This process is for patients to add their personal information.
9	P2.2	Add medicine	This process is for doctor and admin to add new medicine into the system. And for doctor to add medicine list (that patient has to take) for their patients.
10	P2.3	Add food/drink	This process is for users to add a new food, drink or desert that is not existed in the list.
11	P2.4	Add meal	This process is for patients to selected food, drink, or snack that they eat in each meal and record it in the application.
11	P3	Calculate	This process is for phosphorus and calcium calculation.
12	P3.1	Calculate phosphorus	This process will take the selected food, drink, or snack from add meal to calculate amount of phosphorus value and record it with that meal.
13	P4	Notification	This process includes all notifications in the application.

No.	Process	Name	Description
14	P4.1	Alerts exceed phosphorus	This process will be alert the patient when amount of phosphorus that they take in that day exceed the maximum value.
15	P4.2	Notice remains phosphorus quota	This process will display the remain phosphorus value that patient can take in current day.
16	P5	Record	This process shows how the data will be recorded in the storages.
17	P5.1	Store user information	This process is for storing user information like log in information, doctor information, and patient information.
18	P5.2	Store food information	This process is for storing the food, drink, and desert list.
19	P5.3	Store meal record	This process is for storing each meal that patient record with the amount of phosphorus value.
20	P5.2	Store food list	This process is for storing the food, drink, and desert list.
21	P5.3	Store meal record	This process is for storing each meal that patient record with the amount of phosphorus value.
22	P5.4	Store medicine list	This process is to store the list of medicine for the patient. The list of medicine will be taken from pharmacist.
23	P5.5	Store medicine taken	This process is for the doctor to set a list and schedule for their patient to take medicine(s).

Table 3.2: Process Description of Authentication (Process 1)

Process Name	P1- Authentication
Description	This process will authorize the users who can access to use the application.
Inbound data	• Username • Password • Email • Role
Outbound Data	• New account according to role. • Message to alert invalid input. • Message to confirm success status. • Account page according to role.
Logic Summary	P1.1 Register While: the patient goes to register in the application, the application will ask for email, username, and password. If email is duplicated, alert message will appear. If password length less than minimum, alert message will be appeared. If user fill invalid password in repeating password field, alert message will be appeared. Then: Patient will be asked to fill personal information and medical information. Some field patient can skip. If there is invalid input, alert message will be appeared. Else: End while P1.2 Add doctor If user = admin: let user access add doctor page Else: block

	While admin adding new doctor: Admin will be asked to fill doctor's username, password, email. If email is duplicated, alert message will appear. If password length less than minimum, alert message will be appeared. If user fill invalid password in repeating password field, alert message will be appeared. Then: system will ask for real name of doctor. If there is invalid input, alert message will be appeared. Else: End while P1.3 Login Users give username and password. If there is no invalid input, then user can access the application. Else: alert message will be appeared. P1.4 Logout Users send logout request. Application asks to confirm user's action. Log out from the application

Table 3.3: Process Description of Information (Process 2)

Process Name	P2- Information
Description	This process is for adding or updating the information in the application.
Inbound data	• Add new food/drink/desert • Add user information • Add meal • Add medical information

Outbound Data	• Message to alert success or unsuccess status
Logic Summary	P2.1 Add user information In the registration process or When user need to update information. Application will ask user to fill the information in each field. If there is invalid data: alert message appears. Else: information will be stored and end task. P2.3 Add new food/drink/desert User will be asked to type name and take a photo of food. Then application will ask user to confirm ingredients. If there is invalid data: message alert appears. Else: food item will be stored and end task. P2.2 Add medical information Case 1: If the user is patient: Patient will be asked to select the medicine that patient use with the usage (like quantity and time to use). If the medicine does not exist, patient can add new medicine (then the doctor can review the new medicine that patient add) Case 2: If the user is doctor: Doctor will select the patient from list and review or add the medicine to that patient account. If the medicine does not exist, doctor can add new medicine. If the patient did not add medicine yet, doctor can do that for patient. (In case that patient does not know their medicine) Then:

	Medicine information will be stored End task P2.4 Add meal Patient will select date, time, and type of meal. Patient will select food(s) and/or drink(s) and/or desert(s). Patient will check ingredients and quantity of ingredients in each food. If ingredient and/or quantity is invalid: Patient can customize it. The system will show amount of phosphorus of that meal. Then patient can select to add that meal in save list (in case that patient like to eat that menu). Then patient will click to save that meal. End task

Table 3.4: Process Description of Calculation (Process 3)

Process Name	P3- Calculation
Description	This process is for phosphorus and calcium calculation.
Inbound data	• Food list in meal • Medicine taking
Outbound Data	• Phosphorus value in food • Remain phosphorus that can intake • Remain calcium that can intake
Logic Summary	P3.1 Calculate Phosphorus The system gets selected food on a patient. Calculate amount of phosphorus in that meal. Display on meal page and record with that meal. Decrease phosphorus from maximum phosphorus in status bar. The phosphorus value has to reset when starting a new

	date.

Table 3.5: Process Description of Notification (Process 4)

Process Name	P4- Notification
Description	This process includes all notifications in the application.
Inbound data	• Schedule for notification • Content/ medicine list for each patient
Outbound Data	• Medicine take notification (remind patient to take medicine)
Logic Summary	P4.1 Alerts exceed phosphorus – The system alerts patients when they save meal that make total phosphorus value exceed their limitation. P4.3 Notice remains phosphorus quota - The system calculates total phosphorus from all of meal that patient take each day and deduct from the phosphorus quota of that patient. Then show the remaining phosphorus value that patient can take with pie chart.

Table 3.6: Process Description of Record (Process 5)

Process Name	P5- Record
Description	This process shows how the data will be recorded in the storage.
Inbound data	• Data/ Information from user.
Outbound Data	• Generated GUID.
Logic Summary	P5.1 Store user information Receive information of user like sign in information or medical information from client side and send to record in MongoDB database via spring boot server and http protocol. P5.2 Store food information Receive food information (name, ingredients, image, phosphorus value, category) Record in MongoDB database. P5.3 Store meal record Receive meal information. Receive patient ID. Record meal information in MongoDB by mapping meal with patient ID. P5.4 Store medicine list Receive medicine information (name, quantity, unit, image). Record them in MongoDB database.

	P5.5 Store medicine taken Receive medicine taken information like which medicine, which time, how much. Receive patient id. Store them in MongoDB by mapping medicine taken record with patient ID.

3.5.2 Data Stores

This section describes the data stores that exist in the data flow diagram and consists of the Data Store Name, Description, Inbound Data, and Outbound Data.

Table 3.7: List of all Data Stores

No.	Data Store	Name	Description
1	D1	User	Store username, password, email, and role.
2	D2	Patient	Store patient's information and medical information of the patient.
6	D3	Food	Store food list.
7	D4	Meal	Store meal record.
9	D5	Medicine	Store list of medicine.
	D6	MedicineTaken	Store list of medicine that patient must take
	D7	Image	Store images of foods, ingredients and medicine that will map with ID of them.
10	D8	Ingredient	Store list of ingredients.
11	D9	Role	Store roles of users.

Table 3.8: Data Store Description of User (D1)

Data Store Name	D1 - User
Description	Store username, password, email, and role.
Inbound data	• New account • username, password, email, and role
Outbound Data	• kSmart Account • Log in information

Table 3.9: Data Store Description of Patient (D2)

Data Store Name	D2 - Patient
Description	Store patient's information and medical information of the patient.
Inbound data	• Patient information • Medical information (CKD phase, GFR) • Medicine usage
Outbound Data	• Patient profile • Medical profile • Medicine schedule

Table 3.10: Data Store Description of Food (D3)

Data Store Name	D3 - Food
Description	Store food list.
Inbound data	• Food/drink/desert name • ingredients
Outbound Data	• Food list

Table 3.11: Data Store Description of Meal (D4)

Data Store Name	D4 - Meal
Description	Store meal record.
Inbound data	• Food/drink/desert list

	• Date, time
Outbound Data	• Meal record • Phosphate value

Table 3.12: Data Store Description of Medicine (D5)

Data Store Name	D5 - Medicine
Description	Store list of medicine.
Inbound data	• Medicine list (that will use to cure CKD patient) • Unit of medicine
Outbound Data	• List of medicine

Table 3.13: Data Store Description of MedicineTaken (D6)

Data Store Name	D6 - MedicineTaken
Description	Store medicine list that each patient must take.
Inbound data	• Medicine list of each patient • Schedule to take each medicine.
Outbound Data	• List of MedicineTaken

Table 3.14: Data Store Description of Image (D7)

Data Store Name	D7 - Image
Description	Store images of foods, ingredients and medicines.
Inbound data	• Image (binary data type)
Outbound Data	• List of images

Table 3.15: Data Store Description of Ingredient (D8)

Data Store Name	D8 - Ingredient
Description	Store list of ingredients.
Inbound data	• Ingredient information • Phosphorus value
Outbound Data	• List of ingredients

Table 3.16: Data Store Description of Role (D9)

Data Store Name	D9 - Role
Description	Store roles of users.
Inbound data	• Role information
Outbound Data	• List of roles

3.5.3 Data Element

This section describes the data elements or data flows that exist in this system. The table below contains the list of all data elements belonging to their data element name, starting process/source/data store, and ending process/source/data store.

Table 3.17: List of All Data Elements

SEQ	Data Element Name	From Process/Source/Data Store	To Process/Source/Data Store
1	Username	Anonymous, Patient, Doctor, Admin	Authentication (P1)
2	Password	Anonymous, Patient, Doctor, Admin	Authentication (P1)
3	Email	Anonymous, Patient, Doctor, Admin	Authentication (P1)
4	Role	Anonymous, Patient, Doctor, Admin	Authentication (P1)
5	Admin Page	Authentication (P1)	Admin
6	Doctor Page	Authentication (P1)	Doctor
7	Patient Page	Authentication (P1)	Patient
8	New Account	Authentication (P1)	Patient
9	New Doctor Account	Authentication (P1)	Doctor
10	Doctor info.	Authentication (P1)	Doctor (D3)
11	Add information	Patient, Information (P2), Record (P5)	Information (P2), Record (P5), Patient (D2)
12	Add medicine	Patient, Information (P2)	Information (P2), Patient (D2)

SEQ	Data Element Name	From Process/Source/Data Store	To Process/Source/Data Store
13	Add food/drink	Patient, Information (P2), Record (P5)	Information (P2), Record (P5), Food (D6)
14	Alerts exceed phosphorus	Notification (P4)	Patient
15	Remind an appointment	Notification (P4)	Patient
16	Notice remains phosphorus- calcium quota	Notification (P4)	Patient
17	Remind to record meal	Notification (P4)	Patient
18	Status of adding	Record (P5), Information (P4)	Information (P4), Patient
19	Add meal	Patient	Calculation (P3)
20	Add new medicine	Doctor, Information (P2)	Information (P2), Medicine (D9)
21	Add medicine taking	Patient	Calculation (P3)
22	Meal	Calculation (P3), Record (P5)	Record (P5), Meal (D7)
23	Phosphorus	Calculation (P3), Record (P5)	Record (P5), Meal (D7)
24	Add ingredient	Record (P5)	Ingredient (D10)
25	Remain Phosphorus	Calculation (P3)	Patient

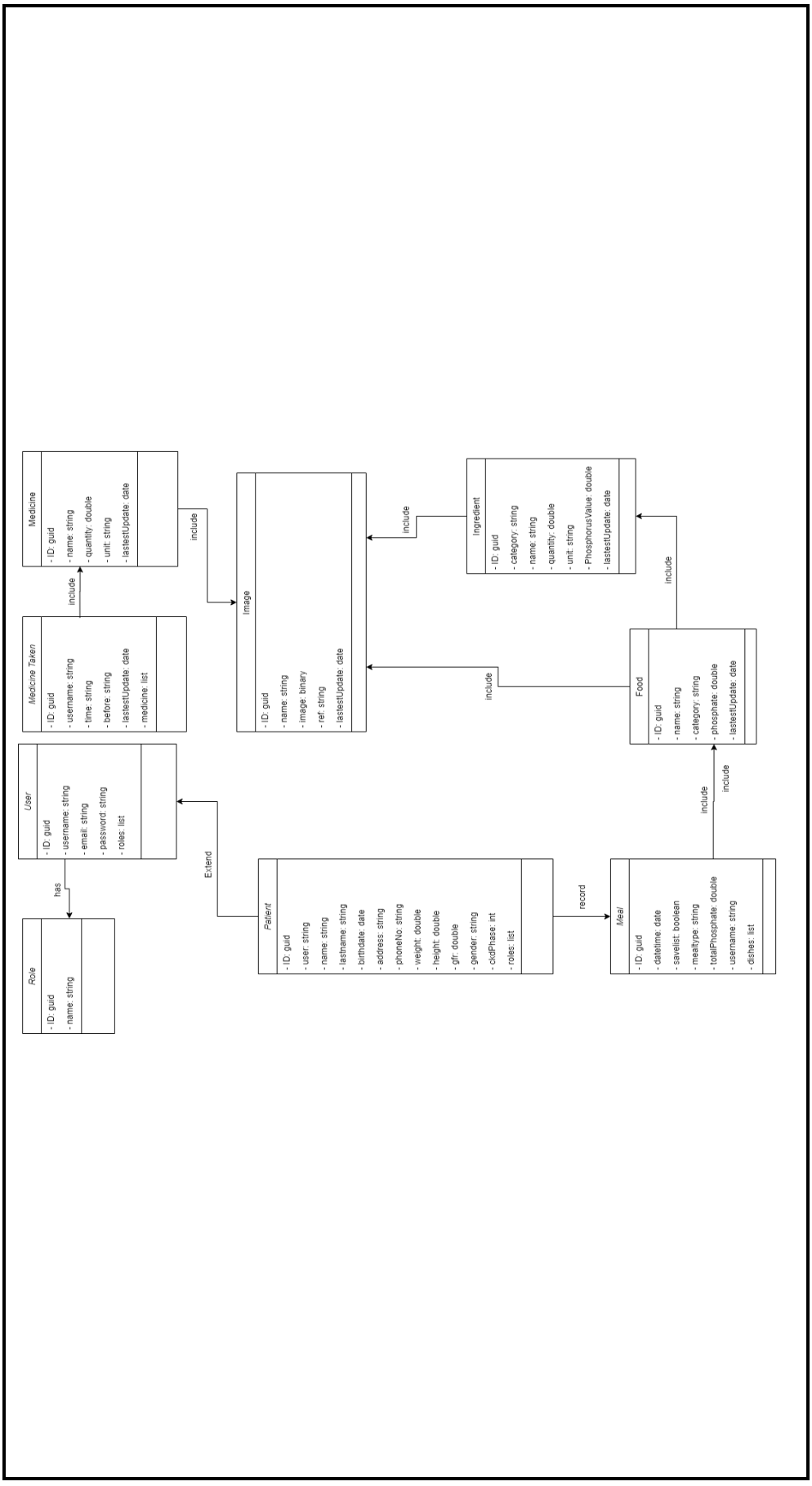
3.6 **Database Analysis and Design**

Database design is the significant part of software design that is called “core of the system” for the most system. The design of database system can determine efficiency of data access. In general, developers use ER diagram (Entry-Relationship Diagram) to design and demonstrate the relationship between the data structure and data tables in the system. Data tables are also called entities.

In this system, there are many relationships and entities. According to requirement analysis and system organization chart, the system contains 11 entities. The following list consists of all entities and the explanation:

- User: contains username, password and email that use in log in process.
- Admin: contains admin information.
- Patient: contains patient information and medical information.
- Doctor: contains doctor information.
- Address: contains patient’s address and phone number.
- Food: contains food/drink/desert list.
- Meal: contains meal (with phosphorus value) record of patient.
- Appointment: contains appointment list.
- Medicine: contains medicine list
- Ingredient: contains ingredient list.
- Role: contains roles of user.
- Category: Contains category list.

3.7 UML Diagram



Project: kSmart Application **System:** kSmart Application System	**Major Advisor:** Dr. Pattanasak Mongkolwat
Description: UML diagram shows all entities in the database of kSmart application system and how they are related.	

Figure 3.6: 6 UML Diagram of kSmart Application

UML Diagram show how the classes in spring boot server are related. It is the same way as how the collections in MongoDB related. This diagram help us when implement the sever to map the data between collection easily. Also it help us have more understanding about the information that we store in database.

3.8 **Relational Schema**

This section describes the attributes of the tables in the database. The attribute notation is shown below.

- **<u>Attributes</u>** – which are bold and underlined are the Primary Keys
- *Attributes* – which are Italic are the Foreign Keys
- ***<u>Attributes</u>*** – which are bold, italic and underlined are both Primary Keys and Foreign Keys

Tables in this system can be divided into 3 groups as follows:

- Master File Table
- Base File Table
- Transaction File Table

Table 3.18: List of all Tables in Our System Database

Table#	Table Name	Table Type	Description
T1	User	Master File	contains username, password and email that uses in log in process.
T2	Patient	Master File	contains patient information and medical information.
T3	Medicine Taken	Transaction File	contains medicine list for each patient.
T4	Image	Transaction File	contains images of foods, ingredients, and medicines.
T5	Food	Transaction File	contains food/drink/desert list.
T6	Meal	Transaction File	contains meal (with phosphorus value) record of patient.
T7	Ingredient	Transaction File	contains ingredient list.
T8	Medicine	Transaction File	contains medicine list.
T9	Role	Master File	contains roles of user.

1. Schema of Master File Tables

User

id	username	password	email	*ObjectID(Role)*	

Patient

patientID	*username*	name	lastname	
birthdate	weight	height	gender	ckdPhase
GFR				

Role

roleID	roleName			

2. Schema of Transaction File Tables

Food

foodID	*ingredientID*	quantity	foodName
type	phosphorus	foodImg	

Meal

mealID	date	time	*List(foodID)*	mealType
totalPhosphorus	isSaveList			

Ingredient

ingredientID	ingredientName	intgredientImg		

Medicine

medID	medName	medMg		

MedicineTaken

medTakenID	medName	medMg	medImg	*List(medID)*

Image

imageID	imgName	*ref*	image	

3.9 **Database Tables (Schemeless)**

This section shows the details of each file component including field name, field description, field data type, field length, null value, primary key and foreign key.

Table 3.19: File Structure of User

Table Name:	User		**Table#T1**			
Table Type:	Master File					
Description:	Store username, password and email that use in log in process.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
username	VARCHAR	20	Username/display name of the user.	PK		
password	VARCHAR	20	Password to log in to the application.			
email	VARCHAR	50	Email of the user.		Should contain '@'	
	Total	90	Bytes			

Table 3.20: File Structure of Patient

Table Name:	Patient		**Table#T2**			
Table Type:	Master File					
Description:	Store patient information.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
username	VARCHAR	20	Username/display name of the user.	FK		

patientID	INT	10	Admin ID.	PK		
address	VARCHAR	100	Patient's address.			
name	VARCHAR	50	Name of patient.			
lastname	VARCHAR	50	Last name of patient.			
birthdate	DATE	-	Birthdate of patient.			
weight	FLOAT	3	Weight of patient.			
height	FLOAT	3	Height of patient.			
Gender	VARCHAR	10	Gender of patient.			
ckdPhase	INT	5	CKD stage of patient.			
GFR	FLOAT	3	GFR value of patient.			
	Total	254	Bytes			

Table 3.21: File Structure of Role

Table Name:	Role	**Table#T3**
Table Type:	Master File	
Description:	Store role of user.	

Field Name	Type	Length	Description	Key	Reference	Null
roleID	INT	10	Role ID	PK		
name	VARCHAR	10	Name of the role.			
Total	20	Bytes				

Table 3.22: File Structure of MedicineTaken

Table Name:	MedicineTaken	Table#T4				
Table Type:	Transaction File					
Description:	Store medicine list for each patient.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
medTakenID	VARCHAR	20	ID of medicine taken record	PK		
List(medID)	ARRAY	20	Array of medicine ID	FK		
date	DATE	-	Date of Appointment			
time	DATETIME	-	Time of Appointment			
	Total	40	Bytes	Total	40	

Table 3.23: File Structure of Image

Table Name:	Image		**Table#T5**			
Table Type:	Transaction File					
Description:	Store images.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
roleID	INT	10	Role ID	PK		
name	VARCHAR	10	Name of the role.			
imageID	VARCHAR	20	ID of an image	PK		
ref	VARCHAR	20	ID of food, ingredient or medicine	FK		
	Total	60	Bytes			

Table 3.24: File Structure of Food

Table Name:	**Food**		**Table#T6**			
Table Type:	Transaction File					
Description:	Store Food.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**

foodID	VARCHAR	20	Food ID	PK		
ingredientID	VARCHAR	20	Ingredient ID.	FK		
quantity	FLOAT	-	Quantity of ingredient			
foodName	VARCHAR	50	Food name			
categoty	VARCHAR	20	Food category			
phosphorus	FLOAT	-	Amount of phosphorus in food.			
foodImg	IMAGE	-	Image of food			
	Total	90	Bytes			

Table 3.25: File Structure of Meal

Table Name:	**Meal**		**Table#T7**			
Table Type:	Transaction File					
Description:	Store Meal.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
mealID	VARCHAR	20	Meal ID	PK		
List(ingredientID)	ARRAY	-	List of Ingredient ID.			

mealtype	VARCHAR	10	Meal type (Breakfast, Lunch, Dinner, Snack)			
date	DATE	-	Date			
time	DATETIME	20	Time			
totalPhosphorus	FLOAT	-	Amount of phosphorus in each meal.			
	Total	50	Bytes			

Table 3.26: File Structure of Ingredient

Table Name:	**Ingredient**		**Table#T8**			
Table Type:	Transaction File					
Description:	Store Ingredient.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
ingredientID	VARCHAR	20	ingredient ID	PK		
name	VARCHAR	10	Ingredient Name			
quantity	FLOAT	-	Amount of ingredient			
unit	VARCHAR	20	Unit of ingredient			
phosphorus	FLOAT	-	Amount of phosphorus in each meal.			

Total	50	Bytes

Table 3.27: File Structure of Medicine

Table Name:	**Medicine**		**Table#T9**			
Table Type:	Transaction File					
Description:	Store medicine.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
medicinetID	VARCHAR	20	Medicine ID	PK		
name	VARCHAR	10	Medicine Name			
quantity	FLOAT	-	Quantity of medicine			
	Total	30	Bytes			

3.6 I/O Design

This section explains the design of the Input and Output User Interface. The section consists of two parts, the interface design and the transition diagram showing transition through the system.

A.1.1 3.7 Interface Design

This project has 2 platform design including mobile application design, and website application design. First, mobile application was designed for patients. Second, website application was designed for hospital officers to use in conjunction with the mobile application.

3.7.1 Mobile application design

Authentication system

This system consists of 2 pages, which are a login page and register page. Users need to login. If the user does not have, they must register first or use another access way to login such as Google login.

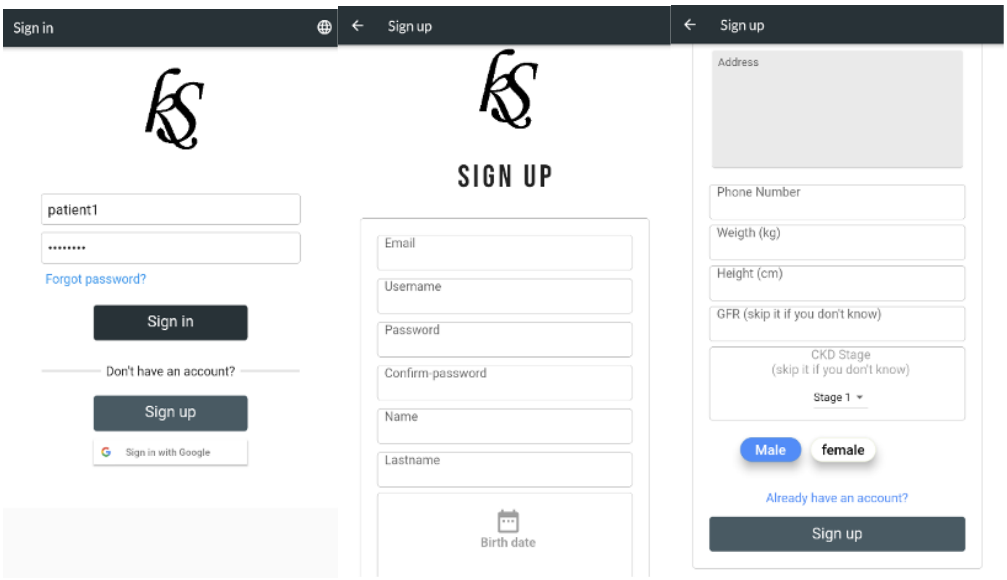


Figure 3.7: Log in and Register pages on kSmart mobile application.

Status bar

As soon as the user login, the first thing they will see is the status bar on the top of the first page. This bar will tell you an estimate amount of phosphorus by the information that users fill in when register.

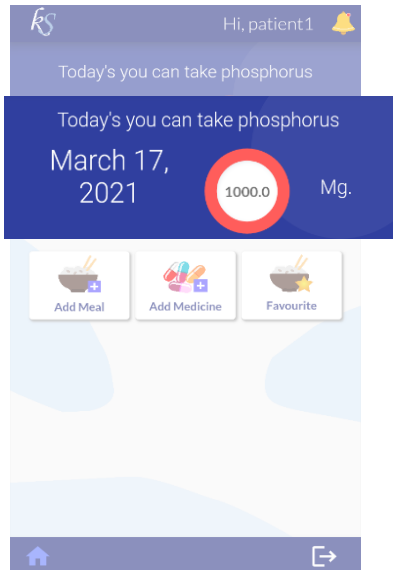


Figure 3.8: Statusbar on kSmart mobile application.

Calendar system

In the calendar page, user can see what they eat from the record just like the history system.

Figure 3.9: Calendar page on kSmart mobile application.

Medication system

In the Medication system, users can select various medicines via select medicine. Then users can then select the time of the day. Users can also set the Before Meal or After Meal.

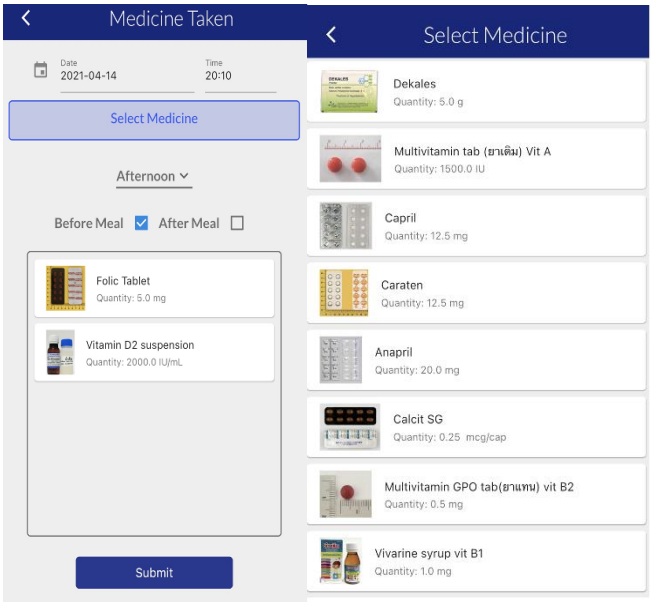


Figure 3.10: Add medicine taken page on kSmart mobile application.

Meal record system

Meal record system is the most important activity or function in this application. Users must use this system to record what they eat each day. The application provides 2 ways to add record for the user

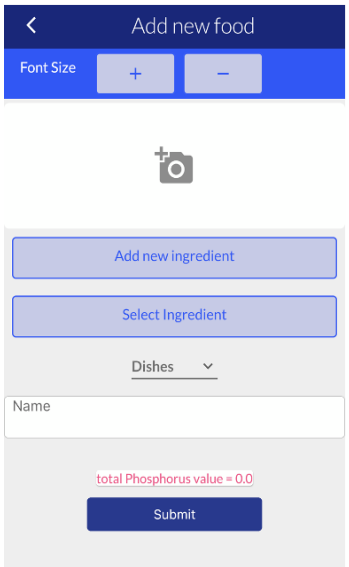


Figure 3.11: Add new food page on kSmart mobile application.

First is add new food, users will have to record what they eat like a customized system. Users must add name, picture, and ingredients by themself then record after adding everything. After adding new food, the food will be recorded in the server. Therefore, users can select it later easily on the existing food list.

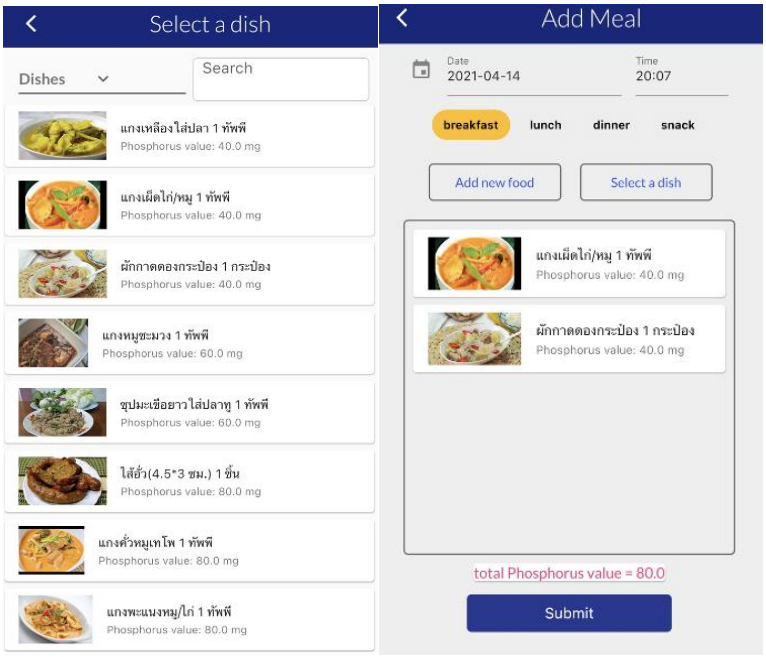


Figure 3.12: Add select food page on kSmart mobile application.

Second is Select a dish, the user will have to select a dish application provided as the default ingredient. Then, users can add additional ingredients if they want. This way will help the users save their precious time. As soon as users finished filling information, the application will automatically calculate the phosphorus. Then, compute with the status bar from the first page and tell the remaining phosphorus that users can take in that day.

Favourite system

In the favourite system, users can select a meal that users had recorded and save it to the Favourite page. In addition, users can find meals based on time of day that they had recorded.

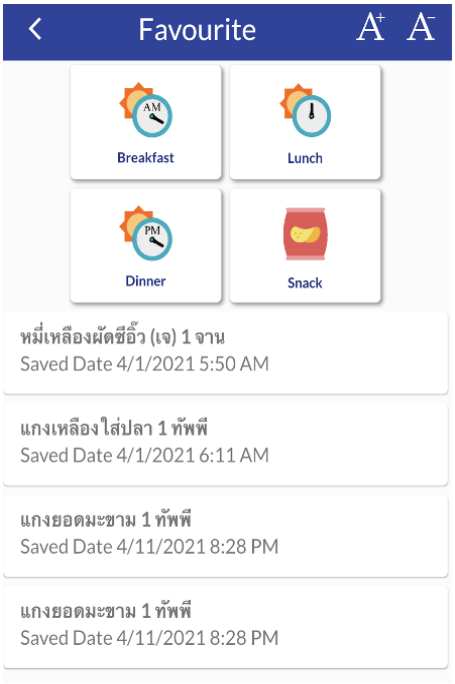


Figure 3.13: Favorite foods page on kSmart mobile application.

3.7.2. Website application design

Target users of this design are for the doctor. The Doctor can use this site to check the patient information and can also edit the patient information. The website also allows the doctor to either check or set the time for meeting then the mobile app will notify the patient. Doctors can use this page to inspect the overall behavior to help the doctor easier to assume the patient in the preliminary step.

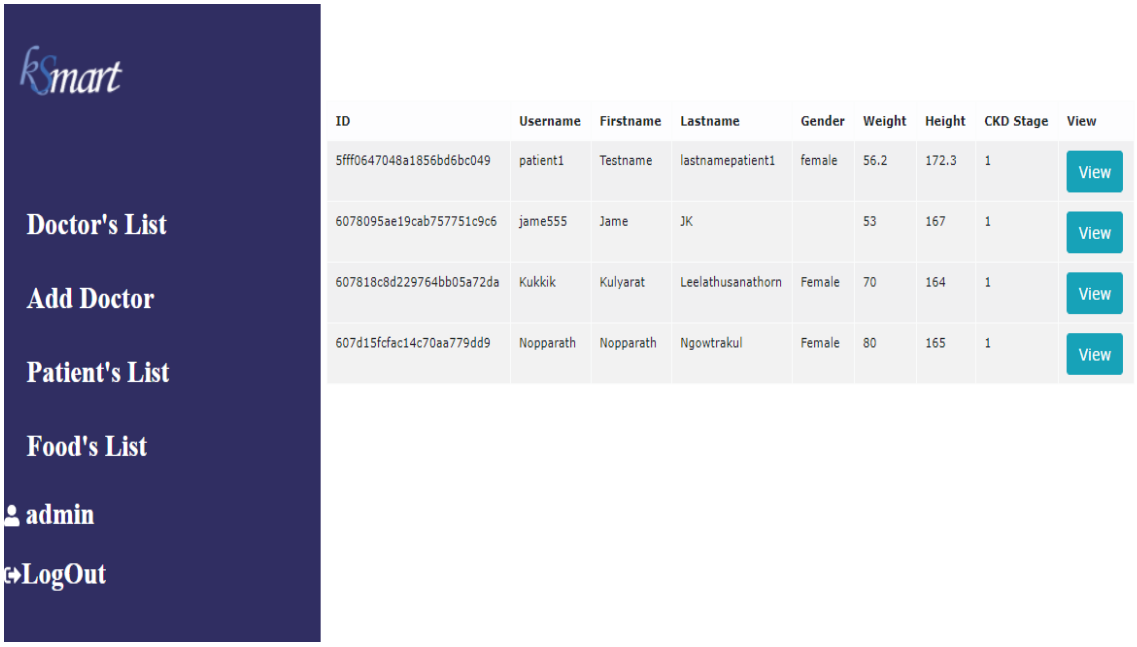


Figure 3.14: Patient list page on kSmart web application.

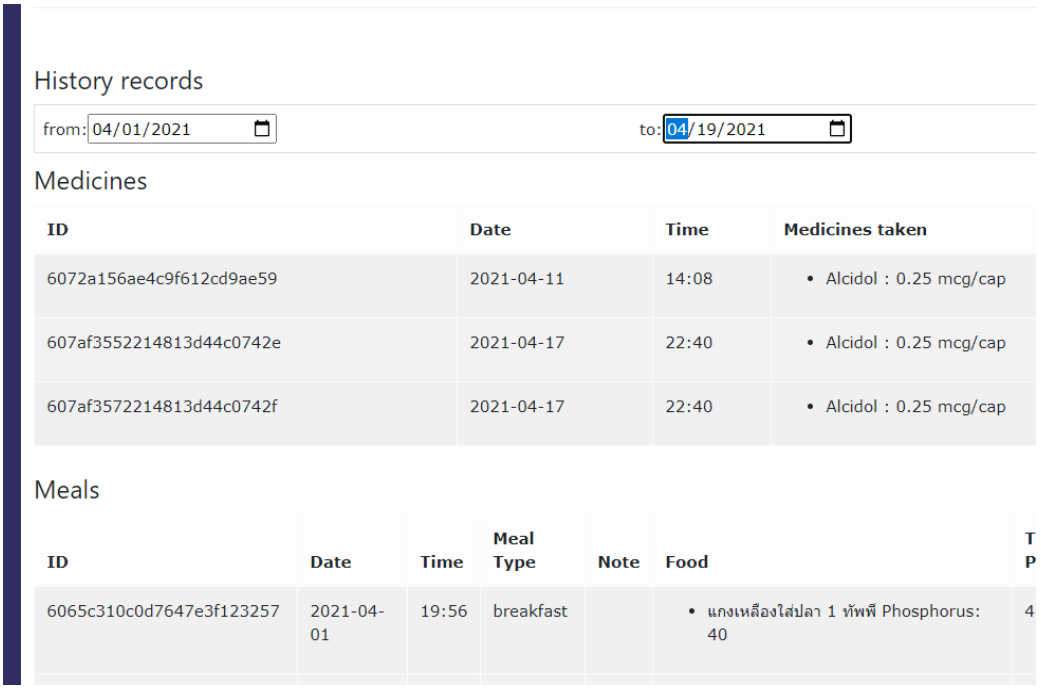


Figure 3.15: Patient's details page on kSmart mobile application.

In this page, doctor can select rage of date to see meal record and medicine taken record of the patient that will show in the table.

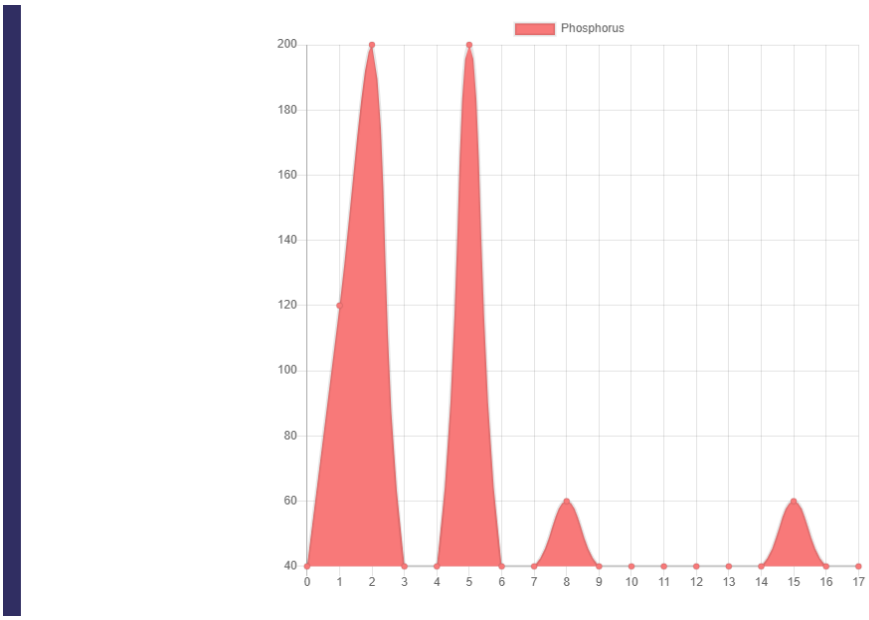


Figure 3.16: Statistic of phosphorus value graph on kSmart mobile application.

This graph shows the phosphorus value and date. X-axis show date and Y-axis show phosphorus value. Doctor can use this graph to investigate user behavior by comparing phosphorus value with the range of date that doctor select. For example, patients might take too high phosphorus in new year range because they have new year party.

CHAPTER 4
IMPLEMENTATION

This chapter describes hardware and software that were used to develop the software along with the system environment, implementation guides and the development techniques.

4.1 Hardware and System Environment

This project mainly includes three main hardware environments which are desktops, laptops and smartphones, which the devices that were used during the implementation of the software are Windows desktops, Windows laptops, MacBook, Android Smartphones and iOS smartphones. It is recommended from the developers that to implement the software, each type of computer should at least meet the specifications below in order to succeed in the tasks in implementation.

  - Laptop Processor - dual core @ 2.4 GHz (i5 or i7 or equivalent AMD)
  - RAM - 8 GB Hard Drive – 256GB SSD or HDD
  - Wireless - 802.11g/n (WPA2 support required) Backup Device - External hard
  - Smartphone
  - Model – Apple’s iPhone or Android Smartphones
  - Storage - 16 GB
  - RAM - 2 GB

- Web Server Software
  - Heroku Server – Deploy Spring Boot application and Vue.js
  - Atlas – Deploy MongoDB to connect with Heroku
- Editor
  - Intelli J – To implement Java Spring Boot project and Flutter project
  - Visual Studio Code – To Edit website
- Database Management System (DBMS)
  - MongoDB – To record data.

- Programming and Scripting Tools
    - Postman – To test services.
    - Swager – To test services.
    - Flutter Framework – To do mobile application.
    - Vue.js - To do web application
    - Spring Boot Framework – To build API services.

Knowledge Source

The websites which were the main resources of work are:

● stackoverflow.com. This site provides a place for developers to create and view forums which contain information about coding problems and solutions.

● github.com. GitHub provides many examples of coding projects in Ionic framework and many similar functions that were required in this project.

● w3schools.com. This website provides information about coding in HTML, CSS, JavaScript, PHP, AngularJS and SQL languages.

● flutter.dev Flutter website provides resources about the Flutter language we can use.

● pub.dev Pub website provides plugins such as calendar, material, local storage, google login etc.

● YouTube provides information and examples of coding videos.

Implementation Guide and Techniques

The developers chose Flutter as a tool for developing the interface of the application. It is a language which can easily experiment, build UIs, add features, fix bugs faster, beautiful Material Design, rich motion APIs, and full native performance on both iOS and Android.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:ksmart/locale/localization.dart';
import 'package:ksmart/locale/language_constants.dart';
import 'package:flutter_localizations/flutter_localizations.dart';
import 'package:ksmart/pages/login.dart';
import 'package:ksmart/globals.dart' as globals;
import 'package:ksmart/pages/statusbar.dart';


//void main() => runApp(new kSmartApp());
void main() {
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations([DeviceOrientation.portraitDown,DeviceOrientation.portraitUp])
    .then((_) {
    runApp(kSmartApp());
  });
}

class kSmartApp extends StatefulWidget {
  const kSmartApp({Key key}) : super(key: key);
  static void setLocale(BuildContext context, Locale newLocale) {
    _kSmartAppState state = context.findAncestorStateOfType<_kSmartAppState>();
    state.setLocale(newLocale);
  }
  @override
  _kSmartAppState createState() => _kSmartAppState();
```

Figure 4.1: Screenshot of Flutter language

This function is for configuring the initial setting of the application before running. The initial setting that appears here are checking locale to set language and allow application to rotate.

```
import Vue from 'vue';
import App from './App.vue';
import { router } from './router';
import store from './store';
import 'bootstrap';
//import PrimeVue from 'primevue/config';
import 'bootstrap/dist/css/bootstrap.min.css';
import {BIcon, } from 'bootstrap-vue'
import VeeValidate from 'vee-validate';
import { library } from '@fortawesome/fontawesome-svg-core';
import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome';
import {
  faHome,
  faUser,
  faUserPlus,
  faSignInAlt,
  faSignOutAlt
} from '@fortawesome/free-solid-svg-icons';


library.add(faHome, faUser, faUserPlus, faSignInAlt, faSignOutAlt);

Vue.config.productionTip = false;

Vue.use(VeeValidate);
Vue.component('BIcon', BIcon)
//Vue.use(PrimeVue);
Vue.component('font-awesome-icon', FontAwesomeIcon);
```

Figure 4.2: Screenshot of the vue.js framework's sample

Another tool we choose is VUE.JS for the website part. This is the main file of vue.js to call all library, component and template of the website.

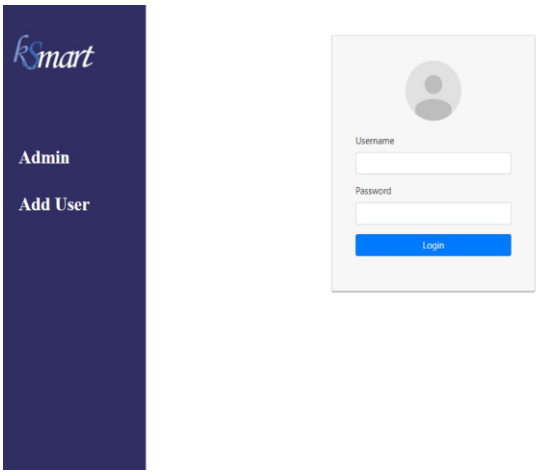


Figure 4.3: Log in page on kSmart web application

4.3 Functions

The application was built under Flutter Framework, which is an open-source development tool for building hybrid mobile applications. It has many benefits over pure native apps, specifically in terms of platform support and speed of development, and it holds UX and UI interactions which works like Bootstrap for Native. Most of the elements in the application contain similar groups of components, which each component consists of a CSS file, an HTML file, a Typescript file and a JSON object. The CSS files hold the color and design of the interface. The HTML files hold the structure and components of the interface. The Typescript files hold the algorithms that activate the functions. The JSON objects hold data in the application which will be sent to or received from the server in the application.

kSmart Application functions

Login and Sign Up

Implementing Login and Sign-Up pages is a simple but necessary step for many websites and applications.

```
@GetMapping("/iscurrent/{email}")
public String isCurrentUser(@PathVariable String email){
    if(userRepository.existsByEmail(email)) return "exist";
    return "not exist";

}


@PostMapping("/signin")
public ResponseEntity<?> authenticateUser(@Valid @RequestBody LoginRequest loginRequest) {

    Authentication authentication = authenticationManager.authenticate(
            new UsernamePasswordAuthenticationToken(loginRequest.getUsername(), loginRequest.getPassword()));


    SecurityContextHolder.getContext().setAuthentication(authentication);
    String jwt = jwtUtils.generateJwtToken(authentication);
    //System.out.println(jwt);


    UserDetailsImpl userDetails = (UserDetailsImpl) authentication.getPrincipal();
    List<String> roles = userDetails.getAuthorities().stream()
            .map(item -> item.getAuthority())
            .collect(Collectors.toList());
    if(roles.contains(ROLE_PATIENT)) {

        User user = userRepository.findByUsername(userDetails.getUsername()).get();
        //System.out.println(user.getUsername());
        Patient patient = patientRepository.findByUser(user).get();
        //  System.out.println(patient.getName());
        //String name, String lastname, Date birthdate, String address, String phoneNo, Double weight, Double height,
        // Double gfr, String gender, int ckdPhase
        PatientResponse patientRes = new PatientResponse(jwt,
```

Figure 4.4: The sample code of sign in method

```
@PreAuthorize("hasRole('ADMIN')")
@PostMapping("/adduser")
public ResponseEntity<?> addUser(@Valid @RequestBody AddUserRequest addUserRequest){
    if (userRepository.existsByUsername(addUserRequest.getUsername())) {
        return ResponseEntity
                .badRequest()
                .body(new MessageResponse("Error: Username is already taken!"));
    }

    if (userRepository.existsByEmail(addUserRequest.getEmail())) {
        return ResponseEntity
                .badRequest()
                .body(new MessageResponse("Error: Email is already in use!"));
    }
    User user = new User(addUserRequest.getUsername(),
            addUserRequest.getEmail(),
            encoder.encode(addUserRequest.getPassword()));

    Set<String> strRoles = addUserRequest.getRoles();
    Set<Role> roles = new HashSet<>();

    strRoles.forEach(role -> {
        switch (role) {
            case "admin":
                Role adminRole = roleRepository.findByName(ERole.ROLE_ADMIN)
                        .orElseThrow(() -> new RuntimeException("Error: Role is not found."));
                roles.add(adminRole);

                break;
            case "doctor":
                Role doctorRole = roleRepository.findByName(ERole.ROLE_DOCTOR)
                        .orElseThrow(() -> new RuntimeException("Error: Role is not found."));
                roles.add(doctorRole);
```

Figure 4.5: The sample code of add doctor method.

Login received username and password from Flutter and sent to Spring Boot, then Spring Boot will generate a token to user and send the data of user back to use in the system.

Register received data consist of username, password, email from Flutter to store in model user. In patients, the other data such as name, surname, address etc. will be stored in the patient model.

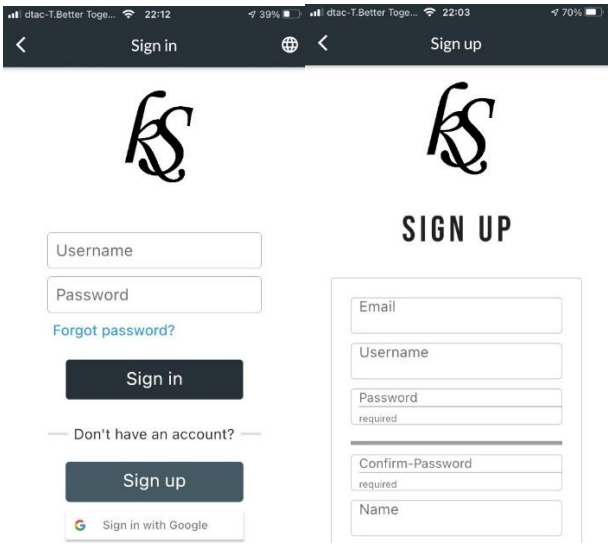


Figure 4.6: Sign in and sign-up pages on mobile application

Status bar and Homepage

```
class StatusPage extends StatefulWidget {

  StatusPage({Key key}) : super(key: key);
  @override
  _StatusPageState createState() => _StatusPageState();
}



class _StatusPageState extends State<StatusPage> {

  //to customize font size
  double fontsize = 20;
  double icon = 16;
  void increaseFontSize() async{
    setState(() {
      fontsize+=2;
      icon+=2;
    });
  }
  void decreaseFontSize() async{
    setState(() {
      fontsize-=2;
      icon-=2;
    });
  }
```

Figure 4.7: The sample code to create Statusbar and homepage

Homepage consists of a status bar, three buttons, and bottom bar. Status bar consists of pie charts that will decrease when the phosphorus taking that is the global variable decreases.

Three buttons consist of add meal button that will lead to add meal page, add medicine button that will lead to add medicine page, and save list button that will lead to save list page.

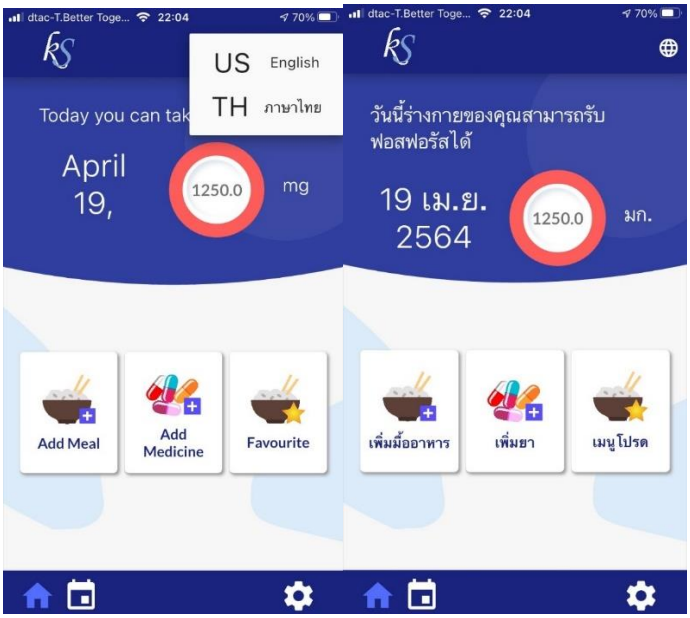


Figure 4.8: Show the process to change language

Add meal

```
@PostMapping("/addfood")
public ResponseEntity<?> addFood(@Valid @RequestBody FoodRequest foodRequest) {
    Food food = new Food(foodRequest.getName(),foodRequest.getIngredients(),
            foodRequest.getCategory(), foodRequest.getPhosphate());
    foodRepo.save(food);

    String id = food.getId();
    service.getRef(id);

    return ResponseEntity.ok(new MessageResponse("Add food successfully!"));
}

@PostMapping("/addingredient")
public ResponseEntity<?> addIngredient(@Valid @RequestBody IngredientRequest ingredientRequest){


    Ingredient ingredient = new Ingredient(ingredientRequest.getCategory(),ingredientRequest.getName(),ingredientRequest.getQuantity(),
            ingredientRequest.getUnit(),ingredientRequest.getPhosphorusValue());
    ingredientRepo.save(ingredient);

    String id = ingredient.getId();
    service.getRef(id);

    return ResponseEntity.ok(new MessageResponse("Add ingredient successfully!"));
}
```

Figure 4.9: The sample code of Add food and Add ingredient method

```
@GetMapping("/ingredients")
public List<Ingredient> ingredientList(){
    long start = System.nanoTime();

    // call the method
    List<Ingredient> ingredients = ingredientRepo.findAll();//ingredientServiceImpl.getIngredients();

    // get the end time
    long end = System.nanoTime();
    double execution = (end - start);
    System.out.println("Execution time: " + String.valueOf(execution)  + " nanoseconds");
    return ingredients;
}

@GetMapping("/foods")
public List<Food> foodList(){
    return foodRepo.findAll();//foodService.getFoods();
}

@PostMapping("/addmeal")
public ResponseEntity<?> addMeal(@Valid @RequestBody MealRequest mealRequest) throws ParseException {


    //Date datetime, boolean savelist, String mealType, String note, List<Food> dishes
    SimpleDateFormat format =new SimpleDateFormat("yyyy-MM-dd hh:mm");
    String str = mealRequest.getDatetime();
    String[] dateStr = str.split("T");
    Date date = format.parse(dateStr[0]+" "+dateStr[1]);
    Meal meal = new Meal(mealRequest.getUser(),date,mealRequest.isSavelist(),mealRequest.getMealType(),
            mealRequest.getDishes());
    mealRepo.save(meal);

    return ResponseEntity.ok(new MessageResponse("Add meal successfully!"));
```

Figure 4.10: The sample code of Add meal, food list and ingredient list

In the Add food page, users can add ingredients by the add ingredient button. Add ingredient receives image data and ingredient data from flutter and send to spring boot, then store the data to model ingredient and model image. Image and ingredient will map together by ID of ingredient and store the data in mongo DB. When we want to get the data, we will get the list from mongo DB in the form of lists through Spring Boot and Flutter. After the user added food, they had to add a name and can also add a food picture. Then, the data of ingredient lists and the filled data that are submitted will be sent to the spring boot and stored to the model food in mongo DB. When users save the record, phosphorus data of the meal that users selected will be saved in model meal and stored to the database.

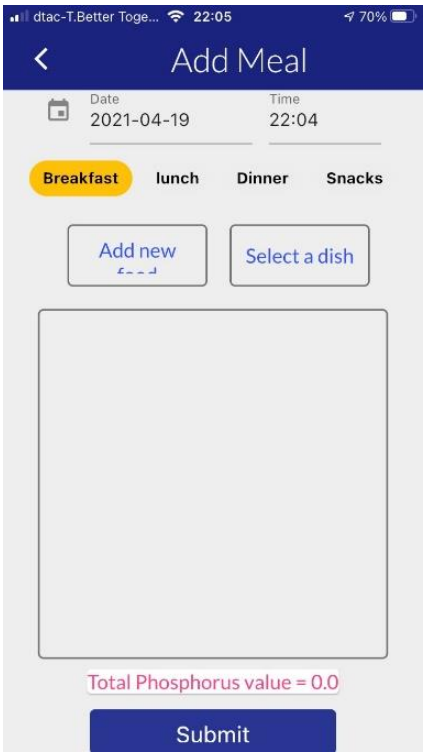


Figure 4.11: Add meal page on kSmart mobile application

Add Medicine

```
@RequestMapping("/api/med")
public class MedicineController {
    /*@Autowired
    MedService medService;*/
    @Autowired
    MedicineRepository medRepo;
    @Autowired
    MedicineTakeRepository medicineTakeRepository;
    @Autowired
    ImageService service;

    @PostMapping("/add")
    public ResponseEntity<?> addMed(@Valid @RequestBody MedicineRequest medicineRequest){
        Medicine med = new Medicine(medicineRequest.getName(),medicineRequest.getQuantity(),medicineRequest.getUnit());
        medRepo.save(med);
        String ref = med.getId();
        service.getRef(ref);

        return ResponseEntity.ok(new MessageResponse("Add medicine successfully!"));

    }
```

Figure 4.12: The sample code of Add medicine process

```
@PostMapping("/take")
public ResponseEntity<?> addMedTake(@Valid @RequestBody MedicineTakeRequest request){
    MedicineTake medicineTake = new MedicineTake(request.getUsername(),request.getTime(),request.getBefore(),
            request.getDate(),request.getMedicines());
    medicineTakeRepository.save(medicineTake);

    return ResponseEntity.ok(new MedicineTakeResponse(medicineTake.getId(),medicineTake.getUsername(),medicineTake.getTime(),
            medicineTake.getBefore(),medicineTake.getDate(),medicineTake.getMedicines()));
}

@GetMapping("/get/record/{username}")
public List<MedicineTake> getMedTakeRecord(@PathVariable("username") String username){
   /* List<MedicineTake> medTake = medicineTakeRepository.findAll();
    List<MedicineTake> myMedTake = new ArrayList<MedicineTake>();
    for(MedicineTake me : medTake){
        if(me.getUsername().equals(username)){
            myMedTake.add(me);
        }
    }
    return medTake;*/
    return  medicineTakeRepository.findByUsername(username);
}

@GetMapping("/get/all/record")
public  List<MedicineTake> getAllRecord(){
    return medicineTakeRepository.findAll();
}
```

Figure 4.13: The sample code of Add medicine taken process

Receiving image and information from Flutter and send to Spring Boot service. The Spring Boot stores images and data in image model and medicine model then stores into MongoDB. Select medicine page will call a list of all drugs from the database via Spring Boot and show them on the medicine page. Add medicine taken will receive information such as date, time, and medicine that the patient takes. Patients must select either take before or after meal box. At the end, Flutter will send the data to Spring Boot via http request and put them in a medicine taken model, then store them in MongoDB.

Figure 4.14: Add medicine taken page on kSmart mobile application.

Save lists.

```
@GetMapping("/getmeal")
public List<Meal> mealList(@RequestParam String username){
    List<Meal> allMeal = mealRepo.findAll();
    List<Meal> meals = new ArrayList<Meal>();
    if(allMeal==null) return meals;
    for(Meal m: allMeal){
        if(m.getUser().equals(username)){
            meals.add(m);
        }
    }
    System.out.println(meals);
    return meals;
}
```

Figure 4.15: The sample code of getting the meal record of specific user.

Request patient's food data from the server via Spring Boot and then Flutter will check which meal is in the save list. If it is a save list, it will show on the save list page. There are breakfast, lunch, and dinner buttons on the save list page. When we

click those buttons, Flutter will check which type of meals that are in the save list whether it be breakfast, dinner, or snacks.

Figure 4.16: Favorite food page on mobile application

Calendar

```
class CalendarPageState extends State<CalendarPage>{
  void _handleNewDate(date) {
    if (mounted){
      setState(() {
      _selectedDay = date;
      _selectedEvents = _events[_selectedDay] ?? [];
    });
    }
    print(_selectedEvents);
  }

  List _selectedEvents;
  DateTime _selectedDay;

  final Map<DateTime, List> _events = {};
```

Figure 4.17: The sample code to create calendar page

```
Widget _buildEventList() {
  return Expanded(
    child: ListView.builder(
      itemBuilder: (BuildContext context, int index) => Container(
        decoration: BoxDecoration(
          border: Border(
            bottom: BorderSide(width: 1.5, color: Colors.black12),
          ),  // Border
        ),  // BoxDecoration
        padding: const EdgeInsets.symmetric(horizontal: 0.0, vertical: 4.0),
        child: ListTile(
          title: Text(_selectedEvents[index]['name'].toString()),
          onTap: () {
            if(mounted) setState(() {
              var show = popUp(_selectedEvents[index]['id']);
              showDialog(context: context, builder: (BuildContext context) => show);
            });
          },
        ),  // ListTile
      ),  // Container
      itemCount: _selectedEvents.length,
    ),  // ListView.builder
  );  // Expanded
}
```

Figure 4.18: The sample code to show the detail of meal's records in each date

In Calendar, we have a clean calendar package that can create tables for the calendar and set events for the schedule such as what you do on this day and on that day the calendar in the app will show what meal you have recorded. As soon as you click on the list, the app will pop the details about meals such as the phosphorus value, the recorded time, the things you record.

Figure 4.19: Calendar page with the list of meal's records in the selected date

Notification

```
//initial workmanager for background process like notify when the app is closed
Workmanager.initialize(
    callbackDispatcher,
    isInDebugMode: true
);

//register the task for workmanager that will execute when the time that we set on duration coming
Workmanager.registerOneOffTask("1",
  "welcome",
  inputData: <String, dynamic>{
    'content':'welcome to kSmart',
    'number': 2
  },
  initialDelay: Duration(seconds: 10),
);
```

Figure 4.20: The sample code of running background task for notification

```
void callbackDispatcher() {
  print('callback');
  int count = 2;
  Workmanager.executeTask((task, inputData) {

    // initialise the plugin of flutterlocalnotifications.
    FlutterLocalNotificationsPlugin flip = new FlutterLocalNotificationsPlugin();

    // app_icon needs to be a added as a drawable
    // resource to the Android head project.
    var android = new AndroidInitializationSettings('@mipmap/ksmart_launcher');
    var IOS = new IOSInitializationSettings();

    // initialise settings for both Android and iOS device.
    var settings = new InitializationSettings(android: android, iOS: IOS);
    flip.initialize(settings);

    _showNotificationWithDefaultSound(flip,'Notification','${inputData['content']}');
    //globals.callbackTime = DateTime.now();
    //DateTime alert = DateTime(DateTime.now().year, DateTime.now().month,DateTime.now().day,23,35);
    //int diff = globals.callbackTime.difference(alert).inSeconds;
    int num = inputData['number'] + 1;
    Workmanager.registerOneOffTask("${inputData['number']}",
      "test",
      inputData: <String, dynamic>{
        'content': 'content',
        'number':num
      },
      initialDelay: Duration(hours: 3),
    );
```

Figure 4.21: The sample code of showing notification

```
Future _showNotificationWithDefaultSound(flip, String time, String message) async {

  // Show a notification after every 15 minute with the first
  // appearance happening a minute after invoking the method
  var androidPlatformChannelSpecifics = new AndroidNotificationDetails(
      '1000',
      'FLUTTER_NOTIFICATION_CHANNEL',
      'FLUTTER_NOTIFICATION_CHANNEL_DETAIL',
      importance: Importance.max,
      priority: Priority.high
  );
  var iOSPlatformChannelSpecifics = new IOSNotificationDetails();

  // initialise channel platform for both Android and iOS device.
  var platformChannelSpecifics = new NotificationDetails(
      android: androidPlatformChannelSpecifics,
      iOS: iOSPlatformChannelSpecifics
  );
  BuildContext context;
  await flip.show(0, time, message,
      platformChannelSpecifics, payload: 'Default_Sound'
  );
}
```

Figure 4.22: The sample code of the structure (title, message) of notification

In mobile application, we decide to use work manager plugin with local notification plugin of Flutter to do the notification schedule even if the application is close (background process) However, the solution is work well only on Android application for the iOS there is an exception while register a task to work manager. Therefore, we have another solution that will be developed in further that is using Firebase messaging.

To use firebase messaging, Spring Boot server have to have a scheduling class to check the time in each schedule in the system and pick a schedule that is near the current time to get a token device of patient and send a notification to that device. This solution is easy to send the specific notification to specific user.

The functions on kSmart Web Application

Edit food list

```
import UserService from '../service/user.service';
    export default {
        name: 'food',
        data () {
            return {
                food : {
                  "id":"",
                  "name":"",
                  "category":"",
                  "phosphorus":""
                }
            }
        },
        computed:{

        },
        created(){
          this.foodId=this.$route.params.foodId
        },
        mounted() {
          UserService.getFood(this.foodId).then(
            response => {
              this.food = response.data;
              console.log(this.food);
            },
            error => {
              this.content =
                (error.response && error.response.data) ||
                error.message ||
                error.toString();
            }
          );
```

Figure 4.23: The sample code of edit food on the web application.

Send a request to call a list of food from the server to show on the page, then if the users click on the list of food, they can edit the information. After they edit, when they click submit, the website will update to the database.

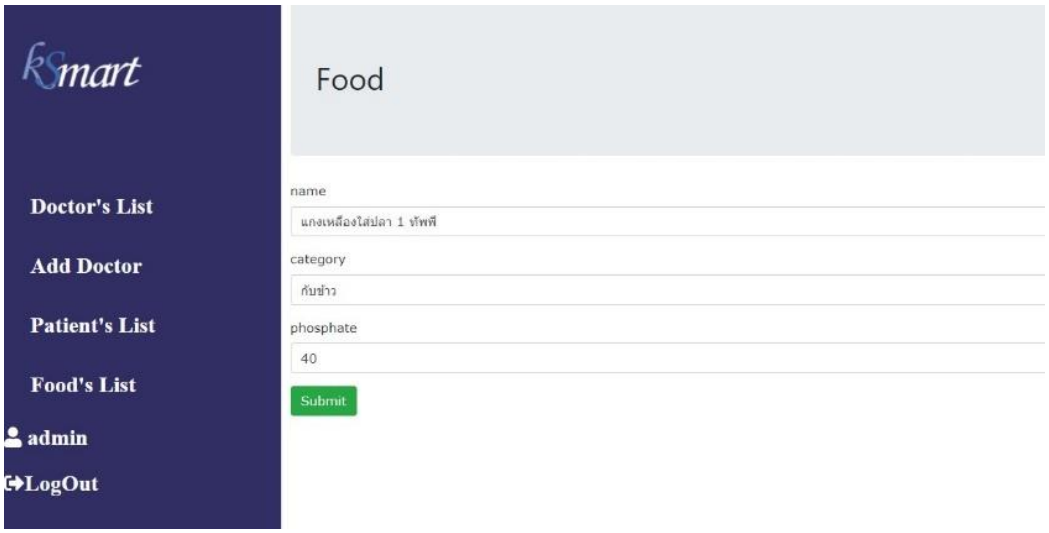


Figure 4.24: Edit food page on web application.

Inspect patient list and doctor list.

```
import UserService from '../service/user.service';

export default {
  name: 'Admin',
  data() {
    return {
      users: []
    };
  },
  mounted() {
    UserService.getDoctor().then(
      response => {
        this.users = response.data;
      },
      error => {
        this.content =
          (error.response && error.response.data) ||
          error.message ||
          error.toString();
      }
```

Figure 4.25: The sample code to get the list of doctors that is existing on the system.

```
import UserService from '../service/user.service';

export default {
  name: 'Doctor',
  data() {
    return {
      users: [],
      username: '',
    };
  },
  mounted() {
    UserService.getPatient().then(
      response => {
        this.users = response.data;
      },
      error => {
        this.content =
          (error.response && error.response.data) ||
          error.message ||
          error.toString();
      }
    );
  },
  methods:{
    gatAge: function(birthdate){
      let today = new Date();
      let age = Math.florr((birthdate - today)/31557600000);
      return age;
    }
  }
```

Figure 4.26: The sample code to get the list of patients that is existing on the system.

Send a request to call a list of patients and doctors from the server to show on the page, then if the users click on the lists, they can edit the information. After they edit, when they click submit, the website will update to the database.

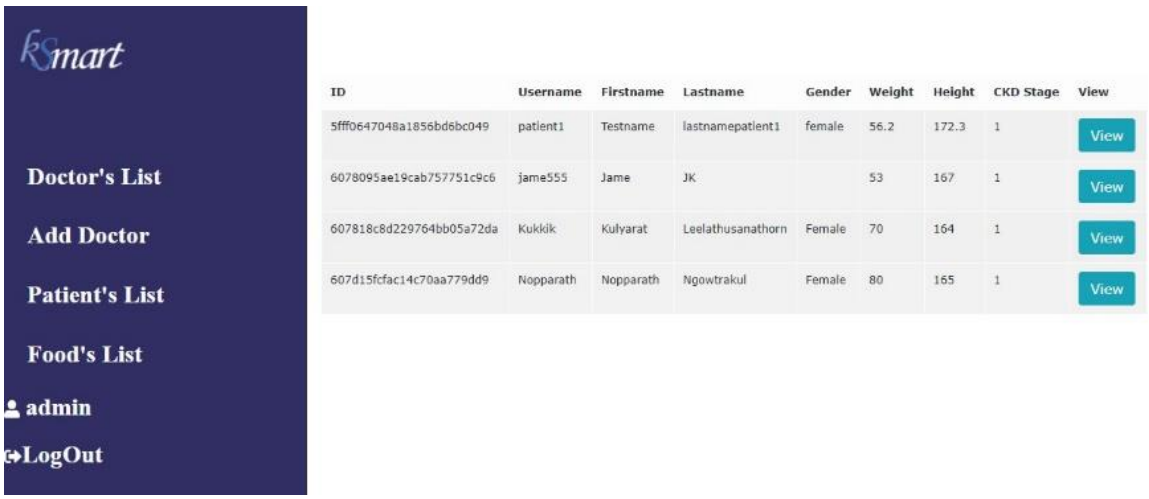


Figure 4.27: The page that shows patient list on the web application.

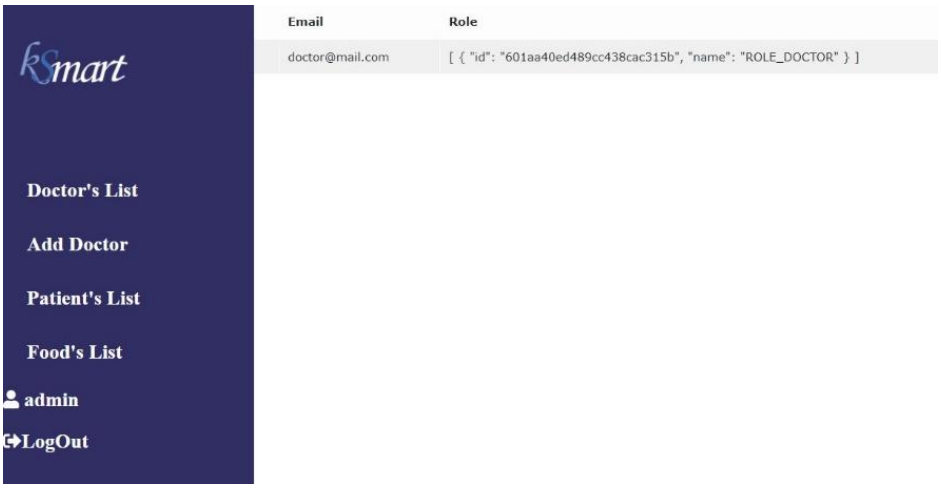


Figure 4.28: The first version of doctor's list page on web application

Add doctor.

```
import User from '../model/user';
import AuthService from '../service/auth.service.js';

export default {
  name: 'Register',
  data() {
    return {
      user: new User('', '', '',["doctor"]),
      submitted: false,
      successful: false,
      message: ''
    };
  },
  methods: {
    handleRegister() {
      this.message = '';
      this.submitted = true;
      this.$validator.validate().then(isValid => {
        if (isValid) {
          AuthService.register(this.user).then(
            data => {
              this.message = data.message;
              this.successful = true;
            },
            error => {
              this.message =
                (error.response && error.response.data) ||
                error.message ||
                error.toString();
              this.successful = false;
```

Figure 4.29: The sample code of Add doctor page on the web application.

Website will allow the admin to input the username, email, and password of the doctor, then the admin will send it to create in the database.

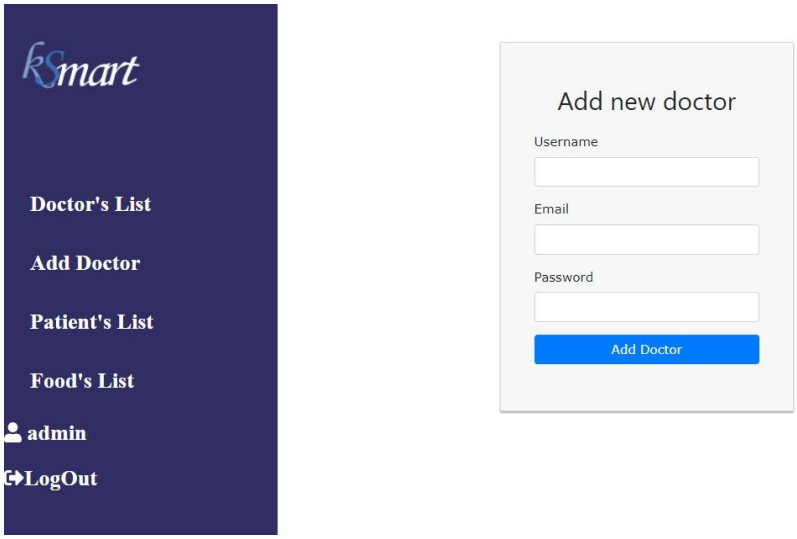


Figure 4.30: Add new doctor page on the web application.

Inspect patient information.

```
import UserService from '../service/user.service';
import PatientService from '../service/patient.service';
import LineChart from './Chart.vue'

    export default {
        components: { LineChart },
        name: 'view',
        data () {
            var today = new Date();
            return {
                user: null,
                content:'',
                username:'',
                //age:22,
                meals:[],
                medmeals:[],
                min: 240,
                max: 240,
                mean: 240,
                loaded: false,
                mealStart: today.toISOString().substr(0, 10),
                mealEnd: today.toISOString().substr(0, 10),
                datacollection: {
                  labels: [10, 11,12,13,14,15],
                  datasets: [
                  {
                    label: 'Phosphorus',
                    backgroundColor: '#f87979',
                    data: [10, 50,30,20,80]
                  }
                  ]
        },
            }
        },
```

Figure 4.31: The sample code of statistic chart in patient's detail page on the web application

Send a request to call a list of patient's information from the server to show on the page including ID, email, name, gender, weight, height, CKD stage, GFR, birthdate, age, address, phone number, history record, and medicine intake.

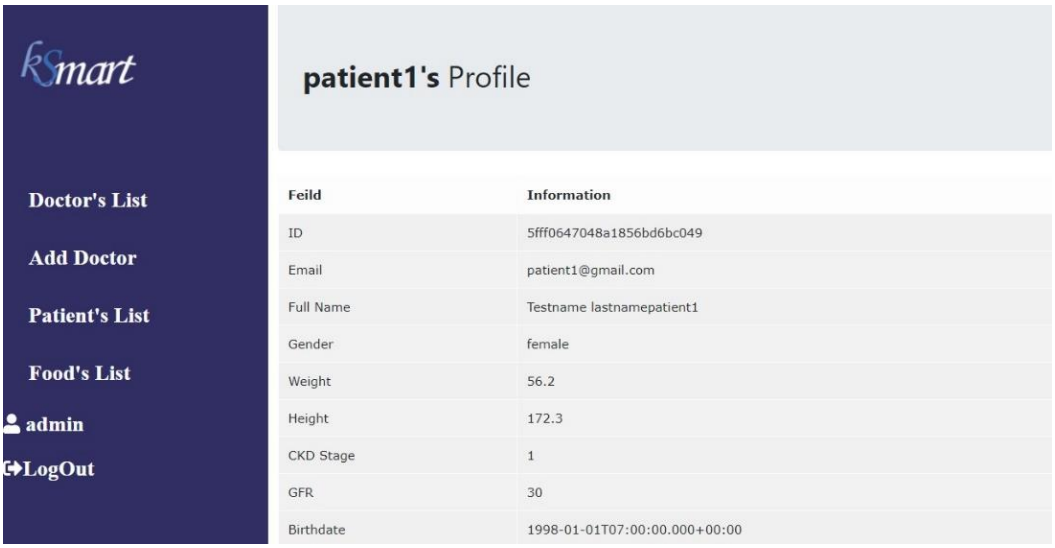


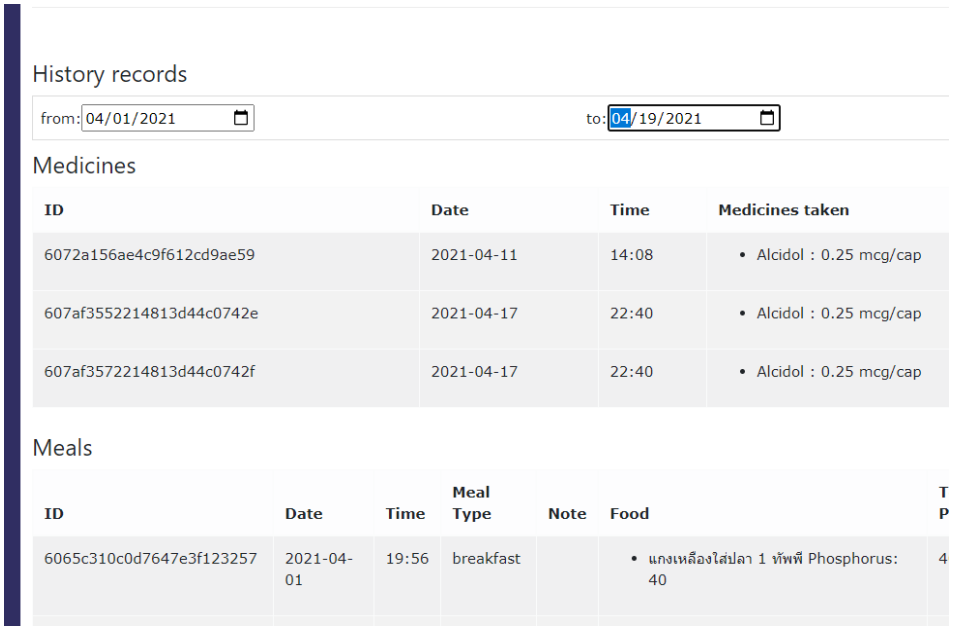
Figure 4.32: Patient's detail page on the web application

```
    UserService.getPatientMeal(this.username,this.mealStart,this.mealEnd).then(
      response => {
        this.meals = response.data;
      },
      error => {
        this.content =
          (error.response && error.response.data) ||
          error.message ||
          error.toString();
      }
    );
  },
  methods: {
      navigate() {
          router.go(-1);
      },
      mealListHandler() {

        UserService.getPatientMeal(this.username,this.mealStart,this.mealEnd).then(
          response => {
          this.meals = response.data;
          if(this.meals.length>0)
          {
            var phos = [];
            let sum = 0;
            this.min = this.meals[0].totalPhosphate;
            this.max = this.meals[0].totalPhosphate;
            var datacollection={
              labels: [],
              datasets: [
                {
                  label: 'Phosphorus',
                  backgroundColor: '#f87979',
                  data: []
                }
              ]
            }
            var labels=[];
            var data=[];
            for(let i=0;i<this.meals.length;i++){
                sum+=this.meals[i].totalPhosphate;
```

Figure 4.33: The sample code to show patient's meal records on the web application.

This part is implemented to calculate min, max and average of the phosphorus, then use that information to plot the graph.



History records

from: 04/01/2021 to: 04/19/2021

Medicines

ID	Date	Time	Medicines taken
6072a156ae4c9f612cd9ae59	2021-04-11	14:08	• Alcidol : 0.25 mcg/cap
607af3552214813d44c0742e	2021-04-17	22:40	• Alcidol : 0.25 mcg/cap
607af3572214813d44c0742f	2021-04-17	22:40	• Alcidol : 0.25 mcg/cap

Meals

ID	Date	Time	Meal Type	Note	Food	T P
6065c310c0d7647e3f123257	2021-04-01	19:56	breakfast		• แกงเหลืองใส่ปลา 1 ทัพพี Phosphorus: 40	4

Figure 4.34: Table to show medicine taken records and meal records.

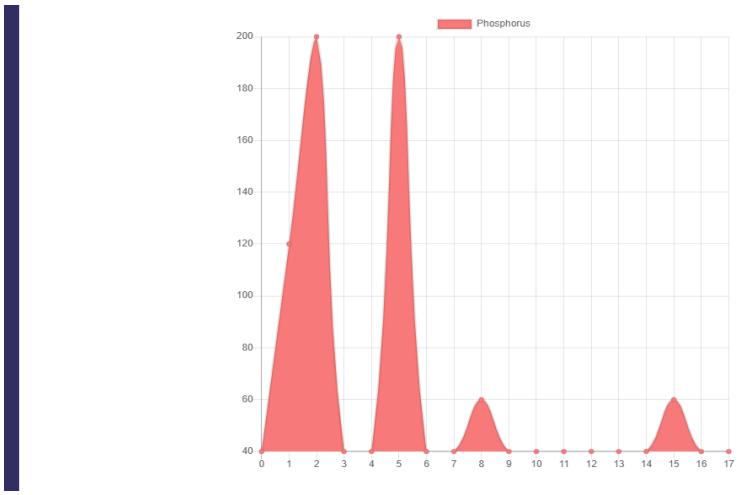


Figure 4.35: The graph of phosphorus value in each date (in the determine date range)

CHAPTER 5
TESTING AND EVALUATION

This chapter provides test results of the implemented system based on unit tests and a system integration test. For unit tests, we separate into two parts, a front-end system, and a back-end system. As for the system integration test, there are two categories that we have tested on the system: test scenarios and a user interface test.

5.1 Unit Tests

For the application unit tests, there are two operating systems to test which are Android operating system and iOS operating system. The unit tests will be based on the application pages from the first page to the last page and separate into two parts which are Front-end system and Back-end system. The process will begin by getting through all buttons and all functions. The list of unit test include:

Front-end System:

Process 1 Logging in page

Process 2 Sign up page, Google login.

Process 3 Main page, Status bar.

Process 4 Add meal page.

Process 4.1 Add new food button in Add meal page.

Process 4.1.1 Add new ingredient button in add new food page.

Process 4.1.2 Select ingredient button in add new food page.

Process 4.2 Select a dish function in Add meal page.

Process 5 Add medicine page.

Process 5.1 Select Medicine button in Add medicine page.

Process 6 Favorite page (save list)

Process 7 Calendar page

Process 8 Setting page

Back-end System:

Process 1 Authentication

Process 2 Recorded data and retrieve data.

Process 3 notification

Process 4 sync data

5.1.1 Android OS:

Front-end System

Table 5.1: Test Process 1 Logging in page

Operation performed	**Condition tested**	**Actual Result**
Inputting username and password	The application will let the users fill in the username and password.	Pass
Click on Forgot password	The application will lead users to the reset password page. Users have to fill in the email that they use to sign up to receive the mail from the system to set a new password.	Pass
Click on Sign in button	The application will lead the users to the main page with the account that they used to sign in.	Pass
Click on Sign up button	The application will lead to a Sign-up page to fill in their information to sign up.	Pass
Click on Sign in with Google button	The application will lead the users to Sign up page with the	Pass

	data that system fetch from the google account.	

Table 5.2: Test Process 2 Sign up page, Google login.

Operation performed	**Condition tested**	**Actual Result**
Click on all text boxes on the page.	The box should let the users fill in the information.	Pass
Click on the calendar box (Date of birth).	The application will allow users to select the day, month, and year of birth.	Pass
Click on the dropdown button.	The button should drop the list of CKD Stage to select.	Pass
Click on the select genders button.	Users can select between Male or Female genders. Which button they are selecting will become blue.	Pass
Click on Already have an account button.	The application will lead the users back to the sign in page.	Pass
Click on Sign up	The app will send that user information to the server. After that the application will lead users back to sign in page.	Pass

Table 5.3: Test Process 3 Main page, Status bar.

Operation performed	**Condition tested**	**Actual Result**
Click on the Language icon	The application changes language between Thai and English	Pass
Phosphorus bar display	When the users add a record, the bar will respond by decreasing the number and flow back of the bar. Daily Reset the phosphorus value.	Pass
Date display	The text shows the corresponding day to the actual day.	Pass
Click on link button	All buttons redirect to other pages correctly.	Pass
Click on calendar icon	The application will lead users to the calendar page	Pass
Click on Gear icon	The application will lead users to the setting page	Pass

Table 5.4: Test Process 4 Add meal

Operation performed	**Condition tested**	**Actual Result**
Date and time	The bar displays the time and date that is corresponding to the actual time when clicking the button Add Meal. Users can change the date and time.	Pass
Meal type selected bar	Users can select all the types of meals.	Pass
Item lists box	When users add items to the list, the box will show all items that users add.	Pass
Food list that users add form add new food and select a dish	The application will pop up the window for the user whether to edit or to delete the list.	Pass
Total value	The text must display the corresponding number to the total amount of phosphorus in the box.	Pass
Link button	All buttons redirect to other pages correctly.	Pass

Table 5.5: Test Process 4.1 Add new food button in Add meal page

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application goes back to the previous page.	Pass
Click on plus and minus icons in Font Size	All text on the page can increase or decrease font size.	Pass
Click on the Camera icon	The application pops the window to ask how to add the image (by camera or gallery). If users choose a camera, the application should lead to a camera to take a picture. On the other hand, If users choose from a gallery, the application will lead to the gallery to choose a picture.	Pass
Click on Add new ingredient button	The application leads users to the Add new ingredient page.	Pass
Click on Select ingredient button	The application ledads users to the Select ingredient page.	Pass
Click on dropdown button	The button drops the list of food types to select.	Pass
Click on the text box name	The box lets the users fill in the information.	Pass

Total Phosphorus value text bar	The bar displays the phosphorus summation value of all ingredients.	Pass
Click on Submit button	The application will pop the window to show that the information is successfully recorded and lead the users to the main page.	Pass

Table 5.6: Test Process 4.1.1 Add new ingredient button in add new food page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application goes back to the previous page.	Pass
Click on plus and minus icons in Font Size	All text in the page can increase or decrease font size.	Pass
Click on the Camera icon	The application pops the window to ask how to add the image (by camera or gallery). If users choose a camera, the application should lead to a camera to take a picture. On the other hand, If users choose from a gallery, the application will lead to the gallery to choose a picture.	Pass

Click on all dropdown buttons in the page	The buttons drop the list of food types and units to select.	Pass
Click on all text boxes in the page	All boxes let the users fill in the information.	Pass
Click on the submit button	The application will add information to the list in the previous page (Add new food) and lead the users to the Add new food page.	Pass

Table 5.7: Test Process 4.1.2 Select ingredient button in add new food page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application goes back to the previous page (add new food)	Pass
Click on dropdown list button	The buttons drop the lists of ingredient types to select.	Pass
Click on the search box	The box lets the users search for the food name by keyword.	Pass
Click on the ingredient lists	The application will take the list that users select to the previous page (Add	Pass

	new food) and lead the users to the Add new food page.	
Scrolling the page	The page goes up or down.	Pass

Table 5.8: Test Process 4.2 Select a dish function in Add meal page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application goes back to the previous page	Pass
Click on dropdown list button	The buttons drop the lists of food types to select.	Pass
Click on the search box	The box lets the users search for the food name by keyword.	Pass
Click on the food lists	The application will take the list that users select to the previous page (Add meal) and lead the users to the Add meal page.	Pass
Scrolling the page	The page should go up or down.	Pass

Table 5.9: Test Process 5 Add medicine

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application goes back	Pass

	to the previous page.	
Date and time	The bar displays the time and date that is corresponding to the actual time when clicking the button Add Meal. Users can change the date and time.	Pass
Click on select Medicine button	The application leads users to the Select Medicine page.	Pass
Click on dropdown button in the page	The buttons drop the lists of time periods to select.	Pass
Click on the check boxes	The users can click on two checkboxes to select one of them which are before meal or after meal. In addition, when one checkbox is selected another box has to remain blank.	Pass
Item lists box	When users add items to the list, the box will show all items that users add.	Pass
Click on the submit button	The application will pop the window to show that the information is	Pass

	successfully recorded and lead the users to the main page.	

Table 5.10: Test Process 5.1 Select Medicine button in Add medicine page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page.	Pass
Click on the medicine lists	The application will take the list that users select to the previous page (Add medicine) and leads the users to the Add medicine page.	Pass

Table 5.11: Test Process 6 Favorite (save list)

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page	Pass
Click on A plus and A minus icon	All text in the page can increase or decrease font size.	Pass
Sort list button	Users can click to sort their save list based on the time period or snacks	Pass
Click on the item(s) lists	The application will take item(s) in the list that	Pass

	users save to the Add meal page and lead the users to the Add meal page.	

Table 5.12: Test Process 7 Calendar

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page	Pass
Click on movement buttons on the calendar page	All the buttons should work properly including the left and right arrow to change the week and month page, the down arrow button to shows the days in one month.	Pass
Click on the number of days in the calendar page	The application should show the users' records that they recorded on that day. Moreover, users can click on each record to see the information including day, date, time, and phosphorus of that record.	Pass

Table 5.13: Test Process 8 Setting

Operation performed	**Condition tested**	**Actual Result**
Click on language button	The application should	Pass

	change language between Thai and English	
Click on the sync data	The application will pop up the loading window. After the loading is finished, the application will receive updated data from the server.	Pass
Click on Clear cache	The application will pop up the loading window. After the loading is finished, the application will clear the cache.	Pass
Click on About us	The application will lead to the about us page that provides the information about the developer and their advisors.	Pass
Click on Sign out	The application will log out from the username they are using, then lead users to the sign in page.	Pass
Click on Notification enable and disable bar	When users click on the bar, If it is enabled, the bar will disable. On the other hand, If it is disable, the bar will be enabled.	Pass

Table 5.14: Test Process 9 Thai Language

Operation performed	**Condition tested**	**Actual Result**
Text displays	All texts have to change to Thai language	Pass
All buttons	All buttons have to change to Thai language and can work properly.	Pass

Back-end System

Table 5.15: Test Process 1 Authentication

Operation performed	**Condition tested**	**Actual Result**
Check username and password	Check if the username and password that user's input are exists on the server, the application will log the users in. If it does not exist on the server, the application will not log the users in.	Pass

Table 5.16: Test Process 2 Recorded data and Retrieve data

Operation performed	**Condition tested**	**Actual Result**
Retrieve data	The application should retrieve data of the users, foods list, ingredients list, and medicines list properly with fast speed.	Pass

Recorded data	The application should send data of the users including food record, medicine taken, and save list to the server without fail.	Pass

Table 5.17: Test Process 3 notification

Operation performed	**Condition tested**	**Actual Result**
Sending notification	If the user enables the notification, when it is time that the user selects, the application will receive the notification by sending the token for calling the notification from the server to send notification to the users.	Pass

Table 5.18: Test Process 4 sync data

Operation performed	**Condition tested**	**Actual Result**
Update data	When the users click on sync data the application will send a token to call the update data from the server.	Pass

5.1.2 iOS:

Front-end System:

Table 5.19: Test iOS Process 1 Logging in page

Operation performed	**Condition tested**	**Actual Result**
Inputting username and password	The application will let the users fill in the username and password.	Pass
Click on Forgot password	The application will lead users to the reset password page. Users have to fill in the email that they use to sign up to receive the mail from the system to set a new password.	Pass
Click on Sign in button	The application will lead the users to the main page with the account that they used to sign in.	Pass
Click on Sign up button	The application will lead to a Sign up page to fill in their information to sign up.	Pass
Click on Sign in with Google button	The application will lead the users to Sign up page with the data that system fetch from the google account.	Pass

Table 5.20: Test iOS Process 2 Sign up page, Google login

Operation performed	**Condition tested**	**Actual Result**
Click on all text boxes on the page.	The box lets the users fill in the information.	Pass
Click on the calendar box (Date of birth).	The application will allow users to select the day, month, and year of birth.	Pass
Click on the dropdown button.	The button drops the list of CKD Stage to select.	Pass
Click on the select genders button.	Users can select between Male or Female genders. Which button they are selecting will become blue.	Pass
Click on Already have an account? button.	The application will lead the users back to the sign in page.	Pass
Click on Sign up	The users will save that username to the server. After that the application will lead users back to sign in page.	Pass

Table 5.21: Test iOS Process 3 Main page, Status bar.

Operation performed	**Condition tested**	**Actual Result**
Click on the Languages icon	The application should change language between Thai and English	Pass
Phosphorus bar display	When the users add a record, the bar will respond by decreasing the number and flow back of the bar.	Pass
Date display	The text shows the corresponding day to the actual day.	Pass
Click on link button	All buttons redirect to other pages correctly.	Pass
Click on calendar icon	The application will lead users to the calendar page	Pass
Click on Gear icon	The application will lead users to the setting page	Pass

Table 5.22: Test iOS Process 4 Add meal

Operation performed	**Condition tested**	**Actual Result**
Date and time	The bar displays the time and date that is corresponding to the actual time when clicking	Pass

	the button Add Meal. Users can change the date and time.	
Meal type selected bar	Users can select all the types of meal.	Pass
Item lists box	When users add items to the list, the box will show all items that users add.	Pass
Food list that users add form add new food and select a dish	The application will pop up the window for the user whether to edit or to delete the list.	Pass
Total value	The text must display the corresponding number to the total amount of phosphorus in the box.	Pass
Link button	All buttons redirect to other pages correctly.	Pass

Table 5.23: Test iOS Process 4.1 Add new food button in Add meal page

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page.	Pass
Click on plus and minus icons in Font Size	All text on the page can increase or decrease font size.	Pass
Click on the Camera icon	The application should pop the window to ask how to add the	Pass

	image (by camera or gallery). If users choose a camera, the application should lead to a camera to take a picture. On the other hand, If users choose from a gallery, the application will lead to the gallery to choose a picture.	
Click on Add new ingredient button	The application should lead users to the Add new ingredient page.	Pass
Click on Select ingredient button	The application should lead users to the Select ingredient page.	Pass
Click on dropdown button	The button should drop the list of food types to select.	Pass
Click on the text box name	The box should let the users fill in the information.	Pass
Total Phosphorus value text bar	The bar should display the phosphorus summation value of all ingredients.	Pass
Click on Submit button	The application will pop the window to show that the information is successfully recorded and lead the users to the main page.	Pass

Table 5.24: Test iOS Process 4.1.1 Add new ingredient button in add new food page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page.	Pass
Click on plus and minus icons in Font Size	All text on the page can increase or decrease font size.	Pass
Click on the Camera icon	The application should pop the window to ask how to add the image (by camera or gallery). If users choose a camera, the application should lead to a camera to take a picture. On the other hand, If users choose from a gallery, the application will lead to the gallery to choose a picture.	Pass
Click on all dropdown buttons in the page	The buttons should drop the list of food types, and units to select.	Pass
Click on all text boxes in the page	All boxes should let the users fill in the information.	Pass
Click on the submit button	The application will add information to the list in	Pass

	the previous page (Add new food) and leads the users to the Add new food page.	

Table 5.25: Test iOS Process 4.2 Select a dish function in Add meal page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page	Pass
Click on dropdown list button	The buttons should drop the lists of food types to select.	Pass
Click on the search box	The box should let the users search for the food name by keyword.	Pass
Click on the food lists	The application will take the list that users select to the previous page (Add meal) and lead the users to the Add meal page.	Pass
Scrolling the page	The page should go up or down.	Pass

Table 5.26: Test iOS Process 5 Add medicine

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go	Pass

	back to the previous page.	
Date and time	The bar should display the time and date that is corresponding to the actual time when clicking the button Add Meal. Users can change the date and time.	Pass
Click on select Medicine button	The application should lead users to the Select Medicine page.	Pass
Click on dropdown button in the page	The buttons should drop the lists of time periods to select.	Pass
Click on the check boxes	The users can click on two checkboxes to select one of them which are before meal or after meal. In addition, when one checkbox is selected another box has to remain blank.	Pass
Item lists box	When users add items to the list, the box will show all items that users add.	Pass
Click on the submit button	The application will pop the window to show that the information is	Pass

	successfully recorded and leads the users to the main page.	

Table 5.27: Test iOS Process 5.1 Select Medicine button in Add medicine page.

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page.	Pass
Click on the medicine lists	The application will take the list that users select to the previous page (Add medicine) and lead the users to the Add medicine page.	Pass

Table 5.28: Test iOS Process 6 Favorite (save list)

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page	Pass
Click on A plus and A minus icon	All text on the page can increase or decrease font size.	Pass
Sort list button	Users can click to sort their save list based on the time period or snacks	Pass
Click on the item(s) lists	The application will take	Pass

	item(s) in the list that users save to the Add meal page and lead the users to the Add meal page.	

Table 5.29: Test iOS Process 7 Calendar

Operation performed	**Condition tested**	**Actual Result**
Click on back arrow icon	The application should go back to the previous page	Pass
Click on movement buttons on the calendar page	All the buttons should work properly including the left and right arrow to change the week and month page, the down arrow button to show the days in one month.	Pass
Click on the number of days in the calendar page	The application should show the users' records that they recorded on that day. Moreover, users can click on each record to see the information including day, date, time, and phosphorus of that record.	Pass

Table 5.30: Test iOS Process 8 Setting

Operation performed	**Condition tested**	**Actual Result**
Click on language button	The application should	Pass

	change language between Thai and English	
Click on the sync data	The application will pop up the loading window. After the loading is finished, the application will receive updated data from the server.	Pass
Click on Clear cache	The application will pop up the loading window. After the loading is finished, the application will clear the cache.	Pass
Click on About us	The application will lead to the about us page that provides the information about the developer and their advisors.	Pass
Click on Sign out	The application will log out from the username they are using, then lead users to the sign in page.	Local storages will not be clean, so if we log in with another account. The data will be wrong.
Click on Notification enable and disable bar	When users click on the bar, If it is enabled, the bar will disable. On the other hand, If it is disable, the bar will be enabled.	Pass

Table 5.31: Test iOS Process 9 Thai Language

Operation performed	**Condition tested**	**Actual Result**
Text displays	All texts have to change to Thai language	Pass
All buttons	All buttons have to change to Thai language and can work properly	Pass

Back-end System:

Table 5.32: Test iOS Process 1 Authentication

Operation performed	**Condition tested**	**Actual Result**
Check username and password	Check if the username and password that users input are exists on the server, the application will log the users in. If it does not exist on the server, the application will not log the users in.	Pass

Table 5.33: Test iOS Process 2 Recorded data and Retrieve data

Operation performed	**Condition tested**	**Actual Result**
Retrieve data	The application should retrieve data of the users, foods list, ingredients list, and medicines list properly with fast speed.	Pass

Recorded data	The application should send data of the users including food record, medicine taken, and save list to the server without fail.	Pass

Table 5.34: Test iOS Process 3 notification

Operation performed	**Condition tested**	**Actual Result**
Sending notification	If the user enables the notification, when it is time that the user selects, the application will receive the notification by sending the token for calling the notification from the server to send notification to the users.	After defense (The code is working. The app store said we didn't turn on the permission of notification, but we did. Therefore, we have some problem with permission)

Table 5.35: Test iOS Process 4 sync data

Operation performed	**Condition tested**	**Actual Result**
Update data	When the users click on sync data the application will send a token to call the update data from the server.	Pass

Website testing (kSmart website)

5.2 Website Unit Tests

For the website unit tests, there are 2 groups of users to test which are admin and doctor. The unit tests will be based on the web pages from the first page to the last page and separate into 2 parts which are Front-end system and Back-end system. The process will begin by getting through all buttons and all functions. The list of unit test include:

Front-end System:

Table 5.36: Test Web Process 1 Logging in page

Operation performed	**Condition tested**	**Actual Result**
Inputting username and password	The website will let the users fill in the username and password.	Pass
Click on Sign in button	The website will lead the users to the main page with the account that they used to sign in.	Pass

Table 5.37: Test Web Process 2 Doctor list (Admin's only)

Operation performed	**Condition tested**	**Actual Result**
Information display	The page should show the information of the doctor including username, email, and role.	Pass

Table 5.38: Test Web Process 3 Add new doctor (Admin's only)

Operation performed	**Condition tested**	**Actual Result**
Inputting username, email, and password	The website will let the users fill in the username, email, and password.	Pass

Table 5.39: Test Web Process 4 Patient's list

Operation performed	**Condition tested**	**Actual Result**
Information display	The page should show the information of the patient including ID, username, full name, gender, weight, height, and CKD stage correctly.	Pass
View button	When the user clicks on the button, the website will lead the user to that patient page.	Pass

Table 5.40: Test Web Process 5 Food's list

Operation performed	**Condition tested**	**Actual Result**
Information display	The page should show the information of the food including ID, name, category, and phosphate correctly.	Pass

View button	When the user clicks on the button, the website will lead the user to edit the information.	Pass

Evaluation

There are two operating systems supported by kSmart application which are Android and iOS. In iOS, the developers have encountered huge limitations about permission. As a result, there are some functions that cannot work such as notification. Moreover, there are some crashes that the developers do not know about because we must use TestFlight to test applications in iOS and the app cannot send the dialog about the error. Therefore, the developers found out that it is difficult to solve the problems.

On the other hand, on Android Operating System, the program works smoothly and has less Bug than the iOS because the developer implements this project based on the Android Operating System. In addition, the developers have found out that some functions in the Android Operating System have to be implemented differently from the iOS.

CHAPTER 6

CONCLUSIONS

This chapter summarizes overall of the project activities including benefits to project developers, benefits to users, problems, limitations, and future work.

6.1 Benefits

kSmart Application is a web and mobile application to help CKD patients record each meal along with calculating and managing phosphorus and calcium levels in each meal. Moreover, it gives the benefits to users and project developers as the following:

6.1.1 Benefits to Project Developers

- Project developers have learned Flutter, Dart, and Android Studio to develop cross platform applications for iOS and Android smartphones.
- Project developers have learned to develop web-based applications.
- Project developers have learned to use Firebase and MongoDB to be a database and cloud storage of the application.
- Project developers can apply the knowledge of HTML, CSS, Java and Spring boot to develop web-based applications.

6.1.2 Benefits to Users

- Provide a simple interface that is easy to use for CKD patients or of young CKD patients.
- The users can use the application to calculate an amount of phosphorus in each food consumed.
- The users can know how much phosphorus they can take each day.
- The user can keep healthy by using the application to monitor amount of phosphorus regularly.
- The doctor can track the patient's consuming behavior.
- The doctor has more information to study CKD patients.

6.2 Problems and Limitations

- **Difficulty in communication;** When developing a project that belongs to a big system, like a hospital information system, requires a lot of information gathering. The developers had to continuously set up a meeting with Siriraj Hospital to gather information which many times could be difficult to do because the hospital staff have busy schedules.

- **Difficulty in gathering the data of user's evaluation;** The developer cannot get the evaluation easily as a result of COVID-19 situation. The target users of this project are the people in the Siriraj hospital. The developers are afraid that they might take risks when they go out to do the testing evaluation, so the data of testing evaluation is not satisfactory for the developer.

- **Certain size of food limitation:** The developer could not provide the most accurate information which is the size of the food proportion because there is no certain size of food proportion in the actual world. Therefore, the developers solved this problem by providing edit functions for food proportions that increase the difficulties to the users.

- **Food data limitation:** The developer could not provide all of the information which is the list of food and ingredients due to there being varieties of food and ingredients that existed in the actual world. At this point, there is a lot of data for the developers to handle. Therefore, the developers solved this problem by continuously adding the data to satisfy the usage of users.

- **Hardware limitation:** All of the applications must face this limitation which is hardware limitation. If the users do not have smartphones, the developer cannot help them to handle this problem.

6.3 Future Work

- Healthy food detection system. This system will provide users about the healthy food nearby them by google map to help the user find healthy food easier. The application can track the location and navigate the user to the destination.

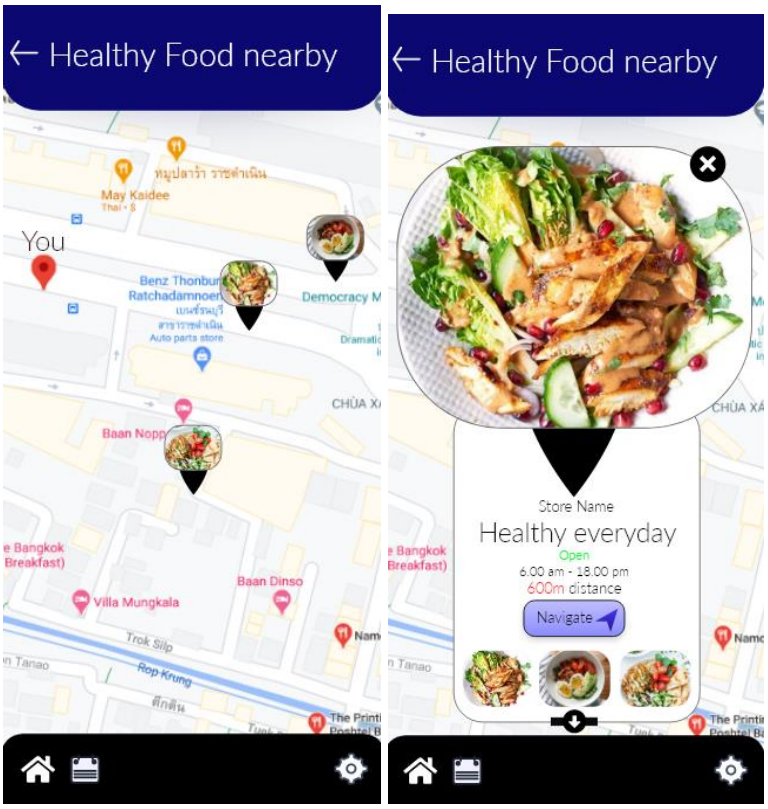


Figure 6.1: The sample design of healthy food nearby page

- Recommended system. The application provides recommended food based on the user information to help the user decide when they want to eat healthy or to remain on a low level of phosphorus.

Figure 6.2: The sample design of recommended food page

- User behavior tracking system. This system will tell the preliminary behaviors of the users by the information that they record every day and notify the users via display on the status bar if the phosphorus level of the user is at risk.

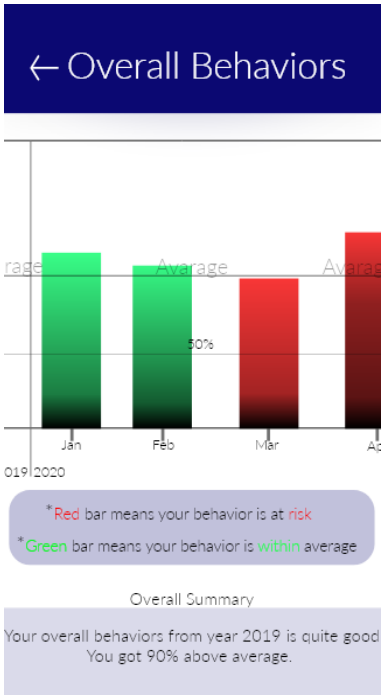


Figure 6.3: The sample page of patient behavior tracking

- News information system. The application will provide medical information or some information that may be useful for the user who needs it at the bottom of the first page. The information will be changed daily to keep in touch with the newest things.

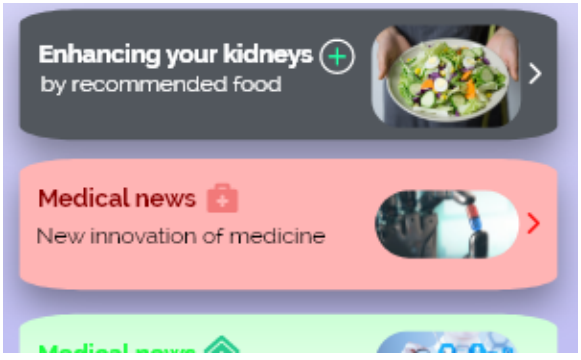


Figure 6.4: The sample design of news section on the homepage of mobile application

References

[1] "โรคไตเรื้อรัง," Param 9 Hodpital, 11 March 2020. [Online]. Available: https://www.praram9.com/articles/%E0%B9%82%E0%B8%A3%E0%B8%84%E0%B9%84%E0%B8%95%E0%B9%80%E0%B8%A3%E0%B8%B7%E0%B9%89%E0%B8%AD%E0%B8%A3%E0%B8%B1%E0%B8%87-2/. [Accessed 4 November 2020].

[2] ร. ภัทรากาญจน์, "โรคไตในเด็ก ป้องกันได," Faculty of Medicine Siriraj Hospital, 19 April 2016. [Online]. Available: https://www.si.mahidol.ac.th/th/healthdetail.asp?aid=1220. [Accessed 9 November 2020].

[3] ผ. ส. กันตชูเวสศิริ, "Patient with chronic kidney diseases(ผู้ป่วยโรคไตเรื้อรัง)," Faculty of Medicine Ramathibodi HospitalMahidol University, [Online]. Available: https://med.mahidol.ac.th/med/sites/default/files/public/pdf/medicinebook1/Patient%20with%20chronic%20kidney%20disease.pdf. [Accessed 8 August 2020].

[4] SiPH, "Food for Kidney," โรงพยาบาลศิริราช ปิยมหาการุณย์, 16 March 2018. [Online]. Available: https://www.siphhospital.com/th/news/article/share/570. [Accessed 8 August 2020].

[5] "Phosphorus," Mount Sinai, [Online]. Available: https://www.mountsinai.org/health-library/supplement/phosphorus. [Accessed 8 August 2020].

[6] N. K. Foundation, "Kidneys basics," National Kidney Foundation, [Online]. Available: https://www.kidney.org/kidney-basics. [Accessed 28 October 2020].

[7] "โรคไตเรื้อรัง (chronic kidney disease)," [Online]. Available: http://www.thaipediatrics.org/Media/media-20191010140730.pdf. [Accessed 9 September 2020].

[8] "Phosphorus and Your Diet," National kidney foundation, [Online]. Available:

https://www.kidney.org/atoz/content/phosphorus. [Accessed 20 July 2020].

[9] "โรคไตเรื้อรัง," [Online]. Available: https://www.bumrungrad.com/th/conditions/ckd-chronic-kidney-disease. [Accessed 8 September 2020].

[10] อ. เ. อ. และนักกำหนดอาหาร, "อาหารและฟอสฟอรัส," [Online]. Available: https://sriphat.med.cmu.ac.th/th/knowledge-63#:~:text=%E0%B8%9F%E0%B8%AD%E0%B8%AA%E0%B8%9F%E0%B8%AD%E0%B8%A3%E0%B8%B1%E0%B8%AA%20%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3&text=%E0%B8%A3%E0%B8%B0%E0%B8%94%E0%B8%B1%E0%B8%9A%E0%B8%9F%E0. [Accessed 10 october 2020].

[11] ศ. สุขขา, "ผลแลปเลือดกับโรคไตเรื้อรัง," 16 October 2016. [Online]. Available: https://pharmacy.mahidol.ac.th/th/knowledge/article/360/ผลแลปเลือดกับโรคไตเรื้อรัง/. [Accessed 8 September 2020].

[12] "โภชนาการสำหรับผู้ป่วยโรคไตเรื้อรัง," [Online]. Available: http://theworldmedicalcenter.com/th/new_site/health_article/detail/?page=โบนาการสำหรับผู้ป่วยโรคไตเรื้อรัง. [Accessed 2020].

[13] G. Thomas, "What is Flutter and Why You Should Learn it in 2020," 12 December 2019. [Online]. Available: https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/. [Accessed 18 July 2020].

[14] S. Montha, "Dart + Flutter is?," 31 July 2018. [Online]. Available: https://medium.com/@iamgique/dart-flutter-is-f61778caa6a0. [Accessed 18 July 2020].

[15] "Flutter," [Online]. Available: https://flutter.dev/docs/get-started/editor. [Accessed 8 August 2020].

[16] "เริ่มต้นสร้าง Android Application พื้นฐานด้วย Android Studio (Lab 3SB04)," 19 February 2018. [Online]. Available: https://medium.com/@palmz/เริ่มต้นสร้าง-android-application-พื้นฐานด้วย-android-studio-lab-3sb04-3fda43b07a1. [Accessed 18 July 2020].

[17] "รู้จักกับ Android Studio ซึ่งเป็น IDE Tool จาก Google ไว้พัฒนา Android โดยเฉพาะ," 26 March 2017. [Online]. Available: https://www.thaicreate.com/mobile/android-studio-ide.html.. [Accessed 10 August 2020].

[18] "IntelliJ IDEA overview," [Online]. Available: https://www.jetbrains.com/help/idea/discover-intellij-idea.html#IntelliJ-IDEA-supported-languages. [Accessed 18 July 2020].

[19] I. Punyasupa, "รู้จักโปรแกรม Adobe XD แบบเจาะลึกทุกฟีเจอร์ พร้อมข้อดีและข้อเสีย," Grappik Digital, 9 March 2020. [Online]. Available: https://www.grappik.com/inside-adobe-xd. [Accessed 19 July 2020].

[20] "ภาษา Java," 15 August 2016. [Online]. Available: http://marcuscode.com/lang/java. [Accessed 18 July 2020].

[21] "What is Kotlin language?," 7 November 2019. [Online]. Available: https://www.div24hr.com/ภาษา-kotlin-คืออะไร/. [Accessed 18 July 2020].

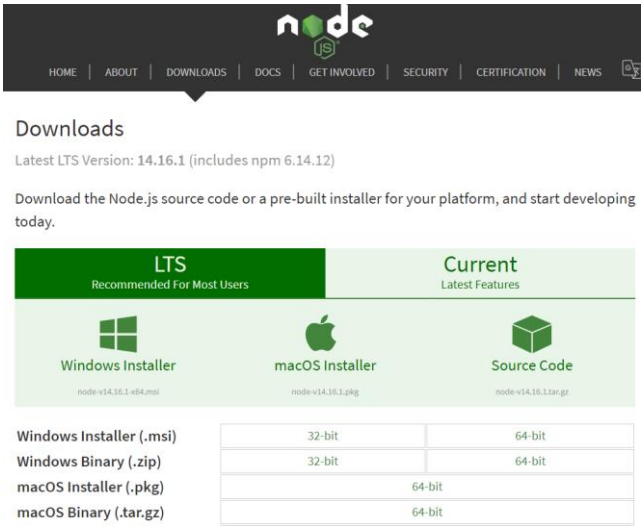
[22] "Swift," Apple, 2020. [Online]. Available: https://developer.apple.com/swift/. [Accessed 20 July 2020].

[23] M. CLINIC, "Chronic kidney disease," MAYO CLINIC, 15 August 2019. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/chronic-kidney-disease/symptoms-causes/syc-20354521. [Accessed 8 August 2020].

APPENDIX A
INSTALLATION MANUAL

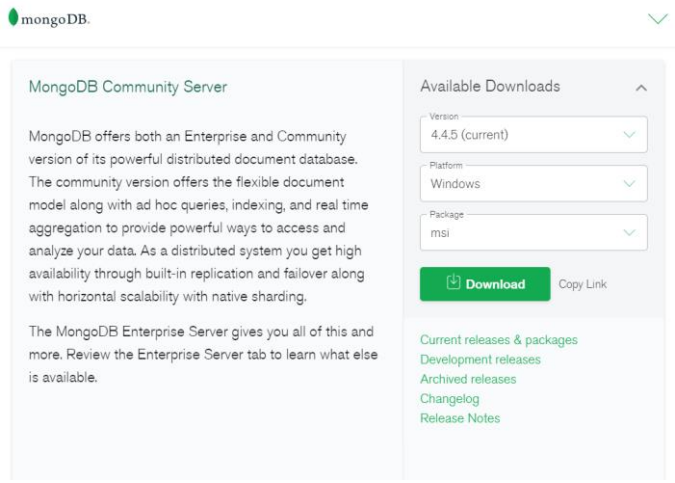
To install our system in local server, the server must be window system (there is no specific requirements for Window OS). Then the server must be installed Java runtime, Maven, Node.js and MongoDB. Java and Maven will be used to run the spring boot application (the system require java jdk 11). Node.js use for running web application. And MongoDB (with MongoDB Compass) use for monitoring the database.

Node.js



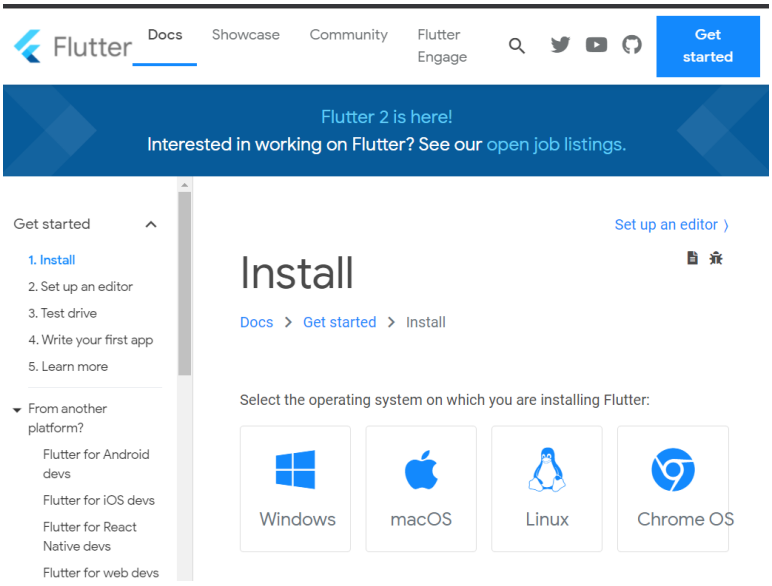
1. Go to https://nodejs.org/en/ to download Node.js installer file. Please select by the host server's Operation System.
2. Install Node.js.

MongoDB



1. Go to https://www.mongodb.com/try/download/community Please select by host server's operating system.
2. Install MongoDB and MongoDB Compass.

Flutter (For the mobile application testing purpose)



1. Go to https://flutter.dev/docs/get-started/install
2. Install flutter following instruction on the website.

The problems that might appear when install flutter.

On MacOS.

android studio plugins not found.

ln -s ~/Library/Application\ Support/Google/AndroidStudio4.1/plugins ~/Library/Application\ Support/AndroidStudio4.1

To install flutter on mac we have to **install cocoapods -> brew install cocoapods** (the most work because brew have a tricky that xcode command does not have)

To use **flutter Java JDk8 or 11 is work.** The latest version of java might not be compatible with flutter.

First time install flutter -> must run command -> **flutter doctor --android-licenses** -> and answer y (yes) to accept the agreement.

xcode cmd tool must install first (before xcode application)

To manage multiple java version -> write command export and switch in **.bash_profile**

On Window.

- The easiest is chocolate command. (It will come with node.js)
- use choco install dart.
- Java jdk 8 or 11

To run spring boot server for kSmart system.

1. We should have a spring boot project (that can be clone from our GitHub repository - https://github.com/Kukkik1998s/kSmart.git).
2. Path to the project in your device. And run mvn install command.
3. If there is no exception, you can continue run mvn run.
    a. If there is an exception about java runtime, you must check java version and java path in your device.

Now, you can use kSmart system.

Additional

If you need to run the web application separate from spring boot server, you can clone the project of the web application and use these commands.

BIOGRAPHIES

NAME	Miss Kulyarat Leelathusanathorn
DATE OF BIRTH	14 December 1998
PLACE OF BIRTH	Samutprakan , Thailand
INSTITUTIONS ATTENDED	Ratwinit Bangkaeo School 2017: High School Diploma
	Mahidol University, 2021: Bachelor of Science (ICT)

NAME	Miss Nopparat Ngowtrakul
DATE OF BIRTH	20 January 1998
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Sarasas Witaed Thonburi School 2017: High School Diploma
	Mahidol University, 2021: Bachelor of Science (ICT)

NAME	Miss Supisara Yingsom
DATE OF BIRTH	02 July 1998
PLACE OF BIRTH	Samutprakan, Thailand
INSTITUTIONS ATTENDED	Ratwinit Bangkaeo School 2017: High School Diploma
	Mahidol University, 2021: Bachelor of Science (ICT)

NAME	Mr. Kritsada Wittayakitjakul
DATE OF BIRTH	02 October 1998
PLACE OF BIRTH	Nakornpathom, Thailand
INSTITUTIONS ATTENDED	Sarasas Witaed Thonburi School 2017: High School Diploma
	Mahidol University, 2021: Bachelor of Science (ICT)