SPORTS INJURY AND ILLNESS RECORDING APPLICATION (SIRA)

แอปพลิเคชันบันทึกอาการบาดเจ็บและการเจ็บป่วยทางการกีฬา

BY

MR. KRISSANAPONG	**PALAKHAM**	**6288102**
MR. PONGSAKORN	**PIBOONPONGPUN**	**6288107**
MR. RATHAKIT	**SRIPRACHAYANUN**	**6288134**

ADVISOR
ASST. PROF. DR. CHARNYOTE PLUEMPITIWIRIYAWEJ

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

2022

ACKNOWLEDGEMENTS

We really would like to thank everyone who has taken part in our initiative and was relevant. This project could not be completed without the participation of all of the participants, including Mr. Krissanapong (Pun), Mr. Pongsakorn (Mark), and Mr. Rathakit Sriprachayanun (Art). We were also grateful to asst. Prof. Dr. Charnyote Pluempitiwiriyawej, who served as our principal advisor for his invaluable assistance, and to all of the advisors from the Faculty of Information and Communication Technology and the Faculty of Sports Sciences for their support. We could learn a lot from this endeavor and put what we have learned into practice as we completed the final report. Finally, we expected that our project would be entirely useful, and grateful and thanks to our parents, who encouraged us with their special support.

Mr. Krissanapong Palakham
Mr. Pongsakorn Piboonpongpun
Mr. Rathakit Sriprachayanun

SPORTS INJURY AND ILLNESS RECORDING APPLICATION

MR. KRISSANAPONG PALAKHAM 6288102 ITCS/B
MR. PONGSAKORN PIBOONPONGPUN 6288107 ITCS/B
MR. RATHAKIT SRIPHACHAYANUN 6288134 ITCS/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: ASST. PROF. DR. CHARNYOTE PLUEMPITIWIRIYAWEJ

ABSTRACT

This project is an application project to collect data on illnesses and injuries which are important problems for athletes and record staff data to record their data. Athletes can be timely by collecting data of athletes that will be saved in paper or Google form where it is likely to be lost and compiled to be difficult to use. Our application is built to record and track injuries and illnesses through the OSTRC Questionnaire and the IOC Injury record which will be shown to the athletes as scores and messages. If the score exceeds the OSTRC star criteria, staff can see the case scores of the athletes who exceed the criteria, they can record and send messages to the athletes so that athletes can receive treatment, relieve injuries and prevent athletes from using their bodies during training or competition.

KEYWORDS: INJURY/ILLNESS/PSYCHOLOGY/ATHLETE/STAFF/
APPLICATION/RECORDING
126 P.

แอปพลิเคชันบันทึกอาการบาดเจ็บและการเจ็บป่วยทางการกีฬา

กฤษณพงศ์ ปาลคำ 6288102 ITCS/B
พงศกร พิบูลพงศ์พันธ์ 6288107 ITCS/B
รัฐกฤษฎิ์ ศรีปรัชญานันท์ 6288134 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ผศ. ดร. ชาญยศ ปลื้มปิติวิริยะเวช

บทคัดย่อ

การทำโครงงานครั้งนี้เป็นโครงงานทำแอพพลิเคชั่นโดยมีวัตถุประสงค์เพื่อใช้สำหรับการเก็บข้อมูลอาการเจ็บป่วยและอาการบาดเจ็บซึ่งเป็นปัญหาสำคัญต่อนักกีฬาและการบันทึกข้อมูลของบุคลากรเพื่อบันทึกข้อมูลของนักกีฬาได้ทันท่วงทีโดยการเก็บข้อมูลของนักกีฬานั้นจะถูกบันทึกไว้ในกระดาษหรือแบบฟอร์มกูเกิล (GOOGLE FORM) ซึ่งมีโอกาสสูญหายและเรียบเรียงนำมาใช้งานได้ยาก แอปพลิเคชันของเราสร้างเพื่อบันทึกและติดตามอาการบาดเจ็บรวมถึงอาการเจ็บป่วยโดยผ่านแบบสอบถามตาม OSTRC และบันทึกข้อมูลอาการบาดเจ็บตาม IOC ซึ่งจะแสดงให้นักกีฬาได้เห็นเป็นคะแนนและข้อความโดยหากคะแนนนั้นเกินเกณฑ์ OSTRC บุคลากรจะสามารถเห็นเคสคะแนนของนักกีฬาที่คะแนนเกินเกณฑ์และสามารถบันทึกและส่งข้อความให้แก่นักกีฬาเพื่อให้นักกีฬาสามารถได้รับการรักษา บรรเทาอาการบาดเจ็บและป้องกันการใช้ร่างกายของนักกีฬาในตอนฝึกซ้อมหรือการแข่งขัน

126 หน้า

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1
INTRODUCTION

In this chapter, there are six parts of the introduction which are Motivation, Problem Statement, Objectives of the Project, Scope of the Project, Expected Benefits of the Project, and Organization of the Document. Their focus will be on how the application was developed, why there is a need for the application, and what were the benefits of the application.

1.1 Motivation

Nowadays, injuries usually happen whether physical, mental, or health, especially for athletes. Injury problems can affect athletes with their training, practice, as well as competition. Athletes' performance would be decreased, resulting in their winning success in a competition or evenly lacking training. In addition, staff (e.g., coaches, doctors, physical therapists, psychologists) would also be affected by athletes' injuries. Due to athlete's injury problems, staff have to create or change a plan to suit the athlete's problem. Thus, it would be difficult to adjust if the athlete's problem information is sent late. In order to prevent those problems, keeping records of the athlete's problems as well as medical recordings from diagnostics are important. Collecting the record from both athletes and staff will assist the athletes' injury problems in order to decrease treatment time, so athletes can heal themselves with the correct method and also understand their injury status and what they need to do. Moreover, staff can plan whether healing, training, or competition for injured athletes. Therefore, building a platform for keeping those records will extremely be useful for both athletes and staff.

1.2 Problem Statement

Recording the athlete's problems happened in Thailand. However, there are two main problems with recording systems:

1.2.1 Recording with Paper

The paper-based recording has several drawbacks, including being hard to find, easily lost, difficult to retrieve, and requiring a large amount of labor. At the moment, we switch to different methods to avoid these issues.

1.2.2 Recording with Google Form

Google Form recordings are changed methodology from the paper based. However, this method of recording injuries and illness has also had some problems whether scattered information, data integration, and difficult processing. Thus, this method must be changed to a new one in order to improve the recording system.

1.3 Objectives of the Project

- Develop a SIRA system capable of managing data on physical injuries, health illnesses, and mental illnesses
- Make athletes know their injury status in order to heal themselves with the correct methodology
- Enable coaches and the medical staff to utilize athlete data to be useful and provide coaching and treatment input to specific athletes

1.4 Scope of the Project

- A mobile platform that can record both athlete’s questionnaires and complaints as well as medical recordings from staff
- The operating systems provide both IOS and Android in order to collect for all athletes and staff data from all user devices
- Applied the Oslo Sports Trauma Research Center (OSTRC) questionnaires and International Olympic Committee (IOC) record form to the systems
- Sending weekly notifications for athletes in order to answer questionnaires every week

1.5 Expected Benefits

In order to build the application, we expected our solution would be assisted by any kind of users that were categorized into two groups: users and developers. Furthermore, users were classified into three groups including athletes, staff, and organization agents. The description is as follows:

- Athlete
  - Receiving a recommendation from staff for healing themselves
  - Regularly receiving health checks from the professional staff
  - Filling out the health questionnaire forms as simply with clearly user interfaces
- Staff
  - Filling out the medical record form as simple with clearly user interfaces
  - Planning for healing, training, competition for injured athletes
- Developer
  - Practicing mobile programming both coding and framework
  - Creating an assistant program for athletes and medical staff
  - Developing an application for both IOS and android operating systems with Dart language and Google Firebase Services
  - Evolving problem-solving, time management, prioritizing tasks, programming, database management, UX/UI, and presentation skill

1.6 Organization of the Document

This document consists of 6 chapters including:

1. Introduction – The motivation, problem statements, project objectives, project scopes, expected benefits for both users and developers, and document organization are all contained in the introduction's first chapter, which also introduces the project.
2. Background – Background knowledge, including a literature review, is included in the second chapter.
3. Analysis and Design – The project's analysis and design are presented in the third chapter. It includes details on the project's design, such as a

system architecture overview, a system structure diagram, and a design for the web-based service.

4. Implementation – The fourth chapter discusses the implementation and includes information on hardware, system environment, implementation techniques, and implementation guide.
5. Testing and Evaluation – The testing and evaluation process, spread pattern results, and discussion are all found in the fifth chapter.
6. Conclusion – Conclusion, benefits, issues and limitations, and future work are all included in the sixth chapter.

CHAPTER 2
BACKGROUND

This chapter includes examples of background knowledge and earlier projects completed by other scholars that we applied some methodology to this project. We gave a summary of the project's content and an example of required knowledge in order to provide more understanding to this project.

2.1 Background Knowledge

2.1.1 The International Olympic Committee Surveillance Form

During the Olympic Games 2012 in London, the International Olympic Committee (IOC) provided a medical record form for all stakeholders whether common medical staff, physician team, physiotherapist, and so on. In order to prevent danger from injuries, all stakeholders could be finished the medical form and analysis a symptom of athletes. There are two types of report including injury and illness report that each type has a different required information. [1]

athlete's accreditation no. 123456789		sport and event (2) athletics, 100m (women)		round, heat or training (3) quarter final / 1st heat		date and time of injury 7.8.2012 - 14:35
injured body part, side (4) wrist, left	code 15	type of injury (5) sprain	code 8	cause of injury (6) slipped and fell	code 21	absence in days (1) 10

Figure 2.1: The example of IOC injury record form

athlete's accreditation no. 1564579587979		sport and event (2) football (men)		diagnosis tonsillitis, cold		occurred on (date) 02.08.2012
affected system (7) respiratory	code 1	main symptom(s) (8) fever, pain	code 1,2	cause of illness (9) infection	code 2	absence in days (1) 2

Figure 2.2: The example of IOC illness record form

2.1.2 The Oslo Sports Trauma Research Center Questionnaire Form

There are questionnaires from the Oslo Sports Trauma Research Center (OSTRC) that are separated into three parts including health, overuse, and sport psychology. The health questionnaire is about the illness of the athletes, for instance, fatigue, fever, and so on. The overuse questionnaire is about the physical injury from any part of the body. The sport psychology questionnaire is about the athletes' mental illness, for instance, readiness to compete, training, and sleep quality. [2]

Table 2.1: Thai and English in OSTRC Health questionnaires

Thai version	**English version**
คุณมีปัญหาด้านอาการเจ็บป่วยหรือไม่ • ใช่ • ไม่มี	Do you have an illness problem? • Yes • No
1. ใน 7 วันที่ผ่านมาปัญหาการบาดเจ็บการเจ็บป่วย หรือปัญหาสุขภาพอื่น ๆ ของท่านทำให้การฝึกซ้อมหรือการแข่งขันกีฬามีปัญหาหรือไม่ • เข้าร่วมการฝึกซ้อมหรือแข่งขันกีฬาได้เต็มที่โดยไม่มีปัญหาการบาดเจ็บ ,เจ็บป่วย หรือปัญหาสุขภาพ • เข้าร่วมการฝึกซ้อมหรือแข่งขันกีฬาได้เต็มที่แต่มีปัญหาการบาดเจ็บ,เจ็บป่วย หรือปัญหาสุขภาพ • เข้าร่วมการฝึกซ้อมหรือแข่งขันกีฬาได้ไม่เต็มที่เพราะมีปัญหาการบาดเจ็บ ,เจ็บป่วย หรือปัญหาสุขภาพ • ไม่สามารถเข้า ร่วมการฝึกซ้อมหรือแข่งขนกีฬาได้ เพราะมีปัญหาการบาดเจ็บ ,เจ็บป่วย หรือปัญหาสุขภาพ	Have you had any difficulties participating in normal training and competition due to injury, illness, or other health problems during the past week? • Full participation without health problems. • Full participation, but with injury/illness. • Reduced participation due to injury/illness. • Cannot participate due to injury/illness.

Thai version	English version
2. ใน 7 วันที่ผ่านมา ปัญหาการบาดเจ็บการเจ็บป่วย หรือปัญหาสุขภาพ ของท่านส่งผลกระทบต่อปริมาณการฝึกซ้อมหรือแข่งขันมากน้อยเท่าไหร่ • ไม่ลดลง • ลดลงเล็กน้อย • ลดลงปานกลาง • ลดลงอย่างมาก • ไม่สามารถเข้าร่วมได้เลย	To what extent have you reduced you training volume due to injury, illness, or other health problems during the past week? • No reduction • To a minor extent • To a moderate extent • To a major extent • Cannot participate at all
3. ใน 7 วันที่ผ่านมา ปัญหาการบาดเจ็บการเจ็บป่วย หรือปัญหาสุขภาพ ของท่านส่งผลกระทบต่อความสามารถ ในการเล่นกีฬามากน้อยเท่าไหร่ • ไม่ลดลง • ลดลงเล็กน้อย • ลดลงปานกลาง • ลดลงอย่างมาก • ไม่สามารถเข้าร่วมได้เลย	To what extent has injury, illness or other health problems affected your performance during the past week? • No effect • To a minor extent • To a moderate extent • To a major extent • Cannot participate at all
4. ใน 7 วันที่ผ่านมา ท่านมีปัญหาการบาดเจ็บ การเจ็บป่วยหรือปัญหาสุขภาพมากน้อยเพียงใด • ไม่สามารถเข้าร่วมได้เลย • มีอาการหรือปัญหาสุขภาพเล็กน้อย • มีอาการหรือปัญหาสุขภาพพอประมาณ • มีอาการหรือปัญหาสุขภาพอย่างมาก	To what extent have you experienced symptoms/health complaints during the past week? • No symptoms/health complaints • To a mild extent • To a moderate extent • To a severe extent • Cannot participate at all

Thai version	English version
5.อาการป่วย • ไข้ • อ่อนล้า • ต่อมอักเสบ • เจ็บคอ • คัดจมูก/น้ำ มูกไหล/จาม • ไอ • หายใจลำบาก • ปวดหัว • คลื่นไส้ • อาเจียน • ท้องเสีย • ท้องผูก • เป็นลม • ผื่นคัน • หัวใจเต้นผิดปกติ • เจ็บหน้าอก • ปวดเมื่อยกล้ามเนื้อเนื้อส่วนท้อง • ความเจ็บปวดอื่น ๆ • ชา • ความวิตกกังวล • หดหู่/เศร้า • หงุดหงิดง่าย • อาการบริเวณตา • อาการบริเวณหู • อาการที่ทางเดินปัสสาวะและอวัยวะเพศ	Illness part • Fever • Fatigue/malaise • Swollen gland • Sore throat • Blocked nose, running nose, sneezing • Cough • Difficulty breathing • Headache • Nausea • Vomiting • Diarrhea • Constipation • Fainting • Rash/itchiness • Irregular pulse/arrhythmia • Chest pain/angina • Abdomen pain • Other pain • Numbness/pins and needles • Anxiety • Depression • Irritability • Eye symptoms • Ear symptoms • Symptoms form urinary tract and genitals

Thai version	English version
9. คุณมีปัญหาอาการเจ็บป่วยอื่น ๆ หรือไม่ • มี • ไม่มี	Do you have another illness problem? • Yes • No

Table 2.2: Thai and English in OSTRC Overuse questionnaires

Thai version	English version
ท่านมีอาการบาดเจ็บหรือไม่ • มี • ไม่มี	Do you have an injury problem? • Yes • No
บริเวณที่บาดเจ็บ • หัวและหน้า • คอ • หัวไหล่/ไหปลาร้า • ต้นแขน • ข้อศอก • แขนท่อนล่าง • ข้อมือ • มือและนิ้ว • หน้าอก/ซี่โครง • หน้าท้อง • กระดูกันหลังทรวงอก • กระดูกสันหลังส่วนล่าง • เชิงกราน/ก้น • สะโพก/ขาหนีบ • ต้นขา • เข่า • ขาท่อนล่าง • ข้อเท้า • เท้าและนิ้วเท้า	Injury part • Head • Neck • Shoulder • Upper arm • Elbow • Lower arm • Wrist • Hand and finger • Chest and ribs • Abdomen • Thoracic spine • Lower spine • Pelvis and buttocks • Hip and groin • Thigh • Knee • Lower leg • Ankle • Feet and toes

Thai version	English version
1.ใน 7 วันที่ผ่านมา ปัญหา (Injury) ของท่านทำให้การเข้าร่วมฝึกซ้อมหรือการแข่งขันกีฬามีปัญหาหรือไม่ • เข้าร่วมการฝึกซ้อมหรือการแข่งขันได้เต็มที่โดยไม่มีปัญหา (Injury) • เข้าร่วมการฝึกซ้อมหรือการแข่งขันกีฬาได้เต็มที่แต่มีปัญหา (Injury) • เข้าร่วมการฝึกซ้อมหรือแข่งขันกีฬาได้ไม่เต็มที่เพราะมีปัญหา (Injury) • ไม่สามารถเข้าร่วมการฝึกซ้อมหรือแข่งขันกีฬาได้เลยเพราะมีปัญหา (Injury)	Have you had any difficulties participating in training and competition due to (Injury) problems during the past 7 days? • Full participation without (Injury) problems. • Full participation, but with (Injury) problems • Reduced participation due to (Injury) problems • Could not participate due to (Injury) problems.
2. ใน 7 วันที่ผ่านมา ปัญหา (Injury) ของท่านส่งผลกระทบต่อการฝึกซ้อมหรือแข่งขันมากน้อยเพียงใด • ไม่ส่งผลกระทบต่อการฝึกซ้อมหรือแข่งขันเลย • การฝึกซ้อมหรือแข่งขันลดลงเล็กน้อย • การฝึกซ้อมหรือแข่งขันลดลงปานกลาง • การฝึกซ้อมหรือแข่งขันลดลงอย่างมาก • ไม่สามารถเข้าร่วมการฝึกซ้อมหรือแข่งขันได้เลย	To what extent have you modified your training or competition due to (Injury) problems during the past 7 days? • No modification • To a minor extent • To a moderate extent • To a major extent • Cannot participate at all

Thai version	English version
3.ใน 7 วันที่ผ่านมา ปัญหา (Injury) ของคุณส่งผลกระทบต่อความสามารถในการเล่นกีฬามากน้อยเพียงใด • ไม่ส่งผลกระทบต่อความสามารถในการเล่นกีฬาเลย • ความสามารถในการเล่นกีฬาลดลงเล็กน้อย	To what extent have (Injury) problems affected your performance during the past 7days? • No effect • To a minor extent • To a moderate extent • To a major extent • Cannot participate at all

Thai version	English version
• ความสามารถในการเล่นกีฬาลดลงปานกลาง	
นอน ไม่หลับภายหลังจากเข้านอนไปแล้วเกินกว่า 30 นาที • ไม่เคยเลยในช่วงระยะเวลา 1 เดือนที่ผ่านมา • น้อยกว่า 1 ครั้งต่อสัปดาห์ • 1 หรือ 2 ครั้งต่อสัปดาห์ • 3 ครั้งหรือมากกว่าต่อสัปดาห์	Cannot get sleep within 30 minutes • Not during the past month • Less than once a week • Once or twice a week • Three or more times a week
4. ใน 7 วันที่ผ่านมา ท่านมีอาการเจ็บปวดของ (Injury) ของท่านซึ่งเป็นผลมาจากการเข้าร่วมการแข่งขัน/ฝึกซ้อมกีฬาอยู่ในระดับใด • ไม่เจ็บเลย • เจ็บเล็กน้อย • เจ็บพอประมาณ • เจ็บมาก	To what extent have you experienced Head pain related to your sport during the past 7 days? • No pain • Mild pain • Moderate pain • Severe pain

Table 2.3: Thai and English in Sport Psychology questionnaires

Thai version	English version
ในช่วงระยะเวลา 1 เดือนที่ผ่านมา ท่านใช้ยาเพื่อช่วยในการนอนหลับบ่อยเพียงใด (ไม่ว่าจะตามใบสั่งแพทย์หรือซื้อมาเอง) • ไม่เคยเลยในช่วงระยะเวลา 1 เดือนที่ผ่านมา • น้อยกว่า 1 ครั้งต่อสัปดาห์ • 1 หรือ 2 ครั้งต่อสัปดาห์ • 3 ครั้งต่อสัปดาห์	During the past month, how often have you take medicine (prescribe or "over the counter") to help you sleep? • Not during the past month • Less than once a week • Once or twice a week • Three or more times a week
รู้สึกตัว ตื่นขึ้นระหว่างนอนหลับกลางดึกหรือตื่นเช้ากว่าเวลาที่ตั้งใจไว้ • ไม่เคยเลยในช่วงระยะเวลา 1 เดือนที่ผ่านมา • น้อยกว่า 1 ครั้งต่อสัปดาห์ • 1 หรือ 2 ครั้งต่อสัปดาห์ • 3 ครั้งต่อสัปดาห์ขึ้นไป	Wake up in the middle of the night or early morning • Not during the past month • Less than once a week • Once or twice a week • Three or more times a week
ในช่วงระยะเวลา 1 เดือนที่ผ่านมา ท่านคิดว่าคุณภาพการนอนหลับโดยรวมของท่านเป็นอย่างไร • ดีมาก • ค่อนข้างดี • ค่อนข้างแย่ • แย่มาก	During the past month, how would you rate your sleep quality overall? • Very good • Fairly good • Fairly bad • Very bad

• 1 หรือ 2 ครั้งต่อสัปดาห์ • 3 ครั้งต่อสัปดาห์ขึ้นไป	• Three or more times a week

2.2 Literature Review

2.2.1 The health problems survey from the Oslo Sports Trauma Research Center

Prior to recently, the majority of research on sports injury prevention consisted of observational studies that outlined injury risk in various activities as well as their incidence, pattern, and severity. However, few studies had been created to offer comprehensive data on injury processes and risk factors data that was necessary in order to suggest appropriate preventative strategies.

Thai version	English version
อาการอื่นๆ ที่รบกวนขณะนอนหลับ โปรดระบุ • เติมลงในช่องว่าง	Other restlessness while you sleep • Fill the blank
ความมั่นใจโดยรวมในการหลับมาเล่นกีฬา • 0 1 2 3 4 5 6 7 8 9 10	My overall confidence to play is: • 0 1 2 3 4 5 6 7 8 9 10
ความมั่นใจในการกลับมาเล่นโดยไม่เจ็บ • 0 1 2 3 4 5 6 7 8 9 10	My confidence to play without pain is: • 0 1 2 3 4 5 6 7 8 9 10
ความมั่นใจที่จะทุ่มเท 100% • 0 1 2 3 4 5 6 7 8 9 10	My confidence to give 100% effort is: • 0 1 2 3 4 5 6 7 8 9 10
ความมั่นใจที่จะไม่กังวลกับส่วนที่เคยบาดเจ็บ • 0 1 2 3 4 5 6 7 8 9 10	My confidence to not concentrate on the injury is: • 0 1 2 3 4 5 6 7 8 9 10
ความมั่นใจว่าส่วนที่เคยบาดเจ็บจะกลับมาเล่นได้ไหว • 0 1 2 3 4 5 6 7 8 9 10	My confidence in the injured body part to handle demands of the situation is: • 0 1 2 3 4 5 6 7 8 9 10
ความมั่นใจในความสามารถของฉัน • 0 1 2 3 4 5 6 7 8 9 10	My confidence in my skill level/ability is: • 0 1 2 3 4 5 6 7 8 9 10

Based on this foundation, the Oslo University Hospital and the Norwegian School of Sport Sciences collaborated to establish the Oslo Sports Trauma Research Center (OSTRC) in May 2000. As a FIFA Medical Center of Excellence, the OSTRC was officially opened in 2009. The facility was also chosen to be one of the first four International Olympic Committee (IOC) Research Centers for injury prevention that year. [3]

2.2.2 Better reporting of sports-related overuse injuries and health issues

The Oslo Sports Trauma Research Center (OSTRC) believed that these improvements would improve the respondents' experience and, as a result, maximized their adherence, and this paper offered updates to the OSTRC surveys. These impressions were influenced by environmental factors, including athlete experience, sport level, sports kind, and season. This implied that data gathered from various athletic cohorts would not necessarily be comparable. We supported additional studies on the psychometric characteristics of the OSTRC questionnaires in various contexts and groups. [4]

2.2.3 Overuse injury questionnaire methodology improvement

According to the research, a new overuse injury questionnaire has been improved in several sessions with many participants such as physiotherapists, doctors, athletes, questionnaire specialists, and others. This questionnaire was used for the management of injuries in overuse areas, especially the knee, lower back, and shoulder. The overuse injury questionnaire has evolved and has changed, such as some specific physical questions not relevant to the sport in which one is playing. In order to replace the questionnaire, all inquiries pertaining to particular work duties were dropped, and the emphasis was placed on documenting the severity of illness, the effects of injury, and athletes' performance. The questions in the questionnaire were chosen as "problem" rather than "injury" because athletes interpret "injury" differently. [5]

2.2.4 Material and Method for Oslo Sports Trauma Research Center Questionnaire on health problem

The Oslo Sports Trauma Research Center questionnaire on overuse injury and health problem developed by Clarsen Etal was translated and adapted to the Thai culture. These questionnaires are tools for injury and illness registration with four essential questions used to assess the severity. Scores in each symptom range from 0 to 100. The range of values in each question is from 0 to 25, with 0 representing no problem and 25 representing the maximum problem level in each question. As a result, questions 1 and 4 have a score of 0-8-17-25, and questions 2 and 3 have a score of 0-6-13-19-25. [6]

2.2.5 Translation and adaptation of Oslo Sports Trauma Research Center into Thai language

Based on the criteria for the process of cross-cultural adaptation of self-report measures, the questionnaire's translation and modification was carried out. This process consists of five stages consists of:

Stage 1 – Forward translation: Two independent translators consist of a knowledge-based translator (T1) and a general translator (T2). They performed a language with setting Thai as their native language, translated the Oslo Sports Trauma Research Center (OSTRC) English questionnaire into Thai language.

Stage 2 – Synthesis of the forward translation: The two translations (T1 and T2) and the researcher combined the results of both translations.

Stage 3 – Back translation: Two additional translators who were proficient in both English and Thai translated the Thai-translated version (T12) back into English (BT1 and BT2).

Stage 4 – Expert committee: The expert committee consisted of the researcher, health professionals, athletes, and the translators. (Forward and backward translators).

Stage 5 – Test of the pre-final version: In this stage, 15 athletes who had the same characteristics as the participants of the study were included to test the pre-final version of injury surveillance. [2]

2.2.6 Sports injuries and illnesses medical record form during Olympic Games 2012

The Olympic Movement Medical Code provided an importance for all athletes who joined the Olympic Games 2012 at London. They advocated all stakeholders to have practicing without any danger from injuries and illnesses, especially to athletes. For analyzing the athletes' performance, they performed a medical record both injury and illness for all staff in order to record a symptom, diagnostic, as well as state of mind that could be affected the athletes. The National Olympic Committee (NOC) staff, London Organizing Committee of the Olympic and Paralympic Games' (LOCOG) medical staff were selected in order to test and evaluate this methodology. In the results, 10,568 athletes participated, which 11% and 7% of athletes were engaged at least one injury or illness respectively. However, an event of injury and illness could occur in variety of sports type that each sport type has a different of danger. Therefore, the medical record must be accompanied by a safe practice session in order to decrease both injuries and illness happening. [7]

CHAPTER 3
ANALYSIS AND DESIGN

This chapter included the analysis and design of our system which contained the system architecture and structure chart that explained the structure and process that happened in the system. Moreover, the system has included the database analysis consisting of ER diagram, Relational Schema, and File Structure which are explained the database on our system what data have to be kept in our system.

3.1 System Architecture Overview

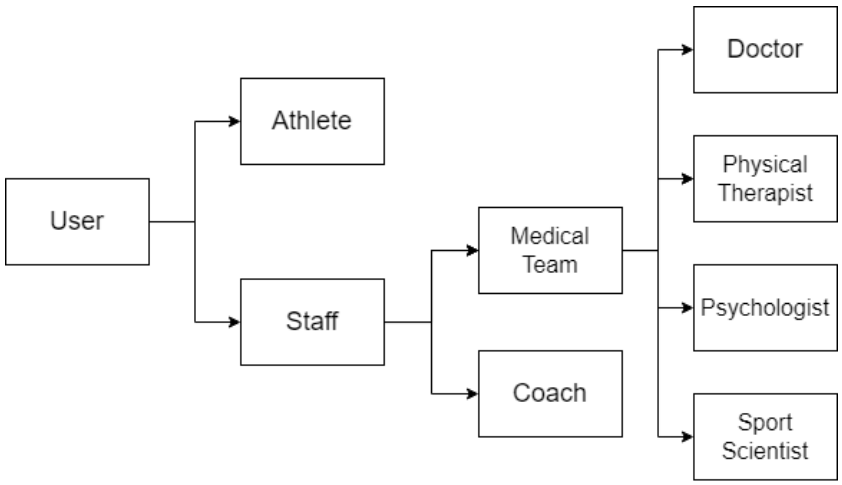


Figure 3.1: User architecture

Figure 3.1 represents the architecture of the user type in Sport Injury and Illness Recording Application (SIRA). Users are separated into main two types including athlete and staff. Staff could break down into two types including coach and medical team which coach can view the overview of athlete and medical team will diagnostic and record for medical document. The medical team is separated into four positions including doctor, physical therapist, psychologist, and sport scientist.

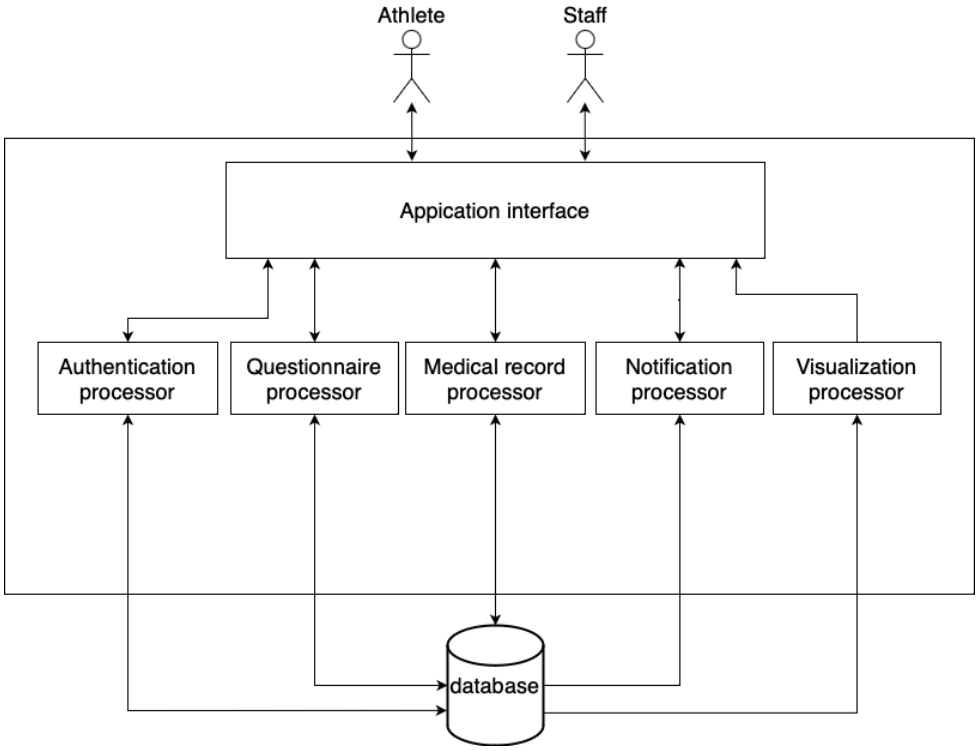


Figure 3.2: SIRA system architecture

Figure 3.2 shows the overviews of our system architecture (Sport Injury and Illness Recording Application). There are two types of users, they are athletes and staff (staff). The athletes can fill out a weekly injury or illness check questionnaire and submit data to firebase for storage and send to the staff to track and view illnesses or injuries outside of athletes from training or competition. The staff can not only track and view athlete symptoms, but also create a record to save diagnoses of athlete symptoms in the firebase for decision-making by the organization. The system includes an application interface, questionnaire processor, authentication processor, medical record processor, and notification processor. The application interface can be used in phones with IOS and Android operating system at this time.

Authentication processor is to manage the account that make the users can login and if users don’t have an account, users can register in application. When a user has account, user can login and use application in his mobile phone.

A questionnaire processor is accessible by athlete, which can be chosen from the application interface and the scores are calculated according to the Oslo Sports Trauma Research Center (OSTRC) and sent to firebase.

Medical record processor is the process of doing record by staff that records have two types, there are illness and injury record. If they do the record and have written advice based on athlete injuries or illness, it can send the message to that athlete in notification navigation bar. When the staff does record finish, it saves to the firebase.

Notification processor is the process of notifying messages to users, which each user is different. If the user is staff department, it can notify the cases that athletes send. If the user is athlete department, it can notify the message that send from staff.

Visualization processor is the process of visualization and summarization the data for athletes and staff. If user is athlete, Athlete can see the scatter plot table that show the overall of the scores from the questionnaires that athlete does. If user is a staff, User can see the table that show the overall of the cases that visualize the number of the case in that week and show the list of the athletes who have the injuries and illnesses in that week.

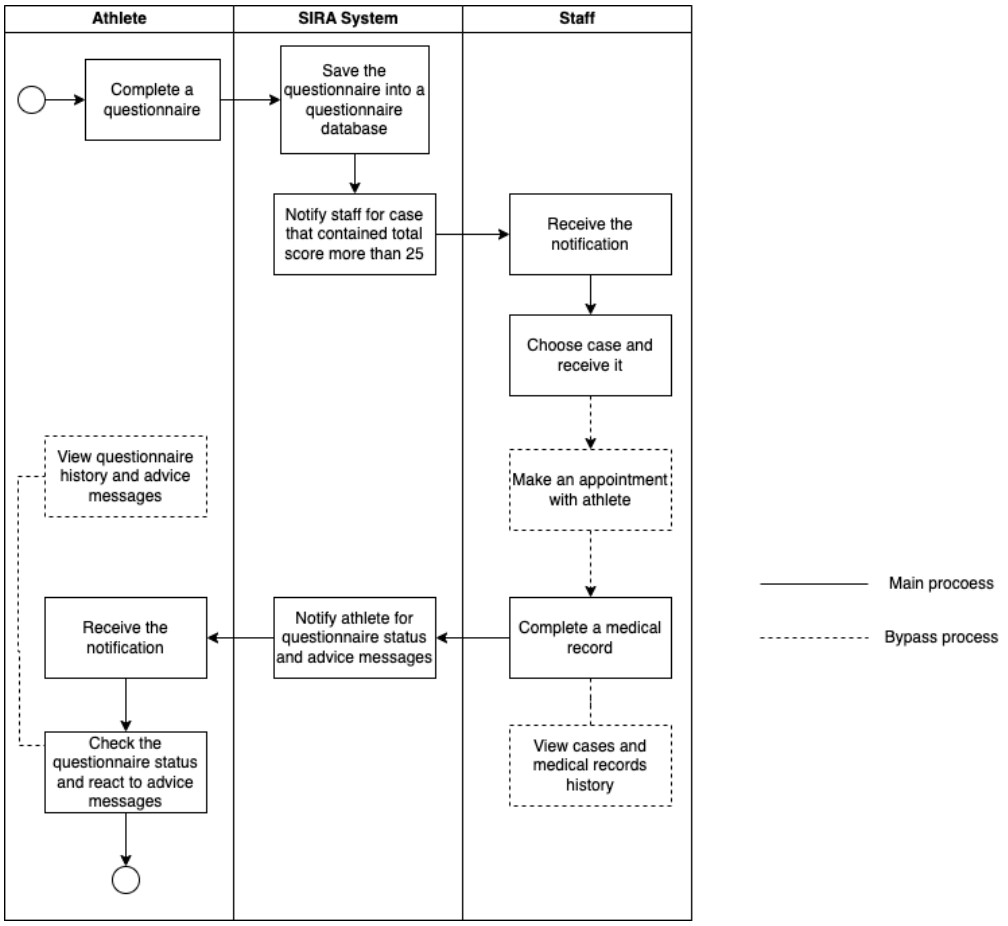
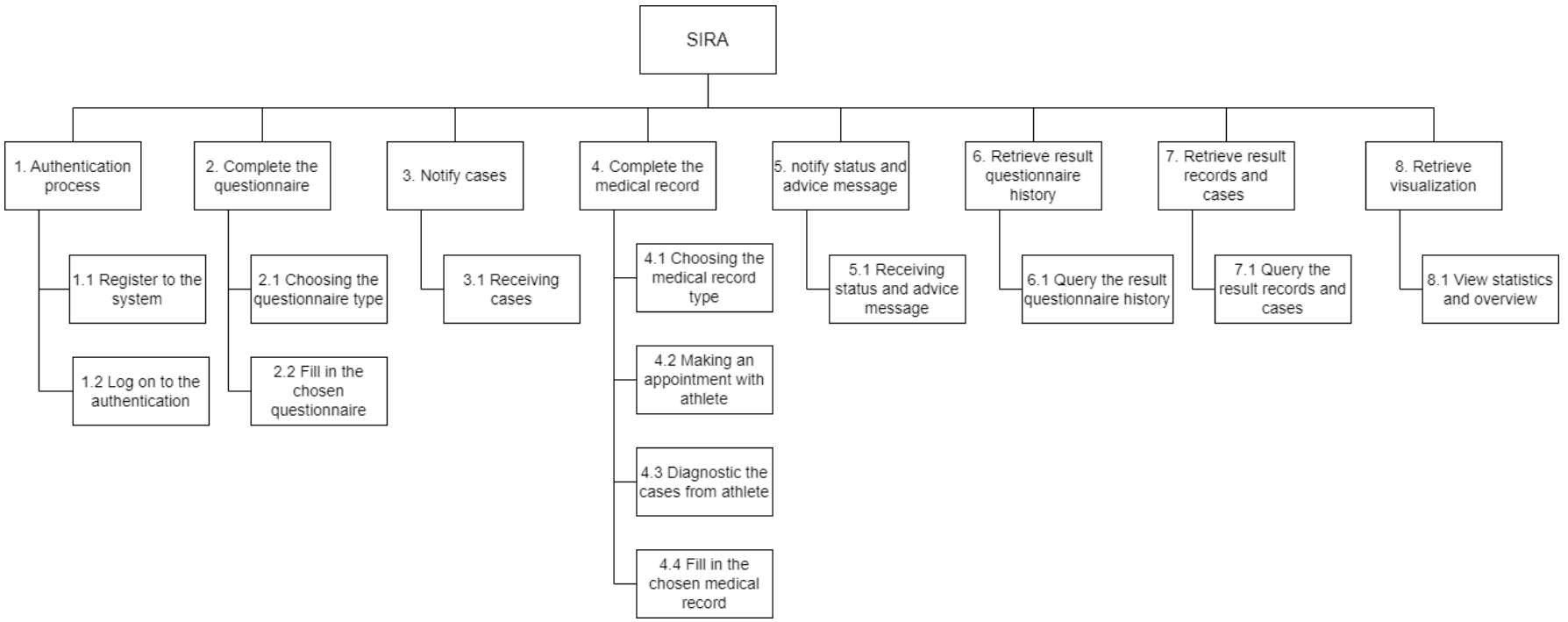


Figure 3.3: SIRA system workflow

Figure 3.3 represents the workflow of all processes that specify whether athletes, staff as well as the SIRA systems. The process starts from completion of questionnaire from athletes in order to save a result into the database (Firebase Firestore). After data being saved into the database, the system will send a notification of which questionnaire contains a total score over 25 points to staff as a case. Then, staff will see the notification and receive whatever they want. Next, staff are required to complete a medical record regarding the received cases, and they can make an appointment with athletes in order to receive more diagnostic information. After they finish the medical record, the system will notify the questionnaire status and advice messages for athletes from the staff who received. Then, athletes will receive the notification and they need to check the questionnaire status and react to advice messages for healing themselves. In addition, staff can view the case and medical record history. Also, athletes can view the questionnaire history and advice messages.

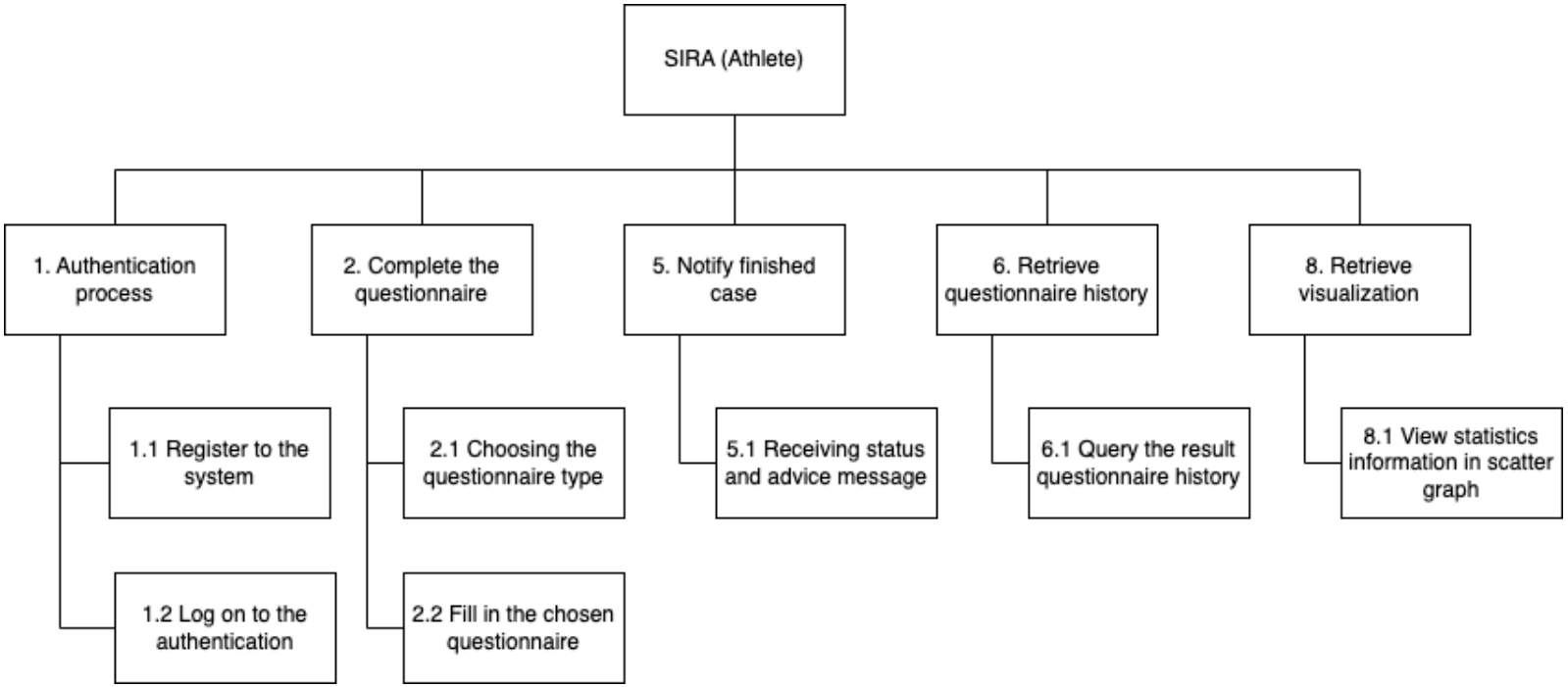
3.2 System Structure Chart

We produced the system structure chart, which illustrated the process all users would urge in terms of a diagram, to demonstrate how the SIRA system would be processed. The essential procedures for the recording system were analyzed and organized in the system structure chart. Additionally, a graphic and description have been used to explain the data that was needed to carry out the process.



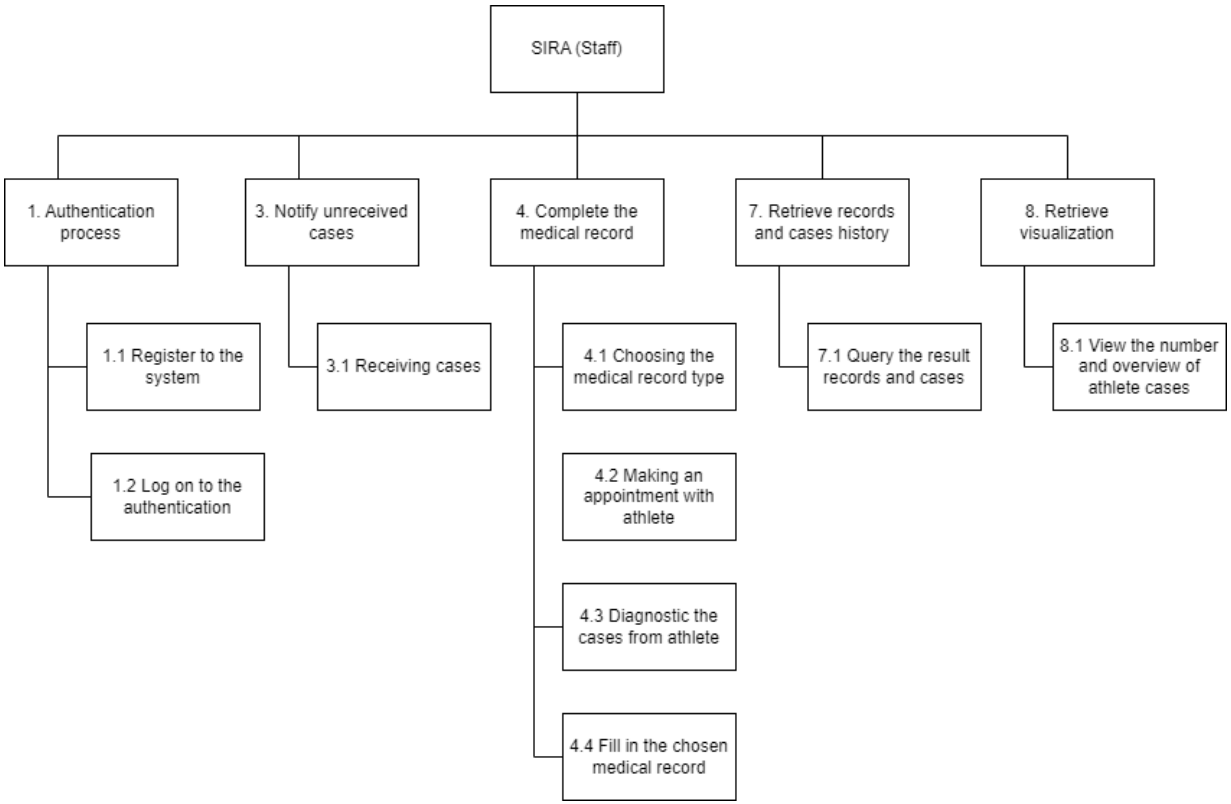
Project : Sport Injury and Illness Recording Application **System :** Recording system	**Major Advisor :** Asst. Prof. Dr. Charnyote Pluempitiwiriyawej
Description : This diagram represents the structure chart of main processes that athlete has to work as a procedure.	

Figure 3.4: Structure chart of SIRA system



Project : Sport Injury and Illness Recording Application **System :** Recording system	**Major Advisor :** Asst. Prof. Dr. Charnyote Pluempitiwiriyawej
Description : This diagram represents the structure chart of main processes that athlete has to work as a procedure.	

Figure 3.5: Structure chart of SIRA system (Athlete)



Project : Sport Injury and Illness Recording Application **System :** Recording system	**Major Advisor :** Asst. Prof. Dr. Charnyote Pluempitiwiriyawej
Description : This diagram represents the structure chart of main processes that staff has to work as a procedure.	

Figure 3.6: Structure chart of SIRA system (Staff)

The subsystem is divided into 2 divisions, namely athletes and staff, which have the same system:

1. **Authentication process** – Users must authenticate to our system both registration and log in in order to access the interactive interfaces.

    1.1. Register to the system – Fill in the personal information, username, and updated password to register to the system.

    1.2. Log on to the authentication – Fill in the email and password to verify the authentication part and access the interfaces.

Each subsystem section, a detailed description of each subsystem is shown below:

Athlete:

2. **Complete the questionnaire** – Athletes choose the questionnaire type, then complete whether questionnaire and complain.

    2.1. Choosing the questionnaire type – Athletes choose the questionnaire type including health questionnaire, physical complaint, or mental questionnaire.

    2.2. Fill in the chosen questionnaire – Fill in or choose an answer to complete the questionnaire.

5. **Notify finished case and appointment** – The system will send the notification of status, advice message, and appointment from staff who received their case.

    5.1 Receiving finished case and appointment – Athletes will receive the notification from staff to verify their questionnaire status and know how to react to their injury or illness and receive the appointment to check the physical condition further from the staff.

6. **Retrieve result questionnaire history** – Athletes can view their questionnaire history to check the past cases.

    6.1 Query the result questionnaire history – Athletes query the desired questionnaire by given options or default settings.

8. **Retrieve visualization** – Athletes can view an overview of their questionnaire scores for each category.

    8.1 Show scores in graph format – Athlete can see the overall as the graph and can select the part that they want to see specifically as well.

Staff:

3. **Notify unfinished cases** – Notification for staff to notify the case from athlete's questionnaire after it is finished.

    3.1 Receiving unfinished cases – Staff receive notification from the system.

4. **Complete the medical record** – Staff must fill in the medical record from case diagnostic.

    4.1 Choosing the medical record type – Staff choose the appropriate medical record type for received cases.

    4.2 Making an appointment with athletes – Staff appoint the athletes to check their health when staff want more information to do record.

    4.3 Diagnostic the cases from athlete – Diagnostic whether health, body injury, and mental problem from athletes' cases

    4.4 Fill in the chosen medical record – Staff complete the chosen medical record from the diagnostic result.

7. **Retrieve result records and cases** – Staff can view their medical record history as well as the cases they received.

    7.1 Query the result records and cases – Staff query the desired record or cases by given options or default setting.

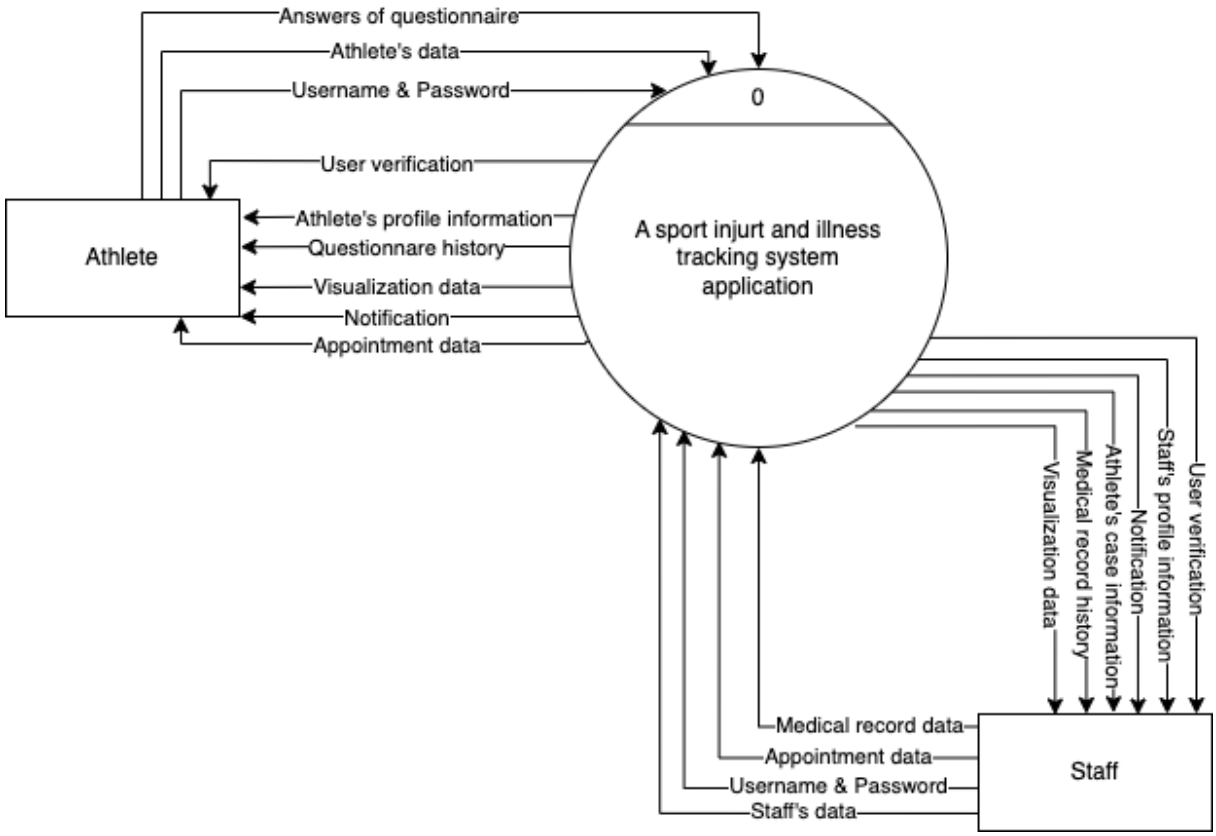
8. **Retrieve visualization** – Staff can see the overview of the number of cases that show both injury and illness.

    8.1 Show the number and overview of athlete cases – Staff can see the overview as a table and show a list of athletes in that week. Moreover, staff can select the information that they want to see specifically such as the range of dates.

3.3 Process Analysis and Design

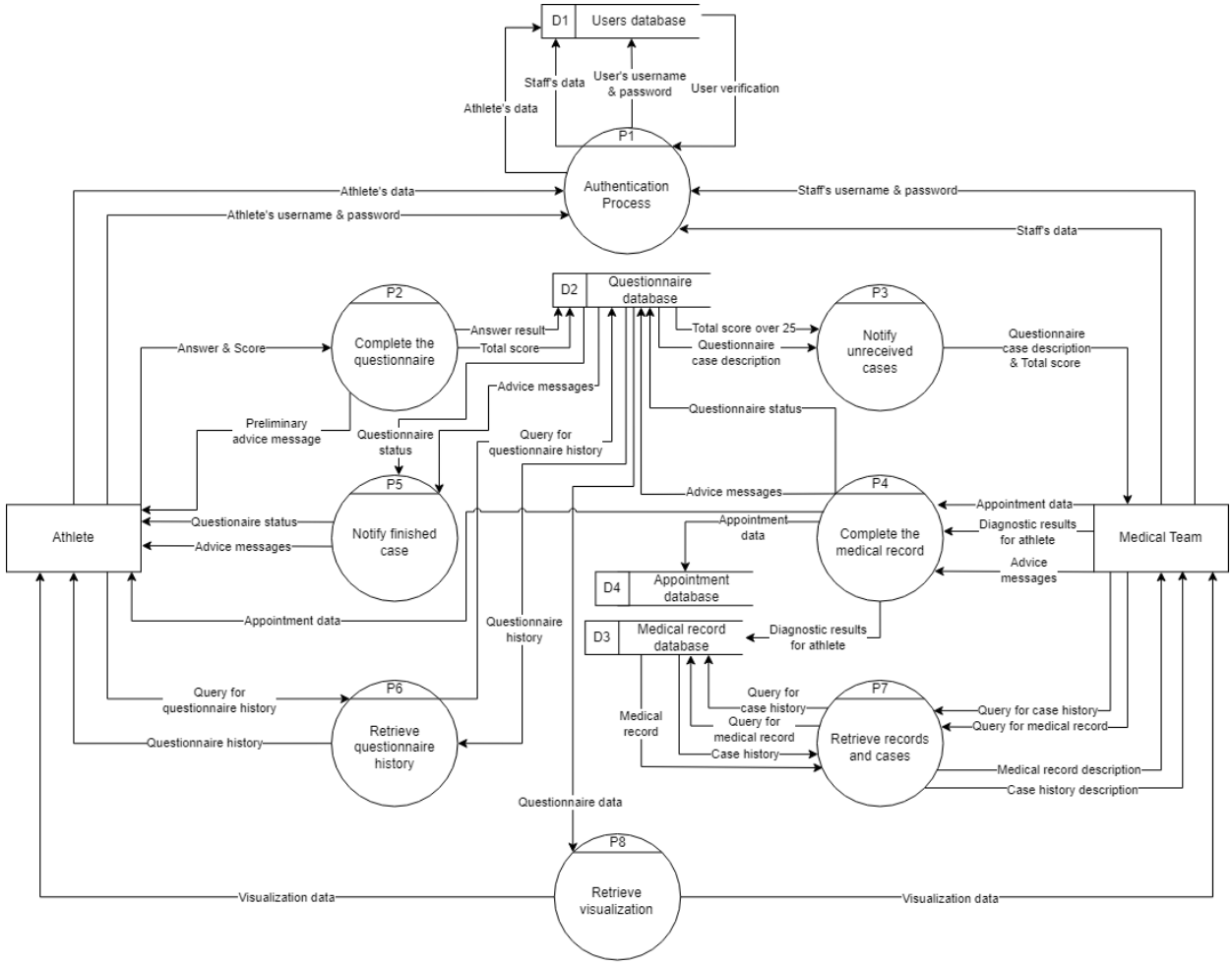
3.3.1 Data Flow Diagram

Our data flow diagram represents the structure and analysis of the processes that can take place in our system and describes the system's flow. The graphic illustrates the processes that our users can carry out and how they create a process in order to produce an output, such as registration, login, completing a questionnaire, and other procedures. The diagram also shows the users who might be the main users, the data that is collected into the database, and the database that is needed for our systems.



Project : Sport Injury and Illness Recording Application **System :** Recording system	**Major Advisor :** Asst. Prof. Dr. Charnyote Pluempitiwiriyawej
Description : This diagram represents how athletes and staff are interacted with the SIRA application.	

Figure 3.7: SIRA Data Flow Diagram Level 0



Project : Sport Injury and Illness Recording Application **System :** Recording system	**Major Advisor :** Asst. Prof. Dr. Charnyote Pluempitiwiriyawej
Description : This diagram deep dives into level 1 of the data flow diagram which makes more clearly in each process through the SIRA application.	

Figure 3.8: SIRA Data Flow Diagram Level 1

3.3.2 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

3.3.2.1 Process Description

This section will provide a 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 process	The process of authentication both register and log in to the system
2	P2	Complete the questionnaire	Fill in the questionnaire in the system
3	P3	Notify unreceived case	Notification for staff to examine the unreceived cases
4	P4	Complete the medical record	Fill in the medical record from a diagnostic result
5	P5	Notify finished case	Notification for athlete to receive the questionnaire status and appointment from staff
6	P6	Retrieve questionnaire history	Query for retrieving the questionnaire result history description
7	P7	Retrieve records and cases history	Query for retrieving the medical result history description and cases history description

No.	Process	Name	Description
8	P8	Retrieve visualization	Both athletes and staff can view a statistic data; athletes can view their overall score; staff can view an overview of athletes

Table 3.2: Process Description of Log on to the application authentication

Process Name	P1 – Authentication process
Description	The process of authentication both register and log in to the system
Inbound data	• Athlete's username and password • Staff's username and password • Athlete's data • Staff's data
Outbound Data	• User's username and password • Athlete's data • Staff's data
Logic Summary	No subsystem

Table 3.3: Process Description of Complete the questionnaire

Process Name	P2- Complete the questionnaire
Description	Fill in the questionnaire in the system
Inbound data	• Answer & Score
Outbound Data	• Answer result • Total score • Preliminary advice message
Logic Summary	No subsystem

Table 3.4: Process Description of Notify unreceived cases

Process Name	P3- Notify unreceived cases
Description	Notification for staff to examine the cases
Inbound data	• Total score over 25 • Questionnaire case description
Outbound Data	• Questionnaire case description and total score
Logic Summary	No subsystem

Table 3.5: Process Description of Complete the medical record

Process Name	P4- Complete the medical record
Description	Fill in the medical record from a diagnostic result
Inbound data	• Diagnostic result for athlete • Advice messages
Outbound Data	• Questionnaire status • Diagnostic results for athlete • Advice messages • Appointment data
Logic Summary	No subsystem

Table 3.6: Process Description of Notify finished case

Process Name	P5- Notify status and appointment
Description	Notification for athlete to receive the questionnaire status and advice messages
Inbound data	• Questionnaire status • Advice messages
Outbound Data	• Questionnaire status • Advice messages • Appointment data
Logic Summary	No subsystem

Table 3.7: Process Description of Retrieve result questionnaire history

Process Name	P6- Retrieve result questionnaire history
Description	Query for retrieving the questionnaire result history description
Inbound data	• Questionnaire history • Query for questionnaire history description
Outbound Data	• Questionnaire history description • Query for questionnaire history description
Logic Summary	No subsystem

Table 3.8: Process Description of Retrieve result records and cases

Process Name	P7- Retrieve result records and cases
Description	Query for retrieving the medical result history description and cases history description
Inbound data	• Query for case history • Query for medical record • Case history • Diagnostic record
Outbound Data	• Medical record description • Case history description • Query for case history • Query for medical record
Logic Summary	No subsystem

Table 3.9: Process Description of Retrieve visualization

Process Name	P8- Retrieve visualization
Description	Query for retrieving the statistical data to be table(staff) or graph(athlete)
Inbound data	• Questionnaire data
Outbound Data	• Visualization data
Logic Summary	No subsystem

3.3.2.2 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 3.10: List of all Data Stores

No.	Data Store	Name	Description
1	D1	Users database	Keeping the staff and athlete data
2	D2	Questionnaire database	Keeping the result of questionnaire from athletes
3	D3	Medical record database	Keeping the result of diagnostic from staff
4	D4	Appointment database	Keeping the appointment that staff want to meet athlete

Table 3.11: Data Store Description of Users database

Data Store Name	D1- Users database
Description	Keeping the staff and athlete data
Inbound data	• Athlete's data • Staff's data • User's username and password
Outbound Data	• User verification

Table 3.12: Data Store Description of Questionnaire database

Data Store Name	D2- Questionnaire database
Description	Keeping the result of questionnaire from athletes
Inbound data	• Answer result • Total score • Questionnaire status • Query for questionnaire history description
Outbound Data	• Total score over 25 • Questionnaire case description • Questionnaire history • Questionnaire status

Table 3.13: Data Store Description of Medical record database

Data Store Name	D3- Medical record database
Description	Keeping the result of diagnostic from staff
Inbound data	• Diagnostic results for athlete • Query for case history • Query for diagnostic record
Outbound Data	• Medical record • Case history

Table 3.14: Data Store Description of Appointment database

Data Store Name	D4- Appointment database
Description	Keeping the appointment that staff want to meet athlete
Inbound data	• Appointment data
Outbound Data	• Appointment data

3.3.2.3 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 3.15: List of All Data Elements

SEQ	Data Element Name	From Process/Source/Data Store	To Process/Source/Data Store
1	Athlete's data	Athlete	P1
2	Athlete's data	P1	D1
3	Staff's data	Medical Team	P1
4	Staff's data	P1	D1
5	Athlete's username & password	Athlete	P1
6	Staff's username & password	Medical Team	P1
7	User's username & password	P1	D1
8	User verification	D1	P1
9	Answer & Score	Athlete	P2
10	Preliminary advice message	P2	Athlete
11	Answer result	P2	D2

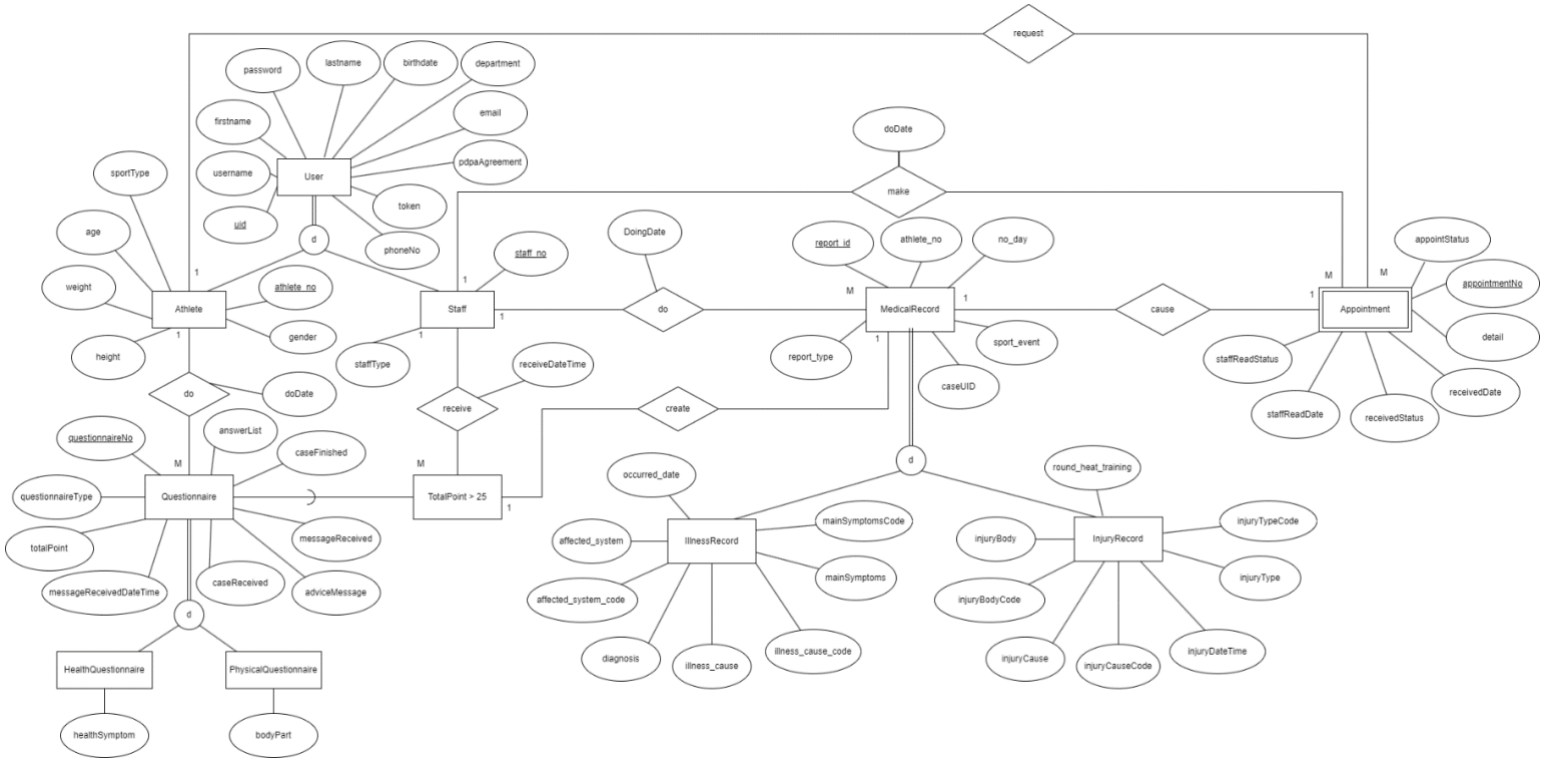
SEQ	Data Element Name	From Process/Source/Data Store	To Process/Source/Data Store
12	Total score	P2	D2
13	Total score over 25	D2	P3
14	Questionnaire case description	D2	P3
15	Questionnaire case description & Total score	P3	Medical Team
16	Appointment data	Medical Team	P4
17	Diagnostic results for athlete	Medical Team	P4
18	Advice messages	Medical Team	P4
19	Questionnaire status	P4	D2
20	Advice messages	P4	D2
21	Appointment data	P4	D4
22	Diagnostic results for athlete	P4	D3
23	Questionnaire status	D2	P5
24	Advice messages	D2	P5
25	Questionnaire status	P5	Athlete
26	Advice messages	P5	Athlete
27	Query for questionnaire history description	Athlete	P6

SEQ	Data Element Name	From Process/Source/Data Store	To Process/Source/Data Store
28	Query for questionnaire history description	P6	D2
29	Questionnaire history	D2	P6
30	Questionnaire history description	P6	Athlete
31	Query for case history	Medical Team	P7
32	Query for diagnostic record	Medical Team	P7
33	Query for case history	P7	D3
34	Query for medical record	P7	D3
35	Case history	D3	P7
36	Diagnostic record	D3	P7
37	Case history description	P7	Medical Team
38	Medical record description	P7	Medical Team
39	Questionnaire data	D2	P8
40	Visualization data	P8	Athlete
41	Visualization data	P8	Medical Team
42	Appointment data	P4	Athlete

3.4 Database Analysis and Design

The systems are created interactive together which contain the entities, relationships, and attributes. Each entity will contain its data as attributes that represent the needed data as well as primary key and foreign key which can be used in SQL. The attribute contains the data that is relevant to them which some data can be null, and some data must be filled in. However, this design and analysis are simply examples but cover all main users and data in our game system.

3.4.1 ER-Diagram



Project: Sport Injury and Illness Recording Application **System:** Recording system	**Major Advisor:** Asst. Prof. Dr. Charnyote Pluempitiwiriyawej
Description: This is ER-Diagram of SIRA including four entities that are superclass. User class has two subclasses including “Athlete” and “Staff”. Questionnaire class has two subclasses including “HealthQuestionnaire” and “PhysicalQuestionnaire”. MedicalRecord class has two subclasses including “IllnessRecord” and “InjuryRecord”. Appointment is weak entity from Medical Record and Staff.	

Figure 3.9: Conceptual ER-Diagram of SIRA Database

3.4.2 Relational Schema

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

- **Attributes** – which are bold and underlined are the Primary Keys
- *Attributes* – which are Italic are the Foreign Keys
- ***Attributes*** – 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 3.16: List of all Tables in Our System Database

Table#	Table Name	Table Type	Description
1	Athlete	Master	It is the subclass that collects the information of athlete.
2	Staff	Master	It is the subclass that collects the information of staff.
3	HealthQuestionnaire	Base	It is the subclass that collects all of health questionnaires that athletes do.
4	PhysicalQuestionnaire	Base	It is the subclass that collects all of physical questionnaires that athletes do.
5	IllnessRecord	Base	It is the subclass that collects all of illness records that staff do.
6	InjuryRecord	Base	It is the subclass that collects all of injury records that staff do.

Table#	Table Name	Table Type	Description
7	Appointment	Base	This table is the weak entity that collects the appointment that occur when the staff want to know more athlete's health body for doing the medical record.

1. Relational Schema of Master File Tables

Athlete

uid	**athlete_no**	username	password	firstname	lastname	sportType	birthdate

department	weight	height	email	age	gender	pdpaAgreement	token

phoneNo

Staff

uid	**staff_no**	username	password	firstname	lastname	stafftType	birthdate

department	email	pdpaAgreement	token	phoneNo

2. Relational Schema of Base File Tables

HealthQuestionnaire

questionnaireNo	***athleteNo***	questionnaireType	totalPoint	caseFinished

caseRecieved	healthSymptom	doDate	***staff_no_received***	Q1	Q2	Q3	Q4

adviceMessage	messageReceived	messageReceivedDateTime

PhysicalQuestionnaire

questionnaireNo	*athleteNo*	questionnaireType	totalPoint	caseFinished

caseRecieved	bodypart	doDate	*staff_no_received*	Q1	Q2	Q3	Q4

adviceMessage	messageReceived	messageReceivedDateTime

IllnessRecord

report_id	*staff_uid*	*athlete_no*	report_type	sport_event	DoingDate	affected_system_code

affected_system	diagnosis	illness_cause	mainSymtoms	mainSymptomsCode	illness_cause_code

occurred_date	no_day	caseUID

InjuryRecord

report_id	*staff_uid*	*athlete_no*	report_type	sport_event	DoingDate	round_heat_training

injuryBody	injuryBodyCode	injuryCause	injuryCauseCode	injuryDateTime	injuryTypeCode

injuryType	no_day	caseUID

Appointment

appointmentNo	*athleteNo*	*staffUID*	*caseID*	appointDate	appointStatus	detail

doDate	receivedDate	receivedStatus	staffReadDate	staffReadStatus

3.4.3 File Structure

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 3.17: File Structure of Athlete

Table Name :	Athlete				**Table#1**	
Table Type :	Master					
Description :	Storing the information of athletes.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
uid	varchar	28	User's ID	PK		NOT
athleteNo	varchar	11	Athlete's ID	PK		NOT
username	varchar	500	Athlete's username			NOT
password	varchar	500	Athlete's password			NOT
firstname	varchar	500	Athlete's firstname			NOT
lastname	varchar	500	Athlete's lastname			NOT
sportType	varchar	500	Athlete's sport type			NOT
birthdate	DATE	100	Athlete's birth date			NOT
department	varchar	500	User's department			NOT
weight	double	11	Athlete's weight			NOT

height	double	11	Athlete’s height			NOT
age	int	2	Athlete’s age			NOT
gender	varchar	6	Athlete’s gender			NOT
email	varchar	500	Athlete’s email			NOT
pdpaAgreement	boolean	5	Confirmation for Personal Data Protection Act			NOT
token	varchar	500	token			NOT
phoneNo	varchar	10	Athlete’s phone number			NOT
	Total	4184	Bytes			

Table 3.18: File Structure of Staff

Table Name :	Staff				**Table#2**
Table Type :	Master				
Description :	Storing the information of staff.				
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
uid	varchar	28	User's ID	PK		NOT
staffNo	varchar	11	Staff's ID	PK		NOT
username	varchar	500	Staff's username			NOT
password	varchar	500	Staff's password			NOT
firstname	varchar	500	Staff's firstname			NOT
lastname	varchar	500	Staff's lastname			NOT
staffType	varchar	500	Staff's type			NOT
birthdate	DATE	100	Staff's birth date			NOT
department	varchar	500	User's department			NOT
email	varchar	500	Staff's email			NOT
pdpaAgreement	boolean	5	Confirmation for Personal Data Protection Act			NOT

token	varchar	500	token			NOT
phoneNo	varchar	10	Athlete’s phone number			NOT
	Total	4184	Bytes			

Table 3.19: File Structure of Health questionnaire

Table Name :	HealthQuestionnaire				**Table#3**	
Table Type :	Base					
Description :	Storing the athlete's health questionnaire answers and calculate the score					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
questionnaireNo	varchar	12	Questionnaire's ID	PK		NOT
athleteNo	varchar	28	Athlete's user ID	FK	Athlete	NOT
questionnaireType	varchar	500	Questionnaire's type			NOT
totalPoint	int	3	Total of points			NOT
caseFinished	varchar	500	Check this case finish or not			NOT
caseReceived	varchar	500	Check this case receive or not			NOT
healthSymptom	varchar	500	Athlete's health symptom			NOT
doDate	DATETIME	100	Questionnaire's do date and time			NOT
staff_no_received	varchar	500	Staff's user ID who receives the athlete's case	FK	Staff	NOT
Q1	int	11	Question 1 point			NOT

Q2	int	11	Question 2 point			NOT
Q3	int	11	Question 3 point			NOT
Q4	int	11	Question 4 point			NOT
adviceMessage	varchar	500	Staff's advice message			NOT
messageReceived	boolean	5	Confirmation message receiving			NOT
messageReceivedDateTime	DATETIME	100	Message receiving date and time			NOT
	Total	3292	Bytes			

Table 3.20: File Structure of Physical questionnaire

Table Name :	PhysicalQuestionnaire				**Table#4**	
Table Type :	Base					
Description :	Storing the athlete's physical questionnaire answers and calculate the score					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
questionnaireNo	varchar	12	Questionnaire's ID	PK		NOT
athleteNo	varchar	28	Athlete's user ID	FK	Athlete	NOT
questionnaireType	varchar	500	Questionnaire's type			NOT
totalPoint	int	3	Total of points			NOT
caseFinished	varchar	500	Check this case finish or not			NOT
caseReceived	varchar	500	Check this case receive or not			NOT
bodyPart	varchar	500	Athlete's body part			NOT
doDate	DATETIME	100	Questionnaire's do date and time			NOT
staff_no_received	varchar	500	Staff's user ID who receives the athlete's case	FK	Staff	NOT
Q1	int	11	Question 1 point			NOT

Q2	int	11	Question 2 point			NOT
Q3	int	11	Question 3 point			NOT
Q4	int	11	Question 4 point			NOT
adviceMessage	varchar	500	Staff's advice message			NOT
messageReceived	boolean	5	Confirmation message receiving			NOT
messageReceivedDateTime	DATETIME	100	Message receiving date and time			NOT
	Total	3292	Bytes			

Table 3.21: File Structure of Illness record

Table Name :	IllnessRecord				**Table#5**	
Table Type :	Base					
Description :	Storing the illness record that staff do.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
report_id	varchar	12	Report's ID	PK		NOT
staff_uid	varchar	28	Staff's user ID	FK	Staff	NOT
athlete_no	varchar	11	Athlete's ID			NOT
report_type	varchar	500	Report's type			NOT
sport_event	varchar	500	Sport that causes illness in the event			NOT
DoingDate	varchar	500	Staff's do date			NOT
affected_system	varchar	500	Affected system that happen to athlete			NOT
affected_system_code	int	2	Affected system code			NOT
diagnosis	varchar	500	The identification of the nature of an illness or other problem by examination			NOT
illness_cause	varchar	500	The cause of illness			NOT

illness_cause_code	int	2	Illness's cause code			NOT
mainSymptoms	varchar	500	Main Symptoms diagnosed			NOT
mainSymptomsCode	int	2	Main Symptoms code			NOT
occurred_date	DATETIME	100	The date and time that occur illness			NOT
no_day	varchar	100	The day to rest or stop training			NOT
caseUID	varchar	28	Case's ID			NOT
	Total	3785	Bytes			

Table 3.22: File Structure of Injury record

Table Name :	InjuryRecord				**Table#6**	
Table Type :	Base					
Description :	Storing the injury record that staff do.					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
report_id	varchar	12	Report's ID	PK		NOT
staff_uid	varchar	28	Staff's user ID	FK	Staff	NOT
athlete_no	varchar	11	Athlete's ID			NOT
report_type	varchar	500	Report's type			NOT
sport_event	varchar	500	Sport that causes injury in the event			NOT
DoingDate	varchar	500	Staff's do date			NOT
injuryBody	varchar	500	The athlete's body part that injuries			NOT
injuryBodyCode	int	2	Injury Body code			NOT
round_heat_training	varchar	500	Round, heat or training during an injury			NOT
injuryCause	varchar	500	The cause of injury			NOT

injuryCauseCode	int	2	Injury’s cause code			NOT
injuryDateTime	DATETIME	100	Injury’s date and time			NOT
injuryType	varchar	500	Injury’s type			NOT
injuryTypeCode	int	2	Injury’s type code			NOT
no_day	varchar	100	The day to rest or stop training			NOT
caseUID	varchar	28	Case’s ID			NOT
	Total	3785	Bytes			

Table 3.23: File Structure of Appointment

Table Name :	Appointment				**Table#7**	
Table Type :	Base					
Description :	Storing the appointments that occur when the staff want to know more athlete's health body for doing the medical record..					
Field Name	**Type**	**Length**	**Description**	**Key**	**Reference**	**Null**
appointmentNo	varchar	12	Appointment's ID	PK		NOT
athleteUID	varchar	28	Athlete's User ID	FK	Athlete	NOT
staffUID	varchar	28	Staff's User ID	FK	Staff	NOT
caseID	varchar	28	Medical record's ID	FK	MedicalRecord	NOT
appointDate	DATETIME	100	Appointment's date and time			NOT
appointStatus	boolean	5	Confirmation appointment status			NOT
detail	varchar	500	Appointment's detail			NOT
doDate	DATETIME	100	Appointment's doing date and time			NOT
receivedDate	DATETIME	100	Athlete's received appointment date and time			NOT

receivedStatus	boolean	5	Confirmation athlete's received appointment			NOT
staffReadDate	DATETIME	100	Staff's reading appointment date and time			NOT
staffReadStatus	boolean	5	Confirmation Staff's reading appointment			NOT
	Total	1011	Bytes			

3.5 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.

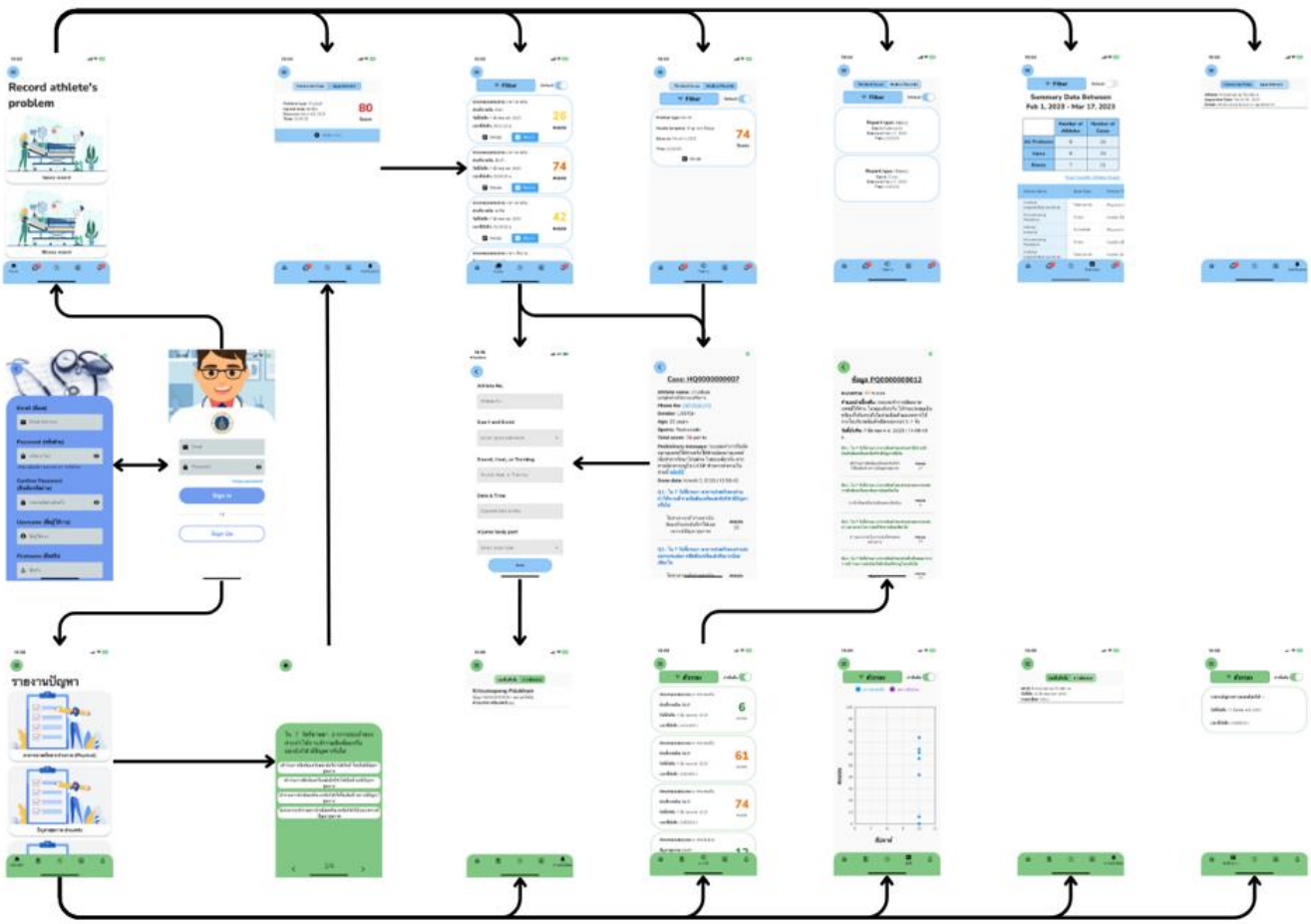


Figure 3.10: User interface workflow

Figure 3.10 represents the user interface workflow of major interfaces in SIRA. The description of each page would be provided as follows:

3.5.1 Interface Design

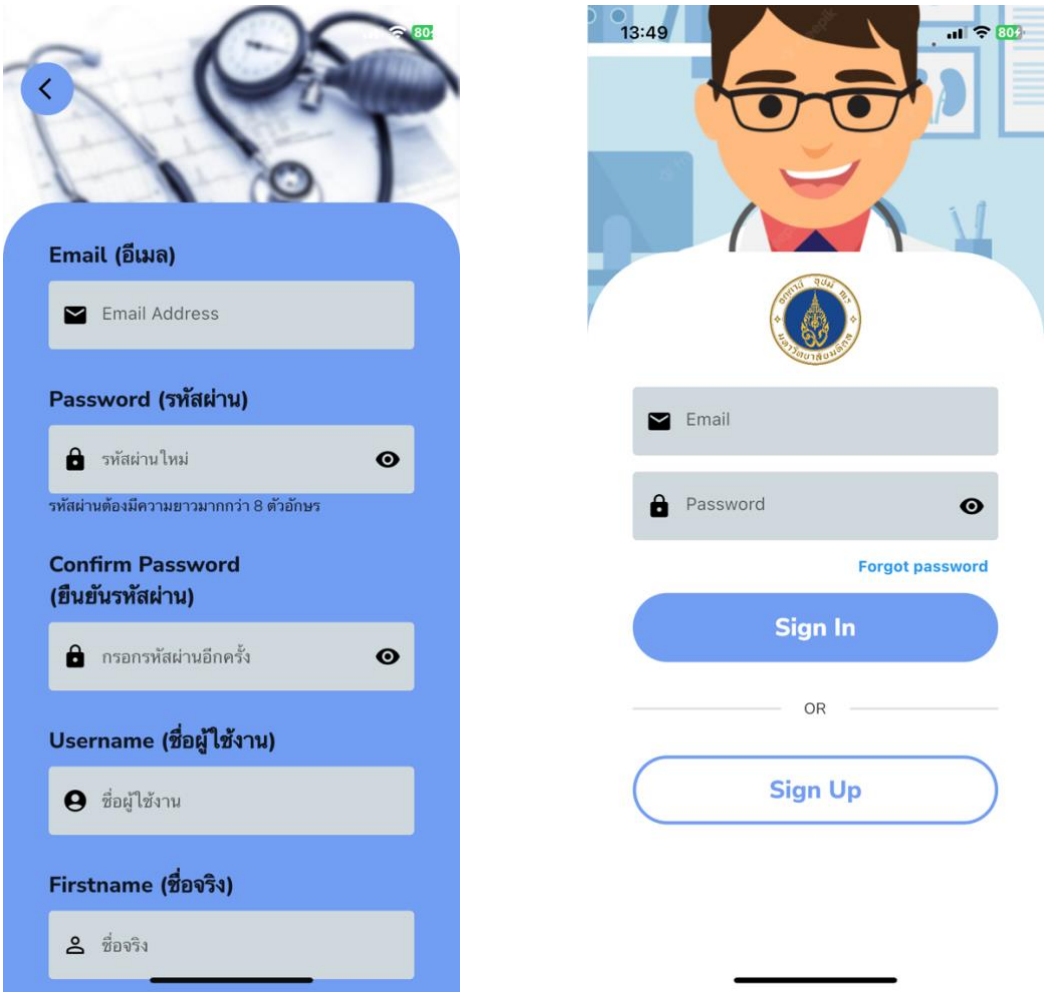


Figure 3.11: Register and Login

According to Figure 3.11, users must sign up via registration page and sign in via log in page in order to get a verification and use of SIRA features.

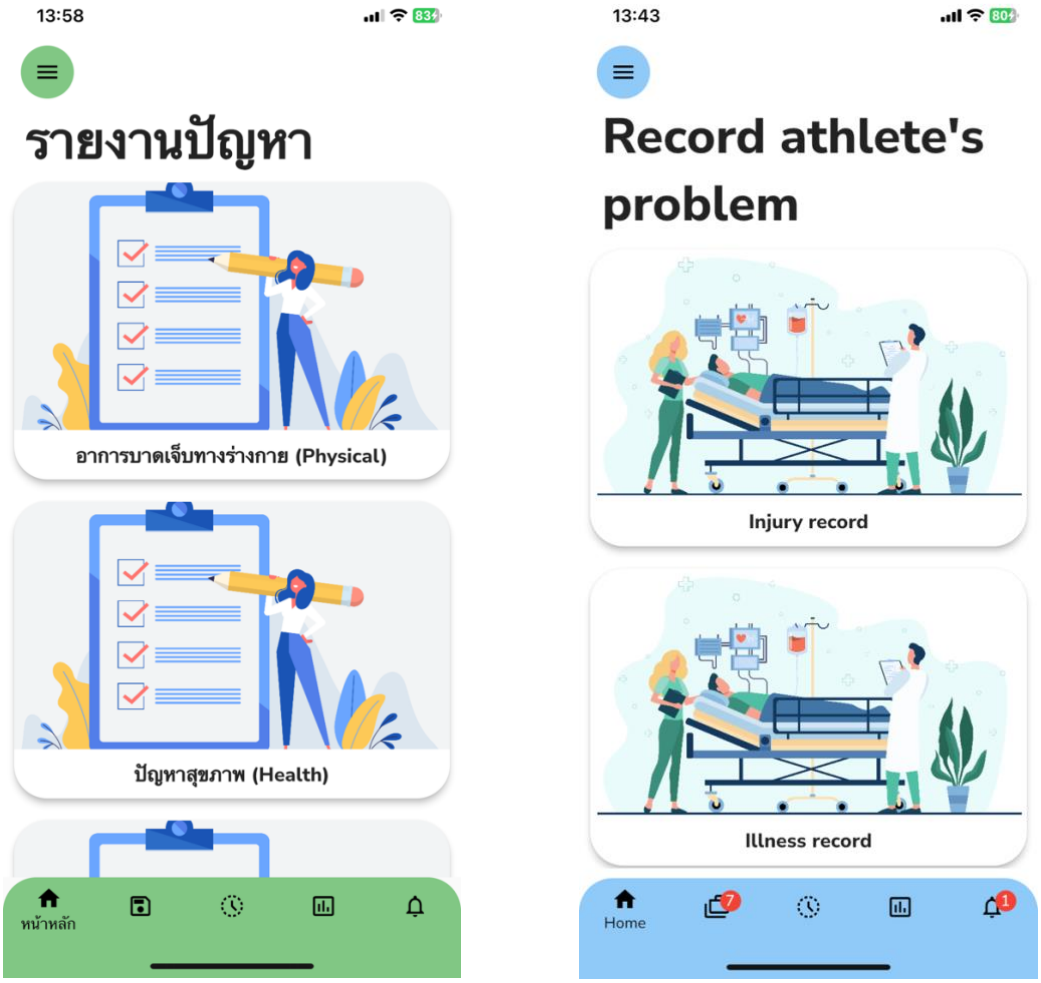


Figure 3.12: Athlete/Staff Homepage

According to Figure 3.12, both interfaces contain athlete homepage (left picture) and staff homepage (right picture). When user's login successfully, user can go to homepage first. If you are athlete, your homepage has the green bottom bar including button of homepage, mental questionnaire history page, health and physical questionnaire history page, statistics page, and notification page. If you are staff, your homepage has the blue bottom bar including button of homepage, received case page, history page, statistics page, and notification page.

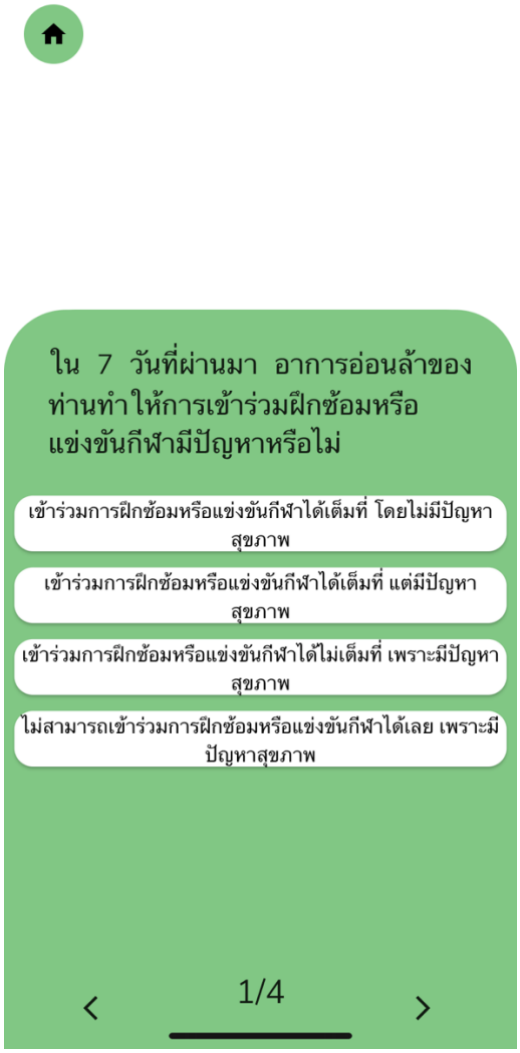


Figure 3.13: Questionnaire page

According to Figure 3.13, athletes have the three questionnaire buttons in their homepage that are health questionnaire, physical complain, and mental questionnaire.

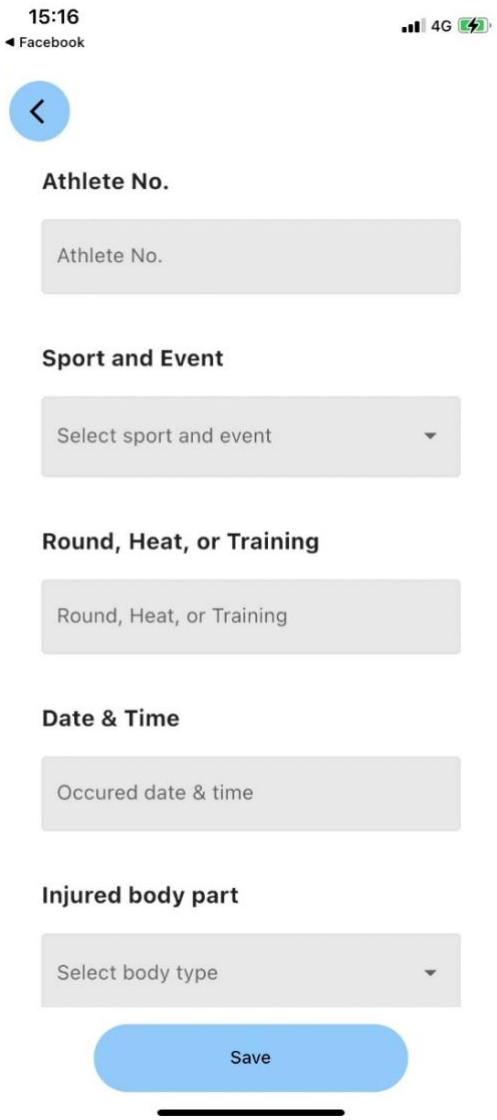


Figure 3.14: Recording page

According to Figure 3.14, athletes have the two recording buttons in their homepage that are illness and injury record.

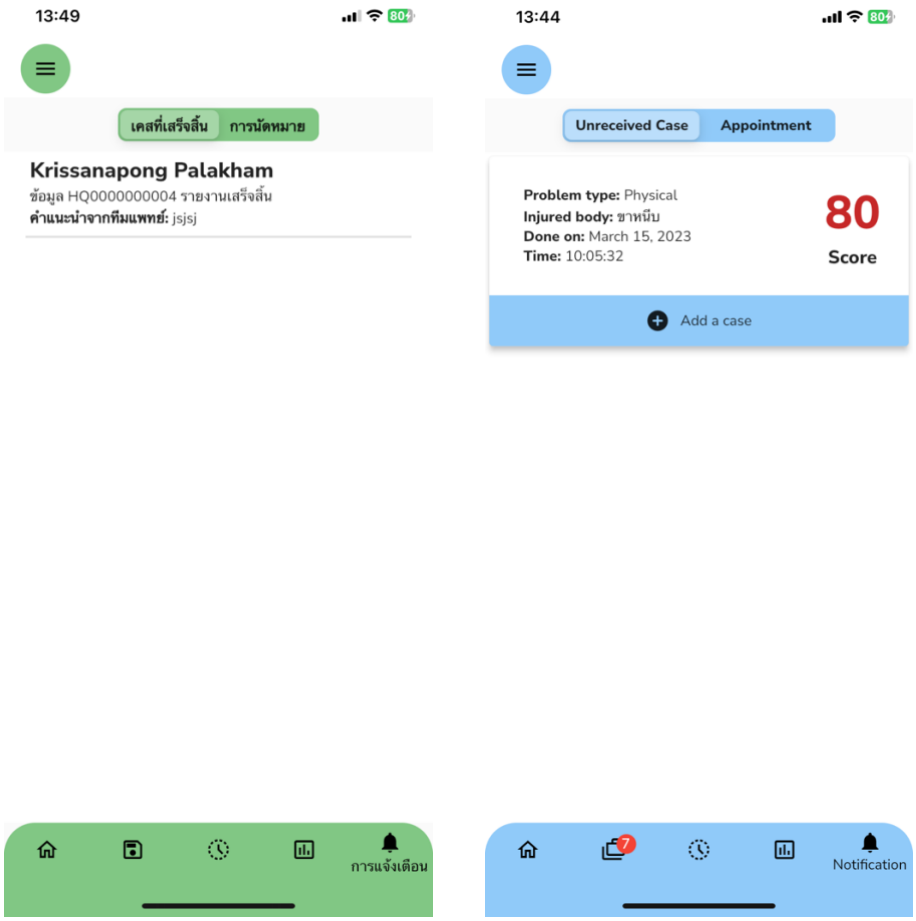


Figure 3.15: Notification page (For case)

According to Figure 3.15, both interfaces represent the athlete notification page (left picture) and the staff notification page (right page). For notification page, Athletes and staff can see the last navigation bar button in homepage representing as a bell button. The bell button properties of both parties are different as follows:

- **Athlete's notification (Left picture)**

  It notifies a finished case including staff name, and preliminary advice from staff who received their case.

- **Staff's notification (Right picture)**

  It notifies the unfinished questionnaire from athletes. Staff can decide to diagnostic a case and click "add a case" button in order to receive cases. Staff can view the preliminary information including problem type, symptom or injured body, date and time when the case have done, and total score.

Figure 3.16: Athlete's history page

According to Figure 3.16, it shows the history of the questionnaires that athletes do it. It shows the questionnaire score, questionnaire type, health symptom or injured body, and date and time of the questionnaire. After we click a history, it would show the deep detail of the selected history including score, automatic advice, date of saving, and score of each question.

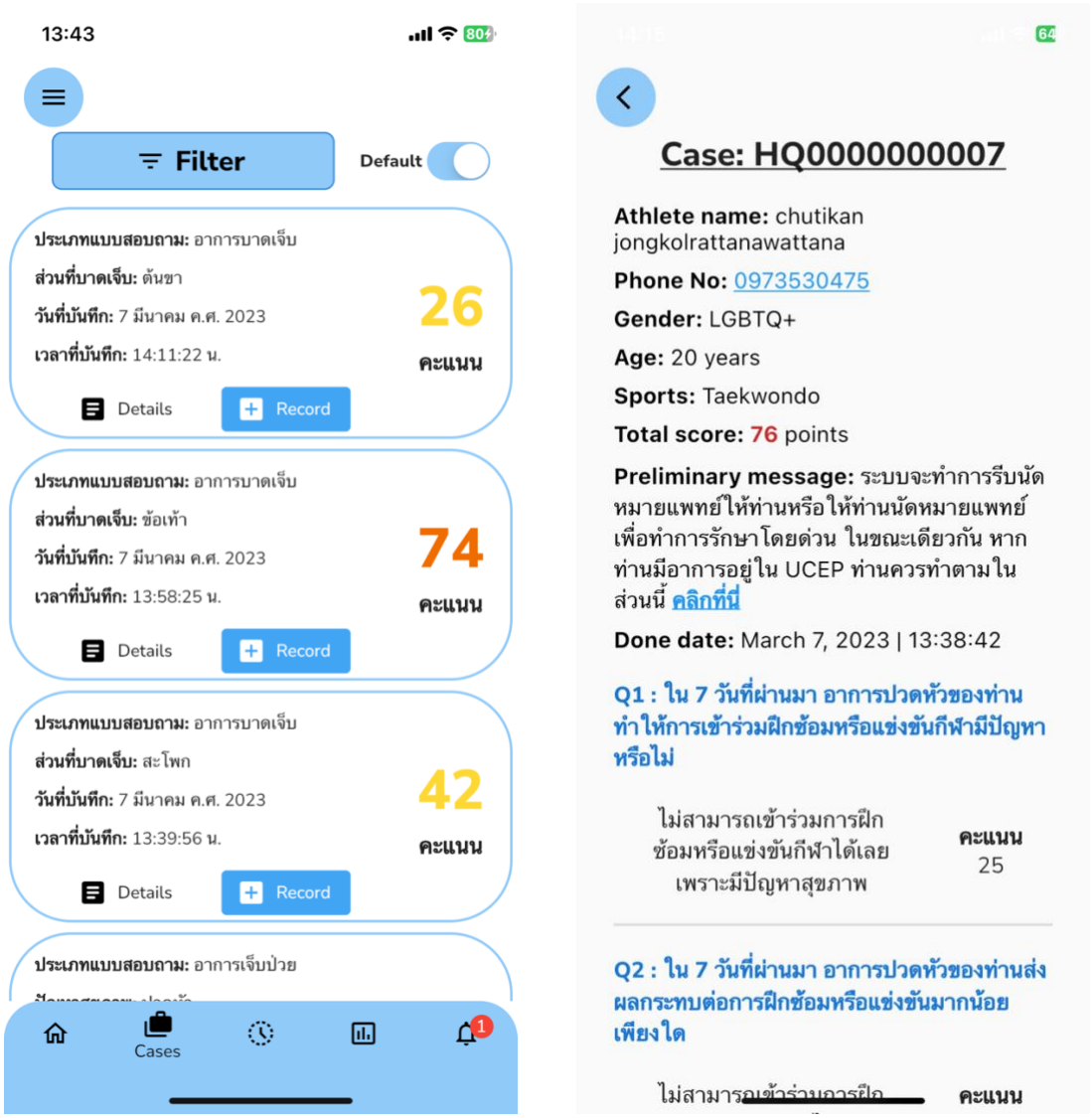


Figure 3.17: Staff's case page

Figure 3.17 shows the received case that represent from athlete questionnaire including type of case, injured body or symptom, date and time of save, and total score. If we click a detail button, it will go to the detail page (right picture). The detail page will show athlete name, phone number, gender, age, sports, total score, automatic advice, date of save, and score of each question.

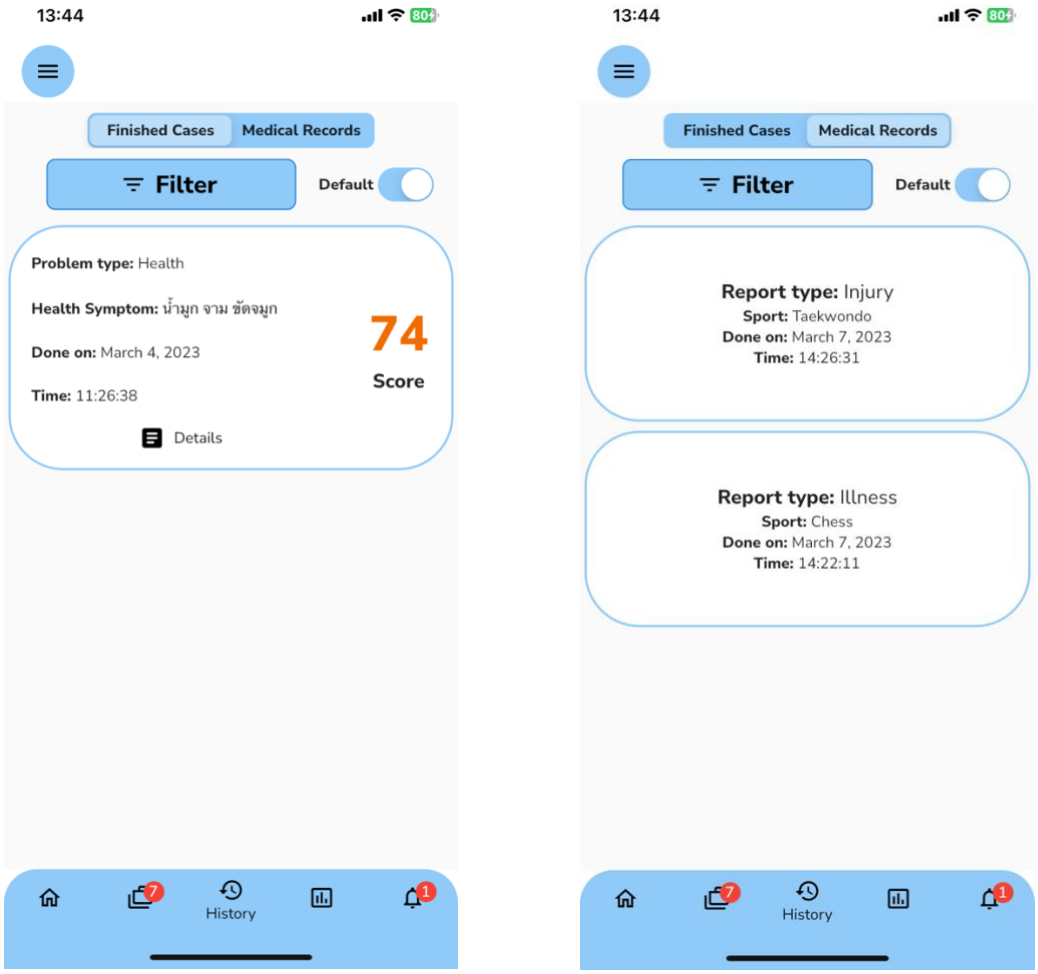


Figure 3.18: Staff's history page

Figure 3.18 shows the history of staff including finished case history and medical record history. The left picture shows the finished case history including problem type, symptom or injured body, date of finishing case, and total score of the case. The right picture shows the medical record history that recorded from illness or injury record. It shows a report type, sports, and date of record.

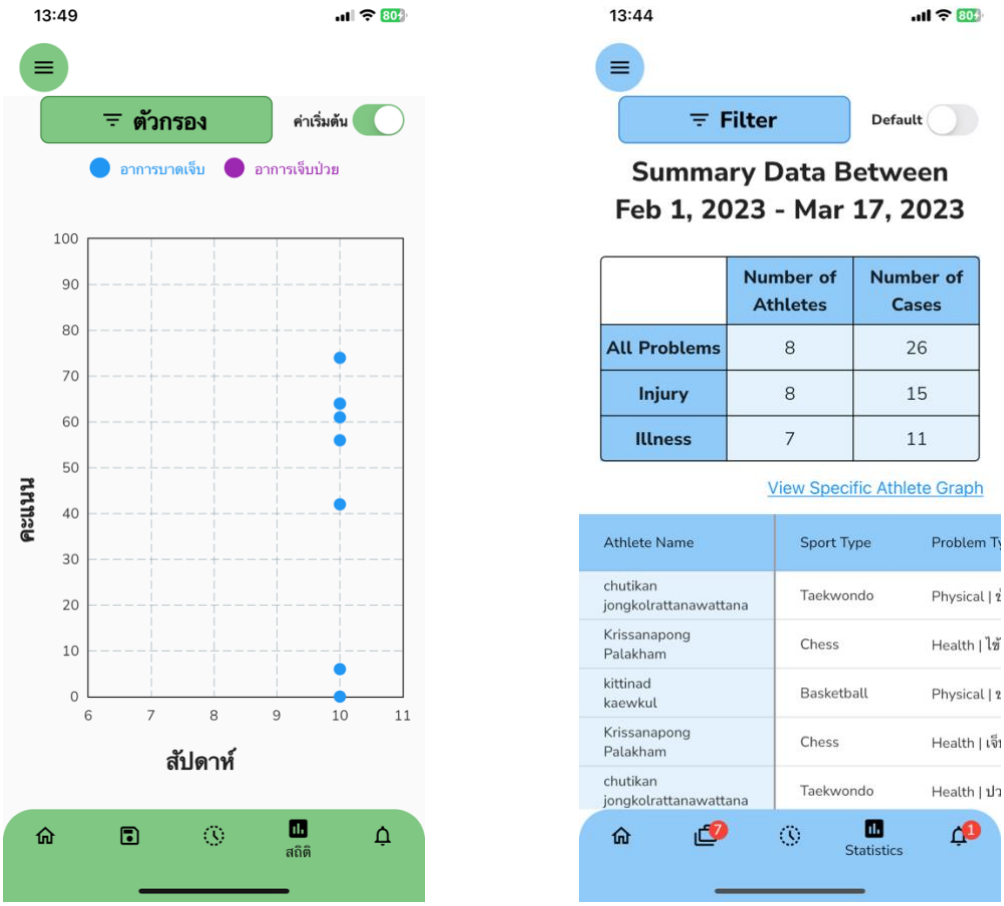


Figure 3.19: Visualization page

Figure 3.19 represents the analysis part including the visualization graph and table. The left picture shows the visualization part of score for each questionnaire as a scatter plot. The blue dot represents score of physical questionnaires and the purple dot represents score of health questionnaires.

The right picture shows the visualization part of staff as a table. There will be represented two table; the above table shows the total case and total athlete of each problem; the below table shows more deep insight for each athlete including sport type, problem type, score, date of questionnaire record.

In addition, it can click on “View Specific Athlete Graph” to show the scatter plot of each athlete in order to get more specific on the selected athlete.

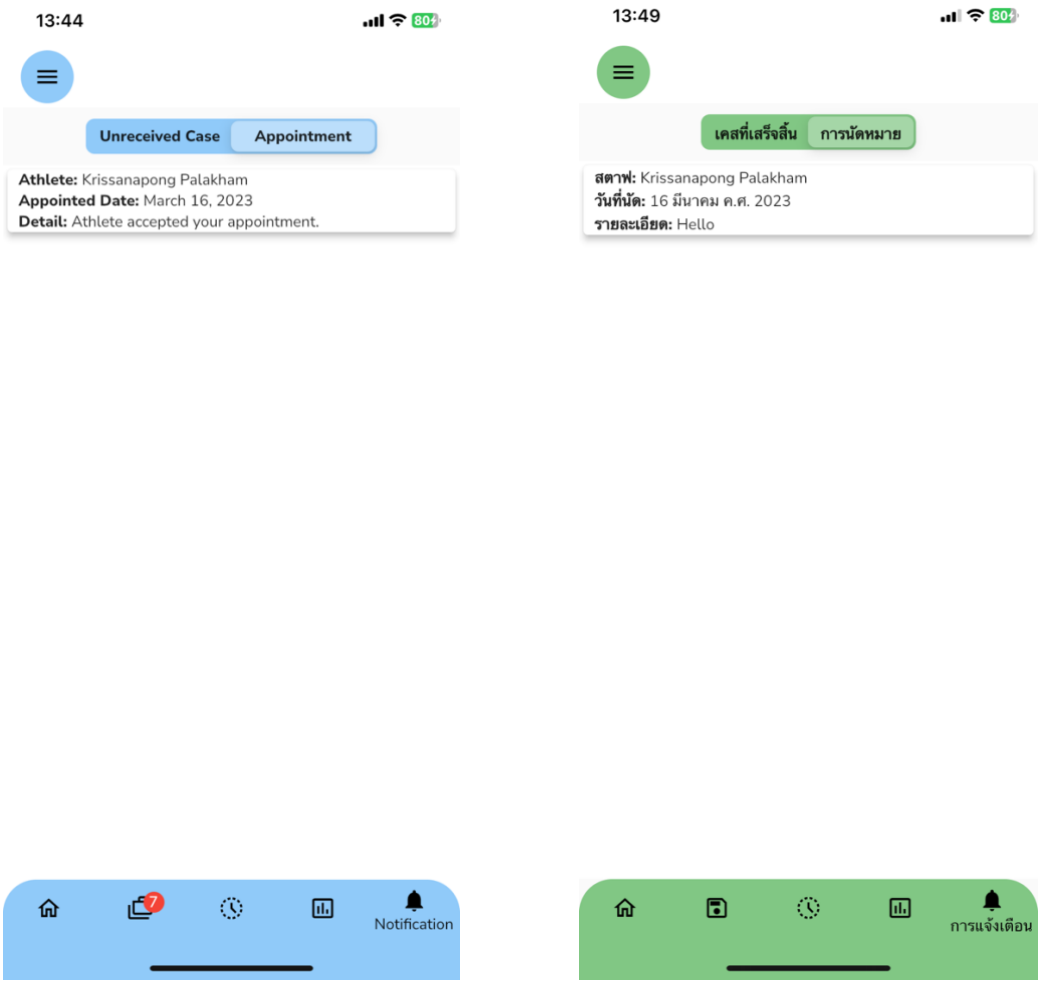


Figure 3.20: Appointment page (Notification)

Figure 3.20 represents an appointment in notification page both staff (left picture) and athlete (right picture). The appointment will notify to athlete when staff finished creating appointment. Athletes required to choose for decline or accept, then the system will notify to staff about athlete's appointment status.

CHAPTER 4
IMPLEMENTATION

This chapter describes the hardware and system environment, as well as the implementation guide and approach employed in this project's coding operations.

4.1 Hardware and System Environment

- Operating System and Utilities Applications
    - iOS Devices
    - Android Devices
    - MacBook Pro (13-inch, 2020)
        - Intel Core i5 8th-generation quad-core 1.4GHz Turbo Boost up to 3.9GHz with 128MB of eDRAM.
        - RAM 8 GB
        - Intel Iris Plus Graphics 645
- Web Browser software
    - Google chrome
        - Canva
        - Draw.io
    - Safari
- Database Management System (DBMS)
    - Google Firebase Firestore
- Programming and Scripting Tools
    - Visual Studio Code
        - Dart
    - Xcode
        - Swift
- Components
    - Email.js
    - Google Firebase Cloud Messaging

4.2 Implementation Guide and Techniques

There are several implementation guides and techniques that have been utilized in this project. The guide starts from creating a project directory in a computer; we linked the project directory to GitHub in order to work with team members comfortably. We write a code “cd <project directory>” in order to open a directory for creating a Flutter folder. We created the Flutter folder by coding “flutter create <project name>” in order to create the first Flutter application. The next step is the SIRA application codes, techniques, and implementation guides, will be provided as follows:

4.2.1 Authentication Page

4.2.1.1 Sign in Function

```
import 'package:firebase_auth/firebase_auth.dart';

await FirebaseAuth.instance
          .signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      )
```

Figure 4.1: Sign in Function

We imported the firebase_auth.dart in order to create the sign in function to the Firebase Authentication Service. Figure 4.1 represents the Firebase function in order to make users sign into Firebase Authentication Service by email and password.

```
import 'package:cloud_firestore/cloud_firestore.dart';


            String uid = value.user.uid;
        final db = FirebaseFirestore.instance;
        DocumentReference athleteRef = db.collection('Athlete').doc(uid);
        DocumentReference staffRef = db.collection('Staff').doc(uid);
        DocumentSnapshot athleteDoc = await athleteRef.get();
        DocumentSnapshot staffDoc = await staffRef.get();
        if (athleteDoc.exists) {
          Navigator.of(context).pushNamedAndRemoveUntil(
              '/athletePageChoosing', (route) => false);
        } else if (staffDoc.exists) {
          Navigator.of(context).pushNamedAndRemoveUntil(
              '/staffPageChoosing', (route) => false);
        } else {
          Navigator.of(context).pushNamed(
            '/register',
          );
        }
```

Figure 4.2: Checking user type

We import the cloud_firestore.dart to query data from Firebase Firestore in order to check user type for navigating to their user interfaces. Figure 4.2 represents the function to query data from collection of users by using “uid”. Value received from Figure 4.1 as a user credential in order to get the user uid. We check if the collection of users uid existed in staff or athlete, the system will navigate to those interfaces. However, if the data doesn’t exist, it will navigate to register page.

4.2.1.2 Sign up Function

```
await FirebaseAuth.instance
          .createUserWithEmailAndPassword(
              email: _emailController.text.trim(),
              password: _passwordController.text.trim())
          .then((value) async {
```

Figure 4.3: Creating account function

Figure 4.3 represents the creating account function in Firebase Authentication Service. This function will be created user account by using email and password.

```
    Athlete athleteModel = Athlete(
  token: token,
  athlete_no: athlete_no,
  username: _usernameController.text.trim(),
  firstname: _firstnameController.text.trim(),
  lastname: _lastnameController.text.trim(),
  sportType: _selectedSport,
  birthdate: _birthdate,
  phoneNo: _phoneNoController.text.trim(),
  department: 'Athlete',
  weight: _selectedWeight,
  height: _selectedHeight,
  email: _emailController.text.trim(),
  gender: _selectedGender,
  age: age,
  pdpaAgreement: true);

Map<String, dynamic> data = athleteModel.toMap();

await FirebaseFirestore.instance
  .collection('Athlete')
  .doc(uid)
  .set(data)
  .then(
    (value) => print('Insert data to Firestore successfully'),
  );
```

Figure 4.4: Insert data function

We take the input data from user to create a data model that Figure 4.4 represented a model of athlete data. Then, we set the data model to document of user in Firebase Firestore.

4.2.2 Home page

4.2.2.1 Staff home page function

```
5   getAppointmentSize() {
6     FirebaseFirestore.instance
7         .collection('AppointmentRecord')
8         .where('staffUID', isEqualTo: uid)
9         .get()
10        .then((snapshot) {
11      int size = 0;
12      snapshot.docs.forEach((element) {
13        if (element['receivedStatus'] == true &&
14            element['staffReadStatus'] == false) {
15          size += 1;
16        }
17      });
18      if (mounted) {
19        setState(() {
20          appointmentSize = size;
21        });
22      }
23    });
24  }
```

Figure 4.5: Get appointment size function

Figure 4.5 represents the function of receiving appointment size of staff. It will process through the Firebase Firestore function in order to query the size of appointment data where “staffUID” equals to user uid. Then, the system will find out the data with following provided conditions and set state to get an appointment size.

```
4       getHealthSize() {
5         FirebaseFirestore.instance
6             .collection('HealthQuestionnaireResult')
7             .where('caseReceived', isEqualTo: false, isNull: false)
8             .get()
9             .then(
10          (snapshot) {
```

```
11            int size = 0;
12            snapshot.docs.forEach((data) {
13              if (data['totalPoint'] > 25) {
14                size += 1;
15              }
16            });
17            if (mounted) {
18              setState(() {
19                healthSize = size;
20              });
21            }
22          },
23        );
24      }
```

Figure 4.6: Get unreceived case size function

Figure 4.6 represents the function of receiving the unreceived health case size of staff from Firebase Firestore. In order to query for the size of unreceived case using in notification part, so we focus on whatever questionnaire contains more than 25 points. It will return only case that have not been finished in order to get size of them. This function is also utilized in unreceived physical case size.

```
4       getUnfinishedHealthCase() {
5         FirebaseFirestore.instance
6           .collection('HealthQuestionnaireResult')
7           .where('staff_uid_received', isEqualTo: uid, isNull: false)
8           .get()
9           .then(
10            (snapshot) {
11            List<QueryDocumentSnapshot<Map<String, dynamic>>> list = snapshot.docs;
12            list.removeWhere((element) => element['caseFinished'] == true);
13            if (mounted) {
14              setState(() {
15                unfinishedHealth = list.length;
16              });
17            }
```

```
18            },
19          );
20        }
```

Figure 4.7: Get unfinished case size function

Figure 4.7 represents the function of receiving the size of unfinished cases. This unfinished case size is queried from “HealthQuestionnaireResult” where “staff_uid_received” equals to user uid. This function is also utilized in unfinished physical case size.

```
4         FirebaseFirestore.instance
5               .collection('Staff')
6               .doc(uid)
7               .get()
8               .then((snapshot) {
9             setState(() {
10              Map data = snapshot.data();
11              staff = Staff.fromMap(data);
12            });
```

Figure 4.8: Get user data function

Figure 4.8 represents the function of receiving the user data from Firebase Firestore. This figure provided the receiving data of staff user by using their uid in order to get a snapshot data. Then, it will set a state to get the staff data. This function will process at the initial state and also be used in athlete data.

```
4     getToken() async {
5       String token = await FirebaseMessaging.instance.getToken();
6       await FirebaseFirestore.instance
7           .collection('Staff')
8           .doc(uid)
9           .update({'token': token});
10    }
```

Figure 4.9: Get notification token function

Figure 4.9 represents the function of receiving the notification token. The "getToken" function will return a Firebase Cloud Messaging (FCM) token of user's device. Then, we will update the token into Firebase Firestore System after user logged on.

4.2.2.2 Athlete home page function

```
4       getUnreceivedMessagePhysicalSize() {
5         FirebaseFirestore.instance
6             .collection('PhysicalQuestionnaireResult')
7             .where('athleteUID', isEqualTo: uid, isNull: false)
8             .get()
9             .then(
10          (snapshot) {
11            List<QueryDocumentSnapshot<Map<String, dynamic>>> list = snapshot.docs;
12            list.retainWhere((element) =>
13                element['caseFinished'] == true &&
14                element['messageReceived'] == false);
15            if (mounted) {
16              setState(() {
17                unreceivedPhysicalSize = list.length;
18              });
19            }
20          },
21        );
22      }
```

Figure 4.10: Get unreceived message size function

Figure 4.10 represents the function of receiving the message from finished case of physical questionnaire. This function is for showing the notification of unread messages for athletes. It also applied to health questionnaires in order to check both problems.

4.2.3 Athlete's questionnaire page

4.2.3.1 Questionnaire method function

```
5   void _resetQuestionnaire() {
6     setState(() {
```

```
7             _questionIndex = 0;
8             _bodyPartRound = 0;
9             isResult = true;
10            hasProblem = true;
11          });
12        }
```

Figure 4.11: Reset questionnaire function

Figure 4.11 represents a function of resetting the questionnaire when athletes finish the questionnaire both physical and health problems.

```
void _answerQuestion(int score) {
    answer_list["Q${_questionIndex + 1}"] = score;
    setState(() {
      _questionIndex += 1;
    });
    print('Index: $_questionIndex');
    if (_questionIndex < _questions.length) {
      print('We have more question');
    }
  }
```

Figure 4.12: Adding score to answer list

Figure 4.12 represents a function of adding score for each question into an answer list. When an athlete answers each question, the system will add a score after athlete chose a choice.

```
int _findTotalScore() {
    int _totalScore = 0;
    answer_list.forEach((key, value) {
      _totalScore += value;
    });
    totalScore = _totalScore;
    return _totalScore;
  }
```

Figure 4.13: Calculating total score function

Figure 4.13 represents a function of calculating total score from answer list in Figure 4.12. This function will be processed when athlete finished all questions and proceed to save the answer to Firebase Firestore.

4.2.3.2 Send notification function

```
void sendPushMessage(
    Athlete athlete, HealthResultData healthResultData) async {
  try {
    await http.post(
      Uri.parse('https://fcm.googleapis.com/fcm/send'),
      headers: <String, String>{
        'Content-Type': 'application/json',
        'Authorization':

'key=AAAAOmXVBT0:APA91bFonAMAsnJl3UDp2LQHXvThSOQd2j7q01EL1afdZl13TP7VEZxRa7q_Od
j3wUL_urjyfS7e0wbgEbwKbUKPkm8p5LFLAVE498z3X4VgNaR5iMF4M9JMpv8s14YsGqI2plf_lCBK',
      },
      body: jsonEncode(<String, dynamic>{
        'priority': 'high',
        'data': <String, dynamic>{
          'click_action': 'FLUTTER_NOTIFICATION_CLICK',
          'status': 'done',
          'title': 'ปัญหาอาการเจ็บป่วยถูกรายงาน',
          'body':
              'ข้อมูล ${healthResultData.questionnaireNo} ถูกรายงานโดยนักกีฬา ${athlete.firstname}
${athlete.lastname} ณ วันที่ ${formatDate(DateTime.now(), 'Athlete')} เวลา ${formatTime(DateTime.now())} น.',
        },
        'notification': {
          'title': 'ปัญหาอาการเจ็บป่วยถูกรายงาน',
          'body':
              'ข้อมูล ${healthResultData.questionnaireNo} ถูกรายงานโดยนักกีฬา ${athlete.firstname}
${athlete.lastname} ณ วันที่ ${formatDate(DateTime.now(), 'Athlete')} เวลา ${formatTime(DateTime.now())} น.',
        },
        'to': '/topics/Staff',
      }),
    );
  } catch (e) {
```

```
    print(e);
  }
}
```

Figure 4.14: Send push message function

Figure 4.14 represents a function of sending notification from athletes to staff. It will send a notification through Firebase Cloud Messaging Service by using "http.post" to https://fcm.googleapis.com/fcm/send. Then, the system will send pass through a key authorization of our account. The data has been sent as a type of JSON data including title, body, to whom.

4.2.4 Staff's medical record page

```
5       void onChangedMethodAffectedValue(String value) {
6         var affectedMapKey = _affectedList.keys
7             .firstWhere((k) => _affectedList[k] == value, orElse: () => null);
8
9         _codeAffectedSystem.text = affectedMapKey;
10      }
```

Figure 4.15: Change value to key function

Figure 4.15 represents a function of changing the value to its key. We set the key to a text controller.

```
String onChangedMethodAffectedKey(String value) {
  String affectedSystemValue = _affectedList[value];
  return affectedSystemValue;
}
```

Figure 4.16: Change key to value function

Figure 4.16 represents a function of changing key to its value. We set the value to a string data.

```
void addMainSymptomList() {
  isRepeat = false;
  valueAdded = true;
  bool addingValidator = _mainSymptomListKey.currentState.validate();
```

```
  if (isVisibleOtherMainSymptom == true) {
    _selectedMainSymptom += ', ${_otherMainSymptom.text.trim()}';
  }
  MainSymptomList newSymptom = MainSymptomList(
      selectedMainSymptom: _selectedMainSymptom,
      selectedMainSymptomCode: _codeMainSymptom.text);
  if (addingValidator) {
    if (int.tryParse(_codeMainSymptom.text) != 0 &&
        int.tryParse(_codeMainSymptom.text) <= 12 &&
        _selectedMainSymptom.isNotEmpty) {
      if (mainSymptoms.every((element) =>
          element.selectedMainSymptomCode !=
          newSymptom.selectedMainSymptomCode)) {
        mainSymptoms.add(MainSymptomList(
          selectedMainSymptom: _selectedMainSymptom,
          selectedMainSymptomCode: _codeMainSymptom.text.toString(),
        ));
      }
    }
  }
}
```

Figure 4.17: Adding symptom into list function

Figure 4.17 represents a function of adding main symptom into list. This function will check a repeat choice and “other” symptom, then the system will add into a list for recording in the Firebase Firestore.

```
Future<void> updateData(String docID) async {
  await FirebaseFirestore.instance
      .collection('HealthQuestionnaireResult')
      .doc(docID)
      .update({
    'caseFinished': true,
    'caseFinishedDateTime': DateTime.now(),
    'adviceMessage': _messageController.text,
    'messageReceived': false,
  }).then((value) {
```

```
    print('Updated data successfully');
  });
 }
}
```

Figure 4.18: Updating case function

Figure 4.18 represents a function of updating athlete's case function. This function updates not only health questionnaire but also physical questionnaire.

4.2.5 Notification page

```
5   import 'package:async/async.dart' show StreamZip;
6   Stream<List<QuerySnapshot>> getData() {
7     Stream healthQuestionnaire = FirebaseFirestore.instance
8       .collection('HealthQuestionnaireResult')
9       .where('caseReceived', isEqualTo: false, isNull: false)
10      .snapshots();
11    Stream physicalQuestionnaire = FirebaseFirestore.instance
12      .collection('PhysicalQuestionnaireResult')
13      .where('caseReceived', isEqualTo: false, isNull: false)
14      .snapshots();
15    return StreamZip([healthQuestionnaire, physicalQuestionnaire]);
16  }
```

Figure 4.19: Get data function

Figure 4.19 represents a function of receiving data with multiple collections. We used StreamZip to aggregate snapshots of health questionnaire and physical questionnaire to be utilized in StreamBuilder. This function process both athlete and staff notification page.

4.2.6 Athlete's history page

```
5   List<Map<String, dynamic>> add_filter(List<Map<String, dynamic>> data) {
6     if (_selectedOrder[0] == true) {
```

```
7           if (_selectedOrderType[0] == true) {
8             data.sort(
9               (a, b) => (b['doDate'].toDate()).compareTo(
10                a['doDate'].toDate(),
11              ),
12            );
13          } else {
14            data.sort(
15              (a, b) => (a['doDate'].toDate()).compareTo(
16                b['doDate'].toDate(),
17              ),
18            );
19          }
20        } else if (_selectedOrder[1] == true) {
21          if (_selectedOrderType[0] == true) {
22            data.sort(
23              (a, b) => (b['totalPoint']).compareTo(
24                a['totalPoint'],
25              ),
26            );
27          } else {
28            data.sort(
29              (a, b) => (a['totalPoint']).compareTo(
30                b['totalPoint'],
31              ),
32            );
33          }
34        }
35
36        if (_selectedQuestionnaire[0] == false) {
37          data.removeWhere((element) => element['questionnaireType'] == 'Physical');
38        }
39        if (_selectedQuestionnaire[1] == false) {
40          data.removeWhere((element) => element['questionnaireType'] == 'Health');
41        }
42
43        data.removeWhere(
44            (element) => element['totalPoint'] < _currentRangeValues.start);
```

```
45        data.removeWhere(
46            (element) => element['totalPoint'] > _currentRangeValues.end);
47        return data;
48      }
```

Figure 4.20: Adding filter function

Figure 4.20 represents a function of adding a filter in order to get more specific history that athletes desired. In addition, staff also could be utilized this function when they desired to get more specific history.

```
List<String> find_answer(List<Map<String, Object>> question) {
    List<String> answerText = [];
    int i = 1;
    question.forEach((map) {
      (map['answerText'] as List<Map<String, Object>>).forEach((answer) {
        print('${answerList['Q$i']} and ${answer['score']}');
        if (answerList['Q$i'] == answer['score']) {
          answerText.add(answer['text'].toString());
        }
      });
      i++;
    });
    return answerText;
  }
```

Figure 4.21: Finding answer details function

Figure 4.21 represents a function of finding answer details when athlete or staff would like to view deeper details. This function will process about a question and its score.

4.2.7 Staff's case page

```
5     void _getAppointmentData() {
6       FirebaseFirestore.instance
7           .collection('AppointmentRecord')
8           .where('caseID', isEqualTo: widget.docID)
9           .get()
10          .then((snapshot) {
11        setState(() {
```

```
12            if (snapshot.docs.isNotEmpty) {
13              latestAppointment = snapshot.docs.first.data();
14            } else {
15              print("This case doesn't have any appointment");
16            }
17          });
18        });
19      }
```

Figure 4.22: Get the latest appointment data

Figure 4.22 represents a function of receiving the latest appointment data from Firebase Firestore in order to check a currently appointment in case of containing more than one appointment of each case.

4.2.8 Staff's history page

```
getIllnessSize() {
  FirebaseFirestore.instance
      .collection('IllnessRecord')
      .where('staff_uid', isEqualTo: uid)
      .get()
      .then(
    (snapshot) {
      setState(() {
        illnessSize = snapshot.docs.length;
      });
    },
  );
 }
```

Figure 4.23: Get medical history size function

Figure 4.23 represents a function of receiving illness history size from a medical record in Firebase Firestore. This function also processes for injury history in order to find out a size of medical record of each staff.

4.2.9 Visualization page

4.2.9.1 Staff graph page

```
5       Future pickDateRange() async {
```

```
6         DateTimeRange newDateRange = await showDateRangePicker(
7           context: context,
8           initialDateRange: dateRange,
9           firstDate: DateTime(1900),
10          lastDate: DateTime(now.year + 1),
11        );
12
13        setState(() {
14          dateRange = newDateRange;
15        });
16      }
```

Figure 4.24: Picking range of data function

Figure 4.24 represents a function of picking range of date when staff wants to get a range of statistics.

```
int countNumberOfAthlete(List<Map<String, dynamic>> inputList) {
   Set targetSet = Set<String>();
   inputList.forEach((element) {
     targetSet.add(element['athleteUID']);
   });
   return targetSet.length;
 }
```

Figure 4.25: Count number of athlete function

Figure 4.25 represents a function of counting number of athlete. When we receive the data of all athletes, there will be contains the repeated athlete data. Therefore, we use Set<String> in order to get data with unique value from athlete. An input list is a list of athlete data provided from between health cases and physical cases.

```
String getProblem(Map<String, dynamic> resultData) {
  String problem;
  if (resultData['questionnaireType'] == 'Health') {
    problem = resultData['healthSymptom'];
  } else {
    problem = resultData['bodyPart'];
  }
  return problem;
}
```

Figure 4.26: Get a problem function

Figure 4.26 represents a function of returning a problem type using in a summary table of staff. The summary table contains the different type of problem, so there is a function to separate between two problems.

```
String getAthleteDepartment(Map<String, dynamic> resultData) {
  String sportType;
  athleteList.forEach((element) {
    if (element['athleteUID'] == resultData['athleteUID']) {
      sportType = element['sportType'];
    }
  });
  return sportType;
}
```

Figure 4.27: Get a department of athlete

Figure 4.27 represents a function of returning a department of athlete in order to figure out the department and sports of each athlete.

4.2.9.2 Athlete's visualization page

```
5         Color getColors(int shade) {
6           Color colors;
7           if (widget.isStaff == true) {
8             colors = Colors.blue[shade];
9           } else {
10            colors = Colors.green[shade];
```

```
11          }
12          return colors;
13        }
```

Figure 4.28: Get user interface color function

Figure 4.28 represents a function of choosing a color function between staff and athlete. Due to the visualization page of athlete, both staff and athlete can view. Therefore, the color of interface should be different.

```
// get week between first date and now
 static double getWeekDay(DateTime from, DateTime to) {
  from = DateTime.utc(from.year, from.month, from.day);
  to = DateTime.utc(to.year, to.month, to.day);
  return (to.difference(from).inDays / 7).ceil().toDouble();
 }
```

Figure 4.29: Get weekday function

Figure 4.29 represents a function of receiving a week between first date of a year and currently date.

```
import 'package:fl_chart/fl_chart.dart';
ScatterSpot getHealthChart(Map<String, dynamic> healthResultDataList) {
  ScatterSpot spot;

  spot = ScatterSpot(
    AthleteLineGraph.getWeekDay(
      AthleteLineGraph.firstJan, (healthResultDataList['doDate']).toDate()),
    healthResultDataList['totalPoint'].toDouble(),
    show: true,
    color: Colors.purple,
  );

  return spot;
 }
```

Figure 4.30: Get statistics chart function

Figure 4.30 represents a function of receiving scatter plot from health data. It is also utilized in physical data. In order to process with ScatterSpot widget, we have to

import fl_chart. In addition, health data will be shown as purple color and physical data will be shown as blue color.

Those all functions are only examples of the function implementing in our code. We provided the meaning, guide, and implementation of function for each page. However, there are other pages that utilized those functions, but it utilized in different ways and variable.

CHAPTER 5
TESTING AND EVALUATION

This chapter describes how the application has been tested and what the results have been.

5.1 Unit Tests

For the unit tests, we selected some important and critical processes for formal unit testing. The processes have two parts, staff and athlete. The selected processes include:

- Athlete
  - Process 1: Sign-up
  - Process 2: Log-in
  - Process 3: Fill the Health Questionnaire
  - Process 4: Fill the Physical Questionnaire
  - Process 5: Fill the Mental Questionnaire
  - Process 6: View history
  - Process 7: Notification
  - Process 8: View graph
- Staff
  - Process 1: Sign-up
  - Process 2: Log-in
  - Process 9: Fill the Illness Record
  - Process 10: Fill the Injury Record
  - Process 11: Collect case
  - Process 12: Watch cases
  - Process 13: View summary table
  - Process 14: View history

5.1.1 Test Performed on Process 1: Sign-up

Table 5.1: Sign-up

Operation Performed	Condition Tested	Actual Result
1. Sign-up	Be able to fill in the sign-up form with athlete and staff role	Pass
2. PDPA	Be able to click accept the condition in PDPA	Pass
3. Sign-up Button	Be able to click the sign-up button	Pass

5.1.2 Test Performed on Process 2: Log in

Table 5.2: Log in

Operation Performed	Condition Tested	Actual Result
1. Username and Password	Be able to fill the username and password	Pass
2. Login Button	Be able to use the username	Pass

5.1.3 Test Performed on Process 3: Fill the Health Questionnaire

Table 5.3: Fill the Health Questionnaire

Operation Performed	Condition Tested	Actual Result
1. Health Questionnaire Button	Be able to click the button to go to health questionnaire	Pass
2. Health Questionnaire	Be able to do the health questionnaire and complete it	Pass
3. Save button	Be able to collect the questionnaires of that person	Pass

5.1.4 Test Performed on Process 4: Fill the Physical Questionnaire

Table 5.4: Fill the Physical Questionnaire

Operation Performed	Condition Tested	Actual Result
1. Physical Questionnaire Button	Be able to click the button to go to health questionnaire	Pass
2. Physical Questionnaire	Be able to do the physical questionnaire and complete it	Pass
3. Save button	Be able to collect the questionnaires of that person	Pass

5.1.5 Test Performed on Process 5: Fill the Mental Questionnaire

Table 5.5: Fill the Mental Questionnaire

Operation Performed	Condition Tested	Actual Result
1. Mental Questionnaire Button	Be able to click the button to go to health questionnaire	Pass
2. Mental Questionnaire	Be able to do the mental questionnaire and complete it	Pass
3. Save button	Be able to collect the questionnaires of that person	Pass

5.1.6 Test Performed on Process 6: View history

Table 5.6: View history (Athlete)

Operation Performed	Condition Tested	Actual Result
1. History data card	Be able to click the card to	Pass

	see additional information of questionnaires taken.	
2. History Filter	Be able to click to filter the information that they want to see.	Pass
3. Default switch	Be able to click to set default filter	Pass

5.1.7 Test Performed on Process 7 Notification

Table 5.7: Notification

Operation Performed	Condition Tested	Actual Result
1. Notification card	Be able to see the message from staff	Pass
2. Appointment card	Be able to see the appointment from staff	Pass

5.1.8 Test Performed on Process 8: View statistics

Table 5.8: View graph

Operation Performed	Condition Tested	Actual Result
1. Graph	Be able to see the graph about the scores of each week	Pass
2. Graph Filter	Be able to click to filter the graph information that they want to see.	Pass
3. Default switch	Be able to click to set default filter	Pass

5.1.9 Test Performed on Process 9: Fill the Illness record

Table 5.9: Fill the Illness record

Operation Performed	Condition Tested	Actual Result
1. Illness record button	Be able to click button to go to the illness record form.	Pass
2. Illness form	Be able to do the Illness record and complete its.	Pass
3. Save button	Be able to collect the illness record of that person into database.	Pass

5.1.10 Test Performed on Process 10: Fill the Injury record

Table 5.10: Fill the Injury record

Operation Performed	Condition Tested	Actual Result
1. Injury record button	Be able to click button to go to the injury record form.	Pass
3. Injury form	Be able to do the Injury record and complete its.	Pass
4. Save button	Be able to collect the injury record of that person into database.	Pass

5.1.11 Test Performed on Process 11: Collect cases

Table 5.11: Collect cases

Operation Performed	Condition Tested	Actual Result
1. Case card	Be able to click card to see	Pass

	more information from the first preliminary information.	
2. Add case	Be able to press button to add the case they want to do.	Pass

5.1.12 Test Performed on Process 12: Watch cases

Table 5.12: Watch cases

Operation Performed	Condition Tested	Actual Result
1. Detail button	Be able to click button to see more information and do record and appointment to athlete.	Pass
2. Record button	Be able to press button to do the record depending on the case type.	Pass

5.1.13 Test Performed on Process 13: View statistics

Table 5.13: View summary table

Operation Performed	Condition Tested	Actual Result
1. Summary table	Be able to show the table about the number of the case and athletes who have the injury and illness problem in that week	Pass
2. Athlete list	Be able to show the list of the athletes and can click the name to show the score graph of that person	Pass

3. Athlete search	Be able to search the specific athlete that the staff wants.	Pass
4. Filter	Be able to filter the summary information that the staff wants such as the range of the days	Pass

5.1.14 Test Performed on Process 14: View finished cases/records

Table 5.14: View finished cases/records

Operation Performed	Condition Tested	Actual Result
1. Case data card	Be able to click the card to see additional information of cases that you finish.	Pass
2. Case Filter	Be able to click to filter the case information that they want to see.	Pass
3. Default switch	Be able to click to set default filter	Pass

5.2 System Integration Test

This activity is performed after the system is completely integrated. The purpose of this testing is to check whether the system can operate correctly according to the required functions or not.

5.2.1 Test Scenario

In order to test all functional aspects of the system thoroughly, we had set up a test scenario which consisted of 11 phases as shown below.

- Sign up successfully
- Log in successfully
- Login failed

- Fill the questionnaire successfully
- Do one more questionnaire
- Fill a record successfully
- Make appointment successfully

Moreover, the test scenario can be used as a user guideline because it covers all the steps necessary in order to use our system. The details of each phase are shown in the next section.

5.2.1.1 Sign up

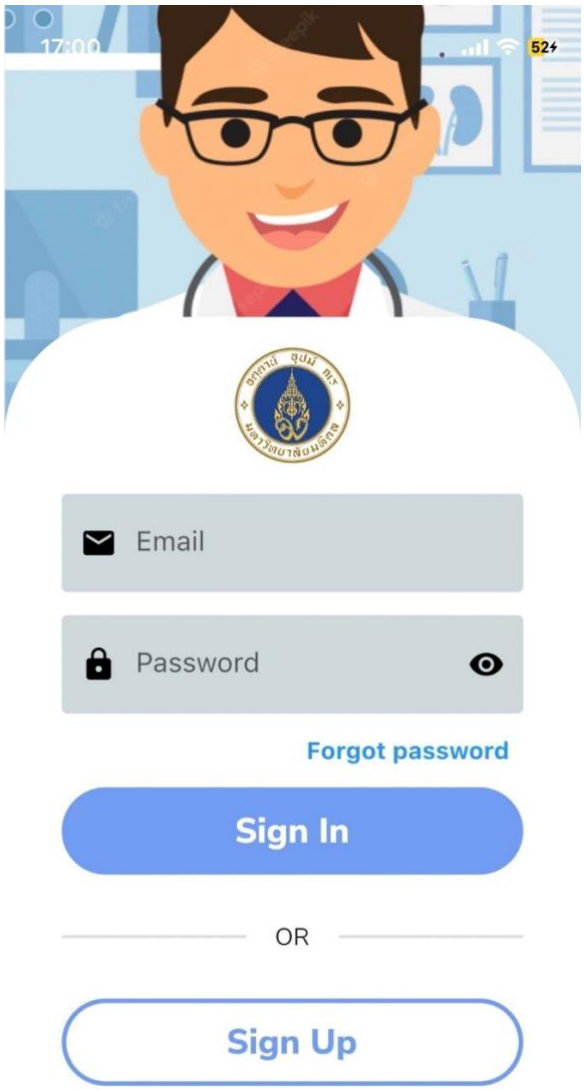


Figure 5.1: Login and Register page

In sign up phase, you click sign up button to go to sign up page.

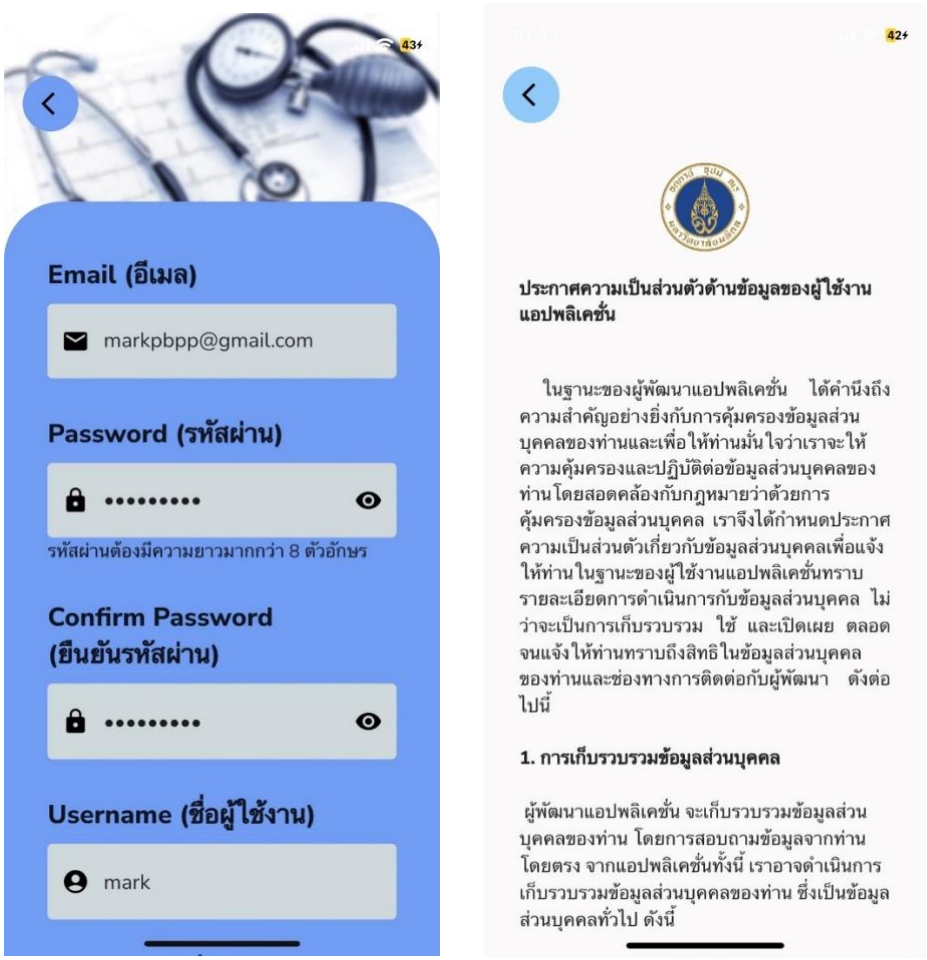


Figure 5.2: Sign up Successfully

When you go to sign up page, you put the information in the form to register and click confirmation for Personal Data Protection Act. If you finish everything, it goes to the home page showing that the account has been created.

5.2.1.2 Login

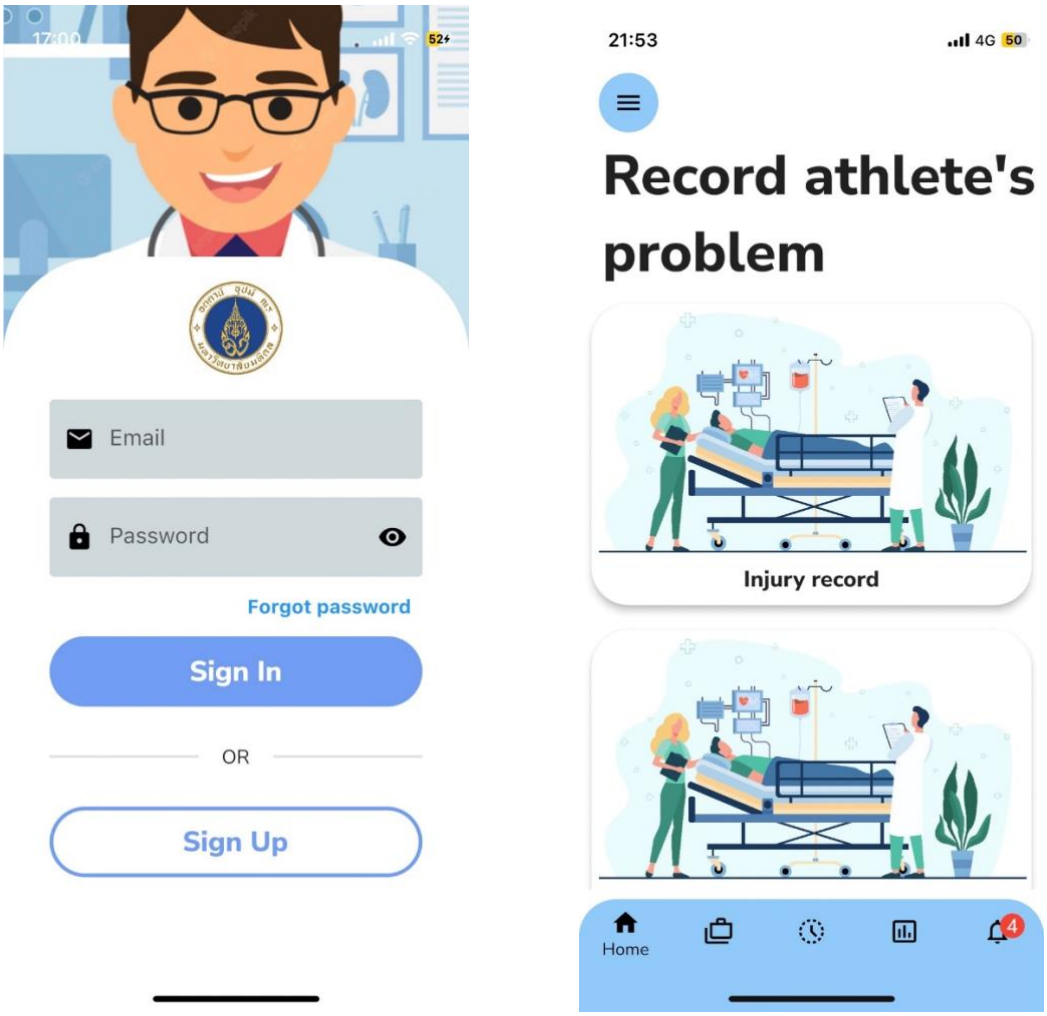


Figure 5.3: Login Successfully

You can put the email and password into sign in page to login to your account. If you enter email and password correct, it goes to the homepage of your account.

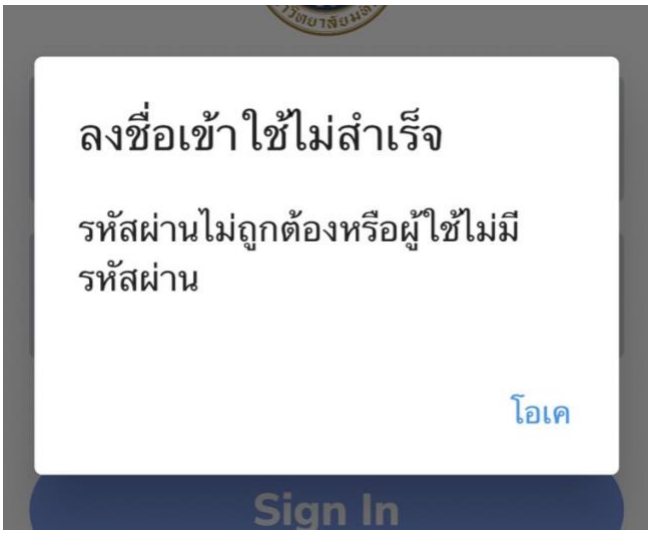


Figure 5.4: Login failed

When you enter email or password incorrect, it shows the pop-up alert this topic “ลงชื่อเข้าใช้ไม่สำเร็จ”.

5.2.1.3 Fill the questionnaire

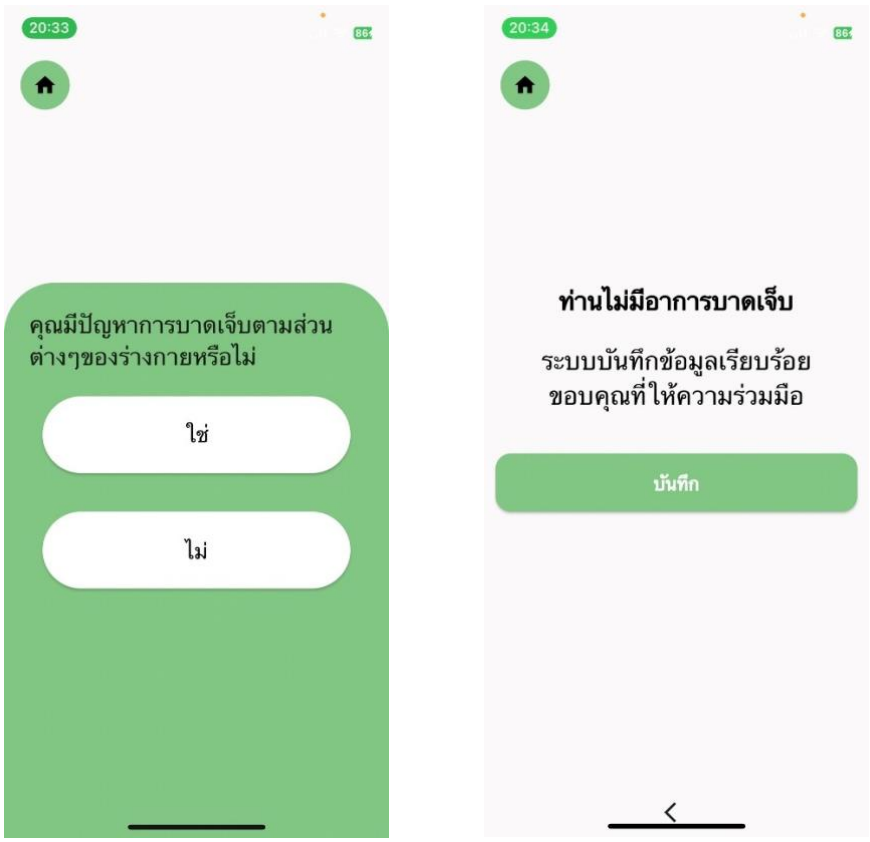


Figure 5.5: Fill the questionnaire Successfully (No pain/health problem)

If you click “no” in the first question, it shows that “ท่านไม่มีอาการบาดเจ็บ” and can save the data into database.

Figure 5.6: Choose pain and do questionnaire

If you click “yes” in the first question, it will go to selecting part of body or illness that it depends on the type of questionnaire. When you select finished, it goes to a questionnaire which asks about the symptoms.

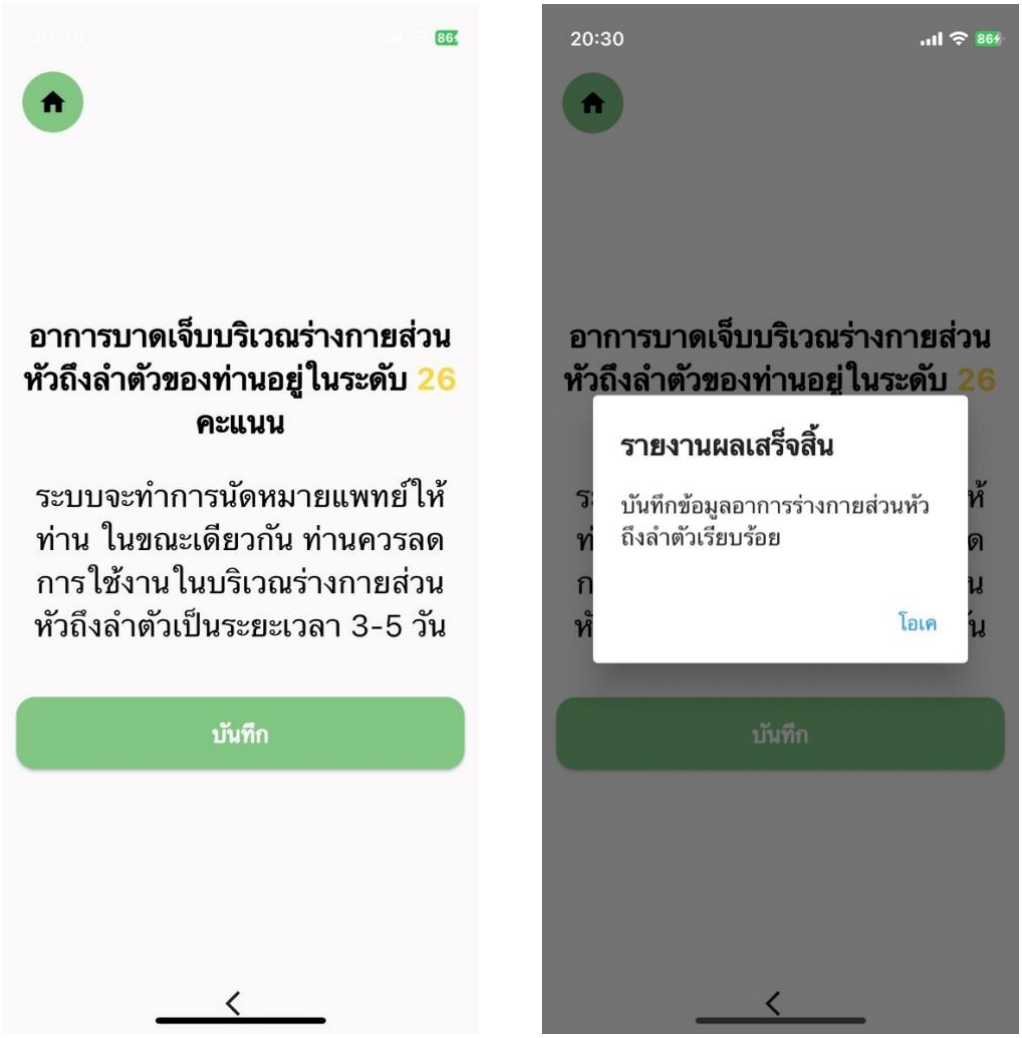


Figure 5.7: Fill the questionnaire Successfully

When you do all of questions, it shows the score of the pain or health symptom and preliminary advice. When you see its finished, it goes to complete the questionnaire and save in the database.

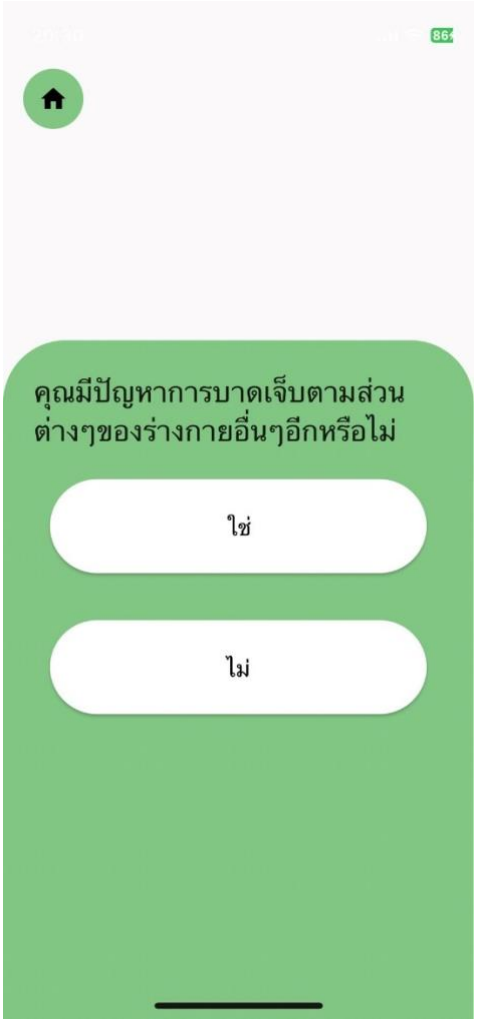


Figure 5.8: Fill questionnaire more than one pain or health problem

When you save the first questionnaire and have more pains and health symptoms, you can click “yes” in this question, and it take you to do another questionnaire for new symptoms.

5.2.1.4 Fill the record

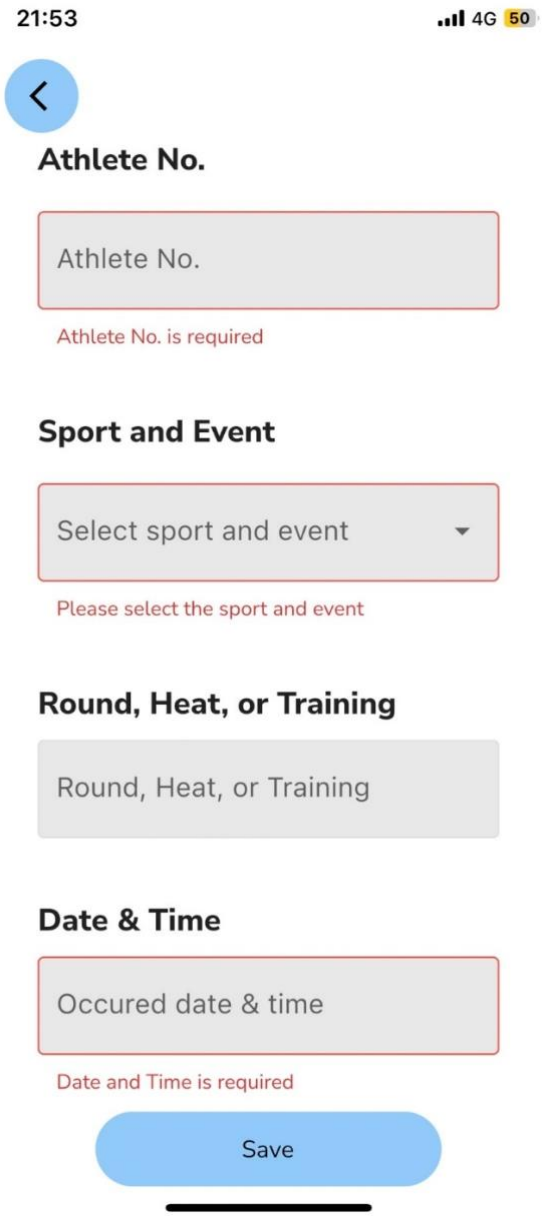


Figure 5.9: Fill the record not Successfully

When you fill the record, you have to fill the required box in form. If you do not fill some box, it alerts the required box that you must fill it.

Figure 5.10: Fill the record Successfully

When you record finished, it shows the pop-up that you insert data successfully in the database.

5.2.1.5 Appointment

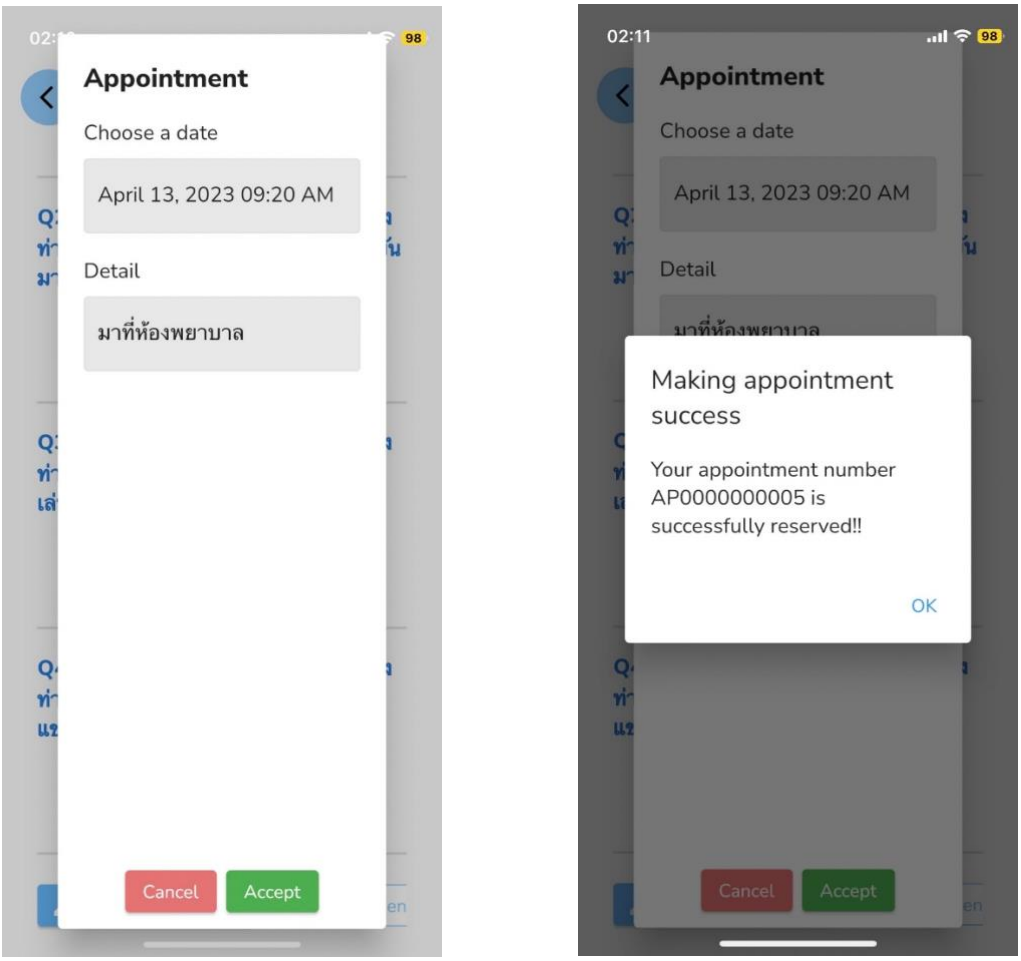


Figure 5.11: Make appointment Successfully

When you want to appoint the athlete, you can do in appointment form. If you do its finished and click accept, it pop up successfully for aappointment athlete.

5.3 Usability and Satisfaction Evaluation

Two categories comprise our usability and satisfaction assessments: usability evaluations for athletes and usability evaluations for staff. The fourteen athletes of the Sport Science Faculty's futsal team, together with seven members of the coaching and staff, volunteered to participate in a usability test. For demonstrating how to use the app we used video tutorials that we created, and we also recorded results and asked for one-on-one advice at the National Sports and Mahidol University.

The summary of the result for the application, we use used time to value learnability for our application and get satisfaction score to scale its usefulness and usability and see the completeness to use our application.

Table 5.15: Usability test from athlete, Learnability

Measuring instrument	**Target level**	**Average time**
Sign in	1 minutes	1 minute 17 seconds
Log in	20 seconds	17 seconds
Fill the health questionnaire	50 seconds	42 seconds
Fill the physical questionnaire	50 seconds	40 seconds
View history	8 seconds	10 seconds
Graph	12 seconds	13 seconds

Table 5.16: Usability test from athlete, Satisfaction

Measuring instrument	**Average score**
Interested to use this application	4.5

Table 5.17: Usability test from athlete, Completeness

Measuring instrument	**Success rate**
Complete the questionnaire	100%
Overall to using application	100%

Table 5.18: Usability test from staff, Learnability

Measuring instrument	**Target level**	**Average time**
Sign in	1 minutes	1 minute 18 seconds
Log in	20 seconds	22 seconds
Fill the injury record	1 minute	1 minute 16 seconds
Fill the illness record	1 minute	1 minute 34 seconds
View history	14 seconds	22 seconds
Collect case (Notification)	8 seconds	13 seconds
Watch cases and do record in case (Case feature)	1 minute 30 seconds	1 minute 49 seconds
Summary table	45 seconds	1 minute

Table 5.19: Usability test from staff, Satisfaction

Measuring instrument	**Average score**
Interested to use this application	4.5

Table 5.20: Usability test from staff, Completeness

Measuring instrument	**Success rate**
Complete the record	100%
Overall to using application	100%

CHAPTER 6
CONCLUSIONS

Due to the inefficient method of collecting Athletes' Physical complaints and Health problem via paper we decided to implement an SIRA application that stand for Sport Injury and Illness Recording Application in order to solve the problem. Apart from that, let's keep at our main point of SIRA, which is developing an application for tracking athletes' weekly scores for staff and doctor to get the athletes' being and how they are improving, they also required to do the weekly questionnaires both physical complain and health problem to let staff know and prepare for treatment. All data in questionnaires will be collected in a database server only accessible for the staff. Staff will analyze how the athletes' being. In addition, SIRA can help use less paper to fix global warming too.

6.1 Benefits

This project gives several benefits to both users and developer.

6.1.1 Benefits to Project Developers

- Developers can use the knowledge studied from the faculty to implement the application
- Developers can learn the mistake and bug in the program while developing the application
- Developers can practice the coding, programming skill from implementing the application
- Developers can evolve many skills such as problem-solving, prioritizing task, management, and presentation skill

6.1.2 Benefits to Users

- Users, especially athletes can get benefits by receiving the recommendation from staff, so they can take care themselves in the correct way
- The staff can follow up the athletes' injury or illness state

6.2 Problems and Limitations

- Some of the questions make athletes confuse and can't answer properly
- Don't have database to handle many associations.
- During questionnaire, user cannot change language

6.3 Future Work

- Expand the database to handle many data associations
- Athletes can make an appointment with the doctor
- Graph showing score will have clear information.

REFERENCES

[1] International Olympic Committee (IOC), "London 2012 Olympic Summer Games Injury & Illness Surveillance Study," 2012. [Online]. Available: https://stillmed.olympic.org/media/Document%20Library/OlympicOrg/Games/Summer-Games/Games-London-2012-Olympic-Games/Anti-doping-and-Medical-Rules/Injury-and-Illness-Surveillance-Study-London-2012.pdf. [Accessed 20 November 2022].

[2] K. Kaewkul, K. Chaijenkij and S. Tongsai, "Validity and Reliability of the Oslo Sports Trauma Research Center (OSTRC) Questionnaire on Overuse Injury and Health Problem in Thai Version," *J Med Assoc Thai 2021,* vol. 104, no. 1, pp. 105-113, 21 August 2020.

[3] L. Engebretsen and R. Bahr, "The main aim of Oslo Sports Trauma Research Center is to prevent injuries and other health problems in sports - through research on risk factors, injury mechanisms and prevention methods," 15 October 2021. [Online]. Available: https://www.med.uio.no/klinmed/english/research/groups/oslo-sports-trauma-research-center-ostrc/index.html. [Accessed 24 November 2022].

[4] B. Clarsen, R. Bahr and G. Myklebust, "Improved reporting of overuse injuries and health problems in sport: an update of the Oslo Sport Trauma Research Center questionnaires," *Br J Sports Med,* vol. 54, pp. 390-396, 14 February 2020.

[5] B. Clarsen, R. Bahr and G. Myklebust, "Development and validation of a new methodfor the registration of overuse injuries in sportsinjury epidemiology," *Br J Sports Med,* vol. 00, pp. 1-8, 4 October 2012.

[6] B. P. Raysmith and M. K. Drew, "Performance success or failure is influenced by weeks lost to injury and illness in elite Australian track and field athletes: A 5-year prospective study," *J Sci Med Sport,* vol. 19, no. 10, pp. 778-783, 1 October 2016.

[7] L. Engebretsen, T. Soligard and K. Steffen, "Sports injuries and illnesses during the LondonSummer Olympic Games 2012," *Br J Sports Med,* vol. 0, pp. 1-8, 20 March 2013.

BIOGRAPHIES

NAME Mr. Krissanapong Palakham

INSTITUTIONS ATTENDED Assumption College Thonburi, 2019:
High School Diploma
Mahidol University, 2023:
Bachelor of Science (ICT)

NAME Mr. Pongsakorn Piboonpongpun

INSTITUTIONS ATTENDED Assumption College Thonburi, 2019:
High School Diploma
Mahidol University, 2023:
Bachelor of Science (ICT)

NAME Mr. Rathakit Sriprachayanun

INSTITUTIONS ATTENDED Assumption College, 2019:
High School Diploma
Mahidol University, 2023:
Bachelor of Science (ICT)