บทที่ 2

ทฤษฎีและระบบงานที่เกี่ยวข้อง

2.1 Flutter และ ไลบรารีที่เกี่ยวข้อง

2.1.1 Flutter Facebook login

คือชุดคำสั่งของ Facebook สำหรับ Android [6] ช่วยให้ผู้ใช้เข้าสู่ระบบแอปพลิเคชันด้วยการเข้าสู่ระบบด้วย Facebook เมื่อเข้าสู่แอปพลิเคชันด้วย Facebook ผู้ใช้สามารถให้สิทธิ์การอนุญาตแก่แอปพลิเคชันเพื่อให้รับข้อมูลหรือดำเนินการบางอย่างบน Facebook แทนผู้ใช้ได้ การเข้าสู่ระบบนี้สามารถใช้ได้บน iOS, Android, เว็บ, แอปพลิเคชันบนเดสก์ท็อป และอุปกรณ์ต่างๆ เช่น สมาร์ททีวีและอุปกรณ์เชื่อมต่ออินเทอร์เน็ตต่าง ๆ สามารถใช้การเข้าสู่ระบบด้วย Facebook ได้ง่าย ๆ เพื่อการยืนยันตัวตนหรือเพื่อทั้งการยืนยันตัวตนและการเข้าถึงข้อมูลการแจ้งเตือนเมื่อผู้ใช้เข้าสู่ระบบผ่าน Facebook โดยการกดปุ่มที่อยู่บนหน้าแอปพลิเคชัน จะเรียกใช้ชุดคำสั่งของ Facebook ในการเข้าสู่ระบบและใช้สิทธิ์อนุญาตในการเข้าสู่ระบบ

2.1.2 Notification

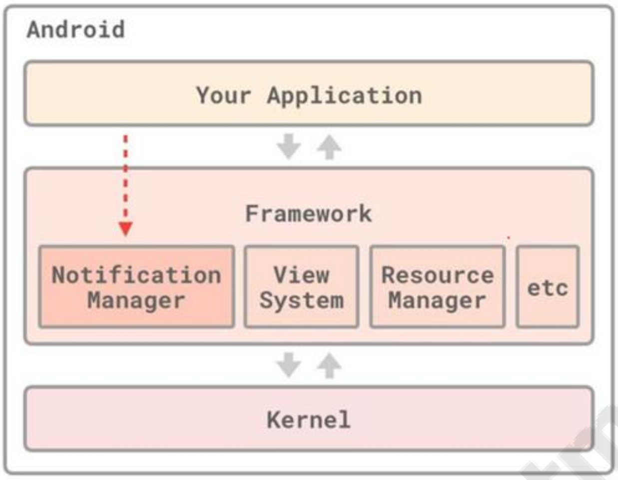
Notification [7] เป็นหนึ่งในช่องทางของแอนดรอยด์ที่เปิดให้แอปพลิเคชันสามารถส่งข้อความให้ผู้ใช้เห็นได้ โดยผู้ใช้ไม่จำเป็นต้องเปิดแอปพลิเคชันขึ้นมา และผู้ใช้ก็สามารถสั่งงานบางอย่างผ่าน Notification ตัวนั้น ๆ กลับมาได้อีกด้วย นี่คือคำจำกัดความสั้น ๆ ของ Notification ซึ่งเป็นหนึ่งในความสามารถของแอนดรอยด์ที่เปิดให้นักพัฒนาสามารถใช้งาน Notification ในสถานการณ์ต่าง ๆ ที่เหมาะสมกับแอปพลิเคชันของตนเองได้

ตามธรรมชาติของ Notification บนอุปกรณ์แอนดรอยด์ที่เป็นแบบ Phone กับ Tablet จะแสดงผลอยู่ที่ด้านบนของหน้าจอ และสามารถดู Notification ที่เคยแสดงได้จาก System Bar ของแอนดรอยด์และเนื่องจาก Notification เป็นแค่ช่องทางหนึ่งในการส่งข้อความให้ผู้ใช้ จึงหมายความว่าตัวมันเองไม่ได้ยึดติดกับการทำงานของแอปพลิเคชันเลย สิ่งที่ Notification สนใจก็มีแค่รายละเอียดของข้อมูลที่ต้องการให้แสดงใน Notification เท่านั้นเอง

2.1.2.1 การทำงานของ Notification บนแอนดรอยด์

การทำงานจะทำงานอยู่บน Android Framework ซึ่งเป็นหัวใจหลักในการทำงานของระบบแอนดรอยด์ที่มีไว้ให้แอปพลิเคชันต่าง ๆ เรียกใช้งานซึ่งใน Android Framework ก็จะมีการแบ่งการทำงานแยกกันออกไปตามหน้าที่ และส่วนที่ทำหน้าที่เกี่ยวกับระบบ Notification ก็จะมีชื่อเรียกว่า Notification Manager

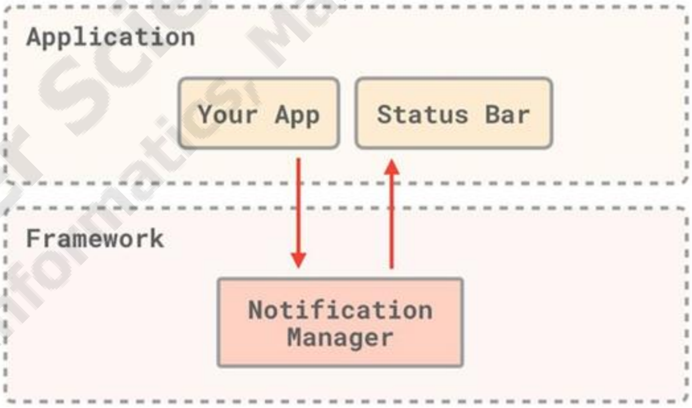
Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 2.1 การทำงานของ Notification บนแอนดรอยด์

ที่มา https://akexorcist.dev/notification-in-android-part-1/

และ Status Bar ของแอนดรอยด์ที่ทำหน้าที่แสดง Notification จากแอปพลิเคชันต่าง ๆ ก็จะรับข้อมูลมาจาก Notification Manager ดังนั้นเมื่อใดก็ตามที่แอปพลิเคชันในเครื่องส่ง Notification ไปให้ Notification Manager ก็จะถูกส่งไปเพื่อแสดงใน Notification Drawer ที่อยู่ใน Status Bar นั่นเอง



ภาพประกอบที่ 2.2 Status Bar ของแอนดรอยด์

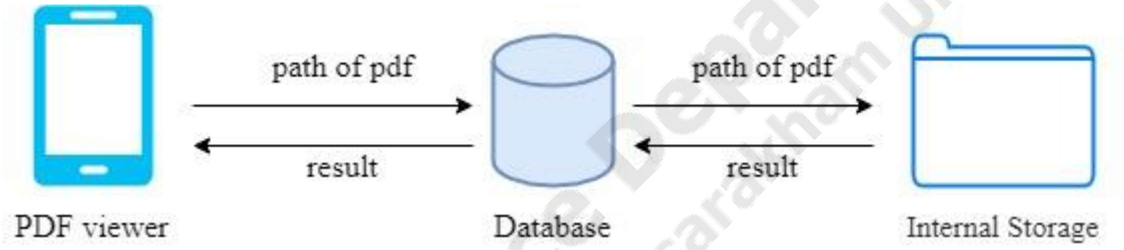
ที่มา https://akexorcist.dev/notification-in-android-part-1/

ด้วยเหตุนี้ จึงเป็นที่มาว่าทำไมทุกครั้งที่นักพัฒนาต้องการแสดงข้อความแบบ Notification ให้กับผู้ใช้ จะต้องเรียกคำสั่งผ่าน Notification Manager อยู่เสมอ

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.1.3 Flutter PDF Viewer (SfPdfViewer)

Flutter PDF Viewer เป็น Package หนึ่งของ Flutter [5] มีคุณสมบัติคือสามารถเปิดอ่านไฟล์ที่มีนามสกุล .pdf ที่มีขนาดใหญ่ได้ สามารถใช้งานได้อย่างมีประสิทธิภาพในภายแพลตฟอร์ม Android ,ios , เว็บ และ macOS พร้อมทั้งมีคุณสมบัติต่างๆ เช่น การขยาย การเลื่อนดูหน้าต่างๆในเอกสาร การนำทางไปยังหน้าที่ต้องการ เลือกข้อความหรือคัดลอกข้อความ ค้นหาข้อความ เปิดหน้าบุ๊คมาร์คจากลิงก์ในไฟล์ได้ เป็นต้น โดยการเรียกที่จะเรียกเปิดดูไฟล์ .pdf นั้น ก่อนอื่นเราต้องเก็บไฟล์ .pdf ไว้ในที่เก็บข้อมูลภายใน แล้วก็จะเก็บตำแหน่งที่เก็บไฟล์ไว้ในฐานข้อมูล พอจะเปิดไฟล์ .pdf ก็จะนำตำแหน่งที่ตั้งที่เก็บไว้มาค้นหาดูจากที่เก็บข้อมูลภายในเพื่อทำการเรียกเปิดไฟล์ .pdf



ภาพประกอบที่ 2.3 การทำงานของ Flutter PDF Viewer
ที่มา https://pub.dev/packages/syncfusion_flutter_pdfviewer

2.1.4 Flutter TimelineTile

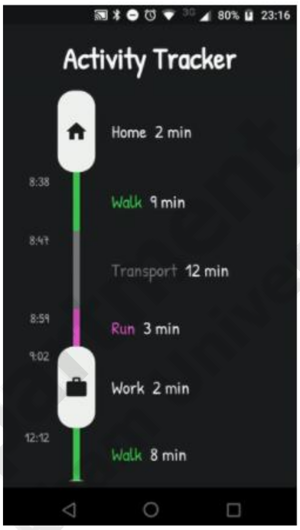
Flutter TimelineTile เป็น package หนึ่งของ flutter [4] ที่ช่วยให้ผู้ใช้สามารถสร้าง Timeline ขึ้นมาให้อยู่ในรูปแบบต่างๆ โดยที่เราจะเรียกใช้ใน flutter นั้น คือ TimelineTile() ซึ่งส่วนประกอบของ component นี้จะมีหลักๆให้ใช้งานคือ verticel ซึ่งตัว verticel นี้จะเป็นเส้นที่นำมารวมกันเป็นเส้น Timeline

ภาพประกอบที่ 2.4 ตัวอย่าง Timeline
ที่มา https://pub.dev/packages/timeline_tile

Computer Science Department
Faculty of Informatics, Mahasarakham University

ซึ่งตัว verticel หรือเส้นเหล่านี้จะสามารถเรียกใช้ได้ 4 แบบ ซึ่งในแต่ละแบบนั้นมีการวางที่ไม่เหมือนกัน โดยเส้น 4 แบบนี้ถ้านำมาจัดวางประกอบกันและนำข้อมูล ไอคอนรูป หรือปุ่มต่างๆ มารวมด้วย ก็จะได้ Timeline รูปแบบต่างๆ ยกตัวอย่าง Timeline ต่างๆ

ภาพประกอบที่ 2.5 ไทม์ไลน์เวลาส่งมอบ



ภาพประกอบที่ 2.6 ไทม์ไลน์กิจกรรม

ที่มา https://pub.dev/packages/timeline_tile

2.1.5 Spring Boot

Spring Boot [8] แอปพลิเคชันอำนวยความสะดวกแก่นักพัฒนาในการ configuration ค่าต่าง ๆ ของ Spring Framework มีชุด Starter Library ให้เรียกใช้งานได้สะดวก ลดขั้นตอนการเพิ่ม Library อื่น ๆ ที่เกี่ยวข้อง มี Auto configuration เพื่อให้ลดขั้นตอนการกำหนดค่าต่าง ๆ มี Built-in web server เช่น Tomcat ช่วยให้สร้างแอปพลิเคชันแบบแยกส่วนการทำงานได้ง่าย

ชื่อ Dependency	คำอธิบาย
spring-boot-starter	ใช้กับแอปพลิเคชันส่วนหลักของ Spring Boot
spring-boot-starter-web	ใช้ในการพัฒนาเว็บแบบ full-stack
spring-boot-starter-jdbc	ใช้กับการติดต่อฐานข้อมูลผ่าน JDBC
spring-boot-starter-data-jpa	สนับสนุน JPA (Java Persistence API) ประกอบด้วย spring-data-jpa, spring-orm และ Hibernate
spring-boot-starter-data-mongodb	สนับสนุน MongoDB NoSQL Database
spring-boot-starter-data-rest	ใช้พัฒนา Web API (RESTful API) จาก spring-data-rest-webmvc
spring-boot-starter-freemarker	ใช้ Render หน้าแสดงผลโดยไม่ใช้โค้ดภาษาจาวาด้วย FreeMarker templating engine
spring-boot-starter-thymeleaf	ใช้ Render หน้าแสดงผลโดยไม่ใช้โค้ดภาษาจาวาด้วย Thymeleaf templating engine
spring-boot-starter-ws	ใช้พัฒนา Web Services
spring-boot-starter-cloud-connectors	ใช้ติดต่อกับ "Spring Cloud Connectors" เช่น Cloud Foundry และ Heroku
spring-boot-starter-security	ใช้จัดการความปลอดภัยในการเข้าถึงระบบ
spring-boot-starter-social-facebook	ใช้ติดต่อกับ Facebook API
spring-boot-starter-social-twitter	ใช้ติดต่อกับ Twitter API

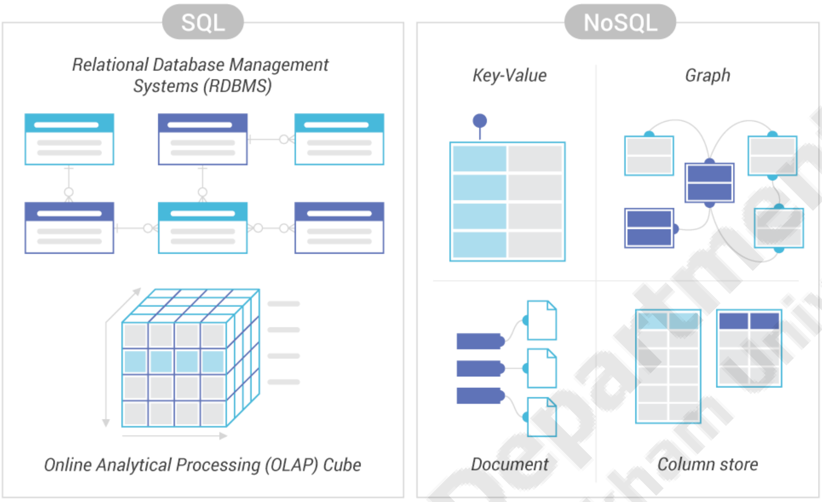
ภาพประกอบที่ 2.7 Library ที่เกี่ยวข้อง

ที่มา https://ttime.in.th/course-jw/slide/11%20-%20Spring%20Boot.pdf

Computer Science Department, Faculty of Informatics, Mahasarakham University

2.2 ฐานข้อมูลที่เกี่ยวข้อง

2.2.1 โนเอสคิวแอล (NoSQL)



ภาพประกอบที่ 2.8 ภาพเปรียบเทียบระหว่าง SQL และ NoSQL

ที่มา https://www.scylladb.com/resources/nosql-vs-sql/

Computer Science Department
Faculty of Informatics, Mahasarakham University

โนเอสคิวแอล (NoSQL) [2] สามารถแบ่งได้เป็น 2 คือ ข้อมูลแบบที่มีโครงสร้าง (structured data) ซึ่งเก็บอยู่ในรูปแบบของตารางซึ่งประกอบไปด้วยแถว (row) และคอลัมน์ (column) และข้อมูลแบบที่ไม่มีโครงสร้าง (unstructured data) ซึ่งไม่อยู่ในรูปแบบของตาราง เช่น ข้อความ (text) กราฟ (graph) หรือเครือข่าย (network) ส่วนใหญ่เมื่อพูดถึงฐานข้อมูลจะคุ้นเคยกับฐานข้อมูลที่เป็นเชิงสัมพันธ์ (relational database) ที่เก็บข้อมูลที่เป็นตารางซึ่งมีจำนวนคอลัมน์หรือฟิลด์ (field) ที่แน่นอน แต่ในบางครั้งข้อมูลบางประเภทไม่สามารถระบุจำนวนคอลัมน์ที่แน่นอนได้ก่อน และอาจจะมีการเพิ่มหรือลดจำนวนคอลัมน์เหล่านี้ ตัวอย่างเช่น ข้อมูลที่อยู่บนเครือข่ายสังคมออนไลน์ (social network) หรือ กระทู้ (topic) ต่างๆ ในเว็บบอร์ด (webboard) ดังนั้นการเก็บข้อมูลเหล่านี้ลงในฐานข้อมูลเชิงสัมพันธ์ที่เป็นตารางจึงไม่เหมาะสมนัก จึงเกิดเป็นเทคโนโลยีใหม่ขึ้นมาเรียกว่า ฐานข้อมูลโนเอสคิวแอล (NoSQL) ซึ่งสามารถเก็บข้อมูลเป็นไฟล์เอกสารที่มีจำนวนฟิลด์ไม่เท่ากันได้หรือเก็บเป็นโครงสร้างของกราฟเลยก็ได้ โดยทั่วไปแล้วฐานข้อมูลโนเอสคิวแอลจะแบ่งเป็น 4 ประเภท คือ

ฐานข้อมูลแบบคีย์-แวลู (key-value database) เป็นการเก็บข้อมูลขนาดเล็ก โดยประกอบด้วยคีย์และแวลู ลักษณะคล้ายกับผลลัพธ์ที่ได้จากการทำแมปรีดิวซ์ ใช้ในการเก็บข้อมูลที่เป็นแคช (cache) จากฐานข้อมูลเชิงสัมพันธ์ หรือ เก็บข้อมูลที่ดึงมาได้จาก IoT (Internet of Things)

ฐานข้อมูลแบบเอกสาร (document database) [3] เป็นการเก็บข้อมูลในลักษณะของไฟล์เอกสาร ซึ่งภายในมีแท็ก (tag) ระบุค่าต่าง ๆ ไว้ โดยหนึ่งแถวหรือเรคอร์ด (record) ในฐานข้อมูลเชิงสัมพันธ์จะเทียบได้กับไฟล์เอกสาร 1 ไฟล์ และแท็กต่างๆ จะเทียบได้กับคอลัมน์หรือฟิลด์ในฐานข้อมูล

เชิงสัมพันธ์ ซึ่งข้อดีของการเก็บข้อมูลแบบนี้คือฟิลด์ในเอกสารไม่จำเป็นต้องมีจำนวนเท่ากันได้ ฐานข้อมูบแบบเอกสารที่นิยมใช้กันอย่างแพร่หลาย คือ ฐานข้อมูล MongoDB ซึ่งเก็บไฟล์ในลักษณะของ เจซอน (JSON)

ฐานข้อมูลแบบคอลัมน์ (column-oriented database) โดยทั่วไปการเก็บข้อมูลในฐานข้อมูลเชิงสัมพันธ์จะเป็นลักษณะของแถว (row-oriented) คือ เก็บข้อมูลในแต่ละแถวเรียงต่อกันในดิสก์ ทำให้การดึงข้อมูลออกมาทั้งตารางหรือบางเรคอร์ดทำได้อย่างมีประสิทธิภาพ แต่ถ้านำมาใช้ในการคำนวณค่าทางสถิติ เช่น การหาค่าเฉลี่ย (average) ของคอลัมน์จะทำงานได้ช้าลง จึงได้มีการคิดฐานข้อมูลแบบคอลัมน์ขึ้นมา โดยการเก็บข้อมูลจะเก็บเรียงเป็นคอลัมน์แทนที่จะเป็นแถว ทำให้เวลาต้องการคำนวณค่าของคอลัมน์ที่ต้องการใช้งานก็สามารถดึงข้อมูลที่เรียงต่อกันในดิสก์ได้เลยซึ่งก็จะเร็วกว่าการเก็บแบบแถว ฐานข้อมูลแบบนี้นิยมใช้ในการทำธุรกิจอัจริยะเพราะการคำนวณค่าทางสถิติไปใส่ไว้ในคลังข้อมูลทำได้อย่างรวดเร็ว

ฐานข้อมูลแบบกราฟ (graph database) ในบางครั้งข้อมูลมีความสัมพันธ์กันในรูปแบบของเครือข่ายหรือกราฟ ทำให้การเก็บข้อมูลในลักษณะของกราฟทำการประมวลผลได้รวดเร็วขึ้น

2.2.2 MongoDB

MongoDB [9] เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มี relation (ความสัมพันธ์) ของตารางแบบ SQL ทั่ว ๆ ไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทน การบันทึกข้อมูลทุก ๆ record ใน MongoDB เราจะเรียกมันว่า **Document** ซึ่งจะเก็บค่าเป็น key และ value จะเห็นว่ามันก็คือ JSON

```
{
    "_id": ObjectId("554b8ee746e04bc5503aef47"),
    "name": "Chai"
}
```

ภาพประกอบที่ 2.9 ตัวอย่างแสดงการเขียน MongoDB

ที่มา https://devahoy.com/blog/2015/08/getting-started-with-mongodb/

และการเก็บข้อมูล document ใน MongoDB จะถูกเก็บไว้ใน **Collections** (เปรียบเทียบได้กับ Table ใน Relational Database ทั่ว ๆ ไป) แต่แตกต่างกันที่ collection ไม่จำเป็นที่จะต้องมี schema เหมือนกันก็สามารถบันทึกข้อมูลได้

2.2.2.1 ข้อดีของ MongoDB

- MongoDB เป็น database แบบ Document-Oriented โดยลักษณะการเก็บข้อมูลจะใช้รูปแบบ format เป็น Json Style โดย Row แต่ละ Row ไม่จำเป็นต้องมีโครงสร้างข้อมูลเหมือนกัน

Computer Science Department
Faculty of Informatics, Mahasarakham University

- MongoDB ใช้ระบบการจัดการ memory แบบเดียวกับ cached memory ใน linux ซึ่งจะปล่อยให้ OS เป็นคนจัดการ Memory

- ใช้ภาษา javascript เป็นคำสั่งในการจัดการข้อมูล

- read-write ข้อมูลรวดเร็ว

- write ข้อมูล แบบ asynchronous (คล้าย INSERT DELAYED ของ MyISAM ใน MySQL) คือไม่ต้องรอ Insert เสร็จจริงก็ทำงานต่อได้

- MongoDB มี Capped Collection ซึ่งจะทยอยลบข้อมูลเก่าที่เก็บไว้นานเกินไป แล้วเอาข้อมูลใหม่มาใส่แทนได้ จะ clear ข้อมูลที่เก็บมานานเกินไปไว้ให้อัตโนมัติ ข้อมูลไม่ใหญ่กว่าที่เรากำหนด

- ค้นหาข้อมูลได้รวดเร็ว

- สามารถใช้เครื่อง server ที่ไม่ต้องคุณภาพสูงมากแต่แบ่งกันทำงานหลาย ๆ เครื่อง ซึ่งประหยัดงบได้มากกว่าใช้เครื่องคุณภาพสูงเพียงเครื่องเดียว

- สามารถเขียนเป็นชุดคำสั่งได้ คล้าย ๆ กับการเขียน PL/SQL

2.2.2.2 ข้อเสีย MongoDB

- ถ้า project เก่ามีการ JOIN กันซับซ้อนก็จะเปลี่ยนมาใช้ MongoDB ได้ยาก

- กินพื้นที่การเก็บข้อมูลมากกว่า MySQL พอสมควร เพราะไม่มี Schema ดังนั้น Schema จริง ๆ จะอยู่ในทุก row ของฐานข้อมูล ทำให้ข้อมูลใหญ่กว่า MySQL

- หากใช้งานจน disk เต็ม จะ clear พื้นที่ disk ให้ใช้งานต่อยาก เพราะการสั่ง delete row ไม่ทำให้ฐานข้อมูลเล็กลง ต้องสั่ง compact เองซึ่งต้องมีที่ว่างที่ disk อีกลูกมากพอๆ กับพื้นที่ข้อมูลที่ใช้อยู่ปัจจุบันเป็น buffer ในการลดขนาด

- หากต้องการใช้งานเป็นฐานข้อมูลหลักแทน MySQL ควรมีเครื่องอย่างน้อย 3 เครื่องที่เป็น physical แยกกันทำ replication กัน เพื่อเพิ่ม durability ของข้อมูล เนื่องจากข้อมูลส่วนใหญ่ของ MongoDB จะเก็บใน Memory เป็นระยะเวลาหนึ่ง หากเครื่องดับไปเครื่อง ข้อมูลที่ยังค้างใน Memory แต่ยังไม่ write ลง disk จะสูญหายทันที

2.3 ระบบงานที่เกี่ยวข้อง

ตัวอย่างระบบที่เกี่ยวข้องกับแอปพลิเคชัน Travel Trip Planner มี 3 ระบบดังนี้

2.3.1 TripIt

TripIt เป็นแอปพลิเคชันที่สามารถเพิ่มทริป เพิ่มรายละเอียดการเดินทาง ที่พัก และกิจกรรมได้ สามารถกำหนดจุดเริ่มต้นและจุดหมายปลายทางต่างๆ ที่เราอยากไปเมื่อกรอกข้อมูลทุกอย่างแล้ว

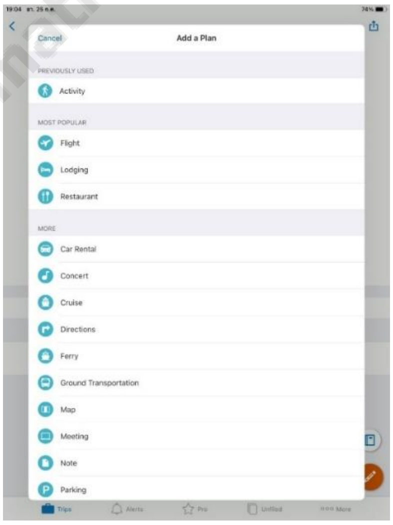
Computer Science Department
Faculty of Informatics, Mahasarakham University

แอปจะสรุปออกมาเป็นไทม์ไลน์ให้เราเลยว่าวันนี้เราต้องทำอะไรที่ไหนอย่างไร และเวลาใด นอกจากจะช่วยทำให้เราเห็นภาพรวมของทริปแล้ว ยังทำให้เราจัดกระเป๋าได้ง่ายขึ้นด้วย เพราะเราได้เห็นว่าต้องไปไหน ไปทำอะไรบ้างนั่นเอง

ภาพประกอบที่ 2.10 ตัวอย่างหน้าเพิ่มทริป (TripIt)

ที่มา แอปพลิเคชัน TripIt

เพิ่มทริป เพิ่มชื่อทริป เมืองที่เราจะเดินทางไป วันที่เริ่มต้นและสิ้นสุด และคำอธิบายทริป

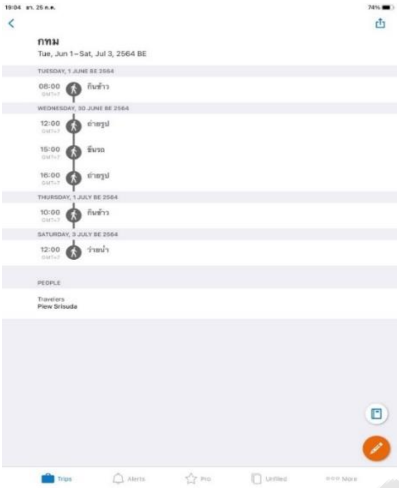


ภาพประกอบที่ 2.11 ตัวอย่างหน้าเพิ่มแพลนในทริป (TripIt)

ที่มา แอปพลิเคชัน TripIt

เพิ่มแพลน เป็นการเพิ่มกิจกรรม การเดินทาง หรือรายละเอียดอื่น ๆ ที่อยู่ในทริป โดยที่กิจกรรมแต่ละประเภทจะมีรายละเอียดที่แตกต่างกันออกไป

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 2.12 ตัวอย่างหน้าไทม์ไลน์ (TripIt)

ที่มา แอปพลิเคชัน TripIt

หน้าไทม์ไลน์ เมื่อกำหนดกิจกรรมเสร็จ แอปพลิเคชันจะสรุปไทม์ไลน์ออกมาให้เราดูว่าในแต่ละวันเรามีกิจกรรมที่ต้องทำอะไรบ้าง

2.3.2 My Trip List

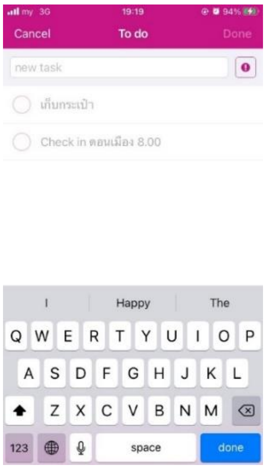
My Trip List สามารถสร้างทริปและเพิ่มการเดินทางต่าง ๆ ได้ และเรายังสามารถเพิ่ม Check Boxes ได้อีกด้วย เหมือนเป็นสิ่งที่ต้องทำทั้งก่อนจะไปเที่ยวและระหว่างเดียว อย่าง ‘อย่าลืมเช็กอินออนไลน์’ , ‘ต้องซื้อของฝากให้แม่’ หรือจะเป็น ‘กินอาหารร้านดัง’เรียกได้ว่าแอปเดียวจบจริง ๆ ตั้งแต่เช็คว่าจองตั๋วหรือยัง, แพ็คของครบไหม ไปจนถึงซื้อของฝากครบหรือเปล่า แบบนี้ต้องโหลดไว้ใช้

ภาพประกอบที่ 2.13 ตัวอย่างหน้าเพิ่มทริป (My Trip List)

ที่มา แอปฟลิเคชัน My Trip List

เพิ่มทริป สามารถใส่ชื่อทริป วันที่เดินทาง และสามารถเลือกได้ว่าเดินทางแบบไหน

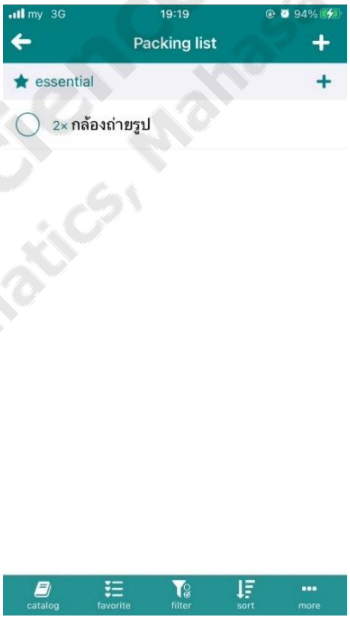
Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 2.14 ตัวอย่างหน้าเพิ่มสิ่งที่ต้องทำ (My Trip List)

ที่มา แอปพลิเคชัน My Trip List

สิ่งที่ต้องทำ สามารถเพิ่มสิ่งที่เราต้องทำก่อนหรือในระหว่างทริปได้ โดยที่เราสามารถมากดเลือกได้ว่าเราทำสิ่งไหนไปบ้างแล้ว



ภาพประกอบที่ 2.15 ตัวอย่างหน้าสิ่งที่ต้องเตรียมจัดกระเป๋า (My Trip List)

ที่มา แอปพลิเคชัน My Trip List

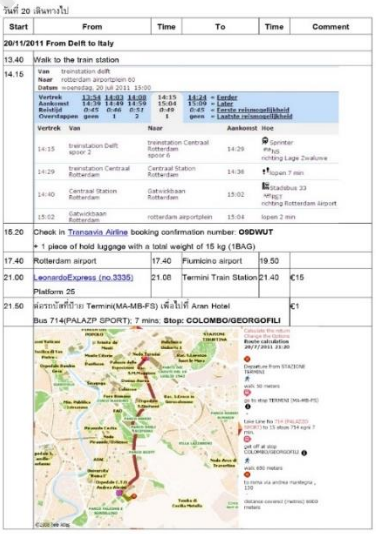
หน้าสิ่งที่ต้องเตรียมใส่กระเป๋าเดินทาง ในหน้านี้เราสามารถกำหนดได้ว่าเราจะเอาอะไรไปบ้างและมีจำนวนเท่าไหร่ เมื่อเราเก็บกระเป๋าเรียบร้อย เราสามารถมาเลือกได้ว่าเราเอาอะไรใส่กระเป๋าไปแล้วบ้าง

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 2.16 ตัวอย่างหน้าเพิ่มรูป (My Trip List)
ที่มา แอปพลิเคชัน My Trip List
หน้าเพิ่มรูป เราสามารถเพิ่มรูปรายละเอียดต่าง ๆ ที่เกี่ยวกับทริปได้

2.3.3 การวางแผนเที่ยวด้วยตนเอง

การวางแผนเที่ยวด้วยตนเองมีมานานแล้วเนื่องจากสมัยก่อนแอปพลิเคชันในมือถือยังไม่อำนวยความสะดวกเหมือนกับยุคปัจจุบัน ซึ่งเราจะเห็นว่านักเดินทางจะมีกระดาษที่วางแผนกิจกรรมและรายละเอียดการเดินทางในแต่ละทริปไว้ ตั้งแต่เอกสารการจองตั๋วเครื่องบิน การจองโรงแรมหรือไม่ว่าจะเป็นแผนที่ หรือรูปถ่ายสถานที่ต่าง ๆ ก็เป็นสิ่งที่ต้องมีในการวางแผนด้วยตนเองเช่นกัน

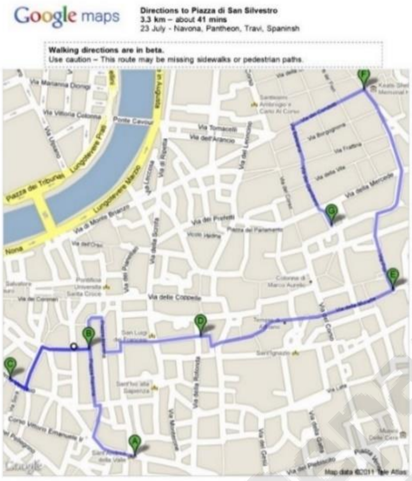

วันที่ 20 เดินทางไป

Start	From	Time	To	Time	Comment
20/11/2011 From Delft to Italy					
13.40	Walk to the train station				
14.15	Van treinstation delft Naar rotterdam airportplein 60				
15.20	Check in Transavia Airline booking confirmation number: O9DWUT + 1 piece of hold luggage with a total weight of 15 kg (1BAG)				
17.40	Rotterdam airport	17.40	Fiumicino airport	19.50	
21.00	LeonardoExpress (no.3335) Platform 25	21.08	Termini Train Station	21.40	€15
21.50	ต่อรถบัสที่ป้าย Termini(MA-MB-FS) เพื่อไปที่ Aran Hotel Bus 714(PALAZP SPORT); 7 mins; Stop: COLOMBO/GEORGOFILI				€1

ภาพประกอบที่ 2.17 ตัวอย่างการวางแผน (การวางแผนเที่ยวด้วยตนเอง)
ที่มา ข้อมูลจากอาจารย์พชระ พฤกษะศรี

Computer Science Department
Faculty of Informatics Mahasarakham University

การวางแผนจะต้องมีเวลาของกิจกรรมต่าง ๆ รวมถึงเวลาของขนส่งสาธารณะที่เราจะใช้เดินทางไปยังปลายทาง เนื่องจากเราไม่มีแอปพลิเคชันจึงต้องมีรายละเอียดของการเดินทางอย่างละเอียด



ภาพประกอบที่ 2.18 ตัวอย่างหน้าแผนที่ของการเดินทาง (การวางแผนเที่ยวด้วยตนเอง)
ที่มา ข้อมูลจากอาจารย์พชระ พฤกษะศรี
แผนที่ แสดงรายละเอียดของเส้นทางที่เราจะใช้เดินทางไปยังจุดหมายปลายทาง

ภาพประกอบที่ 2.19 ตัวอย่างสถานที่ในการเดินทาง (การวางแผนเที่ยวด้วยตนเอง)
ที่มา ข้อมูลจากอาจารย์พชระ พฤกษะศรี

สถานที่ในการเดินทาง เป็นรูปภาพของจุดต่าง ๆ ของการเดินทาง ไม่ว่าจะเป็นจุดขึ้นรถ จุดลงรถ ถนนที่ใช้เดินทางไปยังสถานที่ต่าง ๆ

Computer Science Department
Faculty of Informatics Mahasarakham University

2.4 ตารางเปรียบเทียบ

ตารางที่ 2.1 ตารางเปรียบเทียบระบบที่เกี่ยวข้อง

ฟังก์ชันการทำงาน	TripIt	My Trip List	การวางแผนด้วยตนเอง	ระบบที่พัฒนา
สมัครเป็นสมาชิก	✓			✓
ล็อกอินระบบ	✓			✓
แก้ไขข้อมูลของตนเอง	✓			✓
สามารถเพิ่มทริป	✓	✓	✓	✓
สามารถเพิ่มกิจกรรม	✓	✓	✓	✓
สามารถแชร์ไทม์ไลน์ทริป				✓
สามารถดูทริปของคนอื่น				✓
สามารถโหลดทริปของคนอื่น				✓
สามารถเพิ่มอัลบั้มรูป		✓		✓
สามารถแก้ไขทริป	✓	✓	✓	✓
สามารถแก้ไขกิจกรรม	✓	✓	✓	✓

Computer Science Department
Faculty of Informatics, Mahasarakham University