

ชื่อระบบภาษาไทย
English Project Title

นาย(นางสาว)ชื่อ สกุล

โครงการนี้เป็นส่วนหนึ่งของการศึกษา
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2562
ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ : ชื่อระบบภาษาไทย
English Project Title
โดย : นาย(นางสาว)ชื่อ สกุล
: นาย(นางสาว)ชื่อร่วม สกุล
อาจารย์ที่ปรึกษา : ดร. ชื่อ สกุล
ระดับการศึกษา : วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา : 2562

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงการคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา
(ดร. ชื่อ สกุล)

..... กรรมการ
(ผศ. ชื่อ สกุล)

..... กรรมการ
(อาจารย์ ชื่อ สกุล)

..... หัวหน้าภาควิชา
(ผศ.ดร. สุพจน์ สีบุตร)

วันที่ ... / ... / ...

กิตติกรรมประกาศ

การพัฒนาโครงการระบบ XX สำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลือจากหลายๆ ท่าน ข้าพเจ้าขอขอพระคุณทุกท่าน ที่มีส่วนร่วมในการพัฒนาโครงการนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ชื่อ สกุล อาจารย์ที่ปรึกษาโครงการที่ได้แนะนำทฤษฎี และแนวทางในแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนา ระบบ อีกครั้งยังคอยตรวจสอบความก้าวหน้าของการทำงานเป็นระยะ ๆ รวมทั้งสร้างกำลังใจให้ผู้พัฒนาอยู่เสมอ

ขอขอบพระคุณ ชื่อ สกุล เจ้าหน้าที่งาน XX หน่วยงาน ผู้ให้คำปรึกษาเรื่องขั้นตอนการดำเนินงานและเอกสารที่เกี่ยวข้องของระบบ XX

ขอขอบพระคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ ที่คอยเอื้ออำนวย ความสะดวกทั้งเรื่องอุปกรณ์และสถานที่ต่อการปฏิบัติงานของผู้พัฒนา

ขอขอบคุณบริษัท XX จำกัด ที่ให้โอกาสและความรู้ที่ถูกต้องในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอย์รวมถึงสอนองค์ความรู้ในการวางแผนงานอีกด้วย

ขอกราบขอบพระคุณบิดา มารดา ที่คอยให้กำลังใจ คอยให้ความรักและความห่วงใยเสมอมา ตลอดจนคอยช่วยเหลือทุนทรัพย์ทางการศึกษาและอุปกรณ์ในการพัฒนาโครงการ

นายชื่อ สกุล

เมษายน 62

โครงการ : ชื่อระบบภาษาไทย
โดย : นาย(นางสาว)ชื่อ สกุล
: นาย(นางสาว)ชื่อร่วม สกุล
อาจารย์ที่ปรึกษา : ดร. ชื่อ สกุล
ระดับการศึกษา : วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา : 2562

บทคัดย่อ

ระบบ XX abstract ภาษาไทย

คำสำคัญ: เว็บแอปพลิเคชัน แอนดรอยด์แอปพลิเคชัน วิวเจเอส

Topic : English Project Title
Author : FIRSTNAME LASTNAME
: FIRSTCONAME LASTNAME
Advisor : FIRTNAME LASTNAME, Ph.D.
Degree : Bachelor of Science (Computer Science)
Academic Year : 2019

Abstract

XX System English abstract

Keywords: Web Application, Android Application, Vue.js

สารบัญ

	หน้า
กิตติกรรมประกาศ	ค
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
สารบัญ	ฉ
สารบัญตาราง	ช
สารบัญภาพ	ฌ
บทที่	
1 บทนำ	1
1.1 ที่มาและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)	2
1.5.1 ฮาร์ดแวร์	2
1.5.2 ซอฟต์แวร์ (Software)	2
1.5.3 แผนการดำเนินการ	4
2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์	5
2.1.1 ประวัติความเป็นมาของระบบปฏิบัติการแอนดรอยด์	6
2.1.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์	7
2.1.3 การจัดการเกี่ยวกับวัฏจักรแอคทิวิตี้ของแอปพลิเคชัน	9
2.1.4 กระบวนการเริ่มทำงานของแอคทิวิตี้ (Activity)	10
2.1.5 ทำความรู้จักกับ Lifecycle Callback	10
2.1.6 การกำหนด Launcher Activity ในแอปพลิเคชัน	12
2.1.7 การสร้าง Instance ใหม่	13
2.1.8 การสิ้นสุดการทำงานของแอคทิวิตี้ (Destroy Activity)	16
2.2 ความรู้พื้นฐาน Java	17

2.2.1	ประวัติความเป็นมาของภาษา Java	17
2.2.2	Java Compiler	17
2.2.3	โครงสร้างของภาษา Java	18
2.3	ความรู้พื้นฐาน Java Script	21
2.3.1	ลักษณะการทำงานของ JavaScript	22
2.3.2	JavaScript กับ HTML	24
2.3.3	โครงสร้างภาษา	24
2.4	ความรู้พื้นฐาน Vue.js Fronted Framework	29
2.4.1	การติดตั้ง	29
2.4.2	คุณลักษณะของวิวเจเอส	30
2.4.3	ข้อดีของ Vue.js	31
2.5	ความรู้พื้นฐาน ScanLibrary	31
2.5.1	ขั้นตอนการติดตั้งและใช้งาน	31
2.5.2	ตัวอย่างการใช้งาน	32
2.6	ความรู้พื้นฐานของระบบ XX เดิม	34
2.6.1	ความเป็นมา	34
2.6.2	วิสัยทัศน์และพันธกิจ	34
2.6.3	ขั้นตอนการดำเนินการ XX	34
2.7	เอกสารและงานวิจัยที่เกี่ยวข้อง	35
2.7.1	เว็บไซต์ XX	35
2.7.2	แอปพลิเคชัน XX	35
	บรรณานุกรม	38
	ประวัติผู้เขียน	39

สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน	4

สารบัญภาพ

รูปที่	หน้า
2.1	ส่วนแบ่งการตลาดระบบปฏิบัติการบนสมาร์ทโฟน 6
2.2	โครงสร้างของระบบปฏิบัติการแอนดรอยด์ 7
2.3	วัฏจักรของแอคทิวิตีบนระบบปฏิบัติการแอนดรอยด์ 10
2.4	การกำหนด Launcher Activity ในแอปพลิเคชัน 12
2.5	การกำหนด Launcher Activity ในแอปพลิเคชัน 12
2.6	การกำหนด Launcher Activity ในแอปพลิเคชัน ด้วย minted 13
2.7	การกำหนด Launcher Activity ในแอปพลิเคชัน 13
2.8	การใช้งาน onCreate() ด้วย minted 14
2.9	การใช้งาน onCreate() 15
2.10	การใช้งาน onDestroy() 16
2.11	ตัวอย่างโปรแกรมภาษา Java ด้วย minted 18
2.12	ตัวอย่างโปรแกรมภาษา Java ด้วย minted คลาส MainActivity 19
2.13	การประกาศคลาสในภาษา Java 20
2.14	การประกาศเมธอด main ในภาษา Java 20
2.15	ตัวอย่างคำสั่งภาษา TypeScript ด้วย minted 23
2.16	ตัวแปรอาร์เรย์ 26
2.17	การเปรียบเทียบว่าจริงหรือเท็จ 27
2.18	ตัวดำเนินการเชิงข้อความ 28
2.19	Components ของวิวเจเอส 30
2.20	การเรียกใช้งาน 31
2.21	การรับค่าจากไลบรารี 32
2.22	ถ่ายภาพเอกสาร 33
2.23	ปรับแต่งภาพถ่าย 33
2.24	ผลการทำงาน 34
2.25	หน้าแรกของเว็บไซต์ PSU Studentloan 35
2.26	ข่าวสารจากหน่วยงาน 37

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

เนื่องจากระบบการทำงานเดิม มีข้อจำกัดดังนี้ XX และในปัจจุบัน เทคโนโลยี XX สามารถทำให้ดีขึ้น ดังนี้ XX

ดังนั้น ผู้พัฒนาจึงเสนอระบบงาน XX โดยใช้เทคโนโลยี XX เพื่อแก้ปัญหา โดย มีความสามารถดังต่อไปนี้ XX

1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบแอปพลิเคชัน
2. เพื่ออำนวยความสะดวกสำหรับ

1.3 ขอบเขตของโครงการ

1.3.1 เจ้าหน้าที่

- สามารถจัดการ X
- สามารถเพิ่ม X
- สามารถตรวจสอบอนุมัติ X

1.3.2 นักศึกษา

- สามารถดู X ได้
- สามารถกำหนด X ได้
- สามารถดาวน์โหลด X ได้
- สามารถจอง X ได้
- สามารถส่ง X ได้
- สามารถสนทนากับ X ผ่านระบบได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยอำนวยความสะดวก XX
2. ช่วยกระจายงานของ XX
3. ช่วยจัดระบบ XX

1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)

1.5.1 ฮาร์ดแวร์

1. สมาร์ทโฟน (Smart phone)

- ทำงานบนระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 5.0 หรือ API Level 21
- หน่วยประมวลผลกลาง Mediatek MT6753 Octa-core ความเร็ว 1.3 กิกะเฮิร์ตซ์ (Gigahertz, GHz)
- หน่วยประมวลผลกราฟฟิกลำดับต่ำ Mali-T720MP3
- หน่วยความจำหลักอย่างน้อย 2 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรองอย่างน้อย 16 กิกะไบต์ (Gigabyte, GB)
- หน้าจอแสดงผลความละเอียดอย่างน้อย 1080 x 1920 พิกเซล (Pixel)
- หน้าจอแสดงผลขนาดอย่างน้อย 5 นิ้ว
- กล้องถ่ายรูปความละเอียดอย่างน้อย 13 เมกะพิกเซล (Megapixel)

2. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer)

- ทำงานบนระบบปฏิบัติการ Elementary os พื้นฐานการทำงานบน Linux
- หน่วยประมวลผลกลาง Intel Core i3-3217U ความเร็ว 1.80 กิกะเฮิร์ตซ์ (Gigahertz, GHz)
- หน่วยประมวลผลกราฟฟิก NVIDIA GeForce GT 720M ความจำ 2 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำหลัก 4 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรอง 120 กิกะไบต์ (Gigabyte, GB)

1.5.2 ซอฟต์แวร์ (Software)

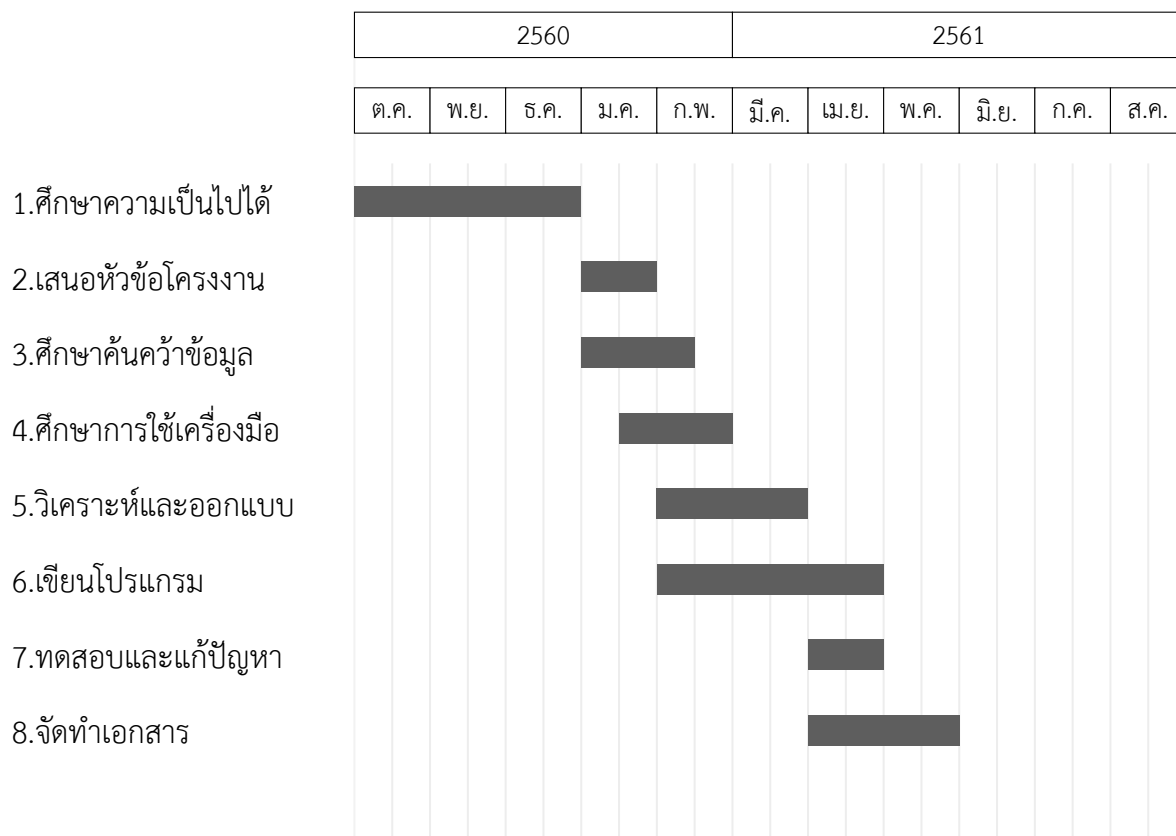
1. XX.js ซึ่งเป็น Frontend Framework สำหรับพัฒนาเว็บไซต์

2. Node.js คือ Cross Platform Runtime Environment หรือเรียกอีกอย่างว่า Backend Framework ใช้สำหรับเป็นเว็บเซิร์ฟเวอร์ (Web Server) ซึ่งเขียนด้วยภาษา JavaScript
3. Android Studio เป็น IDE (Integrated Development Environment) ใช้พัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการแอนดรอยด์
4. Android SDK ชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมสำหรับระบบปฏิบัติการแอนดรอยด์
5. Visual Studio Code เครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชัน

1.5.3 แผนการดำเนินการ

ในการสร้างระบบ XX ผู้พัฒนาได้แบ่งขั้นตอนการดำเนินงานไว้ด้วยกัน 7 ขั้นตอน ดังต่อไปนี้

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน



บทที่ 2

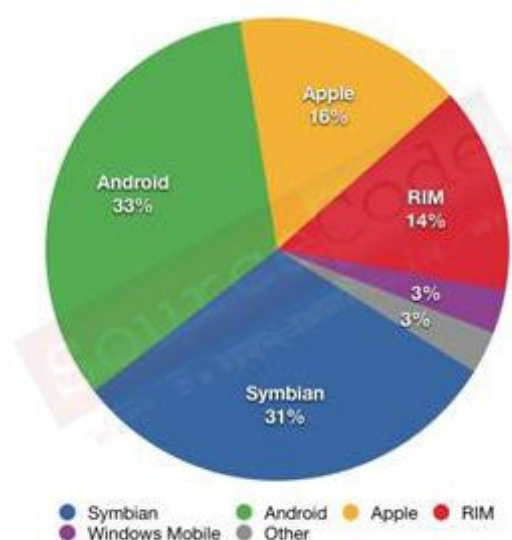
ทฤษฎีที่เกี่ยวข้อง

บทนี้จะเป็นรายละเอียดเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการพัฒนาโปรแกรมในครั้งนี้ โดยที่แต่ละหัวข้อจะมีความสัมพันธ์กันเป็นลำดับ โดยหัวข้อที่หนึ่งจะแนะนำความรู้เรื่อง xxx เพื่อให้เข้าใจพื้นฐานเบื้องต้นเกี่ยวกับที่มาของโครงการ หัวข้อที่สองที่สามจะช่วยเตรียมให้ผู้อ่านเข้าใจเทคโนโลยีที่ใช้ในการออกแบบและพัฒนา ส่วน ...

2.1 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ [1] คือระบบปฏิบัติการแบบเปิดเผยแพร่แวร์ต้นฉบับ (Open Source) โดยบริษัท กูเกิล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลายราคา รวมทั้งสามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอ และความละเอียดแตกต่างกันได้ ทำให้ผู้บริโภคสามารถเลือกได้ตามต้องการและหามองในทิศทางสำหรับนักพัฒนาโปรแกรม (Programmer) แล้วนั้นการพัฒนาโปรแกรมเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ ไม่ใช่เรื่องยาก เพราะมีข้อมูลในการพัฒนารวมทั้ง Android SDK (Software Development Kit) เตรียมไว้ให้กับนักพัฒนาได้เรียนรู้ และเมื่อนักพัฒนาต้องการจะเผยแพร่หรือจำหน่ายโปรแกรมที่พัฒนาแล้วเสร็จแอนดรอยด์ก็ยังมีตลาดในการเผยแพร่โปรแกรม Google PlayStore แต่หากจะกล่าวถึงโครงสร้างภาษาที่ใช้ในการพัฒนานั้น สำหรับ Android SDK จะยึดโครงสร้างของภาษาจาวา (Java language) ในการเขียนโปรแกรม เพราะโปรแกรมที่พัฒนามาได้จะต้องทำงานอยู่ภายใต้ Dalvik Virtual Machine เช่นเดียวกับโปรแกรมจาวา ที่ต้องทำงานอยู่ภายใต้ Java Virtual Machine (Virtual Machine เปรียบได้กับสภาพแวดล้อมที่โปรแกรมทำงานอยู่)

นอกจากนี้แล้วแอนดรอยด์ยังมีโปรแกรมเกมที่เป็นโอเพ่นซอร์สแวร์ต้นฉบับ (Open Source) เป็นจำนวนมาก ทำให้นักพัฒนาที่สนใจสามารถนำซอร์สแวร์ต้นฉบับมาศึกษาได้ประกอบกับความนิยมของแอนดรอยด์ได้เพิ่มขึ้นอย่างมากในปัจจุบัน โดยดูได้จากส่วนแบ่งการตลาด ดังรูปที่ 2.1



รูปที่ 2.1: ส่วนแบ่งการตลาดระบบปฏิบัติการบนสมาร์ทโฟน

ที่มา : <https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าสุด/ระบบปฏิบัติการ-android.html>

2.1.1 ประวัติความเป็นมาของระบบปฏิบัติการแอนดรอยด์

เริ่มต้นระบบปฏิบัติการแอนดรอยด์ถูกพัฒนามาจากบริษัทแอนดรอยด์ (Android Inc.) เมื่อปี พ.ศ 2546 โดยมีนาย แอนดี้ รูบิน (Andy Rubin) ผู้ให้กำเนิดระบบปฏิบัติการนี้และถูกบริษัทกูเกิ้ลเข้าซื้อกิจการเมื่อ เดือนสิงหาคม ปี พ.ศ 2548 โดยบริษัทแอนดรอยด์ ได้กลายมาเป็นบริษัทลูกของบริษัทกูเกิ้ล

ระบบปฏิบัติการแอนดรอยด์ เป็นระบบปฏิบัติการที่พัฒนามาจากการนำเอาแกนกลางของระบบปฏิบัติการลินุกซ์ (Linux Kernel) ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อทำงานเป็นเครื่องให้บริการ (Server) มาพัฒนาต่อ เพื่อให้กลายเป็นระบบปฏิบัติการบนอุปกรณ์พกพา (Mobile Operating System)

ต่อมาเมื่อเดือน พฤศจิกายน ปี พ.ศ 2550 บริษัทกูเกิ้ล ได้ทำการก่อตั้งสมาคม OHA (Open Handset Alliance) เพื่อเป็นหน่วยงานกลางในการกำหนดมาตรฐานกลาง ของอุปกรณ์พกพาและระบบปฏิบัติการแอนดรอยด์ โดยมีสมาชิกในช่วงก่อตั้งจำนวน 34 รายเข้าร่วม ซึ่งประกอบไปด้วยบริษัทชั้นนำที่ดำเนินธุรกิจด้านการสื่อสาร เช่น โรงงานผลิตอุปกรณ์พกพา บริษัทพัฒนาโปรแกรม ผู้

ให้บริการสื่อสาร และผู้ผลิตอะไหล่อุปกรณ์ด้านสื่อสาร [2]

หลังจากนั้น เมื่อเดือนตุลาคม ปี พ.ศ 2551 บริษัท กูเกิ้ล ได้เปิดตัวมือถือตัวแรกที่ใช้ระบบปฏิบัติการแอนดรอยด์ ที่ชื่อ T-Mobile G1 หรืออีกชื่อหนึ่งคือ HTC Dream โดยใช้แอนดรอยด์รุ่น 1.1 และหลังจากนั้น ได้มีการพัฒนาระบบปฏิบัติการเป็นรุ่นใหม่ มาเป็นลำดับ

ช่วงต่อมาได้มีการออกผลิตภัณฑ์จากบริษัทต่าง ๆ ออกมาหลากหลายรุ่น หลากหลายยี่ห้อ ตามการพัฒนาระบบปฏิบัติการแอนดรอยด์ ที่มีอยู่อย่างต่อเนื่อง ทำให้สินค้าของแอนดรอยด์ มีให้เลือกอยู่อย่างมากมาย

2.1.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ [3] ถือว่าเป็นสิ่งสำคัญเพราะ ถ้านักพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบได้ทั้งหมด จะสามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนาเพื่อให้เกิดประสิทธิภาพในการทำงาน



รูปที่ 2.2: โครงสร้างของระบบปฏิบัติการแอนดรอยด์

ที่มา : <https://www.theandroid-mania.com/android-architecture/>

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ในรูปที่ 2.2 จะสังเกตได้ว่า มีการแบ่งออก

เป็นส่วน ๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งคือ ส่วนของ Applications ลำดับถัดมาเป็นองค์ประกอบอื่น ๆ ตามลำดับ และสุดท้ายเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel โครงสร้างของแอนดรอยด์สามารถอธิบายได้ดังนี้

1. Applications ส่วนแอปพลิเคชันหรือส่วนของโปรแกรมที่มากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่าง ๆ ได้โดยตรงซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ด (Code) โปรแกรมเอาไว้
2. Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากมาย เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งานแล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น
 - Activities Manager เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม (Activity)
 - Content Providers เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้
 - View System เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)
 - Telephony Manager เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่น หมายเลขโทรศัพท์ เป็นต้น
 - Resource Manager เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลที่เป็นข้อความและรูปภาพ
 - Location Manager เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์
 - Notification Manager เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรมต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ (Status Bar) ของหน้าจอ
3. Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน เช่น Surface Manage จัดการเกี่ยวกับการแสดงผล Media

Framework จัดการเกี่ยวกับการแสดงผลภาพและเสียง OpenGL|ES และ SGL จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ SQLite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

4. Android Runtime จะมี Dalvik Virtual Machine ที่ถูกออกแบบมาเพื่อให้ทำงานบนอุปกรณ์ที่มีหน่วยความจำ (Memory) หน่วยประมวลผลกลาง (CPU) และพลังงาน (Battery) ที่จำกัดซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงานไปเป็นไฟล์ .DEX ก่อนการทำงานเหตุผลเพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับหน่วยประมวลผลกลางที่มีความเร็วไม่มากส่วนต่อมาก็คือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญโดยถูกเขียนด้วยภาษาจาวา (Java Language)
5. Linux Kernel เป็นส่วนที่ทำหน้าที่หัวใจสำคัญในการจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 2.6. Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.1.3 การจัดการเกี่ยวกับวัฏจักรแอคทิวิตีของแอปพลิเคชัน

ขณะที่ผู้ใช้เปิดใช้งานแอปพลิเคชัน -> ออกจากแอปพลิเคชัน -> แล้วก็กลับเข้ามาในแอปพลิเคชันอีกครั้งแอคทิวิตีจะมีการย้าย Method ต่างๆ เกิดขึ้นในวัฏจักรแอคทิวิตี ยกตัวอย่างเช่น เมื่อแอคทิวิตีเริ่มทำงานครั้งแรกจะแสดงขึ้นมาอยู่ด้านบนสุดของระบบ (Foreground) และรอรับการทำงานจากผู้ใช้ในระหว่างกระบวนการนี้ระบบจะมีการเรียกใช้งาน Callback Method หรือ Method ที่ถูกเรียกใช้งานอัตโนมัติในแอคทิวิตีที่ได้กำหนดการทำงานให้กับ UI และส่วนติดต่ออื่นๆ ไว้ ถ้าผู้ใช้มีการใช้งานใด ๆ ที่เป็นการเรียกแอคทิวิตีอื่นขึ้นมาหรือสลับไปใช้งานแอปพลิเคชันอื่นระบบจะเรียก Callback Method อีกอันขึ้นมา เช่น ซ่อนแอปพลิเคชันไว้ด้านหลัง Background (ไม่แสดงแอคทิวิตีแต่ Instance และ Method นั้นยังทำงานอยู่)

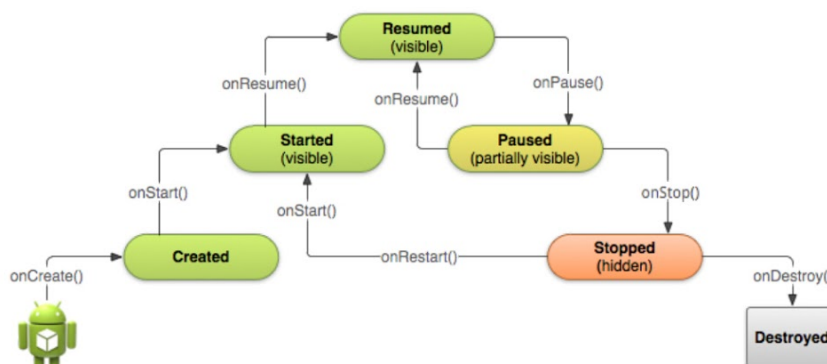
ภายใน Callback Method สามารถกำหนดการทำงานในแอคทิวิตีเมื่อผู้ใช้ออกจากแอปพลิเคชันและกลับเข้ามาใช้งานแอปพลิเคชันใหม่อีกครั้งได้ ตัวอย่าง ถ้าแอปพลิเคชันเป็นแอปพลิเคชัน Streaming Video อาจจะสั่งให้ทำการหยุด Video ชั่วคราว และปิดการเชื่อมต่อ Network ไว้ก่อนเมื่อผู้ใช้สลับไปใช้แอปพลิเคชันอื่น และทันทีที่ผู้ใช้กลับมาใช้งานแอปพลิเคชันต่อ ก็ให้ทำการเชื่อมต่อกับ Network และก๊อปปี้ให้ผู้ใช้กลับไปเล่น Video ในตำแหน่งที่ค้างต่อไปทันทีที่ได้ โดยที่ไม่ต้องเริ่มต้นแอปพลิเคชันใหม่ เป็นต้น

2.1.4 กระบวนการเริ่มทำงานของแอคทิวิตี้ (Activity)

ในระบบแอนดรอย์การกำหนดโค้ดเริ่มต้นไว้ในแอคทิวิตี้โดยสัมพันธ์กับ Method ที่ถูกเรียกใช้งานอัตโนมัติ (Callback Method) อย่างเป็นลำดับ ตั้งแต่เริ่มต้นแอคทิวิตี้ไปจนถึงสิ้นสุดและปิดการทำงานของ Activity ลง

2.1.5 ทำความรู้จักกับ Lifecycle Callback

ในขณะที่แอคทิวิตี้ [4] ทำงานระบบจะเรียกใช้ Callback Method ตามลำดับในลักษณะที่คล้ายกับการก่อพิธีมิต นั่นคือ แต่ละขั้นตอนวัฏจักรของแอคทิวิตี้คือส่วนแยกย่อยแต่ละขั้นของพิธีมิต เช่น เมื่อระบบสร้าง Instance ของแอคทิวิตี้ขึ้นมาใหม่ Method ที่เรียกใช้งานอัตโนมัติ (Callback Method) จะขยับ Activity Method ขึ้นมาด้านบนโดยด้านบนของพิธีมิตคือจุดที่แอคทิวิตี้กำลังทำงานแสดงอยู่ด้านหน้า (Foreground Activity) สุดและผู้ใช้กำลังใช้งานอยู่และเมื่อผู้ใช้กำลังจะออกจากแอคทิวิตี้ระบบจะเรียกใช้ Method อื่นซึ่งทำให้ Activity Method ถอยกลับไปอยู่ด้านล่างของพิธีมิตตามลำดับเพื่อหยุดการทำงานและลบแอคทิวิตี้ออกไป ในบางกรณีแอคทิวิตี้จะย้ายลงมาอยู่บางจุดและรอจังหวะที่จะถูกเรียกกลับขึ้นมาด้านบนอีก เช่น ในกรณีเมื่อผู้ใช้สลับไปใช้งานแอปพลิเคชันอื่นแล้วกลับมาใช้งานอีกครั้ง



รูปที่ 2.3: วัฏจักรของแอคทิวิตี้บนระบบปฏิบัติการแอนดรอยด์

ที่มา : <https://www.dev2qa.com/android-activity-lifecycle-example/>

จากรูปที่ 2.3 แสดงวัฏจักรของแอคทิวิตี้ในรูปแบบโครงสร้างพิธีมิตโดยแสดงให้เห็นว่า Method ที่เรียกใช้งานอัตโนมัติ (Callback Method) ได้แก่ onCreate(), onStart(), onResume() และ onRestart() จะขยับแอคทิวิตี้ขึ้นไปด้านบนสุดที่ Resumed Method และมี Method ได้แก่

onPause(), onStop() และ onDestroy() ที่จะขยับแอคทิวิตีลงมาด้านล่าง แอคทิวิตียังสามารถกลับไปทำงานที่ตำแหน่ง Resumed Method จากตำแหน่ง Paused และ Stopped ได้อีกด้วย

ในบางครั้งไม่จำเป็นต้องเรียกใช้งาน Callback Method ทั้งหมดเสมอไปขึ้นกับความซับซ้อนของแอคทิวิตี อย่างไรก็ตามเป็นสิ่งสำคัญที่นักพัฒนาควรทำความเข้าใจแต่ละ Method เพื่อให้มั่นใจได้ว่าแอปพลิเคชันของที่ได้พัฒนาตอบสนองเป็นไปตามที่ผู้ใช้คาดหวัง ดังนั้น ในการใช้งาน Callback Method ที่ถูกวิธีก็จะช่วยให้แอปพลิเคชันทำงานได้เป็นอย่างดี ดังนี้

- ไม่หยุดการทำงานหรือค้าง กรณีมีสายโทรเข้าหรือมีการสลับไปใช้งานแอปพลิเคชันอื่น
- ไม่ใช้ทรัพยากรที่มีค่าของระบบอย่างสูญเปล่า ถ้าไม่มีการใช้งานแอคทิวิตีใดๆ
- ไม่กระทบต่อกระบวนการในขั้นตอนการใช้งานของผู้ใช้กรณีออกจากแอปพลิเคชันแล้วกลับมาใช้งานอีกครั้ง
- ไม่หยุดการทำงานหรือระบบค้างที่กระทบการใช้งานของผู้ใช้กรณีมีการหมุนหน้าจอแนวนอนและแนวตั้งสลับกัน

เหตุการณ์ที่แอคทิวิตีมีการเปลี่ยน Method ต่าง ๆ ตามแสดงในรูปที่ 2.3 แต่มีอยู่ 3 Method เท่านั้นที่แอคทิวิตีจะยังคงอยู่คงที่ในช่วงเวลาระยะเวลาหนึ่งไม่เปลี่ยนไป Method อื่นในทันที ได้แก่

- Resumed (แสดงอยู่ ทำงานอยู่) ใน Method นี้แอคทิวิตีจะแสดงอยู่ด้านหน้าสุดและผู้ใช้กำลังใช้งานอยู่ บ่อยครั้งจะเรียกว่า Running Method
- Paused (แสดงหน้าจอบางส่วน ไม่ถูกบังสนิท) ใน Method นี้แอคทิวิตีจะถูกบดบังด้วยแอคทิวิตีอื่น เช่น แอคทิวิตีอื่นที่อยู่ด้านหน้าสุดที่แสดงในลักษณะกึ่งโปร่งใสหรือไม่ได้แสดงแบบเต็มหน้าจอ แอคทิวิตีในสถานะนี้จะไม่สามารถรับค่าจากผู้ใช้และทำงานคำสั่งใด ๆ ได้
- Stopped (แสดงหน้าจอแบบ Background ผู้ใช้มองไม่เห็น) ใน Method นี้ แอคทิวิตีจะถูกบดบังอย่างสมบูรณ์และผู้ใช้มองไม่เห็นโดยจะถูกย้ายไปอยู่ด้านหลังในขณะที่อยู่ใน Method นี้ ค่า Activity Instance และตัวแปรทั้งหมดจะยังคงอยู่แต่จะไม่สามารถถูกเรียกมาใช้งานจากโค้ดใด ๆ ได้

ในขณะที่ Method อื่น เช่น Created และ Started จะแสดงชั่วคราวแล้วระบบก็จะเปลี่ยนไป Method อื่นในทันทีที่ Method ถูกเรียกใช้งานอัตโนมัติ นั่นคือ หลังจากทีระบบเรียกใช้งาน

onCreate() แล้วก็จะเรียกใช้งาน onStart() ทันทีและสุดท้ายตามด้วย onResume() ซึ่งก็จะเข้าสู่ Resumed Method ทั้งหมดก็คือวัฏจักรแอคทิวิตีเบื้องต้น

2.1.6 การกำหนด Launcher Activity ในแอปพลิเคชัน

เมื่อผู้ใช้กดที่ Icon App จากหน้า Home Screen เพื่อใช้งาน ระบบก็จะเรียก onCreate() Method ขึ้นมาทำงานอัตโนมัติเพื่อที่จะเปิดใช้งานแอคทิวิตีที่ได้กำหนดให้เป็นแอคทิวิตีหลัก ("launcher" หรือ "main") ซึ่งก็จะเป็นแอคทิวิตีที่เป็นหน้าหลักที่ผู้ใช้จะเห็นเมื่อเข้ามาใช้งานแอปพลิเคชัน สามารถกำหนดได้ว่าแอคทิวิตีใดที่จะใช้เป็น Activity หลักโดยกำหนดค่าในไฟล์ AndroidManifest.xml

แอคทิวิตีหลักจะต้องกำหนด ค่าใน manifest ด้วย <intent-filter>

```
xml<intent-filter>    <action    android:name="android.intent.action.MAIN"    />
<category android:name="android.intent.category.LAUNCHER" /> </intent-filter>
```

รูปที่ 2.4: การกำหนด Launcher Activity ในแอปพลิเคชัน

```
1 <intent-filter>
2 <action android:name="android.intent.action.MAIN" />
3 <category android:name="android.intent.category.
  LAUNCHER" />
4 </intent-filter>
```

รูปที่ 2.5: การกำหนด Launcher Activity ในแอปพลิเคชัน

โดยประกอบไปด้วย MAIN action และ LAUNCHER category ดังตัวอย่างดังนี้

```

1 <Activity android:name=".MainActivity" android:label="@string/
   ↳ app_name">
2 <intent-filter>
3 <action android:name="android.intent.action.MAIN" />
4 <category android:name="android.intent.category.LAUNCHER" />
5 </intent-filter>
6 </Activity>

```

รูปที่ 2.6: การกำหนด Launcher Activity ในแอปพลิเคชัน ด้วย minted

```

1 <Activity android:name=".MainActivity" android:label
   ="@string/app_name">
2 <intent-filter>
3 <action android:name="android.intent.action.MAIN" />
4 <category android:name="android.intent.category.
   LAUNCHER" />
5 </intent-filter>
6 </Activity>

```

รูปที่ 2.7: การกำหนด Launcher Activity ในแอปพลิเคชัน

ใน Android Studio เมื่อสร้างโปรเจก (Project) ขึ้นมาจะมีการประกาศค่าและกำหนดให้มีการเรียกใช้งาน filter ในไฟล์ AndroidManifest.xml มาให้เรียบร้อยแล้ว ถ้าไม่ได้กำหนด MAIN action และ LAUNCHER category แล้ว App Icon ของจะไม่แสดงในหน้าหลักของลิสรายการแอปพลิเคชันบนหน้าจอสมาร์ตโฟนของผู้ใช้

2.1.7 การสร้าง Instance ใหม่

แอปพลิเคชันส่วนใหญ่จะประกอบไปด้วยแอกทิวิตีหลายอันทำงานแตกต่างกันไป ไม่ว่าจะเป็นแอกทิวิตีหลักที่ถูกสร้างขึ้นเมื่อผู้ใช้คลิกที่ icon app หรือแอกทิวิตีต่าง ๆ ที่ถูกเรียกใช้งานโดยผู้ใช้ ทั้งหมดล้วนแล้วเป็นสิ่งที่ระบบสร้าง instance ใหม่ของแอกทิวิตีโดยการเรียกใช้ผ่าน onCreate() Method ต้องใช้ onCreate() Method ในเหตุผลเพื่อเริ่มต้นแอปพลิเคชันซึ่งควรเกิดขึ้นเพียงครั้งเดียวตลอดหนึ่งวัฏจักรของแอกทิวิตีเช่น ใช้ onCreate() กำหนดหน้าตา

ของแอปพลิเคชันหรือประกาศตัวแปรที่จะถูกใช้งานในคลาส (Class) เป็นต้น ตัวอย่างการใช้งาน onCreate() Method ตามตัวอย่างรูปที่ 2.9 แสดง code เพื่อให้เห็นการตั้งค่าเบื้องต้นเช่น การกำหนด User interface(โดยใช้ XML layout ไฟล์) การประกาศตัวแปรต่าง ๆ การตั้งค่ากำหนดเงื่อนไข UI

```

1 TextView mTextView; // การประกาศตัวแปรสำหรับ text view ใน layout
2
3 @Override
4 public void onCreate(Bundle savedInstanceState) {
5     super.onCreate(savedInstanceState);
6
7     // การกำหนด user interface (โดยใช้ XML layout ไฟล์)
8     // ไฟล์ layout ใน project จะอยู่ที่ res/layout/main_Activity.xml
9     setContentView(R.layout.main_Activity);
10
11     // กำหนดค่าให้กับ TextView ดังนั้นสามารถเรียกใช้งานผ่านชื่อตัวแปรได้ภายหลัง
12     mTextView = (TextView) findViewById(R.id.text_message);
13
14     // ตั้งค่าสำหรับกำหนดเงื่อนไข UI
15     // ตรวจสอบเพื่อยืนยันว่ากำลังใช้งานบน android Honeycomb
16     // หรือสูงกว่า เพื่อเรียกใช้ ActionBar API
17     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
18         // กำหนดให้ icon ใน action bar ไม่ทำงานคล้ายปุ่ม Home
19         ActionBar actionBar = getActionBar();
20         actionBar.setHomeButtonEnabled(false);
21     }
22 }

```

รูปที่ 2.8: การใช้งาน onCreate() ด้วย minted

```

1 | TextView mTextView; // การประกาศตัวแปรจำห
   | รับtext view ในlayout
2 |
3 | @Override
4 | public void onCreate(Bundle savedInstanceState) {
5 |     super.onCreate(savedInstanceState);
6 |
7 |     // การกำหนดuser interface โดยใช้( XML layout ไฟล์)
8 |     // ไฟล์layout ในproject จะอยู่
   | ที่res/layout/main_Activity.xml
9 |     setContentView(R.layout.main_Activity);
10 |
11 |    // กำหนดค่าให้กับTextView ดังนั้นสามารถเรียกใช้งานผ่านชื่อตัวแปรได้ภายหลัง
12 |        mTextView = (TextView) findViewById(R.id.
   |            text_message);
13 |
14 |    // ตั้งค่าสำหรับกำหนดเงื่อนไขUI
15 |    // ตรวจสอบเพื่อยืนยันว่ากำลังใช้งานบนandroid Honeycomb
16 |    // หรือสูงกว่าเพื่อเรียกใช้ActionBar API
17 |        if (Build.VERSION.SDK_INT >= Build.
   |            VERSION_CODES.HONEYCOMB) {
18 |            // กำหนดให้icon ในaction bar ไม่ทำงานคล้าย
   |            ปุ่มHome
19 |            ActionBar actionBar = getActionBar();
20 |            actionBar.setHomeButtonEnabled(false);
21 |        }
22 |    }

```

รูปที่ 2.9: การใช้งาน onCreate()

เมื่อ onCreate() ทำงานเสร็จระบบจะเรียก onStart() และ onResume() memthod มาทำงานต่อในทันที แอคทิวิตี้จะไม่อยู่ใน Created Method หรือ Started Method ในทางเทคนิค แอคทิวิตี้จะแสดงขึ้นมาทันทีที่ onStart() ถูกเรียกใช้งานแต่ onResume() จะแสดงตามมาติด ๆ ในทันทีและจะอยู่ใน Resumed Method จนกว่าจะมีการเปลี่ยนแปลงบางอย่างเกิดขึ้นเช่น มีสายโทรเข้าหรือผู้ใช้เปิดไปแอคทิวิตี้อื่นหรือปิดหน้าจอลง

โครงสร้างของวัฏจักรแอคทิวิตี้ที่เน้นไปที่ 3 Method หลักที่ระบบเรียกใช้อัตโนมัติตามลำดับ เมื่อมีการสร้าง Instance ของแอคทิวิตี้ขึ้นมาใหม่ ได้แก่ onCreate(), onStart() และ

onResume() เมื่อลำดับการทำงานเสร็จสิ้นลงแอคทิวิตี้จะ มาอยู่ที่ Resumed Method ซึ่งผู้ใช้ใช้งานอยู่จนกว่าจะเปลี่ยนไปเรียกใช้งานแอคทิวิตี้อื่นขึ้นมา

2.1.8 การสิ้นสุดการทำงานของแอคทิวิตี้ (Destroy Activity)

ในขณะที่วัฏจักรของแอคทิวิตี้มี Method แรกคือ onCreate() และ Method สุดท้ายคือ onDestroy() ระบบจะเรียก Method นี้ในแอคทิวิตี้เป็นสัญญาณว่า Activity instance นี้จะถูกลบออกอย่างสมบูรณ์จากหน่วยความจำของระบบ (System memory)

แอปพลิเคชันส่วนใหญ่ไม่จำเป็นต้องกำหนด Method นี้เพราะ local class ถูกทำลายไปพร้อมกับแอคทิวิตี้และแทบจะถูกลบไปอย่างสมบูรณ์ในระหว่างเรียกใช้งาน onPause() และ onStop() อย่างไรก็ตามถ้าแอคทิวิตี้ของมีการกำหนดให้มีการทำงานแบบ background (ทำงานอยู่เบื้องหลัง) ในขั้นตอน onCreate() ด้วยแล้วหรือมีการใช้งานทรัพยากรของระบบเป็นระยะเวลานานเป็นเหตุให้ความจำถูกใช้งานจำนวนมาก หากไม่ทำการปิดการใช้งาน จึงควรรอยังยี่ที่จะเรียกใช้งาน Method onDestroy() เพื่อคืนค่าหน่วยความจำให้ระบบ

```

1  @Override
2  public void onDestroy() {
3      super.onDestroy(); // เรียกใช้งานsuperclass ทุกครั้ง
        เสมอ
4
5      // หยุดการติดตามหรือตรวจจับMethod ของActivity
6      android.os.Debug.stopMethodTracing();
7  }

```

รูปที่ 2.10: การใช้งาน onDestroy()

ระบบจะเรียกใช้งาน onDestroy() หลังจากเรียกใช้งาน onPause() และ onStop() แล้วในทุกเหตุการณ์ ยกเว้นกรณีมีการกำหนดให้เรียกใช้ finish() ภายใน onCreate() Method ในกรณีเช่น Activity หยุดชั่วคราวเพื่อทำการเรียกใช้งานแอคทิวิตี้อื่นเข้ามาแล้วทำการกำหนดให้มีการเรียกใช้งาน finish() ที่กำหนดใน onCreate() Method เพื่อทำลายแอคทิวิตี้ในกรณีนี้ระบบจะทำการเรียก onDestroy() ทันทีโดยไม่ผ่าน Callback Method อื่นเช่น ไม่ผ่าน onPause() หรือ onStop() แบบนี้เป็นต้น

2.2 ความรู้พื้นฐาน Java

Java [5] เป็นภาษาเขียนโปรแกรมเพื่อวัตถุประสงค์ทั่วไป โดยสามารถทำงานได้พร้อมกัน เป็นภาษาที่สร้างมาจากคลาส และสนับสนุนการเขียนโปรแกรมแบบออบเจกต์อย่างสมบูรณ์ และถูกออกแบบมาให้พร้อมสำหรับการใช้งานมากที่สุด ซึ่งมีเมธอดและคลาสต่าง ๆ อำนวยความสะดวกให้ใช้มากมาย โดยภาษา Java นั้นมีความตั้งใจว่าจะทำให้นักพัฒนาออกแบบและพัฒนาโปรแกรมง่ายขึ้น นั่นคือการเขียนเพียงครั้งเดียว แต่นำไปใช้งานได้ทุกที่หรือทุกแพลตฟอร์ม

แอปพลิเคชันของภาษา Java นั้นโดยปกติแล้วจะคอมไพล์เป็น bytecode ที่สามารถรันได้ใน Java virtual machine (JVM) ขึ้นกับสถาปัตยกรรมของคอมพิวเตอร์นั้นๆ และใน ปี 2016 ภาษา Java เป็นภาษาที่ได้รับความนิยมและใช้มากที่สุดในโลก โดยเฉพาะการใช้พัฒนาเว็บแอปพลิเคชัน

2.2.1 ประวัติความเป็นมาของภาษา Java

James Gosling Mike Sheridan และ Patrick Naughton ได้เริ่มก่อตั้งโปรเจกต์ภาษา Java ของพวกเขาเมื่อปี 1991 โดยในตอนแรกมันถูกพัฒนาสำหรับทีวีที่สามารถมีปฏิสัมพันธ์ได้ เช่น เล่นเกมในทีวีได้ แต่มันยากเกินไปในการที่จะใช้งานกับสายเคเบิลของทีวีดิจิทัลในเวลานั้น ในตอนแรกภาษา Java ใช้ชื่อว่า Oak เพราะว่ามีต้นโอ๊คยื่นออกไปยังออฟฟิศของ Gosling ต่อมาใช้ชื่อว่า Green และในตอนท้ายใช้ชื่อว่า Java มีที่มาจากกาแฟ Java

ภาษาได้รับการออกแบบให้มีรูปแบบทางภาษาเหมือนภาษา C และ C++ ซึ่งทำให้โปรแกรมเมอร์ส่วนมากนั้นคุ้นเคยกับมันได้ดีขึ้น และ Sun Microsystems เผยแพร่ Java 1.0 ในปี 1995 โดยมีคำกล่าวว่า "Write Once, Run Anywhere" (WORA) ซึ่งมันฟรี เขียนเพียงครั้งเดียว และสามารถนำไปรันได้บนทุกแพลตฟอร์ม

2.2.2 Java Compiler

ในการเขียนโปรแกรมในภาษา Java ต้องการ Java Compiler เพื่อทำการแปลงโค้ดของโปรแกรมที่เขียนเป็น bytecode เพื่อนำไปรันในแต่ละแพลตฟอร์มต่อไป โดยเรียกว่า Java Platform (JDK) ซึ่งประกอบไปด้วยคอมไพเลอร์ ในการแปลงโค้ดภาษา Java ให้เป็น Bytecode และ Java virtual machine (JVM) สำหรับรันโปรแกรมของภาษา Java ในแต่ละแพลตฟอร์ม สำหรับในบทเรียนนี้จะใช้ IDE ในการพัฒนาเพื่อความสะดวกและรวดเร็ว

2.2.3 โครงสร้างของภาษา Java

```
1  public class ClassName {  
2      /**  
3       * ฟังก์ชันที่เรียกทำงานทุกครั้งที่ใช้คำสั่ง java ClassName  
4       */  
5      public static void main(String[] args) {  
6          System.out.println("Hello สวัสดี");  
7      }  
8  }
```

รูปที่ 2.11: ตัวอย่างโปรแกรมภาษา Java ด้วย minted

รูปที่ 2.11 แสดงให้เห็นผลลัพธ์ของการใช้ minted เพื่อแสดงคำสั่งในเอกสาร Latex

```
1 package com.ubu.andapp;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.util.Log;
6
7 public class MainActivity extends Activity {
8
9     private String name = "MainActivity";
10
11     /** ฟังก์ชันที่จะทำงานเมื่อมีการเริ่มเรียกใช้งาน MainActivity */
12     @Override
13     public void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         Log.d(msg, "The onCreate() event");
17     }
18
19     /** ฟังก์ชันที่จะทำงานเมื่อมีการแสดง MainActivity ให้ผู้ใช้เห็น */
20     @Override
21     protected void onStart() {
22         super.onStart();
23         Log.d(msg, "The onStart() event");
24     }
25 }
```

รูปที่ 2.12: ตัวอย่างโปรแกรมภาษา Java ด้วย minted คลาส MainActivity

- Package เป็นกลุ่มของคลาสหรือไลบรารีมาตรฐานของภาษา Java ที่มีฟังก์ชันต่าง ๆ ให้ใช้มากมาย
- Class ในส่วนของการประกาศคลาส จะต้องประกาศคลาสให้ชื่อตรงกับไฟล์เสมอ นอกจาก Inner คลาสที่อยู่ในคลาสเดียวกัน โดยชื่อคลาสนั้นควรจะขึ้นต้นด้วยตัวใหญ่และถ้ามีหลายคำให้ใช้ตัวพิมพ์ใหญ่แบ่ง

```

1
2 public class ClassName {
3     /**
4      * ฟังก์ชันที่เรียกทำงานทุกครั้งที่ใช้คำสั่ง java ClassName
5      */
6     public static void main(String[] args) {
7         System.out.println("Hello สวัสดี");
8     }
9 }

```

รูปที่ 2.13: การประกาศคลาสในภาษา Java

- Method หลังจากคลาสสร้างแล้ว จะเป็นประกาศเมธอดภายในคลาส โดยในการที่จะรันโปรแกรมได้จะต้องมีเมธอดที่ชื่อว่า Main ดังตัวอย่างในโปรแกรมด้านบน มันเป็นที่แรกที่โปรแกรมจะเริ่มทำงาน 2.13

```

1 public static void main (String[] args) {
2     ...
3 }

```

รูปที่ 2.14: การประกาศเมธอด main ในภาษา Java

- Methodments เป็นคำสั่งของโปรแกรมเพื่อให้โปรแกรมทำงานตามต้องการ เช่น System.out.println("Hello World!"); เป็นการแสดงผลข้อความออกทางหน้าจอ โดยปกติโปรแกรมมักจะมีหลายคำสั่ง
- Semicolon ทุกคำสั่งการทำงานของโปรแกรมในภาษา Java จะจบด้วยเครื่องหมาย Semicolon (;) นั้นหมายความว่าสามารถเขียนโปรแกรมแบบไหนก็ได้ โดยคอมไพเลอร์จะทราบอัตโนมัติว่าสิ้นสุดคำสั่งที่ไหน
- ในภาษา Java สามารถใช้ White space ได้อย่างอิสระตามที่ต้องการ โดย White space จะประกอบไปด้วย Space bar Tab และ Enter (return) เพราะว่าคอมไพเลอร์ตรวจการสิ้นสุดของคำสั่งด้วย ; ใช้ while space ทำให้โค้ดอ่านเข้าใจง่าย และเป็นระเบียบ
- Literals คือค่าของข้อมูลใดๆ ที่กำหนดให้กับตัวแปรได้ เรียกว่า Constant Literals ทุกค่าที่เป็นไปได้ เช่น "MarcusCode" เป็น String Literals 10 เป็น Integer Literals หรือ true เป็น Boolean Literals โดย Literals เป็นได้แค่ Primitive data type เท่านั้น ตัวอย่าง

การกำหนดค่าหรือ Literals ให้กับตัวแปร

- Expression เป็นการกระทำระหว่างตัวแปรกับตัวดำเนินการเพื่อให้ได้ผลลัพธ์ใหม่ เช่น $4 + 3$ เป็น expression ของการบวกเลขและได้ผลลัพธ์เท่ากับ 7 หรือ $1 == 1$ เป็น expression ของการเปรียบเทียบระหว่างค่าสองค่าว่าเท่ากันหรือไม่ และได้ผลลัพธ์เป็น true
- Keyword คือคำที่สงวนไว้ในภาษา Java นั้นหมายความว่าไม่สามารถนำคำเหล่านี้ไปประกาศเป็นชื่อตัวแปร เมธอด หรือว่าคลาสได้ เพราะว่า Keyword ถูกใช้โดยคอมไพเลอร์เพื่อให้งานมันทำงานได้สมบูรณ์

2.3 ความรู้พื้นฐาน Java Script

JavaScript [6] คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็นภาษาสคริปต์เชิงวัตถุที่เรียกว่า "สคริปต์" (Script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของดุมมีการเคลื่อนไหวสามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (Interpret) หรือเรียกว่า อ็อบเจ็กต์โอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับภาษา HTML และภาษา Java Script ทำงานได้ทั้งทางฝั่งไคลเอนต์ (Client) และทางฝั่งเซิร์ฟเวอร์ (Server)

JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript

JavaScript สามารถทำให้ การสร้างเว็บเพจ มีลูกเล่น ต่าง ๆ มากมายและสามารถโต้ตอบกับผู้ใช้อย่างทันที เช่น การใช้เมาส์คลิกหรือการกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการและมีความน่าสนใจมากขึ้นประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยม

นิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนเบราว์เซอร์ที่สนับสนุน ซึ่งปัจจุบันเบราว์เซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ตามสิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ ๆ ออกมาด้วยดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ไปรันบนเบราว์เซอร์รุ่นเก่าที่ยังไม่สนับสนุนอาจจะทำให้เกิดข้อผิดพลาด (Error) ได้

2.3.1 ลักษณะการทำงานของ JavaScript

JavaScript เป็นภาษาสคริปต์เชิงวัตถุหรือเรียกว่าอ็อบเจกต์โอเรียนเต้ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนเอกสารด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับภาษา HTML และภาษาจาวาได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) โดยมีลักษณะการทำงานดังนี้

```

1 // import ฟังก์ชันสำคัญของ angular/core
2 import { Component, OnInit } from '@angular/core';
3
4 @Component({
5     // ประกาศชื่อ selector เพื่อให้ component อื่นสามารถนำไปใช้ได้
6     selector: 'student',
7     // ระบุชื่อไฟล์ html ที่ใช้ฟังก์ชันใน component โดยตรง
8     templateUrl: './student.component.html',
9     // ระบุ style ที่นำไปใช้กับ html
10    styleUrls: ['./student.component.css']
11 })
12 export class StudentComponent implements OnInit {
13
14    constructor() { }
15
16    ngOnInit() {}
17
18 }

```

รูปที่ 2.15: ตัวอย่างคำสั่งภาษา TypeScript ด้วย minted

- Navigator JavaScript เป็น Client-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลทางฝั่งไคลเอนต์ หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ใช้ไม่ว่าจะเป็นเครื่องพีซี (Personal computer, PC) เครื่องแมคอินทอช (Macintosh) หรืออื่น ๆ จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่
- LiveWire JavaScript เป็น Server-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลทางฝั่งเซิร์ฟเวอร์ (หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ให้บริการเว็บโดยอาจจะเป็นเครื่องของชั้นซิลิคอนกราฟิกส์หรืออื่น ๆ) สามารถใช้ได้เฉพาะกับ LiveWire ของเน็ตสเคป โดยตรง

2.3.2 JavaScript กับ HTML

การเขียน JavaScript อาจเขียนรวมอยู่ในไฟล์เดียวกันกับ HTML ได้ ซึ่งแตกต่างจากการเขียนโปรแกรมภาษา Java ที่ต้อง เขียนแยกออกเป็นไฟล์ต่างหากไม่สามารถเขียนรวมอยู่ในไฟล์เดียวกับ HTML ได้ วิธีการเขียน JavaScript เพื่อสั่งให้เว็บเพจทำงาน มีอยู่ด้วยกัน 2 วิธี ดังนี้

- เขียนด้วยชุดคำสั่งและฟังก์ชันของ JavaScript เอง
- เขียนตามเหตุการณ์ที่เกิดขึ้นตามการใช้งานจากชุดคำสั่งของ HTML

เมื่อเริ่มใช้งานโปรแกรมบราวเซอร์จะอ่านข้อมูลจากส่วนบนของเพจ HTML และทำงานไปตามลำดับจาก บนลงล่าง (top-down) โดยเริ่มที่ส่วน < HEAD >...< /HEAD > ก่อนจากนั้นจึงทำงานในส่วน < BODY >...< /BODY > เป็นลำดับต่อมา การทำงานของ JavaScript ดูไม่แตกต่างไปจาก HTML เท่าใดนัก แต่ HTML จะวางเลย์เอาต์โครงสร้างของอ็อบเจกต์ภายใน และส่วนเชื่อมโยงกับเว็บเพจเท่านั้น ในขณะที่ JavaScript สามารถเพิ่มเติมส่วนของการเขียนโปรแกรมและลอจิกเข้าไป

2.3.3 โครงสร้างภาษา

1. ตัวแปร (Variable) หมายถึง ชื่อหรือสัญลักษณ์ที่ตั้งขึ้นสำหรับการเก็บค่าใด ๆ ที่ไม่คงที่ โดยการจองเนื้อที่ในหน่วยความจำของระบบเครื่องที่เก็บข้อมูลซึ่งสามารถอ้างอิงได้ มีขนาดขึ้นอยู่กับชนิดของข้อมูลและค่าของข้อมูล ซึ่งค่าในตัวแปรนี้สามารถเปลี่ยนแปลงได้ตามคำสั่งในการประมวลผล
2. การตั้งชื่อ (Identifier or Name) เป็นชื่อที่ตั้งขึ้นมาเพื่อกำหนดให้เป็นชื่อของโปรแกรมหลัก, ฟังก์ชัน, ตัวแปร, ค่าคงที่, คำสั่ง และคำสงวน โดยมีหลักการตั้งชื่อว่า
 - เริ่มต้นด้วยตัวอักษรในภาษาอังกฤษ ตามด้วยตัวอักษรหรือตัวเลขใด ๆ ก็ได้
 - ห้ามเว้นช่องว่าง
 - ห้ามใช้สัญลักษณ์พิเศษ ยกเว้นขีดล่าง (_) และดอลลาร์ (\$)
 - สำหรับความยาวของชื่อใน JavaScript จะมีความยาวเท่าใดก็ได้ แต่ที่นิยมใช้ ไม่เกิน 20 ตัวอักษร
 - การตั้งชื่อมีข้อพึงระวังว่า จะต้องไม่ซ้ำกับคำสงวน (Reserve word) และตัวอักษรของชื่อจะจำแนกแตกต่างกันระหว่างอักษรตัวพิมพ์เล็กกับอักษรตัวพิมพ์ใหญ่

- ควรจะตั้งชื่อโดยให้ชื่อนั้นมีสื่อความหมายให้เข้ากับข้อมูล สามารถอ่านและเข้าใจได้ ตัวอย่างชื่อที่ถูกต้อง Hahaha, I_Love_you, Doll\$ เป็นต้น
3. คำสงวน(Reserve word)เป็นคำที่มีความหมายเฉพาะตัวในภาษา JavaScript สงวนไม่ให้เกิดการตั้งชื่อซ้ำกับชื่อโปรแกรม, ฟังก์ชัน, ตัวแปร, ค่าคงที่ และคำสั่ง คำสงวน สามารถเรียกใช้ได้ทันทีโดยไม่ต้องมากำหนดความหมายใหม่แต่อย่างใด
 4. ชนิดของข้อมูลของตัวแปร (Data Type) เป็นการกำหนดประเภทค่าของข้อมูลให้กับตัวแปร เพื่อให้เหมาะสมกับการอ้างอิงข้อมูลจากตัวแปรในการใช้งาน ชนิดข้อมูลของตัวแปรนั้นมีอยู่ด้วยกัน 4 ชนิด ได้แก่
 - number หมายถึง ข้อมูลชนิดตัวเลข ประกอบด้วย เลขจำนวนเต็ม (Integer) และ เลขจำนวนจริง (Floating)
 - logical หมายถึง ข้อมูลทางตรรกะ มี 2 สถานะ คือ จริง (True) และเท็จ (False)
 - string หมายถึง ข้อมูลที่เป็นข้อความ ซึ่งจะต้องกำหนดไว้ในเครื่องหมายคำพูด ("...")
 - null หมายถึง ไม่มีค่าข้อมูลใดๆ ซึ่งค่า null ใช้สำหรับการยกเลิกพื้นที่เก็บค่าของตัวแปรออกจากหน่วยความจำ
 5. การประกาศตัวแปร (Declarations) เป็นการกำหนดชื่อและชนิดข้อมูลให้กับตัวแปรเพื่อนำไปใช้ในโปรแกรม โดยการตั้งชื่อจะต้องคำนึงถึงค่าของข้อมูลและ ชนิดของข้อมูลที่อ้างอิง นอกจากนี้การตั้งชื่อควรให้สื่อความหมายของข้อมูล และอักษรของชื่อจะจำแนกแตกต่างกันระหว่างอักษรตัวพิมพ์เล็กกับอักษรตัวพิมพ์ใหญ่
- รูปแบบ Var ชื่อตัวแปร; เป็นรูปแบบการประกาศตัวแปรปกติหรือ Var ชื่อตัวแปร = ข้อมูล; เป็นรูปแบบการกำหนดค่าเริ่มต้น ในกรณีที่ต้องการกำหนดตัวแปรหลายตัวในบรรทัดเดียวกันให้ใช้เครื่องหมาย คอมม่า (,) คั่นระหว่างชื่อตัวแปรและปิดท้ายด้วยเครื่องหมายเซมิโคลอน (;) การกำหนดค่าให้กับตัวแปร
- รูปแบบ ชื่อตัวแปร = ค่าของข้อมูล โดยที่ค่าของข้อมูล ได้แก่
- ข้อมูลที่เป็นตัวเลข โดยกำหนดตัวเลขไปได้เลย เช่น num = 500
 - ข้อมูลในทางตรรกะ ได้แก่ จริง (True) หรือ เท็จ (False) เช่น test = True;
 - ข้อมูลสตริง ให้กำหนดอยู่ในเครื่องหมายคำพูด ("...") เช่น name = "Adisak";
- ตัวแปรมี 2 จำพวก หากกำหนดชื่อตัวแปรไว้ที่โปรแกรมหลักโดยไม่ได้อยู่ภายในขอบเขตฟังก์ชันใด ๆ เรียกว่าเป็นตัวแปรแบบโกลบอล (Global) ตัวแปรจำพวกนี้จะมีค่าคงอยู่ใน

หน่วยความจำตลอกการทำงานของโปรแกรม ทำให้สามารถเรียกใช้ได้จากทุก ๆ ส่วนของโปรแกรม รวมถึงภายในฟังก์ชันต่าง ๆ ด้วย แต่ถ้ากำหนดตัวแปรไว้ภายในขอบเขตฟังก์ชันใด ๆ จะเรียกว่าเป็นตัวแปรแบบ โลคัล (Local) เพราะจะเป็นตัวแปรที่มีค่าคงอยู่ และสามารถเรียกใช้ได้เฉพาะ ภายในขอบเขตของฟังก์ชันนั้น ๆ เท่านั้น

6. ตัวแปรแบบอาร์เรย์ (Array) หมายถึงตัวแปรซึ่งมีค่าได้หลายค่าโดยใช้ชื่ออ้างอิงเพียงชื่อเดียว ด้วยการให้หมายเลขลำดับเป็นตัวจำแนกความแตกต่างของค่าตั้งแปรแต่ละตัว ถ้าจะเรียกตัวแปรชนิดนี้ว่า "ตัวแปรชุด" ก็เห็นจะไม่ผิดนัก ตัวแปรชนิดนี้มีประโยชน์มาก ลองคิดถึงค่าข้อมูลจำนวน 100 ค่า ที่ต้องการเก็บไว้ในตัวแปรจำนวน 100 ตัว อาจทำให้ต้องกำหนดตัวแปรที่แตกต่างกันมากถึง 100 ชื่อ กรณีอย่างนี้ควรจะทำอย่างไรดี แต่ด้วยการใช้สมบัติอาร์เรย์ สามารถนำตัวแปรหลาย ๆ ตัวมาอยู่รวมเป็นชุดเดียวกันได้และสามารถเรียกใช้ตัวแปรทั้งหมดโดยระบุผ่านชื่อเพียงชื่อเดียวเท่านั้น ด้วยการระบุหมายเลขลำดับ หรือดัชนี (index) กำกับตามหลังชื่อตัวแปร ตัวแปรเพียงชื่อเดียวจึงมีความสามารถเทียบได้กับตัวแปรนับร้อยตัว พันตัว (ตัวที่ 1) ในตัวแปรแบบอาร์เรย์มีดัชนีเป็น 0 ส่วนตัวแปรต่อ ๆ ไปก็จะมีดัชนีเป็น 1,2,3,... ไปตามลำดับ เมื่อต้องการระบุชื่อตัวแปรแบบอาร์เรย์แต่ละตัว ก็จะใช้รูปแบบ name[0], name[1],... เรียงต่อกันไปเรื่อย ๆ สามารถสร้างตัวแปรอาร์เรย์ใหม่ด้วย myArray = new Array() ดังนี้

```
1 myArray[0] = 17;
2 myArray[1] = "Nun" ;
3 myArray[2] = "Stop" ;
```

รูปที่ 2.16: ตัวแปรอาร์เรย์

7. ค่าคงที่ (Literal หรือ Constant) หมายถึง ค่าของข้อมูลที่ไม่ได้กำหนดแล้วจะทำการเปลี่ยนแปลงค่าเป็นอย่างอื่นไม่ได้ ชนิดข้อมูลของค่าคงที่ได้แก่

- เลขจำนวนเต็ม (Integer) เป็นตัวเลขที่ไม่มีเศษทศนิยม สามารถเขียนให้อยู่ในแบบเลขฐานสิบ (0-9), เลขฐานสิบหก (0-9, A-F) หรือ เลขฐานแปด (0-7) โดยการเขียนเลขฐานแปดให้ นำหน้าด้วย O (Octenary) ส่วนการเขียนเลขฐานสิบหกให้นำหน้าด้วย Ox หรือ OX (Hexadenary)
- เลขจำนวนจริง (Floating) ใช้รูปแบบการเขียนโดยประกอบไปด้วยตัวเลข จุดทศนิยม

และตัวเลขยกกำลัง E (Exponential) เช่น \square 5.00E2 จะหมายถึงค่า 5.00 คูณด้วย 10 ยกกำลัง 2 จะมีค่าเป็น 500 \square 3.141E5 จะหมายถึงค่า 3.141 คูณด้วย 10 ยกกำลัง 5 จะมีค่าเป็น 314,1000

- ค่าบูลีน (Boolean) เป็นค่าคงที่เชิงตรรกะ คือมีค่าเป็น จริง(True) และ เท็จ(False) เท่านั้น
- ข้อความสตริง (String) เป็นค่าคงที่แบบข้อความที่อยู่ภายในเครื่องหมายคำพูด ("..." หรือ '...') เช่น "บริษัท เอ็กซ์ทรีม จำกัด", 'นางนฤมล เวชตระกูล'

- รหัสคำสั่งพิเศษ (Character escape code) เป็นการกำหนดรหัสเพื่อควบคุมงานพิมพ์ สตริงโดยใช้เครื่องหมาย Backslash นำหน้าตัวอักษรที่ต้องการกำหนดเป็นรหัส เพื่อให้กลายเป็นรหัสคำสั่งพิเศษ รหัส Character escape code
- นิพจน์ (Expression) เป็นข้อความที่ใช้กำหนดค่าของข้อมูล เช่น การบวกตัวเลข การเปรียบเทียบข้อมูล โดยการกำหนดชื่อของตัวแปร ตามด้วยเครื่องหมายที่ต้องการกระทำ (Operations) ต่อข้อมูลเป็นผลให้เกิดค่าข้อมูลใหม่ค่าหนึ่งให้กับตัวแปรเพื่อนำไปใช้งาน

นิพจน์ JavaScript มีด้วยกัน 3 ชนิดดังนี้

- นิพจน์คณิตศาสตร์ (Arithmetic) เป็นนิพจน์ที่ใช้เครื่องหมายทางคณิตศาสตร์เป็นตัวกระทำ ผลลัพธ์ที่ได้จะมีค่าเป็นตัวเลขให้กับตัวแปร เช่น ให้ตัวแปร num เก็บตัวเลข 5000 จะเขียนได้ดังนี้ num = 5000;
- นิพจน์ตรรกะ (Logical) เป็นนิพจน์ในการเปรียบเทียบข้อมูลโดยใช้เครื่องหมายในการเปรียบเทียบเพื่อตรวจสอบข้อมูลในการเปรียบเทียบว่าจริงหรือเท็จ เช่น กำหนดให้

```
1 a = 50;
2 b = 70;
3 c = b>a;
```

รูปที่ 2.17: การเปรียบเทียบว่าจริงหรือเท็จ

ผลลัพธ์ที่ได้คือ c จะมีค่าเป็นจริง (True)

- นิพจน์ข้อความ (String) เป็นนิพจน์เกี่ยวกับการกำหนดข้อความ การเชื่อมต่อข้อความ ใช้ประมวลผลข้อความในลักษณะต่าง ผลลัพธ์ที่ได้จึงมีค่าเป็นตัวอักษรหรือ

ข้อความเสมอ เช่น ให้ตัวแปร name เก็บชื่อ Adisak จะเขียนได้ดังนี้ name = "Adisak";

10. ตัวดำเนินการ (Operator) หมายถึง เครื่องหมายกำหนดกรรมวิธีทางคณิตศาสตร์, พีชคณิต, บูลีน, การเปรียบเทียบ ระหว่างข้อมูล 2 ตัว ซึ่งเรียกว่า โอเปรานด์ (Operand) โดยอาจมีค่าเป็นตัวเลข ข้อความ ค่าคงที่ หรือตัวแปรต่าง ๆ

11. ชนิดของตัวดำเนินการ

- ตัวดำเนินการคณิตศาสตร์ (Arithmetic operator) หมายถึง ใช้สำหรับคำนวณโอเปรานด์ที่เป็นค่าคงที่หรือตัวแปรก็ได้ โดยให้ค่าผลลัพธ์เป็นตัวเลขค่าเดียว
- ตัวดำเนินการเชิงเปรียบเทียบ (Comparison operator) หมายถึง เครื่องหมายในการเปรียบเทียบข้อมูล ผลลัพธ์ที่ได้จะมีค่าตรรกบูลีนเป็น จริง (True) และ เท็จ (False)
- ตัวดำเนินการเชิงตรรกะ (Logical operator) เป็นเครื่องหมายที่ให้ค่าจริง (True) และ เท็จ (False) ในการเปรียบเทียบ
- ตัวดำเนินการเชิงข้อความ (String operator) เป็นการเชื่อมโยงข้อความเข้าด้วยกัน (concatenation) โดยใช้เครื่องหมายบวก (+) เป็นตัวกระทำ เช่น

```
1 Name = "Bodin";
2 Say = "Hey " + Name;
```

รูปที่ 2.18: ตัวดำเนินการเชิงข้อความ

ผลลัพธ์ที่ได้ Say จะมีข้อความเป็น Hey Bodin

- ตัวดำเนินการระดับบิต (Bitwise operator) เป็นการดำเนินการเชิงตรรกะในระดับบิต โดยจะใช้มุมมองในแบบเลขฐาน 2 มาจัดการกับข้อมูล นั่นคือ ข้อมูลตัวเลขนั้นจะถูกแปลงเป็นเลขฐานสองในหน่วยความจำในขณะที่มีการดำเนินการเชิงตรรกะในระดับบิต ซึ่งโดยปกติแล้วการกระทำใน JavaScript จะอยู่ในระดับตัวอักษร ที่เรียกว่า ระดับไบต์ (byte)

2.4 ความรู้พื้นฐาน Vue.js Fronted Framework

Vue.js [7] เป็นเฟรมเวิร์คที่เน้นเรื่องการทำ User Interface คือเป็นจาวาสคริปต์เฟรมเวิร์คที่เข้ามาช่วยเรื่องการแสดงผล เป็นโอเพนซอร์ส (Open-Source) โดยวัตถุประสงค์หลักของการทำงานใช้วิวเจเอสคือ เพิ่มความสามารถการทำเว็บแอปพลิเคชันแบบ Single Page Application หรือ SPA

วิวเจเอสเป็น Frontend Framework ช่วยจัดการและทำให้การพัฒนาเว็บง่ายขึ้น เน้นพัฒนาส่วนติดต่อผู้ใช้ของเว็บ เช่น components, declarative UI, hot-reloading, time-travel debugging และอื่น ๆ ให้สะดวกต่อการใช้งาน พยายามลดสิ่งที่ไม่จำเป็นต่อการพัฒนาเว็บออก ทำให้มีขนาดเล็กและง่ายต่อการนำไปใช้ของนักพัฒนา

มีสถาปัตยกรรมแบบ Adoptable สามารถนำไปประยุกต์ใช้จากหน้าเพจที่มีอยู่เดิม ทั้งนี้หากต้องการใช้งานความสามารถขั้นสูงสำหรับแอปพลิเคชันที่ซับซ้อน จำเป็นต้องเพิ่มปลั๊กอิน (Plugin) อันได้แก่ Vue-router, Vuex และเครื่องมืออื่น ๆ ตามที่เว็บทางการของวิวเจเอสแนะนำ

2.4.1 การติดตั้ง

การติดตั้งเพื่อใช้งานวิวเจเอสทำได้หลายวิธีในที่นี้ขออธิบายด้วยกัน 2 วิธีได้แก่

- CDN (Content Delivery Network) ทำหน้าที่ให้การให้ User สามารถดาวน์โหลด Resource ต่าง ๆ บนเครือข่ายได้ เช่น jQuery, Bootstrap, jQuery UI, AngularJS เป็นต้น
- Vue Cli เป็นชุดคำสั่งที่ใช้ในการสร้าง Project ด้วย Vue.js ซึ่งได้มีการรวบรวมชุดเครื่องมือรวมถึงไลบรารีต่าง ๆ ที่จำเป็นต่อการพัฒนาไว้ โดยเราสามารถเลือก Templates เพื่อใช้งานตามความเหมาะสมของงานได้
 1. ทำการใส่คำสั่ง `npm install -global vue-cli` เพื่อติดตั้ง Vue CLI
 2. ทำการใส่คำสั่ง `vue init webpack my-project` เพื่อการสร้าง Project
 3. ทำการใส่คำสั่ง `cd my-project` เพื่อเข้าไปที่ path ของ Project
 4. ทำการใส่คำสั่ง `npm install` เพื่อติดตั้ง node module ซึ่งเป็นชุดเครื่องมือที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชัน
 5. ทำการใส่คำสั่ง `npm run dev` เริ่มการทำงานของโปรแกรม

2.4.2 คุณสมบัติของวิวเจเอส

- Templates วิวเจเอสใช้งาน HTML เป็น syntax ซึ่งอนุญาตให้ผู้ใช้ทำการความคุมการแสดงผลภายใต้ DOM โดยทำผ่าน Vue instance's data ความสามารถหลักของ Templates คือ สามารถความคุมการแสดงผลด้วยการเพิ่มคำสั่งหรือเงื่อนไขเข้าไปภายใต้แท็ก HTML ปกติ
- ReActivity ความสามารถที่โดดเด่นที่เห็นได้ชัดเจนคือ การทำงานในส่วนของการ re-rendering ที่รวดเร็ว
- Components เป็นอีกหนึ่งความสามารถพิเศษของวิวเจเอส ในแอปพลิเคชันขนาดใหญ่ จำเป็นต้องมีการแบ่งแอปพลิเคชันทั้งหมดออกเป็นส่วนย่อย ๆ เพื่อให้สามารถนำกลับมาใช้งานได้และทำให้ง่ายต่อการจัดการ

```

1 <div id="tuto">
2   <button v-bind:initial_count="0"></
   buttonclicked>
3 </div>
4 <script>
5   Vue.component('buttonclicked', {
6     props: ["initial_count"],
7     data: function() {return {"count": 0} }
8     ,
9     template: '<button v-on:click="onclick">
10      Clicked {{ count }} times</button>',
11     Methods: {
12       "onclick": function() {
13         this.count = this.count + 1;
14       }
15     },
16     mounted: function() {
17       this.count = this.initial_count;
18     }
19   });
20   new Vue({
21     el: '#tuto',
22   });
23 </script>

```

รูปที่ 2.19: Components ของวิวเจเอส

จากภาพที่ 2.19 แสดงวิวเจเอส component โดยในบรรทัดที่ 5 ถึง 17 เป็นการสร้างวิวเจเอส component

- Routing ในเทคโนโลยี SPA ข้อเสียหนึ่งข้อคือ ไม่สามารถส่งผ่านข้อมูลไปยัง Component อื่นได้ เนื่องจาก SPA เป็น URL-based response เพื่อแก้ปัญหาดังกล่าววิวเจเอสได้สร้างส่วนประสานของตัวเองขึ้น คือ Routing ซึ่งใช้ส่งผ่านข้อมูลไปยัง Component อื่น

2.4.3 ข้อดีของ Vue.js

- ช่วยแยก Logic การตัดสินใจออกจากโค้ด (Code) การแสดงผล
- ช่วยแยกหน้าเว็บออกเป็น Component ทำให้การจัดการง่ายขึ้นและนำกลับมาใช้ได้
- ช่วยจัดการเรื่อง Dynamic data
- มีการเก็บสถานะต่าง ๆ ไว้ที่จุดเดียวเพื่อให้ง่ายต่อการจัดการและการเรียกใช้งานของ Component โดยการใช้งาน Vuex

2.5 ความรู้พื้นฐาน ScanLibrary

ScanLibrary [8] เป็นไลบรารีที่ใช้สแกนเอกสารต่าง ๆ บนแอนดรอยด์ที่พัฒนามาจาก OpenCV ผลจากการประมวลผลจะได้เป็น Uri ซึ่งสามารถนำมาแปลงเป็นข้อมูลรูปภาพเพื่อบันทึกลงฐานข้อมูลได้

2.5.1 ขั้นตอนการติดตั้งและใช้งาน

- ทำการดาวน์โหลดไลบรารีจาก <https://github.com/jhansiredy/AndroidScannerDemo>
- ทำการนำเข้าไลบรารีผ่านคำสั่ง `compile project(':scanlibrary')`
- เมื่อต้องการใช้งาน ให้นำโค้ดดังรูปที่ 2.20 ไปวางไว้ที่เมธอดที่ต้องการเริ่มการทำงาน

```

1  int REQUEST_CODE = 99;
2  int preference = ScanConstants.OPEN_CAMERA;
3  Intent intent = new Intent(this, ScanActivity.
    class);
4  intent.putExtra(ScanConstants.
    OPEN_INTENT_PREFERENCE, preference);
5  startActivityForResult(intent, REQUEST_CODE);

```

รูปที่ 2.20: การเรียกใช้งาน

- ให้ทำการรับค่าที่ถูกส่งกลับจากไลบรารีด้วยคำสั่ง ดังรูปที่ 2.21

```

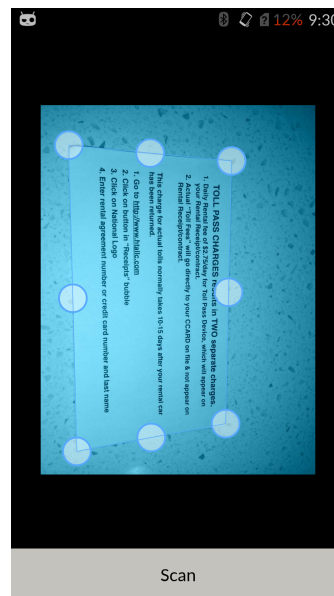
1  @Override
2  protected void onActivityResult(int requestCode, int
    resultCode, Intent data) {
3      super.onActivityResult(requestCode, resultCode,
        data);
4      if (requestCode == REQUEST_CODE && resultCode ==
        Activity.RESULT_OK) {
5          Uri uri = data.getExtras().getParcelable(
            ScanConstants.SCANNED_RESULT);
6          Bitmap bitmap = null;
7          try {
8              bitmap = MediaStore.Images.Media.
                getBitmap(getContentResolver(), uri);
9              getContentResolver().delete(uri, null,
                null);
10             scannedImageView.setImageBitmap(bitmap);
11         } catch (IOException e) {
12             e.printStackTrace();
13         }
14     }
15 }

```

รูปที่ 2.21: การรับค่าจากไลบรารี

2.5.2 ตัวอย่างการใช้งาน

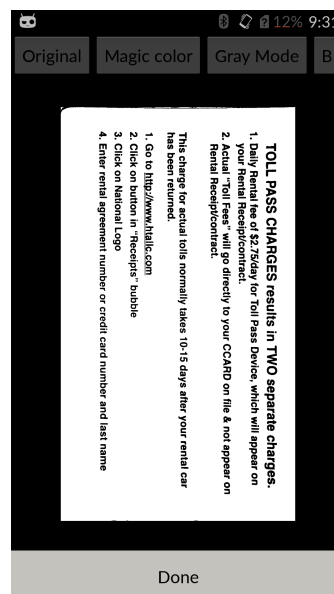
- ผู้ใช้ทำการถ่ายภาพเอกสารบนพื้นเรียบ



รูปที่ 2.22: ถ่ายภาพเอกสาร

ที่มา : <https://github.com/jhansiredy/AndroidScannerDemo>

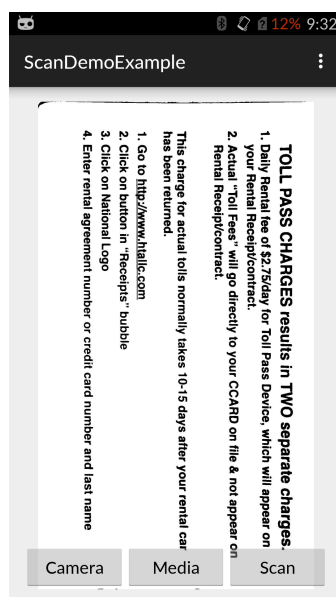
- ทำการปรับรูปภาพ เช่น การปรับสีรูปภาพ การปรับเป็นสีขาวดำ เป็นต้น



รูปที่ 2.23: ปรับแต่งภาพถ่าย

ที่มา : <https://github.com/jhansiredy/AndroidScannerDemo>

- ภาพที่ได้จากการประมวลผล



รูปที่ 2.24: ผลการทำงาน

ที่มา : <https://github.com/jhansiredy/AndroidScannerDemo>

2.6 ความรู้พื้นฐานของระบบ XX เดิม

2.6.1 ความเป็นมา

ระบบ XX เดิม มีประวัติดังนี้

2.6.2 วิสัยทัศน์และพันธกิจ

- วิสัยทัศน์ (Vision) ”เป็นองค์กรหลักที่ XX”
- พันธกิจ (Mission)
 1. สนับสนุนและส่งเสริม XX
 2. พัฒนางค์กรสู่ความเป็นเลิศด้วยนวัตกรรมที่ทันสมัย โดยใช้หลักบริหารจัดการที่ดี

2.6.3 ขั้นตอนการดำเนินการ XX

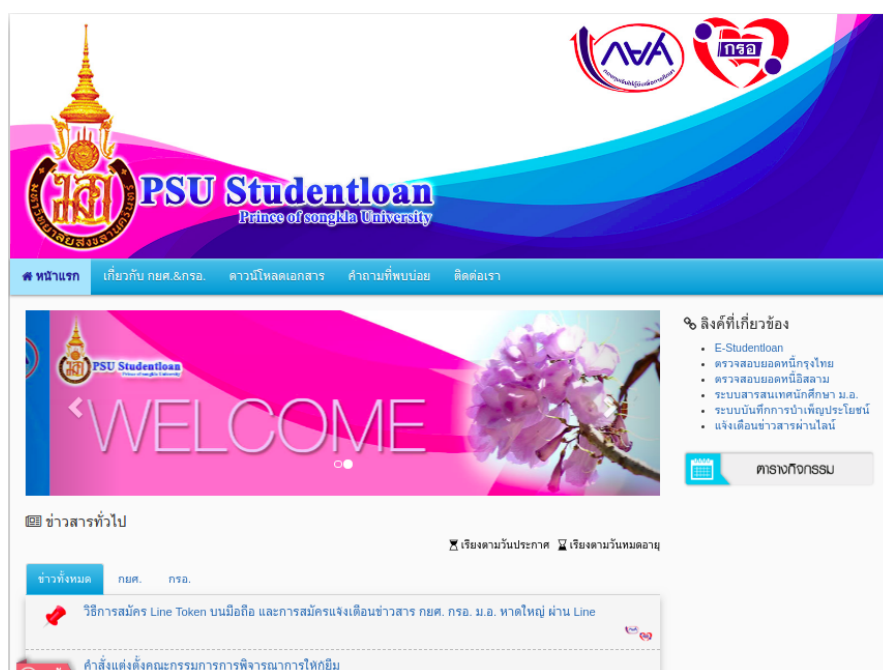
- อธิบายขั้นตอนที่ XX
- อธิบายขั้นตอนที่ XX
- อธิบายขั้นตอนที่ XX

2.7 เอกสารและงานวิจัยที่เกี่ยวข้อง

กล่าวถึงเอกสาร งานวิจัย หรือระบบงานที่คล้ายกันโดยแบ่งเป็น subsection โดยแต่ละหัวข้อให้อธิบายความสำคัญ ฟังก์ชันการทำงาน ข้อจำกัดหรือข้อแตกต่างจะระบบที่จะทำ เช่น

2.7.1 เว็บไซต์ XX

PSU Studentloan [9] เป็นเว็บไซต์ที่ให้บริการนักศึกษากองทุนเงินให้กู้ยืมเพื่อการศึกษา มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ สามารถใช้งานได้ที่ <https://studentloan.psu.ac.th/home> มีฟังก์ชันการทำงานพื้นฐานอันได้แก่ การดูข่าวสารประชาสัมพันธ์ ประมวลภาพกิจกรรม ดาวน์โหลดเอกสารที่เกี่ยวข้อง เป็นต้น



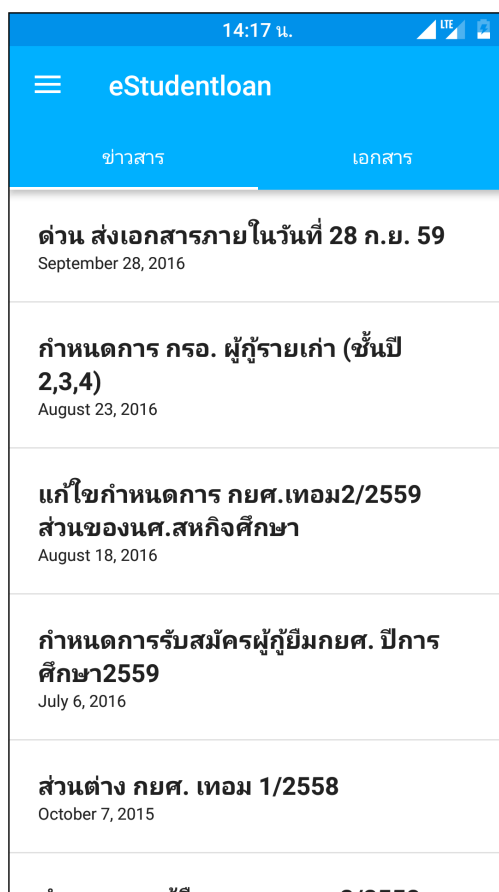
รูปที่ 2.25: หน้าแรกของเว็บไซต์ PSU Studentloan

ที่มา : <https://studentloan.psu.ac.th/home>

2.7.2 แอปพลิเคชัน XX

eStudentloan [10] เป็นแอนดรอยด์แอปพลิเคชันที่พัฒนาโดยฝ่ายกิจการนักศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น เพื่อให้บริการนักศึกษากองทุนเงินให้กู้ยืมเพื่อการศึกษาสถาบันเทคโนโลยีไทย-ญี่ปุ่น สามารถดาวน์โหลดเพื่อใช้งานได้ที่ <https://play.google.com/store/apps/details?id=th.co.dest.anek.studentloan> โดยแอปพลิเคชันมีฟังก์ชันการทำงาน คือ ติดตามข่าวสาร

จากหน่วยงาน



รูปที่ 2.26: ข่าวสารจากหน่วยงาน

ที่มา : <https://play.google.com/store/apps/details?id=th.co.dest.anek.studentloan>

บรรณานุกรม

- [1] ทศพล ต้นสมบัติ. (ม.ป.ป.). ระบบปฏิบัติการ android [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าสุด/ระบบปฏิบัติการ-android.html> .
- [2] Google. (2557). Open handset alliance [ออนไลน์]. สืบค้นเมื่อ 20 พฤษภาคม 2561. จาก <http://www.openhandsetalliance.com/> .
- [3] กอบเกียรติ สระอุบล. 2549. การพัฒนา App Android. กรุงเทพฯ: มีเดีย เนทเวิร์ค.
- [4] Sleeping For Less. (2557). Activity life cycle [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <http://www.akexorcist.com/2016/04/why-do-we-need-to-know-about-activity-life-cycle-th.html> .
- [5] ดร.วีระศักดิ์ ชิงถาวร. 2545. JAVA PROGRAMMING Volume I. กรุงเทพฯ: ซีเอ็ดดูเคชั่น.
- [6] mindphp. (2555). Javascript คืออะไร [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <https://goo.gl/FAeTb2> .
- [7] Vue.js Team. (2560). Introduction [ออนไลน์]. สืบค้นเมื่อ 2 พฤษภาคม 2561. จาก <https://vuejs.org/v2/guide/> .
- [8] jhansireddy. (2557). Androidscannerdemo [ออนไลน์]. สืบค้นเมื่อ 3 เมษายน 2561. จาก <https://www.studentloan.or.th/index.php/aboutus> .
- [9] งานกองทุนเงินให้กู้ยืมเพื่อการศึกษา มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่. (2554). กยศ.ม.อ(psu studentloan) [ออนไลน์]. สืบค้นเมื่อ 20 พฤษภาคม 2561. จาก <https://studentloan.psu.ac.th/home> .
- [10] Tni.Student. (2560). แอปพลิเคชัน estudentloan [ออนไลน์]. สืบค้น เมื่อ 20 พฤษภาคม 2561. จาก <https://play.google.com/store/apps/details?id=th.co.dest.anek.studentloan> .

ประวัติผู้เขียน

ชื่อ-สกุล: นายชื่อ สกุล

รหัสประจำตัวนักศึกษา: 5811400000

วันเกิด: XX YY 25ZZ

ที่อยู่ที่สามารถติดต่อได้: XX ม.XX ต.XX อ.XX จ.XX 34XXX

เบอร์โทรศัพท์: (+66) XX XXX XXXX

อีเมลล์: xxxxxxx.yy.59@ubu.ac.th

ระดับมัธยมต้น: โรงเรียน XX จังหวัด XX

ระดับมัธยมปลาย: โรงเรียน XX จังหวัด XX

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ สาขาวิทยาการ คอมพิวเตอร์ คณะ
วิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี