

TGR 2019 : Almighty (KMUTT)
Project Management
Smart National Historical Park

1. Historical park login system - ระบบ login ผ่าน line

- ฟัง LINE เขียน code เพื่อ train bot ให้ได้ตอบกับนักท่องเที่ยว
- ฟัง LINE เขียน Line ads เพื่อโฆษณาโบราณสถาน และกระตุ้นการท่องเที่ยว

2. Historical information service by using Line beacon (ยื่นใกล้ๆแล้วบอกข้อมูลประวัติศาสตร์)

- ฟัง data analytic เขียน code เพื่อตรวจจับสัญญาณ bluetooth เมื่อมีคนอยู่ในระยะโบราณสถานหรือโบราณวัตถุ จากบอร์ด ESP32 พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดึงจับค่าจาก ESP32 โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฟัง LINE เขียนโค้ดเพื่อแสดงข้อมูลของโบราณสถาน หรือโบราณวัตถุอื่นๆ ผ่าน LINE chat bot โดยอัตโนมัติ

3. Distance routing calculation

- ฟัง LINE เก็บข้อมูล GPS เพื่อคำนวณระยะทางที่สั้นที่สุดในแต่ละจุดของโบราณสถาน แล้วแจ้งเตือนระยะทางกับเวลาโดยประมาณให้กับนักท่องเที่ยว

4. Best viewpoint for taking photo in (ถ่ายรูปอัตโนมัติ โดยจับ gesture)

- ฟัง hardware เขียน code เพื่อตรวจจับท่าทางโดยใช้ gesture-detection sensor จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดึงจับค่าจาก gesture-detection sensor โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฟัง hardware เขียนโค้ดสั่งการ digital output เพื่อถ่ายรูปโดยอัตโนมัติ

5. Tourist security by using voice recognition

- ฟัง hardware เขียน code ให้รับเสียงโดยใช้ omnidirectional microphones จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดึงจับค่าจาก omnidirectional microphones โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฟัง data analytic เขียนเพื่อวิเคราะห์ว่าข้อความที่ได้รับมา สอดคล้องข้อความรูปแบบใด
- ฟัง hardware เขียนโค้ดเพื่อโต้ตอบกลับในรูปแบบเสียง

6. Climate control

- ฟัง hardware เขียน code ให้ค่าจาก sensor เพื่อแจ้งเตือนสภาพแวดล้อมที่มีค่าเกินกำหนด จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดึงจับค่าจาก sensor โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฟัง hardware เขียนโค้ดสั่งการ digital output ให้ปรับสภาพแวดล้อมให้เหมาะสมกับโบราณสถานและวัตถุ
- ฟัง data analytic นำค่าจาก database มาวิเคราะห์สถิติแล้วทำนายค่าอุณหภูมิ ความชื้น และความดันในอนาคตโดยใช้ Tensorflow

7. Intruder detector (กันคนมาทำลายหรือขโมยของในโบราณสถาน)

- ฟัง hardware เขียน code ให้ค่าจาก accelerometer เพื่อตรวจจับการสั่นสะเทือนจากการถูกผู้ไม่ประสงค์ดี ทำความเสียหายต่อโบราณวัตถุหรือโบราณสถาน จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)

- เขียน api สำหรับ server เพื่อดักจับค่าจาก accelerometer โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฝัง hardware เขียนโค้ดสั่งการ digital output เช่น สัญญาณกันขโมย เพื่อแจ้งเตือนผู้ไม่ประสงค์ดี
- ฝัง LINE เขียนโค้ดเพื่อแจ้งเตือนเจ้าหน้าที่ให้รับทราบถึงผู้ไม่ประสงค์ดี

8. Environmental status (temperature, humidity, etc.) by LINE bot - ขออะไรก็ได้ หมอดู

- ฝัง hardware เขียน code ให้ค่าจาก sensor ได้แก่ อุณหภูมิ ความชื้น และความดัน จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดักจับค่าจาก sensors ดังกล่าว โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฝัง LINE เขียนโค้ดเพื่อดึงค่า sensor มาแจ้งเตือนนักท่องเที่ยวผ่าน LINE chat bot
- ฝัง data analytic นำค่าจาก database มาวิเคราะห์สถิติแล้วทำนายค่าอุณหภูมิ ความชื้น และความดันในอนาคตโดยใช้ Tensorflow

9. Falling detection

- ฝัง hardware เขียน code ให้ gyroscope จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดักจับค่าจำนวนคนที่ล้มลง โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ฝัง data analytic เขียนโค้ดเพื่อดึงค่ามาแจ้งเตือนเจ้าหน้าที่รักษาความปลอดภัยผ่าน LINE Official แล้วนำค่าจาก database มาวิเคราะห์สถิตินักท่องเที่ยวที่บาดเจ็บในแต่ละช่วงเวลา ณ แต่ละสถานที่โดยใช้ Tensorflow

10. Tourism counter limiter (prevent overload)

- ฝัง hardware เขียน code ให้ proximity sensor จากบอร์ด STM32 node for iot ส่งไปให้บอร์ด STM32 Lora พร้อมส่งค่าไปที่ server (Cayenne, NodeJS)
- เขียน api สำหรับ server เพื่อดักจับค่าจำนวนคนที่นับได้ โดยใช้ post method แล้วเก็บค่าเข้า database แบบ nosql
- ถ้าจำนวนคนเยอะเกินกำหนด ฝัง hardware เขียนโค้ดเพื่อให้ส่ง digital output ออกมาเตือนว่าจำนวนคนถึงกำหนด กรุณารอสักครู่เพื่อเยี่ยมชม
- ฝัง data analytic เขียนโค้ดเพื่อดึงค่าจาก database แล้วเอามา predict ว่าวันไหนนักท่องเที่ยวมาเยี่ยมชมจุดใดเป็นพิเศษ เพื่อเตรียมพร้อมการรองรับนักท่องเที่ยว