ระบบเพิ่มความปลอดภัยในการพิสูจน์ทราบตัวตนแบบหลายปัจจัยบนระบบปฏิบัติการแอนดรอยด์

MULTI – FACTOR AUTHENTICATION VIA ANDROID APPLICATION

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบเพิ่มความปลอดภัยในการพิสูจน์ทราบตัวตนแบบหลายปัจจัยบนระบบปฏิบัติการแอนดรอยด์

MULTI – FACTOR AUTHENTICATION VIA ANDROID APPLICATION

กิตติธัช สีลาเวียง

เจตพล พุ่มวัฒนกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเพิ่มความปลอดภัยในการพิสูจน์ทราบตัวตนแบบหลายปัจจัย

บนระบบปฏิบัติการแอนดรอยด์

MULTI – FACTOR AUTHENTICATION VIA ANDROID APPLICATION

ผู้จัดทำ

1. นายกิตติธัช สีลาเวียง รหัสนักศึกษา 57010085

2. นายเจตพล พุ่มวัฒนกุล รหัสนักศึกษา 57010216

อาจารย์ที่ปรึกษา

(ดร.ธนัญชัย ตรีภาค)

ระบบเพิ่มความปลอดภัยในการพิสูจน์ทราบตัวตนแบบหลายปัจจัย บนระบบปฏิบัติการแอนดรอยด์

นายกิตติธัช	สีลาเวียง	57010085
นายเจตพล	พุ่มวัฒนกุล	57010216
ดร.ธนัญชัย	ตรีภาค	อาจารย์ที่ปรึกษา

ปีการศึกษา 2560

บทคัดย่อ

เนื่องด้วยปัจจุบัน อินเทอร์เน็ตได้เข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก ไม่ว่าจะเป็นการติดต่อสื่อสาร การทำธุรกรรมทางอิเล็กทรอนิกส์ หรือบัญชีผู้ใช้งานในเว็บไซต์ต่าง ๆ ย่อมต้องการความปลอดภัยในการใช้งาน โดยเฉพาะอย่างยิ่ง เมื่อประมาณ 5–6 ปีที่ผ่านมา สมาร์ตโฟนต่าง ๆ ได้มีการพัฒนาขีดความสามารถจนสามารถทำงานได้เกือบเทียบเท่าคอมพิวเตอร์ หรืออาจเรียกว่าคอมพิวเตอร์ขนาดย่อมเลยก็ได้ เมื่ออินเทอร์เน็ตและสมาร์ตโฟนสามารถเข้าถึงทุกคนได้อย่างง่ายดาย จึงมีกลุ่มบุคคลที่มองเห็นช่องทางในการให้บริการที่สะดวกสบายโดยการใช้งานบริการผ่านสมาร์ตโฟน เช่น การซื้อของออนไลน์, การทำธุรกรรมทางการเงินออนไลน์, การเข้าถึงบัญชีสังคมออนไลน์ เป็นต้น ทว่าเราจะทราบได้อย่างไรว่าบุคคลที่กำลังใช้งานอยู่นั้นเป็นเจ้าของบัญชีจริง ๆ

ผู้ที่เข้าถึงอินเตอร์เน็ตได้นั้น ย่อมมีความเสี่ยงต่าง ๆ ในการใช้งานโดยปริยาย เนื่องจากมีผู้ไม่ประสงค์ดีใช้วิธีการต่าง ๆ ในการล้วงข้อมูลอันเป็นความลับ ทำให้สามารถเข้าถึงระบบได้ จึงได้มีการคิดค้นวิธีการพิสูจน์ทราบตัวตนขั้นที่สองขึ้น เพื่อใช้เป็นสิ่งที่พิสูจน์อีกขั้นหนึ่งว่าเป็นเจ้าของบัญชีจริง ๆ

โครงงานนี้ ได้ทำการศึกษาข้อมูลเกี่ยวกับระบบพิสูจน์ทราบตัวตนขั้นที่สอง และสร้างโปรแกรมประยุกต์ในการพิสูจน์ทราบตัวตนขั้นที่สองขึ้น เพื่อเป็นแนวทางในการศึกษาและป้องกันในภายภาคหน้าต่อไป

MULTI – FACTOR AUTHENTICATION VIA ANDROID APPLICATION

Mr. Kittitad	Seelawiang	57010085
Mr. Jettapol	Pumwattanakul	57010216
Dr. Thanunchai	Threepak	Advisor

Academic Year 2017

ABSTRACT

Nowadays, the internet has become to be an incredibly important role in human's daily life. It is mainly used in communication, E-commerce and many things we can imagine, so it requires a big concern about the security. Moreover, about 5 or 6 years ago, many smartphones had been upgraded to be powerful almost as equal as the personal computer, obviously, it was even named "the mini PC": the replacement of the personal computer. When the internet and smartphones are accessible to everyone easily, some brilliant people found the way to give convenient services through these smartphones, such as an online-shopping, an internet-banking or an access to social-media accounts. However, how do we know that the person who is using these services is the genuine owner of the account?

Those who have access to the internet have taken various risks in using it implicitly. The internet is not thoroughly secured by nature. Malicious users have many methods to take confidential credentials away from us, to do any bad things by any means. To counter this, the use of the multiple-factor authentication has begun. Its function is mainly to prove that someone is really oneself, by give oneself another challenge, generally another password.

This project focuses on the information about this multiple–factor authentication system, and to make the application via smartphone (especially on Android OS) to verify in another step. This may be the guidelines to study, to prevention any risks and to improve any security plans in the future.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้เสร็จสมบูรณ์ได้ตามเป้าหมาย เพราะได้รับความช่วยเหลือและคำแนะนำที่มีประโยชน์อย่างยิ่งต่อการศึกษาค้นคว้าข้อมูลอย่างยิ่ง จากผู้มีพระคุณหลายท่าน อาทิ

ขอขอบคุณดร.ธนัญชัย ตรีภาค ที่ให้คำปรึกษาชี้แจงและแนะนำข้อมูลที่เป็นประโยชน์ต่อการพัฒนาปริญญานิพนธ์ให้สมบูรณ์ทั้งในเวลาและนอกเวลาราชการตั้งแต่การเริ่มเตรียมปริญญานิพนธ์จนถึงการทำปริญญานิพนธ์ให้บรรลุผลที่ตั้งไว้

ขอขอบคุณอาจารย์ทุกท่านในคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่สั่งสอน แนะนำในการเรียนมาตลอดเป็นระยะเวลา 4 ปี

ขอขอบคุณเพื่อน ๆ พี่ ๆ และน้อง ๆ สาขาวิชาวิศวกรรมคอมพิวเตอร์ทุกคนที่ช่วยเหลือและให้กำลังใจมาโดยตลอดไม่ว่าจะมากหรือเล็กน้อยเพียงใด

ขอขอบคุณห้องพัฒนาและวิจัยความปลอดภัยของข้อมูล (Information Security Advisory Group: ISAG) ที่เอื้อเฟื้อปัจจัยต่าง ๆ ในการทำปริญญานิพนธ์เล่มนี้ให้สมบูรณ์

สุดท้ายนี้ขอกราบขอบพระคุณบิดามารดาที่สนับสนุนในด้านการเรียน, คำแนะนำ, กำลังใจและปัจจัยต่าง ๆ ต่อคณะผู้จัดทำมาโดยตลอด

กิตติธัช สีลาเวียง

เจตพล พุ่มวัฒนกุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูป **หน้า**

สารบัญรูป (ต่อ)

รูป **หน้า**

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ปัจจุบันเทคโนโลยีต่าง ๆ สามารถเข้าถึงได้ง่ายกว่าสมัยก่อน อุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ เช่น โทรศัพท์มือถือ แท็บแล็ต คอมพิวเตอร์ต่างก็มีราคาถูกลง เมื่อเทคโนโลยีสามารถเข้าถึงได้ง่าย ก็ย่อมมีความเสี่ยงตามมาเช่นกัน จากที่ได้ยินข่าวในอดีตไม่ว่าจะเป็นการสวมรอยตัวตน (Identity Theft) การขโมยข้อมูล (Data Hijacking) ฯลฯ ทำให้จำเป็นต้องมีการพิสูจน์ทราบตัวตนเกิดขึ้น ซึ่งเป็นการทำให้ผู้ให้บริการมีความเชื่อใจในระดับหนึ่งว่าผู้ที่มาใช้บริการนั้นมีตัวตนจริงในสังคม ส่งผลให้เมื่อเกิดปัญหาแล้วสามารถยกเลิกสิทธิ์ผู้ใช้งาน หรือตามรอยผู้ใช้งานได้ แต่ในปัจจุบันนั้นผู้ที่ไม่ประสงค์ดีได้มีวิธีการใหม่ ๆ ไม่ว่าจะเป็นการสวมรอย หรือขโมยตัวตนของบุคคลนั้น ๆ ที่ทำให้สามารถเข้าถึงระบบได้อย่างไม่ยากเย็น และทำให้การค้นหาผู้ที่กระทำผิดจริง ๆ นั้นยากลำบากมากขึ้น จึงได้มีวิธีการพิสูจน์ทราบตัวตนขั้นที่สองเกิดขึ้น ซึ่งผู้จัดทำได้สนใจในระบบพิสูจน์ทราบตัวตนขั้นที่สองนี้ จึงได้ทำการศึกษา และพัฒนาระบบพิสูจน์ทราบตัวตนขั้นที่สองเพื่อให้การเข้าสู่ระบบมีความปลอดภัยมากขึ้นไปอีก

1.2 เป้าหมายของโครงงาน

สร้างโปรแกรมประยุกต์ในระบบปฏิบัติการ Android ที่มีหน้าที่หลักในการสร้างรหัสผ่านอีกชุดหนึ่งเพื่อใช้พิสูจน์ทราบตัวตนโดยเน้นไปที่ความสะดวกสบายของผู้ใช้เป็นสำคัญ

1.3 วัตถุประสงค์ของโครงงาน

1) เรียนรู้กระบวนการทำงานของระบบการพิสูจน์ทราบตัวตนในรูปแบบต่าง ๆ
2) เรียนรู้เทคนิคการพิสูจน์ทราบตัวตนหลากหลายรูปแบบ
3) ออกแบบโปรแกรมประยุกต์การพิสูจน์ทราบตัวตนขั้นที่สอง
4) ออกแบบระบบพิสูจน์ทราบตัวตนขั้นที่สอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

เทคนิคการยืนยันตัวตนหลากหลายรูปแบบ เช่น การใช้รหัสผ่าน การให้สิทธิ์ในการใช้งานชั่วคราว การใช้รหัสอีกชุดหนึ่งในการยืนยันตัวตน

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การพิสูจน์ทราบตัวตน

การพิสูจน์ทราบตัวตน (Authentication) เป็นสิ่งที่ใช้สำหรับการสร้างความเชื่อมั่นหรือยืนยันว่าสิ่งนั้นหรือผู้นั้น คือตัวจริงหรือตัวแทนจริง ในโลกของการรักษาความปลอดภัยนั้น การพิสูจน์ทราบตัวตนจะหมายถึง การตรวจสอบตัวตนในรูปแบบดิจิทัลของผู้ที่ทำการสื่อสาร เช่น การเข้าสู่ระบบ ผู้ที่สื่อสารอาจเป็นผู้ที่ใช้งานคอมพิวเตอร์ ระบบคอมพิวเตอร์ หรืออาจเป็นโปรแกรมประยุกต์ก็ได้

การพิสูจน์ทราบตัวตนเป็นด่านแรกสำหรับการควบคุมและรักษาความปลอดภัยของระบบคอมพิวเตอร์ ซึ่งก็จะนำไปสู่การตรวจสอบสิทธิ์ในการสร้างและแก้ไขข้อมูลในระบบ รวมไปถึงการจัดเก็บพฤติกรรมการใช้งานระบบ เพื่อสำหรับการตรวจสอบในภายหลัง ก่อนที่จะทำการพิสูจน์ทราบตัวตนได้นั้น ต้องสามารถระบุตัวตน (Identification) ให้ได้ก่อน เป็นขั้นตอนที่ผู้ใช้งานต้องระบุชื่อเฉพาะว่าตนเองเป็นใคร เช่น ชื่อผู้ใช้, หมายเลขประจำตัว, อีเมล, คีย์การ์ด หรือลายนิ้วมือ เป็นต้น สิ่งเหล่านี้จะเป็นตัวแทนของผู้ใช้ระบบคอมพิวเตอร์ เมื่อมีตัวแทนแล้วขั้นตอนต่อไป คือการพิสูจน์ทราบตัวตนโดยผู้ใช้ต้องใช้หลักฐานที่สอดคล้องกับที่ระบบมีอยู่ ยกตัวอย่างเช่น การพิสูจน์ทราบตัวตนโดยใช้ชื่อผู้ใช้ และรหัสผ่าน ตัวแทนของเราคือชื่อผู้ใช้ ส่วนสิ่งที่เป็นหลักฐานในการพิสูจน์ก็คือรหัสผ่านนั่นเอง ซึ่งข้อมูลที่ผู้ใช้งานทราบและสิ่งที่ระบบมีต้องตรงกัน ไม่เช่นนั้นจะถือว่าผู้ใช้งานนั้นไม่ใช่ตัวตนที่ถูกต้อง

2.1.1 ปัจจัยในการพิสูจน์ทราบตัวตน (Authentication Factors)

ปัจจัยในการพิสูจน์ทราบตัวตน มี 3 ปัจจัย

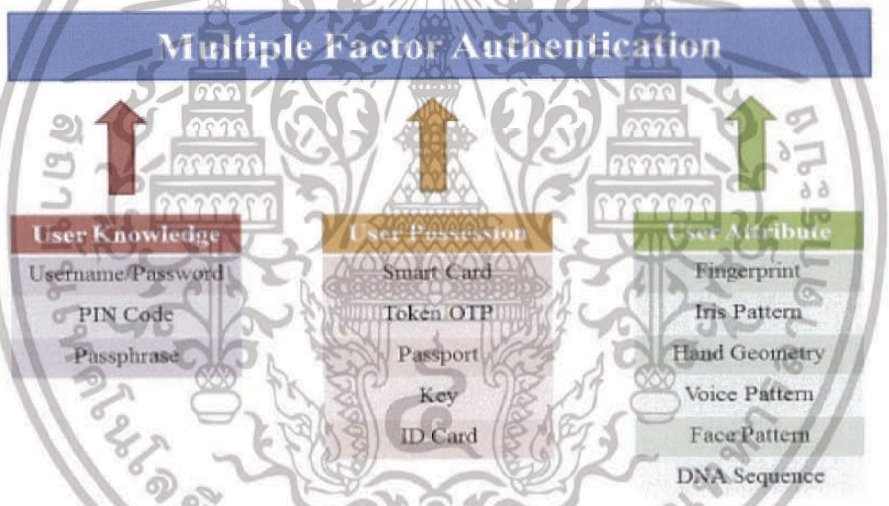
1) สิ่งที่ทราบ (User Knowledge) หมายถึง ข้อมูลหรือรหัสที่ผู้ใช้งานทราบและจดจำได้ เช่น รหัสผ่าน, PIN
2) สิ่งที่มี (User Procession) หมายถึง ข้อมูลหรือรหัสผ่านที่อ่านได้จากสิ่งของหรือวัตถุที่ผู้ใช้งานมี และพกติดตัว เช่น คีย์การ์ด, Token, บัตรประจำตัว
3) สิ่งที่เป็น (User Attribute) หมายถึง ข้อมูลหรือรหัสผ่านที่อ่านได้จากอวัยวะบางส่วนของร่างกาย (ชีวมาตร; Biometrics) เช่น ลายนิ้วมือ, ม่านตา, หน้า, เสียง

ปัจจัยที่นำมาใช้เพียงลักษณะอย่างใดอย่างหนึ่งจะเรียกว่า การพิสูจน์ทราบตัวตนปัจจัยเดียว (Single-factor Authentication) ซึ่งมีข้อจำกัดในการใช้ เช่น อาจจะถูกดักฟัง, เดา หรือขโมยจาก

เครื่องคอมพิวเตอร์ แต่ถ้ามีการนำหลาย ๆ ปัจจัยมาใช้ร่วมกันจะเรียกว่า การพิสูจน์ทราบตัวตนหลายปัจจัย (Multi-factor Authentication)

2.1.2 ปัจจัยในการพิสูจน์ทราบตัวตน (Multiple-Factor Authentication: MFA)

MFA คือ กระบวนการยืนยันตัวตน โดยใช้ปัจจัยมากกว่าหนึ่งปัจจัยร่วมกัน ดังภาพประกอบที่ 2.1 เช่น ใช้ User Knowledge ร่วมกับ User Possession เป็นต้น ดังนั้น MFA จึงทำให้การยืนยันตัวตนมีความมั่นคงมากขึ้น เนื่องจากจำนวนปัจจัยที่เพิ่มมากขึ้นส่งผลให้กระบวนการยืนยันตัวตนมีความซับซ้อนและมีลำดับขั้นตอนเพิ่มมากขึ้น จากงานวิจัยและผลิตภัณฑ์ เช่น Smart Card Authentication , RSA SecurID ซึ่งได้ใช้ MFA เพื่อเพิ่มความมั่นคงในการยืนยันตัวตน แต่ยังพบจุดอ่อนที่เกิดขึ้นแตกต่างกันไป ในวิทยานิพนธ์นี้ได้นำเอาแนวคิดของ MFA เข้ามาใช้งาน โดยพยายามที่จะนำเอาปัจจัยแต่ละประเภทมาแก้ไขจุดอ่อนในกระบวนการยืนยันตัวตน



รูป 2.1 ปัจจัยในการพิสูจน์ตน

2.2 รหัสผ่าน

หลักฐานที่นิยมในการพิสูจน์ทราบตัวตนมากที่สุด คือ รหัสผ่าน (Password) เป็นสิ่งที่ใช้เป็นมาตรฐานทั่วไปในการควบคุมการเข้าถึงระบบปฏิบัติการและโปรแกรมประยุกต์ อย่างไรก็ตามการที่ใช้รหัสผ่านเพียงอย่างเดียวอาจไม่ปลอดภัยเพียงพอ ข้อเสียอย่างหนึ่งของการใช้รหัสผ่าน คือผู้ใช้อาจจะลืมรหัสผ่าน ดังนั้นผู้ผลิตโน้ตบุ๊กหรือสมาร์ตโฟนจึงใช้เทคนิคอื่นเข้ามาช่วย เช่น การสแกนลายนิ้วมือ

2.2.1 รหัสผ่านแบบคงที่

รหัสผ่านแบบคงที่ หรือรหัสผ่านทั่วไป เป็นวิธีการที่ใช้กันอย่างแพร่หลาย รหัสผ่านควรกำหนดให้เฉพาะผู้ที่มีสิทธิ์เท่านั้นที่ทราบ แต่ว่าในปัจจุบันการใช้รหัสผ่านไม่มีประสิทธิภาพเพียงพอในการป้องกันระบบ เพราะเมื่อมีรหัสผ่านที่แตกต่างกันมากมายในแต่ละระบบสำหรับแต่ละผู้ใช้งาน หรือรหัสผ่านมีความยากและยาว ก็อาจเกิดปัญหาในการจำ ทำให้ผู้ใช้บางคนเขียนหรือจดบันทึกรหัสผ่านนั้นไว้ ซึ่งอาจถูกขโมยและนำไปใช้เข้าสู่ระบบได้

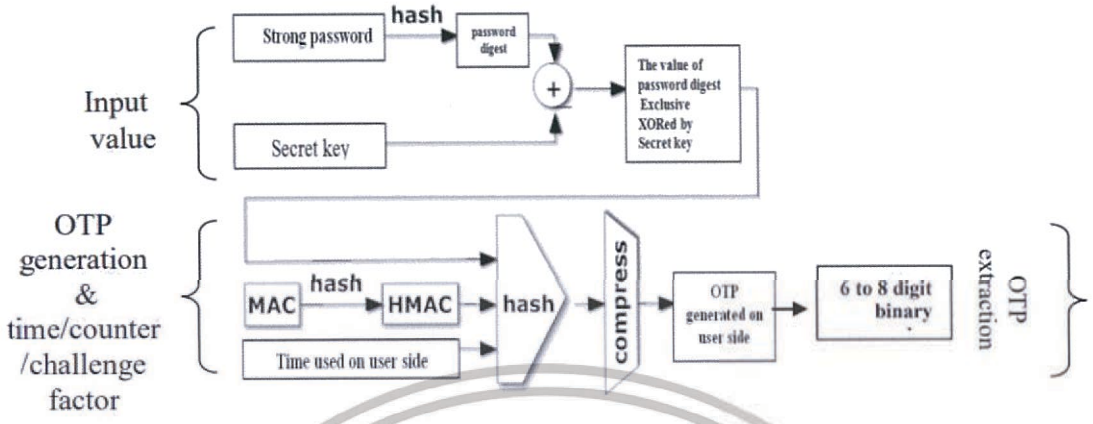
2.2.2 OTP

OTP (One-Time Password) เป็นระบบที่ใช้รหัสผ่านแบบครั้งเดียวในการเข้าสู่ระบบ ถ้าใช้ระบบนี้แล้ว จะทำให้การโจมตีแบบแอบดักฟังรหัสผ่าน (Sniffing) และการใช้รหัสผ่านซ้ำ (Replay Attack) นั้นไม่สามารถเข้าสู่ระบบได้ เพราะรหัสผ่านจะถูกเปลี่ยนทุกครั้งก่อนที่ผู้ใช้งานจะเข้าสู่ระบบ OTP นั้นมีหลักการมาจากการใช้อัลกอริทึมการเข้ารหัสลับ (Cryptographic Algorithms)

ในการสร้างรหัสแบบ OTP นั้น ต้องประกอบข้อมูลเหล่านี้ในการคำนวณด้วย

1) ค่าที่รับเข้ามา (Input Value)
2) กระบวนการสร้างรหัส (OTP generation)
3) กระบวนการแปลงรหัส (OTP extraction)
4) เวลา (Time)

ในรูป 2.2 จะแสดงถึงอัลกอริทึมในการสร้างรหัสแบบ OTP โดยเริ่มจากรับค่าที่ได้รับมา เช่น รหัสผ่านและกุญแจลับ จากนั้นนำรหัสผ่านที่ได้มาทำการ Hashing แบบ MD5 หรืออื่นๆ และใช้ค่าที่ใช้ร่วมกันระหว่างเครื่องแม่ข่ายกับตัวสร้าง OTP นำค่าเวลา, ค่านับ และค่าคำทายที่นำมาใช้สำหรับเป็นกุญแจ และข้อมูลในอัลกอริทึมการเข้ารหัสลับ สุดท้ายจะใช้อัลกอริทึมสุ่มเลือกหลัก ทำการแปลงค่าที่ได้จากการอัลกอริทึมการสร้าง เพื่อให้ได้รหัส OTP ที่แท้จริง



รูป 2.2 กระบวนการสร้างรหัส OTP

2.2.3 เทคนิคที่ประยุกต์ใช้ OTP

1) Lin OTP (Linux One Time Password) เป็น OTP ที่ใช้เพื่อเพิ่มความปลอดภัยของกระบวนการเข้าสู่ระบบทุกประเภทบนระบบปฏิบัติการ Linux
2) MOTP (Mobile One Time Password) เป็น OTPที่เกี่ยวข้องกับการ Synchronize ระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่ายโดยปกติจะมีระยะเวลา 3 นาที ซอฟท์แวร์หลายตัวที่สามารถดาวน์โหลดได้บนโทรศัพท์มือถือสนับสนุนเทคโนโลยีนี้
3) HOTP (HMAC One Time Password algorithm) เป็น OTP ที่ขึ้นอยู่กับค่าตัวนับ (Counter) ที่เพิ่มขึ้นทั้งทางเครื่องลูกข่ายและเครื่องแม่ข่าย โดยตัวนับนี้นับเป็นตัวแปรสำคัญตัวหนึ่งในการเพิ่มไม่ซ้ำซ้อนของรหัสผ่านใช้ครั้งเดียวนี้ สำหรับการใช้ค่าตัวนับนั้น หากทั้งสองรหัสผ่านจากทางผู้ใช้และทางเครื่องแม่ข่ายตรงกัน เครื่องแม่ข่ายจะตรวจสอบและให้สิทธิในระบบกับผู้ใช้และเปลี่ยนแปลงค่าตัวนับนี้โดยนับเพิ่ม (Increment)
4) CROTP (OATH Challenge-Response algorithm) เป็น OTP ที่ขึ้นอยู่กับการส่งคำทาย (Challenge) จาก Authentication Server ขณะที่เครื่องแม่ข่ายส่งคำทายแบบสุ่มประกอบด้วยอักขระ 4 ตัวที่กำหนดเป็น PIN ให้ผู้ใช้ป้อนค่า PIN จากนั้นจะส่งคำตอบ (Response) ต่อเครื่องแม่ข่าย
5) TOTP (Time-based One Time Password) ใช้เป็นข้อมูลปัจจัยเพิ่มเติมในระบบ Two -factor Authentication ที่ผู้ใช้ต้องป้อน OTP หลังจากเข้าสู่ระบบ

2.3 Web Service

Web service คือระบบซอฟต์แวร์ที่ออกแบบมา เพื่อสนับสนุนการแลกเปลี่ยนข้อมูลกัน ระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยที่ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์คือ XML Web service มีอินเทอร์เฟสที่ใช้อธิบายรูปแบบข้อมูลที่เครื่องคอมพิวเตอร์ประมวลผลได้ เช่น WSDL ระบบคอมพิวเตอร์ใช้งานสื่อสารโต้ตอบกับเว็บเซอร์วิสตามรูปแบบที่ได้กำหนดไว้แล้ว โดยการส่งสาส์นตามอินเตอร์เฟสของ Web service นั้น โดยที่สาส์นดังกล่าวอาจแนบไว้ในซอง SOAP หรือส่งตามอินเตอร์เฟสในแนวทางของ REST สาสน์เหล่านี้ปกติแล้วถูกส่งโดยอาศัย HTTP และใช้ XML ร่วมกับมาตรฐานเกี่ยวกับเว็บอื่น ๆ โปรแกรมประยุกต์ที่เขียนโดยภาษาต่าง ๆ และทำงานบนแพลตฟอร์มต่าง ๆ กันสามารถใช้ Web service เพื่อแลกเปลี่ยนข้อมูลผ่านทางเครือข่ายคอมพิวเตอร์ เช่น อินเทอร์เน็ต ในลักษณะเดียวกับการสื่อสารระหว่างโปรแกรม (Inter-process communication) บนเครื่องเดียวกัน ความสามารถในการแลกเปลี่ยนข้อมูลระหว่างระบบที่ต่างกันนี้ (เช่น การแลกเปลี่ยนข้อมูลระหว่าง โปรแกรมที่เขียนโดยภาษา JAVA และโปรแกรมที่เขียนโดยภาษา Python หรือการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ที่ทำงานบน Microsoft Windows และโปรแกรมประยุกต์ที่ทำงานบน Linux) เกิดขึ้นได้เนื่องจากการใช้มาตรฐานเปิด โดย OASIS และ W3C เป็น คณะกรรมการหลักในการรับผิดชอบมาตรฐานและสถาปัตยกรรมของเว็บเซอร์วิส

ในการพัฒนา Web service นั้น สามารถเลือกที่จะพัฒนาแบบ SOAP หรือแบบ REST ก็ได้ ถ้าเราพัฒนา SOAP Web services เราจะต้องมีการส่งข้อความ XML ตามรูปแบบที่กำหนดไว้โดยโปรโตคอล SOAP อีกทั้งต้องมีเอกสารอธิบายการเรียกใช้ Web service ประกอบ ซึ่งเอกสารที่อธิบายนี้ จะเขียนโดยใช้ภาษา WSDL ในแง่ของผู้เรียกใช้ จะต้องมีการเข้าใจเอกสารที่อธิบายการเรียกใช้ SOAP Web services หรือมีเครื่องมือที่จะเข้าใจและเรียกใช้ได้อย่างถูกต้อง ในขณะที่ REST Web service จะเป็นรูปแบบของซอฟต์แวร์ที่มองว่าข้อมูลต่าง ๆ เป็น Resource ซึ่งคนสามารถเรียกใช้ได้ผ่านทางโปรโตคอล HTTP และข้อมูลที่ส่งกลับมาให้ผู้ใช้เป็นข้อมูลรูปแบบ XML ใด ๆ ก็ได้ ในแง่ของผู้เรียกใช้ REST Web service ก็ขอเพียงแค่ให้ทราบ URL ของ REST Web service และการอ่านข้อมูล XML ก็จะดึงข้อมูลที่ตนเองต้องการได้

2.3.1 Representational State Transfer

REST (Representational State Transfer) เป็นแนวคิดในการมองเว็บทั้งหลายเป็นทรัพยากรของตน สามารถที่จะใช้รูปแบบใด ๆ ในการติดต่อกับทรัพยากรเหล่านี้ และสามารถรับรูปแบบใด ๆ ก็ได้ เช่น การใช้เว็บไซต์ผ่านทางเบราเซอร์ ก็ถือเป็น REST แบบหนึ่ง การเรียกใช้ REST แบบ Web service โดยทั่วไปมักจะส่งคำร้องตามด้วยค่ารับเข้า เช่น http://www.google.com/search?q=currency+exchange ทางเครื่องเมื่อประมวลผลเสร็จ ก็จะส่งข้อมูลตอบกลับมาในรูปแบบต่าง ๆ เช่น HTML, XML, JS, JSON แล้วแต่สถานการณ์

จุดอ่อนของ REST ในอดีตคือไม่มีศูนย์ข้อมูลกลางที่เก็บบริการไว้ จึงต้องค้นหาด้วยวิธีปรกติคือใช้ Search Engine ต่างๆ ช่วย เช่น Google, Yahoo โดยทั่วไปเราจะค้นหาบริการโดยใช้คำว่า API (Application Programming Interface) แต่ในปัจจุบันเริ่มมีการเก็บบริการเหล่านี้ไว้มากแล้ว การทำงานของ REST จะมี 2 ส่วนคือโปรแกรมฝั่งผู้รับบริการ (Client) ส่งคำร้องด้วยวิธีการ 4 วิธี คือ GET, PUT, DELETE, POST ไปยังฝั่งผู้ให้บริการ (Server) ซึ่งจะประมวลผลแล้วให้คำตอบกลับมา เราสามารถนำโครงสร้างนี้ไปประยุกต์ใช้อย่างไรก็ได้ เช่น ส่งข้อมูล รับข้อมูล ควบคุมอุปกรณ์เครื่องจักร ฯลฯ

2.3.2 API

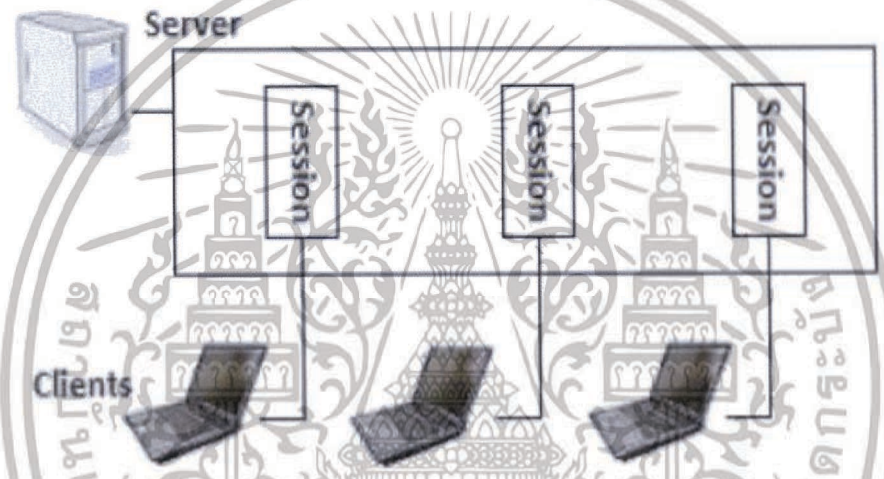
API (Application Programming Interface) คือช่องทางการเชื่อมต่อระหว่างเว็บไซต์หนึ่งไปยังอีกเว็บไซต์หนึ่ง หรือเป็นการเชื่อมต่อระหว่างผู้ใช้งานกับเครื่องแม่ข่าย หรือจากเครื่องแม่ข่ายเชื่อมต่อไปหาเครื่องแม่ข่าย ซึ่ง API นี้เปรียบได้กับภาษาคอมพิวเตอร์ที่ทำให้คอมพิวเตอร์สามารถสื่อสารและแลกเปลี่ยนข้อมูลกันได้อย่างอิสระ โดยที่เราไม่ต้องเข้าใจหรือไปแตะต้องซอร์สโค้ดของ API ขอเพียงทราบว่าต้องทำงานอย่างไร เรียกใช้อย่างไร ส่งค่าอะไร และได้รับค่าอะไรคืนกลับมา

ประโยชน์ของ API มีหลายอย่าง เช่น

1) ช่วยในการพัฒนาเว็บไซต์หรือโปรแกรมประยุกต์ได้ง่ายและรวดเร็วซึ่ง API จะเป็นตัวช่วยที่ไม่ต้องเข้าไปแก้ไขซอร์สโค้ดคำสั่งแม้แต่น้อย ทำให้สะดวกสบายในการนำไปใช้งาน
2) API สามารถรับส่งข้อมูลข้ามเครื่องแม่ข่ายได้ ในปัจจุบันเว็บไซต์ใหญ่ ๆ หลายเว็บไซต์มีการเปิดให้ใช้าน API ซึ่งเราอาจเห็นการใช้งาน API ได้มากขึ้นโดยเฉพาะเว็บไซต์ที่เกี่ยวข้องในด้านการติดต่อสื่อสาร อย่าง Social Network และ E-commerce

2.3.3 Session

Session คือการติดต่อแลกเปลี่ยนข้อมูลแบบกึ่งถาวรระหว่างสิ่ง 2 สิ่ง ตัวเครื่องแม่ข่ายนั้น สามารถสร้าง Session ให้แต่ละเครื่องลูกข่ายใช้เชื่อมต่อได้ ดังรูปที่ 2.3 เพื่อทำการรับรองคำขอที่ผู้ใช้งานส่ง Request มากกว่า 1 Request ในเวลาเดียวกันไปยังหนึ่งเว็บไซต์ ผู้ใช้งานแต่ละคนจะได้การเชื่อมต่อไปยังเครื่องแม่ข่ายด้วย Session ที่แยกกันต่างหาก Session เหมาะสำหรับจัดเก็บข้อมูลที่สำคัญในฝั่งเครื่องแม่ข่ายที่ปลอดภัยจากการโจมตี

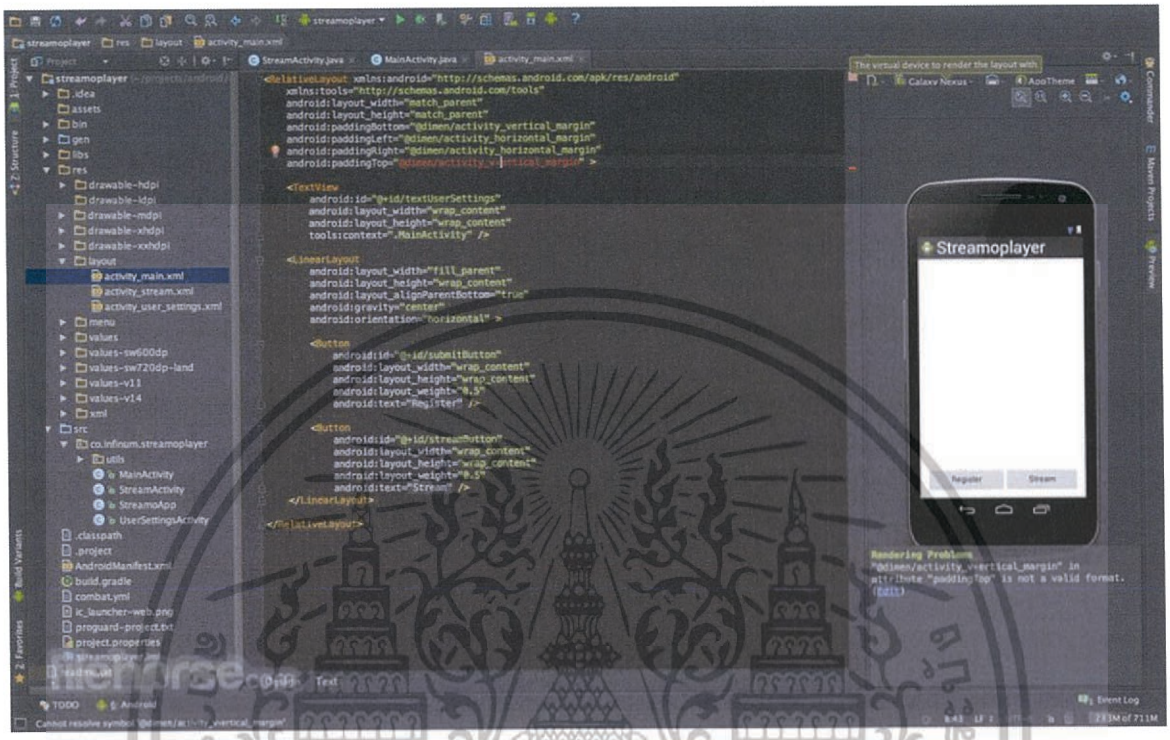


รูป 2.3 เครื่องแม่ข่ายสร้าง Session ให้แต่ละเครื่องลูกข่าย

2.4 Android

Android Application คือโปรแกรมประยุกต์ต่าง ๆ ที่สามารถติดตั้งใช้งานได้กับสมาร์ทโฟนที่ใช้ระบบปฏิบัติการ Android ซึ่งในปัจจุบันได้มีการพัฒนา Android Application กันมากเนื่องจาก Android เป็น OS ที่เป็น Open Source สามารถใช้งานได้ฟรี และติดตั้งได้กับสมาร์ทโฟนหรืออุปกรณ์ที่หลากหลาย และนักพัฒนาก็สามารถพัฒนา Android Application ได้ด้วยโน้ตบุ๊คหรือคอมพิวเตอร์ตั้งโต๊ะธรรมดาได้ เรียกได้ว่าความสามารถของ Android ที่สามารถทำงานร่วมกับ Hardware อย่างเป็นอิสระได้เกือบทุกอย่าง จึงทำให้ได้รับความนิยมจากองค์กรธุรกิจจำนวนมากมายและมีการนำ Android Application มาใช้งานร่วมกับธุรกิจหลาย ๆ ประเภท ทั้งโปรแกรมประยุกต์ที่สามารถโหลดมาใช้งานได้ทันที หรือโปรแกรมประยุกต์ที่ต้องซื้อหรือต้องเสียค่าบริการก็ตาม

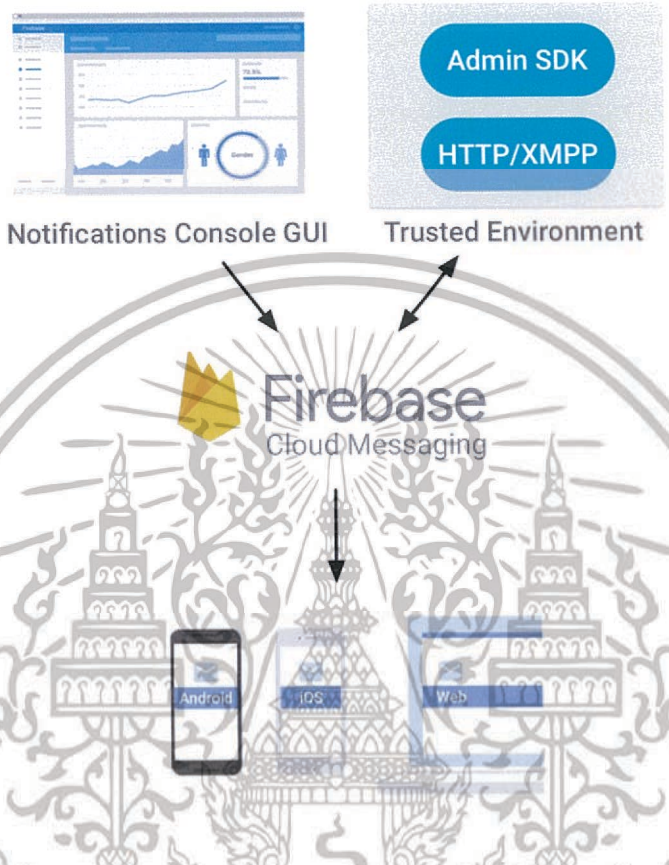
2.4.1 Android Studio



รูป 2.4 หน้าต่าง Android Studio

การพัฒนาโปรแกรมบนระบบปฏิบัติการแอนดรอยด์ไม่ใช่เรื่องยาก เพราะมี Android Studio เป็นเครื่องมือสร้างสภาพแวดล้อมในการพัฒนาอย่างเป็นทางการ (Official Integrated Development Environment) จาก Google เพื่อพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ วัตถุประสงค์ของ Android Studio คือเป็น IDE ที่สามารถพัฒนาโปรแกรมประยุกต์บนแอนดรอยด์โดยเฉพาะ ตัวอย่างหน้าตาของโปรแกรม Android Studio แสดงดังรูป 2.4 ภายในโปรแกรมจะมีส่วนของการเขียนโปรแกรม และ ส่วนการออกแบบ Interface ทำให้สามารถพัฒนาโปรแกรมได้โดยง่าย

2.4.2 Firebase Cloud Messaging



รูป 2.5 ภาพรวมของ Firebase Cloud Messaging

ในรูปที่ 2.5 จะแสดงถึงระบบทั้งหมดของ Firebase Cloud Messaging ซึ่งเป็นบริการหนึ่งของชุดพัฒนา Firebase ของบริษัทกูเกิ้ล โดยจะมีส่วน Console GUI ที่ใช้งานบนเว็บไซต์ที่สามารถควบคุมการส่งข้อความ หรือการแจ้งเตือนต่าง ๆ ไปยังอุปกรณ์โทรศัพท์หรือส่งผ่านเว็บไซต์ต่าง ๆ ของผู้ใช้

2.4.3 QR Code

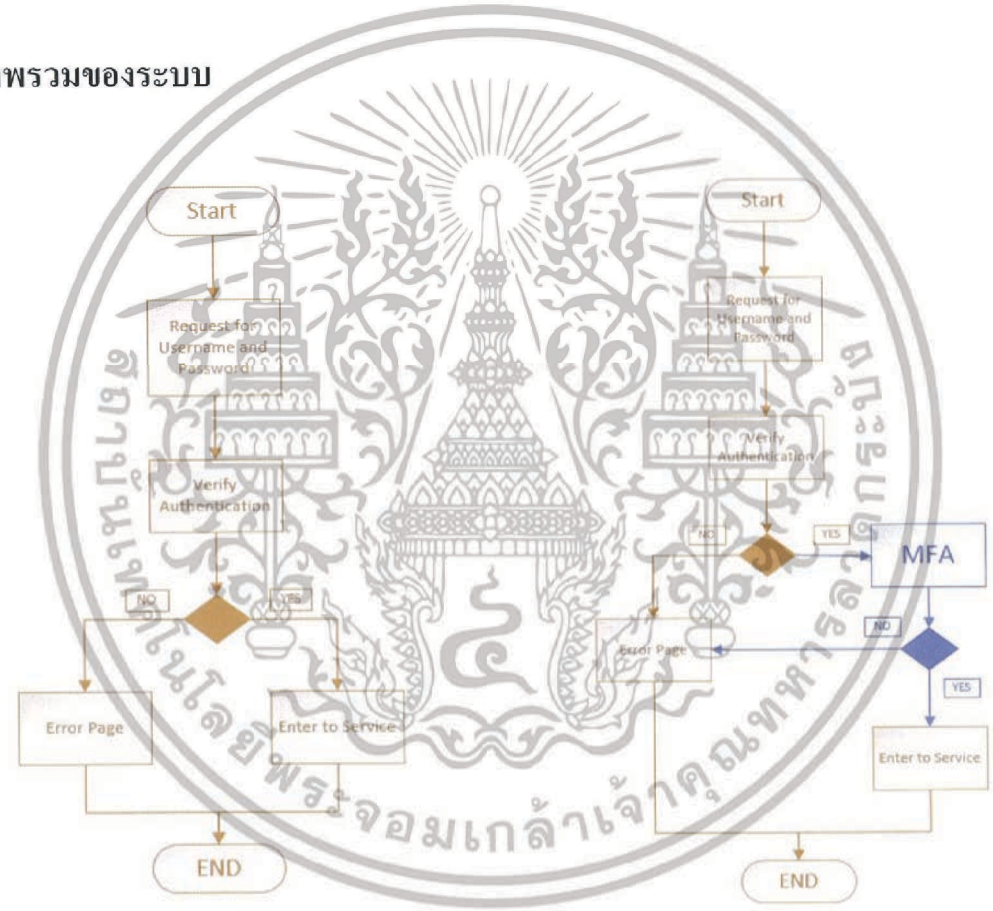
ในปัจจุบัน QR Code (Quick Response Code) ถูกนำมาใช้อย่างกว้างขวางในงานด้านอุตสาหกรรม งานทางธุรกิจ โฆษณา และการใช้งานส่วนบุคคล QR Code เป็นบาร์โค้ดชนิดหนึ่งที่พบเห็น ได้ตามบนหนังสือนิตยสาร หนังสือพิมพ์ ป้ายสินค้า กล่องหรือกระป๋องเครื่องดื่ม เป็นต้น QR Code สามารถอ่านได้โดยใช้โทรศัพท์มือถือที่มีกล้องถ่ายรูปอยู่ในตัว โดยผ่านโปรแกรมการอ่าน QR Code ก็ สามารถแสดงข้อมูลข่าวสาร หรือเว็บไซต์ที่ซ่อนอยู่ในตัว QR Code ได้

บทที่ 3

การออกแบบระบบ

บทนี้จะกล่าวถึงภาพรวมของระบบทั้งหมดว่าระบบประกอบไปด้วยอะไรบ้าง และมีการทำงานอย่างไร จากนั้นจะกล่าวถึงเครื่องมือที่ใช้การทำงาน และเทคนิค OTP ที่จะใช้ภายในโครงงานนี้ตามลำดับ

3.1 ภาพรวมของระบบ



ก) ข)

รูป 3.1 การพิสูจน์ตนในแบบต่าง ๆ

ก) การพิสูจน์ตนแบบปัจจัยเดี่ยว

ข) การพิสูจน์ตนแบบหลายปัจจัย

ระบบการยืนยันตัวตนแบบทั่วไป ดังรูปที่ 3.1ก คือ ผู้ใช้งานจะต้องทำการใส่ Username และ Password จากนั้นระบบก็จะทำการตรวจสอบข้อมูลว่าข้อมูลที่ได้มาถูกต้องหรือไม่ ส่วนในรูปที่ 3.1ข เป็นการนำระบบยืนยันตัวตนรอบที่สองไปติดตั้งหลังจากได้ยืนยันตัวตนรอบแรกเรียบร้อยแล้ว จากนั้นจะต้องยืนยันตัวตนรอบที่สองผ่าน Application การพัฒนาโปรแกรมจะแบ่งออกเป็น 2 ส่วน คือ

1) ฝั่งของ Server โดยพัฒนา Web Service เพื่อนำไปติดตั้งกับ Server เพื่อให้ Server เรียกใช้ งานในการประมวลผลในการลงทะเบียนคำนวณ OTP

2) ฝั่งของ Client โดยพัฒนา Android Application สำหรับ Smart phone เพื่อใช้ในการคำนวณ OTP และรับ Request ในฝั่งของผู้ใช้

ภาพรวมของระบบทั้งหมดจะเป็นดังรูปที่ 3.2

รูป 3.2 ภาพรวมของระบบ

จากรูปที่ 3.2 สามารถอธิบายขั้นตอนการทำงานของระบบได้ดังต่อไปนี้

1) ผู้ใช้งานทำการใส่ชื่อผู้ใช้และรหัสผ่านในหน้าเข้าสู่ระบบ (PHP , HTML)

2) ระบบทำการส่งข้อมูลไปตรวจสอบกับดาต้าเบส (MySQL) เพื่อทำการยืนยันตัวตนผู้ใช้

3) หลังจากทำการตรวจสอบชื่อผู้ใช้และรหัสผ่าน ถ้าชื่อผู้ใช้งานและรหัสผ่านถูกต้องก็จะสามารถเข้าสู่ระบบได้ ถ้าไม่ถูกต้องก็ไม่สามารถเข้าสู่ระบบได้

4) สามารถใช้งานระบบยืนยันตัวต้นขั้นที่สองได้โดยติดตั้งโปรแกรมประยุกต์และสมัครบริการยืนยันตัวตนระบบจะส่ง QR Code ให้ผู้ใช้งานสแกน เพื่อสร้าง สร้างรหัส OTP

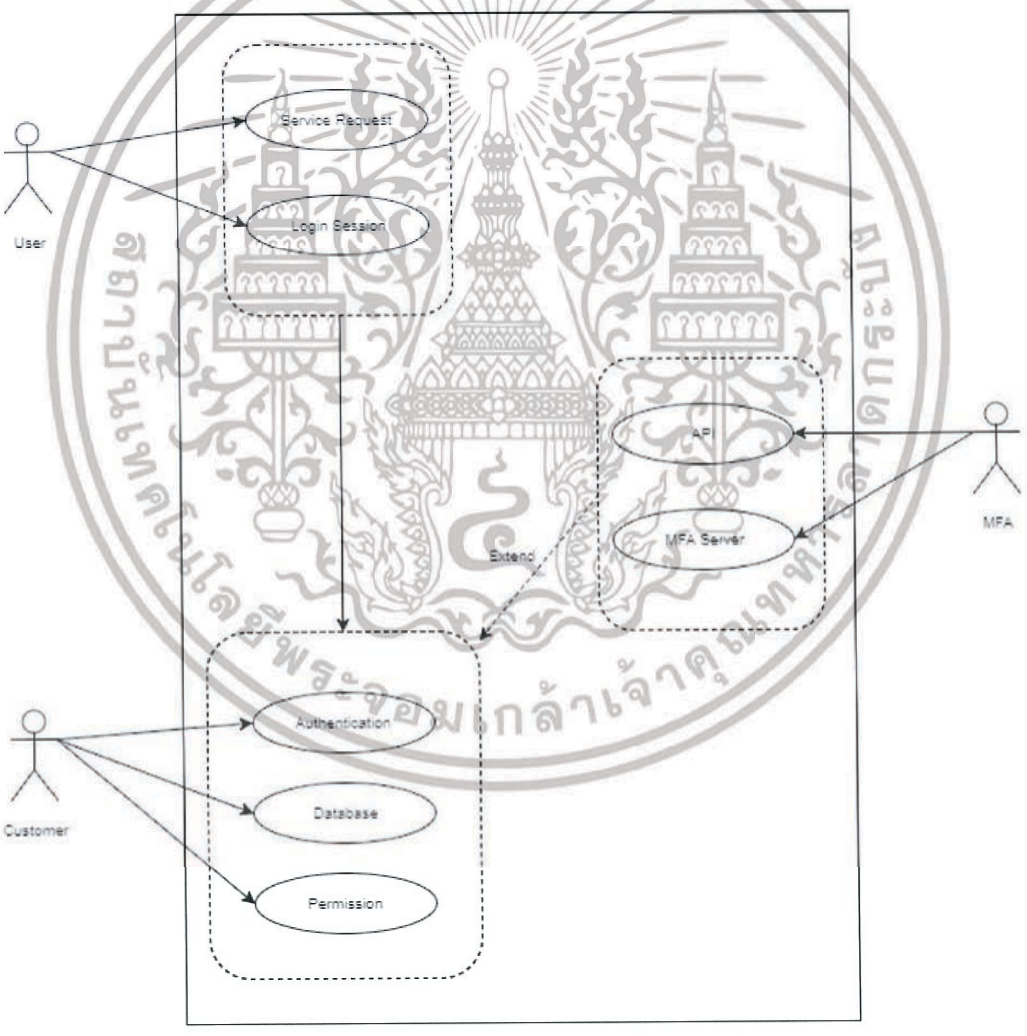
5) ผู้ใช้งานกรอก OTP , Server จะตรวจสอบ OTP ทำการตอบกลับว่ายอมรับหรือปฏิเสธ

6) เมื่อผู้ใช้งานเข้าสู่ระบบ ที่ฝั่งลูกค้าก็จะทำการส่งข้อมูลมาที่ MFA Server

7) ระบบจะทำการส่ง Challenge ไปที่โปรแกรมประยุกต์

8) ผู้ใช้งานทำการตอบกลับมาที่ว่ายอมรับหรือปฏิเสธที่โปรแกรมประยุกต์

9) ถ้าผู้ใช้งานเลือกยอมรับจะสามารถเข้าสู่ระบบได้ ถ้าปฏิเสธก็ไม่สามารถเข้าสู่ระบบได้



รูป 3.3 Use Case Diagram การใช้ระบบยืนยัน

ในรูปที่ 3.3 แสดงถึง Use Case Diagram การใช้ระบบยืนยัน โดยจะแบ่งผู้ที่อยู่ในระบบเป็น 3 ฝั่ง ดังนี้

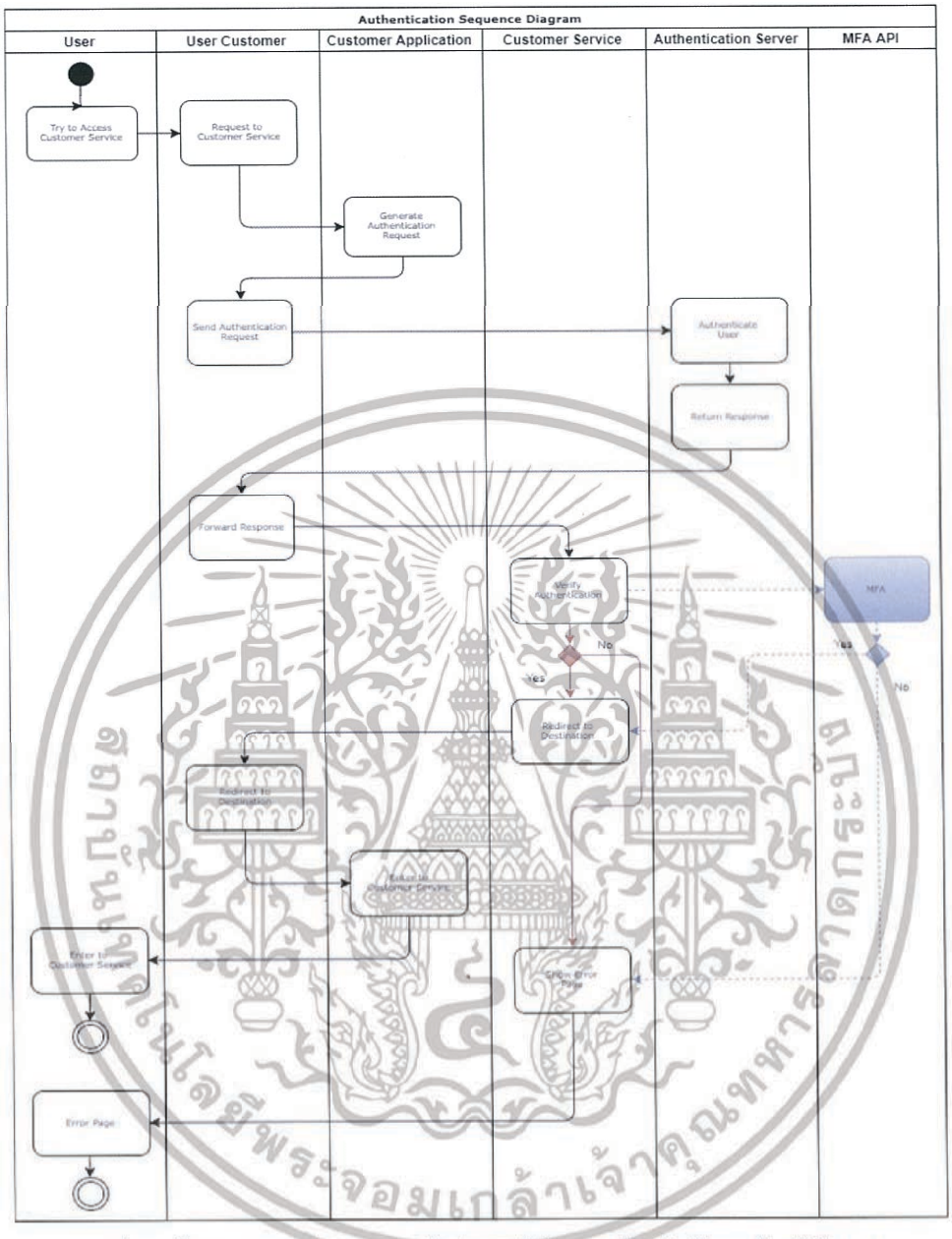
1) ฝั่งของผู้ใช้งาน (User) ผู้ใช้งานจะต้องเป็นคนที่ทำการร้องขอและเรียกใช้งานระบบต่าง ๆ และผู้ใช้งานจะต้องทำการยืนยันตัวตนผ่านระบบของฝั่งลูกค้าผู้ให้บริการ

2) ฝั่งของผู้ให้บริการ (Customer) จะต้องมีระบบยืนยันตัวตนให้ผู้ใช้งานทำการยืนยันตัวตน ต้องมีฐานข้อมูลเพื่อใช้เก็บข้อมูลของผู้ใช้งาน และต้องให้การอนุญาตติดตั้งระบบของการยืนยันตัวขั้นที่สอง

3) ฝั่งยืนยันตัวตนขั้นที่สอง (MFA) โดยทางเราจะมี Server ที่ใช้ประมวลผลในการยืนยันตัวตน และ การให้บริการเรียกใช้งานผ่าน API

3.1.1 ไดอะแกรมก่อนและหลังทำการใช้ระบบยืนยันตัวตนโดยใช้ API

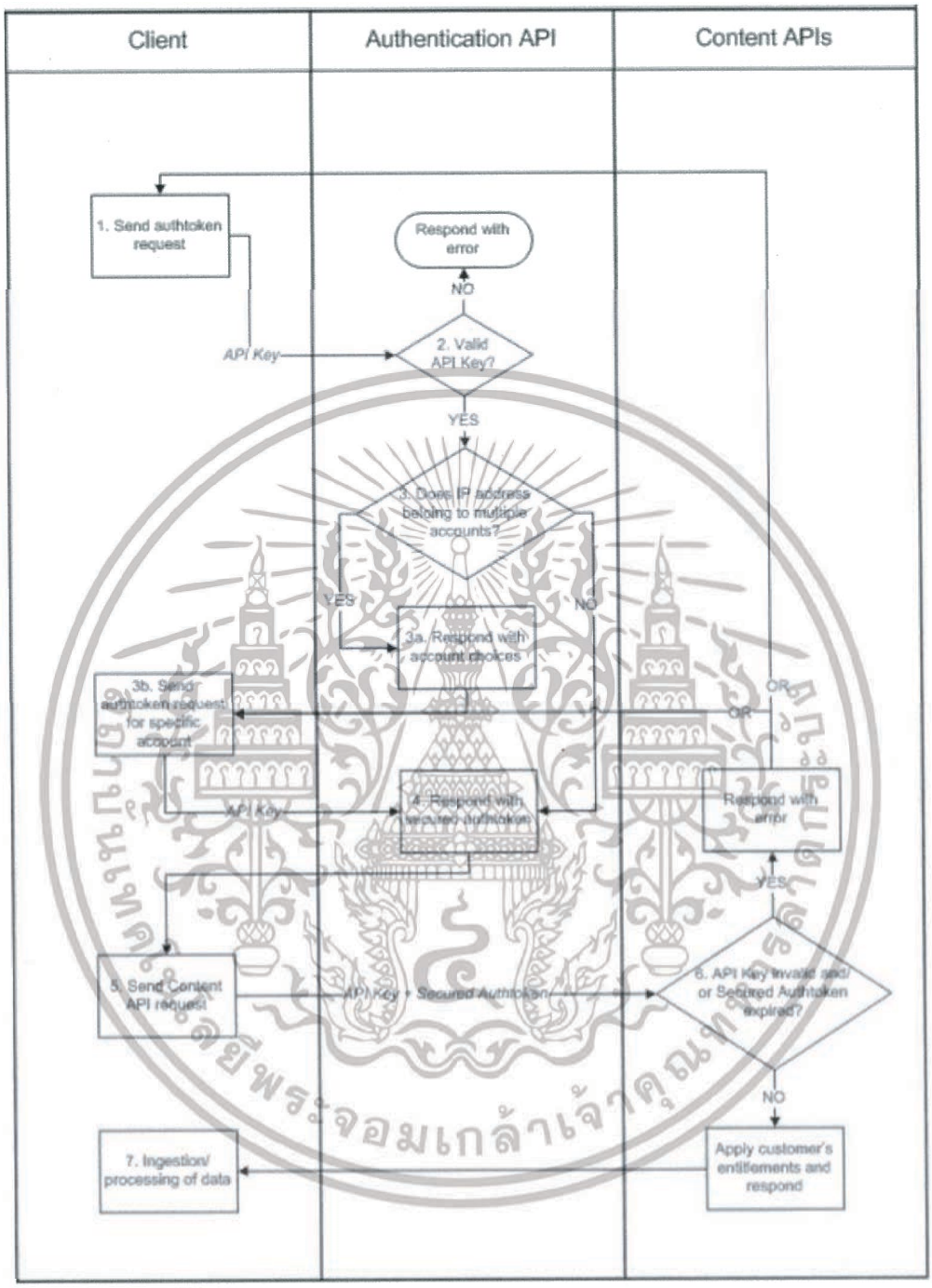
ในรูปที่ 3.4 จะเป็นไดอะแกรมก่อนและหลังทำการใช้ระบบยืนยันตัวตนโดยใช้ API คือ ผู้ใช้งานจะต้องทำการใส่ Username และ Password จากนั้นระบบก็จะทำการตรวจสอบข้อมูลว่าข้อมูลที่ได้มาถูกต้องหรือไม่ ส่วนในกรอบสีฟ้าเป็นการนำระบบยืนยันตัวตนรอบที่สองไปติดตั้งหลังจากได้ยืนยันตัวตนรอบแรกเรียบร้อยแล้ว เพื่อทำให้ระบบยืนยันตัวตนแบบทั่วไปกลายเป็นระบบยืนยันตัวตนสองชั้น



รูป 3.4 ไดอะแกรมก่อนและหลังทำการใช้ระบบยืนยันตัวตนโดยใช้ API

3.1.2 API สำหรับผู้ใช้งานระบบการยืนยันตัวตน

ที่เครื่องแม่ข่ายของผู้ใช้งานระบบยืนยันตัวตนนั้น ไม่จำเป็นต้องทำการลงระบบทั้งหมดไว้ที่เครื่องแม่ข่ายของผู้ใช้งาน แต่ทำการใช้ API ในการเชื่อมต่อกับระบบยืนยันตัวตน ดังรูปที่ 3.5



รูป 3.5 ไดอะแกรม API สำหรับผู้ใช้งานระบบการยืนยันตัวตน

ส่วนของ Code API ที่จะเพิ่มในระบบยืนยันตัวตนแบบเดิมนั้น จะทำการติดตั้งส่วนที่ผู้ใช้งานกดสมัครใช้งานบริการยืนยันตัวตนขั้นที่สอง โดยในระบบต้นแบบได้ทำการเขียนโดยใช้ภาษา PHP และ HTML ในการเรียกใช้งาน API ดังกล่าว ซึ่งผู้ให้บริการลูกค้าไม่จำเป็นต้องทำการเขียนระบบยืนยันตัวตนใหม่ทั้งหมด ซึ่งทำให้มีความง่ายในการติดตั้งระบบเพิ่มเติม แต่เรียกใช้บริการผ่าน API ข้อจำกัดของระบบในตอนนี้คือ ระบบของผู้ให้บริการจะต้องเป็น ภาษา PHP และ HTML เท่านั้น ตัวอย่าง Code ที่เพิ่มในระบบจะเป็นดังโปรแกรมที่ 3.1

โปรแกรม 3.1 API generate-otp

```
<?php
mysql_connect("localhost","root","root");
mysql_select_db("data");
$strSQL = "SELECT * FROM member WHERE
     UserID = '".$_SESSION['UserID']."' ";
$objQuery = mysql_query($strSQL);
$objResult = mysql_fetch_array($objQuery);

$url = 'http://161.246.5.8/generate-otp/totp/provision-uri/';
$ch = curl_init($url);
$jsonData = array(
    'timeout' => 30,
    'name' => $objResult["Username"],
    'issuer_name' => 'Otter'
);
$jsonDataEncoded = json_encode($jsonData);
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonDataEncoded);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json'));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($ch);
curl_close($ch);
?>
```

3.1.3 โครงสร้างโปรแกรมในส่วนของ Server

3.1.3.1 OTP Web Service

Web service เป็นซอฟท์แวร์ที่พัฒนาขึ้นที่มีความสามารถในการคำนวณค่า OTP สำหรับติดตั้งเข้ากับ Server และใช้ Web Application เป็นตัวเรียกใช้งานผ่านค่าพารามิเตอร์และ URL ตามรูปแบบของการเรียกใช้ Web Service ที่พบเห็นโดยทั่วไป โดยซอฟต์แวร์ที่ได้พัฒนาขึ้นนี้ จะนำไปติดตั้งกับองค์กรใด ๆ ที่ต้องการความมั่นคงปลอดภัยในการยืนยันตัวตนให้มีความแข็งแกร่งมากขึ้น จากการยืนยันตัวตนแบบเดิมที่มีเพียงชื่อผู้ใช้และรหัสผ่านเท่านั้น ซึ่งปัจจุบัน สามารถถูกดักจับได้ไม่ยาก เกิดการรั่วไหลได้ง่าย ระบบนี้จึงไม่ได้รับความน่าเชื่อถืออีกต่อไป เมื่อองค์กรนั้นนำซอฟท์แวร์นี้ไปใช้งาน จะช่วยเพิ่มความมั่นใจให้กับองค์กรนั้น ๆ ในระดับที่ดีกว่าเดิมพอควร Web service ถูกพัฒนาขึ้นโดยใช้เทคโนโลยี REST เป็นแนวคิดในการมองเว็บทั้งหลายเป็นทรัพยากรของเรา เราสามารถที่จะใช้รูปแบบใด ๆ ในการติดต่อกับทรัพยากรเหล่านี้ และสามารถรับค่า Response เป็นรูปแบบใด ๆ ก็ได้ การเรียกใช้ REST ในโครงงานปริญญานิพนธ์นี้จะมีการร้องขอผ่าน URL ซึ่งเป็นรูปแบบมาตรฐานในการร้องขอ REST แบบ POST Method

3.1.3.2 Web Application

ใช้สำหรับจำลองหน้า Web Application ซึ่งคล้ายกับหน้าเว็บทั่วไป ภาษาหลักที่ใช้ในการพัฒนา คือ PHP, JavaScript, SQL, HTML, CSS เป็นต้น ภายในมีส่วนประกอบดังนี้

1) Login System คือ ระบบสมาชิกทั่วไป ที่จะต้องมีบัญชีเป็นของตัวเอง มีชื่อผู้ใช้และรหัสผ่านสำหรับทำการเข้าใช้งานระบบ

2) Database คือ ฐานข้อมูลในการจัดเก็บข้อมูลผู้ใช้ โดยใช้ SQL ในการจัดเก็บฐานข้อมูลของผู้ใช้ ซึ่งประกอบไปด้วย รหัสประจำตัวของผู้ใช้ (ID), ชื่อผู้ใช้, รหัสผ่าน และสถานการณ์เปิดใช้งานการยืนยันตัวตนแบบสองขั้นตอน

3) QR Code ที่ได้นำมาฝังไว้นี้จุดประสงค์เพื่อใช้สำหรับเป็นช่องทางในการส่ง Challenge String ไปยังลูกค้าในรูปแบบที่ไม่สามารถอ่านออกได้โดยง่าย และสามารถ ป้องกันการถูกดักจับข้อมูลผ่านเครือข่ายได้ในระดับหนึ่ง ซึ่งแน่นอนว่าจะมีความมั่นคง ปลอดภัยกว่าการ ส่งในรูป Plain text ธรรมดา

3.1.4 โครงสร้างโปรแกรมในส่วนของ Client

3.1.4.1 Mobile Application

ในส่วนของการลงทะเบียนเพื่อให้โปรแกรมประยุกต์นั้นสามารถใช้งานได้ จะต้องมีการกำหนดค่าเริ่มต้นระหว่าง Server และผู้ใช้ก่อน โดยผู้ใช้จำเป็นต้องมีการลงทะเบียนผ่านหน้า Web Application ตามขั้นตอนที่แต่ละองค์กรกำหนด เมื่อการลงทะเบียนผ่านหน้าเว็บเสร็จสมบูรณ์แล้ว ผู้ใช้จะได้รับค่าหนึ่งที่ถูกส่งมาบนหน้าเว็บในรูปแบบของ QR Code เพื่อให้ผู้ใช้นำโทรศัพท์มือถือสมาร์ตโฟนที่ใช้ระบบปฏิบัติการ Android ซึ่งต้องมีคุณสมบัติในการถ่ายภาพ มาสแกน QR Code เพื่อรับค่านั้นไปประมวลผลและบันทึกลงฐานข้อมูลของโทรศัพท์มือถือ

3.1.4.2 Mobile OTP

ในส่วนของ Mobile OTP คือ เมื่อมีการลงทะเบียนลงบนโทรศัพท์มือถือเสร็จแล้ว จะปรากฏรายการบัญชีที่ผู้ใช้ได้ลงทะเบียนไว้ ผู้ใช้สามารถเลือกสลับเพื่อ Generate OTP ได้ การ Generate OTP มีกระบวนการดังนี้

1) เมื่อ Client ต้องการเข้าสู่ระบบ ฝั่ง Server ก็ จะส่งค่า Challenge String มาให้ในรูปแบบของ QR Code แสดงบนหน้า Website

2) Client ต้องนำค่า Challenge String ดังกล่าวมาป้อนบน Mobile app ซึ่งจะใช้วิธีสแกน QR Code

3) จากนั้นโปรแกรมประยุกต์จะสร้างรหัส OTP ขึ้นมา หน้าจอของสมาร์ทโฟนก็จะแสดงผลการคำนวณ OTP เพื่อให้ผู้ใช้นั้นได้นำไปทำการยืนยันตัวตนต่อไป

3.2 เครื่องมือที่ใช้

การดำเนินงานมีความจำเป็นต้องใช้อุปกรณ์ช่วยเหลือในการสร้างรหัส การสร้างระบบ และการใช้งานระบบ ดังต่อไปนี้

3.2.1 Django

Django เป็นเฟรมเวิร์กชนิด MVC (Model-View-Controller) ในการสร้างเว็บไซต์ทางฝั่งเครื่องแม่ข่าย โดยใช้ภาษา Python ในการดำเนินการต่าง ๆ เช่น การสร้างโมเดลของฐานข้อมูล การหาเส้นทางภายในเว็บไซต์ การให้บริการหน้าเว็บไซต์ต่าง ๆ เป็นต้น ซึ่งเป็นเฟรมเวิร์กที่นักพัฒนาและออกแบบระบบบริการเว็บ หรือ REST API นิยมใช้กัน เนื่องจากเป็นเฟรมเวิร์กที่เรียนรู้ได้ง่าย มีเอกสารประกอบที่ถี่ถ้วนชัดเจน อธิบายผ่านตัวอย่างให้นักพัฒนามือใหม่เข้าใจระบบการทำงานของเฟรมเวิร์กได้ ในงานชิ้นนี้ ได้นำ Django มาให้บริการในส่วนของ Authentication Server ซึ่งสร้างในรูปแบบของ REST API ให้เว็บอื่น ๆ ใช้งานบริการการพิสูจน์ตน

3.2.2 Django REST Framework

Django REST Framework เป็นแอพพลิเคชันของ Django (ในเชิงการพัฒนาอาจจะเรียกว่าส่วนเสริมก็ไม่ผิดเท่าใดนัก) ที่อำนวยความสะดวกให้นักพัฒนาในการสร้างและใช้งานแบบ REST ต่าง ๆ ให้สามารถติดต่อกับระบบอื่น ๆ ที่ใช้งาน REST ได้

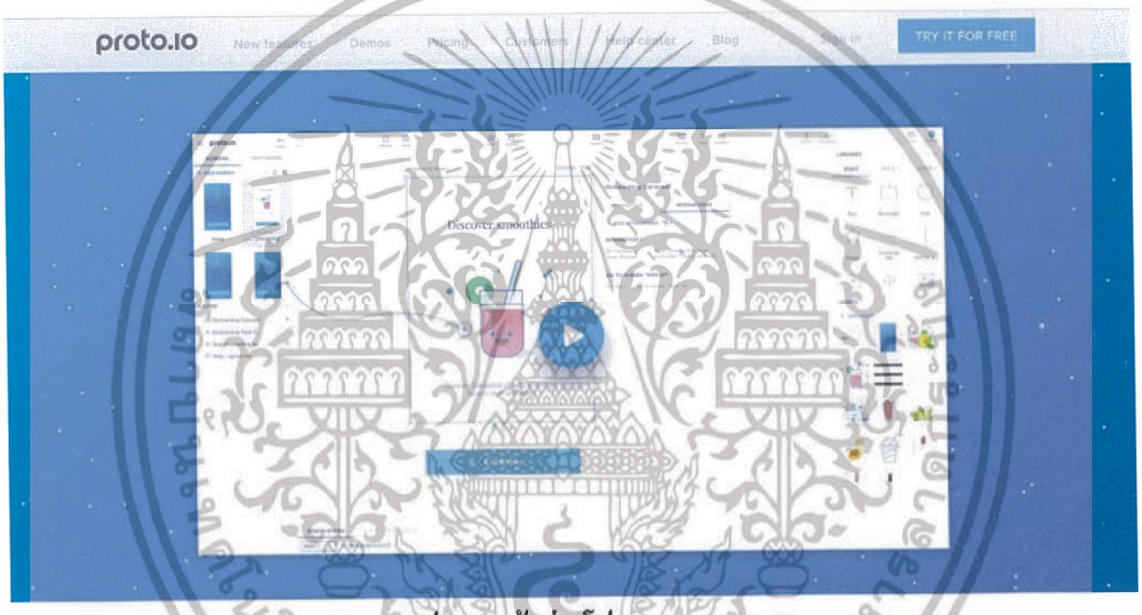
3.2.3 PyOTP

เนื่องจากผู้จัดทำได้ตัดสินใจสร้างระบบให้บริการเว็บของเครื่องแม่ข่ายด้วยภาษา Python (Django) จึงเลือกใช้ไลบรารี OTP ในภาษา Python ชื่อ PyOTP ซึ่งสามารถสร้างรหัส OTP ได้ทั้งแบบ HOTP และ TOTP รวมถึงการสร้างกุญแจลับที่เข้ารหัสแบบ Base32 16 อักขระ ในงานชิ้นนี้ ได้นำ PyOTP มาใช้ใน Authentication Server ในการบริการการสร้างรหัส สร้างกุญแจลับแบบ Base32 และตรวจสอบรหัสที่ส่งมายัง Authentication Server

3.2.4 Aerogear-OTP-Java

การสร้าง OTP ทางด้านของโปรแกรมประยุกต์แอนดรอยด์นั้น จำเป็นจะต้องใช้ไลบรารีที่สร้าง OTP จากกุญแจลับแบบ Base32 เช่นเดียวกับ PyOTP และต้องเป็นไลบรารีที่ใช้งานได้สะดวกและมีความสามารถอำนวยความสะดวกในการสร้างบริการให้ผู้ใช้ได้ เช่น การใช้ QR Code ในการเริ่มต้นการแลกเปลี่ยนกุญแจลับระหว่างทั้งสองฝ่าย ทาง Aerogear ได้สร้างไลบรารีที่สามารถนำไปใช้ได้ทันทีใน Android Studio จึงเลือกใช้ไลบรารีดังกล่าว

3.2.5 Proto.io



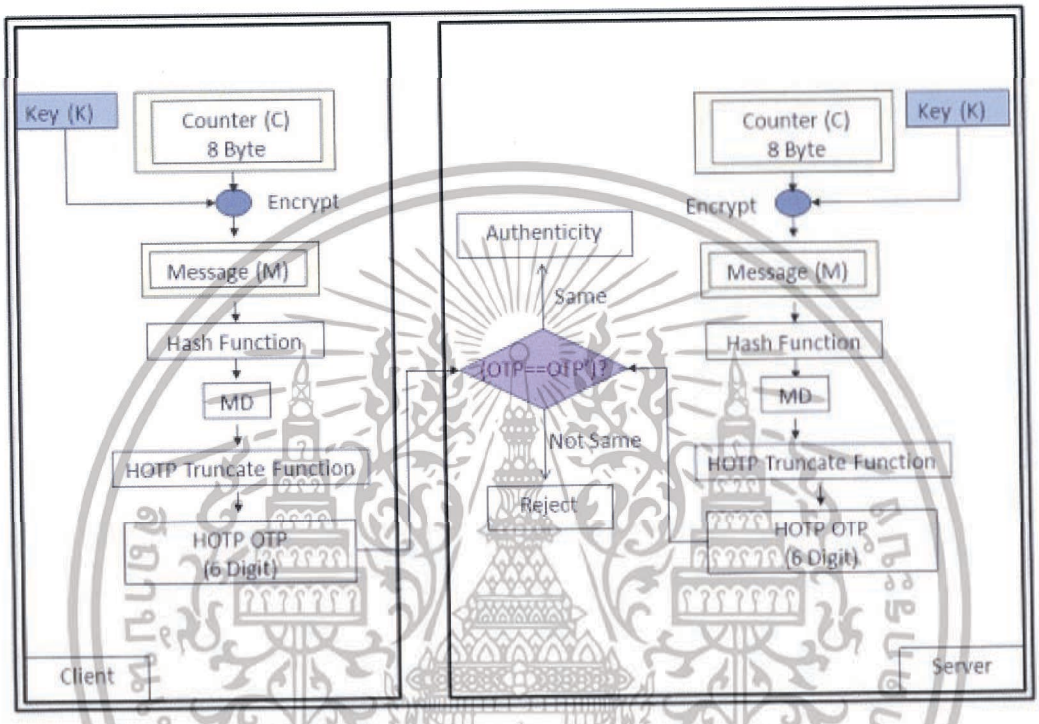
รูป 3.6 หน้าต่างโปรแกรม Proto.io

Proto.io เป็นบริการการออกแบบหน้าตาส่วนผู้ใช้งาน (User Interface) มีประโยชน์ในการร่างงาน กำหนดจุดการให้บริการต่าง ๆ ของโปรแกรมประยุกต์ หน้าตาของ Proto.io จะเป็นดังรูปที่ 3.6 โปรแกรมจะมีอำนวยความสะดวกต่าง ๆ ที่ทำให้ผู้ใช้งานสามารถออกแบบ Prototype ของ Application ก่อนที่จะนำไปใช้งานจริงได้

3.3 เทคนิค OTP

ในการสร้างรหัส OTP นั้น จะมีเทคนิคต่าง ๆ ดังนี้

3.3.1 HOTP



รูป 3.7 กระบวนการทำงานของ HOTP

HOTP ได้มีการพัฒนาขึ้นบนพื้นฐานของ HMAC เพื่อสร้าง OTP จาก Client และตรวจสอบความถูกต้องของ OTP โดย Server ได้ สำหรับการออกแบบ HOTP จะสนับสนุนการนำไปใช้ใน Security Token ที่ผู้ใช้ถือครองอยู่ เรียก Security Token นั้นว่า OTP Token เพื่อสร้าง OTP สำหรับ และใช้เป็น OTP นั้นในการยืนยันตัวตนของผู้ใช้ต่อไป สามารถอธิบายกระบวนการสร้างและตรวจสอบ OTP ได้ดังรูปที่ 3.7 ซึ่งมีขั้นตอนดังต่อไปนี้

1) HOTP Client จะต้องมีการจัดเก็บกุญแจหลัก (Master Key) เพื่อใช้ในขั้นตอนการเข้ารหัสแบบ AES 14

2) HOTP Client สร้างตัวนับ (Counter) เพื่อสร้างข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลา ซึ่ง Counter นี้จะเริ่มจาก 1 และเพิ่มขึ้นเสมอ

3) นำ Key ที่ได้จาก ขั้นตอนที่ 1 มาเข้ารหัสข้อมูล Counter จะได้ Message ในขั้นตอนนี้จะเห็นได้ Message ที่ได้ทุกๆ ครั้ง จะเป็นข้อมูลใหม่เสมอ หากพิจารณาจากขั้นตอนที่ 2

4) นำ Message ที่ได้ไปสร้าง MD ด้วย SHA-1 จะได้ โดยมี Output ทั้งหมด 160 bits (20 Byte) ตามมาตรฐานของ SHA-1

5) นำ MD ที่ได้เข้าสู่ HOTP Truncate Function เป็น Function ที่ต้องการลดจำนวน Output bit จาก 20 Byte ให้เหลือเพียง 4 Byte เพื่อนำ Output ที่ได้มาแปลงเป็น OTP ที่มีลักษณะเป็นตัวเลข 6 ตัว จากนั้นส่ง OTP ที่ได้ให้ Server

6) เมื่อ Server ได้รับ OTP จาก Client ทางฝั่ง Server เองก็จะมีขั้นตอนเช่นเดียวกันกับ Client ทุกประการ นั้นคือใช้ Master Key และ Counter ตัวเดียวกันกับ Client โดย Counter นั้นจะเพิ่มขึ้นตลอดทุกๆ ครั้งที่มีการร้องขอเพื่อยืนยันตัวตน นำ OTP จาก Client มาเปรียบเทียบ กับ OTP ที่ Server สร้างขึ้น หากตรงกันก็แสดงว่าผู้ใช้ยืนยันตัวตนถูกต้อง

3.3.2 TOTP

เทคนิคนี้ทั้งเครื่องลูกข่ายและแม่ข่ายจะมีคุณแจลับที่เหมือนกัน ใช้ขั้นตอนวิธีการคำนวณเช่นเดียวกับ HOTP เพียงแต่ TOTP ใช้เวลาที่ได้จากนาฬิกามาตรฐาน (ที่นิยมคือ Unix Time) ในการคำนวณแทนตัวนับ นั่นคือ รหัส OTP ที่สร้างขึ้นมาได้นั้น จะมีช่วงอายุขัยที่แน่นอน เช่น 30 วินาที, 3 นาที ผู้ใช้ต้องใช้งานรหัสผ่านชุดนั้น ๆ ให้ทันเวลา เนื่องจาก TOTP ใช้เวลาในการสร้างรหัสผ่าน จึงสามารถกล่าวได้โดยสรุปว่า หากเครื่องลูกข่ายและเครื่องแม่ข่ายเก็บค่าเวลาปัจจุบันไม่ตรงกัน จะทำให้ OTP ที่สร้างขึ้นมาจากทั้งสองฝ่ายนั้นไม่ตรงกัน ผู้ใช้จะไม่สามารถพิสูจน์ทราบตัวตนได้

3.3.3 CROTP

เทคนิคนี้จำนวนสุ่ม (PIN) ที่เลือกจากเครื่องแม่ข่ายโดยการตรวจสอบสิทธิ์จะถูกส่งไปยังผู้ใช้งานให้ป้อนค่า PIN แล้วส่งการตอบกลับไปยังเครื่องแม่ข่าย

จากเทคนิคทั้งหมดที่กล่าวข้างต้นนี้ ทางผู้จัดทำได้เลือกใช้การเข้ารหัสลับแบบ TOTP เพราะการเข้ารหัสแบบ HOTP นั้น อยู่ภายใต้ RFC 4226 ที่ประกาศใช้ในเดือนธันวาคมปี 2005 ซึ่งเมื่อเทียบกับ TOTP ที่อยู่ภายใต้มาตรฐาน RFC 6238 ประกาศใช้ในเดือนพฤษภาคมปี 2011 แล้ว TOTP มีความเป็นปัจจุบันกว่า และเมื่อเทียบระหว่าง TOTP กับ HOTP แล้ว TOTP ใช้เวลาในการสร้างรหัสผ่านที่มีอายุขัย ทำให้ผู้ไม่ประสงค์โจมตีโดยการคาดเดารหัสผ่าน (Brute Force Attack) ได้ยากลำบาก

บทที่ 4

วิธีการทดลองและผลการทดลอง

4.1 การสร้างและติดตั้งระบบ

ระบบทั้งระบบมีส่วนสำคัญในการดำเนินงานอยู่ 3 ส่วน ได้แก่

4.1.1 REST API (Django/Django REST Framework)

ก่อนจะกล่าวถึงการดำเนินการต่าง ๆ ของ API จะขอกล่าวถึงการตั้งค่าสภาพแวดล้อมของ Django ให้พร้อมเสียก่อน โดยให้ทำการตั้งค่าระบบ Django ในไฟล์ settings.py ดังโปรแกรมที่ 4.1

โปรแกรม 4.1 settings.py

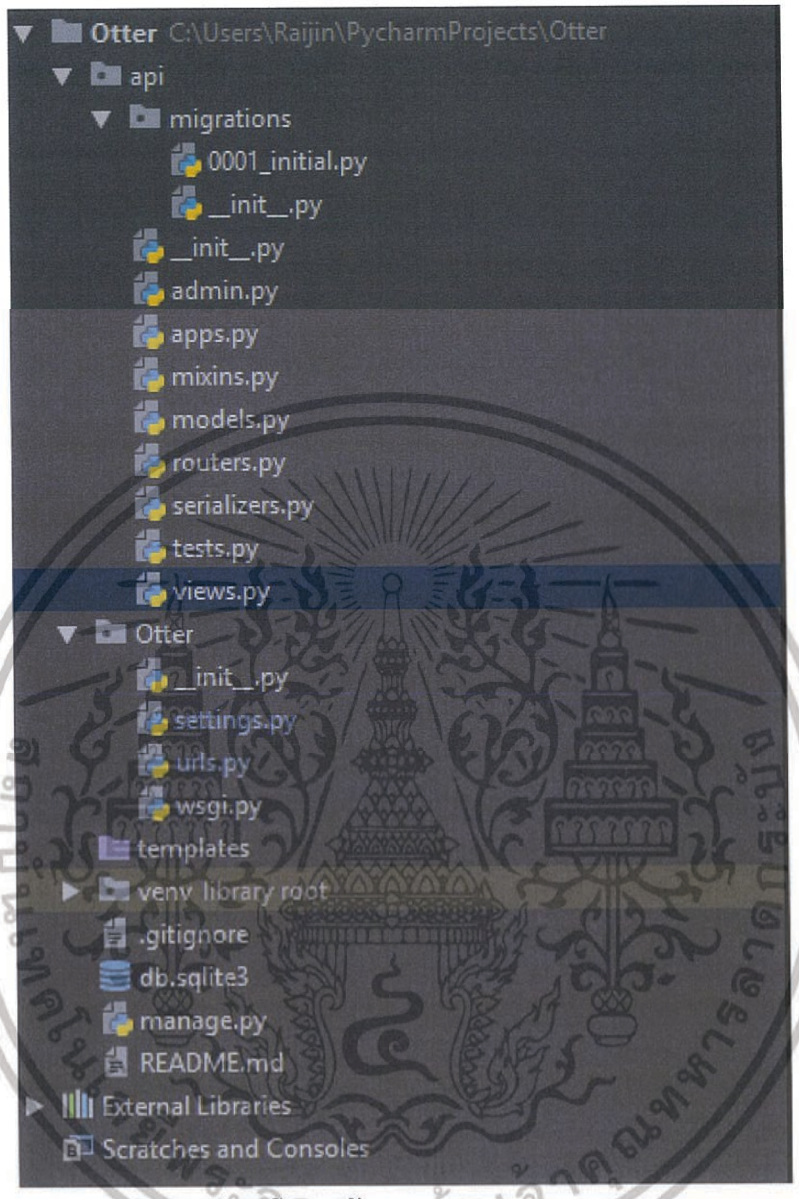
```
INSTALLED_APPS = [
      ...,
      'django.contrib.sites',
      'rest_framework',
      'rest_framework.authtoken',
      'allauth',
      'allauth.account',
      'allauth.socialaccount',
      'rest_auth',
      'rest_auth.registration',
      'fcm_django',
      'api',
]
SITE_ID = 1
EMAIL_BACKEND =
'django.core.mail.backends.console.EmailBackend'
FCM_DJANGO_SETTINGS = {
      'FCM_SERVER_KEY': <FCM_SERVER_KEY>,
}
```

ใน INSTALL_APPS ทำการเรียกใช้ไลบรารีประกอบ 4 ชุด คือ Django REST Framework ใช้สำหรับการทำ REST API ด้วย Django, REST_Auth ใช้สำหรับการจัดการต่าง ๆ เกี่ยวกับผู้ใช้และการพิสูจน์ตนผ่าน API, AllAuth มีไว้ให้บริการ REST_Auth.registration และ FCM_Django สำหรับการจัดการการแจ้งเตือนแบบพุชกับสมาร์ทโฟน โดย FCM_SERVER_KEY นั้น ต้องเป็นกุญแจลับที่ได้จาก Firebase Console ของ Google เท่านั้น

การสร้าง API ที่รองรับการรับ-ส่งข้อมูลแบบ REST จำเป็นต้องกำหนด API Endpoint หรือทางเข้า-ออกของข้อมูลเสียก่อน สำหรับชิ้นงานนี้ มีการทำงานหลัก ๆ กับ API 3 กรณี คือ เมื่อเว็บไซต์ปลายทางต้องการผนวกผู้ใช้กับระบบของเรา, เมื่อเว็บไซต์ปลายทางต้องการตรวจสอบ OTP ที่ผู้ใช้งานโปรแกรมประยุกต์ในสมาร์ตโฟนส่งกลับไปยังเว็บไซต์ และเมื่อเว็บไซต์ปลายทางต้องการตรวจสอบการเข้าใช้งานระบบผ่านทางการแจ้งเตือนของสมาร์ตโฟน จึงมีการกำหนด API Endpoint ดังนี้

- generate-otp/totp/ สร้าง TOTP
- generate-otp/totp/provision_uri/ สร้าง TOTP และ QR Code
- verify-otp/(hotp|totp)/<uuid>/ ตรวจสอบ OTP ผ่าน UUID
- Endpoint อื่น ๆ ของไลบรารี rest-auth ใช้สำหรับการดำเนินการต่าง ๆ ของผู้ใช้

โดยมี Project Directory เป็นลำดับดังรูปที่ 4.1



รูป 4.1 โครงสร้างของ Django Project

จากบรรดาไฟล์ทั้งหมดในชิ้นงานนี้ จะขอยกส่วนทำงานสำคัญ ๆ มาอธิบายดังโปรแกรม 4.2

โปรแกรม 4.2 คลาส PyOTP (บางส่วน)

```
class PyOTP(models.Model):
     uuid = models.UUIDField()
     secret = models.CharField()
     interval = modles.IntegerField()
     name = models.CharField()
     issuer_name = models.CharField()
```

ในส่วนของโมเดลที่ใช้เก็บข้อมูลต่าง ๆ ของ OTP จะใช้คลาส PyOTP ดังโปรแกรม 4.2 สืบทอดมาจากคลาส models.Model ของ Django ทำการกำหนดค่าคุณสมบัติต่าง ๆ ที่ไลบรารี PyOTP จำเป็นต้องใช้ เช่น ค่ากุญแจลับ (secret) ค่าอายุขัยของ OTP (interval) ชื่อองค์กรผู้ออก OTP (issuer_name) ฯลฯ เนื่องจากทางผู้จัดทำมีความคิดเห็นว่าการให้กุญแจลับกับเว็บไซต์ปลายทาง อาจจะมีความไม่ปลอดภัย จึงได้ทำการสร้าง UUID (Universally Unique Identifier) สุ่มขึ้น ใช้เป็นการอ้างถึงตัวผู้ใช้คนหนึ่ง ๆ ที่ใช้งานระบบพิสูจน์ตนขั้นสอง

โปรแกรม 4.3 คลาส OTPMixin

```
class OTPMixin(object):
    def _get_random_base32_string(self):
        return pyotp.random_base32()
    def _create_response(self, otp, instance, otp_type_obj,
data):
        response = {
            'otp_uuid': str(instance.uuid),
            'otp': otp,
        }
        if self.provision_uri is True:
            provisioning_uri =
otp_type_obj.provisioning_uri(**data)
        response = {
            'otp_uuid': str(instance.uuid),
            'provisioning_uri': provisioning_uri,
        }
        return response
    def _generate_totp(self, interval, provision_uri=False,
data={}):
        self.provision_uri = provision_uri
        base32string = self._get_random_base32_string()
        totp = pyotp.TOTP(base32string, interval=interval)
        otp = totp.now()
        return self._create_response(otp, obj, totp, data)
```

จากโปรแกรมที่ 4.3 OTPMixin เป็นคลาสที่ใช้กระทำการสร้าง JSON Object ทั้งหมดของ API สังเกตได้จากฟังก์ชัน _create_response() ที่คืนค่ากลับเป็น JSON Object ที่ประกอบไปด้วยค่า UUID พร้อมกับตัว OTP ที่สร้างมาทันที (เรียกใช้ผ่าน Serializer ซึ่งจะอธิบายต่อไป) หรือจะแนบ URI ที่พร้อมนำไปใช้งานเป็น QR Code OTPMixin มีหน้าที่อื่น ๆ อย่างการสุ่มสายอักขระเข้ารหัสแบบ Base32 เพื่อใช้เป็นกุญแจลับให้กับผู้ใช้คนใหม่

โปรแกรม 4.4 serializers.py

```
class TOTPSerializer(mixins.OTPMixin,
serializers.Serializer):
     timeout = serializers.IntegerField(required=True)
     def create(self, validated_data):
          interval = validated_data.pop('timeout')
          return self._generate_totp(interval,
data=validated_data)

class ProvisionURISerializer(serializers.Serializer):
     name = serializers.CharField()
     issuer_name = serializers.CharField()

class TOTPProvisionURISerializer(TOTPSerializer,
ProvisionURISerializer):
     def create(self, validated_data):
          interval = validated_data.pop('timeout')
          return self._generate_totp(interval,
provision_uri=True, data=validated_data)

class VerifyOTPSerializer(serializers.Serializer):
     otp = serializers.CharField(required=True)
     def verify_otp(self, otp, obj, otp_type):
          if otp_type == 'totp' and obj.interval:
               totp = pyotp.TOTP(obj.secret,
interval=obj.interval)
               return totp.verify(otp)
          return False
```

จากโปรแกรมที่ 4.4 Serializer มีหน้าที่หลัก ๆ ในการจัดรูปแบบหรือเตรียม JSON Object ให้กับระบบ เพื่อให้สะดวกต่อการติดต่อสื่อสารกับระบบอื่น หรือแม้แต่จัดเก็บลงฐานข้อมูลเอง ณ ไฟล์นี้เองที่เกิดกระบวนการตรวจสอบ OTP ขึ้น โดยการสร้าง TOTP Object ใหม่โดยใช้กุญแจลับเดียวกัน หากรหัสตรงกัน ย่อมเป็นตัวตนของผู้ใช้เอง

โปรแกรม 4.5 คลาส PyOTPViewset (บางส่วน)

```
class PyOTPViewset(viewsets.GenericViewset):
     queryset = models.PyOTP.objects.all()
     lookup_field = 'uuid'
     otp_type = None
     def get_serializer_class(self):
          if self.action == 'generate_totp':
               return serializers.TOTPSerializer
          elif self.action ==
'generate_totp_provision_uri':
               return
serializers.TOTPProvisionURISerializer
          elif self.action == 'verify_otp':
               return serializers.VerifyOTPSerializer
          return serializers.NoneSerializer
     def _validate(self, serializer, data):
          serializer_instance = serializer(data=data)

     serializer_instance.is_valid(raise_exception=True)
          return serializer_instance.save()
     def generate_totp(self, request):
          serializer = self.get_serializer_class()
          serializer = self._validate(serializer,
request.data)
          return Response(serializer,
status=status.HTTP_201_CREATED)
     def verify_otp(self, request, otp_type, uuid):
          obj = self.get_object()
          serializer = self.get_serializer_class()
          serializer = serializer(data=request.data)
          serializer.is_valid(raise_exception=True)
          valid_otp =
serializer.verify_otp(serializer.data.get('otp'), obj,
otp_type)
          if not valid_otp:
               return
Response(status=status.HTTP_400_BAD_REQUEST)
          return Response(status=status.HTTP_200_OK)
```

จากโปรแกรมที่ 4.5 Viewset ของ Django REST Framework มีลักษณะคล้าย View ของ Django แต่มอบความรวดเร็วในการผูกกับ URI มากกว่า View ดังจะได้เห็นถัดไป หน้าที่ของ PyOTPViewset คือเรียก Object ทุกตัวในโมเดล PyOTP และตรวจสอบรูปแบบข้อมูลกับ Serializer ที่เหมาะสม

ณ ไฟล์นี้ การตรวจสอบ OTP จะบ่งชี้ถึง HTTP Status โดยขึ้นกับความถูกต้องของ OTP กล่าวคือ หากผู้ใช้กรอก OTP ผิด API จะตอบด้วย HTTP 400 หากถูกต้อง จะตอบด้วย HTTP 200

โปรแกรม 4.6 routers.py (ไม่รวม urls.py ณ Root Directory)

```
UUID_REGEX    =    '[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-
f]{4}-[0-9a-f]{12}'
OTP_TYPE_REGEX = '(hotp|totp)'
verify_otp = views.PyOTPViewset.as_view({'post':
'verify_otp'})
generate_totp      =      views.PyOTPViewset.as_view({'post':
'generate_totp'})
...
urlpatterns = [
     path('generate-otp/totp/', generate_totp,),
     ...
     re_path(r'^verify-
otp/(?P<otp_type>(hotp|totp))/(?P<uuid>{uuid})/$'.format(ot
p_type=OTP_TYPE_REGEX, uuid=UUID_REGEX)
]
```

เมื่อทำการเตรียม Viewset จนครบถ้วนแล้ว จึงเริ่มทำการผูก URI ไว้กับ Viewset ต่าง ๆ ด้วย Viewset.as_view() โดยจากโปรแกรม 4.6 ได้ตั้งค่าให้เรียกใช้งาน Viewset ได้เฉพาะ POST Method เท่านั้น

จากทั้งหมดข้างต้น จะได้ REST API ที่สามารถสร้าง OTP ดังโปรแกรม 4.6 ที่สามารถผนวก Provision URI เพื่อนำไปสร้าง QR Code แลกเปลี่ยนกุญแจลับกับผู้ใช้สมาร์ทโฟนได้

4.1.2 Android Application

การติดตั้งและเตรียมสภาพแวดล้อมการพัฒนาใน Android Studio มีความคล้ายคลึงกับกระบวนการของ Django ที่ได้กล่าวไปแล้ว โดยเริ่มจากการสร้าง Manifest File เพื่อระบุถึงคุณลักษณะต่าง ๆ ที่โปรแกรมประยุกต์ต้องการจะใช้ขณะทำงาน และตั้งค่าใน build.gradle เพื่อระบุถึงเครื่องมือเสริมที่จะใช้ในการพัฒนา ดังโปรแกรม 4.7

โปรแกรม 4.7 AndroidManifest.xml (บางส่วน)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android=http://schemas.android.com/apk/res/android
package="com.jettolo.otter">
     <uses-permission
android:name="android.permission.INTERNET" />
     ...
     <application
          android:hardwareAccelerated="true"
          android:label="Otter"
          <activity
               android:name=".MainPage"
               android:screenOrientation="portrait" />
          ...
     </application>
</manifest>
```

โปรแกรม 4.8 App-level build.gradle (บางส่วน)

```
apply plugin: 'com.android.application'

android {
     compileSdkVersion 27
     defaultConfig {
          ...
          minSdkVersion 19
          targetSdkVersion 27
     }
}
dependencies {
     ...
     implementation 'com.google.firebase:firebase-
messaging:15.0.0'
     implementation 'org.jboss.aerogear:aerogear-otp-
java:1.0.0'
     implementation 'org.journeyapps:zxing-android-
embedded:3.6.0'
}
```

จากโปรแกรม 4.8 build.gradle ทราบว่าได้ใช้ไลบรารีเสริมใดบ้าง เช่น Firebase Cloud Messaging, Aerogear OTP Java หรือ ZXing Barcode Reader เป็นต้น

บทบาทของโปรแกรมประยุกต์แอนดรอยด์ คือการเป็นอุปกรณ์ติดตัวของผู้ใช้เว็บไซต์ปลายทางที่ต้องการความปลอดภัยเพิ่มเติม จึงสมัครใช้บริการพิสูจน์ตนขั้นสอง ซึ่งเริ่มสมัคร/ใช้งานผ่านโปรแกรมประยุกต์ได้ทันที หน้าต่างหลักเป็นศูนย์กลางของ OTP และการแจ้งเตือนแบบพุชสามารถสลับไปมาได้ตามที่ผู้ใช้พึงพอใจ

ในสภาพแวดล้อมพัฒนาดังกล่าว มีโครงสร้างของ Project Directory เป็นดังรูปที่ 4.2

รูป 4.2 โครงสร้างของ Android Development Environment

จะขอยกส่วนของโปรแกรมในการทำงานหลัก ๆ มาอธิบายอย่างเป็นขั้นตอนต่อไป

โปรแกรม 4.9 Activity CodeManually

```
public class CodeManually extends AppCompatActivity {
     private static final int COUNTDOWN_DURATION = 30000;
     private static final int COUNTDOWN_STEP = 100;
     private TextView totpDisplay;
     private Totp totp;
     @Override
     protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_code_manually);
          totpDisplay = (TextView) findViewById(R.id.tOTP);
          FloatingActionButton withQRCode =
(FloatingActionButton)
findViewById(R.id.floatingActionButton);
          withQRCode.setOnClickListener(new
View.OnClickListener() {
               @Override
               public void onClick(View view) {
                    Intent intent = new
Intent(CodeManually.this, QRCodeActivity.class);
                    startActivityForResult(intent, 1);
               }
          });
     }
     @Override
     protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
          super.onActivityResult(requestCode, resultCode,
data);
          if(requestCode == 1) {
               if(resultCode == RESULT_OK) {
                    String otpauth =
data.getStringExtra("otpauth");
                    Uri otpUri = Uri.parse(otpauth);
                    String name =
otpUri.getQueryParameter("issuer");
                    String secret =
otpUri.getQueryParameter("secret");
                    totp = new Totp(secret);
                    updateOTP();
               }
          }
```

โปรแกรม 4.9 Activity CodeManually (ต่อ)

```
            new CountDownTimer(COUNTDOWN_DURATION,
COUNTDOWN_STEP) {
                @Override
                public void onTick(long millisUntilFinished)
{}
                @Override
                public void onFinish() {
                    updateOTP();
                    this.start();
                }
            }.start();
}
private void updateOTP() {
    totpDisplay.setText(totp.now());
}
public void switchMode(View view) {
    Intent intent = new Intent(this, MainPage.class);
    startActivity(intent);
}
```

ในโปรแกรม 4.9 Activity นี้เป็นหน้าต่างที่มอบ TOTP ให้ผู้ใช้ตลอดเวลาจนกว่าจะทำการปิดโปรแกรมประยุกต์นี้ โดยการได้มาของ OTP เกิดจากการสแกน QR Code ที่เว็บไซต์ปลายทางแสดงผลให้ผู้ใช้ ซึ่งการดำเนินการสแกน QR Code นี้อยู่ในอีก Activity หนึ่ง ผู้ใช้สามารถใช้ปุ่ม Floating Action Button (FAB) ในการเข้าถึง Activity ดังกล่าว

กลไกการทำงานหลัก ๆ คือ เมื่อได้รับ Intent ขากลับจาก QRCodeActivity ซึ่งแนบข้อมูล URI มาด้วยแล้ว ให้ทำการแปลงและแยกแยะนำค่าข้อมูลต่าง ๆ มาประกอบในหน้าต่างของ Activity นี้ โดยข้อมูลที่สำคัญคือ secret ซึ่งสามารถนำไปสร้าง TOTP ได้ทันที กลไกต่อมา คือการปรับปรุง TOTP ให้ทันสมัยทุกครั้งที่หมดอายุขัย (ค่าโดยปริยาย คือ 30 วินาที) และแสดงผลที่หน้าจอ

อย่างที่ได้กล่าวไปข้างต้น โปรแกรมประยุกต์นี้สามารถสลับโหมดการใช้งานได้ ด้วยการแตะปุ่ม RECEIVE REQUEST เพื่อเรียกใช้ switchMode() ในการกลับไปยัง Activity MainPage ในการสลับไปใช้วิธีพิสูจน์ตนผ่านการแจ้งเตือนแบบพุช

โปรแกรม 4.10 QRCodeActivity

```
public class QRCodeActivity extends AppConpatActivity {
     private static final String TAG =
QRCodeActivity.class.getName();
     @Override
     protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          scanBarcode();
     }
     private void scanBarcode() {
          new IntentIntegrator(this).initiateScan();
     }
     @Override
     protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
          super.onActivityResult(requestCode, resultCode,
data);
          IntentResult result =
IntentIntegrator.parseActivityResult(requestCode,
resultCode, data);
          if(result != null) {
               if(result.getContents() != null) {
                    Log.d(TAG, result.getContents());
                    Intent intent = new Intent();
                    intent.putExtra("otpauth",
result.getContents());
                    setResult(RESULT_OK, intent);
                    finish();
               } else {
                    Intent intent = new Intent();
                    setResult(RESULT_CANCELED, intent);
                    finish();
               }
          } else {
               super.onActivityResult(requestCode,
resultCode, data);
               showAlertDialog();
          }
}
```

ในโปรแกรม 4.10 Activity นี้ มีเพียงหน้าที่เดียว คือ สแกน QR Code ที่เว็บไซต์ปลายทาง แสดงผลให้ผู้ใช้ แล้วนำข้อมูลใน QR Code URI ส่งกลับไปยัง Activity CodeManually เพื่อแสดงผลต่อไป

4.1.3 Mocking Website

ในส่วนของเว็บไซต์ทดสอบ เพื่อให้ง่ายต่อการตรวจสอบพฤติกรรม ความสามารถต่าง ๆ ของระบบ จึงใช้ภาษา PHP ในการสร้างเว็บไซต์ทดสอบเรียกใช้งาน REST API ดังกล่าว

4.2 ออกแบบและพัฒนาส่วน User Interface

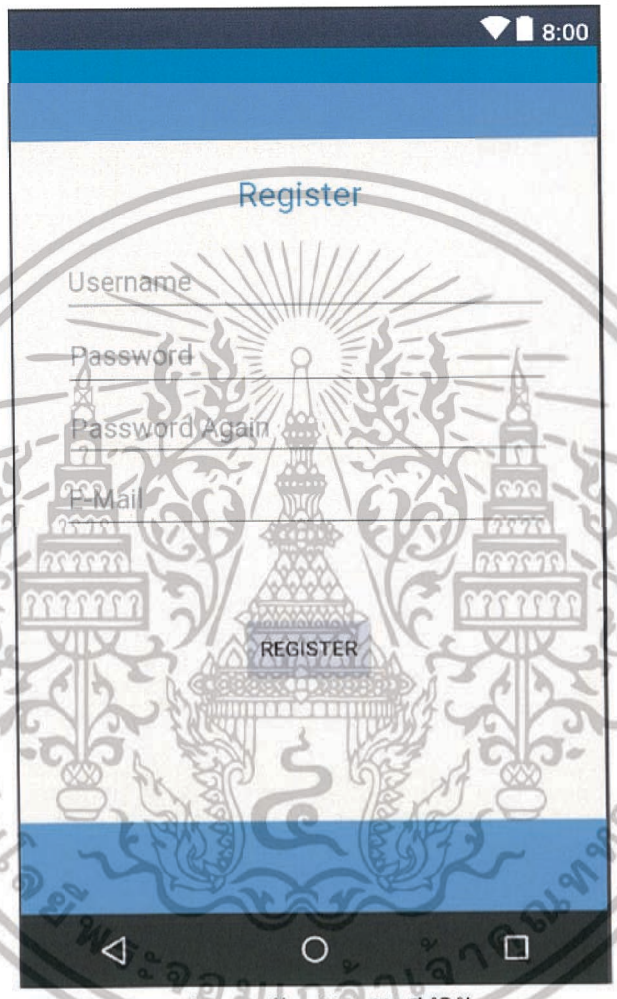
4.2.1 หน้าเข้าสู่ระบบของโปรแกรมประยุกต์

รูป 4.3 หน้าเข้าสู่ระบบ

เมื่อเข้าสู่โปรแกรมประยุกต์ ดังรูปที่ 4.3 จะแสดงให้เห็นโลโก้ของโปรแกรมประยุกต์นี้ จากนั้นจะเข้าสู่หน้าเข้าสู่ระบบ ถ้าผู้ใช้งานมีบัญชีผู้ใช้อยู่แล้ว ก็สามารถทำการเข้าสู่ระบบ เพื่อใช้งาน

โปรแกรมประยุกต์ได้ ถ้าผู้ใช้ยังไม่มีบัญชีของโปรแกรมประยุกต์ ก็สามารถทำการลงทะเบียนเข้าใช้งานโปรแกรมประยุกต์ได้โดยแตะที่คำว่า Register

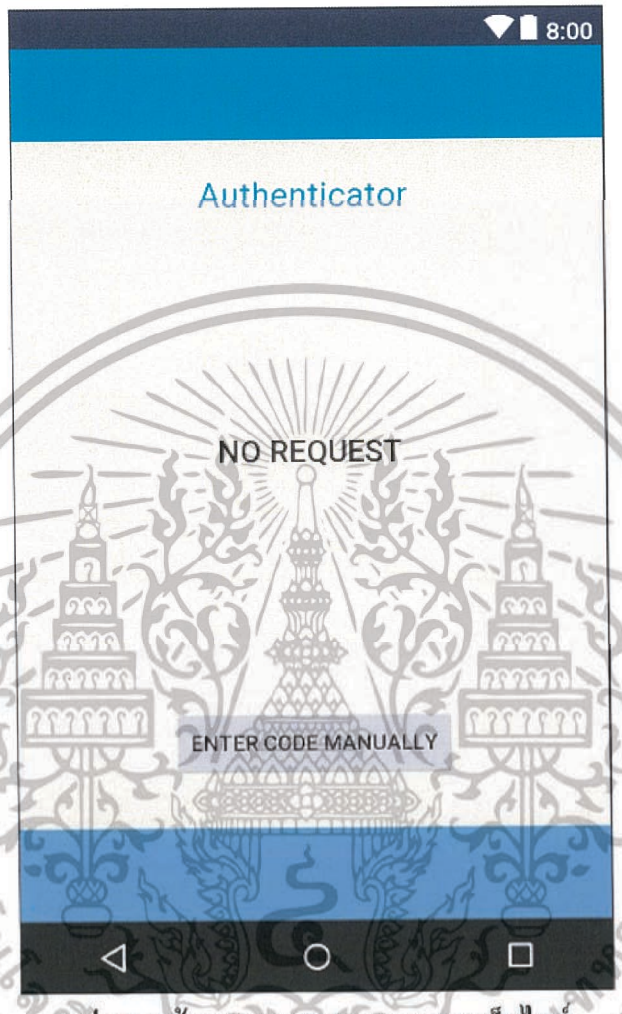
4.2.2 หน้าของการลงทะเบียนเข้าใช้งานโปรแกรมประยุกต์



รูป 4.4 หน้าสมัครบัญชีผู้ใช้

เมื่อทำการแตะปุ่ม Register แล้ว จะเข้าสู่หน้าของการลงทะเบียนเข้าใช้งานโปรแกรมประยุกต์ ดังรูปที่ 4.4 โดยผู้ใช้งานต้องทำการกรอกบัญชีผู้ใช้ อีเมล และรหัสผ่านที่ต้องการใช้งานในระบบ

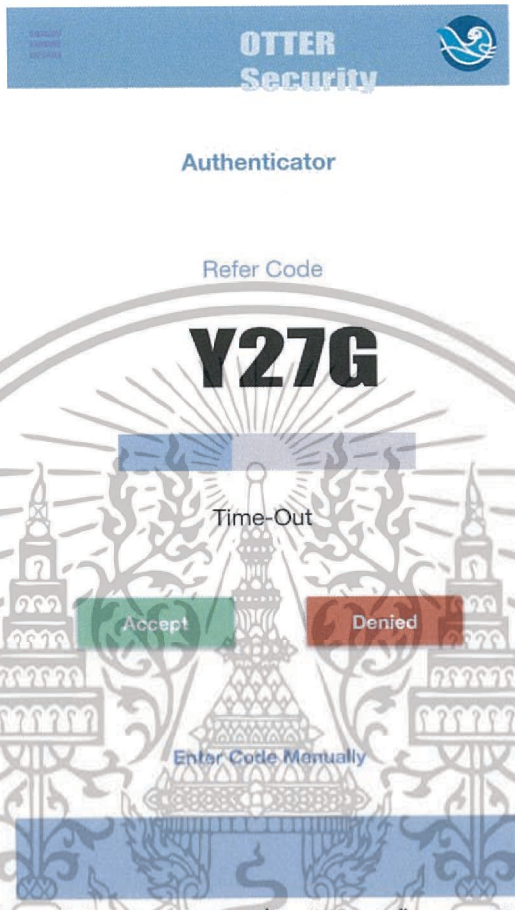
4.2.3 หน้าของการรอ Request จากเว็บไซต์



รูป 4.5 หน้าของการรอ Request จากเว็บไซต์

ถ้ามีบัญชีผู้ใช้งานแล้วหรือผู้ใช้พึ่งทำการลงทะเบียนเข้าใช้งาน โปรแกรมประยุกต์ เมื่อเสร็จสิ้นการเข้าสู่ระบบจะเข้าสู่หน้านี้ของโปรแกรมประยุกต์ ดังรูปที่ 4.5 ทำการรอ Request ที่ได้จากการเรียกใช้ API ของระบบการยืนยันตัวตนของเว็บไซต์

4.2.4 หน้าของโปรแกรมประยุกต์โดยใช้การเลือกยอมรับหรือปฏิเสธ



รูป 4.6 หน้าของโปรแกรมประยุกต์โดยใช้การเลือกยอมรับหรือปฏิเสธ

ในกรณีที่ทำการเข้าสู่ระบบเว็บไซต์ปลายทางและทางผู้ใช้ประสงค์จะรับการแจ้งเตือนแบบพุช Firebase จะทำการส่งการแจ้งเตือนไปยังสมาร์ทโฟนของผู้ใช้ เมื่อผู้ใช้ได้รับการแจ้งเตือนแบบพุช สามารถแตะการแจ้งเตือนนั้นเพื่อเข้าสู่หน้าดังกล่าวในโปรแกรมประยุกต์ได้ทันที และจะทำการแสดงรหัสอ้างอิงที่ควรจะเป็นชุดเดียวกับบนเว็บไซต์ดังรูปที่ 4.6 เมื่อทำการกด Accept ผู้ใช้งานจะสามารถเข้าสู่ระบบของเว็บไซต์ได้ หรือถ้าไม่ใช่รหัสอ้างอิงเดียวกัน เมื่อทำการกด Denied ผู้ใช้งานจะไม่สามารถเข้าสู่ระบบของเว็บไซต์ได้ หลังจากทำการตัดสินใจเลือกแล้ว โปรแกรมประยุกต์จะกลับสู่หน้ารอ Request

สามารถเลือกสลับวิธีการยืนยันตัวตนเป็นแบบใส่รหัสยืนยันตัวตนได้ด้วยตนเองตลอดเวลา โดยทำการกด Enter Code Manually

4.2.5 หน้าของโปรแกรมประยุกต์แบบใส่รหัสยืนยันตัวตนด้วยตนเอง

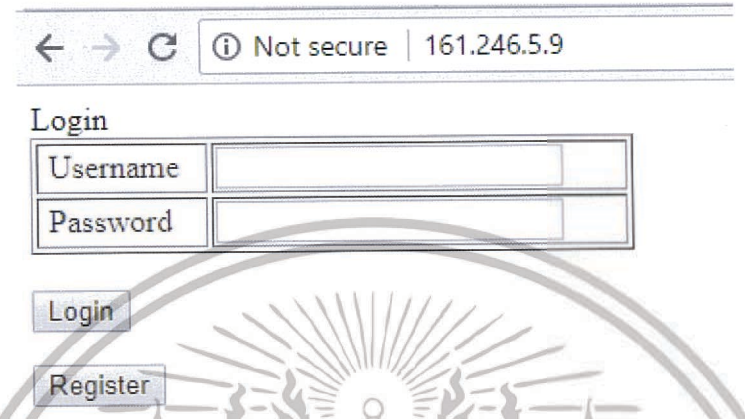
รูป 4.7 หน้าของโปรแกรมประยุกต์แบบใส่รหัสยืนยันตัวตนด้วยตนเอง

โปรแกรมประยุกต์นี้ไม่จำเป็นต้องออนไลน์ตลอดเวลา ณ หน้านี้มีการแสดงรหัสยืนยันตัวที่ได้คำนวณ ณ เวลานั้น ดังรูปที่ 4.7 รหัสผ่านที่คำนวณได้นี้จะมีระยะเวลาในการใช้งาน 30 วินาที ก่อนจะทำการคำนวณรหัสยืนยันตัวตนใหม่ เมื่อทำการใส่รหัสยืนยันตัวตนแล้วมีค่าตรงกับรหัสที่เว็บไซต์คำนวณ ผู้ใช้งานก็จะสามารถเข้าสู่ระบบได้

สามารถเลือกสลับวิธีการยืนยันตัวตนเป็นแบบการเลือกยอมรับหรือปฏิเสธ โดยทำการกด Use One – button

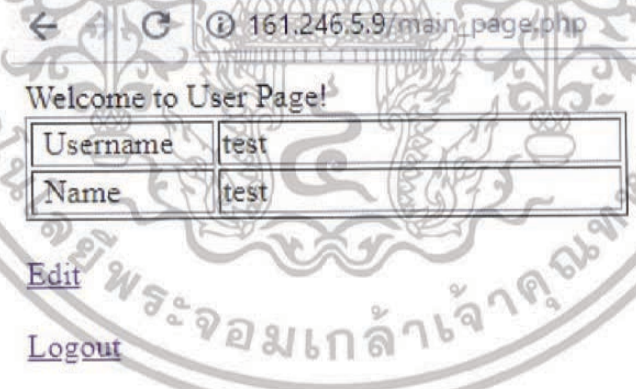
4.3 การทำงานของระบบการยืนยันตัวตนแบบปกติ

ผู้ใช้งานทำการใส่ชื่อผู้ใช้และรหัสผ่านในหน้าเข้าสู่ระบบ ดังรูปที่ 4.8



รูป 4.8 หน้าของเว็บไซต์ผู้ให้บริการที่มีระบบยืนยันตัวตน

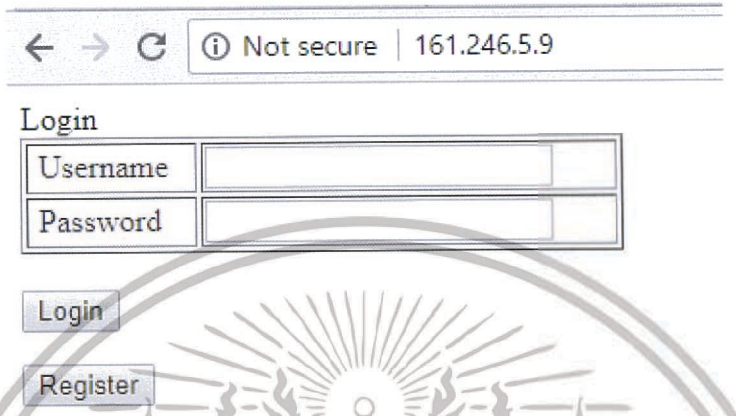
ระบบทำการส่งข้อมูลไปตรวจสอบกับดาต้าเบส (MySQL) เพื่อทำการยืนยันตัวตนผู้ใช้ ดังรูปที่ 4.9



รูป 4.9 หน้าของเว็บไซต์ผู้ให้บริการเมื่อยืนยันตัวตนถูกต้อง

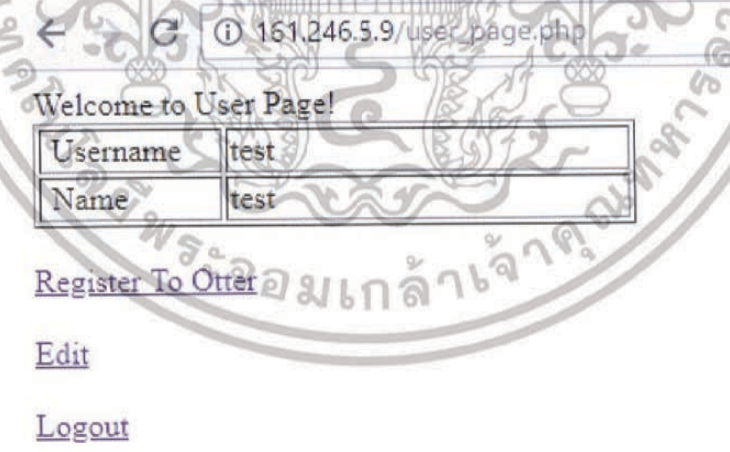
4.4 การทำงานของระบบการยืนยันตัวตนแบบยืนยันตัวตนขั้นที่สอง

หน้าตาเว็บไซต์จะเป็นเหมือนระบบการยืนยันตัวตนแบบปกติ ดังรูปที่ 4.10



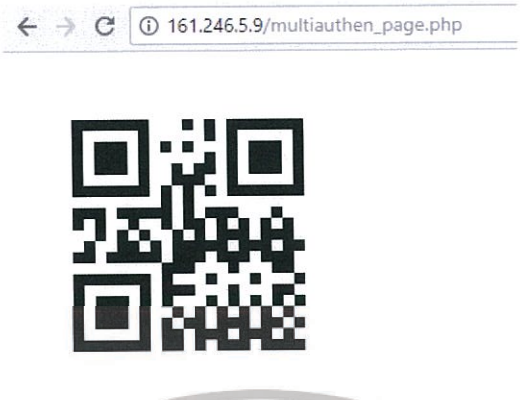
รูป 4.10 หน้าของเว็บไซต์ผู้ให้บริการที่มีระบบยืนยันตัวตน

ระบบทำการส่งข้อมูลไปตรวจสอบกับดาต้าเบส เพื่อทำการยืนยันตัวตนผู้ใช้ดังรูปที่ 4.11 จากนั้นให้ผู้ใช้งานกดที่ Register To Otter เพื่อสมัครใช้การยืนยันตัวตนขั้นที่สอง



รูป 4.11 หน้าของเว็บไซต์ให้ผู้ใช้งานสมัครการยืนยันตัวตนขั้นที่สอง

หลังจากที่ได้ทำการกดที่ Register To Otter ก็จะมี QR Code ให้ผู้ใช้งานทำการสแกน ดังรูปที่ 4.12

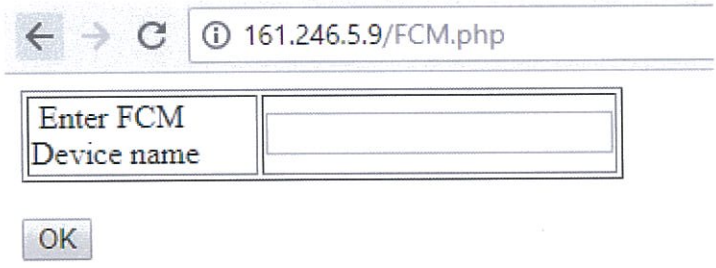


รูป 4.12 หน้าของเว็บไซต์ให้ผู้ใช้งานทำการสแกน QR Code

ผู้ใช้งานทำการสแกน QR Code แล้ว ก็จะขึ้นรหัส OTP ในโทรศัพท์ ดังรูปที่ 4.13

รูป 4.13 หน้าของโปรแกรมประยุกต์แบบใส่รหัสยืนยันตัวตนด้วยตนเอง

หลังผู้ใช้งานทำการใส่รหัสผ่าน OTP แล้ว ก็จะขึ้นหน้าต่างที่ให้ผู้ใช้งานทำการใส่ ชื่อของผู้ใช้งานที่สมัครในโปรแกรมประยุกต์ ดังรูปที่ 4.14

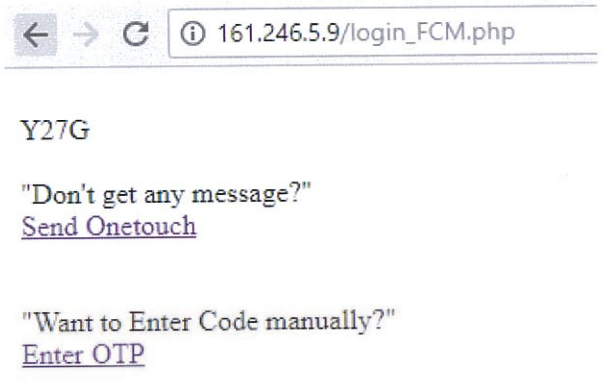


รูป 4.14 หน้าของเว็บไซต์ให้ผู้ใช้งานทำการใส่ชื่อที่สมัครในโปรแกรมประยุกต์

หลังจากนั้นก็จะเข้าสู่หน้าผู้ใช้งานหลัก ดังรูปที่ 4.15

รูป 4.15 หน้าของเว็บไซต์ผู้ให้บริการเมื่อยืนยันตัวตนถูกต้อง

เมื่อผู้ใช้งานเข้าสู่ระบบในครั้งต่อไป ก็จะเข้าสู่หน้าที่มีรหัส 4 ตัวขึ้นที่หน้าเว็บไซต์และ ส่งรหัส 4 ตัวไปที่โปรแกรมประยุกต์ให้ผู้ใช้งานตรวจสอบรหัส และเลือกว่ายอมรับหรือปฏิเสธ ดังรูปที่ 4.16 และรูปที่ 4.17 เมื่อกดยอมรับก็จะสามารถเข้าสู่ระบบได้



รูป 4.16 หน้าของเว็บไซต์ผู้ให้บริการที่มีรหัส 4 หลัก

รูป 4.17 หน้าของโปรแกรมประยุกต์โดยใช้การเลือกยอมรับหรือปฏิเสธ

บทที่ 5
สรุป

5.1 สรุปผลการดำเนินงาน

1) ได้พัฒนาการออกแบบระบบที่สามารถใช้งานได้พอสังเขป
2) ได้จัดตั้งระบบการพิสูจน์ทราบตัวตนแบบหลายปัจจัยในรูปแบบของ API ให้บริการเว็บ
3) ได้โปรแกรมประยุกต์ในระบบปฏิบัติการแอนดรอยด์ที่สามารถใช้งานร่วมกับระบบการพิสูจน์ตน

5.2 อุปสรรคในการดำเนินงาน

1) แหล่งข้อมูลในการศึกษาค้นคว้าเกี่ยวกับการสร้างโปรแกรมประยุกต์สำหรับการพิสูจน์ทราบตัวตนแบบหลายปัจจัยโดยเริ่มต้นสร้างใหม่ทั้งหมดนั้นมีน้อยมาก
2) มีคำเตือนจากหลาย ๆ แหล่งข้อมูลที่สืบค้นว่าไม่ควรดำเนินการสร้างใหม่ทั้งหมดด้วยตนเอง เนื่องจากอาจเกิดความเสียหายต่อผู้ดูแลเว็บไซต์ที่จะนำระบบนี้ไปบูรณาการได้
3) มีความรู้ความเข้าใจในการออกแบบระบบน้อย
4) เนื่องจากตัวระบบนั้นเกี่ยวข้องกับเรื่องเวลามาก เมื่อเกิดปัญหากับโปรโตคอล NTP ขึ้น ทำให้เสียเวลาในการพัฒนางานโดยใช่เหตุ
5) ไม่มีความคิดสร้างสรรค์ในการออกแบบโปรแกรมประยุกต์ให้โดดเด่นพอในตลาด
6) มีความเข้าใจไม่ตรงกันในคณะผู้จัดทำเกี่ยวกับกลไกการดำเนินระบบ

5.3 แนวทางการต่อยอดงาน

1) เพิ่มปัจจัยในการพิสูจน์ตนอีกหนึ่งปัจจัย คือ Inherence factor เช่น ชีวมาตร (ลายนิ้วมือ, เสียง)
2) สร้างระบบกู้คืนในกรณีที่อุปกรณ์ที่ใช้พิสูจน์ตนสูญหายด้วยวิธีที่ดีกว่า
3) เพิ่มความปลอดภัยในการรับ-ส่งข้อมูลผ่าน REST API ด้วยการเข้ารหัสลับข้อความต่าง ๆ

บรรณานุกรม

จตุชัย แพงจันทร์. 2558. **Master in Security 3rd Edition.** นนทบุรี : ไอดีซีฯ

Simon Tabor. 2558. **Building Two-Factor Authentication**. [Online].
Available: https://engineering.gosquared.com/building-two-factor-authentication

Tom Leek. 2556. **How to write solid TOTP implement?**. [Online].
Available: https://security.stackexchange.com/questions/47979/how-to-write-a-rock-solid-totp-implementation

กนกพล เมืองรักษ์. 2555. **ทำความรู้จัก Radius Server**. [Online].
Available: https://arit.rmutsv.ac.th/th/blogs/50-ตอนที่1-ทำความรู้จัก-radius-server-744

Wutipong Wongsakudej. 2557. **Authentication vs Authorization ต่างกันยังไง**. [Online].
Available: http://blog.playground-soft.com/2014/06/Authentication-vs-Authorization-ต่างกันยังไง

Archai. 2560. **ทำความรู้จักกับ Android Studio**. [Online]
Available: http://arctech.in.th/articles/125

Krist Wongsuphasawat. 2560. **API คืออะไร? อธิบายแบบคนไม่เขียนโปรแกรมรู้เรื่องได้ไหม**. [Online].
Available: https://medium.com/skooldio/api-คืออะไร-264ee4186f2c