



รายงานสหกิจศึกษาฉบับสมบูรณ์

แอปพลิเคชันจัดการวันลาหยุด

Leave Request application for organization



อธิพร จันทะไชย

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559



T148603

รายงานสหกิจฉบับสมบูรณ์

แอปพลิเคชันจัดการวันลาหยุด

Leave Request application for organization

อธิพร จันทะไชย

สาขา.....
เลขทะเบียน 148603
วันเดือนปี ๓.6 ๗๕, 2560

.b. 12872696
.i.

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจ แอปพลิเคชันจัดการวันลาหยุด

ชื่อ – สกุล นักศึกษา อธิพร จันทะไชย

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมสารสนเทศ

ชื่อ – สกุล อาจารย์นิเทศ ดร.พิกุลแก้ว ตังติสานนท์

ชื่อ – สกุล ผู้นิเทศงาน นาย กษิต์เดช ยนตรดิษฐถาวร

ชื่อสถานประกอบการ บริษัท เอ็กซ์เซลแบงคอก จำกัด

บทคัดย่อ

เมื่อพนักงานต้องการจะลางานแต่ละครั้งจำเป็นต้องเขียนจดหมายลา และทำการส่งมอบให้กับหัวหน้างานหรือผู้ที่มีอำนาจอนุญาตให้ลางานได้ เพื่อให้ผู้ที่มีอำนาจลงลายมืออนุญาตให้ลางานได้ โดยบางองค์กรต้องมีการอนุญาตจากผู้มีอำนาจมากกว่าหนึ่งคน ยิ่งทำให้การลานั้นยุ่งยากและล่าช้าไปยิ่งกว่าเดิม ยิ่งไปกว่านั้น พนักงานยังไม่สามารถทราบจำนวนวันลาที่เหลือของตน หรือไม่สามารถทราบได้ว่าใบลาที่ส่งไปนั้นได้รับอนุญาตให้ลาหรือไม่จนกว่าผู้ที่มีอำนาจอนุญาตจะแจ้งกลับมา การนำแอปพลิเคชันจัดการวันลาเข้ามาใช้งานจะช่วยจัดการการขอลาที่มีการเก็บข้อมูลที่เกี่ยวข้องกับการลาของพนักงานในองค์กรอย่างเป็นระบบระเบียบมากขึ้น และพนักงานในองค์กรยังสามารถตรวจสอบประวัติการลาหรือจำนวนวันลาที่เหลือได้จากภายในแอปพลิเคชันเพื่อวางแผนการลาได้ รวมถึงสามารถส่งคำขอลาและตรวจสอบสถานะอนุมัติคำขอลาได้อีกด้วย แอปพลิเคชันจัดการวันลานี้จะมีส่วนช่วยให้การลาสามารถดำเนินการได้ง่ายและรวดเร็วยิ่งขึ้น เพราะบุคลากรสามารถส่งคำขอลาได้ทุกเมื่อที่ต้องการ นอกจากนี้ยังมีส่วนช่วยลดการใช้กระดาษในการส่งจดหมายลาด้วย ทำให้ลดการใช้ทรัพยากรสิ่งแวดล้อม

คำสำคัญ : ระบบลาหยุด, ระบบบริหารจัดการแผนบุคคล, ระบบจัดการวันลาอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Co-Operative Title: Leave Request application for organization

Student Intern Name: Athiporn Chantachai

Faculty: Engineering

Department: Information Engineering

Adviser Name: Dr. Pikulkaew Tangtisanon

Mentor Name: Kasidesh Yontaradidthaworn

Company: Excel Bangkok Co., Ltd.

ABSTRACT

Each time when employees require a day off, they have to write a leave of absence letter and submit it to their supervisor or to a person who is in charge of approving the employees' absence in order that the absence can be approved. In some organizations, the letter has to be improved by more than one person in charge causing the complication and the delay. Moreover, the employees are not able to know the number of their remaining absence days or whether the letter sent is approved or not until the person in the charge gives them feedback. Therefore, adopting the application of managing absence days will be useful in requesting a day off for the employees and all of the information related will be collected more systematically. Additionally, the employees will be able to check their absence history and the number of the remaining days from the application so that they can plan for their future absence. This application includes the help of sending requests and checking status of whether requests are approved or not. It will also help the employees to be able to request for a day off more easily and faster because they can send requests anytime they want. Besides, it can help reducing the use of paper in making a leave of absence letter. That is a means of saving environmental resources as well.

Keywords : Leave Request System, Human Resource Management, Automatic Leave Checking System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้จะสำเร็จลุล่วงไปได้ด้วยดีก็ด้วยบุคคลเหล่านี้ ขอขอบคุณ ดร.พิกุลแก้ว ตั้งติสานนท์ ที่คอยให้คำปรึกษาในการดำเนินงานวิจัยและประสานงานทั้งกับทางบริษัท เอ็กซ์เซลล์แองค็อก จำกัด และทางคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อีกทั้งยังช่วยให้คำแนะนำในการแก้ปัญหาต่างๆที่เกิดขึ้นระหว่างการดำเนินงานวิจัยอีกด้วย ขอขอบคุณ นาย กษิทธิ์เดช ยนตรดิษฐถาวร ที่ได้ให้โอกาสได้ดำเนินงานวิจัย ช่วยส่งสอนสิ่งต่างๆที่คิดว่าควรสอน ช่วยติ-ชมผลงานวิจัย ตลอดจนการดำเนินงาน รวมถึงช่วยอำนวยความสะดวกต่างๆในการดำเนินงานวิจัยภายในบริษัท เอ็กซ์เซลล์แองค็อก จำกัด ขอขอบคุณ นาย ณัฐชนน พัฒนเดชะ, นาย บัณฑิต หิรัญกาญจน์ และ นาย อภิสิทธิ์ ไวยโกคา ที่ได้ให้การช่วยเหลือในหลายๆขั้นตอนของการดำเนินงานวิจัย ตั้งแต่กระบวนการคิดพื้นฐาน, สิ่งที่ต้องรู้ในการพัฒนาแอปพลิเคชัน ช่วยให้กำลังใจ ช่วยตักเตือนในเรื่องที่ไม่ถูกไม่ควร ขอขอบคุณ รุ่งพณีในบริษัท เอ็กซ์เซลล์แองค็อก จำกัด ทุกคนที่คอยแนะนำวิธีการผ่อนคลายจากความเครียดระหว่างการดำเนินงานวิจัย ขอขอบคุณที่ทำให้การดำเนินวิจัยครั้งนี้ไม่ตึงเครียดจนเกินไป ขอขอบคุณที่ทำให้ทราบว่าการทำงานก็มีส่วนที่สนุกสนานเช่นกัน ขอขอบคุณเพื่อนๆในสาขาวิศวกรรมสารสนเทศและต่างสาขา พี่ๆน้องๆทั้งร่วมสาขาและต่างสาขาที่คอยให้กำลังใจให้คำปรึกษา แนะนำ และคอยรับฟังปัญหาต่างๆ นับตั้งแต่วันแรกที่รู้จักกันจนผ่านมาถึงวันนี้ ขอขอบคุณจากใจจริง

สุดท้ายนี้ขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียนตลอดจนคอยช่วยเหลือและให้กำลังใจเสมอมาตั้งแต่วันแรกที่เริ่มการศึกษาจนกระทั่งวันสำเร็จการศึกษา

อธิพร จันทะไชย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญ (ต่อ).....	V
สารบัญ (ต่อ).....	VI
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
สารบัญรูป (ต่อ).....	IX
สารบัญรูป (ต่อ).....	X
สารบัญรูป (ต่อ).....	XI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 วิธีการดำเนินการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 ระบบปฏิบัติการ Android.....	5
2.2 ภาษา Java.....	6
2.3 Android SDK.....	7
2.4 Java JDK.....	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5 Android Support Library.....	8
2.6 แนวความคิดในการพัฒนาโปรแกรม แบบ Agile	9
2.7 รูปแบบในการพัฒนาโปรแกรมแบบ Scrum.....	12
2.7.1 ทีมบุคลากรในการพัฒนาโปรแกรม หรือ Role.....	12
2.7.2 วิธีการทำงาน หรือ Process.....	13
2.7.3 การประเมินผลและติดตามผลงาน หรือ Demonstration and Evaluation.....	16
2.8 แบบร่างโปรแกรม หรือ Wireframe.....	17
2.9 แอปพลิเคชันตัวทดลอง หรือ Prototype (Mockup).....	20
2.10 Git.....	22
2.11 ภาษา XML.....	27
2.12 สถาปัตยกรรมซอฟต์แวร์แบบ Model-View-Controller หรือ MVC Pattern.....	28
2.13 Third party library สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android.....	30
2.14 การออกแบบ User Experience และ User Interface (UX/UI).....	32
บทที่ 3 วิธีการดำเนินงานวิจัย.....	34
3.1 เก็บข้อมูล.....	34
3.2 ออกแบบโครงสร้างของระบบ.....	36
3.3 ออกแบบระบบฐานข้อมูล.....	49
3.4 แผนภาพลำดับเหตุการณ์ (Sequence Diagram).....	56
3.5 การออกแบบส่วนติดต่อผู้ใช้.....	60
3.6 การออกแบบแอปพลิเคชันด้วยโครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC	61
บทที่ 4 ผลการวิจัย	66
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
5.1 สรุปผลการวิจัย.....	78
5.2 ปัญหาที่พบบ่อยระหว่างดำเนินงานวิจัย	78
5.3 แนวทางในการแก้ไขปัญหา	78
5.4 ข้อเสนอแนะ	79
เอกสารอ้างอิง	80
เอกสารอ้างอิง (ต่อ)	81



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางแสดงวิธีและช่วงเวลาในการดำเนินงานวิจัย.....	35
3.2 ตารางรายละเอียด Use Case ระบบ Log in เข้าใช้ระบบ Leave Request.....	41
3.3 ตารางรายละเอียด Use Case ระบบ Authentication ของระบบ Leave Request.....	42
3.4 ตารางรายละเอียด Use Case ระบบคำขอลา ของระบบ Leave Request.....	43
3.5 ตารางรายละเอียด Use Case ระบบส่งคำอนุญาตหรือไม่อนุญาต ของระบบ Leave Request.....	45
3.6 ตารางรายละเอียด Use Case ระบบตรวจสอบจำนวนวันลาที่เหลือ ของระบบ Leave Request.....	46
3.7 ตารางรายละเอียด Use Case ระบบตรวจสอบคำขอลาที่ผ่านมา ของระบบ Leave Request.....	47
3.8 ตารางเก็บข้อมูลส่วนตัวของผู้ใช้ (UserInfo).....	50
3.9 ตารางเก็บรายละเอียดการจัดการแอดเคาท์ของผู้ใช้ (Manages).....	51
3.10 ตารางเก็บข้อมูลตำแหน่ง Staff, Supervisor และ Head of operation ของผู้ใช้แต่ละคน.....	52
3.11 ตารางเก็บชื่อและรหัสของตำแหน่ง Staff, Supervisor และ Head of operation (Role).....	52
3.12 ตารางเก็บรายละเอียดของประเภทการลา (LeaveRequestType).....	53
3.13 ตารางเก็บรายละเอียดของคำขอลา (Leave).....	53
3.14 ตารางเก็บสถานะของคำขอลา (Status).....	54
3.15 ตารางเก็บช่วงเวลาที่ต้องการลา (Time).....	54
3.16 ตารางเก็บข้อมูลของการปฏิเสธคำขอลาพร้อมเหตุผล (Reply).....	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้ามอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 วิวัฒนาการของโลโก้ระบบปฏิบัติการ Android ตั้งแต่ ปี 2008 – 2013.....	5
2.2 สถิติจำนวนแอปพลิเคชันบน Google Play Store ตั้งแต่ปี 2009 – 2016.....	6
2.3 โลโก้ของภาษา Java.....	6
2.4 ตัวอย่าง Android Virtual Device ที่มีมาให้ใน Android SDK.....	7
2.5 ตัวอย่างหน้าต่างของ User Interface ที่มีให้เลือกใช้ใน Android Support Library.....	8
2.6 กระบวนการพัฒนาโปรแกรมตามแนวความคิดแบบ Agile Methodology.....	11
2.7.1 Backlog ของระบบ Leave Request ในรูปแบบลิสต์รายการ.....	13
2.7.2 Backlog ของระบบ Leave Request ในรูปแบบกระดาน.....	14
2.7.3 ตัวอย่าง Sprint Backlog ของระบบ Leave Request ใน Sprint ที่ 5 ในรูปแบบลำดับ.....	14
2.7.4 ตัวอย่าง Sprint Backlog ของระบบ Leave Request ใน Sprint ที่ 5 ในรูปแบบกระดาน.....	15
2.7.5 การประชุม daily scrum ภายในองค์กร.....	16
2.7.6 ตัวอย่างกราฟ Burn-Down Chart.....	16
2.8.1 ตัวอย่าง Wireframe ด้วยวิธีการวาด.....	17
2.8.2 Wireframe ของแอปพลิเคชัน Leave Request ที่ทำด้วยโปรแกรม Photoshop.....	18
2.8.3 ภาพรวม Wireframe ของแอปพลิเคชัน Leave Request.....	19
2.9.1 Prototype ของแอปพลิเคชัน Leave Request ที่ออกแบบโดยใช้เครื่องมือจากเว็บไซต์ www.marvelapp.com.....	21
2.9.2 ภาพรวม Prototype ของแอปพลิเคชัน Leave Request ที่ออกแบบโดยใช้เครื่องมือจากเว็บไซต์ www.marvelapp.com.....	21
2.9.3 แสดงถึงจุดที่สามารถใช้งานแล้วมีการตอบสนองของตัว Prototype ของแอปพลิเคชัน Leave Request ที่ออกแบบโดยใช้เครื่องมือจากเว็บไซต์ www.marvelapp.com.....	22
2.10.1 Version Control Systems แบบ Local.....	23
2.10.2 Version Control Systems แบบรวมศูนย์.....	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.10.3 Version Control Systems แบบกระจายศูนย์.....	25
2.10.4 โปรแกรม Source Tree.....	26
2.10.5 ตัวอย่าง เว็บไซต์ www.github.com เว็บไซต์สำหรับใช้งาน Git ที่ได้รับความนิยมในปัจจุบัน.....	26
2.11.1 ตัวอย่างภาษา XML ที่ใช้ในการพัฒนาแอปพลิเคชัน Leave Request.....	27
2.11.2 ตัวอย่างส่วนติดต่อผู้ใช้ของแอปพลิเคชัน Leave Request ที่ถูกกำหนดด้วยภาษา XML.....	28
2.12.1 ตัวอย่างการแบ่งส่วนของซอฟต์แวร์ตามสถาปัตยกรรมซอฟต์แวร์แบบ MVC ของแอปพลิเคชัน Leave Request.....	29
2.12.2 Workflow Diagram ของสถาปัตยกรรมซอฟต์แวร์แบบ MVC.....	29
2.12.3 หลักการทำงานของซอฟต์แวร์ที่ใช้โครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC.....	30
2.13.1 ตัวอย่าง Third party library “Picasso”.....	31
2.13.2 ตัวอย่างหน้าเว็บไซต์ www.android-arsenal.com ที่เป็นแหล่งรวบรวม Third party library..	32
2.14.1 ตัวอย่างเปรียบเทียบการออกแบบให้ได้ User Experience ที่ดีและไม่ดี.....	33
3.2.1 โครงสร้างการทำงานของระบบ Leave Request ส่วนของแอปพลิเคชัน.....	36
3.2.2 Flowchart diagram ของแอปพลิเคชัน Leave Request	37
3.2.3 Use case diagram โดยรวมของระบบ Leave Request.....	38
3.2.4 Use case diagram พี่เจอร์ Log in ของแอปพลิเคชัน Leave Request.....	39
3.2.5 Use case diagram พี่เจอร์เกี่ยวกับการส่งคำขอลา และการตอบรับคำขอลาของแอปพลิเคชัน Leave Request.....	40
3.2.6 Use case diagram พี่เจอร์ตรวจสอบจำนวนวันลาที่เหลือและประวัติการขอลาที่ผ่านมาของแอปพลิเคชัน Leave Request.....	40
3.3.1 แผนภาพ ER Diagram ระบบฐานข้อมูลของระบบ Leave Request.....	49
3.4.1 แผนภาพลำดับเหตุการณ์ระบบล็อกอินเข้าสู่ระบบ Leave Request.....	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และX้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.4.2 แผนภาพลำดับเหตุการณ์ระบบตรวจสอบจำนวนวันลาที่เหลือของระบบ Leave Request.....	56
3.4.3 แผนภาพลำดับเหตุการณ์ระบบตรวจสอบวันลาที่ผ่านมาของระบบ Leave Request.....	57
3.4.4 แผนภาพลำดับเหตุการณ์ระบบส่งคำขอลาของระบบ Leave Request.....	58
3.4.5 แผนภาพลำดับเหตุการณ์การอนุญาตให้ลาได้ของระบบ Leave Request.....	58
3.4.6 แผนภาพลำดับเหตุการณ์การไม่อนุญาตให้ลาของระบบ Leave Request.....	59
3.5.1 แผนภาพแสดงส่วนติดต่อผู้ใช้ของแอปพลิเคชัน Leave Request ทั้งหมด.....	60
3.6.1 ไฟล์กลุ่ม Model ของแอปพลิเคชัน Leave Request	62
3.6.2 ไฟล์ Fragment ที่ทำหน้าที่เป็น View ของแอปพลิเคชัน Leave Request	62
3.6.3 ไฟล์ที่ทำหน้าที่เป็น Controller หลักของแอปพลิเคชัน Leave Request	63
3.6.4 ข้อมูลของผู้ใช้ที่นำมาแสดงผลบน Navigation Drawer	64
4.1 หน้าลือคอินของแอปพลิเคชัน	66
4.2 ข้อความแจ้งเตือนผู้เข้าผู้ที่ไม่ได้กรอก password และ username ไม่ถูกต้องตามรูปแบบที่แอปพลิเคชันกำหนด	67
4.3 ข้อความแจ้งเตือนว่าไม่สามารถลือคอินเข้าสู่ระบบได้.....	67
4.4 ข้อความแจ้งเตือนปัญหาที่เกิดขึ้นทางฝั่งเซิร์ฟเวอร์.....	68
4.5 หน้าแสดงจำนวนวันลา และข้อความตรวจสอบสถานะการลือคอินของผู้ใช้.....	69
4.6 หน้ากรอกแบบฟอร์มคำขอลาเมื่อผู้ใช้กดปุ่ม “คำขอลา”.....	69
4.7 drop down list แสดงตัวเลือกประเภทวันลาและช่วงเวลาที่ต้องการลา.....	70
4.8 ปฏิทินที่ใช้สำหรับเลือกวันที่ต้องการลา.....	70
4.9 หน้ากรอกแบบฟอร์มที่มีรายละเอียดครบถ้วน และข้อความเมื่อส่งคำขอลาสำเร็จ.....	71
4.10 หน้าแสดงลิสต์คำขอลาล่าสุด.....	71
4.11 หน้ารายละเอียดคำขอลา.....	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
4.12 เปรียบเทียบเมนู Navigation drawer ของผู้ใช้แต่ละตำแหน่ง.....	72
4.13 หน้าแสดงลิสต์ของคำขอลาที่ส่งเข้ามาถึง Supervisor หรือ Head of operation.....	73
4.14 หน้าประวัติการลาที่ผ่านมา และหน้าคำขอลาที่เข้ามา เมื่อไม่มีข้อมูล	74
4.15 หน้าแสดงรายละเอียดของคำขอลาที่เข้ามา.....	74
4.16 ข้อความยืนยันว่าผู้ใช้ต้องการอนุญาตให้ลา.....	75
4.17 ขั้นตอนการไม่อนุญาตให้ลา.....	75
4.18 อีเมลล์แจ้งเตือนว่ามีผู้ใช้ส่งคำขอลามา.....	76
4.19 อีเมลล์แจ้งว่าคำขอได้รับการอนุญาตให้ลาได้.....	76
4.20 อีเมลล์แจ้งว่าคำขอไม่ได้รับการอนุญาต.....	77
4.21 การเปลี่ยนภาษาของแอปพลิเคชัน.....	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ **ขอ** อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

แม้ว่าในปัจจุบันได้มีการพัฒนาเทคโนโลยีการจัดเก็บข้อมูลทะเบียนประวัติพนักงานในระบบโปรแกรมซอฟต์แวร์ ทั้งชนิด ยี่ห้อหรือรุ่นต่างๆ มากมาย เพื่อใช้ในการจัดเก็บเป็นฐานข้อมูลซึ่งเป็นการประหยัดพื้นที่และลดการใช้เอกสารการจัดเก็บข้อมูลส่วนตัวของพนักงาน แต่สำหรับเอกสารบางประเภทยังคงต้องมีการจัดเก็บเป็นรูปแบบกระดาษหรือรายงานอยู่เพื่อเป็นหลักฐานที่ต้องมีการลงลายมือชื่อของพนักงานและหัวหน้าหรือผู้ดูแลของพนักงานคนนั้นๆ ทำให้หลายๆองค์กรยังคงต้องมีการเก็บเอกสารดังกล่าวนี้ด้วยรูปแบบกระดาษหรือเอกสารแบบยุคก่อนมีเทคโนโลยีอยู่

เอกสารคำขอลา ถือเป็นเอกสารสำคัญที่ยังคงมีการเขียนด้วยลายมือและต้องมีการลงลายมือชื่อของพนักงานที่มีความประสงค์จะลา รวมถึงลายมือชื่อของผู้มีอำนาจที่สามารถลงชื่ออนุญาตให้พนักงานแต่ละคนลาได้ด้วย นอกจากนี้ขั้นตอนการลงรายละเอียดการขอลาและการลงชื่อเพื่ออนุญาตและขอลาในแต่ละครั้งแล้ว ยังต้องมีการเก็บเอกสารดังกล่าวรวบรวมเป็นแฟ้มเอกสารอีกโดยเป็นหน้าที่ของฝ่ายจัดสรรทรัพยากรมนุษย์ของแต่ละองค์กร (Human Resource : HR) จากกระบวนการดังกล่าวทำให้การส่งคำขอลาแต่ละครั้งเป็นเรื่องที่ค่อนข้างเสียเวลาในการยื่นคำขอลาแล้วต้องรอผลการอนุมัติการลาจากทางผู้มีอำนาจ หลังจากนั้น ทางฝ่าย HR ของแต่ละองค์กรจะเป็นฝ่ายรับเอกสารที่ได้รับการอนุมัติการลาแล้วมาจัดการหักล้างวันลาที่เหลือของพนักงานที่ต้องการลาตามจำนวนวันลาที่พนักงานแจ้งขอลาไว้ในส่วนนี้ทำให้เป็นการยากขึ้นไปอีกสำหรับพนักงานที่ต้องการทราบวันลาที่เหลือของตัวเอง ก็ไม่สามารถทำได้ด้วยตัวเอง ต้องไปสอบถามจากทางฝ่าย HR เอง ซึ่งถ้าหากวันใดเกิดฝ่าย HR ลาหยุด แล้วพนักงานต้องการทราบวันลาของตัวเอง อาจจะไม่สามารถทราบได้ และอาจไม่สามารถขอลาได้ด้วย

การลาหยุดของแต่ละองค์กรมักจะมีการกำหนดไว้หลักๆอยู่ 3 ประเภท ดังนี้

- ลาป่วย (Sick Leave)
- ลากิจ (Personal Leave)
- ลาพักร้อน (Annual Leave)

ซึ่งการลาแต่ละประเภท แต่ละองค์กรก็อาจจะมีการลาประเภทอื่นๆเพิ่มเติมอีก เช่น ลาไปเข้าพิธีอุปสมบท, ลาไปเกณฑ์ทหาร, ลาคลอดบุตร เป็นต้น ซึ่งแต่ละองค์กรก็จะกำหนดจำนวนวันที่ลาได้ในแต่ละประเภทการลาแตกต่างกันไปอีก นี่จึงเป็นอีกเหตุผลสำคัญที่ทุกครั้งที่จะลาไม่ว่าจะด้วยเหตุผลใด พนักงานจำเป็นต้องเข้ามาตรวจสอบจำนวนวันลาที่เหลือในการลาแต่ละประเภทของตนเอง เพื่อจะได้มีการวางแผนเพื่อที่จะลาหยุดและรักษาผลประโยชน์ของตัวเองไว้ให้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้แล้วในหลายๆองค์กรที่มีพนักงานแต่ละแผนกอย่างจำกัด หากจะมีพนักงานคนใดคนหนึ่งลาจะต้องมีการตรวจสอบด้วยว่า ในระหว่างที่พนักงานคนนั้นลา งานที่ทางแผนกกำลังดำเนินอยู่นี้จะสามารถดำเนินงานได้ต่อหรือไม่ แม้จะมีพนักงานที่ทำตำแหน่งเดียวกันหลายคน แต่ถ้าหากพนักงานทุกคนลาพร้อมกันก็ทำให้งานไม่สามารถดำเนินต่อไปได้ ทำให้การลาแต่ละครั้งพนักงานจะต้องมีการวางแผนการลาให้เหมาะสมและไม่ตรงกันกับพนักงานในตำแหน่งเดียวกันด้วย เพื่อไม่ให้เกิดการลากระทบกับการดำเนินงาน

ด้วยปัญหาต่างๆ ดังที่กล่าวมาข้างต้นนี้ จึงได้มีการนำเทคโนโลยีที่มีอยู่ใกล้ตัว มาประยุกต์ใช้กับระบบการลาที่ทั้งระบบ ซึ่งจะมีส่วนช่วยให้การลาหยุดแต่ละครั้งดำเนินการได้รวดเร็วมากขึ้น พนักงานสามารถตรวจสอบวันลาได้ตามต้องการโดยไม่ต้องอาศัยการตรวจสอบผ่านฝ่าย HR นอกจากนี้ยังสามารถวางแผนการลาร่วมกันในแผนกเดียวกัน เพื่อไม่ให้เกิดการลาของพนักงานต้องกระทบกับการดำเนินงานด้วย

1.2 วัตถุประสงค์ของการวิจัย

- เพื่อลดเวลาในการส่งคำขอเพื่อลาหยุดแต่ละครั้ง
- เพื่อลดการใช้กระดาษในการส่งคำขอลาแต่ละครั้ง
- เพื่อลดภาระและหน้าที่ของฝ่าย HR
- เพื่อช่วยให้การลาของพนักงานไม่ส่งผลกระทบต่อการทำงาน
- เพื่อช่วยให้การเก็บเอกสารที่เกี่ยวข้องกับการลาทำได้ง่ายและเป็นระเบียบมากขึ้น
- เพื่อช่วยให้การตรวจสอบประวัติการขอลาหยุดหรือจำนวนวันลาที่เหลือของพนักงานทำได้สะดวกมากขึ้น

1.3 ขอบเขตของการวิจัย

1.3.1 พนักงานทั่วไป (User)

- สามารถตรวจสอบวันลาที่เหลือของตนเองได้
- สามารถส่งคำขอลาไปยังผู้ที่มีอำนาจอนุญาตให้ลาหยุดได้
- สามารถตรวจสอบประวัติการลาที่ผ่านมาของตนเองได้

1.3.2 หัวหน้างาน หรือผู้ที่มีอำนาจอนุญาตให้พนักงานลาได้ (Supervisor, Head of operation)

- สามารถตรวจสอบวันลาของตัวเองได้
- สามารถตรวจสอบวันลาของพนักงานภายใต้การดูแลของตัวเองได้
- สามารถอนุมัติคำขอลาหรือปฏิเสธคำขอลาได้
- สามารถรับ Email แจ้งเตือนเมื่อมีคำขอลาส่งมาถึงได้

1.3.3 ฝ่าย HR หรือแผนกที่เกี่ยวข้องเกี่ยวกับเอกสารของพนักงาน (Admin)

- สามารถจัดการเกี่ยวกับระบบฐานข้อมูลของระบบการลานั้นได้
- สามารถตรวจสอบวันลาของพนักงานได้ทุกคน
- สามารถเข้าถึงข้อมูลการลาของพนักงานทุกอย่างโดยสมบูรณ์
- สามารถนำข้อมูลการลาของพนักงานทุกคนมาสรุปลงเอกสารรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **ประจำปีขององค์กร** ได้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีดำเนินการวิจัย

1.4.1 วิเคราะห์และแลกเปลี่ยนความคิดเห็นกับพนักงานที่ปรึกษาถึงปัญหาที่เกิดขึ้นในการทำงานและชีวิตประจำวัน โดยเริ่มวิเคราะห์ตั้งแต่สาเหตุของปัญหา ประเภทของปัญหา หลังจากนั้นจึงคิดหาวิธีแก้ไขปัญหาดังกล่าว จนกระทั่งเป็นหัวข้อของรายงานชิ้นนี้

1.4.2 นำปัญหาและวิธีแก้ไขไปสำรวจว่ารายงานชิ้นนี้จะสามารถช่วยแก้ปัญหาที่เกิดขึ้นดังที่กล่าวไปในหัวข้อที่ 1.1 ได้จริงหรือไม่ด้วยการนำเสนอและสอบถามกับกลุ่มเป้าหมาย ซึ่งก็คือบุคลากรภายในองค์กรแผนกต่างๆทุกแผนก และทุกคน โดยกลุ่มเป้าหมายดังกล่าวมีจำนวนประมาณ 30 - 40 คน หลังจากนั้นจึงนำผลการสำรวจนี้มาสรุปว่าควรทำรายงานชิ้นนี้หรือไม่

1.4.3 เริ่มวางแผนการดำเนินงานว่าจะเริ่มพัฒนาจากส่วนใดของแอปพลิเคชัน เพื่อให้แอปพลิเคชันสามารถนำไปใช้งานได้จริงเร็วที่สุด โดยมีการแบ่งช่วงพัฒนาของแอปพลิเคชันเป็นดังนี้

ช่วงที่ 1 : บุคลากรแต่ละคนสามารถตรวจสอบวันลาที่เหลือของตนได้จากแอปพลิเคชัน

ช่วงที่ 2 : บุคลากรแต่ละคนสามารถส่งคำขอลาไปยังผู้มีอำนาจพิจารณาคำขอลาได้ด้วยแอปพลิเคชัน

ช่วงที่ 3 : ผู้ที่มีอำนาจอนุญาตการลา สามารถตอบรับหรือปฏิเสธคำขอลาจากบุคลากรที่ส่งมาถึงตนได้ผ่านแอปพลิเคชัน

ช่วงที่ 4 : บุคลากรแต่ละคนสามารถตรวจสอบประวัติการขอลาที่ผ่านมาของตนได้ผ่านแอปพลิเคชัน

ช่วงที่ 5 : บุคลากรสามารถตรวจสอบวันลางวดหน้าของบุคลากรคนอื่นๆที่อยู่ในแผนกเดียวกันได้ เพื่อร่วมกันวางแผนการดำเนินงานต่อไป

1.4.4 ออกแบบโครงสร้างและส่วนติดต่อผู้ใช้ (User Interface) ของแอปพลิเคชันเพื่อให้ผู้ใช้งานแอปพลิเคชันสามารถใช้งานแอปพลิเคชันได้ง่ายและถูกต้อง

1.4.5 เริ่มพัฒนาแอปพลิเคชันด้วยโปรแกรม Android Studio โดยเริ่มพัฒนาเป็นฟีเจอร์แบ่งตามช่วงดังข้อที่ 1.4.3 และได้มีการปรับปรุงตัว Source code รวมถึงแก้ไขปัญหบางจุดที่เกิดขึ้นเมื่อทำการทดสอบแอปพลิเคชันเป็นระยะๆ

1.4.6 เมื่อพัฒนาแอปพลิเคชันจนสามารถใช้งานได้ครบทุกฟีเจอร์ตามที่กำหนดแล้ว จึงนำไปให้กลุ่มตัวอย่างผู้ใช้ ได้ทดลองใช้ และบันทึกผลการใช้งาน ก่อนจะนำปัญหาที่เกิดขึ้นกลับมาแก้ไขปัญหาต่อไปจนแอปพลิเคชันเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- ช่วยให้การส่งคำขอลาเป็นระบบระเบียบยิ่งขึ้น
- ช่วยให้การเก็บข้อมูลของบุคลากรภายในองค์กรจากที่เก็บเป็นกระดาษเอกสาร จะเปลี่ยนมาเป็นการเก็บในระบบฐานข้อมูลโดยสมบูรณ์
- ช่วยลดการใช้กระดาษในการนำไปทำเป็นใบขออนุญาตลา
- ช่วยแบ่งเบาภาระของฝ่ายจัดสรรทรัพยากรมนุษย์ (Human Resource)
- ช่วยให้การวางแผนเพื่อการทำงานร่วมกันเป็นทีมเป็นไปในทางที่ดีขึ้น
- ช่วยให้บุคลากรภายในองค์กรสามารถวางแผนในการลาหยุดของตัวเองได้เป็นระยะๆ
- ช่วยลดรายจ่ายในงบประมาณให้องค์กรได้



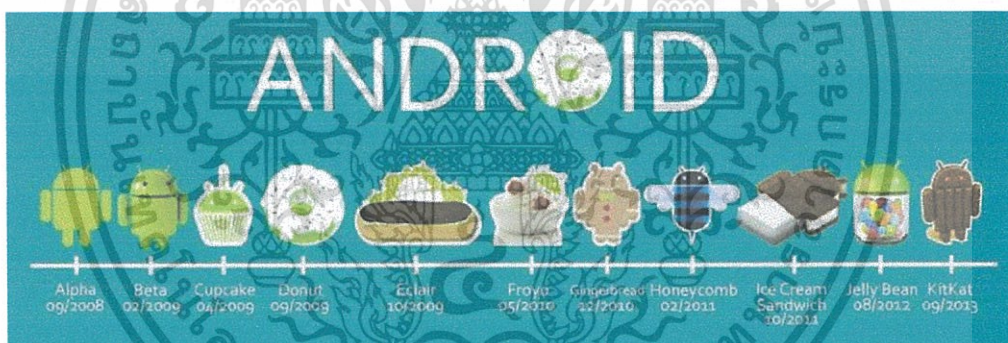
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบปฏิบัติการ Android

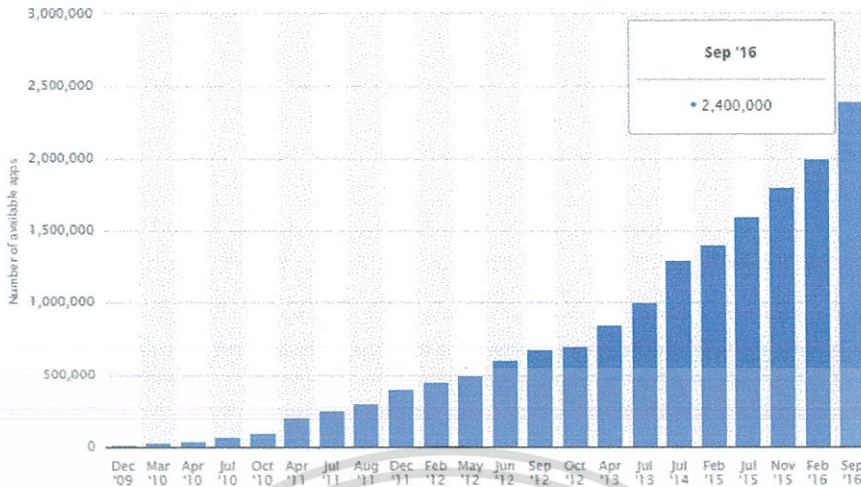
Android คือ ระบบปฏิบัติการบนสมาร์ทโฟน (Smart Phone) และแท็บเล็ต (Tablet) แบบ Open source ที่มีพื้นฐานอยู่บนระบบปฏิบัติการ Linux อีกชั้นหนึ่ง (Base on Linux) โดยเริ่มคิดค้นและพัฒนาโดยบริษัท Android, Inc. ก่อนที่บริษัท Google จะเข้าซื้อกิจการในปี พ.ศ. 2548 หลังจากนั้นในปี พ.ศ. 2550 Google จึงได้ทำการเปิดตัวระบบปฏิบัติการ Android อย่างเป็นทางการ และสมาร์ทโฟนรุ่นแรกที่ได้ใช้ระบบปฏิบัติการ Android คือ HTC Dream ที่วางจำหน่ายในปี พ.ศ. 2551 ซึ่งในปัจจุบัน Android นับเป็นระบบปฏิบัติการบนสมาร์ทโฟนที่ได้รับความนิยมมากที่สุด นอกจากนี้แล้วในปัจจุบันยังถูกนำไปใช้บนอุปกรณ์อิเล็กทรอนิกส์อื่นๆอีก เช่น กล้องดิจิทัลระบบ Android, Smart TV ระบบ Android, Android Wear (เช่น นาฬิกาข้อมือ, แว่นตา) เป็นต้น



ภาพที่ 2.1 วิวัฒนาการของโลโก้ระบบปฏิบัติการ Android ตั้งแต่ ปี 2008 – 2013

ที่มา : <http://www.xda-developers.com/wp-content/uploads/2013/11/android-evolution.jpg>

Android เป็นระบบปฏิบัติการแบบเปิดหรือที่เรียกว่า Open source ซึ่งอนุญาตให้ผู้ผลิต, นักพัฒนา และผู้ให้บริการเครือข่าย สามารถปรับแต่งและจัดจำหน่ายได้ (ภายใต้เงื่อนไขที่ทาง Google ได้กำหนดไว้) นอกจากนี้การพัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการ Android ยังใช้ภาษาที่ใช้กันอย่างแพร่หลายอย่างภาษา Java เป็นภาษาหลัก ทำให้ง่ายต่อการพัฒนาและด้วยตัวช่วยหลายๆอย่างที่ทาง Google ได้จัดสรรไว้ให้สำหรับนักพัฒนาทุกระดับได้ใช้ในการพัฒนาแอปพลิเคชันทำให้ใครๆก็สามารถพัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการ Android ได้ ส่งผลให้บน Google Play Store ที่เป็นแหล่งให้ดาวน์โหลดแอปพลิเคชันสำหรับระบบปฏิบัติการ Android มีจำนวนแอปพลิเคชันสูงถึง 2,000,000 แอปพลิเคชันในปัจจุบัน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.2 สถิติจำนวนแอปพลิเคชันบน Google Play Store ตั้งแต่ปี 2009 – 2016

ที่มา : <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

2.2 ภาษา Java

ภาษา Java เป็นภาษาเพื่อการพัฒนาโปรแกรมเชิงวัตถุ หรือ Object Oriented Programming (OOP) ถูกคิดค้นและพัฒนาขึ้นในปี พ.ศ. 2534 ก่อนจะสำเร็จและเปิดตัวอย่างเป็นทางการในปี พ.ศ. 2538 โดย James Gosling และทีมวิศวกร ของบริษัท Sun Micro System ซึ่งมีจุดประสงค์หลักในการพัฒนาภาษา Java เพื่อนำมาใช้แทนภาษา C++ และจากรูปแบบของภาษา Java ที่ถูกเพิ่มขึ้นมาทำให้ตัวภาษาส่วนที่เพิ่มเติมขึ้นมา มีความคล้ายกับภาษา Objective-C

ชื่อภาษา Java นั้นมีที่มาจากกาแฟ และมีความคล้ายคลึงกับชื่อภาษา Javascript แต่ตัวภาษา และการนำไปใช้นั้นแตกต่างกันเป็นอย่างมาก ทั้งภาษา Java และภาษา Javascript ไม่มีส่วนเกี่ยวข้องใดๆกัน ทั้งสิ้น



ภาพที่ 2.3 โลโก้ของภาษา Java

ที่มา : <https://www.kinvey.com/wp-content/uploads/2013/11/java-logo.jpg?w=300>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Android SDK

Android SDK หรือ Android Software Development Kit เป็นชุดโปรแกรมที่ทางบริษัท Google ได้คิดค้นและพัฒนาขึ้นเพื่อให้ นักพัฒนาได้ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยทางบริษัท Google ได้แจกจ่าย Android SDK ให้ใช้โดยไม่คิดค่าใช้จ่ายใดๆทั้งสิ้น ด้วยเหตุนี้เองจึงทำให้ระบบปฏิบัติการ Android มีแอปพลิเคชันเพิ่มขึ้นอย่างรวดเร็ว

ใน Android SDK นี้จะประกอบไปด้วยโปรแกรมช่วยเหลือและไลบรารีต่างๆที่จำเป็นและช่วยให้การพัฒนาแอปพลิเคชันทำได้สะดวกสบายยิ่งขึ้น เช่น Android Virtual Device หรือ Emulator ที่ช่วยให้ผู้ที่พัฒนาแอปพลิเคชันสามารถทดสอบแอปพลิเคชันบน Emulator ได้โดยไม่จำเป็นต้องมีสมาร์ตโฟน เพราะ Android Virtual Device นี้มีการจำลองสภาพแวดล้อม, ทรัพยากรตัวเครื่องของสมาร์ตโฟนระบบปฏิบัติการ Android ที่สามารถใช้งานได้ใกล้เคียงกับสมาร์ตโฟนระบบปฏิบัติการ Android เครื่องจริงเกือบทั้งหมด



ภาพที่ 2.4 ตัวอย่าง Android Virtual Device ที่มีมาใน Android SDK

2.4 Java JDK

Java JDK หรือ Java Development Kit เปรียบเสมือนชุดเครื่องมือที่ใช้ในการพัฒนาโปรแกรมด้วยภาษา Java ที่ประกอบไปด้วย Java compiler, Java debugger, Java doc และ Java interpreter หรือ Java VM ถ้าหากนักพัฒนาไม่ทำการติดตั้ง Java JDK แล้วจะไม่สามารถทำการ compile หรือ run โปรแกรมหรือแอปพลิเคชันได้ ทั้งนี้ Java SDK แบ่งย่อยได้ 3 รุ่น ดังนี้

- Java SE (Standard Edition) สำหรับพัฒนาโปรแกรมบนคอมพิวเตอร์ทั่วไป
- Java ME (Micro Edition) สำหรับพัฒนาโปรแกรมบนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ

หรือแท็บเล็ต ส่วนมากใช้ในการพัฒนาเกม

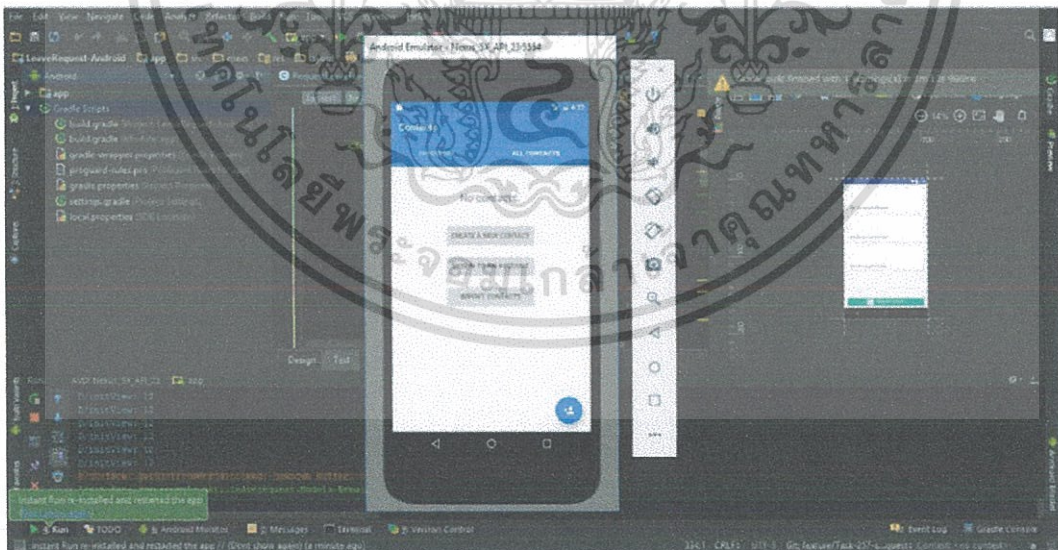
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนขึ้นเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Java EE (Enterprise Edition) สำหรับพัฒนาโปรแกรมในองค์กรใหญ่ๆ หรือมีขอบเขตของโครงการกว้างมาก

2.5 Android Support Library

ด้วยสาเหตุที่ระบบปฏิบัติการ Android นั้นมีการพัฒนาอย่างต่อเนื่องและรวดเร็วมาก ทำให้การพัฒนาแอปพลิเคชันเพื่อให้รองรับกับระบบปฏิบัติการ Android เวอร์ชันใหม่เสมอ ในขณะเดียวกัน ก็ยังมีผู้ใช้ระบบปฏิบัติการ Android เวอร์ชันเก่าก็ยังคงมีอยู่เป็นจำนวนมาก ทางนักพัฒนาจึงจำเป็นต้องพัฒนาแอปพลิเคชันให้รองรับทั้งระบบปฏิบัติการ Android เกือบทุกเวอร์ชัน แต่บางคำสั่งในการพัฒนาแอปพลิเคชันเพื่อให้รองรับกับระบบปฏิบัติการ Android ในเวอร์ชันที่แตกต่างกัน ก็อาจจะต้องใช้คำสั่งในการพัฒนาที่แตกต่างกัน

ปัญหาดังกล่าวถือเป็นปัญหาสำคัญในการพัฒนาแอปพลิเคชันมาก ทางบริษัท Google จึงได้คิดค้นและพัฒนา Android Support Library ที่เปรียบเสมือนชุดคำสั่งที่เป็นมาตรฐานในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android ที่ช่วยให้นักพัฒนาเขียนคำสั่งเพียงชุดเดียวก็สามารถใช้คำสั่งนี้ได้กับระบบปฏิบัติการ Android ได้แบบครอบคลุมเกือบทุกเวอร์ชัน และถ้าหากมีการพัฒนาระบบปฏิบัติการ Android เวอร์ชันใหม่ออกมาอีก ทางนักพัฒนาแอปพลิเคชัน ก็เพียงแค่อัปเดต Android Support Library ให้เป็นเวอร์ชันใหม่ตามคำสั่งในแอปพลิเคชันก็จะถูกปรับปรุงให้ตัวระบบปฏิบัติการเข้าใจได้เอง โดยที่นักพัฒนาแอปพลิเคชัน ไม่จำเป็นต้องไปแก้ไขในระดับชั้น Source code แต่อย่างใด



ภาพที่ 2.5 ตัวอย่างหน้าต่างของ User Interface ที่มีให้เลือกใช้ใน Android Support Library

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 แนวความคิดในการพัฒนาโปรแกรม แบบ Agile

ในยุคแรกของการพัฒนาโปรแกรมคอมพิวเตอร์ มีความต้องการให้โปรแกรมที่พัฒนามีความถูกต้องตามความต้องการของผู้ใช้งาน จึงได้มีการคิดค้นแนวคิดในการพัฒนาโปรแกรมแบบ Waterfall Model ที่มีการแบ่งขั้นตอนการพัฒนาออกเป็น 5 ขั้นตอนซึ่งมีเงื่อนไขว่า จะต้องทำขั้นตอนก่อนหน้าให้เสร็จสิ้นสมบูรณ์ก่อนจึงจะทำขั้นตอนต่อไปได้ และเมื่อเสร็จแต่ละขั้นตอนแล้วอยากกลับมาแก้ไขโปรแกรมในขั้นตอนที่ผ่านไปแล้วจะไม่สามารถย้อนกลับมายังขั้นตอนก่อนหน้าได้ แต่จะต้องไปเริ่มทำตั้งแต่ขั้นตอนแรกใหม่เลย ทำให้ในยุคต่อมาที่โปรแกรมมีความซับซ้อนและผู้จัดการมีการเปลี่ยนแปลงความต้องการได้เรื่อยๆ ทำให้เกิดการแก้ไขโปรแกรมอยู่บ่อยครั้ง ทำให้แนวคิดในการพัฒนาโปรแกรมแบบ Waterfall ไม่สามารถทำให้การพัฒนาโปรแกรมเสร็จสิ้นตามเวลาที่กำหนดได้ จนทำให้เกิดแนวความคิดในการพัฒนาโปรแกรมแบบ Agile ขึ้นในช่วงกลางปี พ.ศ. 2533

แนวความคิดในการพัฒนาโปรแกรมแบบ Agile นี้มีความยืดหยุ่นในการปรับเปลี่ยนความต้องการของระบบมากกว่าแบบ Waterfall มาก ทำให้เมื่อมีการเปลี่ยนแปลงความต้องการของระบบที่พัฒนาสามารถย้อนกลับไปแก้ไขได้เรื่อยๆ ไม่ต้องเริ่มพัฒนาใหม่หมดตั้งแต่ขั้นตอนแรกเหมือนแนวคิดในการพัฒนาโปรแกรมแบบ Waterfall นับจากปีแรกที่แนวคิดในการพัฒนาโปรแกรมแบบ Agile ถูกคิดค้นขึ้นก็ได้รับการพัฒนาอย่างต่อเนื่องจนกระทั่งปี พ.ศ. 2544 ทีมนักพัฒนาแนวคิดในการพัฒนาโปรแกรมแบบ Agile ก็ได้ตั้งชื่อให้กับแนวคิดนี้โดยให้มีชื่ออย่างเป็นทางการว่า “Agile Method” หลังจากนั้นก็ได้มีองค์กรที่ไม่ประสงค์จะออกนามได้เข้ามาสนับสนุนการพัฒนาแนวคิด Agile Method นี้ด้วย โดยภาพรวมของแนวคิดในการพัฒนาโปรแกรมแบบ Agile Method นี้มีหลักการ ดังนี้

- ต้องมีการส่งผลลัพธ์ที่เกิดจากการพัฒนาโปรแกรมให้กับผู้ใช้งานอย่างต่อเนื่องและรวดเร็ว
- ต้องมีการรองรับการปรับเปลี่ยนความต้องการของโปรแกรมหรือระบบตามความต้องการของผู้ใช้งานเสมอ แม้จะเป็นช่วงสุดท้ายของการพัฒนาโปรแกรมแล้วก็ตาม เพื่อให้เกิดความได้เปรียบเชิงธุรกิจกับผู้ใช้งานมากที่สุด
- ช่วงเวลาที่ต้องส่งผลลัพธ์ที่เกิดจากการพัฒนาโปรแกรมให้กับผู้ใช้งานนั้น อาจกำหนดให้เป็น ทุกสัปดาห์, ทุกสองสัปดาห์ หรือทุกเดือนก็ได้ แต่ต้องมีการส่งผลลัพธ์ให้กับผู้ใช้งานอย่างสม่ำเสมอ และไม่ทิ้งช่วงนานเกินไป
- บุคลากรที่เป็นผู้ใช้งานและผู้พัฒนาระบบจำเป็นต้องทำงานร่วมกันเสมอ ตลอดจนการพัฒนาโปรแกรมเสร็จสิ้นโดยสมบูรณ์
- ต้องมีการสร้างแรงกระตุ้นให้กับบุคลากรทุกคนที่เกี่ยวข้องในการพัฒนา ด้วยการสร้างสภาพแวดล้อมการทำงานที่ดี ให้การสนับสนุนทุกความต้องการของบุคลากร เพื่อให้ได้ผลงานที่มีศักยภาพสูงสุด
- มีการแลกเปลี่ยนแนวความคิดที่ใช้ในการพัฒนาโปรแกรมแบบเป็นทีม, ทักเซ, ความก้าวหน้าของชิ้นงาน รวมถึงปัญหาในการพัฒนาของนักพัฒนาแต่ละคนอย่างสม่ำเสมอ และตรงไปตรงมา ไม่มีความลับต่อกันในทีมพัฒนา วิธีการที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่สุดคือการประชุมกันต่อหน้า ซึ่งสามารถทำได้ทุกวัน และใช้เวลาในการประชุมต่อครั้งไม่ยาวนานมาก เพื่อไม่ให้กระทบต่อเวลาในการพัฒนาโปรแกรม
- ผลลัพธ์ที่เกิดจากการพัฒนาโปรแกรมในแต่ละครั้งถือเป็นความก้าวหน้าของชิ้นงาน
 - เน้นการพัฒนาโปรแกรมแบบค่อยเป็นค่อยไป และมีข้อจำกัดในการพัฒนาโปรแกรมน้อยที่สุดเท่าที่จะเป็นไปได้
 - เน้นความเรียบง่าย หากมีการพัฒนาที่ไม่สมบูรณ์ในบางครั้งก็ควรรอบการพัฒนาในแต่ละสัปดาห์อาจไม่ยึดติดว่าเป็นผลเสียทั้งหมด แต่สามารถมองกลับเป็นความยืดหยุ่นในการพัฒนาโปรแกรมได้
 - ต้องมีการบริหารทีมในการพัฒนาที่ต่อเนื่อง หากเกิดปัญหาขึ้นทุกคนต้องช่วยกันแก้ปัญหา และหากมีใครทราบถึงปัญหาต้องรีบแจ้งให้ในทีมทราบ เพื่อให้เกิดผลดีต่อชิ้นงานที่พัฒนามากที่สุด

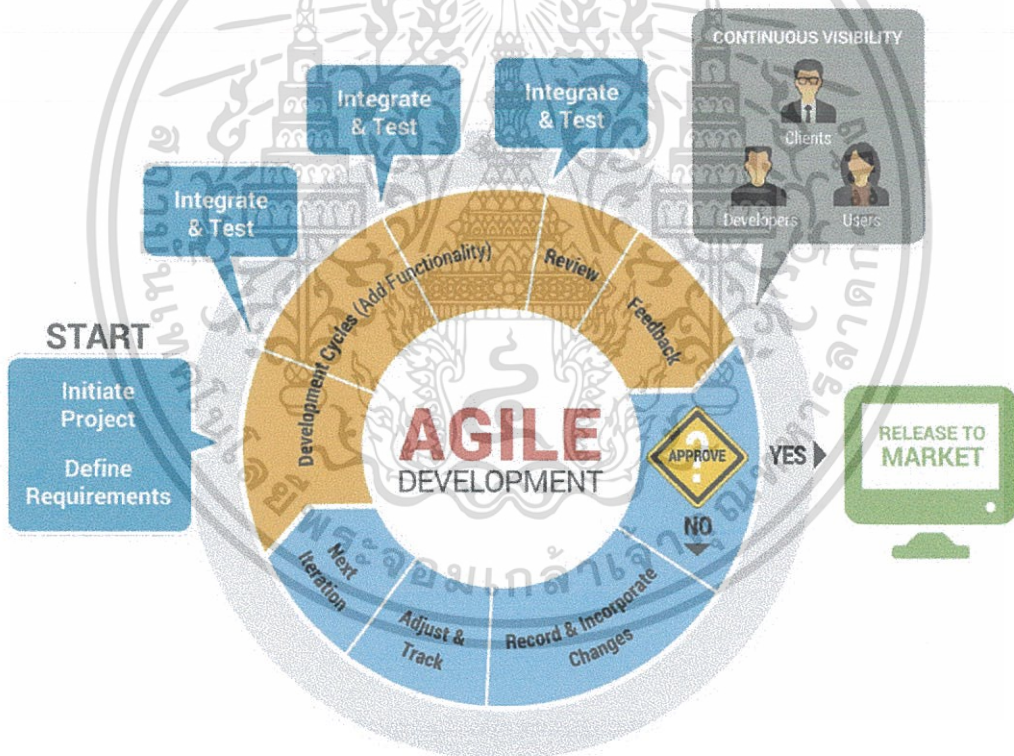
แนวคิดในการพัฒนาโปรแกรมแบบ Agile Method นี้มีจุดประสงค์เพื่อเน้นถึงความถนัดของแต่ละบุคคลในทีมพัฒนาออกมาใช้ในการพัฒนาโปรแกรมให้ได้มากที่สุดเท่าที่จะทำได้และเน้นการพูดคุยแลกเปลี่ยนความคิดเพื่อการพัฒนาตัวเองและพัฒนาโปรแกรมควบคู่กันไปมากกว่าการยึดติดที่กระบวนการพัฒนาหรือเครื่องมือที่ใช้ในการพัฒนา นอกจากนี้ยังมีการทำงานที่ยืดหยุ่นต่อผลลัพธ์ที่เกิดจากการพัฒนาโปรแกรมเป็นหลัก ซึ่งแตกต่างจากแนวคิดในการพัฒนาโปรแกรมแบบ Waterfall ในยุคก่อนที่ยึดติดกับเอกสารที่มีการระบุรายละเอียดของโปรแกรมที่ต้องพัฒนาอย่างละเอียด โดยในอดีตก่อนการพัฒนาจะต้องมีการทำเอกสารที่ระบุรายละเอียดของโปรแกรมที่จะพัฒนาเพื่อนำไปเสนอให้กับผู้จ้างงานก่อนจะตกลงกันว่า ต้องการโปรแกรมแบบนี้จริงหรือไม่ ซึ่งในช่วงแรกผู้จ้างงานอาจตกลงตามเงื่อนไขต่างๆในเอกสารแต่เมื่อเวลาผ่านไป ผู้จ้างงานอาจจะมีความต้องการที่จะเปลี่ยนแปลงรูปแบบของโปรแกรมที่พัฒนาทำให้ต้องเริ่มทำเอกสารเพื่อเสนอให้ผู้จ้างงานเข้าใจในการพัฒนาและตัวโปรแกรมที่จะได้รับอีกครั้ง แต่แนวคิดแบบ Agile จะมีการพัฒนาโปรแกรมทีละส่วนและนำเสนอต่อผู้จ้างงานหรือผู้ใช้เรื่อยๆ หากมีจุดใดที่ผู้ใช้หรือผู้จ้างงานยังไม่พอใจสามารถบอกให้แก้ไขหรือปรับปรุงในการพัฒนาครั้งต่อมาได้แล้วจึงนำกลับไปเสนอต่อผู้จ้างงานหรือผู้ใช้อีกครั้ง ทำให้ไม่ต้องใช้เวลาในการทำเอกสารแบบเสียเปล่า แต่นำเอาเวลาตรงส่วนนี้ไปพบปะ พูดคุยกันโดยตรงจะให้ความชัดเจนได้รวดเร็วมากกว่า

ถึงแม้แนวคิดในการพัฒนาโปรแกรมแบบ Agile Method จะดูเอื้ออำนวยความสะดวกต่อการพัฒนาโปรแกรมแบบเป็นทีมอย่างไรก็ตามแนวความคิดนี้ยังคงมีข้อจำกัดอยู่บางส่วน ดังต่อไปนี้

- วัฒนธรรมภายในองค์กรที่ต้องการใช้แนวความคิดในการพัฒนาโปรแกรมแบบ Agile Method ต้องมีการยอมรับในเรื่องของการเจรจาต่อรองอย่างสม่ำเสมอในการพัฒนาโปรแกรม
- บุคลากรภายในองค์กรต้องเชื่อถือได้ ต้องมีการทำงานร่วมกันอย่างโปร่งใสและตรงไปตรงมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บุคลากรในองค์กรหรือในทีมพัฒนาโปรแกรมต้องมีจำนวนไม่มากจนเกินไป เพื่อให้การเจรจาแลกเปลี่ยนความคิดเห็นกันเป็นไปอย่างเรียบง่ายที่สุด
- บุคลากรภายในองค์กรหรือในทีมพัฒนาควรมีทักษะและความสามารถที่จำเป็นต่อการพัฒนาโปรแกรมมากกว่าบุคลากรที่ใช้แนวคิดในการพัฒนาโปรแกรมแบบอื่นๆ
- องค์กรต้องยอมรับในการตัดสินใจของทีมพัฒนาให้ได้
- องค์กรจะต้องมีสภาพแวดล้อมในการทำงานที่เอื้ออำนวยต่อการติดต่อสื่อสารกันภายในทีมพัฒนาที่สะดวกและรวดเร็วที่สุดเท่าที่เป็นไปได้
- ทีมพัฒนาจะไม่สามารถเร่งรัดเอารายละเอียดของตัวโปรแกรมจากผู้จ้างงานได้ เพราะอาจนำมาซึ่งการแก้ปัญหาที่ไม่ตรงจุดและอาจเกิดปัญหาใหม่มากขึ้นได้เรื่อยๆ เพราะความต้องการของผู้ใช้ที่มีต่อตัวโปรแกรม ควรเกิดจากการไตร่ตรองอย่างละเอียดและถี่ถ้วนของผู้ใช้



ภาพที่ 2.6 กระบวนการพัฒนาโปรแกรมตามแนวความคิดแบบ Agile Methodology

ที่มา : <https://www.xandermar.com/sites/default/files/agile.jpeg>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 รูปแบบในการพัฒนาโปรแกรมแบบ Scrum

Scrum เป็นรูปแบบการพัฒนาในรูปแบบหนึ่งที่น่าแนวความคิดในการพัฒนาโปรแกรม Agile Method มาใช้หรืออาจกล่าวอีกนัยได้ว่า Scrum เป็นรูปแบบในการพัฒนาโปรแกรมรูปแบบหนึ่งของแนวความคิดในการพัฒนาโปรแกรมแบบ Agile Method

Scrum มีจุดประสงค์ที่สำคัญคือ ผู้จ้างวานได้เห็นความก้าวหน้าของโปรแกรมที่สามารถนำไปใช้งานได้จริงอยู่เสมอ และสามารถเปลี่ยนแปลงรายละเอียดของตัวโปรแกรมได้เสมอตามความต้องการ ในขณะที่ทีมพัฒนายังมีความสุขในการพัฒนาโปรแกรมเพราะมีการแบ่งการทำงานอย่างชัดเจนในแต่ละช่วงเวลา มองเห็นว่าอนาคตของโปรแกรมที่กำลังพัฒนาจะออกมาในรูปแบบใดภายในระยะเวลาเท่าใด ซึ่งการแบ่งการทำงานตามช่วงระยะเวลาที่กำหนดทำให้ทีมพัฒนาได้ทำงานแต่พอดีไม่มากหรือน้อยจนเกินความสามารถของทีมพัฒนา นอกจากนี้รูปแบบการพัฒนาโปรแกรมแบบ Scrum จะมีบุคลากรที่รับผิดชอบในชิ้นงานที่รับมาทั้งหมด เรียกว่า Project Manager ที่มีหน้าที่ในการติดตามความก้าวหน้าของการพัฒนา มีการรายงานความก้าวหน้าพร้อมนำผลลัพธ์ที่เกิดจากการพัฒนาโปรแกรมในแต่ละรอบการทำงาน (Sprint) นำเสนอต่อผู้ใช้หรือผู้จ้างวานอยู่เสมอ และด้วยการนำรูปแบบการพัฒนาโปรแกรมแบบ Scrum มาใช้ทำให้ Project Manager มีการติดตามงานที่ง่ายขึ้นเพราะในแต่ละรอบการทำงาน (Sprint) มีการกำหนดไว้ชัดเจนว่าโปรแกรมจะมีส่วนใดพร้อมใช้งานบ้าง นอกจากนี้การที่มีการติดต่อสื่อสารกับผู้ใช้งาน และทีมพัฒนาอย่างสม่ำเสมอ ทำให้การปรับเปลี่ยนรายละเอียดของโปรแกรมเป็นไปได้โดยง่าย

Scrum จะมีหลักการทั้งหมดอยู่บนช่วงเวลาแบ่งเป็นรอบการทำงานครั้งละ 2 – 4 สัปดาห์ เรียกว่า Sprint และมีหลักการตามแนวคิดหลักๆ 3 หลักการ ดังนี้

2.7.1 ทีมบุคลากรในการพัฒนาโปรแกรม หรือ Role

สำหรับทีมบุคลากรในการพัฒนาโปรแกรมด้วยรูปแบบ Scrum จะมีตำแหน่งหน้าที่หลักๆด้วยกัน 3 ตำแหน่ง คือ Scrum Team, Product Owner, Scrum Master

- Scrum Team หรือ ทีมนักพัฒนาโปรแกรมรวมถึงทีมออกแบบด้วย โดยในตำแหน่งนี้จะมีด้วยกันประมาณ 5 – 9 คน ทุกคนจะไม่มีหน้าที่ที่ถูกกำหนดตายตัว ทุกคนสามารถทำงานแทนกันและกันได้ สามารถกำหนดและแจกจ่ายงานกันเองภายในทีมรวมถึงการประเมินระดับของชิ้นงาน ก็สามารถทำกันเองภายใน Scrum Team เลย
- Product Owner เป็นตัวแทนจากทางผู้จ้างวาน มีหน้าที่ในการแบ่งแยกย่อยโปรแกรมทั้งโปรแกรมออกมาเป็นพีเจอรี่ย่อยๆ โดยการกำหนดเหตุการณ์สมมติขึ้นมาเพื่อให้จินตนาการได้ถึงพีเจอรี่ที่ผู้ใช้ต้องการจากโปรแกรมนั้นๆ ก่อนนำผลลัพธ์ที่ได้จากการกำหนดเหตุการณ์สมมตินั้น ไปนำเสนอกับ Scrum Team เพื่อนำไปพัฒนาเป็นโปรแกรมต่อไป โดยต้องวิเคราะห์ออกมาให้ Scrum Team เข้าใจตรงกันได้ง่ายที่สุด

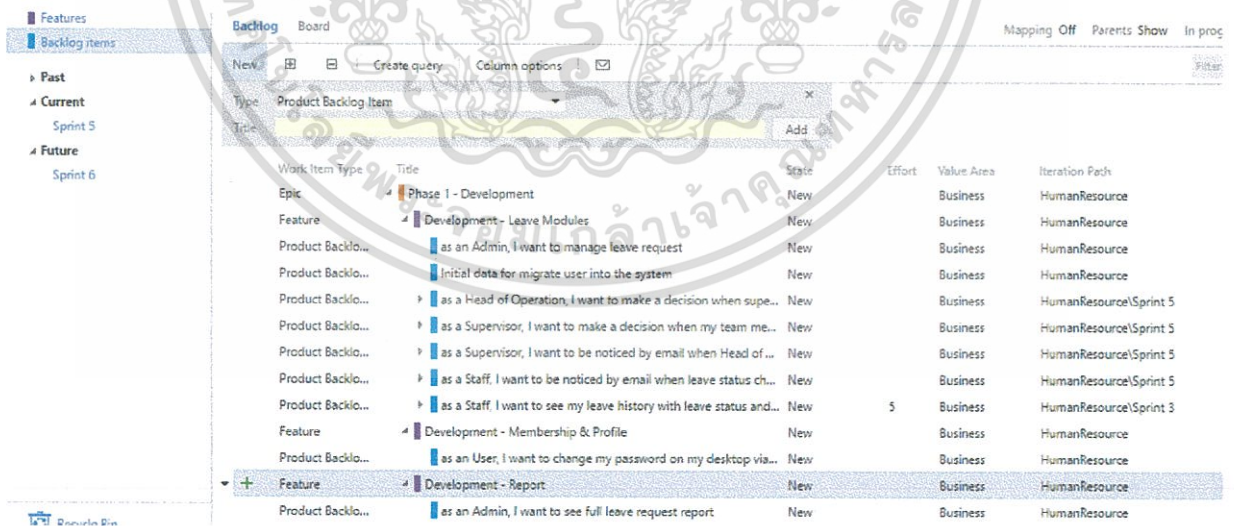
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Scrum Master มีหน้าที่ดูแล, ให้คำแนะนำ, ช่วยแก้ปัญหาให้กับ Scrum Team ทุกคน เปรียบเสมือนโค้ชของ Scrum Team นอกจากนี้ยังมีหน้าที่รับผิดชอบคุณภาพของผลงานที่ถูกพัฒนาออกมาด้วย จัดลำดับความสำคัญของงานที่ต้องทำ ว่าควรทำงานใดก่อนหรือหลัง วิเคราะห์ข้อมูลที่ทาง Product Owner คิดมาให้ และแบ่งเป็นชิ้นงานย่อยๆให้ได้ละเอียดที่สุด แล้วจึงแบ่งงานไปให้ Scrum Team ได้พัฒนาในแต่ละรอบการทำงาน Scrum Master นอกจากจะเป็นโค้ชให้กับ Scrum Team แล้ว ยังเป็นเหมือนผู้นำของ Scrum Team ด้วย เพราะต้องเป็นผู้นำพา Scrum Team เริ่มประชุมในแต่ละครั้งด้วย เพื่อจะได้ทราบถึงความก้าวหน้าของงานและปัญหาที่เกิดขึ้น ก่อนจะเป็นผู้นำที่ช่วยแก้ปัญหาที่ภายใน Scrum Team ร่วมกันแก้เองไม่ได้

2.7.2 วิธีการทำงาน หรือ Process

ในกระบวนการทำงานของรูปแบบการพัฒนาโปรแกรมแบบ Scrum มีดังนี้

- Backlog คือรายการของพีเจอร์ของโปรแกรม ในที่นี้รวมถึงปัญหาที่เกิดจากความผิดพลาดในการเขียนโปรแกรม, รายละเอียดของโปรแกรมที่ผู้ใช้งานหรือผู้ใช้ต้องการปรับเปลี่ยน โดยผู้ที่รับหน้าที่จัดการในส่วนนี้คือ Product Owner ที่จะเข้ามาจัดการลำดับความสำคัญของแต่ละ Backlog ก่อนนำไปให้ Scrum Team เพื่อนำไปพัฒนาโปรแกรมต่อไปในแต่ละรอบการทำงาน



ภาพที่ 2.7.1 Backlog ของระบบ Leave Request ในรูปแบบลิสต์รายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Backlog items

New	Approved	0/5 Committed	0/5 Done
<ul style="list-style-type: none"> + New item as an Admin, I want to manage leave request Initial data for migrate user into the system as an Admin, I want to see full leave request report as a Head of Operation, I want to make a decision when supervisor approved staff's leave request 4/5 as an User, I want to change my password on my desktop via browser as a Supervisor, I want to make a decision when my team member submit leave request 8/10 as a Supervisor, I want to be noticed by email when Head of Operation changed status of my team member request 1/2 as a Staff, I want to be noticed by email when leave status changed 			<ul style="list-style-type: none"> as a Head of Operation, I want to be notice by email when supervisor approved staff's leave request 1/1 as a Supervisor, I want to be notice by email when my team's member request to leave 2/2 As a Staff, when I log in to application I don't want to see request that was sent by other one 2/2 as a Staff, I want to know my remaining leave days in this year for each leave type so that I can plan my leave 5/5 as an Admin, I want to manage supervisor on each staffs 2/2 as an Admin, I want to manage users who can use this application

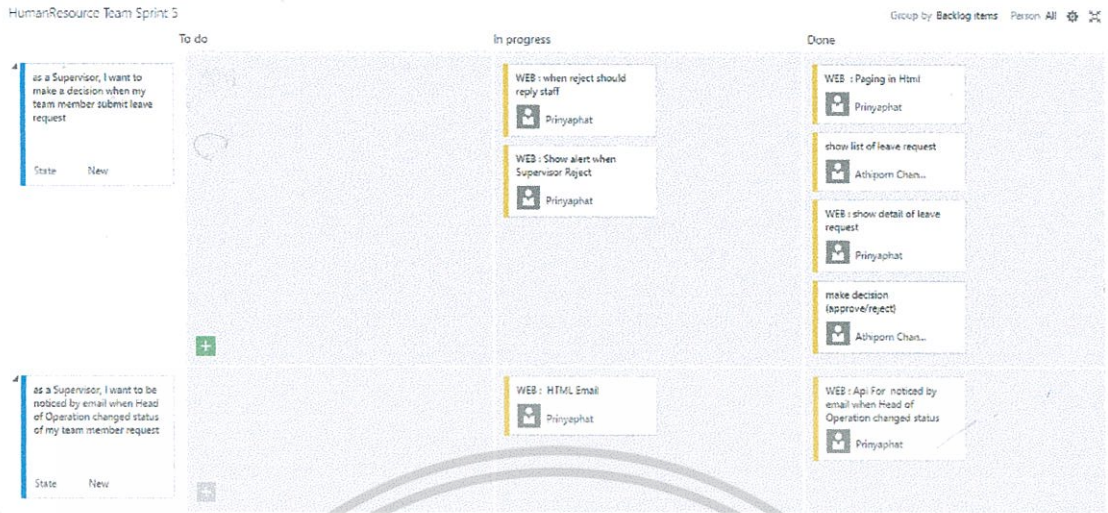
ภาพที่ 2.7.2 Backlog ของระบบ Leave Request ในรูปแบบกระดาน

- Sprint phase คือช่วงเวลาในการทำงานโดยจะแบ่งเป็นระยะเวลาที่ยาวนานเท่าเดิมเสมอตลอดการพัฒนาโปรแกรมนั้นๆ ซึ่งจะแบ่งแต่ละ Sprint เป็นครั้งละ 2 - 4 สัปดาห์ ก่อนเริ่ม Sprint จะมีการนำ Backlog มาจัดลำดับความสำคัญเพื่อนำเข้ามาเป็น Sprint Backlog หรือรายการงานที่ต้องทำให้สำเร็จในแต่ละ Sprint จากนั้น Scrum Team จะเข้ามาวิเคราะห์และพิจารณาร่วมกันในทีมว่า Sprint Backlog ต้องใช้เวลานานเพียงใดในแต่ละ Sprint Backlog ถ้าหากมี Sprint Backlog ใดที่ในทีมตัดสินใจร่วมกันแล้วว่าไม่สามารถทำสำเร็จได้ภายใน Sprint นี้ สามารถต่อรองกับทาง Product Owner เพื่อขอเลื่อนการทำ Sprint Backlog นั้นออกไปทำใน Sprint อื่นได้

Title	State	Assigned To	Rema...
<ul style="list-style-type: none"> as a Head of Operation, I want to make a decision when supervisor ... WEB : Paging in Html implement detail page 	New	Prinyaphat	
<ul style="list-style-type: none"> as a Supervisor, I want to make a decision when my team member ... WEB : Paging in Html show list of leave request WEB : show detail of leave request make decision (approve/reject) WEB : when reject should reply staff WEB : Show alert when Supervisor Reject 	New	Prinyaphat	
<ul style="list-style-type: none"> as a Supervisor, I want to be noticed by email when Head of Opera... WEB : Api For noticed by email when Head of Operation change... WEB : HTML Email 	New	Prinyaphat	
<ul style="list-style-type: none"> as a Staff, I want to be noticed by email when leave status changed WEB : Api For noticed by email when leave status changed WEB : HTML Email 	New	Prinyaphat	

ภาพที่ 2.7.3 ตัวอย่าง Sprint Backlog ของระบบ Leave Request ใน Sprint ที่ 5 ในรูปแบบลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.7.4 ตัวอย่าง Sprint Backlog ของระบบ Leave Request ใน Sprint ที่ 5 ในรูปแบบกระดาน

- Daily Scrum เป็นรูปแบบการประชุมกันโดย Scrum Team และ Scrum Master ที่มีจะมีขึ้นทุกวัน โดยใช้เวลาในการแลกเปลี่ยนความคิดเห็นกัน ประมาณ 10 – 20 นาที ในแต่ละครั้งที่ประชุมกัน ด้วยเวลาการประชุมที่ไม่มากนักจึงนับเป็นการประชุมแบบไม่เป็นทางการมากนัก จึงมักจะจัดการประชุมแบบที่เรียกว่า Stand Up Meeting หรือการประชุมแบบยืนคุยกัน หัวข้อในการประชุมหลักๆ คือ เมื่อวานได้ทำอะไรไปแล้วบ้าง, วันนี้จะทำอะไรบ้าง, ปัญหาที่เจอระหว่างการทำงานในเมื่อวานหรือที่ผ่านมา, ปัญหาดังกล่าวได้รับการแก้ไขแล้วหรือไม่ และสามารถแก้ปัญหาดังกล่าวได้อย่างไร นอกจากนี้ในที่ประชุมอาจมีการแลกเปลี่ยนความคิดเห็นเพิ่มเติมอะไรก็ได้ ที่ช่วยให้การทำงานเป็นไปในทางที่ดีขึ้น ซึ่งทุกคนในที่ประชุมจะต้องประชุมกันด้วยความซื่อตรงและโปร่งใสที่สุดเพื่อประโยชน์สูงสุดในการทำงาน

นอกจากกระบวนการทำงานหลักๆ ทั้ง 3 ข้อที่ได้กล่าวไปแล้วรูปแบบการพัฒนาโปรแกรมแบบ Scrum ยังมีอุปกรณ์อื่นๆ และการทำงานอื่นๆ ที่ช่วยให้การพัฒนาโปรแกรมได้ผลลัพธ์ที่มีศักยภาพของผลงานมากที่สุดด้วย เช่น การประชุมสรุปงานหลังจากจบ Sprint Phase ที่เรียกว่า Sprint Retrospective ที่จะมีการประชุมกันทั้ง Scrum Team และ Scrum Master เพื่อวิเคราะห์และพิจารณาผลงานและการทำงานตลอดช่วง Sprint ที่ผ่านมา เป็นต้น นอกจากนี้แล้วในแต่ละกระบวนการการทำงานของรูปแบบการพัฒนาโปรแกรมแบบ Scrum ยังมีอุปกรณ์และเครื่องมือช่วยให้ทำงานร่วมกันภายในทีมสะดวกและรวดเร็วขึ้นด้วย เช่น Planning Poker ที่เป็นการวัดให้คะแนนระดับความยากของแต่ละ Backlog จะถูกใช้ในการให้คะแนนเพื่อแบ่ง Sprint Backlog ให้เหมาะสมที่สุดในการแบ่งการทำงานในแต่ละ Sprint ซึ่งจะช่วยให้ Scrum Team ได้พิจารณาและทำงานตามศักยภาพที่มีภายในระยะเวลาที่กำหนดด้วย

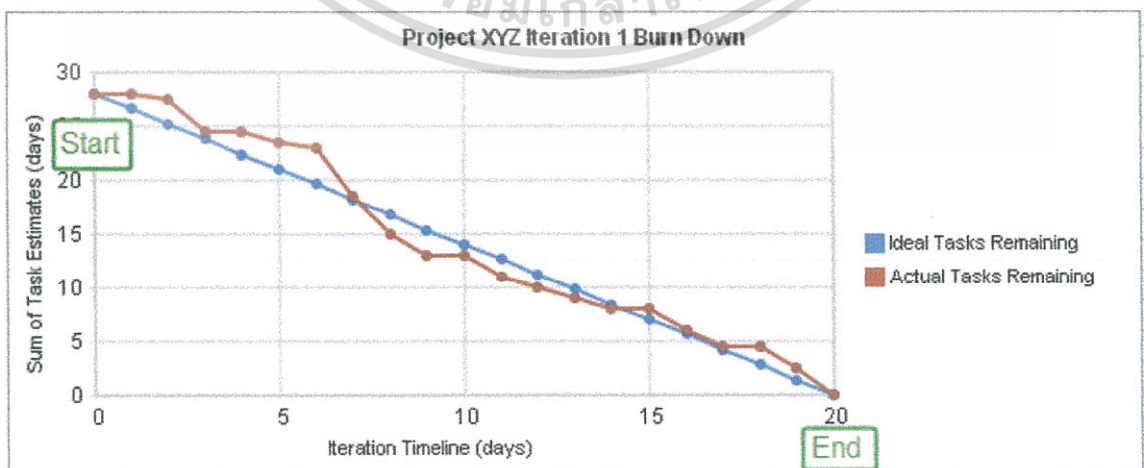
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.7.5 การประชุม daily scrum ภายในองค์กร

2.7.3 การประเมินผลและติดตามผลงาน หรือ Demonstration and Evaluation

จุดเด่นของรูปแบบการพัฒนาโปรแกรมแบบ Scrum คือการที่เราสามารถวัดผลการทำงานของเราได้ง่ายและมีประสิทธิภาพเป็นอย่างมาก โดยในรูปแบบการพัฒนาโปรแกรมแบบ Scrum จะมีกราฟแสดงงานที่ทีมได้ทำสำเร็จไปแล้ว เรียกว่า กราฟ Burn-Down Chart ที่มีหลักการ คือ แกนตั้งของกราฟจะแสดงถึงงานทั้งหมดภายใน Sprint ส่วนแกนนอนจะแสดงถึงจำนวนวันที่เหลือภายใน Sprint โดยในการประชุม Daily Scrum แต่ละวัน จะมีการอัปเดตกราฟเพื่อให้ทราบถึงงานที่ทำไปแล้ว, งานที่ยังเหลือ, เวลาที่เหลือภายใน Sprint และเวลาจริงที่ใช้ในการทำงานมากหรือน้อยกว่าเวลาที่คาดคะเนกันไว้ เมื่อครบกำหนด Sprint ก็จะมีการนำกราฟ Burn-Down Chart มาเพื่อวิเคราะห์และประเมินผลการทำงานกันอีกครั้ง

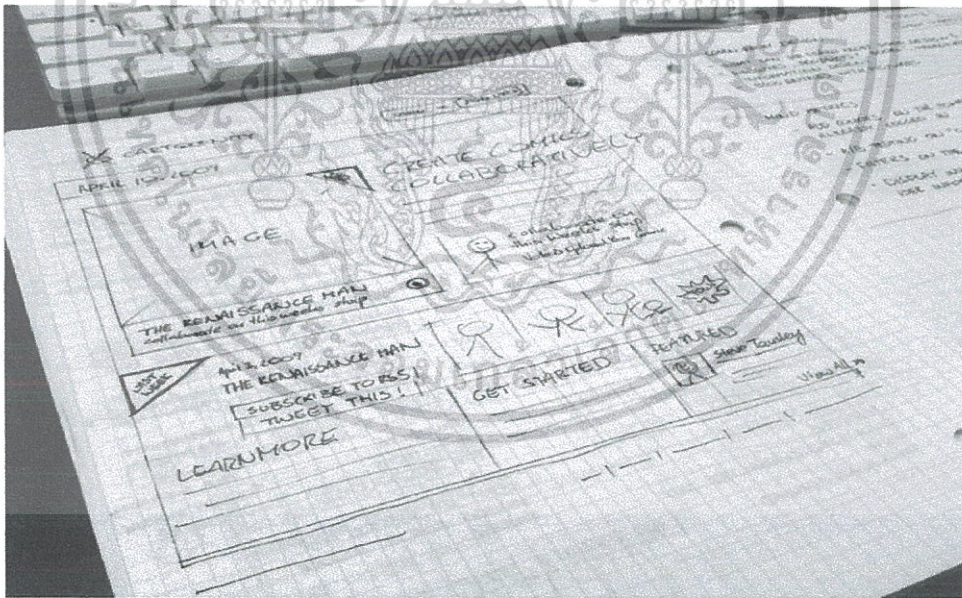


ภาพที่ 2.7.6 ตัวอย่างกราฟ Burn-Down Chart

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเขียนเพื่อใช้เท่านั้น ไม่อนุญาตให้เผยแพร่หรือนำไปใช้ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 แบบร่างโปรแกรม หรือ Wireframe

ในการพัฒนาโปรแกรม, แอปพลิเคชัน, หรือเว็บไซต์ มักจะมีข้อมูลและรายละเอียดที่ซับซ้อนหลายจุด เพื่อเป็นการสื่อสารให้เข้าใจตรงกันระหว่างผู้ใช้และทีมพัฒนา จะต้องมีการอธิบายให้เข้าใจตรงกันว่าแต่ละหน้าของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์จะมีรายละเอียดอะไรบ้าง ข้อมูลแต่ละอย่าง จะอยู่ตรงส่วนใดของหน้าโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ ซึ่งการทำแบบร่างของโปรแกรมหรือ Wireframe จะเป็นการนำรายละเอียดต่างๆมาจัดเรียงบนหน้าโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ โดยการทำให้ Wireframe ไม่จำเป็นต้องเขียนโปรแกรมเพื่อแสดงรายละเอียดต่างๆออกมา แต่สามารถทำได้หลายวิธี เช่น วาดลงบนกระดาษ, สร้าง Wireframe ด้วยโปรแกรมตัดต่อภาพ (เช่น Photoshop, Sketch, Illustrator เป็นต้น), สร้าง Wireframe ด้วยเว็บไซต์ (เช่น www.justinmind.com, www.pencil.evolus.vn เป็นต้น) หรือใช้โปรแกรมในการสร้าง Wireframe โดยเฉพาะ (เช่น โปรแกรม pencil เป็นต้น) ในยุคแรกเริ่มของการทำ Wireframe มักนิยมใช้วิธีวาดลงกระดาษ เพราะใช้เวลาไม่มาก และสามารถทำให้ผู้ใช้พอจะเข้าใจว่าตัวโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์จะมีรายละเอียดอะไรบ้าง และสามารถทำอะไรได้บ้าง นอกจากนี้การวาด Wireframe ลงบนกระดาษยังถือเป็นวิธีที่ประหยัดที่สุดด้วย เพราะโปรแกรมสำหรับทำ Wireframe โดยเฉพาะหรือเว็บสำหรับช่วยทำ Wireframe มักจะเก็บค่าบริการ หรืออาจต้องซื้อโปรแกรมหรือแพ็คเกจเสริมจากทางผู้ให้บริการ เพื่อให้ใช้งานตัวโปรแกรมหรือเว็บไซต์ได้อย่างเต็มประสิทธิภาพที่สุด

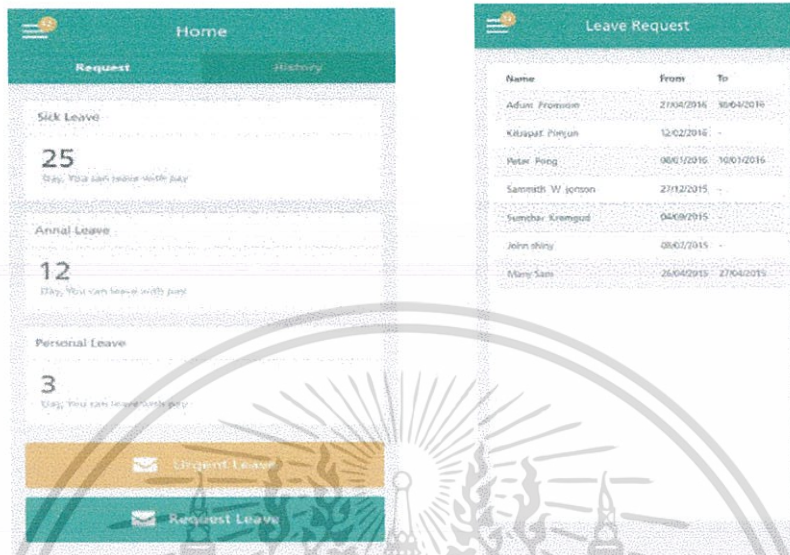


ภาพที่ 2.8.1 ตัวอย่าง Wireframe ด้วยวิธีการวาด

ที่มา : <http://www.designil.com/wp-content/uploads/2013/12/cartoonity-wireframe.jpg>

ในยุคปัจจุบันที่มีอินเทอร์เน็ตเข้ามาช่วยให้ชีวิตง่ายขึ้น การทำ Wireframe สำหรับภายในองค์กรมักนิยมใช้เว็บไซต์สำหรับช่วยทำ Wireframe เพราะหลังจากการทำ Wireframe เสร็จจำเป็นจะต้องนำเสนอต่อผู้ใช้ การถือกระดาษที่วาด Wireframe ไปนำเสนอต่อผู้ใช้จึงเป็นวิธีที่ไม่เหมาะสมอีกเอกสารต่อไป เมื่อปัจจุบันเก็บเว็บไซต์สำหรับช่วยทำ Wireframe สามารถนึกถึงคีย์บนที่ก Wireframe ไว้แล้วส่งต่อให้ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้จำนวนมากได้อย่างรวดเร็ว ถึงแม้จะมีค่าใช้จ่ายที่เพิ่มขึ้นเล็กน้อย แต่ในหลายๆองค์กรก็นิยมใช้วิธีนี้ เพราะเป็นวิธีที่สะดวก รวดเร็ว และได้ผลลัพธ์ที่ดีกว่า



ภาพที่ 2.8.2 Wireframe ของแอปพลิเคชัน Leave Request ที่ทำด้วยโปรแกรม Photoshop

จะเห็นแล้วว่าในปัจจุบันการทำ Wireframe เป็นเรื่องที่ยากขึ้นและจำเป็นอย่างมากในการสื่อสารให้ผู้ใช้และทีมพัฒนาเข้าใจตรงกัน โดย Wireframe ที่ดีนั้นจำเป็นจะต้องมีหลักในการทำ Wireframe ด้วยกัน 4 ข้อ ดังนี้

- ในขั้นตอนของการวาด Wireframe เป็นขั้นตอนที่สำคัญมาก ควรให้ทุกคนที่มีส่วนร่วมในการพัฒนาโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์นั้นๆมีส่วนร่วมในการร่างแบบขึ้นมา หากมีจุดใดที่ควรแก้ไขหรือปรับปรุง จะทำได้ง่ายในขั้นตอนนี้ หากข้ามขั้นตอนนี้ไปจนถึงขั้นตอนการออกแบบส่วนติดต่อผู้ใช้แล้ว และต้องแก้ไขอาจต้องทำการออกแบบหน้าส่วนติดต่อผู้ใช้กันใหม่ ซึ่งเป็นสิ่งที่ทำให้เสียเวลาไปโดยไม่สมควร ดังนั้นในขั้นตอนการวาดแบบร่าง Wireframe จึงควรให้ทุกคนในทีมพัฒนาเข้ามามีส่วนร่วมช่วยกันพิจารณา หากเป็นไปได้ให้ผู้ใช้เข้าร่วมด้วยจะเป็นประโยชน์อย่างมาก
- ควรกำหนดขนาดหน้าจอของอุปกรณ์ที่เป็นเป้าหมายหลักของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์นั้นๆ เพื่อที่เวลาออกแบบและในขั้นตอนการพัฒนาโปรแกรมจะไม่เกิดข้อผิดพลาดของสัดส่วนขนาดหน้าจอ เพราะหากเริ่มพัฒนาโปรแกรมไปแล้ว พบว่าการวางรูปแบบปุ่มหรือช่องรับค่าต่างๆภายในโปรแกรมผิดเพี้ยนไปจากตอนออกแบบหรือตอนวาดแบบร่าง Wireframe จะทำให้ต้องเริ่มออกแบบกันใหม่หรืออาจจะต้องกลับไปวาดแบบร่าง Wireframe กันใหม่เลย
- สิ่งสำคัญสำหรับแบบร่าง Wireframe คือสามารถสื่อสารออกมาได้พอสังเขปว่าโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์นี้สามารถทำอะไรได้ แสดงผลอะไรได้ ส่งต่อไปยังหน้าใดได้ เพราะฉะนั้นหากเริ่มวาดแบบร่าง Wireframe แล้วไม่สามารถวาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมาได้สวยงามชัดเจน ก็ควรวาดออกมาให้ผู้ที่มาเห็น Wireframe เข้าใจได้ว่า โปรแกรม, แอปพลิเคชัน หรือเว็บไซต์นี้สามารถแสดงหรือรับค่าใดได้บ้างในหน้านั้นๆ

- ในการออกแบบ Wireframe นั้นควรวาดออกมาแค่พอสังเขปว่าโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์นั้นๆสามารถทำอะไรได้บ้าง เพราะจะใช้เวลาในการวาดไม่นาน แต่สิ่งที่ควรทำอีกอย่างคือ ควรออกแบบ Wireframe มากกว่า 1 แบบ เพื่อให้คนในทีมมีตัวเลือกมาก หรือสามารถก่อเกิดความคิดใหม่ๆเพื่อใช้ในการออกแบบขั้นตอนต่อไปได้

หากออกแบบ Wireframe ได้ตามหลักทั้ง 4 ข้อดังกล่าว จึงจะถือว่า Wireframe ที่ถูกออกแบบมานั้นเป็น Wireframe ที่ดีและสามารถนำไปใช้งานได้จริง รวมถึงยังช่วยแบ่งเบาภาระของทีมออกแบบ เพราะอีกหนึ่งปัญหาสำคัญของการพัฒนาโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์นั้นก็คือ การที่ทีมออกแบบมา แต่ทีมพัฒนาไม่สามารถพัฒนาออกมาได้ตามที่ออกแบบ การทำ Wireframe ที่ดีจะช่วยให้ทีมออกแบบและทีมพัฒนาได้ทำการตกลงกันอย่างเห็นภาพก่อนว่า สิ่งใดสามารถทำได้จริง สิ่งใดไม่สามารถทำได้ และจะไม่จำเป็นต้องมาแก้ไขการออกแบบใหม่หมดเหมือนในอดีตที่ผ่านมา



ภาพที่ 2.8.3 ภาพรวม Wireframe ของแอปพลิเคชัน Leave Request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 แอปพลิเคชันตัวทดลอง หรือ Prototype (Mockup)

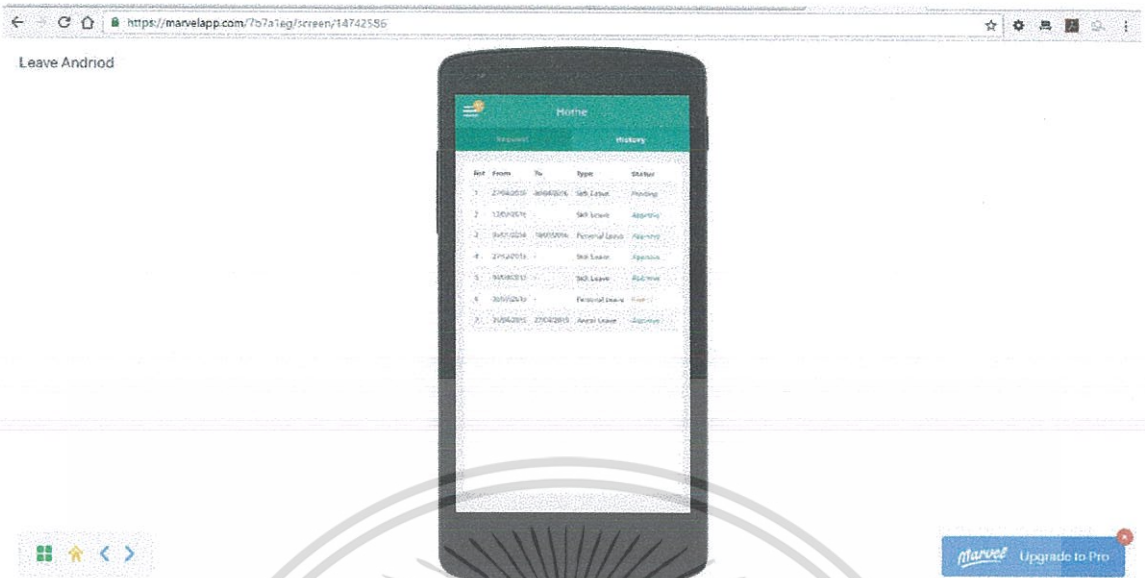
หลังจากออกแบบวาด Wireframe เสร็จสิ้นแล้ว แม้จะช่วยให้ติดต่อสื่อสารระหว่างทีมพัฒนาทีมออกแบบ และผู้ใช้หรือผู้จ้างวาน เข้าใจตรงกันแล้วว่าโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์จะมีรายละเอียดหลักๆอย่างไรบ้าง แต่ก็ยังไม่สามารถทำให้ผู้ใช้เข้าใจถึงการใช้งานโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ ได้อย่างแท้จริง จึงเป็นหน้าที่ของทีมออกแบบที่จะต้องออกแบบและลงรายละเอียด Wireframe ให้มีรายละเอียดที่ชัดเจนมากยิ่งขึ้นจนเกิดเป็น Prototype หรือ Mockup ที่นอกจากจะมีรายละเอียดของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ที่ชัดเจนขึ้นแล้ว ยังช่วยให้ผู้ใช้เข้าใจการใช้งานของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์มากขึ้นด้วย เพราะตัว Prototype เปรียบเสมือนตัวต้นแบบของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ ที่มีการตอบสนองต่อการใช้งานของผู้ใช้ เช่น กดปุ่มเปลี่ยนหน้าแล้วหน้าแอปพลิเคชันก็เปลี่ยนหน้า, กดไฮเปอร์ลิงค์บนเว็บไซต์แล้วเปิดหน้าเว็บไซต์ใหม่ขึ้นมา เป็นต้น ซึ่งหลังจากออกแบบ Prototype เสร็จสิ้นแล้วจะต้องมีการนำไปทดสอบกับกลุ่มผู้ใช้ว่าผู้ใช้เข้าใจการออกแบบ สามารถใช้งานโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ได้จริง หากผู้ใช้อยู่รู้สึกว่าโปรแกรม, แอปพลิเคชัน, เว็บไซต์นี้ยังใช้งานยากอยู่ อาจจะต้องย้อนกลับไปออกแบบกันใหม่

การออกแบบ Prototype และนำไปทดสอบกับกลุ่มผู้ใช้งานก่อน ช่วยให้ทีมออกแบบออกแบบหน้าตาของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ได้ตรงตามความต้องการของผู้ใช้มากขึ้น รวมถึงทีมพัฒนา ก็สามารถพัฒนาโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ได้อย่างถูกต้อง ไม่ต้องแก้ไขในส่วนหน้าติดต่อผู้ใช้บ่อยๆ ซึ่งจะทำให้การพัฒนาเป็นไปอย่างล่าช้าและมีความซับซ้อนเป็นอย่างมาก

การออกแบบ Prototype สามารถทำได้หลายวิธี ไม่ว่าจะเป็นการออกแบบโดยใช้โปรแกรมบนคอมพิวเตอร์, ออกแบบโดยใช้แอปพลิเคชันบนสมาร์ตโฟนที่มีกันอย่างแพร่หลายในปัจจุบัน, เว็บไซต์สำหรับออกแบบ Prototype หรือแม้แต่การพัฒนาโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์แบบมีการตอบสนองต่อการใช้งานของผู้ใช้ ออกมาก็สามารถทำได้ แต่ที่นิยมกันมากที่สุดในปัจจุบันสำหรับทีมออกแบบคือ การออกแบบโดยใช้โปรแกรมสำหรับออกแบบ Prototype และการออกแบบโดยใช้เว็บไซต์สำหรับออกแบบ Prototype เพราะมีให้เลือกใช้กันอย่างแพร่หลาย และสามารถส่งต่อระหว่างทีมออกแบบด้วยกัน, ส่งต่อไปยังทีมพัฒนา หรือแม้แต่ส่งไปให้ผู้ใช้ได้ทดลองใช้ก็สามารถทำได้อย่างสะดวกรวดเร็ว โดยเฉพาะการออกแบบด้วยเว็บไซต์สำหรับออกแบบ Prototype เช่น เว็บไซต์ www.marvelapp.com เป็นต้น มักจะมีฟีเจอร์ที่อนุญาตให้สามารถเผยแพร่ชิ้นงาน Prototype ที่ออกแบบไปยังสมาชิกในทีม หรือใครก็ตามได้ตามต้องการ โดยสามารถกำหนดได้ว่าจะส่งต่อไปยังใคร หากใครจะเข้าใช้ชิ้นงานนี้แต่ไม่ได้รับอนุญาตโดยเจ้าของชิ้นงานจะไม่สามารถเข้าชมหรือเข้าใช้ชิ้นงานนี้ได้ ซึ่งก็ถือเป็นความปลอดภัยอย่างหนึ่งที่เหมาะสม

สำหรับการออกแบบ Prototype ที่ดีจะมีหลักการออกแบบเหมือนกันกับ Wireframe เพราะตัว Prototype คือการออกแบบต่อยอดต่อจาก Wireframe ที่ในปัจจุบันมักจะออกแบบต่อยอดกันหรืออาจออกแบบควบคู่กันมาได้เลย ตามแต่ความชำนาญของผู้ที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

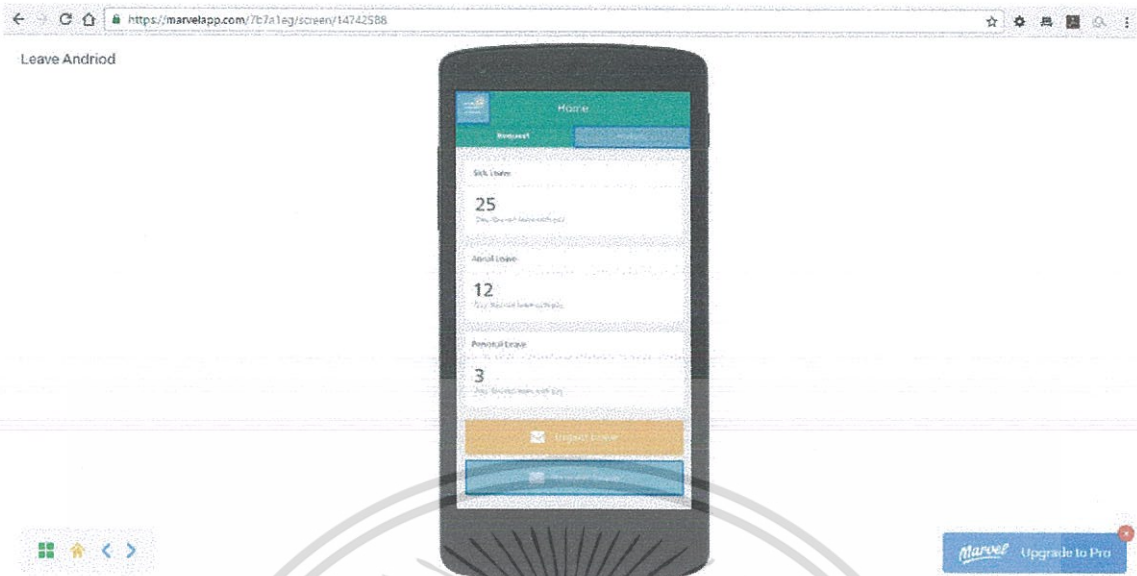


ภาพที่ 2.9.1 Prototype ของแอปพลิเคชัน Leave Request ที่ออกแบบโดยใช้เครื่องมือจากเว็บไซต์ www.marvelapp.com



ภาพที่ 2.9.2 ภาพรวม Prototype ของแอปพลิเคชัน Leave Request ที่ออกแบบโดยใช้เครื่องมือจากเว็บไซต์ www.marvelapp.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.9.3 แสดงถึงจุดที่สามารถใช้งานแล้วมีการตอบสนองของตัว Prototype ของแอปพลิเคชัน Leave Request ที่ออกแบบโดยใช้เครื่องจากเว็บไซต์ www.marvelapp.com

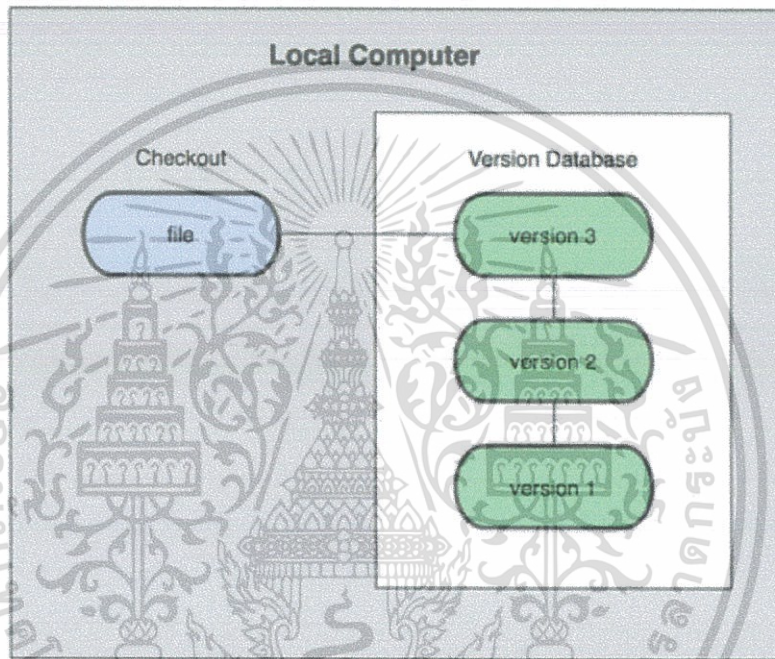
2.10 Git

ในการพัฒนาโปรแกรมแต่ละครั้ง มักมีการพัฒนาโปรแกรมที่ซับซ้อน และต้องทำต่อเนื่อง ไม่สามารถทำสำเร็จได้ภายในวันเดียวหรือการเขียนโปรแกรมเพียงครั้งเดียว ทำให้ต้องมีการบันทึกไฟล์ลงบนเครื่องของผู้ที่ทำการเขียนโค้ดหรือโปรแกรมนั้นขึ้นมา เมื่อพัฒนาโปรแกรมที่มีขนาดใหญ่มากขึ้นหรือมีความซับซ้อนมากขึ้นจนไม่สามารถพัฒนาด้วยตัวคนเดียวได้ จำเป็นต้องพัฒนาร่วมกันเป็นทีม แต่เกิดปัญหาในการกระจายโค้ดหรือไฟล์โปรแกรมที่กำลังพัฒนาไปยังนักพัฒนาโปรแกรมคนอื่นๆ โดยวิธีที่ง่ายที่สุดในอดีตคือการคัดลอกไฟล์โปรแกรมที่กำลังพัฒนาไปยังเครื่องคอมพิวเตอร์ของนักพัฒนาคนอื่นๆ โดยใช้ Thumb Drive หรือในยุครุ่นอินเทอร์เน็ตเริ่มแพร่หลาย อาจมีการอัปโหลดขึ้นเว็บไซต์สำหรับอัปโหลดไฟล์เพื่อให้นักพัฒนาคนอื่นๆ มาดาวน์โหลดไฟล์โปรแกรมไปพัฒนากันต่อ และถ้าหากจะส่งไฟล์ที่พัฒนาเสร็จสมบูรณ์แล้วกลับมาก็ต้องใช้วิธีเดิม ซึ่งยังไม่รวดเร็วพอ และในกรณีที่มีการพัฒนาไปแล้วเกิดอุบัติเหตุส่งผลให้ไฟล์ที่พัฒนาที่ผ่านมาหายไปจากเครื่องคอมพิวเตอร์ของนักพัฒนา ซึ่งเป็นปัญหาใหญ่มากสำหรับทีมพัฒนา ทำให้ในยุคต่อมาได้มีการคิดค้นและพัฒนาระบบที่เรียกว่า Git ขึ้นมาโดยมีพื้นฐานหลักของ Git คือการช่วยให้ทีมพัฒนาสามารถร่วมกันพัฒนาโปรแกรมได้อย่างสะดวกสบายมากขึ้น และสามารถทำการสำรองไฟล์ของชิ้นงานที่กำลังพัฒนาไว้ได้ ป้องกันกรณีเกิดอุบัติเหตุไฟล์ในคอมพิวเตอร์สูญหาย

ก่อนหน้าจะมี Git ให้ใช้นั้น ได้มีการคิดค้นระบบที่จัดการการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อที่คุณสามารถเรียกเวอร์ชันใดเวอร์ชันหนึ่งกลับมาดูเมื่อต้องการที่ถูกเรียกว่า Version Control โดยระบบนี้ได้ถูกพัฒนาต่อเนื่องมาเรื่อยๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

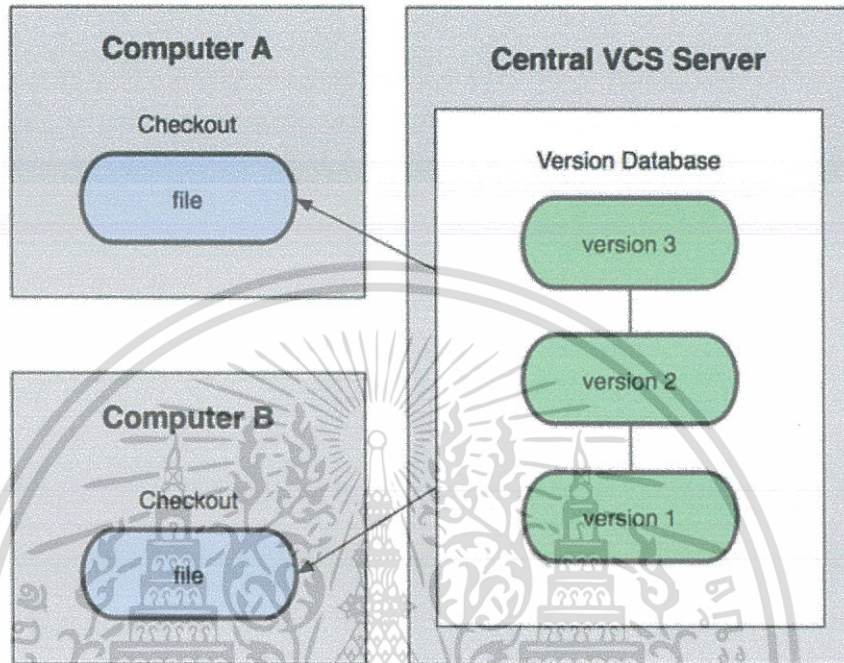
- Version Control Systems แบบ Local เป็นยุคแรกที่มีการสำรองไฟล์ไว้ในเครื่องคอมพิวเตอร์เครื่องที่ทำการแก้ไขไฟล์นั้นๆ เพื่อป้องกันกรณีต้องกลับมาแก้ไขไฟล์ในเวอร์ชันก่อนๆ ด้วยการบันทึกไฟล์เวอร์ชันใหม่ด้วยไดเรกทอรีใหม่ โดยอาจมีการตั้งชื่อเป็นวันเวลาที่ทำการสำรองไฟล์นั้นๆ เพราะถือเป็นวิธีที่ง่ายที่สุดเท่าที่จะทำได้ในยุคนั้น แต่ก็เกิดปัญหาใหม่คือ สิ้นเปลืองพื้นที่เก็บข้อมูลภายในคอมพิวเตอร์เป็นอย่างมาก จึงเกิดการพัฒนาระบบ Version Control Systems แบบรวมศูนย์ขึ้น



ภาพที่ 2.10.1 Version Control Systems แบบ Local
ที่มา: <https://git-scm.com/figures/18333fig0101-tn.png>

- Version Control Systems แบบรวมศูนย์ ถูกพัฒนาเพื่อแก้ปัญหาเรื่องความสิ้นเปลืองพื้นที่จัดเก็บข้อมูลภายในคอมพิวเตอร์และยังเข้ามาช่วยในเรื่องการพัฒนาซอฟต์แวร์แบบเป็นทีมอีกด้วย โดยระบบนี้จะมีเซิร์ฟเวอร์กลางที่เก็บไฟล์ทั้งหมดไว้ในที่ที่เดียวและผู้ใช้หลายๆคนสามารถเชื่อมต่อเข้ามาเพื่อดาวน์โหลดไฟล์จากเซิร์ฟเวอร์ศูนย์กลางนี้ไปแก้ไขได้ นอกจากนี้นักพัฒนาทุกคนในทีมสามารถรู้ได้ว่านักพัฒนาคนอื่นในทีมกำลังทำอะไร ผู้ควบคุมระบบสามารถควบคุมได้อย่างละเอียดว่าใครสามารถแก้ไขอะไรได้บ้าง การจัดการแบบรวมศูนย์ในที่ที่เดียวทำได้ง่ายกว่าการจัดการฐานข้อมูลบนคอมพิวเตอร์แต่ละเครื่องเป็นอย่างมาก แต่ก็ยังคงมีปัญหาคือถ้าหากเซิร์ฟเวอร์ศูนย์กลางที่มีเพียงตัวเดียวล่มจะไม่สามารถเข้าไปทำอะไรกับไฟล์บนเซิร์ฟเวอร์ได้เลย หรือถ้าหากฮาร์ดดิสก์ของเซิร์ฟเวอร์ศูนย์กลางเกิดชำรุดเสียหาย เท่ากับว่าไฟล์ที่สำรองไว้บนเซิร์ฟเวอร์ทั้งหมดก็ชำรุดเสียหาย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

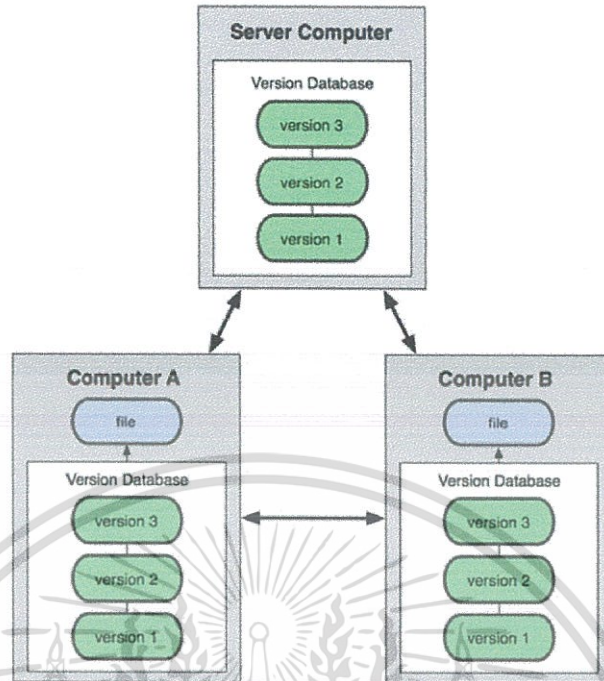
โดยไม่มีการสำรองไว้ที่อื่นอีกด้วย นอกจากไฟล์ที่อยู่บนเครื่องคอมพิวเตอร์แต่ละเครื่องที่ทำการดาวน์โหลดไว้ก่อนหน้านี้



ภาพที่ 2.10.2 Version Control Systems แบบรวมศูนย์
ที่มา: <https://git-scm.com/figures/18333fig0102-tn.png>

- Version Control Systems แบบกระจายศูนย์ เป็นระบบที่เข้ามาแก้ปัญหาที่เกิดขึ้นในระบบแบบรวมศูนย์ โดยระบบแบบกระจายศูนย์นี้ เมื่อมีการดาวน์โหลดไฟล์จากเซิร์ฟเวอร์สู่เครื่องคอมพิวเตอร์จะไม่ได้มีการดาวน์โหลดมาเพียงไฟล์เวอร์ชันล่าสุดเท่านั้น แต่ยังรวมถึงเวอร์ชันทั้งหมดที่ผ่านมาของไฟล์นั้นเข้ามายังเครื่องด้วย ทำให้ต่อให้เซิร์ฟเวอร์ล่มนักพัฒนา ก็ยังสามารถแก้ไขไฟล์ในเวอร์ชันต่างๆ ได้จากไฟล์ที่ถูกดาวน์โหลดมาในเครื่องคอมพิวเตอร์แล้ว และเมื่อเซิร์ฟเวอร์กลับมาใช้งานได้ ก็สามารถอัปเดตไฟล์ที่แก้ไขแล้วกลับไปยังเซิร์ฟเวอร์ได้ นอกจากนี้ในระบบแบบกระจายศูนย์ยังช่วยให้นักพัฒนาสามารถดาวน์โหลดไฟล์จากหลายๆ โปรเจกต์มาทำได้ ทำให้สามารถพัฒนาและสำรองไฟล์ได้หลายโปรเจกต์ในเวลาเดียวกัน และ Git ถือเป็นระบบ Version Control Systems แบบกระจายศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



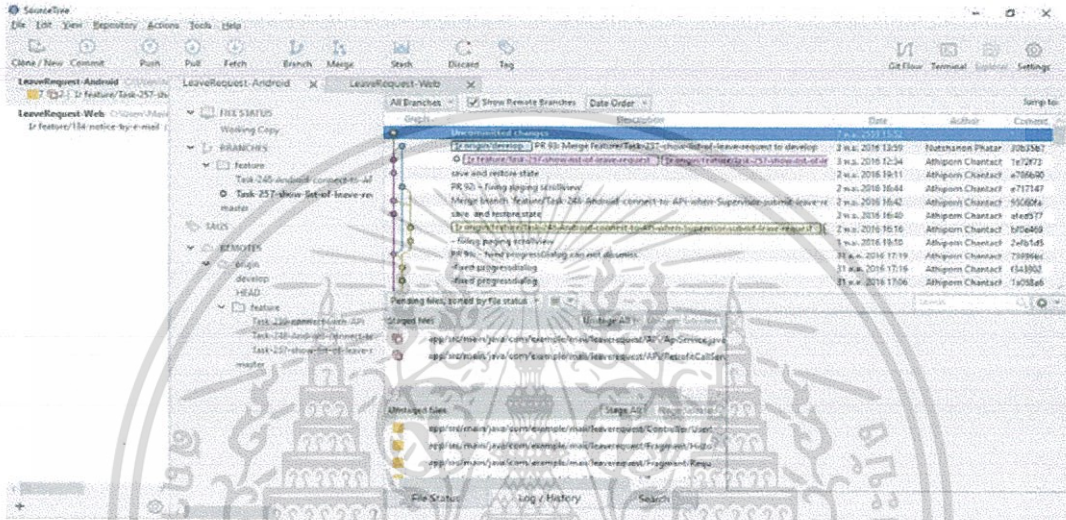
ภาพที่ 2.10.3 Version Control Systems แบบกระจายศูนย์
ที่มา: <https://git-scm.com/figures/18333fig0103-tn.png>

Git เป็นระบบที่มีการจัดการและบันทึกการเปลี่ยนแปลงของไฟล์ที่พัฒนาโปรแกรม โดยจะมีการสำรองไฟล์ที่พัฒนาโปรแกรมลงบนเซิร์ฟเวอร์ของระบบ ที่ช่วยให้นักพัฒนาสามารถที่จะเรียกดูหรือย้อนกลับไปแก้ไขเวอร์ชันต่างๆของโปรเจกต์ที่ใด เวลาใดก็ได้ หรือแม้แต่ว่าไฟล์นั้นๆใครทำการเพิ่มหรือแก้ไข หรือว่าจะดูว่าไฟล์นั้นๆถูกเขียนโดยนักพัฒนาคนใดก็สามารถทำได้

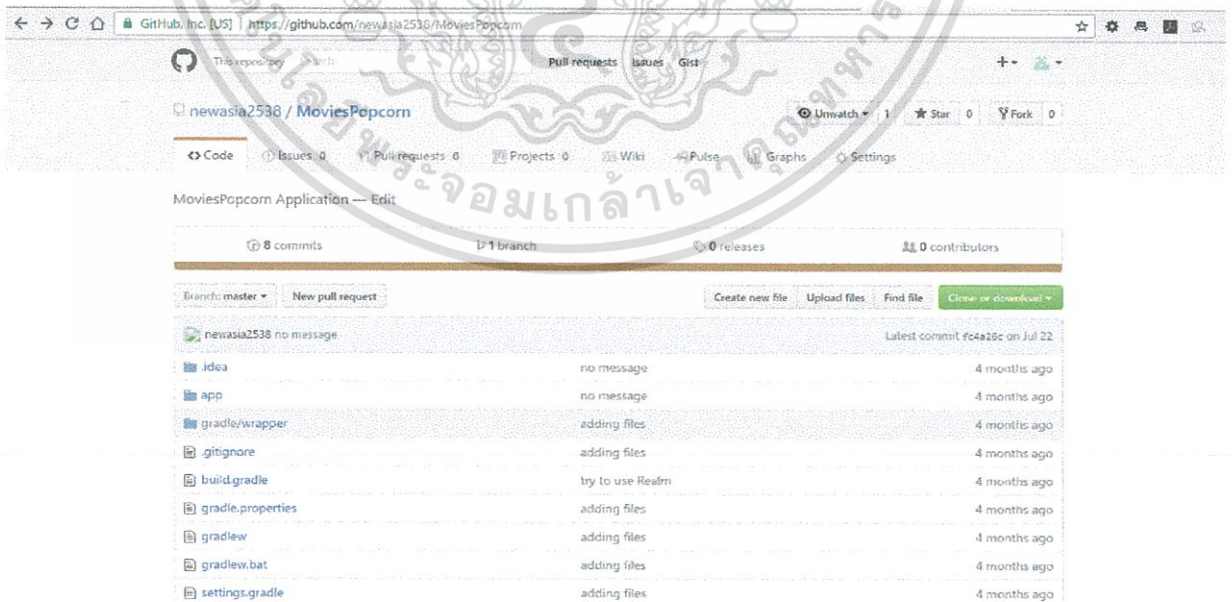
ในกรณีที่มีการแก้ไขไฟล์เดียวกันหรือโปรเจกต์เดียวกันโดยนักพัฒนาหลายคน มักจะเกิดปัญหาคือมีการเปลี่ยนแปลงและแก้ไขโค้ด แตกต่างกันไปเมื่อจะนำไฟล์หรือโปรเจกต์ที่มีการแก้ไขร่วมกันนี้ไปใช้งานจะต้องนำแต่ละส่วนที่นักพัฒนาแต่ละคนแก้ไขไปนั้นมารวมกันด้วยตัวเอง ถ้าหากทำด้วยตัวเองแล้วจะเป็นการยากมาก เพราะนักพัฒนาจะต้องไล่หาจุดที่แก้ไขเอง ซึ่งจะเสียเวลาตรงส่วนนี้ไปโดยใช้เหตุ Git เข้ามาแก้ไขปัญหานี้โดย Git จะทำการรวมโค้ดที่แก้ไขในไฟล์เดียวกันให้โดยอัตโนมัติเมื่อนักพัฒนาทำการส่งไฟล์ที่ร่วมกันแก้ไขเข้าสู่ระบบ หากไฟล์ที่ทำการแก้ไขไม่มีการทับซ้อนกันของโค้ดที่แก้ไข หรือไม่มีปัญหาใดๆเลย Git จะทำการรวมไฟล์ให้เสร็จสมบูรณ์ แต่ถ้าหากไฟล์ที่ส่งเข้าสู่ระบบนั้นมีการทับซ้อนกันของโค้ด หรือมีการแก้ไขไฟล์ในจุดๆเดียวกัน โดยนักพัฒนามากกว่าหนึ่งคน Git จะมีการแจ้งเตือนว่าการรวมไฟล์นี้จะไม่สามารถทำได้โดยอัตโนมัติ จำเป็นต้องให้นักพัฒนาจัดการรวมไฟล์ในส่วนนี้ด้วยตนเอง ซึ่งก็ถือว่าช่วยลดเวลาไปได้มากเช่นกัน

สำหรับการใช้งาน Git ในปัจจุบันสามารถทำได้โดยใช้งานผ่าน Command Prompt ของระบบปฏิบัติการ Window หรือผ่าน Terminal ของระบบปฏิบัติการ Mac OS นอกจากนี้ยังสามารถใช้ผ่านตัวโปรแกรมประเภท GUI (Git User Interface) เช่น โปรแกรม Source Tree เป็นต้น การใช้งาน Git ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่จำเป็นต้องใช้อินเทอร์เน็ตทุกคำสั่ง นักพัฒนาสามารถแก้ไขไฟล์และทำการ commit ไฟล์นั้นเพื่อให้ Git จัดจำไว้ว่าไฟล์นี้มีการเปลี่ยนแปลงเกิดขึ้น หรือย้อนกลับไปยัง commit ที่ผ่านมานักพัฒนาสามารถใช้งานได้โดยไม่ต้องต่ออินเทอร์เน็ต แต่การส่งไฟล์เข้าสู่ระบบ (push) หรือการดึงไฟล์จากระบบเข้าสู่เครื่องคอมพิวเตอร์ (pull) คำสั่งพวกนี้จำเป็นต้องต่ออินเทอร์เน็ต เพราะมีการติดต่อกับทางเซิร์ฟเวอร์ของระบบ



ภาพที่ 2.10.4 โปรแกรม Source Tree



ภาพที่ 2.10.5 ตัวอย่าง เว็บไซต์ www.github.com เว็บไซต์สำหรับใช้งาน Git ที่ได้รับความนิยมในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 ภาษา XML

ภาษา XML หรือ ภาษา Extensible Markup Language เป็นภาษาที่ใช้ในการแสดงผลข้อมูล ซึ่งภาษา XML จะมีความสามารถใกล้เคียงกับภาษา HTML แต่ภาษา XML จะมีความสามารถที่มากกว่า HTML ตรงที่ภาษา XML ไม่ใช่เพียงแค่แสดงผลข้อมูลแต่สามารถเก็บข้อมูลได้ด้วย หรือสามารถเก็บรายละเอียดของข้อมูลที่แสดงที่เรียกว่า Meta data เพราะจุดประสงค์ของภาษา XML จริงๆแล้วถูกคิดค้นมาเพื่อเก็บข้อมูล ส่วนการแสดงผลจริงๆจะใช้ภาษาเฉพาะที่เรียกว่า ภาษา XSL หรือภาษา Extensible Stylesheet Language

อาจกล่าวได้ว่าภาษา XML เป็นส่วนเสริมของภาษาต่างๆ เพราะภาษา XML ไม่สามารถแสดงผลข้อมูลได้ด้วยตัวเอง เพราะฉะนั้นหากต้องการจะแสดงผลข้อมูลจำเป็นต้องใช้ร่วมกับภาษาอื่นๆ เช่น HTML, PHP, JSP หรือภาษาอื่นๆที่สนับสนุน

ในปัจจุบันนอกจากมีการนำภาษา XML มาใช้ร่วมกับภาษา HTML เพื่อแสดงผลของข้อมูลและติดต่อกับผู้ใช้จากโครงสร้างของข้อมูลแล้ว ยังมีการนำมาใช้ร่วมกับภาษา Java เพื่อการแสดงผลเป็นส่วนติดต่อกับผู้ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android อีกด้วย โดยจะสามารถกำหนดรายละเอียดการแสดงผลต่างๆได้อย่างละเอียด เช่น ขนาดตัวอักษร, ตัวอักษรหนา เอียง บาง, ขนาดความกว้างของหน้าจอแอปพลิเคชัน, ซ่อนหรือแสดงเลย์เอาต์ต่างๆภายในแอปพลิเคชัน เป็นต้น ซึ่งบางอย่างสามารถกำหนดค่าได้ด้วยภาษา Java แต่หลายอย่างที่ภาษา Java ไม่สามารถทำได้ จึงจำเป็นต้องมีภาษา XML เข้ามาเป็นส่วนเติมเต็มในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="8dp"
    android:paddingLeft="8dp"
    android:paddingRight="8dp"
    android:paddingTop="16dp">

    <ScrollView
        android:id="@+id/scrollView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/requestButton"
    >

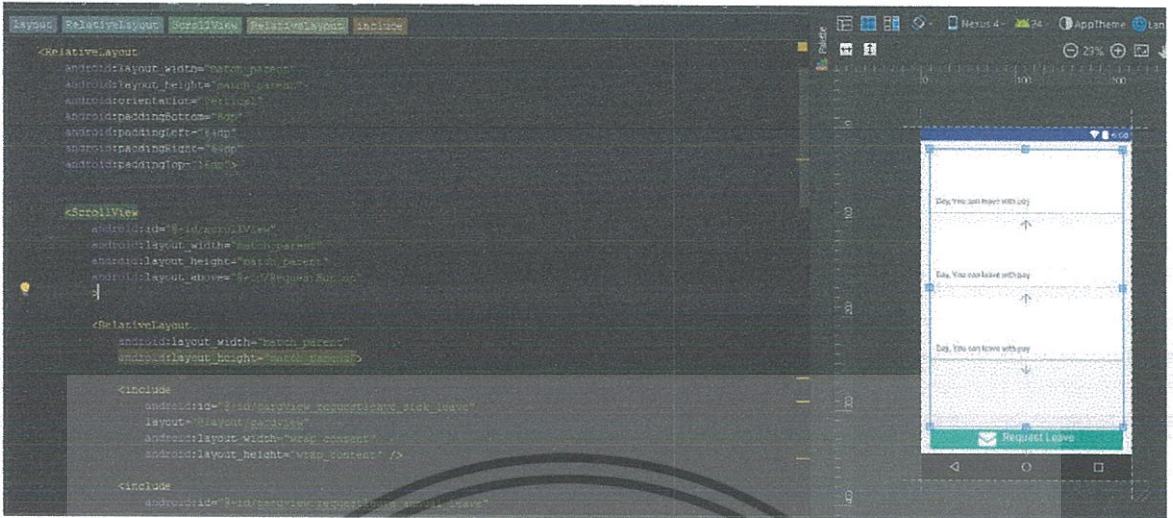
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <include
                android:id="@+id/cardview_request_leave_sick_leave"
                layout="@layout/cardview"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
    </RelativeLayout>
</RelativeLayout>

```

ภาพที่ 2.11.1 ตัวอย่างภาษา XML ที่ใช้ในการพัฒนาแอปพลิเคชัน Leave Request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.11.2 ตัวอย่างส่วนติดต่อผู้ใช้ของแอปพลิเคชัน Leave Request ที่ถูกกำหนดด้วยภาษา XML

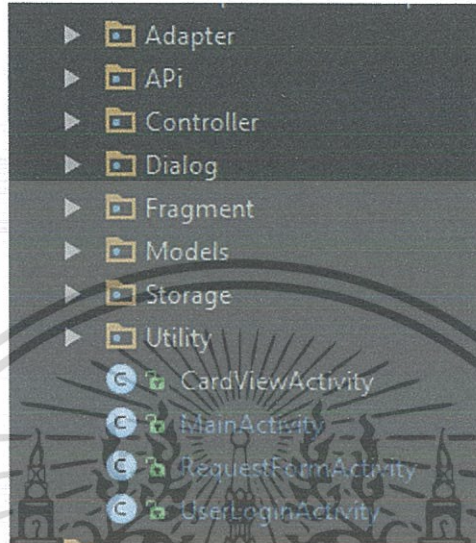
2.12 สถาปัตยกรรมซอฟต์แวร์แบบ Model-View-Controller หรือ MVC Pattern

สถาปัตยกรรมซอฟต์แวร์แบบ Model-View-Controller หรือที่นิยมเรียกกันว่า MVC Pattern หรือที่นักพัฒนามักจะนิยมเรียกกันว่า MVC เป็นโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่นิยมใช้เป็นอย่างมากในการพัฒนาซอฟต์แวร์ในปัจจุบัน เพราะโครงสร้างแบบ MVC นี้จะมีการแยกส่วนของโปรแกรมเป็น 3 ส่วนคือ ส่วน Model, View และ Controller ซึ่งทั้ง 3 ส่วนจะแบ่งแยกหน้าที่การทำงานกันอย่างชัดเจน ซึ่งช่วยให้การพัฒนา การทดสอบ และการดูแลรักษาซอฟต์แวร์เป็นไปได้อย่างง่ายดายมากขึ้น แม้ผู้ที่เข้ามาทำการพัฒนาต่อ หรือทดสอบซอฟต์แวร์นั้นๆ ก็สามารถตรวจสอบและเข้าใจโครงสร้างของซอฟต์แวร์นั้นได้ง่ายขึ้น โดยหน้าที่หลักของทั้ง 3 ส่วนนั้นมีดังนี้

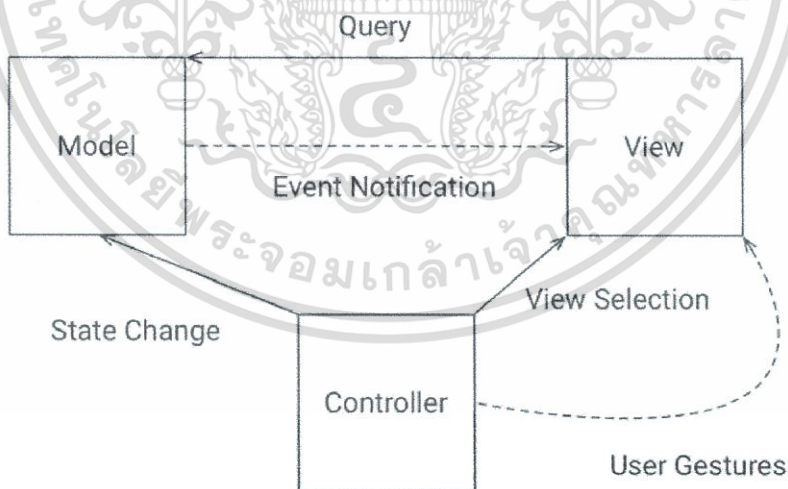
- Model เป็นส่วนของซอฟต์แวร์ที่มีไว้เพื่อทำงานร่วมกับตัวข้อมูลของซอฟต์แวร์ เช่น การเก็บข้อมูล การส่งต่อข้อมูลไปยังส่วนอื่นของโปรแกรม เป็นต้น หรือจะเปรียบได้ว่า Model เป็นเสมือนระบบฐานข้อมูลของซอฟต์แวร์นั้นๆ ทั่วๆ ไป และการจัดการเกี่ยวกับข้อมูลทุกอย่าง จะถูกจัดการในส่วนของ Model ทั้งหมด
- View เป็นส่วนของซอฟต์แวร์ที่ทำหน้าที่แสดงผลและเป็นส่วนติดต่อกับผู้ใช้ของซอฟต์แวร์นั้นๆ ไม่สามารถทำอะไรได้มากไปกว่านั้น
- Controller เป็นส่วนของซอฟต์แวร์ที่ทำหน้าที่รับค่าข้อมูลต่างๆ จาก Model มาประมวลผลจนเสร็จสิ้นสมบูรณ์ ก่อนจะส่งต่อไปแสดงผลที่ส่วนของ View หรืออาจจะรับค่ามาจากส่วน View แล้วทำหน้าที่ประมวลผล ก่อนส่งเข้าไปจัดเก็บลงในส่วนของ Model

จะเห็นแล้วว่าซอฟต์แวร์ที่ใช้โครงสร้างสถาปัตยกรรมแบบ MVC ในการพัฒนาจะทำให้ส่วนต่างๆ ของซอฟต์แวร์แบ่งหน้าที่กันอย่างชัดเจน และไม่มีการข้ามไปทำหน้าที่ของส่วนอื่นๆ ของซอฟต์แวร์ได้ แยกแยะด้วยการแบ่งแยกส่วนต่างๆ และหน้าที่กันอย่างชัดเจน ทำให้ผู้ที่ไม่เคยเห็นโค้ดที่ถูกพัฒนาขึ้นมาก่อนก็เข้าใจไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเข้าใจโครงสร้างของซอฟต์แวร์นั้นๆ ได้ง่ายขึ้นเป็นอย่างมาก และทำให้การพัฒนาต่อยอดซอฟต์แวร์ หรือการปรับปรุง ดูแลรักษา ตรวจสอบความถูกต้องของซอฟต์แวร์ก็สามารถทำได้ง่ายขึ้นเช่นกัน



ภาพที่ 2.12.1 ตัวอย่างการแบ่งส่วนของซอฟต์แวร์ตามสถาปัตยกรรมซอฟต์แวร์แบบ MVC ของแอปพลิเคชัน Leave Request



ภาพที่ 2.12.2 Workflow Diagram ของสถาปัตยกรรมซอฟต์แวร์แบบ MVC

ที่มา : https://i0.wp.com/www.arondora.in.th/wp-content/uploads/2015/11/mvc_model.png

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับซอฟต์แวร์ที่มีการใช้โครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC นั้นจะมีหลักการทำงานคล้ายๆกัน ดังนี้

1. เมื่อผู้ใช้ ทำการร้องขอ(Request) เข้ามายังซอฟต์แวร์ ซอฟต์แวร์จะส่งคำขอตกลงไปที่ Controller ก่อนเพื่อให้ Controller พิจารณาว่าต้องทำหน้าที่อะไร
2. หากคำขอที่ส่งเข้ามามีความต้องการใช้ข้อมูล Controller จะส่งหน้าที่การทำงานต่อไปยัง Model เพื่อไปหาข้อมูลภายใน Model มาให้
3. เมื่อ Controller ได้รับข้อมูลที่ได้รับการประมวลผลเพื่อนำไปใช้งานต่อแล้ว Controller จะส่งข้อมูลที่ได้รับการประมวลผลแล้วนั้นไปให้ View
4. สุดท้าย View ทำการแสดงผลให้กับผู้ใช้
5. และเมื่อผู้ใช้ทำการใช้งานซอฟต์แวร์ส่วนต่างๆต่อไป ซอฟต์แวร์ก็จะมีการทำงานตามขั้นตอนที่ 1 - 4 ทุกครั้งที่มีการส่งคำขอไปยังซอฟต์แวร์ หรือต้องติดต่อกับข้อมูลภายในซอฟต์แวร์นั้นๆ
6. กรณีที่ไม่จำเป็นต้องใช้งานข้อมูลที่เก็บอยู่ภายใน Model เมื่อ Controller รับคำขอจากผู้ใช้แล้ว Controller จะทำการประมวลผลแล้วส่งผลลัพธ์ที่ได้ไปยัง View เลย โดยไม่ต้องมีการทำงานร่วมกับส่วน Model



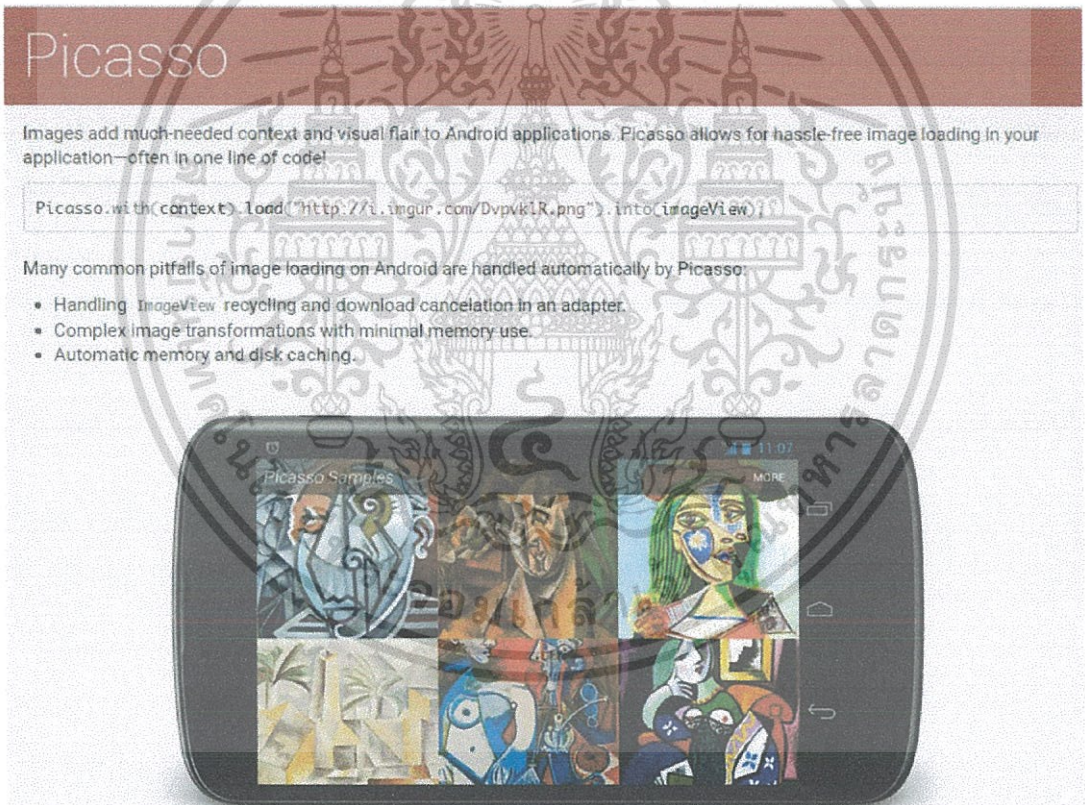
ภาพที่ 2.12.3 หลักการทำงานของซอฟต์แวร์ที่ใช้โครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC ที่มา : http://2.bp.blogspot.com/_DHQYWBAmhO4/TRQro3TSCLI/AAAAAAAAAEo/Ec472SdLcLQ/s1600/basic_mvc.png

2.13 Third party library สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android

Third party library หรือที่นักพัฒนาในประเทศไทยนิยมเรียกกันว่า “ไลบรารีนอก” คือ ส่วนเสริมหรือ Function สำเร็จรูปที่มิ้นักพัฒนาทำไว้สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android ซึ่ง Third party library ในปัจจุบันก็มีให้เลือกใช้มากมาย และ Third party library แต่ละอย่างมีความสามารถที่แตกต่างกัน โดยจะมีตั้งแต่ช่วยเสริมในเรื่องการจัดการส่วนติดต่อกับผู้ใช้, จัดการระบบฐานข้อมูลบนตัวอุปกรณ์, เรียกใช้หรือติดต่อกับระบบเซิร์ฟเวอร์ หรือการดึงข้อมูลจากไฮเปอร์ลิงค์มาแสดงบนหน้าแอปพลิเคชันก็สามารถทำได้ ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

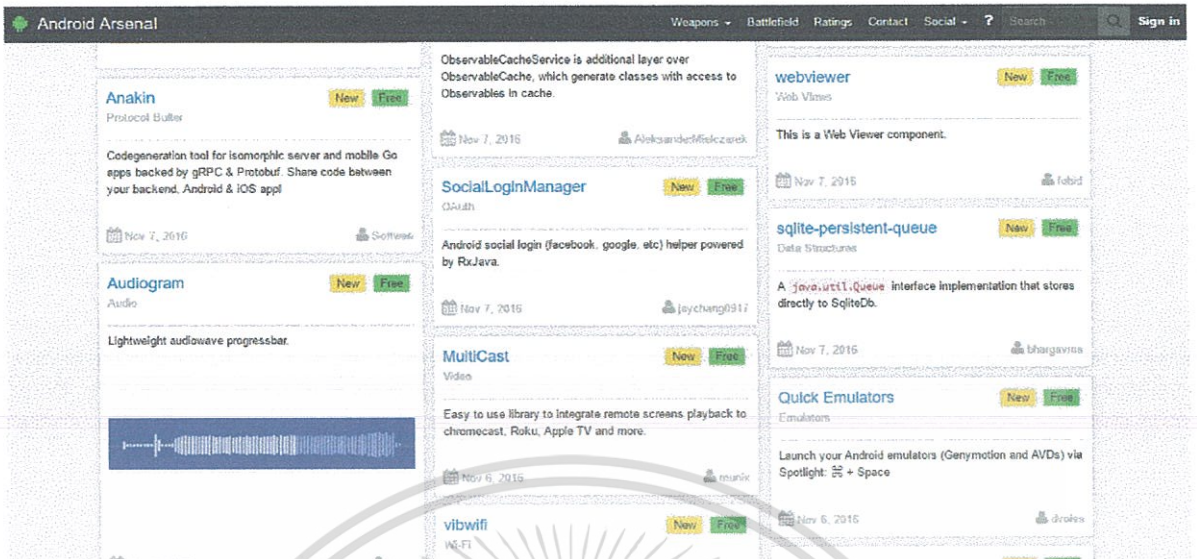
ถึงแม้บาง Function ทางโปรแกรม Android Studio จะมีมาให้ หรือบาง Function ที่นักพัฒนาสามารถเขียนได้เอง แต่ก็ต้องยอมรับว่า Third party library บางตัวที่ได้รับความนิยมมากๆ ยังคงมีประสิทธิภาพโดยรวมที่ดีกว่า สำหรับ Third party library ที่กำลังเป็นที่นิยมขณะนี้ เช่น Realm ที่ทำหน้าที่เป็นตัวจัดการระบบฐานข้อมูลที่สามารถจัดการทั้งระบบฐานข้อมูลบนเซิร์ฟเวอร์หรือสามารถจัดเก็บข้อมูลลงบนระบบฐานข้อมูลบนตัวอุปกรณ์ก็ได้, Picasso ที่ทำหน้าที่ดึงข้อมูลจากไฮเปอร์ลิงค์, เซิร์ฟเวอร์ของระบบฐานข้อมูล หรือแม้แต่ข้อมูลบนอุปกรณ์ที่เป็นข้อมูลประเภทรูปภาพมาแสดงบนหน้าแอปพลิเคชันได้ เป็นต้น

สำหรับการหา Third party library มาใช้งานในปัจจุบันนั้นสามารถหาได้ง่ายๆตามเว็บไซต์ต่างๆเช่น www.android-arsenal.com หรือหาได้จากแอปพลิเคชันที่รวบรวม Third party library ไว้ให้แล้ว เช่น libraries for developers ก็มีให้เลือกใช้มากมาย



ภาพที่ 2.13.1 ตัวอย่าง Third party library “Picasso”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.13.2 ตัวอย่างหน้าเว็บไซต์ www.android-arsenal.com ที่เป็นแหล่งรวบรวม Third party library

2.14 การออกแบบ User Experience และ User Interface (UX/UI)

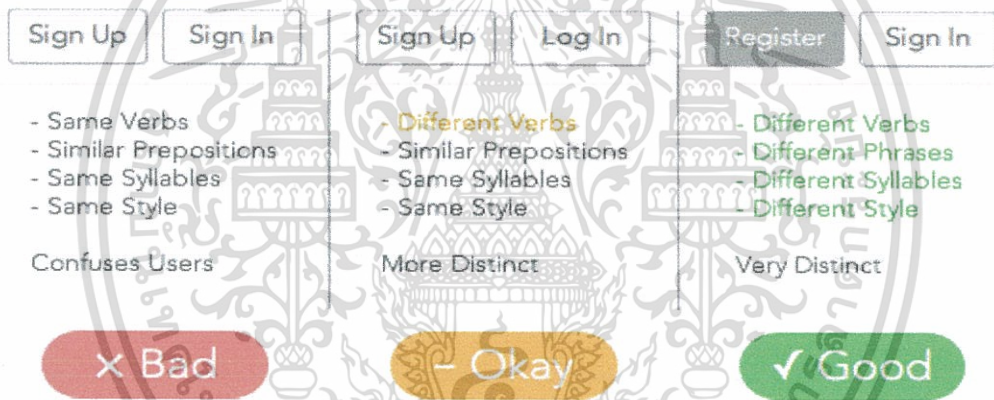
ในการออกแบบซอฟต์แวร์หรือเว็บไซต์นั้น ในปัจจุบันจำเป็นต้องมีการออกแบบอยู่ 2 ส่วนใหญ่ๆ ที่ผู้ที่ออกแบบจำเป็นต้องคำนึงถึง คือส่วนของ User Interface ที่เป็นส่วนติดต่อกับผู้ใช้ และ User Experience ที่เป็นเหมือนศาสตร์ของการออกแบบแขนงหนึ่งที่กำลังได้รับความนิยมอย่างมากในการพัฒนาซอฟต์แวร์หรือเว็บไซต์ สำหรับความหมายและหลักการในการออกแบบของทั้ง 2 ส่วน มีดังนี้

- User Interface (UI) Design หรือการออกแบบส่วนติดต่อผู้ใช้ ส่วนติดต่อผู้ใช้เป็นส่วนสำคัญอย่างมากในการดึงดูดให้ผู้ใช้สนใจมาใช้ซอฟต์แวร์หรือเว็บไซต์ เพราะถ้าซอฟต์แวร์หรือเว็บไซต์ได้มีการออกแบบ UI ออกมาได้สวยงาม สบายตา ผู้ใช้ ผู้ใช้ก็อยากจะใช้งานซอฟต์แวร์หรือเว็บไซต์นั้นๆ มากกว่าซอฟต์แวร์หรือเว็บไซต์ที่มีการออกแบบ UI แบบไม่โดดเด่นหรือนำดึงดูดให้ใช้ แต่อย่างไรก็ตามในปัจจุบันมีการทดลองมากมายที่พิสูจน์ว่าการออกแบบ UI ให้สวยงามเพียงอย่างเดียว ไม่สามารถทำให้ผู้ใช้สนใจใช้ซอฟต์แวร์หรือเว็บไซต์นั้นๆ ได้มากพอ เพราะฉะนั้นจำเป็นต้องมีการออกแบบ User Experience ควบคู่กันไปด้วย
- User Experience (UX) หรือแปลตามตัวได้ว่า ประสบการณ์ของผู้ใช้ นับตั้งแต่มีการคิดค้นและวิจัยเรื่อง User Experience จนกระทั่งปัจจุบันยังไม่สามารถให้คำนิยามที่ตายตัวของ UX ได้ โดยผู้ที่ได้ศึกษาเกี่ยวกับ UX มักให้คำอธิบายโดยรวมว่า UX ไม่ใช่เพียงความสวยงามของซอฟต์แวร์หรือเว็บไซต์ แต่เป็นความพึงพอใจของผู้ใช้ที่มีต่อซอฟต์แวร์หรือเว็บไซต์นั้นๆ ด้วย ความพึงพอใจของผู้ใช้จึงเกิดจากหลายปัจจัยที่เกี่ยวข้องกับซอฟต์แวร์หรือเว็บไซต์นั้นๆ มีด้วยกัน 3 ข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักๆดังนี้ มีความสวยงาม, ใช้งานง่าย, ผู้ใช้สามารถเรียนรู้การใช้งานได้อย่างรวดเร็วโดยอาศัยการเรียนรู้จากคู่มือน้อยที่สุดหรืออาจไม่จำเป็นต้องศึกษาวิธีใช้จากคู่มือเลย หากซอฟต์แวร์หรือเว็บไซต์ที่มีการออกแบบ UX ที่ดีจำเป็นจะต้องมีปัจจัยทั้ง 3 อย่างที่ได้กล่าวไป

จะเห็นได้ว่าการทำให้ผู้ใช้พึงพอใจในซอฟต์แวร์หรือเว็บไซต์จำเป็นต้องมอบ User Experience ที่ดีให้กับผู้ใช้ และการที่ซอฟต์แวร์หรือเว็บไซต์จะสามารถมอบ User Experience ที่ดีให้กับผู้ใช้ได้จำเป็นจะต้องมีการออกแบบ User Interface ที่ดีเป็นหนึ่งในปัจจัยหลักด้วย นอกจากนี้ทีมพัฒนาและบุคลากรทางด้านธุรกิจก็ยังส่งผลต่อ User Experience ที่ดีด้วย เพราะหากทีมออกแบบได้ออกแบบ User Interface ที่สวยงามมา และทำการทดลองว่า User Interface นี้ใช้งานง่ายเพียงใดแล้ว แต่ทีมพัฒนาไม่สามารถพัฒนาซอฟต์แวร์หรือเว็บไซต์ให้ออกมาดีและมีการใช้งานที่ลื่นไหลแล้ว ซอฟต์แวร์หรือเว็บไซต์นั้นๆ จะไม่สามารถมอบ User Experience ที่ดีให้กับผู้ใช้ได้เลย



ภาพที่ 2.14.1 ตัวอย่างเปรียบเทียบการออกแบบให้ได้ User Experience ที่ดีและไม่ดี ที่มา : <https://s-media-cache-ak0.pinimg.com/originals/13/f0/79/13f07917b0dc1fb8644316aac0c52b3a.png>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานวิจัย

ในการดำเนินงานวิจัยให้สำเร็จลุล่วงไปได้ด้วยดีความจุดประสงค์และระยะเวลาที่กำหนดนั้น จำเป็นต้องมีการวางแผนการดำเนินงานอย่างเป็นระบบระเบียบ ดังนี้

- เก็บข้อมูลที่ต้องใช้ในการพิจารณาว่าระบบที่ต้องการพัฒนาควรมีความสามารถอย่างไรบ้าง และอะไรที่ผู้ใช้ต้องการ
- ออกแบบโครงสร้างของระบบ และระบบฐานข้อมูลของระบบ
- ออกแบบส่วนติดต่อผู้ใช้ของแอปพลิเคชัน
- พัฒนาระบบตามที่ได้วางแผนและออกแบบไว้
- ทดลองระบบและปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้นภายในระบบ

3.1 วางแผนและเก็บข้อมูล

ก่อนจะเริ่มการวางแผนการทำงาน จำเป็นต้องทราบข้อมูลต่างๆของกลุ่มผู้ใช้ ซึ่งหาได้จากการสอบถามโดยตรงกับกลุ่มผู้ใช้งานในองค์กร ว่ากลุ่มผู้ใช้งานต้องการอะไรที่จะทำให้การใช้งานแอปพลิเคชันหรือระบบนี้เป็นไปได้โดยง่ายและมีความน่าใช้ที่ผู้ใช้งานเห็นแล้วอยากจะใช้ หรือต้องเปลี่ยนจากการเขียนใบลาที่เคยทำมาใช้แอปพลิเคชันนี้ส่งคำขอลาแทน

หลังจากสอบถามและเก็บข้อมูลจากกลุ่มผู้ใช้งานในองค์กรมาแล้ว สามารถสรุปสิ่งที่กลุ่มผู้ใช้งานต้องการออกมาเป็นหัวข้อได้ ดังนี้

- ส่งคำขอลาได้ทันทีแทนการเขียนใบลาแล้วต้องเดินไปยื่นให้กับหัวหน้าหรือผู้มีอำนาจพิจารณาคำขอลา
- สามารถตรวจสอบสถานะคำขอลาที่ส่งไปแล้วได้ว่าได้รับการอนุญาตหรือไม่
- สามารถตรวจสอบจำนวนวันลาที่เหลือของผู้ใช้ได้ด้วยตัวเอง
- ผู้มีอำนาจพิจารณาคำขอลาสามารถตอบรับคำขอลาที่ถูกส่งมาได้จากภายในแอปพลิเคชัน
- มีการเก็บประวัติการลาอย่างเป็นระบบระเบียบที่สุดเท่าที่จะสามารถทำได้
- สามารถตรวจสอบประวัติการลาของตนเองได้จากภายในแอปพลิเคชัน

จากหัวข้อต่างๆดังกล่าวนับเป็นความสามารถหลักของตัวแอปพลิเคชันและระบบการลา (Leave Request) นอกจากความสามารถหลักๆดังกล่าวแล้ว ผู้ใช้บางคนยังมีความต้องการบางอย่างเพิ่มเติมด้วย เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้นเพื่อใช้ในการพัฒนาและเรียนรู้เท่านั้นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนติดต่อผู้ใช้งานของแอปพลิเคชันต้องใช้งานได้ง่าย สามารถเรียนรู้การใช้งานแอปพลิเคชันด้วยตัวของผู้ใช้เองได้
- ผู้ใช้แต่ละคนต้องมีการยืนยันตัวตนก่อนการเข้าใช้แอปพลิเคชัน เพื่อจะได้สามารถระบุได้ว่าคำขอที่ถูกส่งออกไปมาจากผู้ใช้คนใดโดยไม่จำเป็นต้องกรอกชื่อของผู้ใช้ทุกครั้งที่ต้องการส่งคำขอ

หลังจากรวบรวมข้อมูลได้พอสมควรจึงนำข้อมูลที่ได้ทั้งหมดมาออกแบบโครงสร้างและการทำงานโดยรวมของระบบ Leave Request

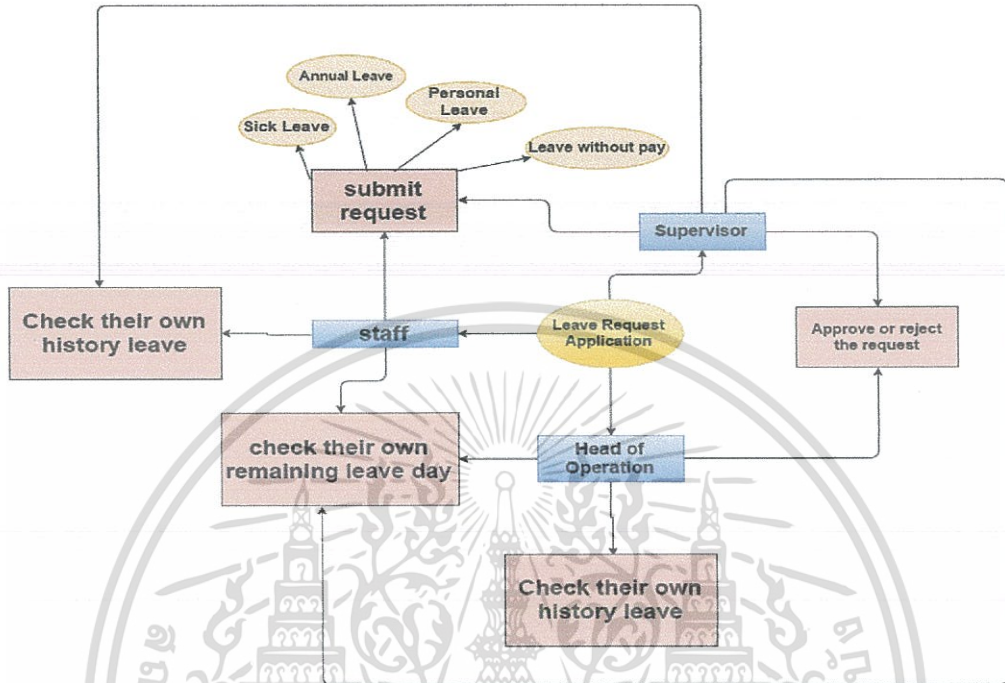
ตารางที่ 3.1 ตารางแสดงขั้นตอนในการดำเนินงานวิจัยและช่วงเวลาในการดำเนินงานวิจัย

สัปดาห์ที่	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
วางแผน	→																
เก็บข้อมูล		→															
ออกแบบระบบ			→														
ออกแบบส่วนติดต่อผู้ใช้				→													
พัฒนาระบบ																	
ทดลอง																	
ปรับปรุงแก้ไข																	

จากตารางที่ 3.1 แสดงถึงแผนการดำเนินงานทั้งหมดตั้งแต่เริ่มวางแผน เก็บข้อมูล ไปจนถึงการพัฒนา ระบบ ทดลองและปรับปรุงแก้ไขระบบให้ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

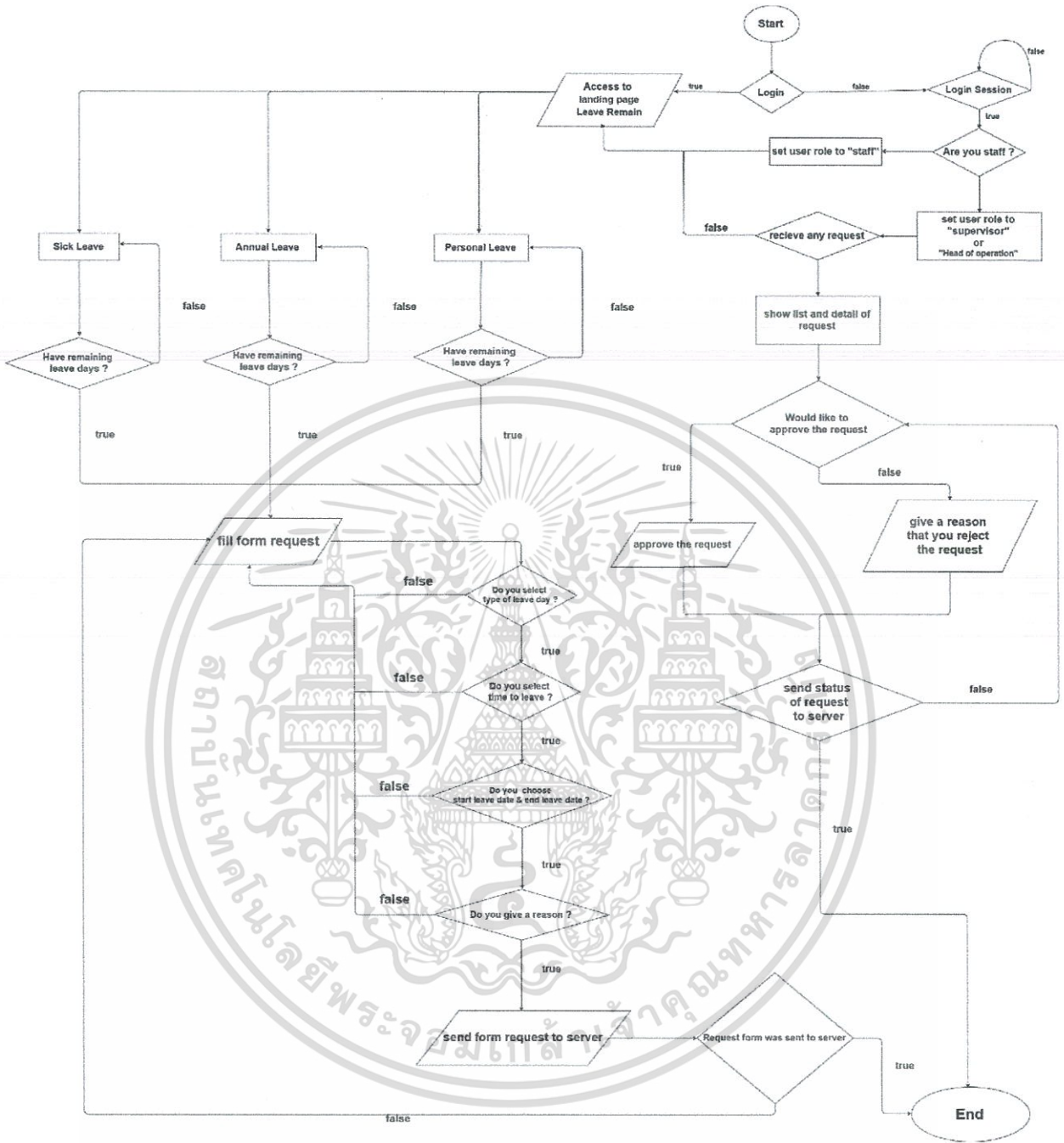
3.2 ออกแบบโครงสร้างของระบบ



ภาพที่ 3.2.1 โครงสร้างการทำงานของระบบ Leave Request ส่วนของแอปพลิเคชัน

ในการออกแบบขั้นต้นนั้นได้มีการร่าง Workflow diagram ของทั้งระบบเพื่อกำหนดว่าระบบและแอปพลิเคชัน Leave Request จะสามารถทำอะไรได้และด้วยเวลาที่จำกัดจำเป็นต้องลดความสามารถบางอย่างลงไป หรือสามารถเพิ่มเติมความสามารถบางอย่างเพิ่มเข้ามาในกรณีที่มีการพัฒนาเป็นไปอย่างรวดเร็วที่วางแผนไว้ แต่อย่างไรก็ตาม ความสามารถหลักของแอปพลิเคชัน ยังคงต้องมีและไม่สามารถลดลงได้ ทำให้ได้ Workflow diagram ดังภาพที่ 3.2.1 ที่แสดงถึงการทำงานทั้งระบบ

หลังจากนั้นจึงได้ออกแบบ Flowchart ของแอปพลิเคชัน Leave Request โดยที่แอปพลิเคชันต้องมีความสามารถหลักทั้งหมดตามที่ได้เก็บข้อมูลมา จึงทำให้ได้ Flowchart ดังภาพที่ 3.2.2



ภาพที่ 3.2.2 Flowchart diagram ของแอปพลิเคชัน Leave Request

สำหรับโครงสร้างของแอปพลิเคชันที่ได้ออกแบบมานั้น เมื่อเปิดเข้ามาหน้าแรกของแอปพลิเคชัน Leave Request แล้วแอปพลิเคชันจะมีการตรวจสอบก่อนว่าผู้ใช้ได้ทำการล็อกอินเข้าใช้ในแอปพลิเคชันหรือไม่ หากยังไม่ได้ล็อกอิน จะนำผู้ใช้ไปสู่หน้าล็อกอินเพื่อทำการล็อกอิน เมื่อล็อกอินแล้วระบบจะตรวจสอบว่าผู้ใช้ที่ล็อกอินเข้ามานั้นคือบุคลากรตำแหน่งใด โดยได้แบ่งเป็น 3 ตำแหน่งตามโครงสร้างขององค์กรเป้าหมาย คือ พนักงานทั่วไป เป็น Staff, หัวหน้าทีมหรือแผนกต่างๆ เป็น Supervisor และระดับหัวหน้าสาขาหรือระดับผู้จัดการที่มีอำนาจสูงสุดภายในองค์กร เป็น Head of operation เมื่อไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

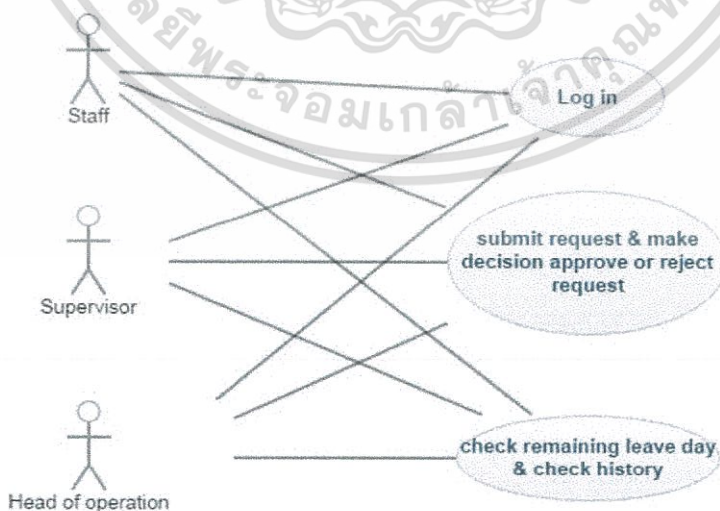
ตรวจสอบแล้ว จะนำผู้ใช้ไปสู่หน้าแรกที่แสดงวันลาที่เหลือของผู้ใช้รายนั้น และผู้ใช้สามารถเข้าสู่หน้ากรอกแบบฟอร์มคำขอลาได้ที่หน้านี้ แล้วแอปพลิเคชันจะนำไปสู่หน้ากรอกแบบฟอร์มคำขอลา หากมีช่องใดในแบบฟอร์มที่ผู้ใช้ไม่ได้กรอก จะมีการแจ้งเตือนว่าช่องใดที่ผู้ใช้ไม่ได้กรอก เมื่อกรอกข้อมูลครบถ้วนถูกต้อง จึงจะสามารถกดส่งคำขอลาได้

ในกรณีที่ผู้ใช้ที่ล็อกอินเข้ามาเป็น Supervisor หรือ Head of operation ก็จะมาสู่หน้าเดียวกันกับ Staff แต่ Supervisor และ Head of operation จะมีเมนูคำขอที่เข้ามาเพื่อตรวจสอบคำขอลาถูกส่งมาถึงตนเอง และจะสามารถกดเข้าไปดูรายละเอียดและสามารถอนุญาตให้ลาหรือปฏิเสธคำขอลาได้ ถ้าหากเป็น Supervisor ถ้าอนุญาตให้ลาได้ คำขอลาจึงจะถูกส่งต่อไปยัง Head of operation เพื่อให้ Head of operation ยืนยันการอนุญาตให้ลาอีกครั้ง บุคลากรที่ส่งคำขอลามาจึงจะสามารถลาได้

Head of operation ถือเป็นผู้มีอำนาจสูงสุดในองค์กร หากต้องการลาหยุดจึงไม่จำเป็นต้องส่งคำขอลาเพื่อขออนุญาต แต่ Supervisor ยังถือว่าอยู่ภายใต้ Head of operation หากต้องการลาจึงต้องส่งคำขอลาไปยัง Head of operation เพื่อขออนุญาตลาก่อนเช่นเดียวกับ Staff

ทั้งนี้ผู้ใช้ทั้ง 3 ตำแหน่ง สามารถตรวจสอบวันลาที่เหลือของตัวเองและประวัติการลาของตัวเองได้ภายในแอปพลิเคชัน เพื่อวางแผนการลาในอนาคตของตนได้ด้วย

นอกจากการออกแบบ Flowchart การทำงานของแอปพลิเคชันแล้วยังได้ทำการออกแบบ Flowchart ของฝั่งระบบเซิร์ฟเวอร์ด้วย เนื่องจากระบบ Leave Request มีการทำงานที่ต้องเชื่อมต่อกับระบบฐานข้อมูลขององค์กร จึงจำเป็นต้องออกแบบให้แอปพลิเคชันทำงานควบคู่ไปกับการทำงานของระบบเซิร์ฟเวอร์ภายในองค์กรด้วย

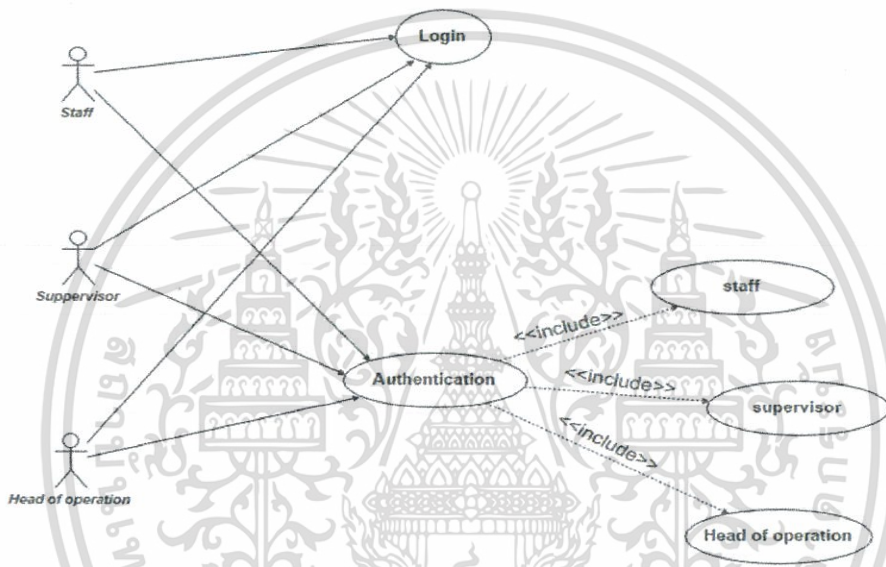


ภาพที่ 3.2.3 Use case diagram โดยรวมของระบบ Leave Request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 3.2.3 เป็น Use case diagram ที่แสดงถึงภาพรวมของระบบ Leave Request ว่าระบบมีความสามารถหลักๆ ดังนี้

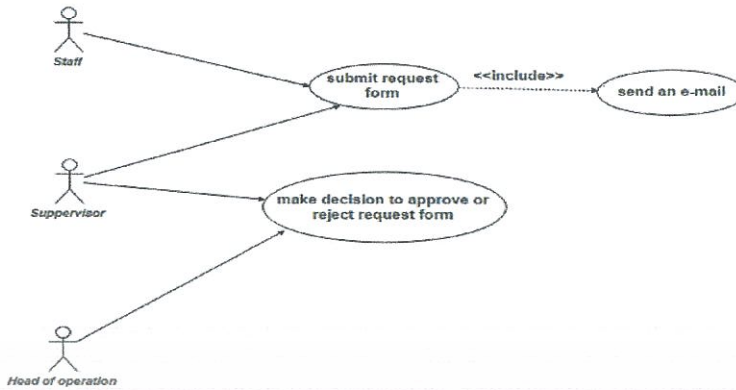
- ผู้ใช้ทุกคนสามารถทำการล็อกอินเข้าใช้ระบบได้
- ผู้ใช้ทุกคนสามารถส่งคำขอลาได้ และผู้ใช้ที่มีอำนาจในการพิจารณาคำขอลาสามารถตัดสินใจได้ว่าจะอนุญาตให้คำขอลาที่ส่งเข้ามาสามารถลาหยุดได้หรือไม่
- ผู้ใช้ทุกคนสามารถตรวจสอบคำขอลาที่เหลือ และคำขอลาที่เคยส่งออกไปแล้วได้



ภาพที่ 3.2.4 Use case diagram ฟังก์ชัน Log in ของแอปพลิเคชัน Leave Request

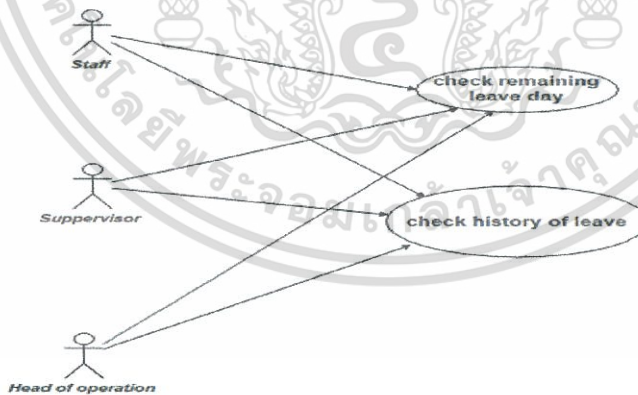
จากภาพที่ 3.2.4 เป็น Use case diagram แสดงถึงการล็อกอินเข้าใช้งานระบบ ผู้ใช้ทุกคนสามารถทำการล็อกอินเข้าใช้ระบบได้ และเมื่อมีการล็อกอินเข้ามาภายในระบบ ระบบจะมีการตรวจสอบและระบุว่าผู้ใช้นั้นมีตำแหน่งเป็น Staff, Supervisor หรือ Head of Operation ซึ่งส่วนติดต่อผู้ใช้ของผู้ใช้แต่ละตำแหน่ง จะมีบางฟังก์ชันที่แตกต่างกันเล็กน้อย คือผู้ใช้ที่เป็น Staff จะไม่สามารถพิจารณาคำขอลาของผู้ใช้รายอื่นได้ จึงไม่มีเมนูในการแสดงคำขอลาที่ส่งเข้ามา แต่ผู้ใช้ที่เป็น Supervisor หรือ Head of operation จะมีเมนูนี้แสดงในส่วนติดต่อผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2.5 Use case diagram ที่เกี่ยวข้องกับกรส่งคำขอลา และการตอบรับคำขอลา ของแอปพลิเคชัน Leave Request

จากภาพที่ 3.2.5 เป็น Use case diagram ที่แสดงถึงฟีเจอร์ในการส่งคำขอลาของผู้ใช้ที่มีตำแหน่งเป็น Staff และ Supervisor ที่เมื่อมีการส่งคำขอลาออกไปแล้ว ระบบจะทำการส่งอีเมลไปยังผู้มีอำนาจในการพิจารณาคำขอลาขั้นแรก ถ้าหากเป็นผู้ใช้ที่เป็น Staff ส่งคำขอลา อีเมลล์จะถูกส่งไปยัง Supervisor และจะส่งต่อไปยัง Head of operation เมื่อ Supervisor ทำการอนุมัติคำขอลานั้นแล้ว แต่ถ้าหากผู้ใช้ที่เป็น Supervisor เป็นคนส่งคำขอลาอีเมลล์จะถูกส่งไปยัง Head of operation ทันที



ภาพที่ 3.2.6 Use case diagram ฟีเจอร์ตรวจสอบจำนวนวันลาที่เหลือและประวัติการขอลาที่ผ่านมา ของแอปพลิเคชัน Leave Request

จากภาพที่ 3.2.6 เป็น Use case diagram แสดงถึงฟีเจอร์ที่ผู้ใช้ทุกคนสามารถตรวจสอบคำขอลาที่เหลือของตนเอง และตรวจสอบคำขอลาที่ผ่านมาของตนเองได้ ในส่วนนี้ผู้ใช้ทุกคนจะสามารถตรวจสอบข้อมูลของตัวเองได้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 ตารางรายละเอียด Use Case ระบบ Log in เข้าสู่ระบบ Leave Request

Use Case ID:	1
Use Case Name:	ระบบ Log in
Actor:	ผู้ที่ต้องการเข้าใช้ระบบทุกคน (User)
Description:	<p>ผู้ที่มีความต้องการที่จะเข้าใช้ระบบ Leave Request จำเป็นต้องทำการ Log in เพื่อระบุตัวตนว่าผู้ใช้แต่ละรายนั้น เป็นตำแหน่งใดภายในองค์กร โดยแบ่งเป็นหลักๆได้ ดังนี้</p> <ul style="list-style-type: none"> - Staff - Supervisor - Head of operations
Trigger:	ผู้ใช้ทำการกรอก Username และ password ก่อนกดปุ่ม Log in หลังจากนั้นระบบจะทำการตรวจสอบ Username และ password ว่าถูกต้องหรือไม่
Precondition:	ผู้ใช้ทำการกรอก Username และ password ก่อนกดปุ่ม Log in
Post-condition:	ระบบทำการตรวจสอบว่า Username และ password
Normal Flow:	<ol style="list-style-type: none"> 1.) ผู้ใช้กรอก Username และ password 2.) ผู้ใช้กดปุ่ม Log in 3.) ระบบทำการตรวจสอบ Username และ password
Exception:	<p>ถ้าหากไม่มีการกรอก Username หรือ password หรือมีการกรอก Username หรือ password ไม่ถูกต้อง แอปพลิเคชันจะมีการแจ้งเตือนกับผู้ใช้ว่า “ กรุณากรอก username ให้ถูกต้อง ” หรือ “ กรุณากรอก password ให้ถูกต้อง ”</p>
Note and Issue:	-

จากตารางที่ 3.2 แสดงถึงข้อมูลของ Use case diagram ในส่วนของการล็อกอินเข้าใช้ระบบซึ่งได้อธิบายไว้อย่างละเอียดว่า ผู้ใช้ที่ต้องการเข้าใช้ระบบ Leave Request จำเป็นจะต้องทำการล็อกอินเข้าสู่ระบบโดยการกรอก Username และ Password ในการเข้าใช้ระบบ ถ้าหากผู้ใช้กรอก Username และ Password ถูกต้องครบถ้วนแอปพลิเคชันจะนำผู้ใช้เข้าสู่หน้าแรกของแอปพลิเคชัน แต่หากมีการกรอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Username หรือ Password ผิด จะมีข้อความแจ้งเตือนผู้ใช้งานว่า “ กรุณากรอก username ให้ถูกต้อง ” หรือ “ กรุณากรอก password ให้ถูกต้อง ”

ตารางที่ 3.3 ตารางรายละเอียด Use Case ระบบ Authentication ของระบบ Leave Request

Use Case ID:	2
Use Case Name:	ระบบ Authentication
Actor:	ผู้ที่ต้องการเข้าใช้ระบบทุกคน (User)
Description:	หลังจากผู้ใช้งานทำการกดปุ่ม Log in แล้วระบบจะทำการตรวจสอบ Username และ password ที่ผู้ใช้งานกรอกเข้ามาว่าถูกต้องหรือไม่ ถ้าถูกต้องแล้ว ผู้ใช้รายนั้นกำลังดำรงตำแหน่งใดภายในองค์กร ซึ่งแบ่งได้หลักๆ 3 ตำแหน่ง ดังนี้ <ul style="list-style-type: none"> - Staff - Supervisor - Head of operation
Trigger:	ผู้ใช้งานกรอก Username และ password ก่อนกดปุ่ม Log in หลังจากนั้นระบบจะทำการตรวจสอบ Username และ password ว่าถูกต้องหรือไม่
Precondition:	ผู้ใช้งานกรอก Username และ password ก่อนกดปุ่ม Log in
Post-condition:	ระบบทำการตรวจสอบว่า Username และ password และถ้าถูกต้องระบบจะระบุว่าเป็น Staff, Supervisor หรือ Head of operation
Normal Flow:	<ol style="list-style-type: none"> 1.) ผู้ใช้กรอก Username และ password 2.) ผู้ใช้กดปุ่ม Log in 3.) ระบบทำการตรวจสอบ Username และ password 4.) ระบบระบุตัวตนของผู้ใช้ว่าเป็น Staff, Supervisor หรือ Head of operation
Exception:	ถ้าหาก Username หรือ password ไม่ถูกต้อง ระบบจะมีข้อความแจ้งเตือนผู้ใช้งานว่า “ กรุณากรอก username ให้ถูกต้อง ” หรือ “ กรุณากรอก password ให้ถูกต้อง ”
Note and Issue:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.3 แสดงถึงข้อมูลของ Use case diagram ระบบ Authentication ของระบบ Leave Request ที่เมื่อผู้ใช้ทำการกรอก Username และ Password อย่างถูกต้องครบถ้วนแล้ว ระบบจะทำการตรวจสอบว่าผู้ใช้นั้นมีตำแหน่งเป็นใคร Staff, Supervisor หรือ Head of operation เพื่อแสดงเมนูภายในแอปพลิเคชันให้กับผู้ใช้ได้อย่างถูกต้อง

ตารางที่ 3.4 ตารางรายละเอียด Use Case ระบบส่งคำขอลา ของระบบ Leave Request

Use Case ID:	3
Use Case Name:	ระบบส่งคำขอลา
Actor:	ผู้ที่ต้องการส่งคำขอลา (Staff, Supervisor)
Description:	<p>ผู้ใช้สามารถกรอกแบบฟอร์มคำขอลา และสามารถกดส่งเข้าสู่ระบบเพื่อส่งต่อคำขอลานี้ไปถึงผู้ที่มีอำนาจอนุญาตให้ลาได้</p> <ul style="list-style-type: none"> - ประเภทของวันลา - ช่วงเวลาที่ต้องการลา (ครึ่งวัน, เต็มวัน) - วันแรกที่ต้องการลาจนถึงวันสุดท้ายของการลา - จำนวนวันลาที่ต้องการลา - เหตุผลของการลา
Trigger:	ผู้ใช้กรอกแบบฟอร์มคำขอลาครบถ้วน และกดปุ่มส่งคำขอลา คำขอลาจะถูกส่งเข้าสู่ระบบฐานข้อมูลพร้อมกับส่งต่อไปยังผู้มีอำนาจพิจารณาคำขอลา
Precondition:	ผู้ใช้กดปุ่ม “คำขอลา” เพื่อเข้าสู่หน้ากรอกแบบฟอร์มคำขอลา
Post-condition:	ผู้ใช้กดปุ่ม “ส่งคำขอลา” ระบบทำการเก็บแบบฟอร์มคำขอลาเข้าสู่ระบบฐานข้อมูลและส่งต่อคำขอลาไปยังผู้มีอำนาจพิจารณาคำขอลา และคำขอลามีสถานะเป็น “อยู่ระหว่างดำเนินการ”
Normal Flow:	<ol style="list-style-type: none"> 1.) ผู้ใช้กดปุ่ม “คำขอลา” เพื่อเข้าสู่หน้ากรอกแบบฟอร์มคำขอลา 2.) ผู้ใช้กรอกแบบฟอร์มคำขอลาให้ครบถ้วน 3.) ผู้ใช้กดปุ่ม “ส่งคำขอลา” 4.) ระบบส่งแบบฟอร์มคำขอลาเข้าสู่ระบบฐานข้อมูล 5.) ระบบส่งแบบฟอร์มคำขอลาไปยังผู้มีอำนาจพิจารณาคำขอลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Exception:	<p>ถ้าหากผู้ใช้กรอกแบบฟอร์มไม่ครบถ้วนจะมีข้อความเตือนผู้ใช้ ดังนี้</p> <ul style="list-style-type: none"> - “กรุณาเลือกประเภทการลา” - “กรุณาเลือกช่วงเวลาที่ต้องการลา” - “กรุณากรอกวันแรกที่ต้องการลา” - “กรุณากรอกวันสุดท้ายที่ต้องการลา” - “กรุณากรอกจำนวนวันลาทั้งหมด” - “กรุณากรอกเหตุผลที่ต้องการลา” <p>โดยข้อความจะขึ้นตามช่องของแบบฟอร์มที่ผู้ใช้ไม่ได้ทำการกรอกข้อมูล</p>
Note and Issue:	-

จากตารางที่ 3.4 แสดงถึงข้อมูลของ Use case diagram ในส่วนของพีเจอรส์ส่งคำขอลา โดยผู้ใช้ทุกคนสามารถส่งคำขอลาได้ เพียงแต่ Head of operation ไม่จำเป็นต้องส่งเพราะถือเป็นตำแหน่งสูงสุดภายในองค์กรแล้วหากต้องการลาหยุดไม่จำเป็นต้องขออนุญาต ดังนั้นผู้ที่ต้องใช้งานพีเจอรส์นี้จึงเหลือเพียงผู้ใช้ที่มีตำแหน่ง Staff และ Supervisor ซึ่งเมื่อผู้ใช้เข้ามาถึงหน้ากรอกแบบฟอร์มคำขอลา ผู้ใช้จำเป็นต้องกรอกรายละเอียดภายในแบบฟอร์มคำขอลาให้ถูกต้องและครบถ้วน ดังนี้

- ประเภทของวันลา
- ช่วงเวลาที่ต้องการลา (ครึ่งวัน, เต็มวัน)
- วันแรกที่ต้องการลาจนถึงวันสุดท้ายของการลา
- จำนวนวันลาที่ต้องการลา
- เหตุผลของการลา

ซึ่งเมื่อผู้ใช้กรอกรายละเอียดตามแบบฟอร์มคำขอลาอย่างครบถ้วนสมบูรณ์แล้วผู้ใช้จึงสามารถกดปุ่มส่งคำขอลาได้ หลังจากนั้นระบบจะทำการเก็บคำขอลาที่ถูกส่งออกไปเข้าระบบฐานข้อมูลและส่งอีเมลล์แจ้งเตือนไปยังผู้มีอำนาจพิจารณาคำขอลาของผู้ใช้รายนั้น แต่หากผู้ใช้กรอกข้อมูลตามแบบฟอร์มไม่ครบถ้วน แอปพลิเคชันจะมีการแจ้งเตือนผู้ใช้ให้กรอกข้อมูลให้ครบถ้วนตามแต่ช่องว่างที่ผู้ใช้ไม่ได้ทำการกรอกข้อมูล ดังนี้

- “กรุณาเลือกประเภทการลา”
- “กรุณาเลือกช่วงเวลาที่ต้องการลา”
- “กรุณากรอกวันแรกที่ต้องการลา”
- “กรุณากรอกวันสุดท้ายที่ต้องการลา”
- “กรุณากรอกจำนวนวันลาทั้งหมด”
- “กรุณากรอกเหตุผลที่ต้องการลา”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 ตารางรายละเอียด Use Case ระบบส่งคำอนุญาตหรือไม่อนุญาต ของระบบ Leave Request

Use Case ID:	4
Use Case Name:	ระบบส่งคำอนุญาตหรือไม่อนุญาต
Actor:	ผู้ที่มีอำนาจอนุญาตให้ลาได้ (Supervisor, Head of operation)
Description:	<p>เมื่อคำขอลาถูกส่งมาถึงผู้มีอำนาจพิจารณาคำขอลาก็จะมีการแสดงที่หน้าแอปพลิเคชันหรือเว็บไซต์ของผู้มีอำนาจว่ามีคำขอส่งมาถึงในหน้า “คำขอลาที่ได้รับ” โดยจะแสดงเป็นลิสต์ สามารถเข้าดูรายละเอียดของแต่ละคำขอได้ ในหน้า “รายละเอียดคำขอลา” จะแสดงรายละเอียดของคำขอลาที่ส่งมาถึง ดังนี้</p> <ul style="list-style-type: none"> - ประเภทของวันลา - ช่วงเวลาที่ต้องการลา (ครึ่งวัน, เต็มวัน) - วันแรกที่ต้องการลาจนถึงวันสุดท้ายของการลา - จำนวนวันลาที่ต้องการลา - เหตุผลของการลา <p>โดยผู้ที่มีอำนาจสามารถตัดสินใจได้ว่าจะอนุญาตให้ลาหรือไม่ได้จากหน้านี้</p>
Trigger:	เมื่อระบบส่งคำขอลามาถึงผู้ที่มีอำนาจอนุญาตให้ลาได้ ผู้ที่มีอำนาจสามารถกดเพื่อดูรายละเอียดคำขอลาและตัดสินใจได้ว่าจะอนุญาตให้ลาหรือไม่ โดยการกดปุ่ม “อนุญาต” หรือ “ไม่อนุญาต”
Precondition:	ผู้มีอำนาจพิจารณาคำขอลาที่มีคำขอลาส่งมาถึง กดเข้าไปดูรายละเอียดของคำขอลาโดยกดที่คำขอลาที่แสดงเป็นลิสต์ได้ทันที หลังจากนั้นแอปพลิเคชันหรือเว็บไซต์จะแสดงรายละเอียดคำขอลาที่ได้กดเข้าไป พร้อมปุ่ม “อนุญาต” และ “ไม่อนุญาต” ที่ผู้มีอำนาจพิจารณาคำขอลาสามารถกดเพื่ออนุญาตหรือไม่อนุญาตให้ลาได้ทันที
Post-condition:	เมื่อผู้ที่มีอำนาจอนุญาตให้ลาได้ กดปุ่ม “อนุญาต” หรือ “ไม่อนุญาต” ระบบจะส่งสถานะนี้ไปยังระบบฐานข้อมูล และส่งต่อไปยังผู้มีอำนาจสูงสุด หากไม่มีผู้มีอำนาจสูงสุดอีกแล้ว ระบบจะเปลี่ยนสถานะของคำขอเป็น “ยืนยัน” กรณีที่ต้องได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	อนุญาตจากผู้ที่มีอำนาจสูงสุดก็ต้องให้ผู้ที่มีอำนาจสูงสุดกดปุ่ม “อนุญาต” สถานะของคำขอลาจึงเปลี่ยนเป็น “ยืนยัน” ที่แสดงว่าคำขอลานี้ได้ถูกอนุมัติให้ลาได้เป็นที่เรียบร้อยแล้ว หากไม่ได้รับอนุญาต จะมีการเปลี่ยนสถานะของคำขอลาดังกล่าวเป็นสถานะ “ปฏิเสธ”
Normal Flow:	<ol style="list-style-type: none"> 1.) ผู้มีอำนาจพิจารณาคำขอลา ได้รับคำขอลา 2.) ผู้มีอำนาจพิจารณาคำขอลาเข้าดูรายละเอียดของคำขอลา 3.) ผู้มีอำนาจพิจารณาคำขอลากดปุ่ม “อนุญาต” หรือ “ไม่อนุญาต” 4.) ระบบส่งสถานะ “ยืนยัน” หรือ “ปฏิเสธ” คำขอลาเข้าระบบฐานข้อมูล
Exception:	ถ้าหากไม่มีคำขอลาถูกส่งมายังผู้ที่มีอำนาจอนุญาตให้ลาได้ จะไม่มีการแสดงลิสต์ของคำขอลา
Note and Issue:	-

จากตารางที่ 3.5 แสดงถึงข้อมูลของ Use case diagram ในการตอบรับหรือปฏิเสธคำขอลาที่ส่งเข้ามาถึงผู้ใช้ที่มีตำแหน่ง Supervisor หรือ Head of operation ซึ่งเมื่อผู้ใช้กดเข้าไปดูคำขอลาที่ส่งเข้ามา แอปพลิเคชันจะแสดงรายละเอียดของคำขอลาตามแบบฟอร์มคำขอลาที่ถูกส่งเข้าทุกอย่าง และผู้ใช้สามารถตอบรับหรือปฏิเสธคำขอลาได้โดยการกดปุ่ม “อนุญาต” หรือ “ไม่อนุญาต” ที่อยู่ในส่วนล่างสุดของหน้าจอ เมื่อผู้ใช้ทำการตอบรับหรือปฏิเสธคำขอลาแล้ว จะมีการส่งสถานะของคำขอลานั้นเป็น “ยืนยัน” หรือ “ปฏิเสธ” ลงในระบบฐานข้อมูล

ตารางที่ 3.6 ตารางรายละเอียด Use Case ระบบตรวจสอบจำนวนวันลาที่เหลือ ของระบบ Leave Request

Use Case ID:	5
Use Case Name:	ระบบตรวจสอบจำนวนวันลาที่เหลือ
Actor:	ผู้ใช้ทุกคน
Description:	<p>ผู้ใช้สามารถตรวจสอบจำนวนวันลาที่เหลือของตนเองได้จากหน้านี้ โดยจะแบ่งแสดงเป็นแต่ละประเภทวันลา ดังนี้</p> <ul style="list-style-type: none"> - ลาป่วย (Sick Leave) - ลาพักร้อน (Annual Leave) - ลากิจ (Personal Leave)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ภายนอก
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Trigger:	ผู้ใช้กดที่แท็บบาร์ “วันลาที่เหลือ” ด้านบนของแอปพลิเคชัน หลังจากนั้นจะมีการแสดงจำนวนวันลาที่เหลือออกมา
Precondition:	ผู้ใช้เข้าใช้งานแอปพลิเคชัน และทำการ Log in เข้าสู่แอปพลิเคชัน
Post-condition:	ระบบนำผู้ใช้เข้าสู่หน้า “วันลาที่เหลือ” และแสดงจำนวนวันลาที่เหลือของผู้ใช้รายนั้น
Normal Flow:	<ol style="list-style-type: none"> 1.) ผู้ใช้เข้าสู่หน้าแรกของแอปพลิเคชัน 2.) ผู้ใช้กดแท็บบาร์เพื่อเข้าสู่หน้า “วันลาที่เหลือ” 3.) ระบบแสดงรายละเอียดจำนวนวันลาที่เหลือแต่ละประเภทวันลาของผู้ใช้รายนั้น
Exception:	หากเข้าสู่ระบบไม่ได้หรือไม่สามารถเข้าถึงข้อมูลจำนวนวันลาที่เหลือได้ จะมีข้อความแจ้งถึงสาเหตุของปัญหาที่ไม่สามารถเข้าถึงข้อมูลได้
Note and Issue:	-

จากตารางที่ 3.6 แสดงถึงข้อมูลของ Use case diagram ระบบตรวจสอบวันลาที่เหลือ ผู้ใช้ทุกคนสามารถเข้ามาตรวจสอบจำนวนวันลาที่เหลือของตนได้ โดยการกดที่ส่วนของแท็บบาร์ “วันลาที่เหลือ” ทางด้านบนของแอปพลิเคชัน หลังจากนั้นแอปพลิเคชันจะแสดงจำนวนวันลาโดยแบ่งเป็น 3 ประเภทวันลาดังนี้

- ลาป่วย (Sick Leave)
- ลาพักร้อน (Annual Leave)
- ลากิจ (Personal Leave)

หากเข้าสู่ระบบไม่ได้หรือไม่สามารถเข้าถึงข้อมูลจำนวนวันลาที่เหลือได้ จะมีข้อความแจ้งถึงสาเหตุของปัญหาที่ไม่สามารถเข้าถึงข้อมูลได้

ตารางที่ 3.7 ตารางรายละเอียด Use Case ระบบตรวจสอบคำขอลาที่ผ่านมา ของระบบ Leave Request

Use Case ID:	6
Use Case Name:	ระบบตรวจสอบคำขอลาที่ผ่านมา
Actor:	ผู้ใช้ทุกคน
Description:	ผู้ใช้สามารถตรวจสอบคำขอลาที่ผ่านมาของตนเองได้จากหน้านี้ โดยจะมีการแสดงเป็นลิสต์ของประวัติการลาที่ผ่านมาและสามารถกดเพื่อดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ในเชิงการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รายละเอียดของคำขอลาแต่ละครั้งได้จากการกดเข้าที่วันลาที่ระบุบนลิสต์ได้เลย
Trigger:	ผู้ใช้กดที่แท็บบาร์ “คำขอลาที่ผ่านมา” ที่อยู่ทางด้านบนของแอปพลิเคชัน หลังจากนั้นระบบจะแสดงประวัติคำขอลาที่ผ่านมาของผู้ใช้รายนั้น
Precondition:	ผู้ใช้เข้าใช้งานแอปพลิเคชัน และทำการ Log in เข้าสู่แอปพลิเคชัน หลังจากนั้นกดที่แท็บบาร์ “คำขอลาที่ผ่านมา”
Post-condition:	ระบบนำผู้ใช้เข้าสู่หน้า “คำขอลาที่ผ่านมา” และแสดงคำขอลาที่เคยส่งออกไปของผู้ใช้รายนั้น
Normal Flow:	<ol style="list-style-type: none"> 1.) ผู้ใช้เข้าสู่หน้าแรกของแอปพลิเคชัน 2.) ผู้ใช้กดแท็บบาร์เพื่อเข้าสู่หน้า “คำขอลาที่ผ่านมา” 3.) ระบบแสดงลิสต์ของคำขอลาที่ผ่านมาของผู้ใช้รายนั้น
Exception:	หากเข้าสู่ระบบไม่ได้หรือไม่สามารถเข้าถึงข้อมูลวันลาที่ผ่านมาได้ จะมีข้อความแจ้งถึงสาเหตุของปัญหาที่ไม่สามารถเข้าถึงข้อมูลได้
Note and Issue:	-

จากตารางที่ 3.7 แสดงถึงข้อมูลของ Use case diagram ของระบบตรวจสอบวันลาที่ผ่านมาของตนเอง ผู้ใช้ทุกคนสามารถตรวจสอบวันลาที่ผ่านมาของตนเองได้ โดยการกดที่แท็บบาร์ “คำขอลาที่ผ่านมา” ที่อยู่ทางด้านบนของแอปพลิเคชัน หลังจากนั้นระบบจะแสดงประวัติคำขอลาที่ผ่านมาของผู้ใช้รายนั้น โดยจะมีการแสดงเป็นลิสต์ของประวัติการลาที่ผ่านมาและสามารถกดเข้าดูรายละเอียดของคำขอลาแต่ละครั้งได้จากการกดเข้าที่วันลานั้นๆบนลิสต์ได้เลย

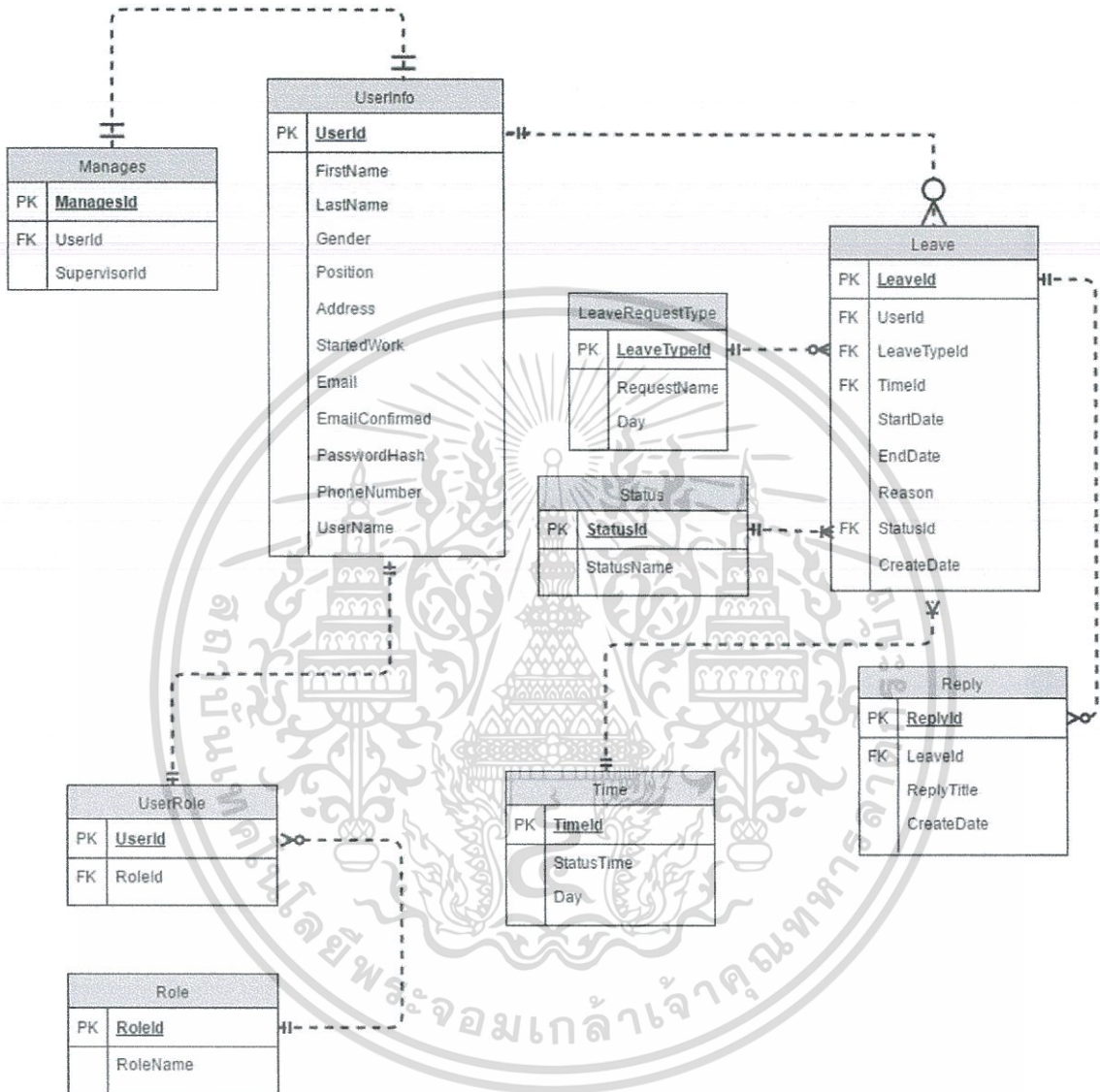
หากผู้ใช้เข้าสู่ระบบไม่ได้หรือไม่สามารถเข้าถึงข้อมูลวันลาที่ผ่านมาได้ จะมีข้อความแจ้งถึงสาเหตุของปัญหาที่ไม่สามารถเข้าถึงข้อมูลได้

จากการออกแบบระบบและแอปพลิเคชัน Leave Request นั้นสามารถแยกสิ่งที่ระบบ Leave Request สามารถทำได้เป็นหลัก ดังนี้

- ระบบ Log in
- ระบบระบุตัวตนของผู้ใช้ หรือ Authentication
- ระบบส่งคำขอลา
- ระบบอนุญาตคำขอลา
- ระบบตรวจสอบจำนวนวันลาที่เหลือของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของระบบตรวจสอบคำขอลาที่ผ่านมาของผู้ใช้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ออกแบบระบบฐานข้อมูล



ภาพที่ 3.3.1 แผนภาพ ER Diagram ระบบฐานข้อมูลของระบบ Leave Request

จากภาพที่ 3.3.1 ที่แสดงแผนภาพ ER Diagram ระบบฐานข้อมูลทั้งหมดของระบบ Leave Request ทำให้สามารถแบ่งระบบฐานข้อมูลออกเป็นตารางแสดงรายละเอียดได้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 ตารางเก็บข้อมูลส่วนตัวของผู้ใช้ (UserInfo)

Name	Type	Key	Meaning	Example
UserId	Integer	PK	รหัสเฉพาะตัวของผู้ใช้	0,1,2,3, ...
FirstName	varchar(30)		ชื่อจริงของผู้ใช้	อธิพร
LastName	varchar(30)		นามสกุลจริงของผู้ใช้	จันทร์ไชย
Gender	varchar(8)		เพศของผู้ใช้	ชาย
Position	varchar(30)		ตำแหน่งของผู้ใช้	Developer
Address	varchar(200)		ที่อยู่ของผู้ใช้	1 Chalong Krung ถนน ฉลองกรุง แขวง ลาดกระบัง เขต ลาดกระบัง กรุงเทพมหานคร 10520
StartedWork	Date		วันเริ่มทำงานของผู้ใช้	12/12/2016
Email	varchar(30)		อีเมลล์ของผู้ใช้	user1@gmail.com
EmailConfirmed	varchar(30)		อีเมลล์ของผู้ใช้ที่ใช้ตรวจสอบว่าตรงกับอีเมลล์ของผู้ใช้ในช่อง Email หรือไม่	user1@gmail.com
PasswordHash	varchar(10)		รหัสผ่านที่ใช้เพื่อเข้าสู่ระบบของผู้ใช้	1p2as3wor4d
PhoneNumber	Integer		หมายเลขโทรศัพท์ของผู้ใช้	0811911991
UserName	varchar(30)		username ที่ใช้ในการเข้าสู่ระบบของผู้ใช้	user1@gmail.com

จากตารางที่ 3.8 แสดงถึงข้อมูลของ ตารางเก็บข้อมูลส่วนตัวของผู้ใช้ ภายในระบบฐานข้อมูล ซึ่งประกอบไปด้วยข้อมูลดังนี้

- รหัสเฉพาะตัวของผู้ใช้ (UserId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- ชื่อจริงของผู้ใช้ (FirstName) เก็บเป็นข้อมูลประเภท varchar

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพศของผู้ใช้ (Gender) เก็บเป็นข้อมูลประเภท varchar
- ตำแหน่งงานของผู้ใช้ (Position) เก็บเป็นข้อมูลประเภท varchar
- ที่อยู่ของผู้ใช้ (Address) เก็บเป็นข้อมูลประเภท varchar
- วันแรกที่ใช้เริ่มทำงาน (StartedWork) เก็บเป็นข้อมูลประเภท Date
- อีเมลล์ของผู้ใช้ (Email) เก็บเป็นข้อมูลประเภท varchar
- อีเมลล์ของผู้ใช้ที่ใช้ตรวจสอบว่าตรงกับอีเมลล์ของผู้ใช้ในช่อง Email หรือไม่ (EmailConfirmed) เก็บเป็นข้อมูลประเภท varchar
- รหัสผ่านของผู้ใช้ที่ใช้เข้าสู่ระบบ (PasswordHash) เก็บเป็นข้อมูลประเภท varchar
- เบอร์โทรศัพท์ของผู้ใช้ (PhoneNumber) เก็บเป็นข้อมูลประเภท Integer
- Username ที่ผู้ใช้ใช้เข้าสู่ระบบ (UserName) เก็บเป็นข้อมูลประเภท varchar

ตารางที่ 3.9 ตารางเก็บรายละเอียดการจัดการแอดเคาท์ของผู้ใช้ (Manages)

Name	Type	Key	Meaning	Example
MangesId	Integer	PK	รหัสระบบการจัดการแอดเคาท์ของผู้ใช้	0,1,2,3, ...
UserId	Integer	FK	รหัสเฉพาะตัวของผู้ใช้	0,1,2,3, ...
SupervisorId	Integer		รหัสระบบตัวตนของบุคลากรในตำแหน่ง Supervisor แต่ละคน	0,1,2,3, ...

จากตารางที่ 3.9 แสดงถึงข้อมูลภายในตารางเก็บรายละเอียดการจัดการแอดเคาท์ของผู้ใช้ (Manages) ซึ่งมีข้อมูล ดังนี้

- รหัสระบบการจัดการแอดเคาท์ของผู้ใช้ (MangesId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- รหัสเฉพาะตัวของผู้ใช้ (UserId) เก็บเป็นข้อมูลประเภท Integer และเป็น Foreign Key
- รหัสระบบตัวตนของบุคลากรในตำแหน่ง Supervisor แต่ละคน (SupervisorId) เก็บเป็นข้อมูลประเภท Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 ตารางเก็บข้อมูลตำแหน่ง Staff, Supervisor และ Head of operation ของผู้ใช้แต่ละคน (UserRole)

Name	Type	Key	Meaning	Example
UserId	Integer	PK	รหัสเฉพาะตัวของผู้ใช้	0,1,2,3, ...
RoleId	Integer	FK	รหัสเฉพาะของแต่ละตำแหน่ง	0,1,2

จากตารางที่ 3.10 แสดงข้อมูลภายในตารางเก็บข้อมูลตำแหน่งงานของผู้ใช้ (UserRole) ซึ่งมีรายละเอียด ดังนี้

- รหัสเฉพาะตัวของผู้ใช้ (UserId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- รหัสเฉพาะของแต่ละตำแหน่ง (RoleId) เก็บเป็นข้อมูลประเภท Integer และเป็น Foreign Key

ตารางที่ 3.11 ตารางเก็บชื่อและรหัสของตำแหน่ง Staff, Supervisor และ Head of operation (Role)

Name	Type	Key	Meaning	Example
RoleId	Integer	PK	รหัสเฉพาะของแต่ละตำแหน่ง	0,1,2
RoleName	varchar(20)	FK	ชื่อตำแหน่ง	Staff, Supervisor, Head of operation

จากตารางที่ 3.11 แสดงข้อมูลภายในตารางเก็บชื่อและรหัสของตำแหน่งงานของผู้ใช้ (Role) ซึ่งมีรายละเอียด ดังนี้

- รหัสเฉพาะของแต่ละตำแหน่ง (RoleId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- ชื่อตำแหน่ง (RoleName) เก็บเป็นข้อมูลประเภท varchar และเป็น Foreign Key

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 ตารางเก็บรายละเอียดของประเภทการลา (LeaveRequestType)

Name	Type	Key	Meaning	Example
LeaveTypeIId	Integer	PK	รหัสเฉพาะของแต่ละประเภทวันลา	0,1,2,3
RequestName	varchar(20)		ประเภทวันลา	Sick Leave
Day	Float		จำนวนวันลาคงเหลือ	30

จากตารางที่ 3.12 แสดงข้อมูลภายในตารางเก็บรายละเอียดของประเภทการลา (LeaveRequestType) มีข้อมูล ดังนี้

- รหัสเฉพาะของแต่ละประเภทวันลา (LeaveTypeIId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- ประเภทวันลา (RequestName) เก็บเป็นข้อมูลประเภท varchar
- จำนวนวันลาคงเหลือ (Day) เก็บเป็นข้อมูลประเภท Float

ตารางที่ 3.13 ตารางเก็บรายละเอียดของคำขอลา (Leave)

Name	Type	Key	Meaning	Example
LeaveIId	Integer	PK	รหัสเฉพาะของแต่ละคำขอลา	0,1,2,3, ...
UserIId	Integer	FK	รหัสเฉพาะตัวของผู้ใช้	0,1,2,3, ...
LeaveType	varchar(20)	FK	ประเภทของวันลา	Sick Leave
TimeIId	Integer	FK	รหัสเฉพาะของแต่ละช่วงเวลาที่ต้องการลา	All day = 1
StartDate	Date		วันแรกที่ต้องการขอลา	12/12/2016
EndDate	Date		วันสุดท้ายที่ต้องการขอลา	12/12/2016
Reason	varchar(200)		เหตุผลที่ต้องการลา	ต้องการลาป่วย เพราะไม่สบาย
StatusIId	Integer	FK	รหัสเฉพาะของสถานะคำขอที่ถูกส่งไป	0,1,2,3
CreateDate	Date		วันและเวลาที่ส่งคำขอลานั้นๆ	12/12/2016 16:04:55

จากตารางที่ 3.13 แสดงข้อมูลภายในตารางเก็บรายละเอียดของคำขอลา (Leave) มีข้อมูล ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รหัสเฉพาะของแต่ละคำขอลา (Leaveld) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- รหัสเฉพาะตัวของผู้ใช้ (UserId) เก็บเป็นข้อมูลประเภท Integer และเป็น Foreign Key
- ประเภทของวันลา (LeaveType) เก็บเป็นข้อมูลประเภท varchar และเป็น Foreign Key
- รหัสเฉพาะของแต่ละช่วงเวลาที่ต้องการลา (Timeld) เก็บเป็นข้อมูลประเภท Integer และเป็น Foreign Key
- วันแรกที่ต้องการขอลา (StartDate) และวันสุดท้ายที่ต้องการขอลา (EndDate) เก็บเป็นข้อมูลประเภท Date
- เหตุผลที่ต้องการลา (Reason) เก็บเป็นข้อมูลประเภท varchar
- รหัสเฉพาะของสถานะคำขอที่ถูกส่งไป (StatusId) เก็บเป็นข้อมูลประเภท Integer และเป็น Foreign Key
- วันและเวลาที่ส่งคำขอลานั้นๆ (CreateDate) เก็บเป็นข้อมูลประเภท Date

ตารางที่ 3.14 ตารางเก็บสถานะของคำขอลา (Status)

Name	Type	Key	Meaning	Example
StatusId	Integer	PK	รหัสเฉพาะของแต่ละสถานะ	0,1,2,3
StatusName	varchar(10)		สถานะคำขอลา	Pending

จากตารางที่ 3.14 แสดงข้อมูลภายในตารางเก็บสถานะของคำขอลา (Status) มีข้อมูล ดังนี้

- รหัสเฉพาะของแต่ละสถานะ (StatusId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- สถานะคำขอลา (StatusName) เก็บเป็นข้อมูลประเภท varchar

ตารางที่ 3.15 ตารางเก็บช่วงเวลาที่ต้องการลา (Time)

Name	Type	Key	Meaning	Example
Timeld	Integer	PK	รหัสเฉพาะของแต่ละช่วงเวลาที่ต้องการลา	0,1,2
StatusTime	varchar(20)		ช่วงเวลาที่ต้องการลา	Half-Day (Morning)
Day	Float		ช่วงเวลาในการลา	0.5 วัน (ครึ่งวัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.15 แสดงถึงข้อมูลภายในตารางเก็บช่วงเวลาที่ต้องการลา (Time) มีข้อมูล ดังนี้

- รหัสเฉพาะของแต่ละช่วงเวลาที่ต้องการลา (TimeId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- ช่วงเวลาที่ต้องการลา (StatusTime) เก็บเป็นข้อมูลประเภท varchar
- ช่วงเวลาในการลา (Day) เก็บเป็นข้อมูลประเภท Float

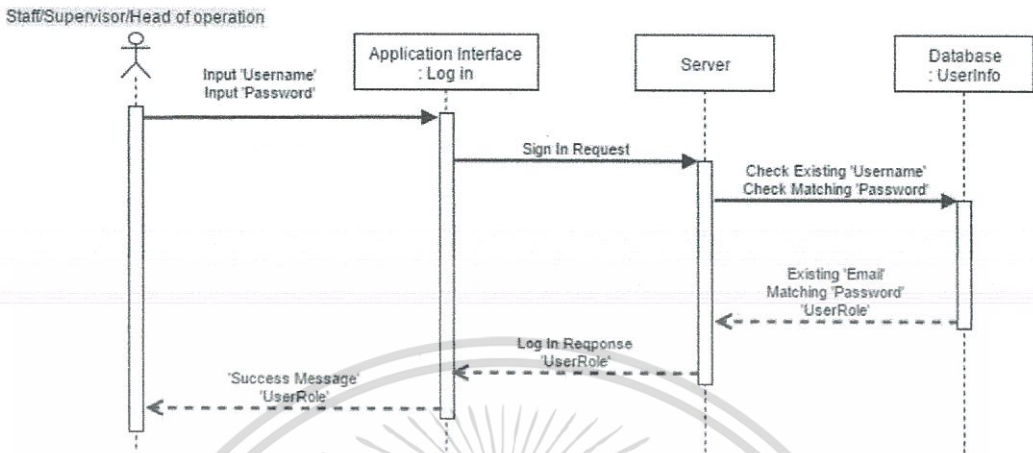
ตารางที่ 3.16 ตารางเก็บข้อมูลของการปฏิเสธคำขอลาพร้อมเหตุผล (Reply)

Name	Type	Key	Meaning	Example
ReplyId	Integer	PK	รหัสเฉพาะของการปฏิเสธคำขอลาแต่ละครั้ง	0,1,2,3, ...
Leaveld	Integer	FK	รหัสเฉพาะของแต่ละคำขอลา	0,1,2,3, ...
ReplyTitle	varchar(200)		เหตุผลที่ปฏิเสธคำขอลา	ไม่ให้เธอไป
CreateDate	Date		วันและเวลาที่ปฏิเสธคำขอลา	12/12/2016 16:04:55

จากตารางที่ 3.16 แสดงถึงข้อมูลภายในตารางเก็บข้อมูลของการปฏิเสธคำขอลาพร้อมเหตุผล (Reply) มีข้อมูล ดังนี้

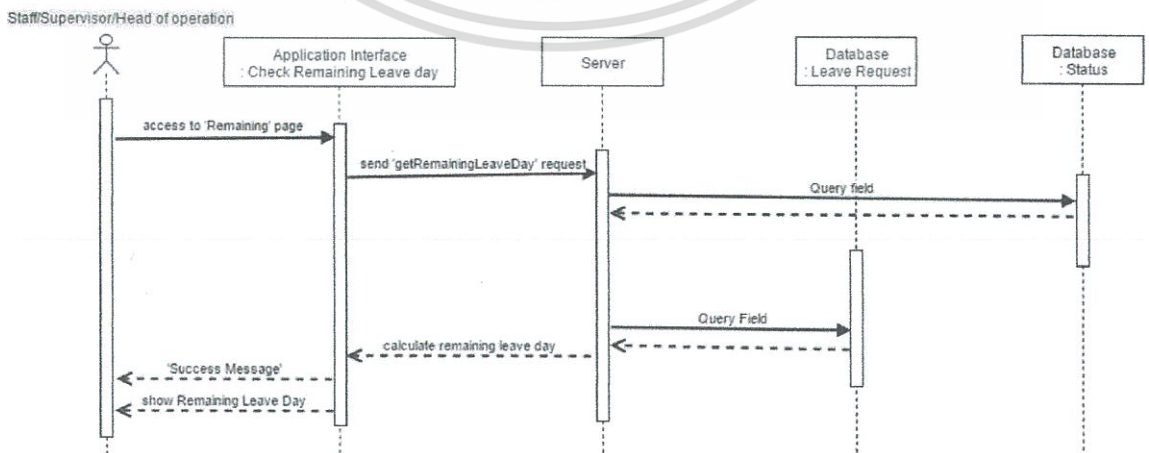
- รหัสเฉพาะของการปฏิเสธคำขอลาแต่ละครั้ง (ReplyId) เก็บเป็นข้อมูลประเภท Integer และเป็น Primary Key
- รหัสเฉพาะของแต่ละคำขอลา (Leaveld) เก็บเป็นข้อมูลประเภท Integer และเป็น Foreign Key
- เหตุผลที่ปฏิเสธคำขอลา (ReplyTitle) เก็บเป็นข้อมูลประเภท varchar
- วันและเวลาที่ปฏิเสธคำขอลา (CreateDate) เก็บเป็นข้อมูลประเภท Date

3.4 แผนภาพลำดับเหตุการณ์ (Sequence Diagram)



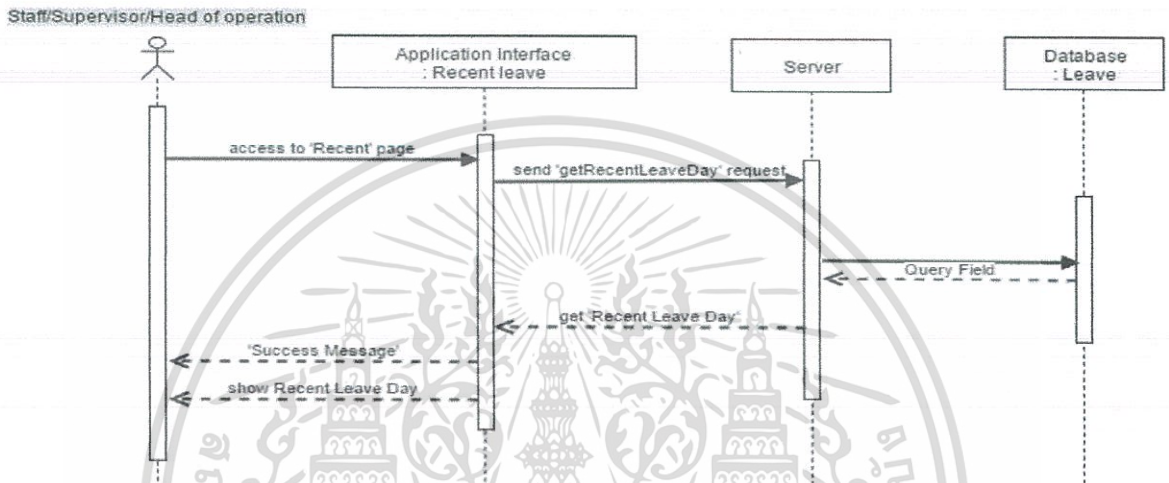
ภาพที่ 3.4.1 แผนภาพลำดับเหตุการณ์ระบบล็อกอินเข้าสู่ระบบ Leave Request

จากภาพที่ 3.4.1 เป็นแผนภาพลำดับเหตุการณ์ระบบล็อกอินเข้าสู่ระบบ Leave Request ที่ผู้ใช้ทุกคนจะต้องทำการล็อกอินเมื่อเข้าใช้งานแอปพลิเคชัน Leave Request เป็นครั้งแรก โดยเมื่อผู้ใช้เข้าสู่แอปพลิเคชันเป็นครั้งแรกแอปพลิเคชันจะนำผู้ใช้เข้าสู่หน้าล็อกอิน เพื่อทำการล็อกอินด้วยการกรอก Username และ Password เมื่อผู้ใช้ทำการกรอก Username และ Password แล้วกดปุ่ม Log in ระบบจะทำการส่ง Username และ Password ไปตรวจสอบว่ามีข้อมูลตรงตามที่ผู้ใช้ได้กรอกเข้ามาในระบบฐานข้อมูลหรือไม่ หากข้อมูลถูกต้องและมีข้อมูลในระบบฐานข้อมูลแล้วแอปพลิเคชันจะมีข้อความแจ้งกับผู้ใช้ว่า “ ล็อกอินสำเร็จ ” และนำผู้ใช้เข้าสู่หน้าตรวจสอบจำนวนวันลาที่เหลือซึ่งเป็นหน้าแรกของแอปพลิเคชันต่อไป



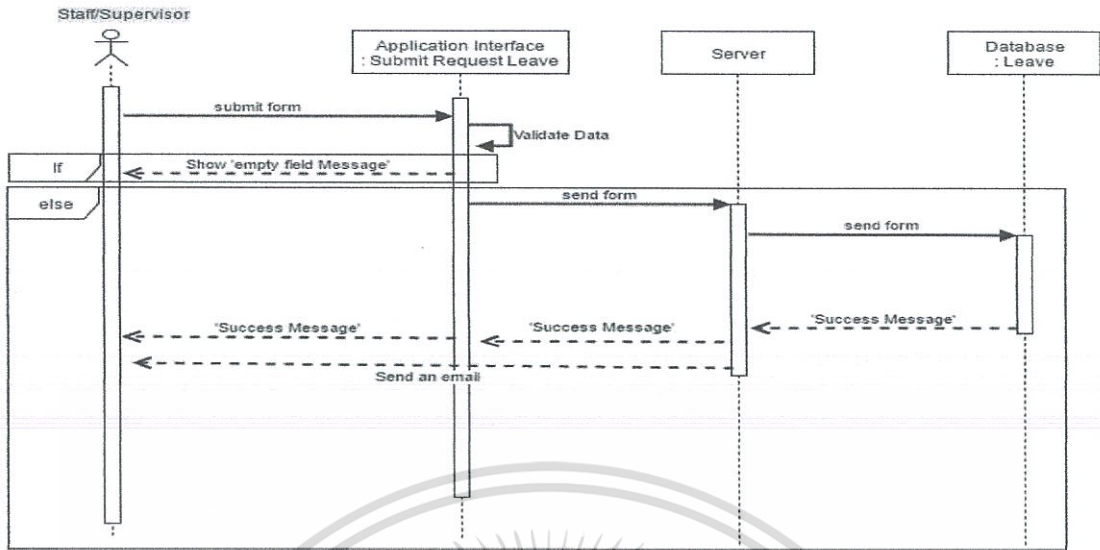
ภาพที่ 3.4.2 แผนภาพลำดับเหตุการณ์ระบบตรวจสอบจำนวนวันลาที่เหลือของระบบ Leave Request เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 3.4.2 แสดงถึงลำดับเหตุการณ์การทำงานของระบบ Leave Request ในส่วนของการตรวจสอบจำนวนวันลาที่เหลือของผู้ใช้ เมื่อผู้ใช้เข้าสู่หน้าตรวจสอบจำนวนวันลาที่เหลือทั้งหลังจากการลือคอิน หรือแม้แต่การเข้าแอปพลิเคชันมาอีกครั้งหลังจากที่เคยลือคอินไว้แล้ว แอปพลิเคชันก็จะนำผู้ใช้เข้าสู่หน้าตรวจสอบจำนวนวันลาที่เหลือ และระบบจะมีการตรวจสอบและดึงข้อมูลจากระบบฐานข้อมูลมาคำนวณจำนวนวันลาที่เหลือ ก่อนนำมาแสดงผ่านหน้าตรวจสอบจำนวนวันลาที่เหลือให้กับผู้ใช้



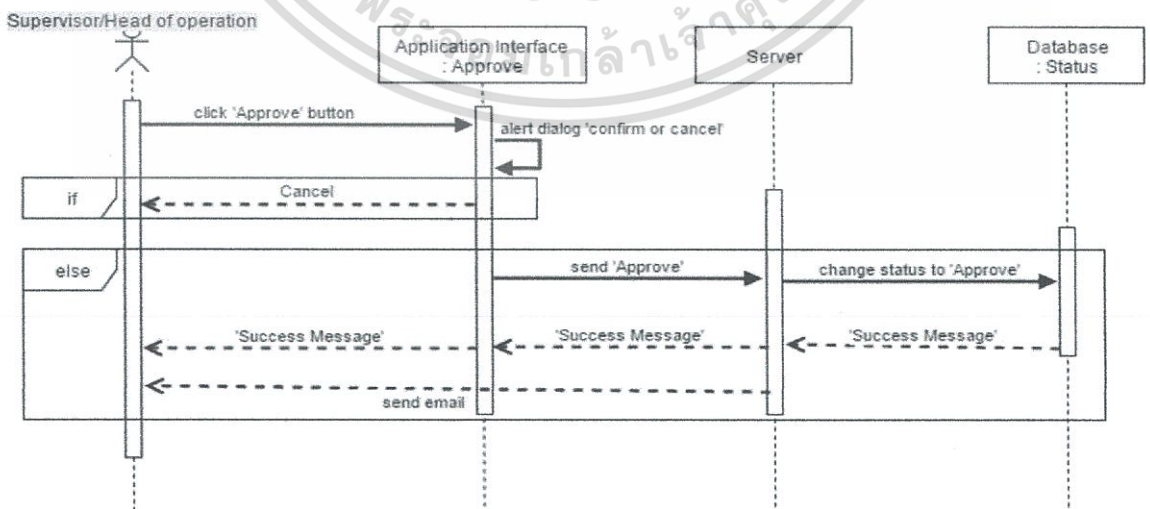
ภาพที่ 3.4.3 แผนภาพลำดับเหตุการณ์ระบบตรวจสอบวันลาที่ผ่านมาของระบบ Leave Request

จากภาพที่ 3.4.3 จะเป็นแผนภาพลำดับเหตุการณ์ระบบตรวจสอบวันลาที่ผ่านมาของผู้ใช้ เมื่อผู้ใช้เข้าสู่หน้าตรวจสอบวันลาที่ผ่านมา ระบบจะทำการตรวจสอบข้อมูลของวันลาที่ผ่านมาของผู้ใช้รายนั้นภายในระบบฐานข้อมูล ก่อนจะส่งกลับมาให้ทางแอปพลิเคชันแสดงในหน้าตรวจสอบวันลาที่ผ่านมาให้แก่ผู้ใช้ต่อไป



ภาพที่ 3.4.4 แผนภาพลำดับเหตุการณ์ระบบส่งคำขอลาของระบบ Leave Request

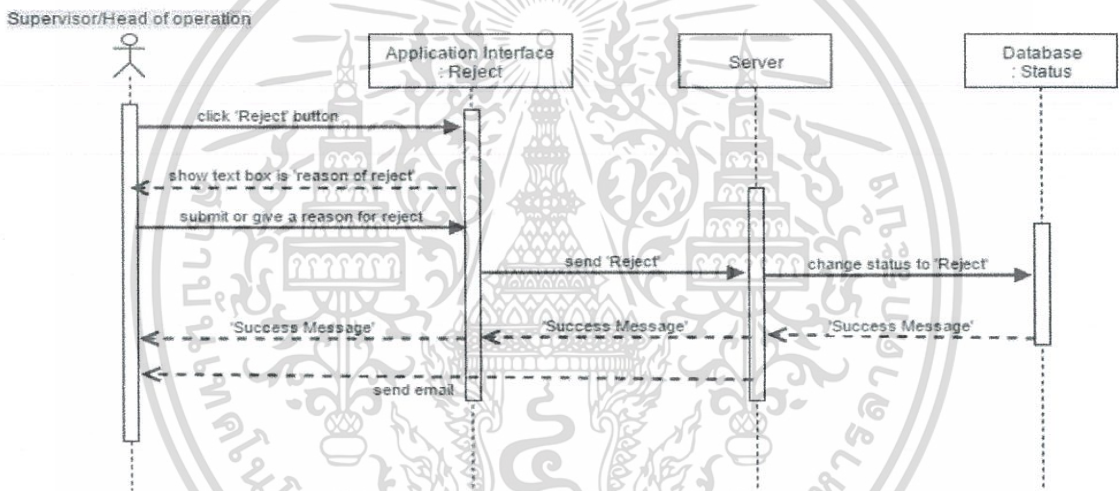
จากภาพที่ 3.4.4 จะเป็นแผนภาพลำดับเหตุการณ์ระบบส่งคำขอลา โดยเมื่อผู้ใช้เข้าสู่หน้ากรอกแบบฟอร์มคำขอลาแล้ว ผู้ใช้จะต้องทำการกรอกข้อมูลภายในแบบฟอร์มคำขอลาให้ครบถ้วน ก่อนกดปุ่มส่งคำขอลา หากมีข้อมูลใดภายในแบบฟอร์มคำขอลาที่ไม่ถูกกรอก จะมีข้อความขึ้นมาเตือนผู้ใช้ให้กรอกข้อมูลให้ครบถ้วนและถูกต้อง หลังจากผู้ใช้กรอกข้อมูลอย่างถูกต้องและครบถ้วนและกดปุ่มส่งคำขอลาแล้ว ระบบจะส่งคำขอลาดังกล่าวเข้าสู่ระบบฐานข้อมูล เมื่อคำขอลาถูกส่งเข้าสู่ระบบฐานข้อมูลแล้วจะมีข้อความแจ้งเตือนผ่านหน้าแอปพลิเคชันว่าคำขอลาได้ถูกส่งแล้ว และจะมีการส่งอีเมลไปแจ้งผู้ใช้งานที่มีอำนาจในการอนุญาตให้ลาได้ เพื่อให้พิจารณาคำขอลาที่ถูกส่งมาถึง



ภาพที่ 3.4.5 แผนภาพลำดับเหตุการณ์การอนุญาตให้ลาได้ของระบบ Leave Request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 3.4.5 จะเป็นแผนภาพลำดับเหตุการณ์การอนุญาตให้ลาได้ของระบบ Leave Request ที่เมื่อผู้ที่มีอำนาจอนุญาตให้ลาได้กดเข้ามาในหน้าแสดงรายละเอียดของคำขอลาที่ส่งมาถึงแล้ว จะสามารถพิจารณารายละเอียดต่างๆในคำขอลานั้นและหากต้องการอนุญาตให้ลาได้ผู้ใช้สามารถกดปุ่ม “อนุญาต” หลังจากนั้นจะมีข้อความขึ้นมาเพื่อให้ผู้ใช้กด “ยืนยัน” หรือ “ยกเลิก” การอนุญาตนี้เพื่อกลับไปพิจารณาคำขอลาอีกครั้งได้ ในกรณีที่ผู้ใช้กดปุ่ม “ยืนยัน” ระบบจะทำการส่งสถานะ “อนุญาต” ในกรณีที่ผู้ใช้ที่กดปุ่มอนุญาตนี้เป็น Supervisor หรือส่งสถานะ “ยืนยัน” ในกรณีที่ผู้ใช้ที่กดปุ่มอนุญาตนี้เป็น Head of operation ซึ่งสถานะดังกล่าวนี้จะถูกส่งเข้าไปเก็บไว้ในระบบฐานข้อมูล เมื่อมีการเก็บสถานะดังกล่าวลงระบบฐานข้อมูลได้อย่างถูกต้องครบถ้วนแล้ว จะมีการส่งข้อความผ่านทางระบบ เพื่อบอกว่าการเก็บข้อมูลลงระบบฐานข้อมูลสำเร็จ และแอปพลิเคชันจะมีข้อความเพื่อแจ้งให้ผู้ใช้ทราบว่า การอนุญาตนี้สำเร็จแล้ว

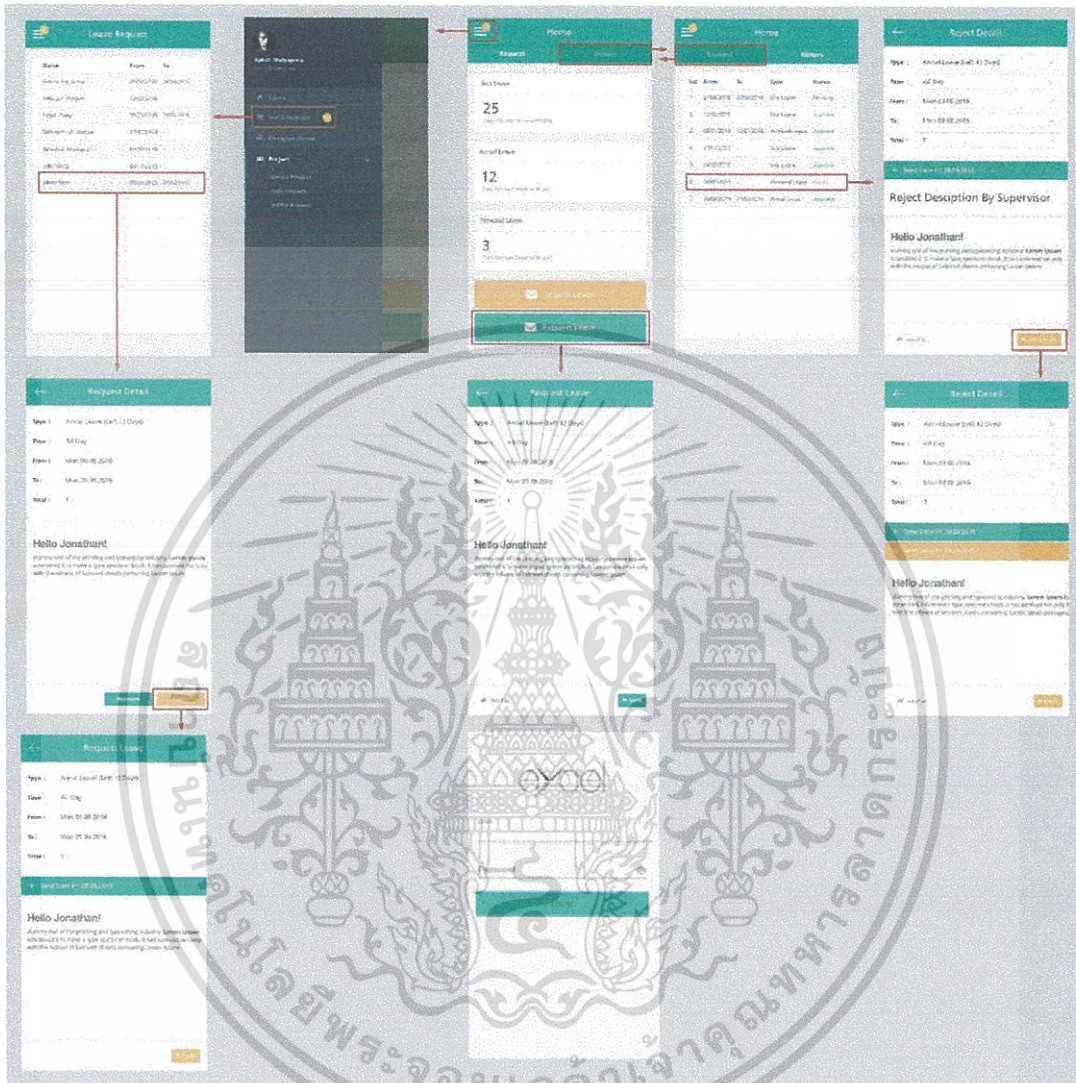


ภาพที่ 3.4.6 แผนภาพลำดับเหตุการณ์การไม่อนุญาตให้ลาของระบบ Leave Request

จากภาพที่ 3.4.6 คือแผนภาพลำดับเหตุการณ์การไม่อนุญาตให้ลาของระบบ Leave Request ที่หลังจากผู้ใช้พิจารณาคำขอลาที่ถูกส่งเข้ามาแล้ว และไม่ต้องการอนุญาตให้ลา ผู้ใช้สามารถกดปุ่ม “ไม่อนุญาต” หลังจากนั้นแอปพลิเคชันจะแสดงช่องให้กรอกเหตุผลที่ไม่อนุญาตให้ลา หลังจากกรอกเหตุผลที่ไม่อนุญาตให้ลาแล้ว ผู้ใช้สามารถกดปุ่มตอบกลับเพื่อส่งคำอนุมัติว่าไม่อนุญาตให้ลานั้นไปยังระบบฐานข้อมูล เมื่อมีการเก็บข้อมูลทั้งหมดลงระบบฐานข้อมูลอย่างครบถ้วนแล้ว จะมีการตอบกลับจากระบบฐานข้อมูลมายังแอปพลิเคชันเพื่อแจ้งว่าได้เก็บข้อมูลเข้าสู่ระบบอย่างถูกต้องครบถ้วนแล้ว และจะมีอีเมลล์ไปแจ้งเตือนผู้ที่ส่งคำขอลาว่าคำขอลานั้นๆ ไม่ได้รับการอนุญาตให้ลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบส่วนติดต่อผู้ใช้



ภาพที่ 3.5.1 แผนภาพแสดงส่วนติดต่อผู้ใช้ของแอปพลิเคชัน Leave Request ทั้งหมด

จากภาพที่ 3.5.1 จะเป็นการออกแบบส่วนติดต่อผู้ใช้ของแอปพลิเคชัน Leave Request ทั้งหมด เมื่อผู้ใช้เข้าใช้งานแอปพลิเคชันครั้งแรกจะเข้าสู่หน้าล็อกอิน เพื่อให้ผู้ใช้ทำการล็อกอินเข้าสู่ระบบ เมื่อผู้ใช้ล็อกอินเข้าสู่ระบบสำเร็จ แอปพลิเคชันจะนำผู้ใช้เข้าสู่หน้าตรวจสอบจำนวนวันลาที่เหลือซึ่งเป็นหน้าหลักของแอปพลิเคชัน และหากผู้ใช้ไม่ได้ทำการล็อกเอาต์ออกจากระบบ เมื่อผู้ใช้เปิดใช้งานแอปพลิเคชัน ครั้งต่อไปแอปพลิเคชันก็จะนำผู้ใช้มาสู่หน้าหลักนี้ทันที ในหน้าตรวจสอบจำนวนวันลาที่เหลือนี้ จะแสดงจำนวนวันลาที่เหลือของผู้ใช้โดยแบ่งเป็นวันลาแต่ละประเภทหลักๆ ดังนี้ วันลาป่วยที่เหลือ (Sick Leave), วันลาพักร้อนที่เหลือ (Annual Leave) และ วันลากิจที่เหลือ (Personal Leave) และในหน้าหลักนี้ผู้ใช้สามารถกดปุ่มเข้าสู่หน้ากรอกแบบฟอร์มคำขอลาได้จากปุ่ม “ส่งคำขอลา” ที่อยู่ทางด้านล่างของเอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอ ในหน้ากรอกแบบฟอร์มจะมีช่องให้ผู้กรอกรายละเอียดของคำขอลา โดยรายละเอียดที่กรอกจะ เป็นไปตามแบบฟอร์มคำขอลาที่ทางองค์กรกำหนด โดยจะมีรายละเอียด ดังนี้

- ประเภทของวันลาที่ต้องการใช้ลา
- ช่วงเวลาที่ต้องการลา
- วันที่ต้องการเริ่มลา
- วันสุดท้ายที่ต้องการลา
- เหตุผลที่ต้องการลา

เมื่อผู้ใช้กรอกรายละเอียดจนครบ จึงสามารถกดปุ่มส่งคำขอลาที่อยู่ทางมุมล่างขวาได้เพื่อ ส่งคำขอลาไปยังระบบ

จากหน้าหลักหรือหน้าตรวจสอบจำนวนวันลาที่เหลือผู้ใช้สามารถเลื่อนไปยังแท็บบาร์ ซ่างๆ เพื่อเข้าสู่หน้าตรวจสอบวันลาที่ผ่านมาได้ โดยในหน้าแสดงวันลาที่ผ่านมาจะมีการแสดงวันลาที่ผ่านมา ของผู้ใช้เป็นแบบลิสต์ และสามารถกดเข้าไปดูรายละเอียดแบบสมบูรณ์ของการขอลาแต่ละครั้งได้โดย การกดที่วันที่ ของวันลาที่ต้องการดูได้เลย แล้วแอปพลิเคชันจะนำไปสู่หน้าแสดงรายละเอียดวันลาที่ผ่านมา

ผู้ใช้สามารถเรียกใช้ Navigation drawer ได้โดยการสไลด์หน้าจอจากทางด้านซ้ายมือไป ทางด้านขวามือ เพื่อเรียกใช้เมนู ล็อคเอาต์ หรือสำหรับผู้ใช้ที่มีตำแหน่งเป็น Supervisor หรือ Head of operation จะสามารถเข้าดูคำขอลาที่เข้ามาได้จากเมนู “คำขอลาที่เข้ามา” ได้ด้วย

ในหน้า “คำขอลาที่เข้ามา” จะมีการแสดงคำขอลาที่ส่งเข้ามาเป็นลิสต์เช่นเดียวกับหน้า แสดงลิสต์ “วันลาที่ผ่านมา” และผู้ใช้สามารถเข้าไปดูรายละเอียดของแต่ละคำขอลาที่เข้ามาได้โดยการกด ที่แต่ละคำขอลาที่ต้องการดูรายละเอียดได้เลย หลังจากนั้นแอปพลิเคชันจะแสดงหน้ารายละเอียดของคำขอลา ที่ถูกกดเข้าไปซึ่งจะมีรายละเอียดตามหน้ากรอกแบบฟอร์มคำขอลาทุกประการ เพียงแต่ผู้ใช้จะไม่สามารถแก้ไขรายละเอียดในช่องต่างๆได้ และในหน้านี้ผู้ใช้จะสามารถตัดสินใจได้ว่าจะอนุญาตคำขอลานี้ หรือไม่ โดยการกดที่ปุ่ม “ยืนยัน” หรือ “ปฏิเสธ” ได้ทางมุมขวาล่างของจอ ในกรณีที่กดปุ่ม “ปฏิเสธ” แอปพลิเคชันจะแสดงช่องให้กรอกเหตุผลที่ปฏิเสธคำขอลาหลังจากนั้นผู้ใช้สามารถกดปุ่ม “ตอบกลับ” เพื่อ ทำการปฏิเสธคำขอลา

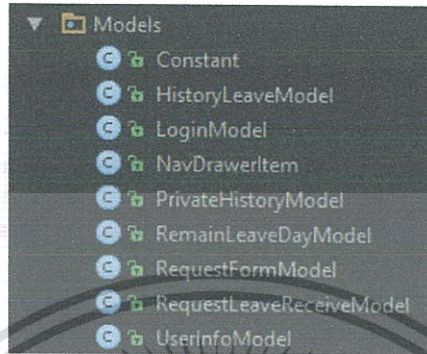
3.6 การออกแบบแอปพลิเคชันด้วยโครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC

ในขั้นตอนการพัฒนาแอปพลิเคชันได้มีการนำโครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC (Model-View-Controller) มาประยุกต์ใช้เพื่อให้การทำความเข้าใจและตรวจสอบโค้ดเป็นไป ได้โดยง่าย ซึ่งได้แบ่งไฟล์ต่างๆออกเป็น 3 กลุ่ม ดังนี้

- จากภาพที่ 3.6.1 แสดงถึง Model หรือไฟล์ที่ทำหน้าที่เป็นเหมือนอ็อบเจกต์ที่ทำการเก็บ ไฟล์ข้อมูลที่มีการรับและส่งระหว่าง Web API และแอปพลิเคชัน หรือเมื่อมีการติดต่อ

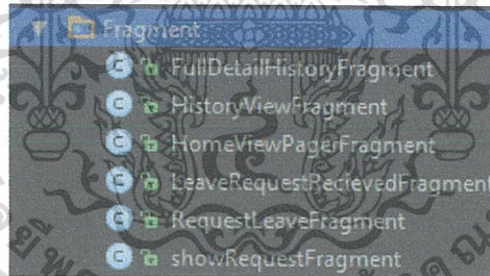
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Controller ให้รับและส่งข้อมูลที่เกี่ยวข้องในการประมวลผล ซึ่งข้อมูลส่วนนี้ก็จะถูกเก็บอยู่ในไฟล์ Model ทั้งสิ้น



ภาพที่ 3.6.1 ไฟล์กลุ่ม Model ของแอปพลิเคชัน Leave Request

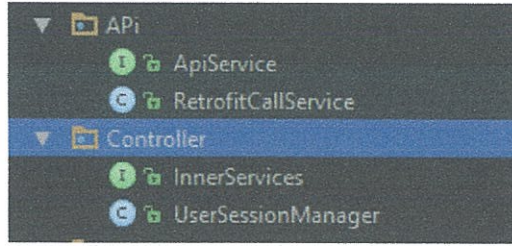
- จากภาพที่ 3.6.2 แสดงถึง View เป็นไฟล์ที่ทำหน้าที่เป็นส่วนติดต่อกับผู้ใช้ของแอปพลิเคชัน เพื่อรับและแสดงผลให้กับผู้ใช้เข้าใจได้



ภาพที่ 3.6.2 ไฟล์ Fragment ที่ทำหน้าที่เป็น View ของแอปพลิเคชัน Leave Request

- จากภาพที่ 3.6.3 แสดงถึง Controller ไฟล์ที่ทำหน้าที่เป็นส่วนรับข้อมูลมาเพื่อประมวลผลก่อนจะส่งข้อมูลที่ประมวลผลแล้วกลับไปยัง View เพื่อให้แสดงผลต่อผู้ใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6.3 ไฟล์ที่ทำหน้าที่เป็น Controller หลักของแอปพลิเคชัน Leave Request

สำหรับแอปพลิเคชัน Leave Request จะมีรายละเอียดและหน้าที่ของแต่ละไฟล์ ดังนี้

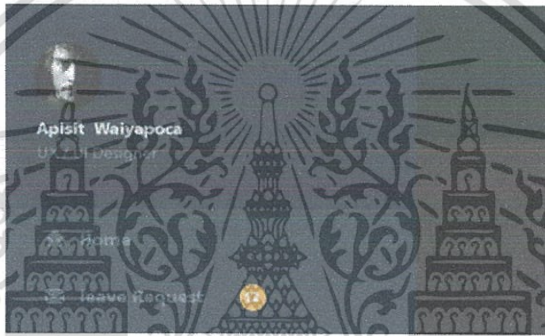
ไฟล์กลุ่ม Model

- Constant ทำหน้าที่เก็บค่าคงที่ต่างๆที่มีการเรียกใช้งานค่าคงที่นั้นบ่อยๆ เช่น ค่าหน่วยเวลา (Delay) เป็นเวลา 2 วินาที จะมีการเก็บเป็นค่าคงที่ 2000 ms เป็นต้น
- HistoryLeaveModel ทำหน้าที่รับและเก็บค่าข้อมูลที่ได้รับมาจาก Web API โดยเป็นข้อมูลของประวัติการขอลาที่ผ่านมา และจะต้องมีการทำงานร่วมกับไฟล์ FullDetailHistoryViewFragment (View) หรือร่วมกับไฟล์ HistoryViewFragment (View) เพื่อนำไปแสดงผลให้กับผู้ใช้ โดยจะมีการเรียกใช้งาน Web API ผ่านไฟล์ RetrofitCallService (Controller) ทุกครั้งที่ต้องการเรียกใช้ข้อมูลจากระบบฐานข้อมูล
- LoginModel เก็บข้อมูลเกี่ยวกับการเข้าสู่ระบบของผู้ใช้เช่น Token, Email ของผู้ใช้ เพื่อทำการเก็บสถานะการเข้าสู่ระบบของผู้ใช้ไว้เมื่อผู้ใช้เข้าใช้งานแอปพลิเคชันครั้งต่อไปจะมีการดึงข้อมูลส่วนนี้ไปตรวจสอบว่าผู้ใช้เคยทำการเข้าสู่ระบบไว้แล้วหรือไม่ โดยจะมีการทำงานร่วมกับไฟล์ UserLoginActivity (View และ Activity) เพื่อตรวจสอบว่าผู้ใช้เคยเข้าสู่ระบบแล้วหรือไม่ หากไม่เคยจะแสดงหน้าแรกที่ใช้เข้าสู่ระบบให้กับผู้ใช้ และจะมีการทำงานร่วมกับไฟล์ RetrofitCallService (Controller) เพื่อเรียกใช้งาน Web API ที่ใช้สำหรับการเข้าสู่ระบบ
- PrivateHistoryModel เป็นไฟล์ Model ที่ใช้ทดสอบกับ Mock-up ของ Web API ในช่วงเริ่มต้นพัฒนาแอปพลิเคชัน ปัจจุบันไม่ได้ใช้งานแล้วภายในแอปพลิเคชัน
- RemainLeaveDayModel ทำหน้าที่รับข้อมูลเกี่ยวกับจำนวนวันลาที่เหลือของผู้ใช้ และเก็บบันทึกไว้ โดยจะมีการทำงานร่วมกับไฟล์ RetrofitCallService (Controller) ที่จะเรียกใช้งาน Web API สำหรับนำข้อมูลจากระบบฐานข้อมูลมาแสดงผลร่วมกับไฟล์ RequestLeaveFragment (View)
- RequestFormModel ทำหน้าที่ส่งข้อมูลเกี่ยวกับคำขอลาไปยังระบบฐานข้อมูลโดยเมื่อผู้ใช้กรอกคำขอลาครบถ้วนสมบูรณ์และกดปุ่มส่งคำขอลาจากหน้าแบบฟอร์มคำขอลาผ่านไฟล์ RequestFormActivity (View และ Activity) ก็จะมีการส่งรายละเอียดคำขอลาทั้งหมดผ่านไฟล์ RequestFormModel แล้วส่งต่อไปยังระบบฐานข้อมูลผ่าน Web API

เอกสารนี้เป็นเอกสารที่โดยการเรียกใช้ผ่านไฟล์ RetrofitCallService (Controller) ที่นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RequestLeaveReceiveModel ทำหน้าที่รับข้อมูลเกี่ยวกับคำขอลาที่พนักงานได้ส่งมาถึงหัวหน้า โดยจะทำงานร่วมกับไฟล์ LeaveRequestRecievedFragment (View) หรือไฟล์ showRequestFragment (View) เพื่อนำข้อมูลเกี่ยวกับคำขอลาที่ได้รับมาแสดงผลให้ผู้ใช้ ซึ่งจะต้องเรียกข้อมูลดังกล่าวผ่าน Web API ด้วยไฟล์ RetrofitCallService (Controller)
- UserInfoModel ทำหน้าที่รับข้อมูลของผู้ใช้มาเก็บไว้เพื่อให้ไฟล์ MainActivity (Activity) นำข้อมูลของผู้ใช้ เช่น ชื่อ-สกุล, ตำแหน่งงาน เป็นต้น ไปแสดงบน Navigation Drawer ตามภาพที่ 3.6.4 โดยข้อมูลของผู้ใช้จะถูกเรียกใช้ได้จาก Web API ผ่านไฟล์ RetrofitCallService (Controller)



ภาพที่ 3.6.4 ข้อมูลของผู้ใช้ที่นำมาแสดงผลบน Navigation Drawer

ไฟล์กลุ่ม View

- FullDetailHistoryFragment หน้าแสดงรายละเอียดของคำขอลาที่ผ่านมาของผู้ใช้จะมีหน้าตาคล้ายๆ หน้ากรอกแบบฟอร์มคำขอลา แต่ไม่สามารถแก้ไขรายละเอียดต่างๆ ได้ โดยจะมีการเรียกข้อมูลจากไฟล์ HistoryLeaveModel (Model) มาแสดง
- HistoryViewFragment หน้าแสดงลิสต์ของคำขอลาที่ผ่านมาของผู้ใช้ โดยจะมีการนำข้อมูลจากไฟล์ HistoryLeaveModel (Model) มาแสดง โดยการแสดงข้อมูลเป็นลิสต์นั้น จะทำการกำหนดการแสดงผลต่างๆ ผ่านไฟล์ PagerAdapter ที่นับเป็นไฟล์ Controller สำหรับจัดการการแสดงผลเป็นลิสต์สำหรับไฟล์ HistoryViewFragment นี้
- HomeViewPagerFragment เป็นหน้าที่แสดงผลและจัดการเรื่องของการแสดงผลแบบ ViewPager ที่แสดงผลบน MainActivity อีกที่
- LeaveRequestRecievedFragment เป็นหน้าที่แสดงลิสต์ของคำขอลาที่ถูกส่งมาถึงหัวหน้า โดยมีการเรียกใช้ Web API ผ่านไฟล์ RetrofitCallService (Controller) เพื่อนำข้อมูลบันทึกเข้าสู่ไฟล์ RequestLeaveReceiveModel (Model) ก่อนนำมาแสดงผลผ่านไฟล์ LeaveRequestRecievedFragment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RequestLeaveFragment เป็นหน้าแสดงจำนวนวันลาที่เหลือของผู้ใช้ โดยจะเรียกข้อมูลวันลาที่เหลือผ่านไฟล์ RetrofitCallService (Controller) แล้วจัดเก็บลงในไฟล์ RemainLeaveDayModel (Model) แล้วจึงนำข้อมูลที่ได้นี้มาแสดงผลผ่านไฟล์ RequestLeaveFragment
- showRequestFragment เป็นหน้าแสดงรายละเอียดคำขอลาที่ถูกส่งมาถึงหัวหน้า โดยจะรับข้อมูลจากไฟล์ RequestLeaveReceiveModel (Model) มาแสดงสำหรับหน้านี้ ผู้ใช้จะสามารถกดปุ่มอนุญาตหรือไม่อนุญาตให้ลาได้ ซึ่งเมื่อมีการกดปุ่มใดปุ่มหนึ่งจะมีการส่งสถานะคำขอลาไปยังระบบฐานข้อมูลผ่าน Web API โดยเรียกใช้ผ่านไฟล์ RetrofitCallService (Controller)

ไฟล์กลุ่ม Controller

- ApiService เป็นไฟล์ที่เก็บชื่อของ Web API ที่จำเป็นต้องใช้งานทั้งหมด และทำหน้าที่ระบุชื่อ Web API ที่ถูกเรียกใช้ผ่านไฟล์ RetrofitCallService (Controller) ที่ทำหน้าที่เรียกใช้ Web API
- RetrofitCallService เป็นไฟล์ที่ทำหน้าที่เรียกใช้งาน Web API โดยจะต้องทราบชื่อ Web API ที่ต้องการเรียกใช้ โดยรายชื่อ Web API ที่เรียกใช้งานได้ทั้งหมดภายในแอปพลิเคชันสามารถดูได้จากไฟล์ ApiService
- InnerService เป็นไฟล์ Controller ที่ทำหน้าที่คอยจัดการการแสดงผลของปุ่มกด, Header หรือปุ่มกด ที่มีการเรียกใช้และแสดงผลบ่อยๆ แต่มีการแสดงผลแตกต่างกันไปภายในแอปพลิเคชัน เช่น เมื่อเปลี่ยนจากหน้าแสดงจำนวนวันลาที่เหลือไปยังหน้าแสดงคำขอลาที่ถูกส่งมาถึงหัวหน้า ส่วนของ Header จะเปลี่ยนจาก “Home” เป็น “Leave Request” ก็จะมีการเรียกใช้งานไฟล์ InnerService
- UserSessionManager เป็นไฟล์ที่จะตรวจสอบสถานะการเข้าสู่ระบบของผู้ใช้ว่าผู้ใช้เคยเข้าสู่ระบบไว้แล้วหรือไม่ หากเข้าสู่ระบบไว้แล้วจะมีการส่งค่าไปบอกยังไฟล์ MainActivity (Activity) เพื่อนำผู้ใช้เข้าสู่หน้าแรกของแอปพลิเคชัน ถ้าผู้ใช้ไม่ได้ทำการเข้าสู่ระบบค้างไว้ จะมีการส่งค่าไปยัง MainActivity (Activity) ก่อนจะนำผู้ใช้ไปสู่หน้าเข้าสู่ระบบเพื่อทำการเข้าสู่ระบบ

บทที่ 4

ผลการวิจัย

หลังจากได้เก็บข้อมูลที่จำเป็นต่อการสร้างระบบและแอปพลิเคชัน หลังจากนั้นจึงได้ดำเนินการวิจัยตามแผนที่ได้วางไว้ และพัฒนาระบบควบคู่กับแอปพลิเคชันไปพร้อมๆกันตามข้อมูลที่ได้เก็บมาในช่วงต้นของการดำเนินงานวิจัย และในท้ายที่สุดจึงได้แอปพลิเคชันจัดการระบบวันลาที่สามารถใช้งานได้ ดังนี้

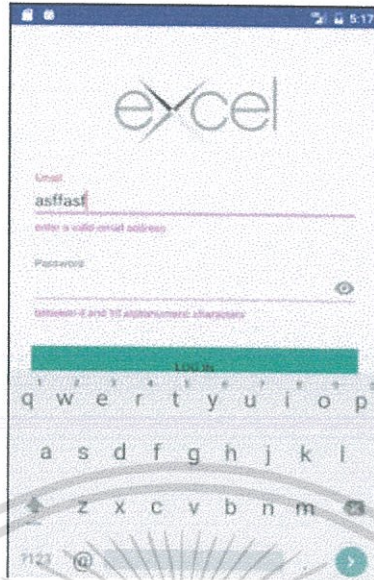
เมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชันครั้งแรก ผู้ใช้จะถูกนำมาสู่หน้าล็อกอินเข้าใช้ระบบ ดังภาพที่ 4.1



ภาพที่ 4.1 หน้าล็อกอินของแอปพลิเคชัน

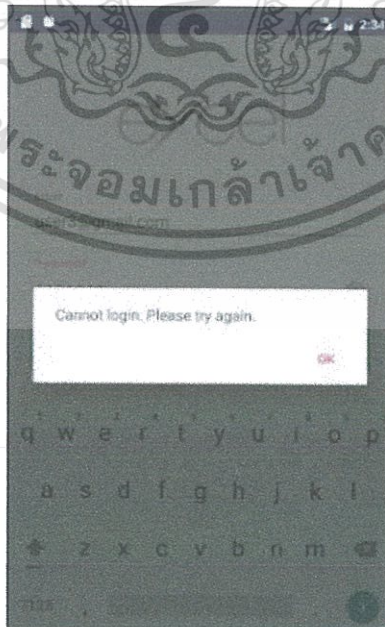
ในหน้าล็อกอินจะมีช่องเพื่อให้ผู้ใช้กรอก username และ password เพื่อทำการล็อกอินเข้าระบบ ถ้าหากผู้ใช้ไม่ได้ทำการกรอก username หรือ password หรือแม้แต่มิมีการกรอกแต่ไม่ถูกต้องตามรูปแบบมาตรฐานที่ทางแอปพลิเคชันกำหนดให้ username มีรูปแบบเป็นอีเมลล์ แอปพลิเคชันจะมีการแจ้งเตือนดังภาพที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.2 ข้อความแจ้งเตือนผู้ใช้งานไม่ได้กรอก password และ username ไม่ถูกต้องตามรูปแบบที่แอปพลิเคชันกำหนด

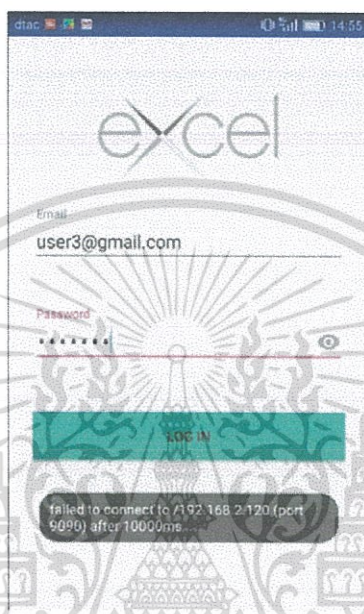
ถ้าหากผู้ใช้กรอก username และ password ถูกต้องตามรูปแบบที่แอปพลิเคชันกำหนดทุกอย่าง แต่ username และ password ไม่ตรงกับข้อมูลที่มีในระบบผู้ใช้งานจะไม่สามารถทำการล็อกอินเข้าสู่ระบบได้ และจะมีข้อความแจ้งเตือนกับผู้ใช้งานว่าไม่สามารถเข้าสู่ระบบได้ ดังรูปที่ 4.3



ภาพที่ 4.3 ข้อความแจ้งเตือนว่าไม่สามารถล็อกอินเข้าสู่ระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ผู้ใช้กรอกข้อมูลถูกต้องครบถ้วนแล้วกดปุ่มล็อกอิน แต่ทางเซิร์ฟเวอร์มีปัญหาเกิดขึ้น แอปพลิเคชันจะมีข้อความแจ้งเตือนถึงปัญหาที่เกิดขึ้นทางฝั่งเซิร์ฟเวอร์กลับมา ดังภาพที่ 4.4 ซึ่งเป็นข้อความที่แจ้งเตือนปัญหาว่าได้ทำการเชื่อมต่อและรอการตอบรับจากทางฝั่งเซิร์ฟเวอร์จนครบกำหนดเวลา (Time out) แต่ก็ไม่มีการตอบรับจากทางฝั่งเซิร์ฟเวอร์แต่อย่างใด



ภาพที่ 4.4 ข้อความแจ้งเตือนถึงปัญหาที่เกิดขึ้นทางฝั่งเซิร์ฟเวอร์

เมื่อผู้ใช้ทำการล็อกอินเข้าสู่ระบบได้สำเร็จ แอปพลิเคชันจะนำผู้ใช้เข้าสู่หน้าจอแสดงจำนวนวันลาที่เหลือ และแสดงข้อความที่แสดงถึงผลการตรวจสอบว่าผู้ใช้ทำการล็อกอินแล้วหรือไม่ ถ้ามีการล็อกอินแล้ว จะแสดงผลการตรวจสอบออกมาเป็น “true” ถ้าไม่ได้ล็อกอินจะแสดงเป็น “false” ซึ่งกรณีที่ผู้ใช้ทำการล็อกอินเข้าสู่ระบบสำเร็จแล้วจะมีการแสดงผลการตรวจสอบออกมาเป็น “true” เสมอ ดังภาพที่ 4.5 และแอปพลิเคชันจะมีผลการตรวจสอบเป็น “false” เมื่อผู้ใช้ทำการล็อกเอาต์ออกจากระบบ และทำการเปิดแอปพลิเคชันขึ้นมาใหม่

ในหน้าจอแสดงจำนวนวันลาที่เหลือนี้จะแบ่งจำนวนวันลาเป็นแต่ละประเภทวันลาอย่างชัดเจน และทางด้านล่างจะมีปุ่มสำหรับส่งคำขอลาที่เมื่อผู้ใช้ทำการกดปุ่มนี้ แอปพลิเคชันจะนำผู้ใช้เข้าสู่หน้าจอกรอกแบบฟอร์มคำขอลา ดังภาพที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

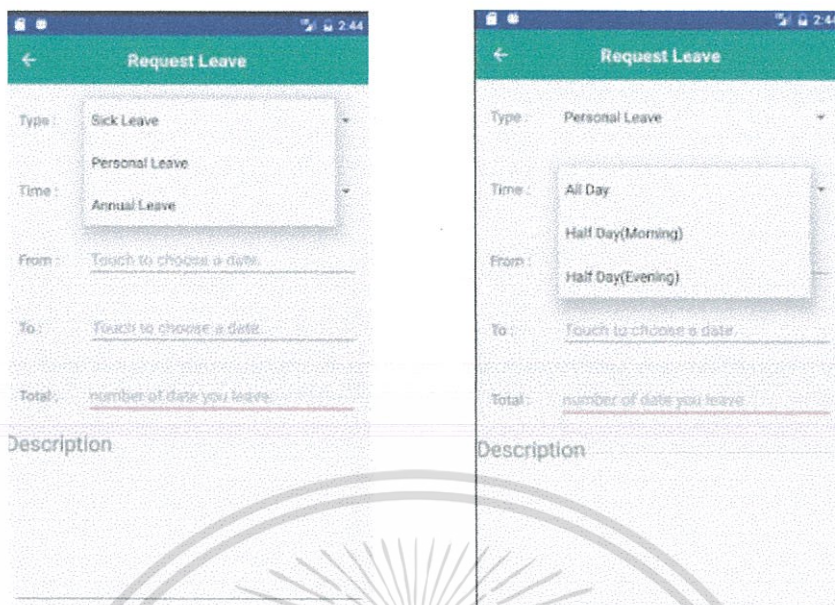


ภาพที่ 4.5 หน้าแสดงจำนวนวันลา และข้อความตรวจสอบสถานะการล็อกอินของผู้ใช้

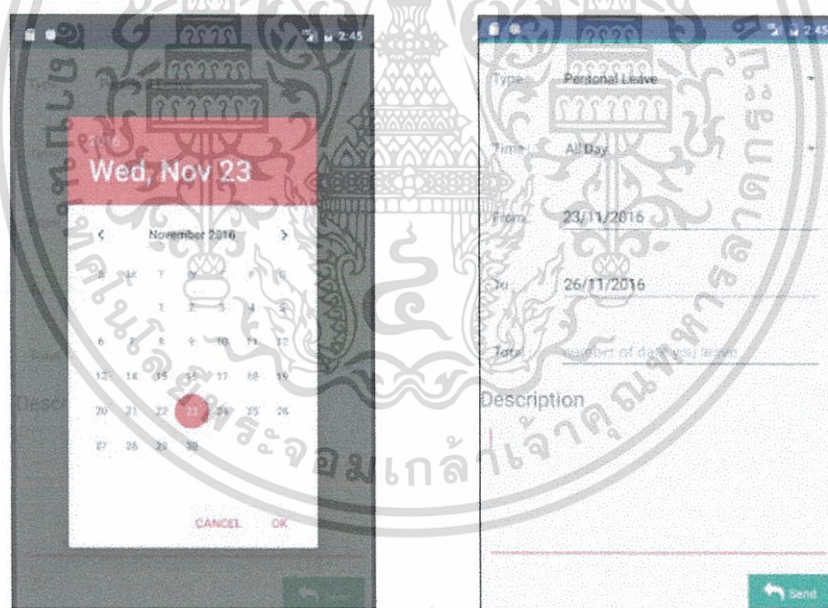


ภาพที่ 4.6 หน้ากรอกแบบฟอร์มคำขอลาเมื่อผู้ใช้กดปุ่ม “คำขอลา”

ในหน้ากรอกคำขอลาก็จะมีช่องให้กรอก ดังที่เคยกล่าวไปในบทที่ผ่านมา โดยช่องสำหรับกรอก “จำนวนวันที่ต้องการลาทั้งหมด (Total)” และ “เหตุผลที่ต้องการลา หรือ คำอธิบาย (Description)” ช่องให้พิมพ์กรอกรายละเอียดได้ด้วยตนเอง สำหรับช่องกรอก “ประเภทการลา (Type)” และ “ช่วงเวลาที่ต้องการลา (Time)” เมื่อผู้ใช้กดเข้าไปที่ช่องดังกล่าว จะมีตัวเลือกประเภท drop down list มาให้ผู้ใช้เลือกประเภทและช่วงเวลาที่ต้องการจะลา ดังภาพที่ 4.7 และสำหรับช่อง “วันแรกที่ต้องการลา (From)” และช่อง “วันสุดท้ายที่ต้องการลา (To)” เมื่อผู้ใช้กดที่ช่องดังกล่าวจะมีปฏิทินขึ้นมาให้ผู้ใช้กดเลือกวันที่ต้องการได้ทันทีจากปฏิทินนี้ โดยค่าเริ่มต้นเมื่อเปิดปฏิทินขึ้นมาจะเป็นวันเวลา ณ ขณะนั้น ดังภาพที่ 4.8 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ในทางอื่นไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



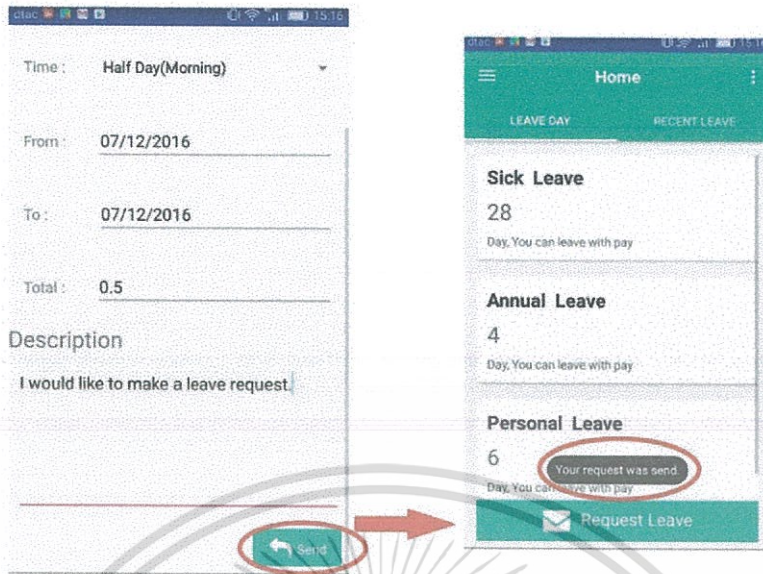
ภาพที่ 4.7 drop down list แสดงตัวเลือกประเภทวันลาและช่วงเวลาที่ต้องการลา



ภาพที่ 4.8 ปฏิทินที่ใช้สำหรับเลือกวันที่ต้องการลา

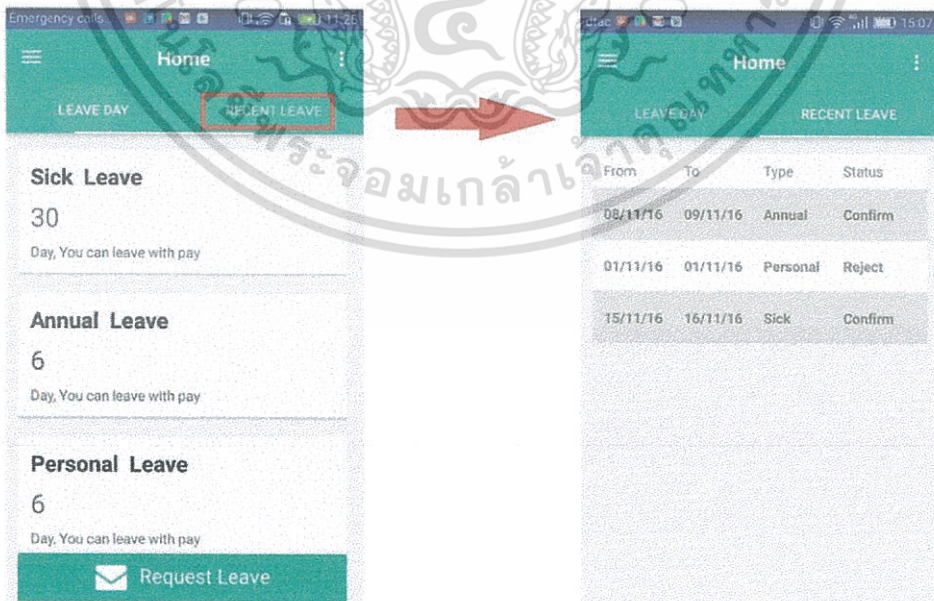
เมื่อผู้ใช้กรอกรายละเอียดตามแบบฟอร์มของคำขอลาจนครบถ้วนสมบูรณ์แล้ว ผู้ใช้จึงจะสามารถกดปุ่มส่งคำขอลาได้ และเมื่อคำขอลาถูกส่งออกไปแล้วเก็บเข้าสู่ระบบฐานข้อมูลได้สำเร็จ แอปพลิเคชันจะมีข้อความแจ้งกับผู้ใช้ว่า “คำขอได้ถูกส่งแล้ว (Your request was sent)” และนำผู้ใช้กลับมาสู่หน้าแสดงจำนวนวันลาที่เหลือ ดังภาพที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



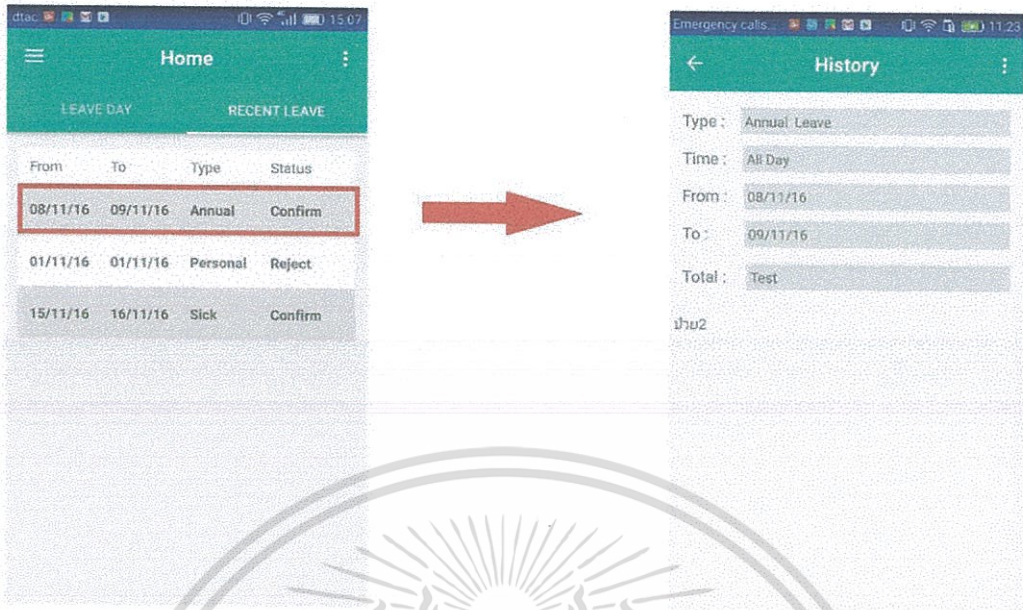
ภาพที่ 4.9 หน้ากรอกแบบฟอร์มที่มีรายละเอียดครบถ้วน และข้อความเมื่อส่งคำขอลาสำเร็จ

จากหน้าแสดงจำนวนวันลาผู้ใช้สามารถเลื่อนหน้าจออกจากทางขวาไปทางซ้าย หรือกดที่แท็บบาร์ “วันลาล่าสุด (Recent Leave)” ทางด้านบนของจอเพื่อเข้าสู่หน้าแสดงลิสต์วันลาล่าสุดได้ และจากหน้าแสดงลิสต์วันลาล่าสุด ผู้ใช้สามารถกดเข้าที่คำขอลาแต่ละครั้งเพื่อดูรายละเอียดการลาแต่ละครั้งได้ ดังภาพที่ 4.10



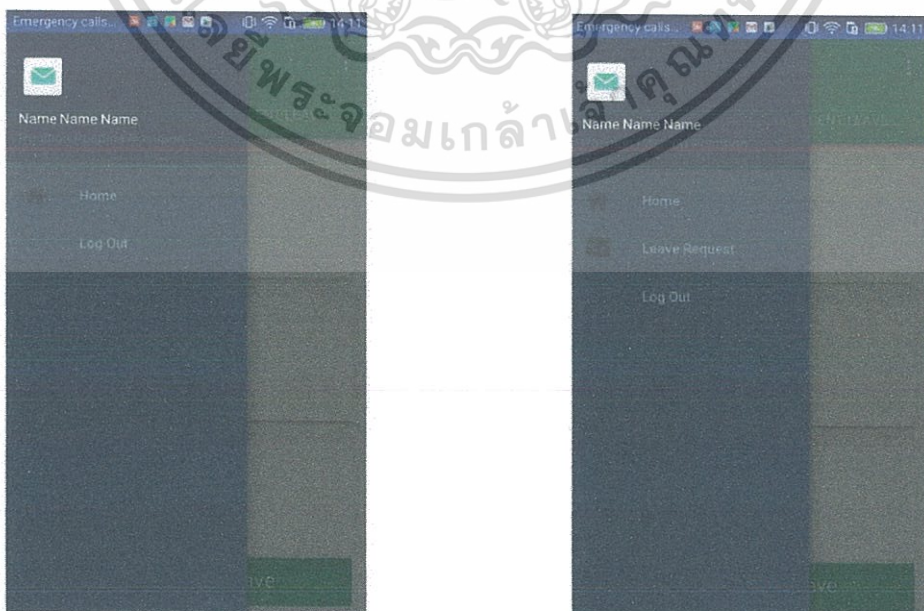
ภาพที่ 4.10 หน้าแสดงลิสต์คำขอลาล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.11 หน้ารายละเอียดคำขอลา

สำหรับผู้ที่มีตำแหน่งเป็น Supervisor และ Head of operation เมื่อเข้าสู่แอปพลิเคชันจะเจอหน้าตาแอปพลิเคชันเหมือนกับผู้ใช้ที่มีตำแหน่งเป็น Staff แต่เมื่อผู้ใช้ทำการเลื่อนแถบ Navigation drawer ออกมาจากทางด้านซ้ายของหน้าจอ จะมีเมนูที่เพิ่มเติมขึ้นมา คือเมนู “คำขอลา (Leave Request)” ซึ่งเป็นหน้าแสดงลิสต์ของคำขอลาที่ได้รับจาก Staff

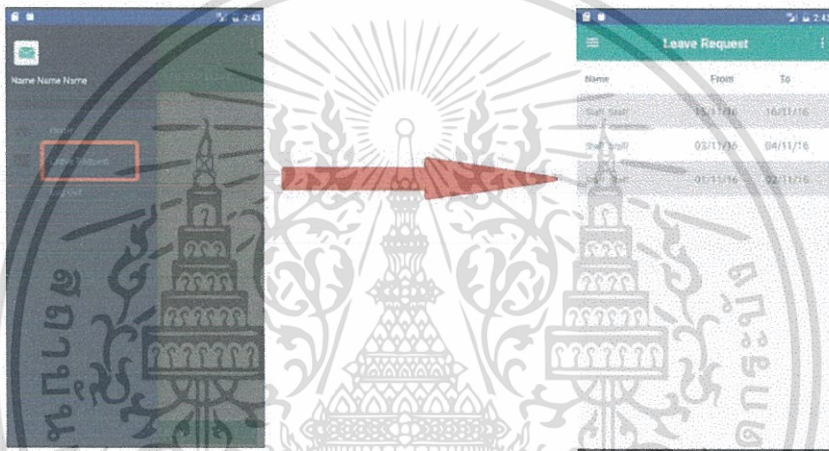


Staff

Supervisor & Head of operation

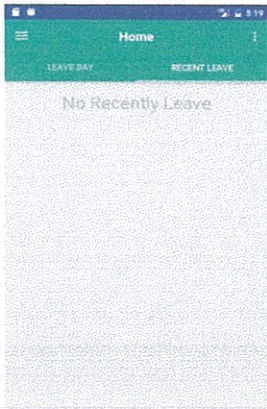
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยคณะกรรมาธิการการศึกษาระดับอุดมศึกษา กระทรวงศึกษาธิการ ไม่สามารถนำข้อมูลไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ที่มีตำแหน่งเป็น Supervisor หรือ Head of operation จะสามารถเข้าดูลิสต์ของคำขอลาที่ส่งเข้ามาได้โดยการกดที่เมนู “คำขอลา (Leave Request)” ที่อยู่ใน Navigation drawer ซึ่งคำขอลาที่แสดงในหน้าแอปพลิเคชันของผู้ใช้ที่มีตำแหน่งเป็น Supervisor จะเป็นคำขอลาที่บุคลากรที่อยู่ภายใต้การดูแลของ Supervisor คนนั้นๆ ส่งเข้ามา และสำหรับคำขอลาที่จะแสดงในหน้าแอปพลิเคชันของผู้ใช้ที่มีตำแหน่งเป็น Head of operation จะเป็นคำขอลาที่ Supervisor ได้ทำการอนุมัติให้คำขอลาที่มาจาก Staff นั้นสามารถลาได้ คำขอลาดังกล่าวจึงถูกส่งต่อมายัง Head of operation เพื่อให้ Head of operation ทำการพิจารณาเพื่อทำการอนุมัติขั้นสุดท้ายหรือปฏิเสธคำขอลา ตามแต่ Head of operation เห็นสมควร



ภาพที่ 4.13 หน้าแสดงลิสต์ของคำขอลาที่ส่งเข้ามาถึง Supervisor หรือ Head of operation

ถ้าหากในหน้าที่แสดงลิสต์ของคำขอลาที่ผ่านมาและหน้าแสดงลิสต์คำขอลาที่เข้ามา ไม่มีข้อมูลใดๆ ที่จะแสดง หรือกล่าวคือ ผู้ใช้อาจไม่เคยส่งคำขอลาหรือไม่มีประวัติการลาในระบบ สำหรับ Supervisor หรือ Head of operation อาจไม่มีคำขอลาที่ถูกส่งเข้ามาเลย แอปพลิเคชันจะมีการแสดงหน้าลิสต์ดังกล่าว ดังภาพที่ 4.14



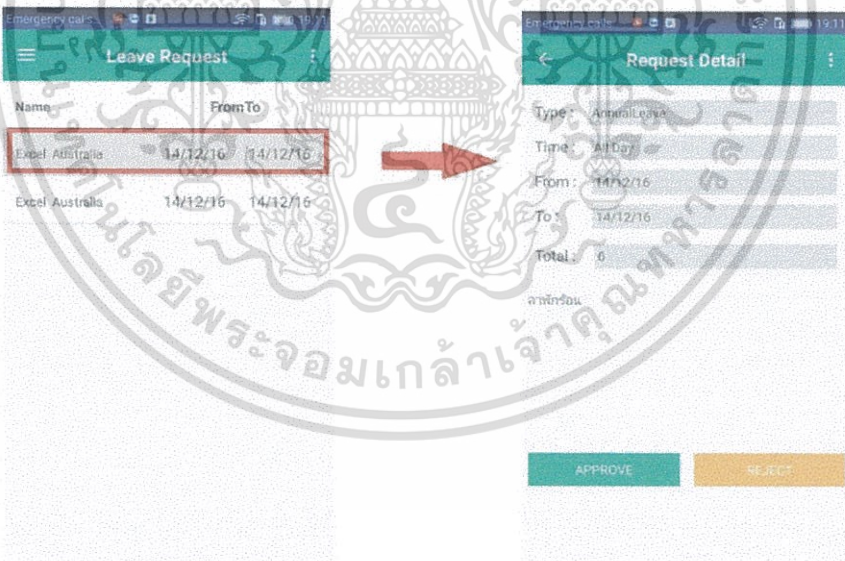
Recent Leave page



Leave Request page

ภาพที่ 4.14 หน้าประวัติการลาที่ผ่านมา และหน้าคำขอลาที่เข้ามา เมื่อไม่มีข้อมูล

เมื่อผู้ใช้ที่มีตำแหน่ง Supervisor หรือ Head of operation มีคำขอเข้ามา ผู้ใช้สามารถกดเข้าที่คำขอลาที่เข้ามาเพื่อดูรายละเอียดของคำขอลาแต่ละคำขอได้ ดังภาพที่ 4.15

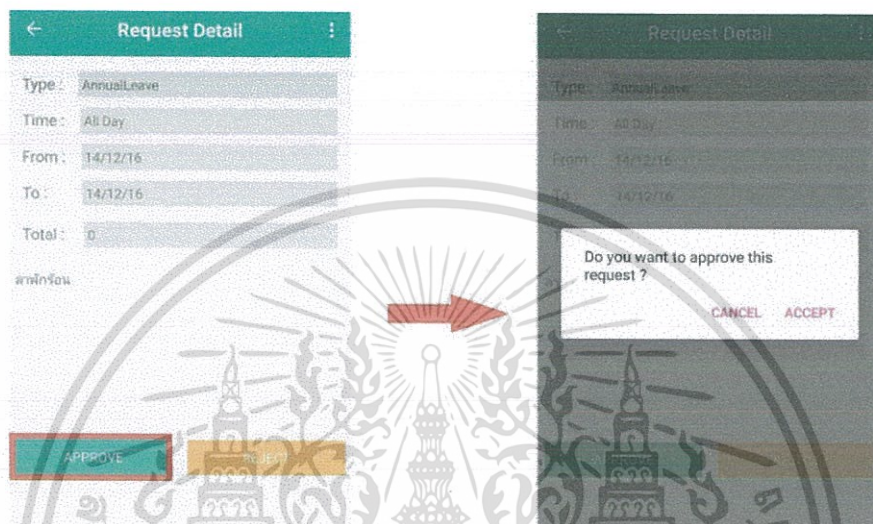


ภาพที่ 4.15 หน้าแสดงรายละเอียดของคำขอลาที่เข้ามา

เมื่อเข้าถึงหน้าแสดงรายละเอียดของคำขอลาแล้วผู้ใช้สามารถทำการตัดสินใจได้ว่าจะอนุญาตให้คำขอลานี้ผ่านหรือไม่ผ่านด้วยการ กดปุ่ม “อนุญาต (Approve)” หรือปุ่ม “ไม่อนุญาต (Reject)”

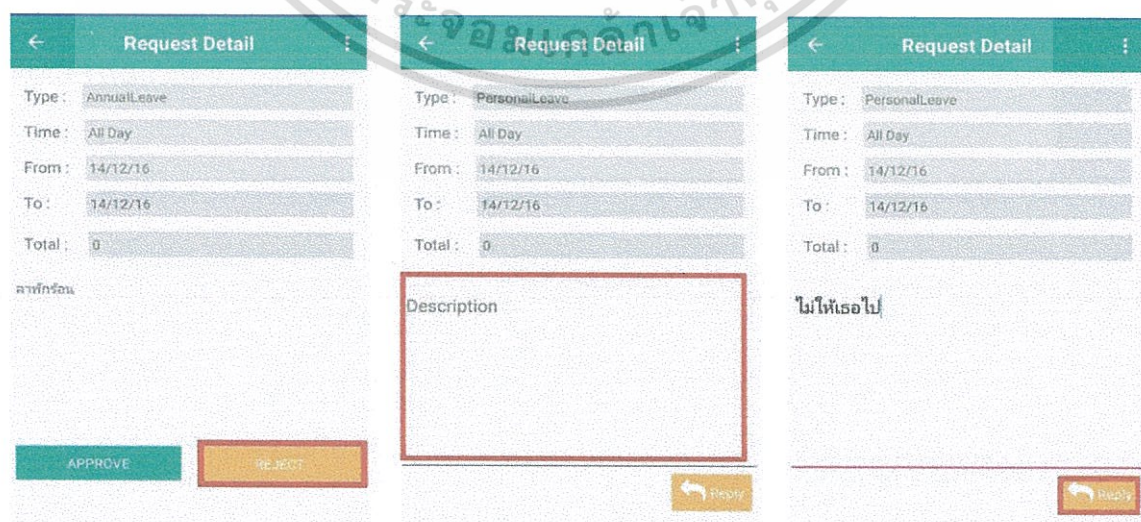
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหากผู้ใช้งานต้องการอนุญาตคำขอลานี้จะสามารถทำได้โดยการกดปุ่ม “อนุญาต (Approve)” หลังจากนั้นจะมีข้อความเพื่อให้ผู้ใช้งานยืนยันอีกครั้ง จึงจะถือว่าคำขอนี้ได้รับการอนุญาตให้ลา ถ้าหากผู้กดยกอนุญาตคำขอลานี้เป็น Supervisor คำขอลานี้จะถูกส่งต่อไปยัง Head of operation เพื่อทำการพิจารณาอีกครั้ง



ภาพที่ 4.16 ข้อความยืนยันว่าผู้ใช้งานต้องการอนุญาตให้ลา

ถ้าหากผู้ใช้งานไม่ต้องการอนุญาตคำขอลา ผู้ใช้สามารถทำการกดปุ่ม “ไม่อนุญาต (Reject)” เมื่อผู้ใช้งานกดปุ่มดังกล่าวแล้ว แอปพลิเคชันจะแสดงช่องให้กรอกเหตุผลที่ผู้ใช้ไม่อนุญาตให้ลา หลังจากนั้นผู้ใช้งานต้องกดปุ่ม “ตอบกลับ (Reply)” เพื่อยืนยันคำขอลานี้ว่าไม่อนุญาตให้ลา ดังภาพที่ 4.17



ภาพที่ 4.17 ขั้นตอนการไม่อนุญาตให้ลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้มีการส่งคำขอลา จะมีอีเมลล์แจ้งเตือนถึง Supervisor ของผู้ใช้นั้น และหาก Supervisor ทำการอนุญาตคำขอลาของผู้ใช้ ก็จะมีอีเมลล์แจ้งไปยัง Head of operation เพื่อให้ Head of operation ทำการพิจารณาคำขอลาเป็นขั้นตอนสุดท้ายของการอนุญาตคำขอลานั้นๆ โดยอีเมลล์จะมีลักษณะคล้ายๆ ภาพที่ 4.18



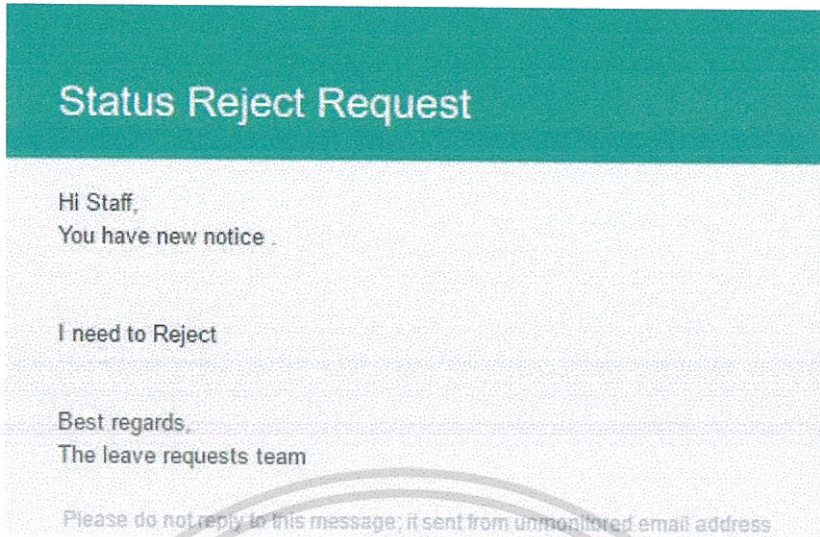
ภาพที่ 4.18 อีเมลล์แจ้งเตือนว่ามีผู้ใช้ส่งคำขอลามา

หากคำขอลานั้นได้รับการอนุญาตหรือไม่อนุญาตก็ตาม จะมีอีเมลล์แจ้งไปยังผู้ใช้ที่ส่งคำขอลาเข้ามาว่าคำขอลานั้นได้รับการอนุญาตหรือไม่ ดังภาพที่ 4.19 และภาพที่ 4.20



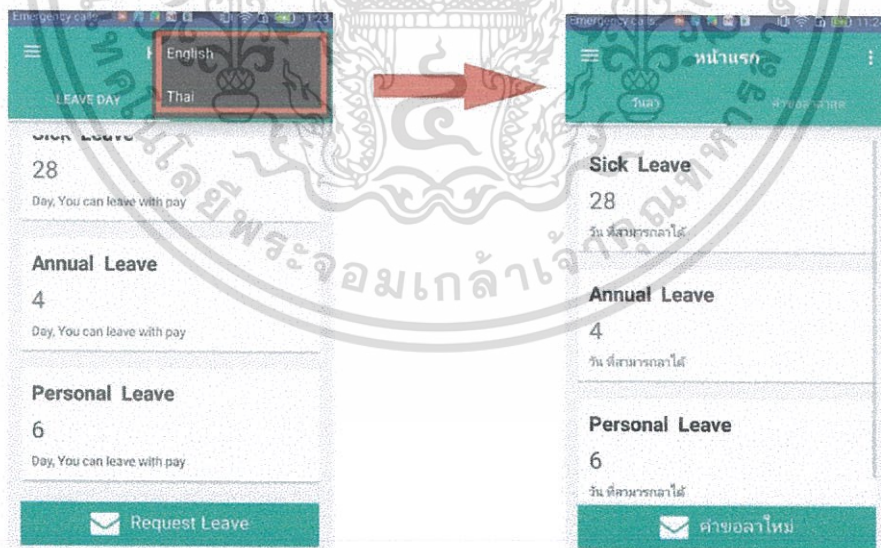
ภาพที่ 4.19 อีเมลล์แจ้งว่าคำขอได้รับการอนุญาตให้ลาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.20 อีเมลแจ้งว่าคำขอไม่ได้รับการอนุญาต

นอกจากนี้แล้วแอปพลิเคชันยังมี 2 ภาษาให้เลือกใช้ ทั้งภาษาไทย และภาษาอังกฤษ โดยผู้ใช้สามารถเปลี่ยนภาษาของแอปพลิเคชันได้จากปุ่มเมนูทางมุมขวาบน เมื่อผู้ใช้กดเปลี่ยนภาษาแอปพลิเคชัน จะทำการปิดตัวและเปิดขึ้นมาอีกครั้งโดยอัตโนมัติ ดังภาพที่ 4.21



ภาพที่ 4.21 การเปลี่ยนภาษาของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

แอปพลิเคชันสามารถเชื่อมต่อและทำงานร่วมกับเซิร์ฟเวอร์ผ่านทาง API ได้อย่างครบถ้วน โดยผู้ใช้งานสามารถล็อกอินเข้าใช้แอปพลิเคชัน ทางแอปพลิเคชันสามารถตรวจสอบและระบุได้ว่าผู้ใช้ที่ทำการล็อกอินเข้าใช้งานแอปพลิเคชันมีตำแหน่งภายในองค์กรเป็นตำแหน่งใด ผู้ใช้งานสามารถส่งคำขอลาไปยังผู้ที่มีอำนาจอนุญาตให้ตนเองลาได้อย่างถูกต้อง เมื่อมีการอนุญาตคำขอลาแล้วมีการส่งอีเมลแจ้งเตือนทั้งทางผู้ที่ยื่นคำขอลาและผู้ที่ต้องพิจารณาคำขอลาในลำดับต่อไป ผู้ใช้ที่มีอำนาจในการพิจารณาคำขอลาสามารถพิจารณาและตัดสินใจได้ว่าจะอนุญาตหรือไม่อนุญาตคำขอลาที่ถูกส่งมา ผู้ใช้งานสามารถตรวจสอบจำนวนวันลาคงเหลือของตนเองได้จากภายในแอปพลิเคชัน นอกจากนี้ยังสามารถตรวจสอบวันลาที่ผ่านมาของตนเองได้ด้วย สำหรับผู้ใช้งานที่ต้องการใช้แอปพลิเคชันในภาษาอังกฤษหรือภาษาไทยก็สามารถเปลี่ยนภาษาได้จากภายในแอปพลิเคชัน

ระหว่างการพัฒนาผู้ใช้งานสามารถเปลี่ยนแปลงความต้องการได้เสมอ ทำให้มีการต้องปรับปรุงแก้ไขแอปพลิเคชันควบคู่ไปกับการพัฒนาแอปพลิเคชันตลอดระยะเวลาการวิจัย ในส่วนนี้การนำแนวคิดในการพัฒนาซอฟต์แวร์แบบ Scrum มาใช้ ถือว่ามีส่วนช่วยให้การพัฒนาและการปรับปรุงแก้ไขแอปพลิเคชันเป็นไปตามเวลาที่กำหนดที่สุดเท่าที่จะทำได้

5.2 ปัญหาที่พบระหว่างดำเนินงานวิจัย

- การใช้ Third party library ใหม่ที่ไม่เคยใช้ ทำให้ต้องเสียเวลาบางส่วนระหว่างดำเนินงานไปกับการศึกษาวิธีใช้งานและการนำไปใช้
- ผู้ใช้มีความต้องการเปลี่ยนแปลงความสามารถของระบบได้ตลอดช่วงระยะเวลาดำเนินงานวิจัย
- เมื่อเกิดการเปลี่ยนแปลงความสามารถของระบบทำให้ต้อง เพิ่มระยะเวลาในการปรับปรุงแก้ไข และลดระยะเวลาในการพัฒนาระบบลง

5.3 แนวทางในการแก้ไขปัญหา

- พยายามลดระยะเวลาการเรียนรู้ หรือใช้เวลาว่างจากการดำเนินงานวิจัยในการศึกษาวิธีการใช้งานและการนำ Third party library มาใช้งานจริง หรืออาจเปลี่ยนไปพัฒนาแอปพลิเคชันในแบบที่เชี่ยวชาญแทน ถ้าหากประสิทธิภาพของแอปพลิเคชันยังอยู่ในระดับที่น่าพึงพอใจ ก็อาจไม่จำเป็นต้องนำ Third party library ใหม่ ๆ มาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การนำแนวคิดในการพัฒนาซอฟต์แวร์แบบ Scrum หรือ Agile มาใช้แทนแนวคิดการพัฒนาซอฟต์แวร์แบบ waterfall ทำให้การแก้ไขระบบหรือแอปพลิเคชันระหว่างการพัฒนา เป็นไปได้โดยง่ายและช่วยให้การพัฒนาสำเร็จคล่องภายในระยะที่กำหนดได้
- การใช้แนวคิดในการพัฒนาซอฟต์แวร์แบบ Scrum หรือ Agile ทำให้ช่วงเวลาในการพัฒนาระบบหรือแอปพลิเคชัน สามารถปรับเปลี่ยนได้อย่างยืดหยุ่น ช่วยให้การแก้ไขและปรับปรุงระบบและแอปพลิเคชันสามารถทำได้ตลอดระยะเวลาการพัฒนาระบบหรือแอปพลิเคชัน

5.4 ข้อเสนอแนะ

- ในการพัฒนาย่อมเกิดการเปลี่ยนแปลงความสามารถของแอปพลิเคชันอยู่บ่อยครั้งตามความต้องการของผู้ใช้ ในการพัฒนาแอปพลิเคชันหรือซอฟต์แวร์ใดๆก็ตาม ควรนำแนวคิดในการพัฒนาซอฟต์แวร์แบบ Scrum หรือ Agile มาปรับใช้เพื่อให้การพัฒนาสำเร็จทันเวลา
- ในการพัฒนาแอปพลิเคชันที่ต้องมีการทำงานร่วมกับฝั่งเซิร์ฟเวอร์ ผู้พัฒนาแอปพลิเคชันจำเป็นต้องมีความรู้ในภาษาของการพัฒนาฝั่งเซิร์ฟเวอร์ด้วย เพื่อใช้ในการแก้ปัญหาเฉพาะหน้า หรือแม้แต่ทำความเข้าใจกับข้อความที่ทางเซิร์ฟเวอร์ตอบรับกลับมา

ในการพัฒนาแอปพลิเคชันให้ผู้ใช้ได้รับประสบการณ์การใช้งานที่ดีนั้น ผู้พัฒนา, ทีมออกแบบ และผู้ใช้ควรได้ออกความเห็นร่วมกันบ่อยๆ เพื่อให้ทีมออกแบบและทีมพัฒนาเข้าใจว่าผู้ใช้ต้องการอะไรจากแอปพลิเคชัน จึงจะทำให้ผู้ใช้พอใจในการใช้งาน

เอกสารอ้างอิง

- [1] [Dev Tip] MVC คืออะไร ?. (1 พฤศจิกายน 2558). เข้าถึงได้จาก:
<https://www.arnondora.in.th/what-is-mvc/>
- [2] [Android Code] Let's Fragment - วงจรชีวิตของ Fragment (Fragment Lifecycle). (11 พฤศจิกายน 2557). เข้าถึงได้จาก:
<http://www.akexorcist.com/2014/11/lets-fragment-lifecycle.html>
- [3] แอนดรอยด์ (ระบบปฏิบัติการ). (29 พฤศจิกายน 2559). เข้าถึงได้จาก:
[https://th.wikipedia.org/wiki/%E0%B9%81%E0%B8%AD%E0%B8%99%E0%B8%94%E0%B8%A3%E0%B8%AD%E0%B8%A2%E0%B8%94%E0%B9%8C_\(%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A%E0%B8%9B%E0%B8%8F%E0%B8%B4%E0%B8%9A%E0%B8%B1%E0%B8%95%E0%B8%B4%E0%B8%81%E0%B8%B2%E0%B8%A3\)](https://th.wikipedia.org/wiki/%E0%B9%81%E0%B8%AD%E0%B8%99%E0%B8%94%E0%B8%A3%E0%B8%AD%E0%B8%A2%E0%B8%94%E0%B9%8C_(%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A%E0%B8%9B%E0%B8%8F%E0%B8%B4%E0%B8%9A%E0%B8%B1%E0%B8%95%E0%B8%B4%E0%B8%81%E0%B8%B2%E0%B8%A3))
- [4] Android Story ตอนที่ 5 Android SDK ชุดพัฒนาซอฟต์แวร์สำหรับแอนดรอยด์. (4 มกราคม 2555). เข้าถึงได้จาก:
<http://1st.phet.in.th/2011/01/android-story-5-android-sdk/>
- [5] ภาษาจาวา. (29 เมษายน 2559). เข้าถึงได้จาก:
<https://th.wikipedia.org/wiki/%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%E0%B8%88%E0%B8%B2%E0%B8%A7%E0%B8%B2>
- [6] JDK คืออะไร เจดีเค คือชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรม JAVA. (24 เมษายน 2555). เข้าถึงได้จาก:
<http://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2232-jdk-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html>
- [7] [Android Dev Tips] มาลองเรียกใช้งาน Android Support Library กันเถอะ. (7 ตุลาคม 2557). เข้าถึงได้จาก:
<http://www.akexorcist.com/2014/07/android-dev-tips-android-support-library.html>
- [8] Agile Method: วิธีการพัฒนาซอฟต์แวร์สำหรับปัจจุบันและอนาคต. (10 ตุลาคม 2555). เข้าถึงได้จาก:
http://noc.mcu.ac.th/index.php?option=com_content&view=article&id=2:agile-method-&catid=2:2012-10-10-02-09-41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

- [9] **scrum คืออะไร.** (30 กันยายน 2555). เข้าถึงได้จาก:
<https://opeaw.wordpress.com/2012/09/30/scrum/>
- [10] **Wireframe ที่ดี ต้องมีองค์ประกอบ 4 ข้อนี้.** (28 พฤษภาคม 2558). เข้าถึงได้จาก:
<http://uxacademy.in.th/study/wireframe/2015/05/28/good-wireframe-have-4-rule/>
- [11] **Wireframe คืออะไร ทำไมดีไซน์เนอร์ต้องรู้ พร้อมวิธีทำ + Tool ฟรี !!** (2 ธันวาคม 2556).
เข้าถึงได้จาก: <http://www.designil.com/wireframe-website-free-download.html>
- [12] **[Android Dev Tips] ปฏิบัติการพิมพ์ Mockup แบบฉบับเร่งด่วนโคตรๆแต่ไม่ธรรมดาด้วยเวลาแค่ 2 วัน.** (28 มกราคม 2559). เข้าถึงได้จาก:
<http://www.akexorcist.com/2016/01/mockup-android-app-in-two-days-operation.html>
- [13] **Git คืออะไร ? + พร้อมสอนใช้งาน Git และ Github.** (23 สิงหาคม 2558). เข้าถึงได้จาก:
<https://devahoy.com/posts/introduction-to-git-and-github/>
- [14] **XML คืออะไร เอ็กซ์เอ็มแอล คือ ภาษาที่ใช้กำหนดรูปแบบของคำสั่งภาษา HTML.** (30 เมษายน 2555). เข้าถึงได้จาก:
<http://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2129-xml-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html>
- [15] **MVC คืออะไร.** (12 สิงหาคม 2556). เข้าถึงได้จาก: <http://blog.bossturteam.com/mvc/>
- [16] **เริ่มต้นใช้งาน - เกี่ยวกับ Version Control.** (15 ธันวาคม 2559). เข้าถึงได้จาก: <https://git-scm.com/book/th/v1/%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-%E0%B9%80%E0%B8%81%E0%B8%B5%E0%B9%88%E0%B8%A2%E0%B8%A7%E0%B8%81%E0%B8%B1%E0%B8%9A-Version-Control>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้