

แอปพลิเคชันจัดเวรแพทย์และพยาบาล

Doctor and nurse organization application



ธีรวิช สุขประเสริฐกุล

Tirawich Sukprasertkul

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการ 2

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

แอปพลิเคชันจัดเวรแพทย์และพยาบาล

Doctor and nurse organization application



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการงาน2

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2562

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันจัดเวรแพทย์และพยาบาล

Doctor and nurse organization application



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงงาน2

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2562

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองวิชาโครงการ 2

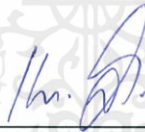
รายงานวิชาการประยุกต์วงจรรอิเล็กทรอนิกส์ 2 ปีการศึกษา 2562

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์
คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แอปพลิเคชันจัดเวรแพทย์และพยาบาล
doctor/nurse organization application
ผู้จัดทำ อธิวิษ สุขประเสริฐกุล รหัสประจำตัว 58010608

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

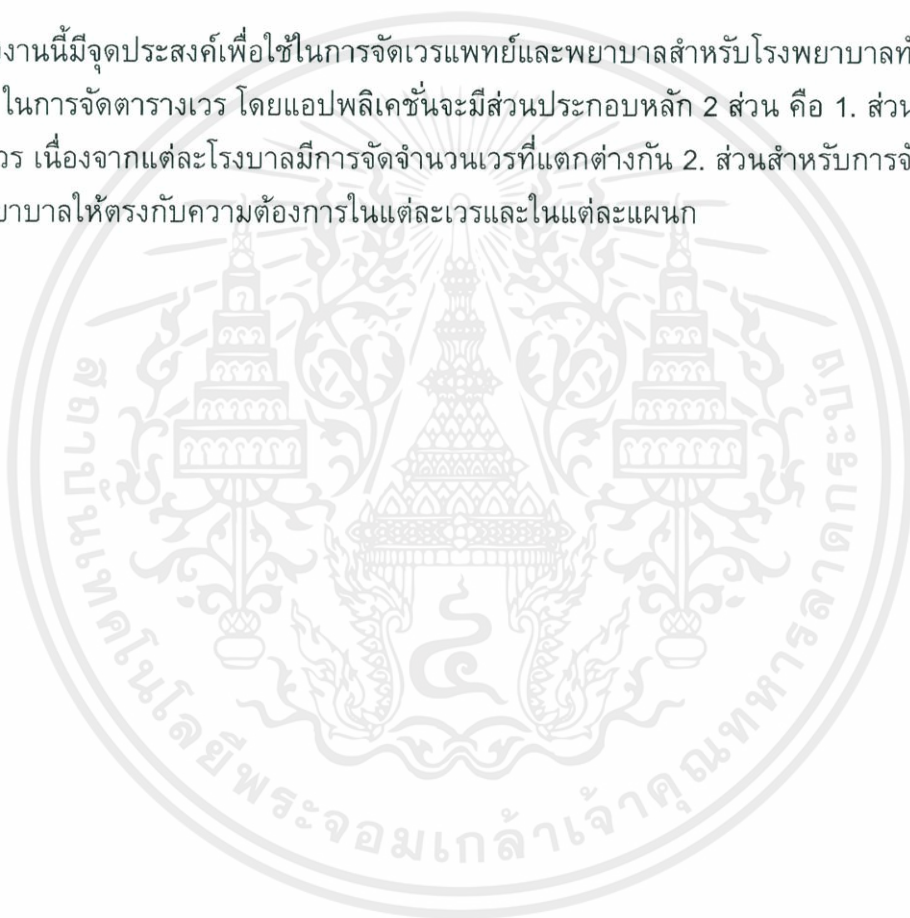


ดร.เทอดศักดิ์ ลีวหาทอง
อาจารย์ที่ปรึกษา

หัวข้อโครงการ	แอปพลิเคชันจัดเวรแพทย์และพยาบาล
นักศึกษา	ธีรวิษ สุขประเสริฐกุล รหัสประจำตัว 58010608
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2562
อาจารย์ที่ปรึกษาโครงการ	ดร.เทอดศักดิ์ ลีวหาทอง

บทคัดย่อ

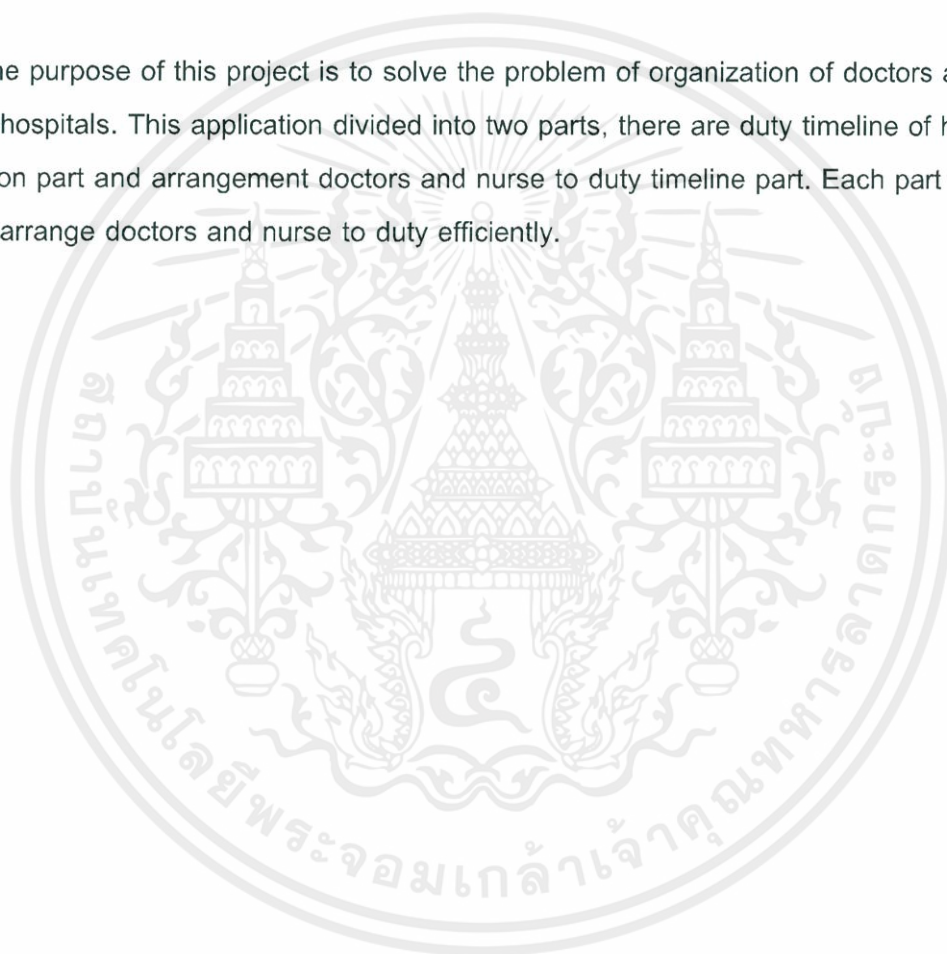
โครงการนี้มีจุดประสงค์เพื่อใช้ในการจัดเวรแพทย์และพยาบาลสำหรับโรงพยาบาลทั่วไป เพื่อแก้ไขปัญหาในการจัดตารางเวร โดยแอปพลิเคชันจะมีส่วนประกอบหลัก 2 ส่วน คือ 1. ส่วนสำหรับการปรับจำนวนเวร เนื่องจากแต่ละโรงพยาบาลมีการจัดจำนวนเวรที่แตกต่างกัน 2. ส่วนสำหรับการจัดเรียงแพทย์และพยาบาลให้ตรงกับความต้องการในแต่ละเวรและในแต่ละแผนก



Project Title	doctor/nurse organization application	
Student	Tirawich Sukprasertkul	Student ID 58010608
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2019	
Project Advisor	Dr.Thurdsak Leauhatong	

Abstract

The purpose of this project is to solve the problem of organization of doctors and nurses in hospitals. This application divided into two parts, there are duty timeline of hospital organization part and arrangement doctors and nurse to duty timeline part. Each part will co-work and arrange doctors and nurse to duty efficiently.



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการแอปพลิเคชันจัดเวรแพทย์และพยาบาล (doctor/nurse organization application) สามารถบรรลุเป้าหมายมาได้ ด้วยความช่วยเหลือและให้คำปรึกษาจาก ดร.เทอดศักดิ์ ลีวาทอง อาจารย์ที่ปรึกษาที่คอยให้คำปรึกษาและกระตุ้น รวมถึงติดตามงานอยู่ตลอด ผู้จัดทำโครงการขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณบิดาและมารดาของคณะผู้จัดทำที่คอยเป็นกำลังใจหลักและเป็นผู้สนับสนุนเงินทุนหลักในการทำโครงการ ให้คำแนะนำ รวมทั้งช่วยกันในการแก้ปัญหาที่เกิดขึ้นในระหว่างการทำโครงการนี้ทั้งหมดทำให้ผลของโครงการสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ ผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้สนใจและผู้นำผลงานนี้ไปใช้งานได้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	1
1.3 สมมุติฐานของการศึกษา	1
1.4 ขอบเขตการศึกษา	1
1.5 ระยะเวลาในการทำโครงการ	2
ประโยชน์ที่คาดว่าจะได้รับ	2
หลักการและทฤษฎี	3
2.1 ส่วน register	4
2.2 ส่วน header หรือ หัวหน้าแผนก	4
2.3 ส่วนของ หมอเฉพาะทาง และ พยาบาล	4
การออกแบบ	5
3.1 ฐานข้อมูลที่ออกแบบ	5
3.2 Algorithm การทำงาน	7
ผลการทดลอง	10
4.1 การทดลองสมัครสมาชิก	10
4.2 interfaceในสำหรับแพทย์/พยาบาล	13
4.3 interface ในสำหรับหัวหน้าแผนก	13
สรุปผลการทดลอง	16
5.1 สรุปผลการทดลอง	16
5.2 ปัญหาอุปสรรคและการแก้ไข	16
บรรณานุกรม	17
ภาคผนวก	18

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1.1 แสดงระยะเวลาในการทำโครงการ	2
3.1 staffs	5
3.2 departments	5
3.3 works	6



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
4.1.1 หน้า register ข้อมูลผู้ใช้	10
4.1.2 เลือกแผนกจากspinner	10
4.1.3 เลือกตำแหน่งของผู้ใช้	11
4.1.4 กรอกข้อมูลผู้ใช้	11
4.1.5 แสดง email ใน firebaseAuth	12
4.1.6 แสดงข้อมูลผู้ใช้ใน firebasefirestore	12
4.2 ตารางในส่วนแพทย์/พยาบาล	13
4.3.1 ตารางในส่วนหัวหน้าแผนก	13
4.3.2 ส่วนกำหนดงาน	14
4.3.3 ตารางงานใน firebasefirestore	14
4.3.4 เมื่อลงเวรติดกัน 3 เวร	15
4.3.5 ชั่วโมงการทำงานเกิน 56 ชั่วโมง	15

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันหลายโรงพยาบาลต่างกำลังประสบปัญหาหนักในการจัดเวรแพทย์และพยาบาลเพื่อประจำหน้าที่ในแต่ละแผนก เนื่องจากการขาดแคลนบุคลากรและการจัดเวรที่ยังไม่มีประสิทธิภาพพอ จึงทำให้แพทย์และพยาบาลหลายคนต้องทำงานอย่างหนัก ข้ามวันข้ามคืนจนเกิดอาการเหนื่อยล้าทั้งร่างกาย และ จิตใจ เมื่อเหนื่อยล้าอ่อนเพลียผลที่ตามมาคือ ประสิทธิภาพในการรักษาคนไข้ค่อยลง บางครั้งอาจหมายถึงชีวิตของคนไข้อีกด้วย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อสร้างการจัดระเบียบแพทย์และพยาบาลให้เป็นระเบียบขึ้น และ ลดความเหนื่อยล้าจากการทำงานเกินเวลา (56 ชั่วโมง ต่อ สัปดาห์)

1.3 สมมติฐานของการศึกษา

แอปพลิเคชันนี้จะสามารถช่วยให้การจัดการเวลาการทำงานของแพทย์และพยาบาลที่กำลังเป็นปัญหาอยู่ในตอนนี้ในแต่ละรพ ให้มีประสิทธิภาพมากขึ้น หรือ บรรเทาปัญหานี้ให้ลดน้อยลง โดยการใช้อปพลิเคชันนี้มาจัดการ และแสดงตารางการทำงานของแพทย์ พยาบาลอย่างชัดเจนโดยมีการแบ่งเป็น สองแบบในการlogin ป้องกันความสับสน

1.4 ขอบเขตการศึกษา

แอปพลิเคชันนี้จะทำการเก็บข้อมูลแพทย์และพยาบาลทั้งหมดเข้าไว้ใน ฐานข้อมูล firebase ในรูปแบบ array list จากนั้นหัวหน้าแต่ละแผนกจะเป็นผู้กำหนดว่าใครจะเป็นกำหนดเวลาในการลงตามความเหมาะสม และ จากนั้นในส่วนของแพทย์และพยาบาลจะแสดงตารางงานออกมาในรอบ อาทิตย์ และ เดือน

โดยจะแบ่ง role ในการเข้าใช้เป็น 2 ส่วน คือ

1. แพทย์และพยาบาล
2. หัวหน้าแผนก

และสามารถแสดงผลเวลาการทำงานได้มากที่สุด 1 เดือน เพื่อการวางแผนงานอย่างยืดหยุ่นและมีประสิทธิภาพมากยิ่งขึ้น

1.5 ระยะเวลาในการทำโครงการ

ตั้งแต่วันที่ 1 กุมภาพันธ์ 2562 ถึง 22 เมษายน 2562

ตารางที่ 1.1 แสดงระยะเวลาในการทำโครงการ

รายละเอียด	มกราคม 2562	กุมภาพันธ์ 2562	มีนาคม 2562	เมษายน 2562
1.ได้รับโครงสร้างของโครงการ	←→			
2.รวบรวมข้อมูลเรื่องการจัดเวร		←→		
3.ศึกษาโปรแกรมเพิ่มเติมและกำหนดโครงสร้างเพิ่มเติม			←→	
4.เริ่มสร้างต้นแบบและทดลองการใช้			←→	
5.แก้ปัญหาและสรุปผลโครงการ				←→
6.จัดทำรูปเล่มโครงการ				←→

1.6 ประโยชน์ที่คาดว่าจะได้รับ

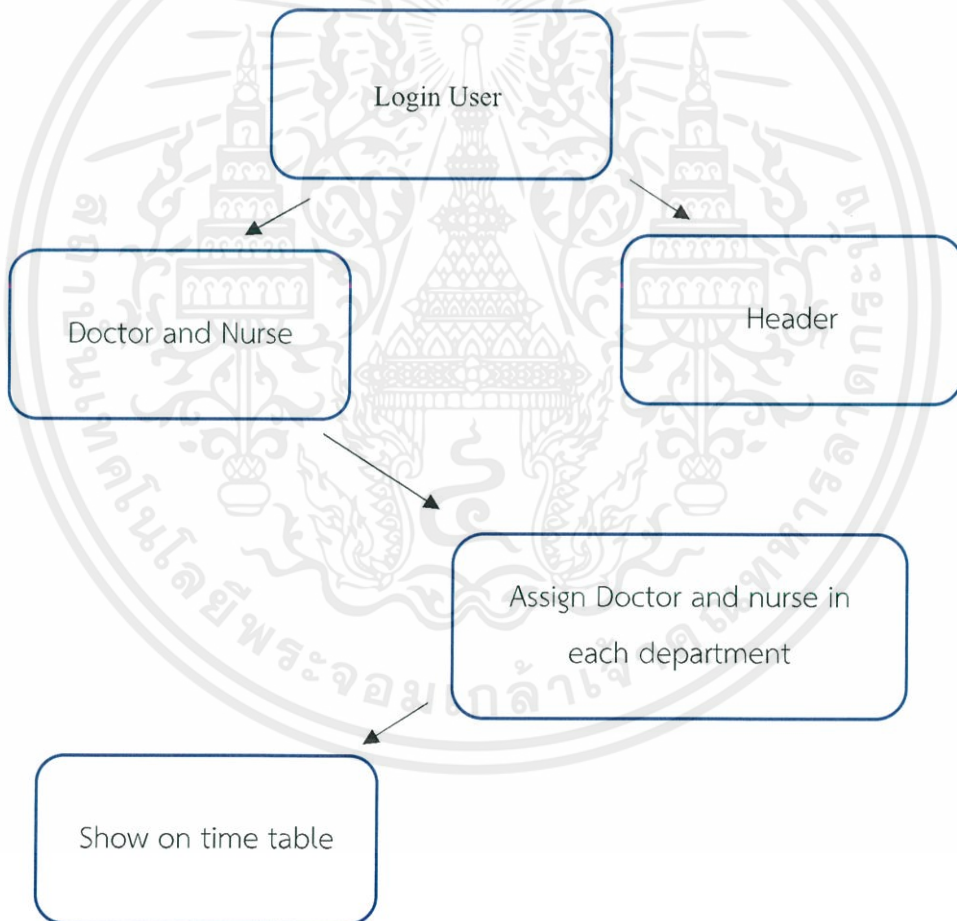
คาดว่าจะสามารถช่วยเหลือโรงพยาบาลที่กำลังประสบปัญหาในการจัดตารางเวลาได้

บทที่ 2

หลักการและทฤษฎี

โครงสร้างหลักของแอปพลิเคชันจัดเวรแพทย์และพยาบาลโดยแสดงในรูปที่ 2.1 ซึ่งประกอบด้วย ส่วนหลักคือ

1. ส่วน login/register สำหรับสมัครใช้งาน แบ่งเป็น 2 ส่วนคือ แพทย์และพยาบาล , หัวหน้าแผนก
2. ส่วน Assign work ส่วนนี้จะเป็นส่วนของหัวหน้าแผนกใช้ในการจัดการเวรให้แพทย์และพยาบาลในการทำหน้าที่ โดยสามารถจัดได้ 1 เดือน
3. ส่วน time table จะใช้แสดงว่าแต่ละวันมีแพทย์และพยาบาลต้องประจำเวรในแต่ละวัน



รูป 2.1 แสดงโครงสร้างหลักของแอปพลิเคชัน

2.1 ส่วนของ register

ในส่วนของการลงทะเบียนนั้น ผู้ใช้จะต้องทำการกรอกข้อมูล คือ อีเมล รหัสผ่าน ชื่อ นามสกุล แผนก(แบ่งเป็น ศัลยกรรม,สูติ-นารีเวชกรรม,กุมารเวชกรรม,อายุรกรรม,ศัลยกรรมกระดูก) และตำแหน่ง(หัวหน้าแผนก,หมอเฉพาะทางและพยาบาล) เป็นต้น โดยเมื่อลงข้อมูลครบแล้วแอปพลิเคชันจะทำการเก็บข้อมูลของผู้ใช้ทั้งหมดลงใน firebase โดยการแยกแยะจะใช้ ตำแหน่งในการกำหนดสถานะหน้าที่ของแต่ละคนซึ่งจะอธิบายในส่วนถัดไป

2.2 ส่วนของ header หรือ หัวหน้าแผนก

ในส่วนจะเป็นส่วนกลางในการกำหนดงาน โดยหัวหน้าแผนกจะสามารถดูตารางงานของทุกคนในแผนกได้และสามารถกำหนดงานให้ หมอและพยาบาลแต่ละคนได้ โดยเข้าไปตรงในส่วนกำหนดงานจากนั้น เลือกชื่อส่วนที่ทำงาน จากนั้นแอปพลิเคชันจะทำการบันทึกตารางงานของแต่ละคนลงใน firebasefirestore

2.3 ส่วนของ หมอเฉพาะทางและพยาบาล

ในส่วนนี้เปรียบเสมือนตารางงานส่วนตัวของแต่ละคนที่สามารถเข้าไปดูได้ทุกเมื่อ เพื่อเช็คตารางงานในแต่ละวัน

กล่าวโดยสรุปคือแอปพลิเคชันนี้ firebase ในการเก็บข้อมูลทั้งหมด เพื่ออัปเดตแบบ realtime ในส่วนของการแก้ไขปัญหาของแอปพลิเคชันนั้น จะมีการกำหนดเงื่อนไขอยู่ 2 อย่างหลัก (เนื่องจากเป็นสากลและใช้กันได้เกือบทุกโรงพยาบาล) คือ

1. ห้ามเข้าเวรติดต่อกันเกิน 2 ครั้ง เนื่องจากอาจเกิดการเหนื่อยล้า และ ข้อผิดพลาดในการปฏิบัติงาน
 2. ห้ามชั่วโมงทำงานเกิน 56 ชั่วโมง (ตามที่กฎหมายกำหนด) เนื่องจากว่า จำนวนที่ทำงานมากไปนั้นส่งผลโดยตรงกับบุคลากร ทั้งด้านจิตใจและสุขภาพร่างกาย
- เมื่อมีการจำกัดเงื่อนไขเหล่านี้ไว้แล้วจะส่งผลในสามารถลดอาการเหนื่อยล้าของแพทย์และพยาบาลได้อย่างมีนัยยะ

บทที่ 3

การออกแบบ

3.1 ฐานข้อมูลที่ออกแบบ

ใช้ Firebase Authentication สำหรับการเข้าใช้งานระบบ และใช้ Firebase Cloud Firestore เป็นฐานข้อมูลที่ใช้จัดเก็บข้อมูล

ตารางที่ 3.1 Staffs (พนักงาน)

FIELD_NAME	FIELD_TYPE	DESCRIPTION
ID	String	Firebase Firestore Document Id
email	String	ข้อมูลอีเมล
firstName	String	ชื่อ
lastName	String	นามสกุล
role	String	ตำแหน่ง เป็นได้สามค่า= (NURSE, DOCTOR, LEADER)
departmentId	String	ไอดีของแผนก

ตารางที่ 3.2 Departments (แผนก)

FIELD_NAME	FIELD_TYPE	DESCRIPTION
ID	String	Firebase Firestore Document ID
nameInitial	String	ชื่อแผนกย่อ
nameEng	String	ชื่อแผนก (ภาษาอังกฤษ)
nameTh	String	ชื่อแผนก (ภาษาไทย)
Description	String	รายละเอียดของแผนก

ตารางที่ 3.3 Works (ข้อมูลเวร)

FIELD_NAME	FIELD_TYPE	DESCRIPTION
ID	String	Firebase Firestore Document ID
workAssignedDate	Date (timestamp)	วันที่ที่ที่ได้รับมอบหมายงาน ตัวอย่าง 13/4/2019 เวลา 00:00 = 1555113600000L
assignedPeriod	String	ช่วงของเวร เป็นได้ 3 ค่า (MORNING, EVENING, NIGHT)
staffId	String	ไอดีของพนักงาน
departmentId	String	ไอดีของแผนก

3.2 Algorithm การทำงาน

Algorithm การเข้าใช้งานระบบ

1. รับค่า Email
2. รับค่า Password
3. ถ้า Email มีค่าว่างหรือยังไม่ได้กรอก ไปข้อ 7
4. ถ้า Password มีค่าว่างหรือยังไม่ได้กรอก ไปข้อ 2
5. ทำการ Login โดยใช้ email และ password
6. ถ้า Login สำเร็จแล้ว ไปข้อ 9 ถ้าไม่ ไปข้อ 10
7. แสดงข้อความว่า “กรุณากรอก Email” จากนั้น ไปข้อ 11
8. แสดงข้อความว่า “กรุณากรอก Password” จากนั้น ไปข้อ 11
9. ไปยังหน้าหลัก จากนั้น ไปข้อ 11
10. แสดงข้อความว่า “ข้อมูล Email หรือ Password ไม่ถูกต้อง” จากนั้น ไปข้อ 11
11. จบการทำงาน

Algorithm การสมัครบัญชีเข้าใช้งานระบบ

1. รับค่า Email
2. รับค่า Password
3. รับค่า Confirmation Password

4. รับค่า ชื่อ นามสกุล ตำแหน่งหน้าที่ และแผนก
5. ถ้า Email มีค่าว่างหรือยังไม่ได้กรอก ไปข้อ 9
6. ถ้า Password มีค่าว่างหรือยังไม่ได้กรอก ไปข้อ 10
7. ถ้า Password ไม่ตรงกันกับ Confirmation Password ไปข้อ 11
8. ถ้ายังไม่ได้กรอกชื่อ นามสกุล และถ้ายังเลือกตำแหน่งหน้าที่ และ แผนก ไปข้อ 12
9. แสดงข้อความว่า “กรุณากรอก Email” จากนั้นไปข้อ 17
10. แสดงข้อความว่า “กรุณากรอก Password” จากนั้นไปข้อ 17
11. แสดงข้อความว่า “Password และ Confirmation Password ต้องตรงกัน” จากนั้นไปข้อ 17
12. แสดงข้อความว่า “กรุณากรอกข้อมูลทั่วไป” จากนั้นไปข้อ 17
13. ทำการสมัครสมาชิก และบันทึกข้อมูลลงฐานข้อมูล
14. ถ้าสมัครสมาชิกสำเร็จแล้ว ไปข้อ 15 ถ้าไม่ไปข้อ 16
15. แสดงข้อความว่า “สมัครสมาชิกเสร็จสมบูรณ์” จากนั้นกลับสู่หน้าเข้าสู่ระบบ จากนั้นไปข้อ 17
16. แสดงข้อความขอผิดพลาดที่ได้รับจากเซิร์ฟเวอร์ จากนั้นไปข้อ 17
17. จบการทำงาน

Algorithm การออกจากระบบ

1. มีข้อความแจ้งเตือนว่า “ต้องการออกจากระบบใช่หรือไม่” ถ้าเลือก “ออกจากระบบ” ไปข้อ 2 ถ้าไม่ ไปข้อ 3
2. ทำการออกจากระบบ จากนั้นไปยังหน้าเข้าสู่ระบบ
3. จบการทำงาน

Algorithm การดึงข้อมูลเวรของตัวเองแต่ละวัน

1. รับวันที่ที่ต้องการดึงข้อมูลเวรของตัวเอง
2. ทำการดึงข้อมูลเวรของตัวเองจากวันที่ที่ต้องการ
3. นำข้อมูลไปแสดงผล
4. จบการทำงาน

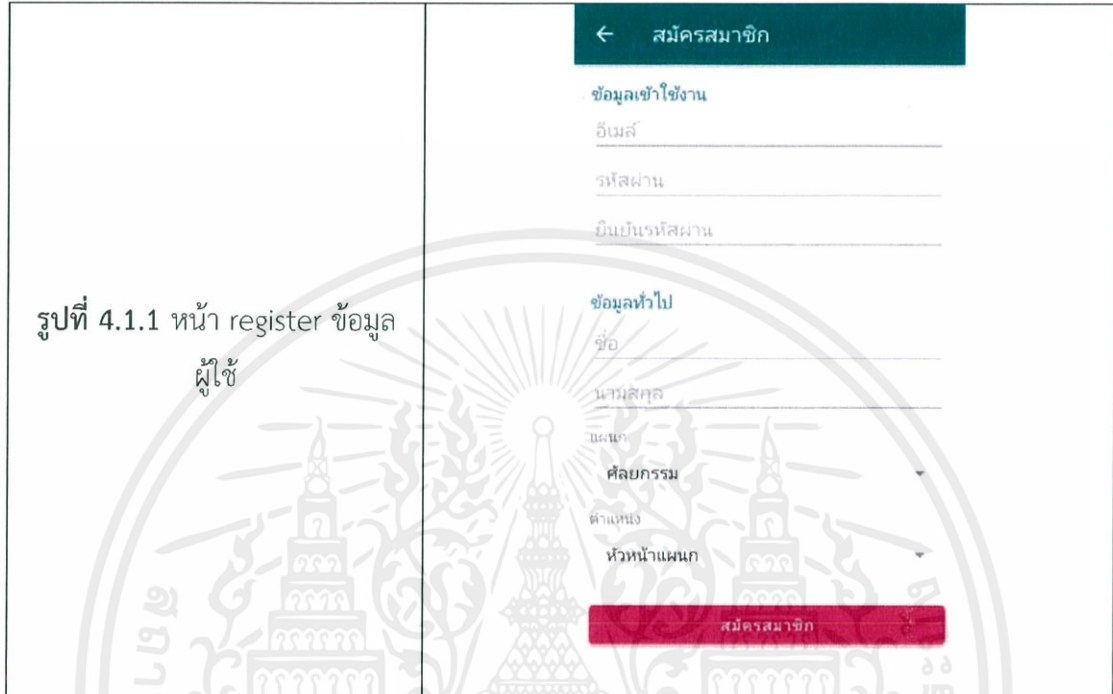
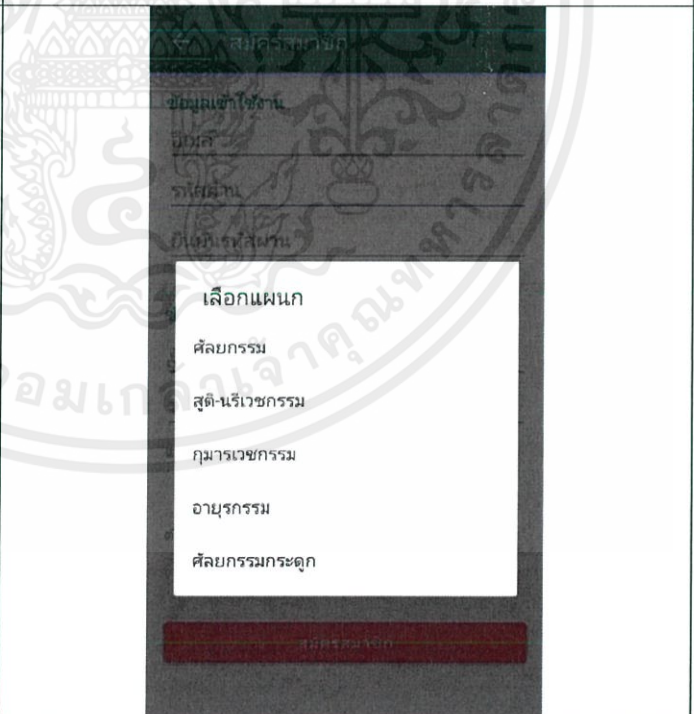
Algorithm การดึงข้อมูลแผนก

1. ทำการดึงข้อมูลแผนกทั้งหมด
2. นำข้อมูลที่ได้รับไปแสดงผล

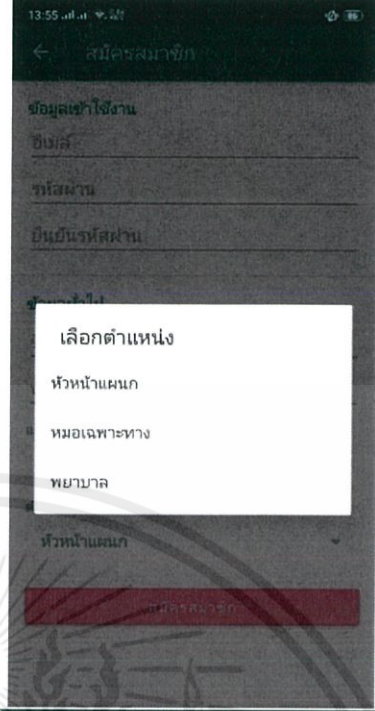
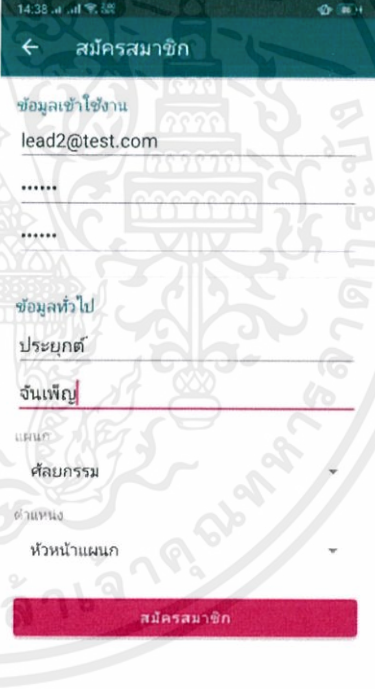
บทที่ 4

ผลการทดลอง

4.1 การทดลองสมัครสมาชิก

<p>รูปที่ 4.1.1 หน้า register ข้อมูล ผู้ใช้</p>	
<p>รูปที่ 4.1.2 เลือกแผนกจาก spinner</p>	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่น การค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

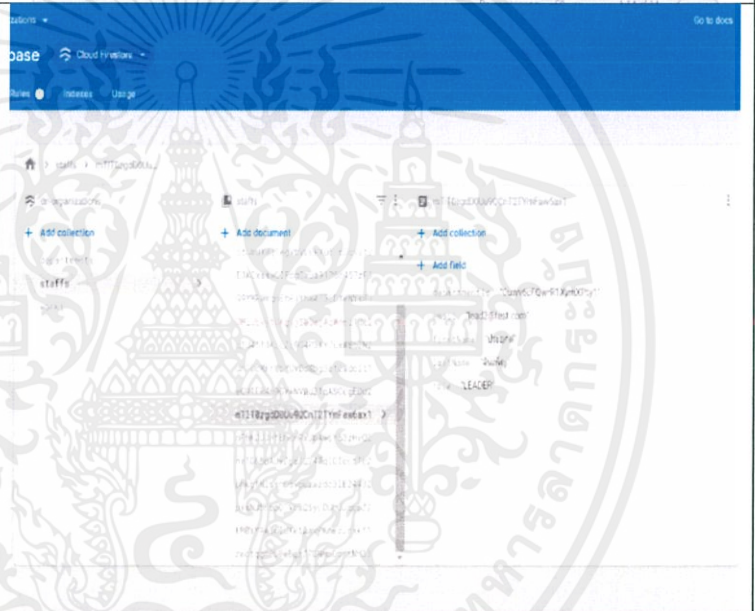
<p>รูปที่ 4.1.3 เลือกตำแหน่งของผู้ใช้</p>	
<p>รูปที่ 4.1.4 กรอกข้อมูลผู้ใช้</p>	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่น การค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1.5 แสดง email ใน firebaseAuth

Authentication				
name@text.com	✓	7 Apr 2019	7 Apr 2019	1N2akE6Bngj7Imz7Wd31Ev9G6
id@text.com	✓	17 Apr 2019	17 Apr 2019	5u0JX07ag65v13Fu61kxv7d2
name@text.com	✓	7 Apr 2019	7 Apr 2019	ESkDxawQ7pogkua91P5c457zF3
58010712gkxmbuacth	✓	22 Apr 2019	22 Apr 2019	69VM5vovNf9Jkx4T30Dkxv1
name4@text.com	✓	7 Apr 2019	7 Apr 2019	on10dMdeYV955Mog19E5oa91
name3@text.com	✓	7 Apr 2019	7 Apr 2019	wG9FFAe9G9uKwv21U4N0qg8
name2@text.com	✓	22 Apr 2019	22 Apr 2019	inT72zqD00uU9U172TmFae6
name@text.com	✓	7 Apr 2019	7 Apr 2019	nPKADJ4H8m64uJ6u6v537ru
name1@text.com	✓	7 Apr 2019	17 Apr 2019	exT0F6pA6u7g2466v0ku17d2
name@text.com	✓	17 Apr 2019	17 Apr 2019	gkxjN5v197poukzoc3BQ4402
name7@text.com	✓	7 Apr 2019	7 Apr 2019	peohvB8ep01Vc405E8D4H0zgg
name2@text.com	✓	7 Apr 2019	7 Apr 2019	1M5VYAG98kxwv9kxvzjwa11
name1@text.com	✓	7 Apr 2019	7 Apr 2019	zwcgpnF6e8g77001g0an0c33

รูปที่ 4.1.6 แสดงข้อมูลผู้ใช้ ใน firebasefirestore



4.2 interfaceในสำหรับแพทย์/พยาบาล

รูปที่ 4.2 ตารางในส่วนแพทย์/พยาบาล

สวัสดีคุณ น1 น1
ตำแหน่งใหม่ : พยาบาล

← พฤษภาคม 2019 →

อา จ อ พ พก ศ ส

21 22 23 24 25 26 27

เช้า (08:00-16:00)

น1 น1
25 เมษายน 2562

เย็น (16:00-24:00)

น1 น1
25 เมษายน 2562

4.3 interface ในสำหรับหัวหน้าแผนก

รูปที่ 4.3.1 ตารางในส่วนหัวหน้าแผนก (สามารถดูงานของลูกแผนกได้)

← จัดการเวร กำหนดเวร

← พฤษภาคม 2019 →

อา จ อ พ พก ศ ส

21 22 23 24 25 26 27

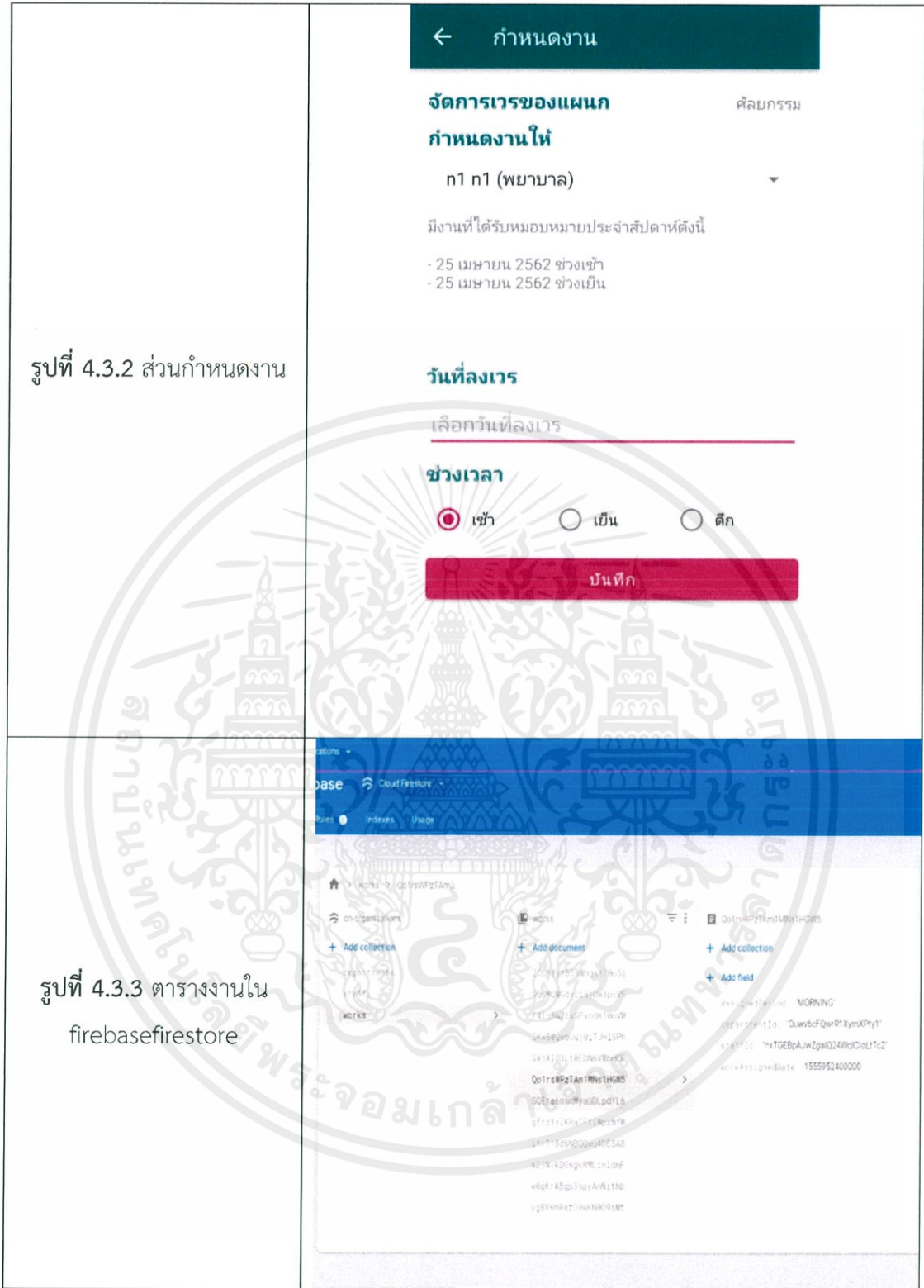
บุคคลที่เข้าเวรวันนี้

กิตติภัทร ชุมเรือง
ตำแหน่งที่ : NURSE
เข้าเวรช่วง : เย็น

น1 น1
ตำแหน่งที่ : NURSE
เข้าเวรช่วง : เย็น

น1 น1
ตำแหน่งที่ : NURSE
เข้าเวรช่วง : เช้า

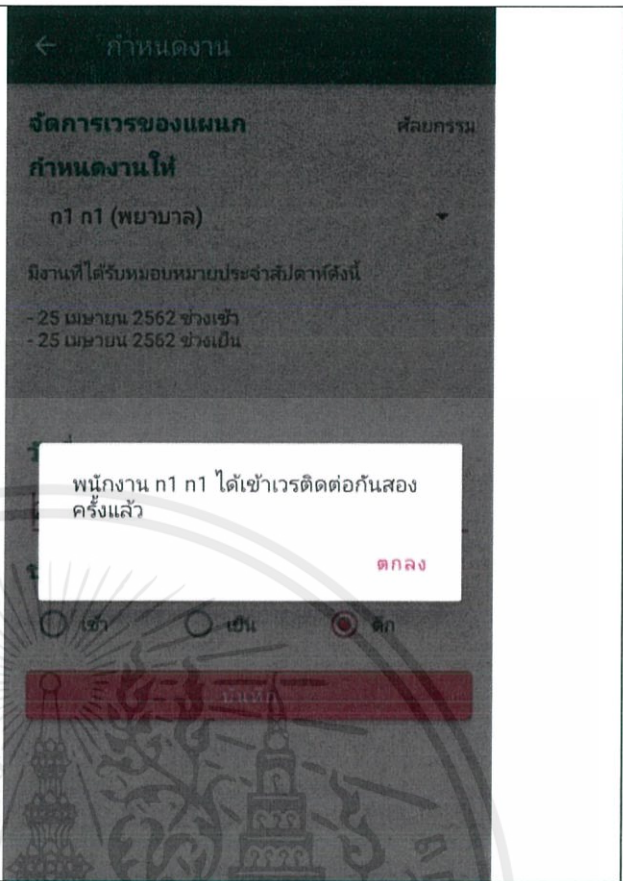

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3.2 ส่วนกำหนดงาน

รูปที่ 4.3.3 ตารางงานใน
firebasefirestore

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<p>รูปที่ 4.3.4 เมื่อลงเวรติดกัน 3 เวน</p>	
<p>รูปที่ 4.3.5 ชั่วโมงการทำงาน เกิน 56 ชั่วโมง</p>	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่น การค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

การทดลอง 4.1 เป็นการสมัครสมาชิก โดยในหน้าสมัครจะรับค่า email , password , ชื่อ-นามสกุล , แผนกและ ตำแหน่ง สร้าง user โดย firebaseauth และ สร้าง string array ใน firebasefirestore เพื่อเก็บข้อมูลของสมาชิกแต่ละประเภท

การทดลอง 4.2 เป็นหน้า main ของแพทย์และพยาบาล ในการใช้ดูตารางงานของตัวเอง

การทดลองที่ 4.3 เป็นตารางของหัวหน้าแผนกที่สามารถดูเวลาและกำหนดงานให้ลูกน้องได้ โดยสามารถกำหนดงานได้มากที่สุด 1 เดือนและแต่ละวันจะแบ่งเป็น 3 ช่วง คือ เช้า บ่าย ดึก จากนั้นแอปพลิเคชันจะทำการเก็บข้อมูลงานไว้ที่ firebasefirestore และทำการทดสอบเงื่อนไขต่างๆเช่น เมื่อลงเวร 3 เวรติดต่อกันจะทำการเตือนว่า ลง 2 เวรติดต่อกันไปแล้ว ต้องเว้น 1 เวร เพื่อประสิทธิภาพในการทำงาน และ ไม่สามารถลงเวลางานได้เกิน 56 ชั่วโมงต่อสัปดาห์

5.2 ปัญหาอุปสรรคและการแก้ไข

เนื่องจากประสบการณ์และความรู้ของผู้จัดทำยังไม่มีความพอ จึงทำให้เกิดข้อผิดพลาดในแอปพอสมควรและการออกแบบบางที่อาจจะใช้ยากไปบ้าง จึงทำให้ผู้ใช้งานอาจเกิดความสับสน ปัญหานี้แก้ไขได้ด้วยการหาข้อมูลเพิ่มเติมและปรึกษาผู้รู้

บรรณานุกรม

- [1] <https://www.voicetv.co.th/read/532519>
- [2] <https://docs.oracle.com/javase/tutorial/>
- [3] https://www.tutorialspoint.com/android/android_login_screen.html
- [4] <https://www.youtube.com/watch?v=wW-AAXOy4u4&t=791s>
- [5] <https://www.javatpoint.com/java-tutorial>
- [6] <https://developer.android.com/studio/write>
- [7] <https://www.youtube.com/watch?v=wW-AAXOy4u4&t=791s>



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 16 ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดในส่วน register (doCreateAccount())

```
private void doCreateAccount() {  
    showLoadingDialog();  
  
    //สร้าง account โดยรับค่า email และ password และสร้าง account  
    Task<AuthResult> createAccountTask = FirebaseAuth.getInstance()  
        .createUserWithEmailAndPassword(getEmail(), getPassword());  
  
    createAccountTask.continueWithTask(new Continuation<AuthResult,  
Task<Void>>()) {  
        @Override  
        public Task<Void> then(@NonNull Task<AuthResult> task) throws Exception  
        {  
            //ถ้า register ไม่สำเร็จทำ exception ถ้าสำเร็จทำการดึงข้อมูลประเภทแผนก staff จาก  
spinner  
            โดยสร้าง object ของ คลาส staff เพื่อเก็บข้อมูลแผนกลงใน FirebaseFirestore(returnค่า) และทำ  
            การดำเนินเมธอดต่อไป  
  
            if (!task.isSuccessful()) throw task.getException();  
            FirebaseUser firebaseUser = task.getResult().getUser();  
            String uid = firebaseUser.getUid();  
            String email = firebaseUser.getEmail();  
            Staff staff = new Staff(uid, getName(), getSurname(), email,  
                (StaffRole) spinnerRoleList.getSelectedItemAt(),  
                ((Department) spinnerDepartmentList.getSelectedItemAt()).getId()  
            );  
            return FirebaseFirestore.getInstance()  
                .collection(Staff.COLLECTION)
```

```

        .document(uid)

        .set(staff.toMap(), SetOptions.merge());
    }
}).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        dismissLoadingDialog();

        // ถ้าทำการ create account ใน FirebaseAuthication และ ทำการแทรกข้อมูล
        แผนก(role) ของ staff ใน FirebaseFirestore สำเร็จ จากนั้นทำการ สร้าง alertDialog เพื่อบอกว่า
        ทำการ register สำเร็จแล้ว แต่ถ้าไม่สำเร็จ exception แสดง error

        if (task.isSuccessful()) {
            showCreateAccountSucceededDialog();
        } else {
            showErrorMessage(task.getException().getLocalizedMessage());
        }
    }
});
}

```

เมธอดในส่วน login (doLogin())

```
private void doLogin() {
    showLoadingDialog();

    //ทำการ login โดยอิงข้อมูลจาก firebaseAuth ด้วยเมธอด
    signInWithEmailAndPassword ของ firebase
    Task<AuthResult> firebaseLoginTask =
    FirebaseAuth.getInstance().signInWithEmailAndPassword(
        getEmail(),
        getPassword()
    );

    firebaseLoginTask.continueWithTask(new Continuation<AuthResult,
    Task<DocumentSnapshot>>() {
        @Override
        public Task<DocumentSnapshot> then(@NonNull
        Task<AuthResult> task) throws Exception {
            //ถ้าทำการ login แล้วไม่สำเร็จ ก็ไปที่ FirebaseAuthException
            //ทำการตรวจสอบจาก Uid จาก firebase authentication
            if (!task.isSuccessful()) throw task.getException();
            return FirebaseFirestore.getInstance()
                .collection(Staff.COLLECTION)
                .document(task.getResult().getUser().getUid())
                .get();
        }
    }).addOnCompleteListener(new
    OnCompleteListener<DocumentSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot>
        task) {
```

```

dismissLoadingDialog();
try {

    //ทำการตรวจสอบว่ามีข้อมูลผู้ใช้อยู่หรือไม่ ถ้ามีให้ไปหน้าของ
    mainactivity ถ้าไม่ให้ไปหน้า register เพื่อลงทะเบียนข้อมูล
    DocumentSnapshot document = task.getResult();
    if (document != null && document.exists()) {

        Staff staff = new Staff(document);
        sessionManager.create(staff);
        launchMainActivity();
    } else {
        sessionManager.deleteAndAlsoDoFirebaseLogout();
        launchRegisterActivity();
    }
} catch (Exception e) {
    //getRealErrorMessage
    //Input "Error: errorMessage"
    //Output "errorMessage"

    String realErrorMessage =
    e.getLocalizedMessage().split(":")[1].substring(1);
    showErrorMessage(realErrorMessage);
}
});
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 20 ำานการค้ำ
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดในส่วน logout (doLogout())

```
private void doLogout() {
```

//ทำการแจ้งเตือน(AlertDialog) ว่า ต้องการออกจากระบบหรือไม่ โดยแสดงตัวเลือก
ถ้าคลิก “ออกจากระบบ” จะทำการดึงเมธอด.deleteAndAlsoDoFirebaseLogout(); มาใช้เพื่อทำ
การ logout จากระบบและเปิด LancherActivity แต่ถ้าคลิก “ยกเลิก” ก็กลับหน้าปกติ

```
new AlertDialog.Builder(this)
    .setMessage("ต้องการออกจากระบบหรือไม่")
    .setPositiveButton("ออกจากระบบ", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            sessionManager.deleteAndAlsoDoFirebaseLogout();
            startActivity(new Intent(MainActivity.this, LancherActivity.class));
            finish();
        }
    })
    .setNegativeButton("ยกเลิก", null)
    .show();
}

public void deleteAndAlsoDoFirebaseLogout() {
    // Clear staff data from SharedPreferences
    getSharedPreferences().edit().clear().apply();

    // Do FirebaseAuthLogout
    FirebaseAuth.getInstance().signOut();
}
```

เมธอดในส่วน inquiryWork (inquiryWorkAssigned)

```
//เมธอดนี้รับค่าเวลาเข้าไป (LocalDateTime date) และทำการตรวจสอบและรับ work
จาก firebasefirestore
private void inquiryWorkAssigned(LocalDateTime date) {
    WorkAssignService.getInstance().getWorkAssign(sessionManager.get().getId(), date,
new Result.OnCompleteListener<List<WorkAssign>>() {
    @Override
    public void onComplete(@NonNull Result<List<WorkAssign>> result) {
        if (result.isSuccessful()) {
            //เรียงข้อมูลโดยโชว์หัวหน้าแผนก หมอ พยายามตามลำดับ
            Collections.sort(result.getData(), new Comparator<WorkAssign>() {
                @Override
                public int compare(WorkAssign o1, WorkAssign o2) {
                    int firstPeriod = o1.getPeriod().ordinal();
                    int lastPeriod = o2.getPeriod().ordinal();
                    return Integer.compare(firstPeriod, lastPeriod);
                }
            });
            //set adapter โดยใช้ workAssignedListAdapter ในการรับค่าข้อมูล ของ work จาก
firebase
            workAssignedListAdapter.submitList(result.getData());
        } else {
            Toast.makeText(MainActivity.this,
result.getException().getLocalizedMessage(), Toast.LENGTH_SHORT).show();
        }
    }
});
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 22 ้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดในส่วน assignwork (informAssignWork())

//ทำ assign work ให้ถูกต้อง ถ้าไม่ แสดง error message ถ้าถูกต้องทำเมธอด doAssignWork()

```
private void informAssignWork() {
    int radioButtonId = radioGroupPeriodOfWork.getCheckedRadioButtonId();
    String dateAssignedString = editTextWorkDate.getText().toString();
    if (radioButtonId == -1) return;
    if (dateAssignedString.isEmpty()) return;

    final Date dateAssigned;
    try {
        dateAssigned = dateFormat.parse(dateAssignedString);
    } catch (ParseException e) {
        e.printStackTrace();
        return;
    }

    final Staff staff = ((Staff) spinnerAssignee.getSelectedItem());
    if (staff == null) return;

    //กำหนดช่วงเวลาทำงาน work period โดยทำการกำหนดค่าเงื่อนไขให้แต่ละปุ่มกด
    WorkPeriod workPeriod = null;
    if (radioButtonId == R.id.radioButtonPeriodMorning) {
        workPeriod = WorkPeriod.MORNING;
    } else if (radioButtonId == R.id.radioButtonPeriodEvening) {
        workPeriod = WorkPeriod.EVENING;
    } else if (radioButtonId == R.id.radioButtonPeriodNight) {
        workPeriod = WorkPeriod.NIGHT;
    }

    //ถ้างาน(วันและเวลานั้น)ถูกมอบหมายไปแล้วให้แสดงข้อความ
    if (workAvailable.isAlreadyAssigned(LocalDateTimeUtils.fromDate(dateAssigned),
    workPeriod)) {
        showMessage("งานที่กำลังมอบหมายให้ ได้รับมอบหมายเรียบร้อยแล้ว");
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 23 ้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return;
    }
    //ถ้ามีการลงงานติดต่อกันสองเวอร์แล้ว ไม่สามารถลงงานได้ในเวลาติดต่อกัน(เวอร์ที่3) และแสดง
    ข้อความเตือน
    if
    (workAvailable.isAlreadyWorkTwoTimesInARow(LocalDateTimeUtils.fromDate(dateAssigned), workPeriod)) {
        showMessage("พนักงาน " + staff.getFullName() + " ได้เข้าเวรติดต่อกันสองครั้งแล้ว");
        return;
    }

    //เมื่อเวลางานว่างแสดงข้อความกำหนดค่า ถ้าคลิก ตกลง ให้ทำเมธอด doAssignWork ต่อไป ถ้า
    คลิก ยกเลิก จะเป็น null และกลับเป็นหน้าเดิม

    String messageAlert = "ต้องการกำหนดงานให้กับข้อมูลดังกล่าวหรือไม่ ?\n\n" +
        "ชื่อ " + staff.getFullName() + "\n" +
        "เวร " + workPeriod.getPeriodTitle() + "\n" +
        "วันที่ขึ้นเวร " + editTextWorkDate.getText().toString();
    final WorkPeriod period = workPeriod;
    new AlertDialog.Builder(this)
        .setTitle("ยืนยันข้อมูล")
        .setMessage(messageAlert)
        .setPositiveButton("ตกลง", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                doAssignWork(staff, dateAssigned, period);
            }
        })
        .setNegativeButton("ยกเลิก", null)
        .show();
    }

```

//ในทำการ assign work เป็นการเพิ่มงานลงไป ใน FirebaseFirestore ถ้าสำเร็จจะแสดงข้อความ
ออกผ่านเมธอด (showAssignWorkSucceededDialog();) แต่ถ้าไม่สำเร็จก็จะแสดงข้อความที่ได้จาก
exception

คือ

```
Toast.makeText(AssignWorkActivity.this,e.getLocalizedName(),Toast.LENGTH_SHORT  
.show());
```

```
private void doAssignWork(Staff staff, Date dateAssigned, WorkPeriod period) {  
    showLoadingDialog();  
    Work work = new Work(staff, dateAssigned, period, department);  
    FirebaseFirestore.getInstance()  
        .collection(Work.COLLECTION)  
        .add(work.toMap())  
        .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {  
            @Override  
            public void onSuccess(DocumentReference documentReference) {  
                dismissLoadingDialog();  
                showAssignWorkSucceededDialog();  
            }  
        })  
        .addOnFailureListener(new OnFailureListener() {  
            @Override  
            public void onFailure(@NonNull Exception e) {  
                dismissLoadingDialog();  
                Toast.makeText(AssignWorkActivity.this, e.getLocalizedName(),  
                    Toast.LENGTH_SHORT).show();  
            }  
        });  
}
```

LuncherActivity (Splash Screen)

```
public class LauncherActivity extends AppCompatActivity implements  
FirebaseAuth.AuthStateListener {
```

```
    private SessionManager sessionManager;
```

```
    @Override
```

```
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        sessionManager = SessionManager.getInstance(this);  
        FirebaseAuth.getInstance().addAuthStateListener(this);  
    }
```

```
    @Override
```

```
    public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  
        FirebaseAuth.getInstance().removeAuthStateListener(this);  
        Intent intent;  
  
        if (firebaseAuth.getCurrentUser() != null && sessionManager.hasCreated()) {  
            intent = new Intent(this, MainActivity.class);  
        } else {  
            sessionManager.deleteAndAlsoDoFirebaseLogout();  
            intent = new Intent(this, LoginActivity.class);  
        }  
  
        startActivity(intent);  
        finish();  
        overridePendingTransition(android.R.anim.fade_in, android.R.anim.fade_out);  
    }  
}  
  
//sessionmanager จะเป็นตัวเก็บข้อมูลตอน login เพื่อนำข้อมูลมาใช้
```