**ระบบดูแลผู้สูงอายุและผู้ป่วยติดเตียงส่วนการพัฒนาส่วนหน้าที่มองเห็นได้ทันที**

**Elderly Visiting System and Bedridden care part front end development**

**นางสาวนภาพร ตั้งใจ**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาตร์บัณฑิต**

**สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2565**

ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบดูแลผู้สูงอายุและผู้ป่วยติดเตียงส่วนการพัฒนาส่วนหน้าที่มองเห็นได้ทันที

Elderly Visiting System and Bedridden care part front end development

ผู้จัดทำ

1. นางสาวนภาพร ตั้งใจ รหัสนักศึกษา 62010462

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ อาจารย์ที่ปรึกษา

(ผศ.เกียรตินรงค์ ทองประเสริฐ)

**ระบบดูแลผู้สูงอายุและผู้ป่วยติดเตียงส่วนการพัฒนาส่วนหน้าที่มองเห็นได้ทันที**

นางสาวนภาพร ตั้งใจ 62010462

ผศ.เกียรตินรงค์ ทองประเสริฐ อาจารย์ที่ปรึกษา

ปีการศึกษา 2565

**บทคัดย่อ**

โครงงานนี้จัดทำเพื่อพัฒนาแอพพลิเคชันที่สามารถให้ความสะดวกสบายแก่การดูแลผู้ป่วยติดเตียงหรือผู้สูงอายุต่างๆ ทั้งนี้เพื่อให้ผู้ที่สนใจสามารถศึกษาหรือทดลองใช้งานในการดูแลต่างๆ เพื่อเพิ่มความสะดวกสบาย ความไม่ยุ่งยาก และความเป็นระเบียบวินัยแก่ตัวผู้ดูแล และผู้สูงอายุผู้ป่วยติดเตียงเราจะจัดทำให้เป็นระบบระเบียบยิ่งขึ้น โดยหลักๆจะมีไม่กี่ขั้นตอนในการเข้าดูแล หรือเข้าถึงผู้ป่วย เรียงลำดับก็คือ Care Manager > Care Giver > Elder > Longtercare ซึ่งระบบของเราหรือแอพพลิเคชันของเราที่จะมาช่วยก็คือ แอพพลิเคชัน Longterm care ดังนั้นแอพพลิเคชันนี้จะมาช่วยในการจดบันทึกต่างๆเพื่อเสริมสร้างความเป็นระเบียบแก่ผู้สูงอายุ เพราะภายในแอพพลิเคชันเราจะมี Care Plan คือแผนการดูแล ในระบบของเราคือจะมี Care plan กระจายไปยังผู้สูงอายุ เพื่อให้ทุกคนได้ดูแลตาม Care plan ที่วางไว้ หลักๆหน้าที่ของแอพพลิเคชันคือ บันทึกในสิ่งที่ Care Giver ไปปฏิบัติต่อ Elder และอำนวยความสะดวกต่อ Care Giver

**Elderly Visiting System and Bedridden care**

**ABSTRACT**

# กิตติกรรมประกาศ

นภาพร ตั้งใจ

# บทที่ 1

# บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันโลกของเรามีผู้คนที่เป็นผ้สูงอายุเยอะและเป็นผู้ป่วยที่ติดเตียงที่ต้องการคนดูแลหรือคนเฝ้าอย่างใกล้ชิดเยอะ และปัจจุบัน Care Giver หรือคนดูแลผู้ป่วยผู้สูงอายุอาจจะไม่เพียงพอ หรือต้องการความช่วยเหลือทางด้านเทคโนโลยี เพื่อที่จะมาอำนวยความสะดวก แก่การดูแลการเฝ้าต่างๆ เพื่อให้มันเป็นระบบ เป็นระเบียบมากยิ่งขึ้น

ทางเราได้เห็นถึงความสำคัญในการสร้างโครงงานนี้ขึ้นมาเพื่อทำให้การดูแลผู้ป่วยหรือการเฝ้าต่างๆ มีความสสะดวกสบายมากขึ้น แก่การจดบันทึกในแต่ละวันที่ Care Giver ได้ไปปฏิบัติต่อผู้สูงอายุหรือผู้ป่วยติดเตียง หรือบางวันอาจจะให้ผู้สูงอายุได้ดูแลตัวเองผ่าน Care plan ซึ่งก็จะมีอยู่ในแอพพลิเคชันของเรา เราจะคิดที่จะสร้างแอพพลิเคชันที่มีชื่อว่า Longtermcare ขึ้นมา

1.2 วัตถุประสงค์

1) เพื่อพัฒนาแอพพลิเคชันที่สามารถเรียกใช้งาน Care plan

2) เพื่อพัฒนาแอพพลิเคชันที่สามารถบันทึกสิ่งต่างๆที่ต้องการบันทึกได้

3) เพื่อที่จะสามารถให้ Care Manager Care Giver Elder เชื่อมต่อกันได้

1.3 ขอบเขตของโครงงาน

1) สามารถใช้งานแอพพลิเคชันได้

2) สามารถบันทึกในสิ่งที่ Care Giver ไปปฏิบัติต่อ Elder ได้

3) สามารถที่จะดู Care plan หรือแผนการดูแลได้

4) สามารถให้ผู้สูงอายุออกกำลังกายตามคลิปวิดิโอได้

5) สามารถให้ Care Manager Care Giver Elder เชื่อมต่อกันได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถสร้างแอพพลิเคชันที่สามารถใช้งานดูแลผู้ป่วยติดเตียงหรือผู้สูงอายุได้อย่างง่ายดาย

2) สามารถสร้างแอพพลิเคชันที่สามารถให้ผู้สูงอายุออกกำลังกายด้วยตัวเองได้

3) สามารถบันทึกสิ่งต่างๆที่ Care Giver ไปปฏิบัติต่อผู้สูงอายุหรือผู้ป่วยติดเตียงในแต่ละวันได้

1.5 แผนการดำเนินงาน

**ตาราง 1.5 แผนการดำเนินงาน**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| หัวข้อ | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. |
| กำหนดขอบเขต |  |  |  |  |  |  |  |  |
| ศึกษาและออกแบบ |  |  |  |  |  |  |  |  |
| พัฒนาระบบ |  |  |  |  |  |  |  |  |
| ทดสอบและปรับปรุง |  |  |  |  |  |  |  |  |
| จัดเล่มรายงานและนำเสนอ |  |  |  |  |  |  |  |  |

**บทที่ 2**

**ทฤษฎีที่เกี่ยวข้อง**

2.1 React

React คือ JavaScript Library ที่ทีม Facebook เป็นคนพัฒนาขึ้นมาและเปิดให้คนทั่วไปมาใช้ฟรีซึ่งเว็บไซต์ในปัจจุบันของ Facebook.com ก็ใช้ React ขั้นตอนการเขียน React มี 3 Concept

1. Component ส่วนต่าง ๆ ในเว็บจะมองเป็น Component

2. State ข้อมูลที่อยู่ใน Component แต่ละชิ้นเรียกว่า State

3. Props ข้อมูลที่ถูกส่งต่อจาก Component ชั้นบนลงไปชั้นล่างเรียกว่า Props (Properties)

สำหรับการเรียน Component นั้น เหมือนกับการเขียน HTML React ใช้สิ่งที่เรียกว่า JSX ในการแสดงผลเว็บไซต์จะเหมือน HTML แตกต่างตรงเขียนเข้าไปในไฟล์ JavaScript แทนไฟล์ HTML

ข้อดีของ React

1. เทรนด์เว็บไซต์ได้กลายเป็น API เว็บไซต์ในปัจจุบันมี API เช่น Facebook, Google, Twitter etc.

2. Concept เข้าใจง่าย

3. React เกิดมาหลายปีและยงัคงมีคนให้ความสนใจอยู่มาก

4. เครื่องมือทำงานจะต้องศึกษาเครื่องมือเพิ่มเติมเพื่อใช้ React

5. React Native เป็นการเขียน JavaScript แล้วน ามาแปลงเป็น App แบบ Native ได้ทั้งบนAndroid และ IOS

6.ผู้ใช้สามารถหา code ให้ศึกษาได้ง่ายมี Best Practice จากเว็บไซต์จริงให้ศึกษา

7. Community แข็งแกร่งหากมีคำถามจะมีคนตอบ

2.2 React Native Expo

React Native Expo เป็น framework และแพลตฟอร์มที่พัฒนาขึ้นจาก React Native เป็นตัวช่วยให้เราพัฒนาแอพพลิเคชั่นได้อย่างรวดเร็ว และง่ายยิ่งขึ้นสามารถใช้ JavaScript หรือ TypeScript ก็ได้ ซึ่งสามารถพัฒนาได้ทั้ง Android และ iOS ได้พร้อมกัน ไม่ต้องเข้ามาจัดการเขียน Native Module และมี Expo CLI ช่วยในการดู log การทำงานต่างๆ และการทดสอบ Mobile และความสามารถพิเศษอีกอย่างของ Expo คือให้เราสามารถ Run แอพพลิเคชั่น Expo Go ของเราได้ผ่านมือถือจริง ได้โดยไม่ต้องเชื่อมต่อ USB เลย เครื่องคอมที่ใช้พัฒนาและมือถืออยู่ในต้องอยู่วง Lan เดียวกันจึงจะสามารถแสดงผลลัพธ์ได้

ข้อดีของ React-Native

* ไม่ต้องรอ Compile App แค่ทำการ Save ก็จะเห็นผลลัพธ์
* Dev Tools, Debugging tools เครื่องมือที่ช่วยให้เรา dex ได้ productive มากขึ้นและรวมถึง tools ที่ช่วยแก้ปัญหาได้เร็วขึ้น
* สามารถเขียนทีเดียวใช้ได้หลาย Platform

ข้อเสียของ React-Native

* ใช้เวลา Setup นาน คือต้อง setup ตั้งแต่พวก JS tools รวมถึงทุกอย่างที่ iOS/Android Native Developer ใช้งาน เช่น XCode, Android SDK, Android Emulator เป็นต้น ซึ่งใช้เวลาถึง 2–3 ชั่วโมงในการติดตั้งครั้งแรกและยังต้องติดตั้งตัวเครื่องมือต่างๆเพิ่มเติมในการทำงาน
* Cross platform แต่ต้องการเครื่องที่ Compile ภาษาที่ต้องการได้ หรือก็คือต้องการ Mac เพื่อที่จะได้สร้าง iOS Application ได้นั่นเอง

ต้องมีพื้นฐานภาษา javascript ระดับหนึ่ง

2.3 Nativebase

จากที่ลองศึกษา React Native ซึ่งเป็น Framework สำหรับพัฒนา Mobile Applications ที่สามารถทำงานได้ทั้ง Android และ iOS พยายามค้นดูว่า น่าจะมี Framework อะไรบ้างที่ช่วยทำให้หน้าตา App ของเราสวยงามและน่าใช้งานมากขึ้น ซึ่งหากเปรียบเทียบกับการพัฒนา Web Application คงจะเป็นกลุ่ม CSS Framework เช่น Bootstrap Foundation GUMBY เป็นต้น



NativeBase ซึ่งเป็น Framework ที่ช่วยสร้าง Components ต่างๆ มากมายสำหรับพัฒนา App โดยสามารถใช้ React Native เป็นฐานหลักสำหรับพัฒนา App ได้

ข้อดีอย่างหนึ่งที่เห็นได้ชัดสำหรับ NatvieBase นั้นก็คือ หากจะต้องสร้างอะไรซักอย่าง ซึ่งหากเขียนสคริปต์ด้วย React Native แบบเพียวๆไม่มีของผสม ก็คงต้องเขียนกันหลายบรรทัด แต่สำหรับ NativeBase เขียนสคริปต์ไม่กี่บรรทัดก็สามารถสร้างได้เหมือนกัน

2.4 Expo Go

เครื่องมือ run และแสดงผลที่เร็วที่สุด ในการเริ่มต้นใช้งานคือการใช้ แอปไคลเอนต์ Expo Goบนอุปกรณ์ iOS หรือ Android ของคุณ สามารถใช้งานผ่านแอปที่ให้บริการผ่าน Expo CLI สามารถใช้ผ่านอุปกรณ์จริงทั้ง iOS App Store และ Android Play Store

2.5 MongoDB

เป็น NoSQL Database ที่สามารถใช้ข้าม Platform ได้ ซึ่งหลายคนอาจจะสงสัยว่าแล้ว NoSQL มันคือการเข้าถึงข้อมูลต่างๆ โดยจะใช้เทคนิคของการดึงข้อมูลผ่าน Key Pair Value ซึ่งข้อดีของตัวนี้ เหมาะกับองค์กรที่อยากจะเข้าถึงข้อมูลได้อย่างรวดเร็ว อีกทั้งยังเหมาะกับการทำ Big Data

1. ตัว MongoDB สามารถที่จะสร้างเป็น Cluster เพื่อที่จะตอบสนองของคำว่า High Availability (HA) ได้ ซึ่งเราอาจจะเลือก Region ที่เราอยากจะ Deploy บน Cloud Provider

2. รวดเร็วในการเข้าถึงข้อมูล เพราะว่าตัว Database ของมันไม่มี Schema ซึ่งจะต่างกับ SQL โดยพวกนั้นจะอิงจากฐานข้อมูลที่มาจาก Table

3. สามารถทำ Auto Scale ได้ไม่ว่าจะมีการใช้งานมากน้อยแค่ไหน ตัวมันก็สามารถ Adapt กับ Environment นั้นๆ ได้ครับ

4. รองรับ Multiple Cloud Provider ซึ่งข้อดีข้อนี้มันจะทำให้ Database ของเรามี High Availability มากขึ้นโดยเราไม่จำเป็นที่จะต้องยึดติดกับ Cloud Provider เจ้าใดเจ้าหนึ่ง

2.2.4 GraphQL

GraphQL เป็นเว็บเซอร์วิสที่เข้ามามีบทบาทสำคัญในการพัฒนา API ปัจจุบัน เพื่อทดแทนความบกพร่องบางประการของเว็บเซอร์วิสแบบ RESTful ทั้งการประหยัดแบนวิธที่ใช้ในการสื่อสาร รวมถึงความเรียบง่ายต่อการใช้งาน นั่นทำให้หลายบริษัทขนาดใหญ่ เช่น Facebook หรือ GitHub เลือกใช้ GraphQL เป็น API สามารถใช้งานร่วมกับหลายๆ Database เช่น SQL, MongoDB

GraphQL ที่ใช้บน Server ช่วยให้คุณสามารถกำหนดวิธี Fetch Data จากนั้นเมื่ออยู่บน Client จะสามารถ Execute Query ที่มีรูปแบบข้อมูลตามที่คุณต้องการ และข้อมูลที่คุณจะได้รับกลับมา ก็จะมีรูปแบบตามที่คุณ Query

* Query คือ การดึงข้อมูลต่างๆ ถ้าเทียบกับ Restful API ก็คงเป็น GET
* Mutation คือ การเปลี่ยนแปลงข้อมูล ถ้าเทียบกับ Restful API ก็คงเป็นพวก POST, PUT, DELETE
* Schema, Type definition คือ การนิยามชนิดของข้อมูลที่รับเข้า และส่งออก
* Resolver อันนี้คือ ส่วนของการทำงานของทั้งตั; Query, Mutation และ Type definition นั่นเอง

2.6 React navigation

package ที่จะเข้ามาช่วย navigate เพื่อเข้าถึงไปยังหน้าต่าง ๆ ภายในแอพ เช่น เมื่อเปิดแอพขึ้นมาแล้วให้แสดงหน้าอะไร หรือเมื่อ user กดปุ่มแล้วให้ไปหน้าไหน ช่วยให้การจัดการและการเขียน Front end ง่ายและช่วยลดการเขียนโค้ดลงไปได้อีก

2.7 Android Studio

Android Studio เป็น IDE Tool จาก Google ไว้พัฒนา Android สำหรับ Android Studio เป็น IDE Tools ล่าสุดจาก Google ไว้พัฒนาโปรแกรม Android โดยเฉพาะ โดยพัฒนาจากแนวคิดพื้นฐานมาจาก InteliJ IDEA คล้าย ๆ กับการทำงานของ Eclipse และ Android ADT Plugin โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน Smart Phone แต่ล่ะรุ่น สามารถแสดงผลบางอย่างได้ทันทีโดนไม่ต้องทำการรัน App บน Emulator รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน

Android SDK

Android Software Development Kit (Android SDK) เปรียบเสมือน Library ที่ใช้ในการพัฒนา Application สำหรับ Android เนื่องจากตัว Android มีหลายเวอร์ชั่นและแต่ละเวอร์ชั่นมี Feature, GUI ที่ไม่เหมือนกันทำให้เกิด Android SDK ออกมาหลายเวอร์ชั่นให้เลือกใช้งาน

**บทที่ 3**

**การออกแบบ และการพัฒนา**

3.1 วิเคราะห์ความต้องการ

- ทดลองคิดความต้องการที่ผู้ใช้จะต้องได้ใช้งานจริงๆ

- ประชุมกับ Super User ของแต่ละอนามัยเพื่อวิเคราะห์ความต้องการ

3.2 Get Requirement

**ตารางที่ 3.2 Requirement**

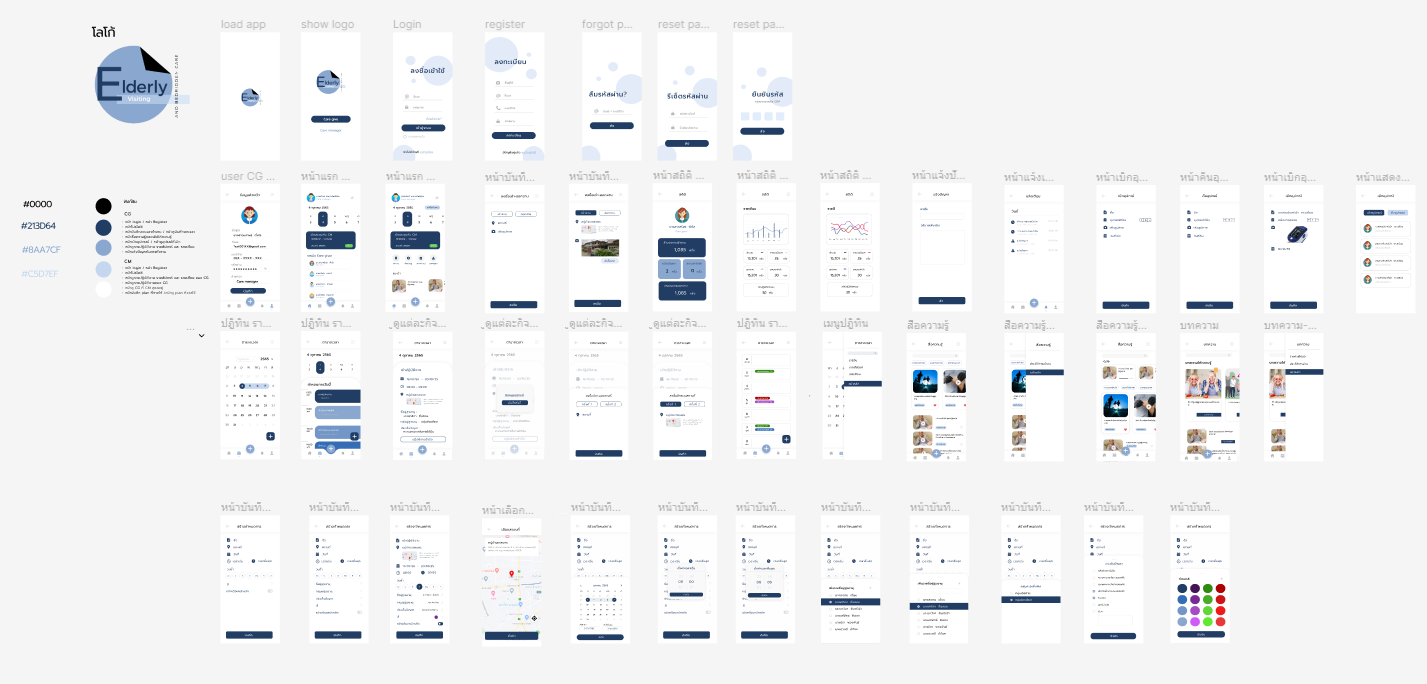
|  |  |  |
| --- | --- | --- |
| **Req ID** | **Feature or Module Name** | **Description** |
| R1 | ลงทะเบียน | CM หรือ CG สามารถลงทะเบียนใส่ข้อมูลในระบบ - ชื่อผู้ใช้ - อีเมล - เบอร์โทร - รหัสผ่าน ครบถ้วนเรียบร้อย |
| R2 | ลงทะเบียน | CM หรือ CG สามารถลงทะเบียนใส่ข้อมูลในระบบ - ชื่อผู้ใช้ - อีเมล - เบอร์โทร - รหัสผ่าน ไม่ครบถ้วน แสดง " กรุณากรอกข้อมูลให้ครบทุกช่อง " |
| R3 | ลงชื่อเข้าใช้ | CM หรือ CG ลงชื่อเข้าสู่ระบบ กรอกข้อมูลครบทั้ง ชื่อผู้ใช้ และ รหัสผ่านที่ถูกต้อง |
| R4 | ลงชื่อเข้าใช้ | CM หรือ CG ลงชื่อเข้าสู่ระบบ กรอกข้อมูลไม่ครบ แสดง " กรุณากรอกข้อมูลให้ครบทุกช่อง " |
| R5 | ลงชื่อเข้าใช้ | CM หรือ CG ลงชื่อเข้าสู่ระบบ กรอกข้อมูลชื่อผู้ใช้ หรือ รหัสผ่านไม่ถูกต้อง แสดง " ชื่อผู้ใช้หรือรหัสผ่านไม่ถูกต้อง โปรดลองอีกครั้ง " |
| R6 | หน้าแรก ของ CM | - แสดงวันปัจจุบัน - แสดงกำหนดการสำคัญของวันที่ใกล้จะถึง เมื่อผ่านกำหนดการก็แสดง กำหนดการต่อไป - แสดงรายชื่อ CG ที่ดูแลอยู่ |
| R7 | หน้าแรก ของ CG | - แสดงวันปัจจุบัน - แสดงกำหนดการสำคัญของวันที่ใกล้จะถึง เมื่อผ่านกำหนดการก็แสดง กำหนดการต่อไป - ปุ่มสื่อการเรียนรู้ - ปุ่มบทความให้ความรู้ - ปุ่มเบิกอุปกรณ์ - ปุ่มแจ้งปัญหาในการปฎิบัติงาน - แสดงสื่อการเรียนรู้และบทความล่าสุด |
| R8 | โปรไฟล์ผู้ใช้ | แสดงข้อมูลรายละเอียดของผู้ใช้ ที่ได้ลงทะเบียนไว้ |
| R9 | โปรไฟล์ผู้ใช้ | สามารถแก้ไขและเพิ่มข้อมูลรายละเอียดของผู้ใช้ได้ |
| R10 | บันทึกลงเวลาทำงานของ CG | สามารถบันทึกข้อมูลเริ่มการปฎิบัติงานได้ - ลงวันที่/เดือน/ปีที่ปฎิบัติงาน - ลงเวลาเข้างาน - ปักหมุดแผนที่ที่ปฎิบัติงานอยู่ |
| R11 | บันทึกลงเวลาทำงานของ CG | เมื่อกรอกข้อมูล วันที่/เดือน/ปีที่ปฎิบัติงาน เวลาเข้างาน และ ปักหมุด แผนที่ที่ปฎิบัติงานกรอกข้อมูลไม่ครบ แสดง " กรุณากรอกข้อมูลให้ ครบทุกช่อง " |
| R12 | บันทึกลงเวลาทำงานของ CG | สามารถบันทึกข้อมูลการปฎิบัติงานได้ - ลงเวลาออกงาน - รูปถ่ายการปฎิบัติงาน - กิจกรรมที่ปฎิบัติงาน |
| R13 | บันทึกลงเวลาทำงานของ CG | เมื่อกรอกข้อมูล ลงเวลาออกงาน รูปถ่ายการปฎิบัติงาน และ กิจกรรมที่ ปฎิบัติงานกรอกข้อมูลไม่ครบ แสดง " กรุณากรอกข้อมูลให้ครบทุกช่อง " |
| R14 | แสดงการบันทึกลงเวลาของ CG | แสดงข้อมูลเข้าปฎิบัติงานและออกจากการปฎิบัติงาน CG สามารถดูได้ |
| R15 | แสดงการบันทึกลงเวลาของ CG | แสดงข้อมูลเข้าปฎิบัติงานและออกจากการปฎิบัติงานของ CG ที่ CM สามารถตรวจเช็คดูได้ |
| R16 | แสดงสถิติการบันทึกลงเวลาของ CG | แสดงรายงานสถิติการบันทึกลงเวลาเป็นรายสัปดาห์ รายเดือน |
| R17 | แสดงสถิติการบันทึกลงเวลาของ CG | แสดงรายงานสถิติการบันทึกลงเวลาเป็นรายสัปดาห์ รายเดือน ของ CG ให้ CM สามารถดูได้ |
| R18 | แสดงสื่อและบทความให้ความรู้ | แสดงสื่อที่ให้ความรู้ ทั้งคลิปและบทความ |
| R19 | เบิกอุปกรณ์ | กรอกข้อมูลเบิกอุปกรณ์ - ชื่ออุปกรณ์ที่เบิก - รูปถ่ายอุปกรณ์ที่เบิก |
| R20 | แสดงการเบิกรายการอุปกรณ์ | แสดงรายการและรายละเอียดอุปกรณ์ที่ได้เบิกไป |
| R21 | แสดงการเบิกรายการอุปกรณ์ | แสดงรายการและรายละเอียดอุปกรณ์ที่ได้เบิกไปของ CG ของ CG ให้ CM สามารถดูได้ |
| R22 | แจ้งปัญหาในการปฏิบัติติงาน | กรอกข้อมูลแจ้งปัญหาในการปฏิบัติติงาน |
| R23 | แสดงการแจ้งปัญหาในการปฏิบัติติงาน | แสดงการแจ้งปัญหาในการปฏิบัติติงาน - แจ้งเมื่อเข้าปฏิบัติติจริงไม่ตรงกับ Care plan ที่วางไว้ |
| R24 | แสดงการแจ้งปัญหาในการปฏิบัติติงาน | แสดงการแจ้งปัญหาในการปฏิบัติติงานของ CG ที่ CM สามารถดูได้ |
| R25 | บันทึก Care plan ที่ CM วางไว้ | กรอกข้อมูล Care plan - วัน/เดือน/ปี ปฏิบัติติงานของ CG แต่ละคน - เวลาปฎิบัติงานของ CG แต่ละคน |
| R26 | บันทึก Care plan ที่ CM วางไว้ | เมื่อกรอกข้อมูล วัน/เดือน/ปี และเวลาปฎิบัติงานใน plan แล้วกดบันทึก Care plan แล้วเวลาเกิดซ้ำกัน แสดงแจ้งเตือน " เวลาปฎิบัติงานซ้ำกัน โปรดลองอีกครั้ง "แสดงรายละเอียด Care plan |
| R27 | แสดงการบันทึก Care plan ที่ CM วางไว้ | แสดงรายละเอียด Care plan ทั้งวัน/เดือน/ปี และเวลาในปฎิบัติงานของ CG แต่ละคน |
| R28 | แจ้งเตือนเมื่อถึงกำหนดการของ CG | แจ้งเตือนเมื่อเวลาเข้างาน แจ้งเตือนเมื่อถึงเวลาประชุม แจ้งเตือนเมื่อถึงเวลาทำกิจกรรม |
| R29 | แจ้งเตือนเมื่อถึงกำหนดการของ CM | แจ้งเตือนเมื่อถึงเวลาประชุม แจ้งเตือนเมื่อ CG แจ้งปัญหาเข้ามา |

**3.3 Use Case Diagram**

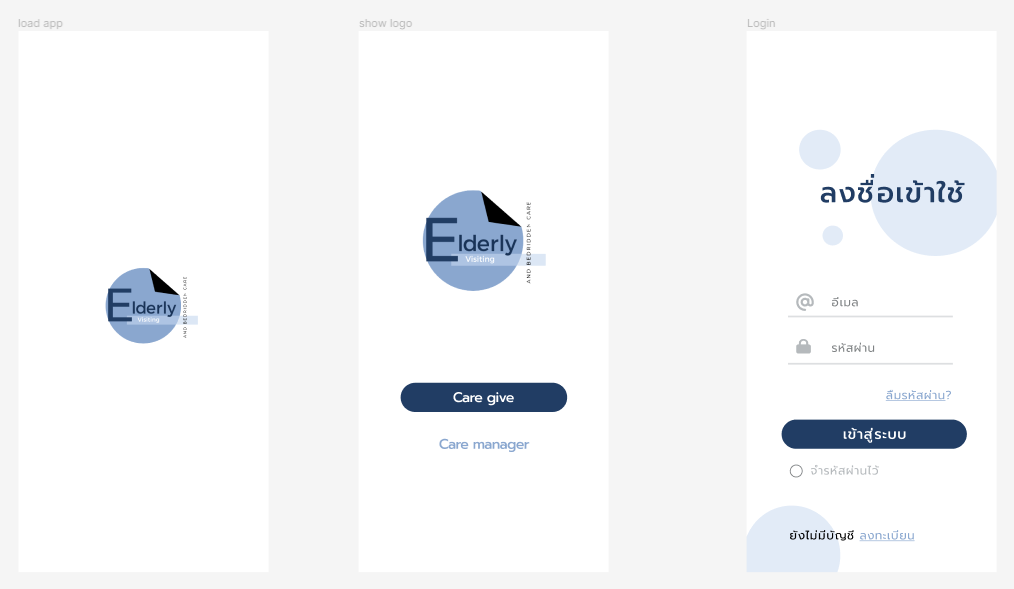


**รูปที่ 3.2 Use Case Diagram**

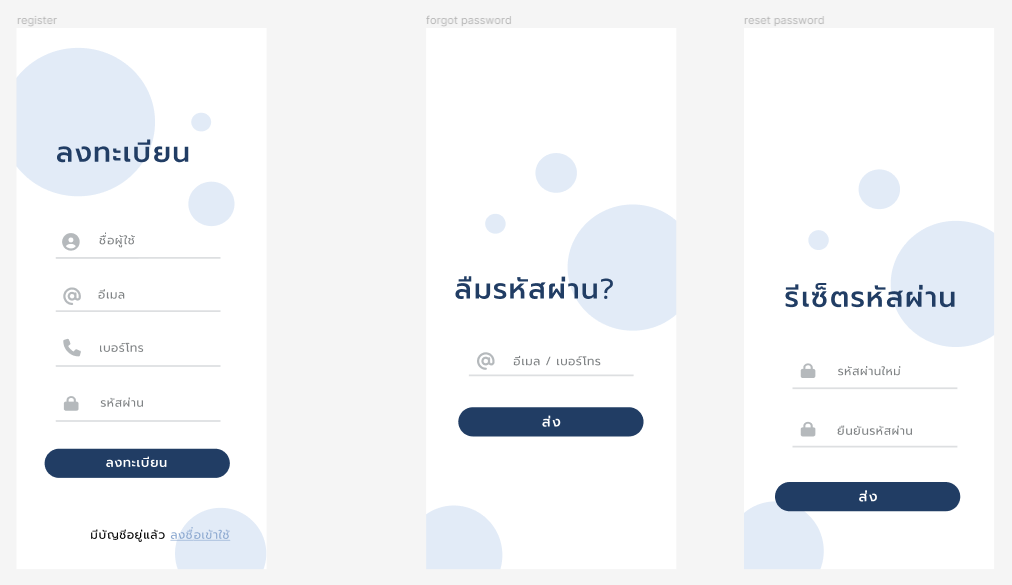
**3.4 ภาพรวมและตัวอย่าง UX UI Design**



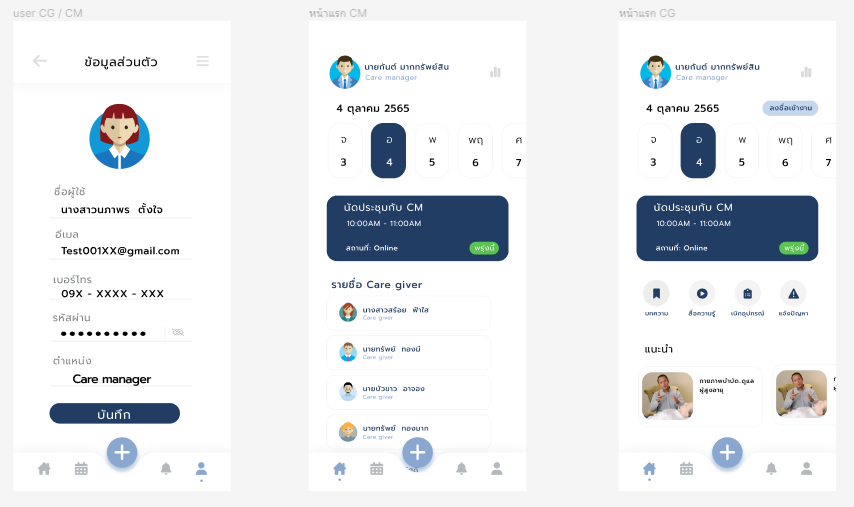
**รูปที่ 3.4.1 ภาพรวม UXUI Design**



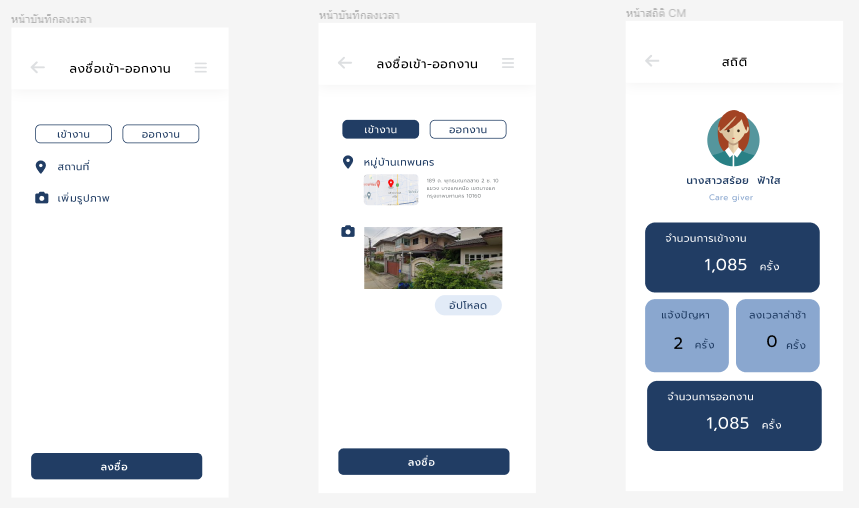
**รูปที่ 3.4.2 ตัวอย่าง UXUI Design**



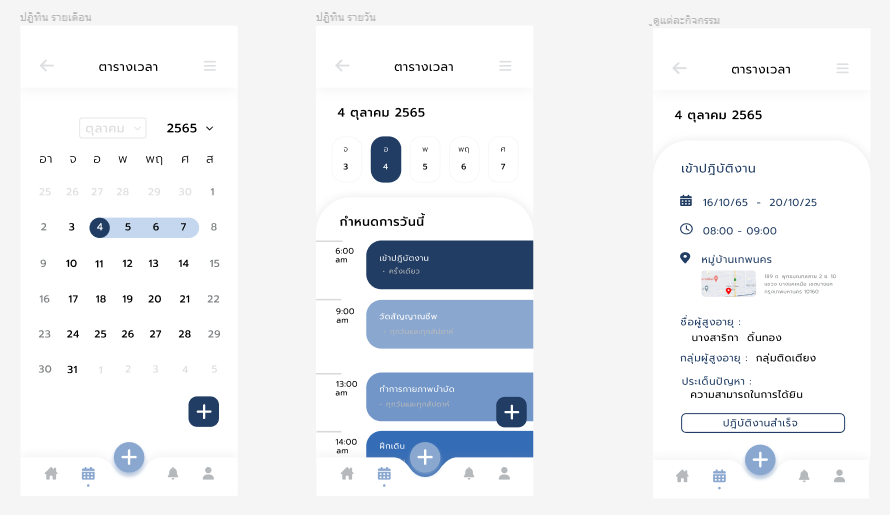
**รูปที่ 3.4.3 ตัวอย่าง UXUI Design**



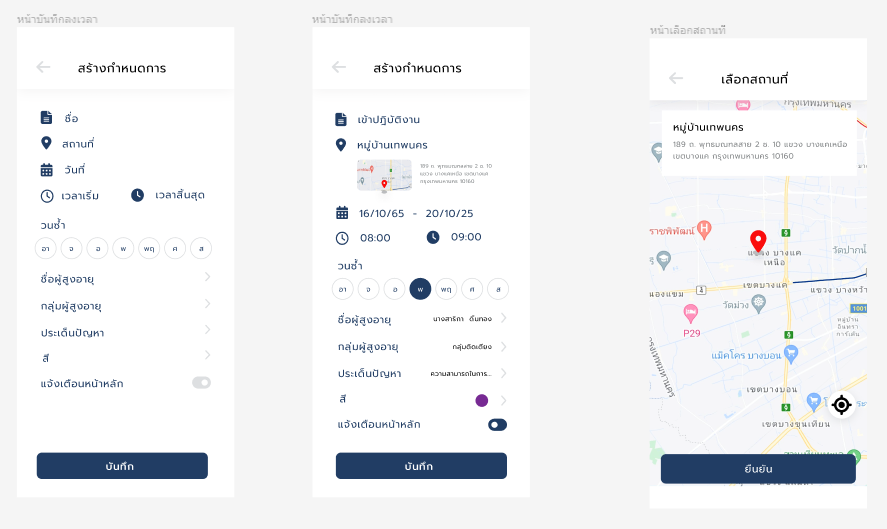
**รูปที่ 3.4.4 ตัวอย่าง UXUI Design**



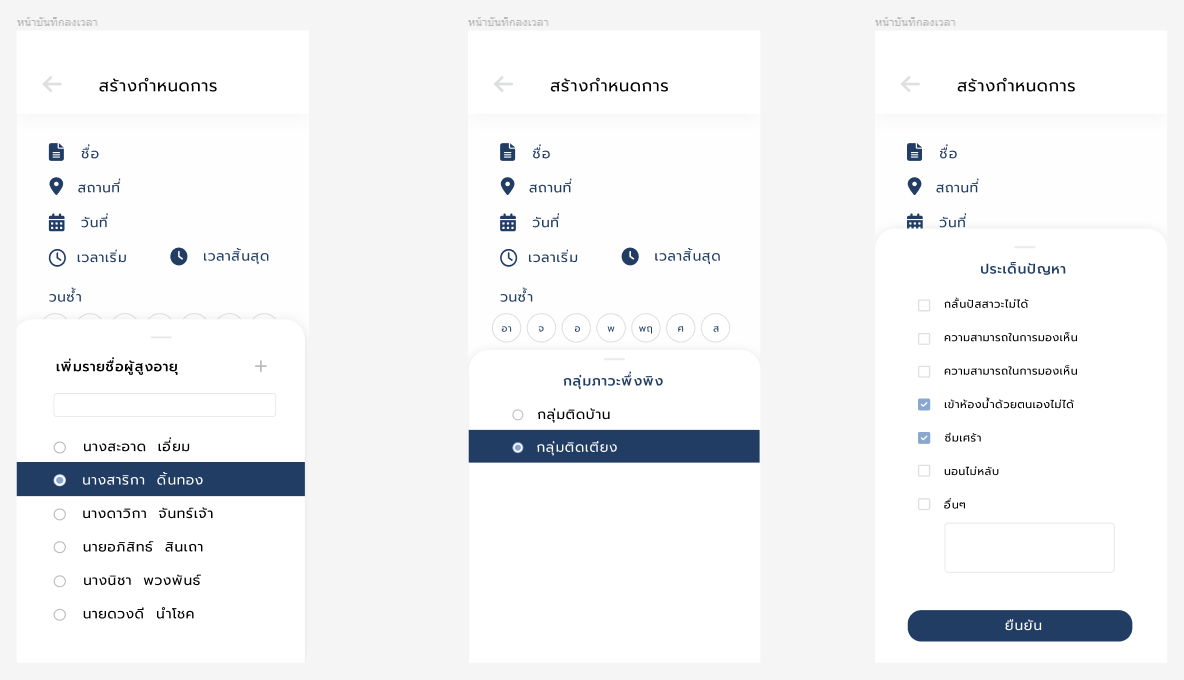
**รูปที่ 3.4.5 ตัวอย่าง UXUI Design**



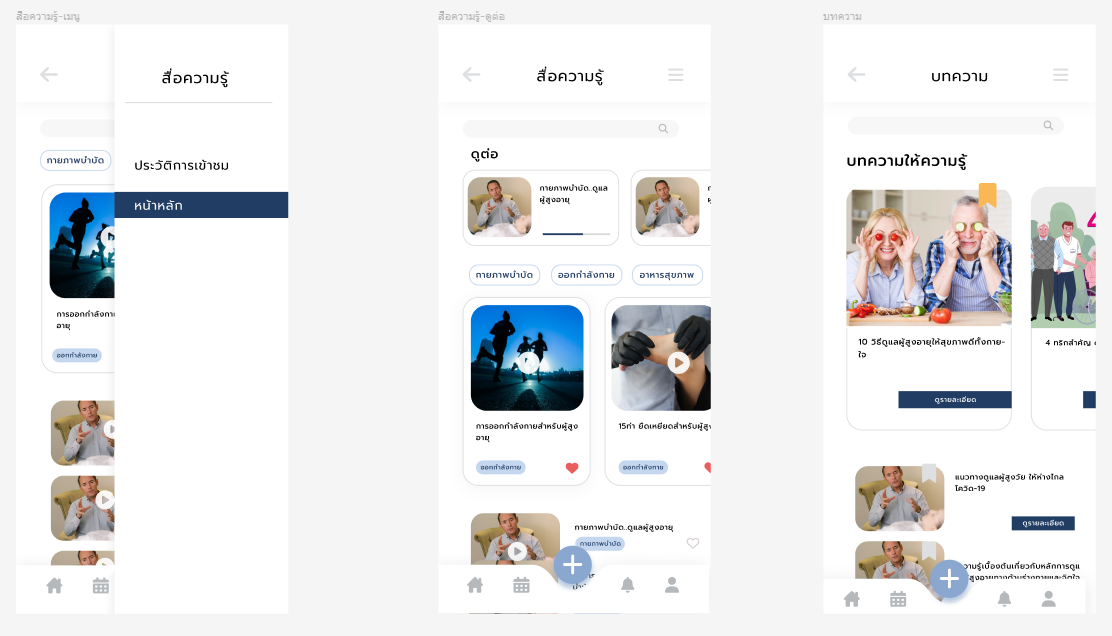
**รูปที่ 3.4.6 ตัวอย่าง UXUI Design**



**รูปที่ 3.4.7 ตัวอย่าง UXUI Design**

****

**รูปที่ 3.4.8 ตัวอย่าง UXUI Design**

****

**รูปที่ 3.4.9 ตัวอย่าง UXUI Design**

**บทที่ 4**

**ผลการดำเนินงาน**

**4.1 สร้าง Front end โดยใช้ React Native Expo**

ติดตั้ง npm i -g expo-cli และ npx create-expo-app my-app

สร้างโปรเจค app โดย run แล้วแสดงผลผ่านอุปกรณ์จริงด้วยแอป Expo Go หรือ Emulator

เครื่องมือที่เป็นตัวช่วยในการทำ Front end ด้วย Native Base เป็น UI library

รูปภาพประกอบด้วย ข้อความ, จอภาพ, คอมพิวเตอร์, สีดำ

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.1 หน้า Home ของ CM**

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.2 หน้า Home ของ CG**

ในการสร้างหน้า Home ของ CM กับ หน้า Home ของ CG มีส่วนประกอบ Layout เหมือนกันจึงมีการทำแยกไฟล์แบ่งเป็น screens กับ components โดยทำส่วนประกอบไว้ components แล้ว import ไปไว้ใน screens

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, สีดำ

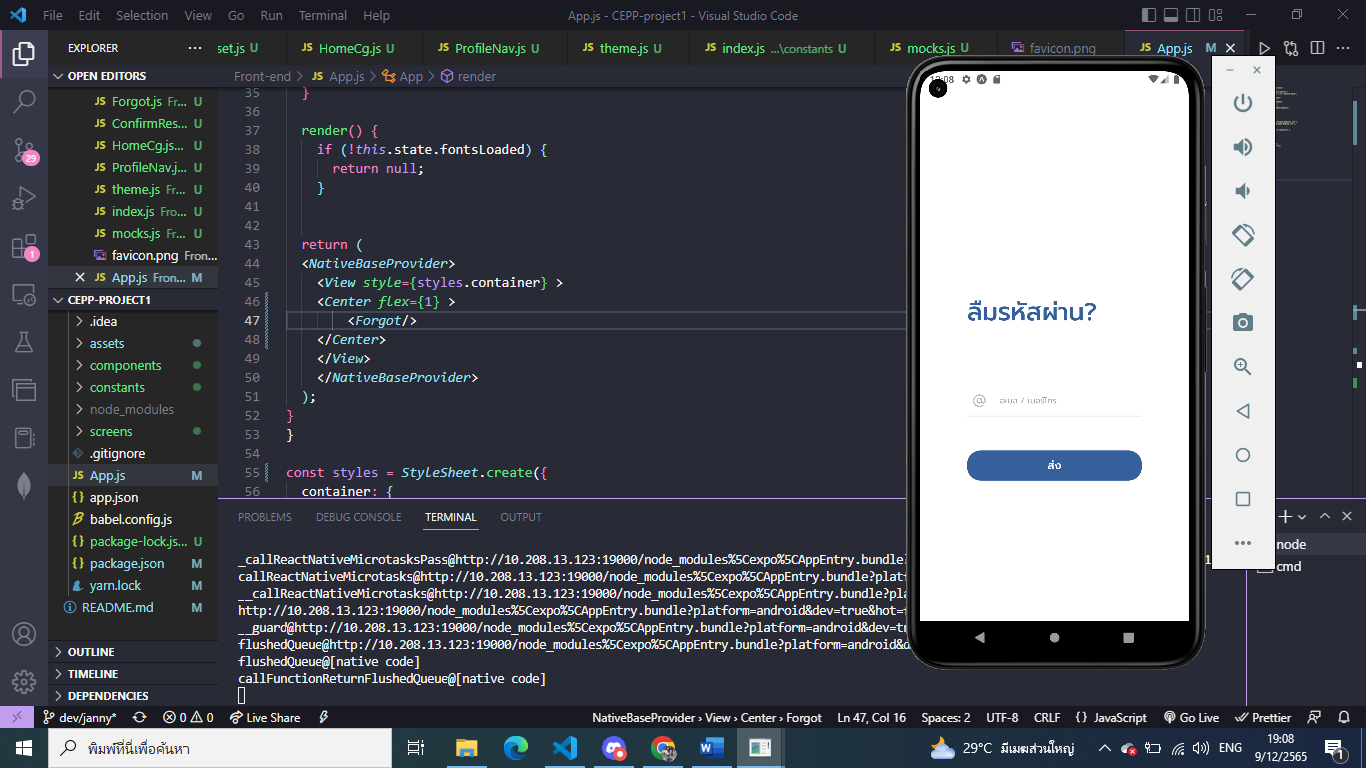
คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.3 หน้าลงชื่อเข้าใช้**

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.4 หน้าลงทะเบียน**



**รูปที่ 4.1.5 หน้าลืมรหัสผ่าน**

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.6 รีเซ็ตรหัสผ่าน**

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.7 ยืนยันรหัสผ่าน**

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.8 ตัวอย่างโค้ด Front end**

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.1.9 ไฟล์เก็บข้อมูลที่มีจำนวนมาก**

ในไฟล์ constants ใช้ในการดึงข้อมูลเยอะๆมาไว้แสดงผล เพื่อลดไม่ให้โค้ดแต่ละหน้าเยอะเกินไป

**4.2 สร้าง Back end โดยใช้ MongoDB ร่วมกับ GraphQL Apollo**

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, สีดำ

คำอธิบายที่สร้างโดยอัตโนมัติ

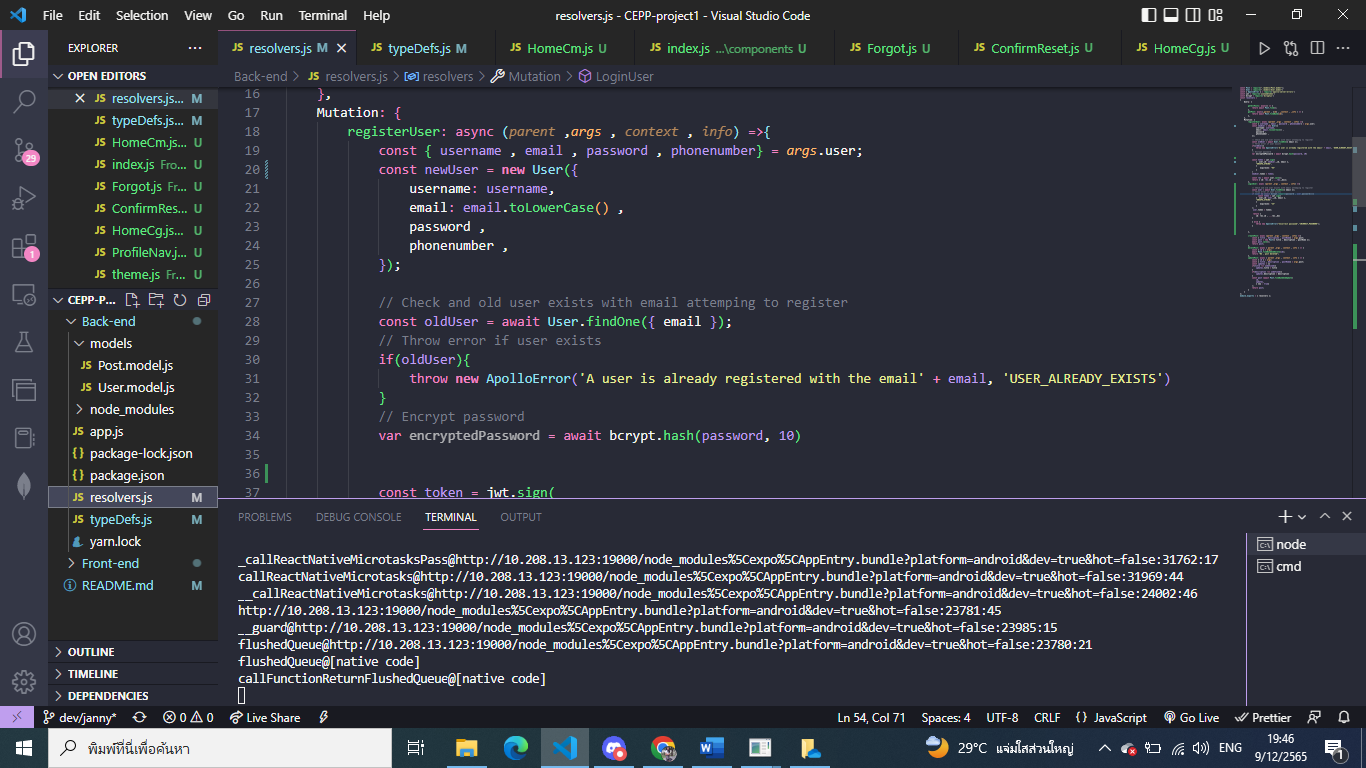
**รูปที่ 4.2.1 จัดเก็บข้อมูลของ User**

ในไฟล์ models จัดเก็บข้อมูล User ได้รวมการพึ่งพาเพิ่มเติมสำหรับโครงการของเราที่เรียกว่า mongoose เป็นเฟรมเวิร์กในการโต้ตอบกับ mongoDB

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, สีดำ

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.2.2 กำหนด Mongoose schema และ model**



**รูปที่ 4.2.3 สร้าง** **Apollo Server**

สร้าง Apollo Server สามารถเพิ่ม ลบ อัพเดท ในการใช้งานร่วมกับ Apollo และส่งข้อมูลใน typedefs และresolver เป็นพารามิเตอร์

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์

คำอธิบายที่สร้างโดยอัตโนมัติ

**รูปที่ 4.2.4 เรียกใช้** **Apollo Server**

เรียกใช้งานโดยมีการใช้ข้อมูลในไฟล์ typedefs และ resolver สามารถ input ข้อมูลเข้า Server ได้และสามารถเรียกดูข้อมูลที่ input ได้ด้วย

**บทที่ 5**

**สรุป**

**5.1 สิ่งที่ได้ทำไป**

5.1.1 Fornt end

- Integrate หน้าต่างๆ

5.1.2 ดึงระบบ Care plan จากส่วนกลาง

- ศึกษา meteor เพื่อที่จะไปดึงข้อมูล Care Plan ที่ได้รับมา

- ดูข้อมูล Care Plan ที่ได้รับมา เพื่อไปแก้ไข UXUI ใหม่บางส่วน

5.1.3 Back end

- แก้ไข Bsck end ที่ใช้ใหม่ ยังไม่เสร็จสมบูรณ์

**5.2 ปัญหาและแนวทางการแก้ไข**

5.2.1 Back end ที่ใช้มีปัญหา

ต้องแก้ไข Back end ที่จะใช้งาน เนื่องจากศึกษาและค้นคว้าตอนแรกไม่ดีพอ ต้องแก้ไขในส่วนของ Back end

5.2.2 การพัฒนาระบบทั้งหมดด้วยเครื่องคอมพิวเตอร์ส่วนตัว

เนื่องจากระบบทั้งหมดประกอบด้วย Service ต่างๆมากมายทำให้การทดสอบจำเป็นต้องเปิด Service พร้อมกันทำให้เครื่องคอมพิวเตอร์รับไม่ไหวและทำให้การทำงานล่าช้า

5.2.3 แก้ไขในส่วนของ UXUI บางส่วน

เนื่องจากที่ออกแบบมา ออกแบบก่อนที่จะดึงข้อมูล Care plan จึงไม่ทราบถึงการกรอกข้อมูล หรือใส่ข้อมูลต่างๆ ว่างพื้นที่ที่ใช้เพียงพอหรือไม่ จะแก้ไขโดยการดึงข้อมูล Care plan ให้เสร็จสิ้นละจะมาแก้ไข UXUI บางส่วนใหม่

**5.3 แนวทางการพัฒนาต่อ**

ตารางแนวทางการพัฒนาต่อ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| หัวข้อ | ธ.ค. | ม.ค. | ก.พ. | มี.ค. |
| ดูระบบ Care Plan |  |  |  |  |
| พัฒนาระบบ front end |  |  |  |  |
| พัฒนาระบบ Back end |  |  |  |  |
| ทดสอบและปรับปรุงแก้ไข |  |  |  |  |
| จัดเล่มรายงานและนำเสนอ |  |  |  |  |

**5.4 เสนอแนะการพัฒนาต่อ**

1) ปรับปรุงส่วนต่างๆให้รองรับกับระบบ Care Plan ที่ได้มามากขึ้น

2) แก้ไขให้สามารถใช้งานกับระบบได้หลากหลายมากขึ้น

**บรรณานุกรม**

# ภาคผนวก ก

# การพิจารณาปริมาณงานขั้นต่ำสำหรับโครงงาน 1

ก.1 ความก้าวหน้าของโครงงาน

โครงงาน ระบบตรวจเยี่ยมผู้สูงอายุและดูแลผู้ป่วยติดเตียงมีปริมาณงานทั้งหมด .. Tasks คิดเป็น …. ชั่วโมง เป็นการนับชั่วโมงแบบหาก 2 คนทำ Task เดียวกันจะคิดเวลา 2 คนรวมกัน และมี Task ที่ทำขนานกันทำให้ชั่วโมงเยอะเกินจริง ทำเสร็จไปแล้ว 38 % ของปริมาณงานทั้งหมด