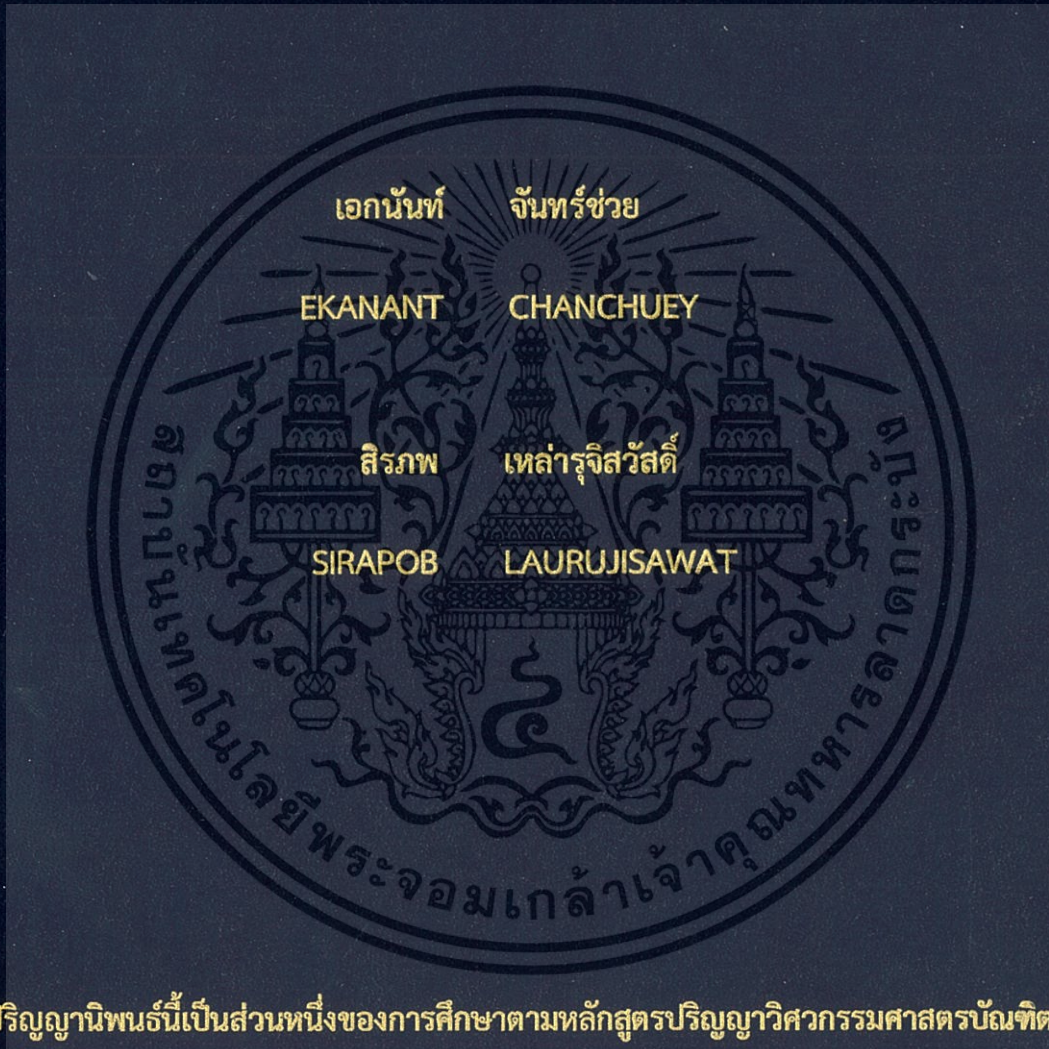


แอปพลิเคชันจัดการการแข่งขันทัวร์นาเมนต์ไฟบริดจ์
Bridge Tournament Management Application



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

แอปพลิเคชันจัดการการแข่งขันทัวร์นาเมนต์ไฟบริดจ์

Bridge Tournament Management Application



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bridge Tournament Management Application

EKANANT CHANCHUEY

SIRAPOB LAURUJISAWAT

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาบัตร

แอปพลิเคชันจัดการการแข่งขันทัวร์นาเมนต์ไฟบริดจ์

รายชื่อนักศึกษา

นายเอกนันท์ จันทรช่วย รหัสนักศึกษา 58011467

นายสิรภพ เหล่ารุจิสวัสดิ์ รหัสนักศึกษา 58011310

ระดับปริญญา

วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมสารสนเทศ

พ.ศ.

2561

อาจารย์ที่ปรึกษาปริญญาบัตร อ.นิจจารีย์ สัตยารักษ์

ปริญญาบัตรฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง



(อ.นิจจารีย์ สัตยารักษ์)

อาจารย์ผู้ควบคุมปริญญาบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์	แอปพลิเคชันจัดการการแข่งขันทัวร์นาเมนต์ไฟเบรตจ์	
รายชื่อนักศึกษา	นายเอกนันท์ จันทร์ช่วย	รหัสนักศึกษา 58011467
	นายสิรภพ เหล่าจุสวัสต์	รหัสนักศึกษา 58011425
ระดับปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	2561	
อาจารย์ที่ปรึกษาปริญญาานิพนธ์	อ.นิจจารีย์ สัตยารักษ์	

บทคัดย่อ

ในปัจจุบันเพื่อการจัดการแข่งขันบริตจ์อย่างมีประสิทธิภาพจำเป็นต้องใช้อุปกรณ์บันทึกผลการเล่นบริตจ์(บริตจ์เมท)ที่มีราคาแพง ต้องใช้เครื่องเป็นจำนวนมากในการแข่งขันแต่ละครั้ง ดังนั้นโครงงานนี้จึงแก้ปัญหาด้วยการพัฒนาแอปพลิเคชันที่มีความสามารถเทียบเท่ากับบริตจ์เมท เพื่อนำไปใช้ทดแทนในการแข่งขันจริง เพื่อเป็นการลดต้นทุนในการจัดการแข่งขันแต่ยังคงประสิทธิภาพ สามารถใช้งานได้ง่ายในแอปพลิเคชันเดียว ใช้งานผ่านมือถือและคอมพิวเตอร์ได้ โดยไม่มีปัญหาในด้านการแสดงผล สามารถบอกได้ว่าใครเจอกับใคร รอบไหน โຕะที่เท่าไร ใช้บอร์ดไฟชุดไหน สามารถบอกผลประมวลของการประมวลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Bridge Tournament Management Application	
Student	Mr. Ekanant Chanchuey	Student ID. 58011467
	Mr. Sirapob Laurujisawat	Student ID. 58011310
Degree	Bachelor of Engineering	
Program	Information Engineering	
Department	Computer Engineering	
Academic Year	2018	
Thesis Advisor	Nitjaree Satayarak	

ABSTRACT

Recently, to manage competition efficiently, organizer must use expensive bridge equipment's (Bridge Mate) in large volume. This project aims to solve the problem so develop application that equivalent to the efficiency of Bridge Mate. As the result, we can apply this application in tournament, include show table of team, player, board and movement, also record final score, compute result and compare result with other team.

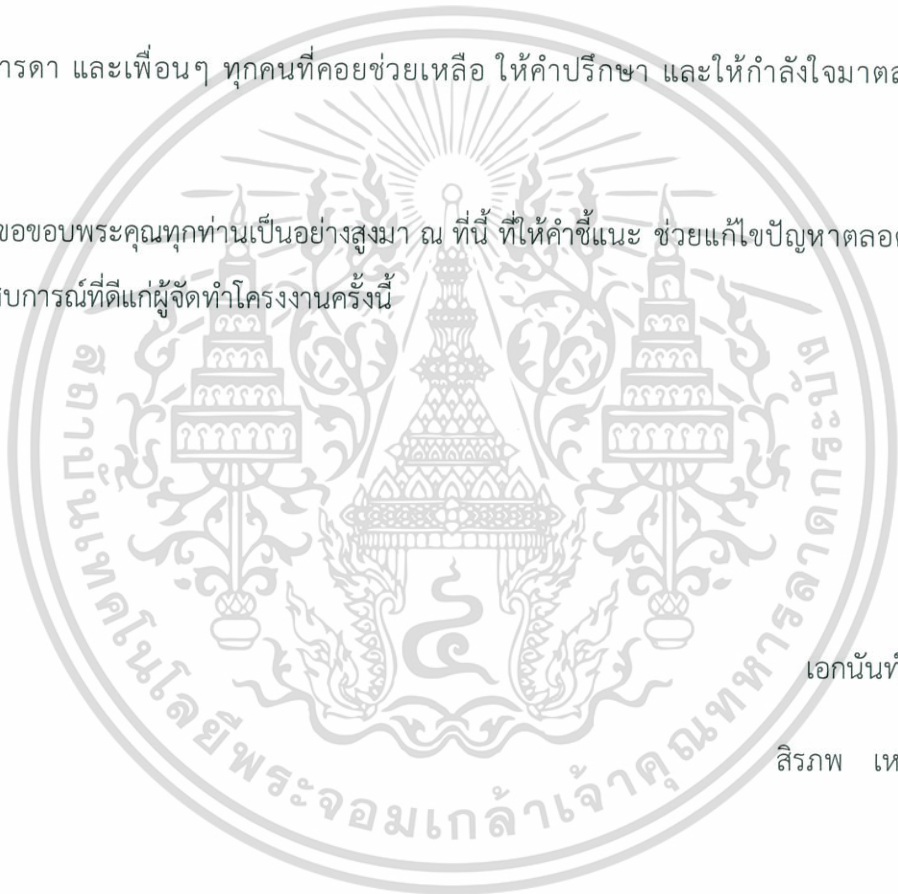
กิตติกรรมประกาศ

การดำเนินโครงการนี้ จะไม่สามารถสำเร็จลุล่วงไปด้วยดี หากขาดการสนับสนุนจากหลายๆฝ่าย ได้แก่

อาจารย์ที่ปรึกษา อ.นิจารีย์ สัตยารักษ์ ที่เป็นผู้ให้คำปรึกษาและแนะแนวทางในการจัดทำโครงการ อีกทั้งยังช่วยปรับปรุงและนำเสนอวิธีการแก้ไขปัญหาต่างๆ ของโครงการที่ทำออกมาเพื่อให้มีประสิทธิภาพมากขึ้น ผู้จัดทำขอขอบพระคุณเป็นอย่างสูง ไว้ ณ โอกาสนี้

บิดา มารดา และเพื่อนๆ ทุกคนที่คอยช่วยเหลือ ให้คำปรึกษา และให้กำลังใจมาตลอดการทำโครงการ

ผู้จัดทำขอขอบพระคุณทุกท่านเป็นอย่างสูงมา ณ ที่นี้ ที่ให้คำชี้แนะ ช่วยแก้ไขปัญหาตลอดจนการให้ความรู้และประสบการณ์ที่ดีแก่ผู้จัดทำโครงการครั้งนี้



เอกนันท์ จันทร์ช่วย

สิรภาพ เหล่ารุจิสวัสดิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 แอพพลิเคชันจัดการการแข่งขันทัวร์นาเมนต์ไฟฟ์บริดจ์.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ภาพรวม หรือโครงสร้างรวมของโครงการ.....	1
1.2.1 ระบบการลงทะเบียนแข่งขัน และระบบจับคู่การแข่งขัน.....	1
1.2.2 ระบบการส่งผลการประมวลผลและผลการแข่งขัน.....	2
1.3 วัตถุประสงค์ของการจัดสร้างโครงการ.....	3
1.4 ขอบเขตของโครงการ.....	3
1.5 ขั้นตอนการดำเนินงานโครงการ.....	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.7 ตารางการดำเนินการโครงการ.....	4
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้.....	5
2.1 ทฤษฎีเกี่ยวกับกีฬา บริดจ์(Bridge).....	5
2.1.1 ความหมายของบริดจ์.....	5
2.1.2 โครงสร้างของกีฬาบริดจ์.....	6
2.1.3 นิยามศัพท์.....	7
2.1.4 แนวคิดในการเล่นกีฬาบริดจ์.....	10
2.1.5 เรื่องของการคิดคะแนน.....	11
2.1.5.1 คะแนนของผู้เดินไพ่.....	11
2.1.5.2 คะแนนของฝ่ายป้องกัน.....	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.1.5.3	คะแนนการแข่งขันกีฬาบริดจ์.....	12
2.2	ทฤษฎีเกี่ยวกับ Firebase platform.....	12
2.2.1	Firebase.....	12
2.2.2	บริการแต่ละส่วนของ Firebase.....	13
2.2.2.1	Build better apps.....	13
2.2.2.2	Improve app quality.....	14
2.2.2.3	Grow your business.....	14
2.3	ทฤษฎีเกี่ยวกับ Web Application.....	15
2.4	ทฤษฎีเกี่ยวกับ JavaScript.....	15
2.5	ทฤษฎีเกี่ยวกับ Node.js.....	16
บทที่ 3	การออกแบบ	18
3.1	ยูสเคสไดอะแกรม(Use case diagram).....	18
3.1.1	ยูสเคสไดอะแกรม ระบบการลงทะเบียนและการจับคู่.....	18
3.1.2	ยูสเคสไดอะแกรม ระบบการส่งคะแนน	19
3.1.3	ยูสเคสไดอะแกรม ระบบการแสดงผล	20
3.2	การออกแบบการทำงานของระบบ.....	21
3.2.1	ระบบการลงทะเบียนการแข่งขัน และระบบจับคู่การแข่งขัน.....	21
3.2.2	ระบบการส่งผลการแข่งขัน	22
บทที่ 4	ผลการทดลอง.....	23
4.1	ระบบการลงทะเบียนการแข่งขัน และระบบจับคู่การแข่งขัน	23
4.2	ระบบการส่งผลการประมวล และผลการแข่งขัน.....	24
4.3	การส่งข้อมูลจาก Firebase ไปยัง Googlesheet.....	28
บทที่ 5	สรุปผลการทดลอง.....	30
5.1	สรุปผลการทดลอง.....	30
5.2	ปัญหาที่พบระหว่างการทดลอง.....	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5.3 แนวทางการปรับปรุง และพัฒนาต่อยอดในอนาคต	31
บรรณานุกรม	32
ภาคผนวก	34
ภาคผนวก ก โปสเตอร์	35
ภาคผนวก ข การติดตั้งโปรแกรมสำหรับการใช้งาน.....	36
ภาคผนวก ค คำอธิบายส่วนประกอบของโปรแกรม	42
ภาคผนวก ง รูปภาพวันโปรเจคต์	52



สารบัญตาราง

หน้า

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน.....	4
---------------------------------------	---



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 ภาพจำลองการจับคู่ด้วยระบบ “พบกันหมด” (Round Robin).....	1
รูปที่ 1.2 อุปกรณ์ “Bridgmate”	2
รูปที่ 2.1 การจำลองเล่น “บริดจ์”	5
รูปที่ 2.2 โครงสร้างตำแหน่งที่นั่งในการแข่งขันของกีฬา “บริดจ์”	6
รูปที่ 2.3 คำเรียกหน้าไพ่ สัญลักษณ์ และอักษรย่อของทรมพ์.....	7
รูปที่ 2.4 โลโก้บริการ Firebase.....	13
รูปที่ 2.5 Javascript.....	15
รูปที่ 2.6 Node.js.....	16
รูปที่ 3.1 ยูสเคสไดอะแกรม ระบบการลงทะเบียนและการจับคู่.....	18
รูปที่ 3.2 ยูสเคสไดอะแกรม ระบบการส่งคะแนน.....	19
รูปที่ 3.3 ยูสเคสไดอะแกรม ระบบการแสดงผล.....	20
รูปที่ 3.4 การออกแบบการทำงานของระบบการลงทะเบียนการแข่งขันและระบบจับคู่การแข่งขัน.....	21
รูปที่ 3.5 การออกแบบการทำงานของระบบส่งผลการแข่งขัน.....	22
รูปที่ 4.1 ภาพหน้าจอการจับคู่การแข่งขัน และการบันทึกลงใน Cloud Firebase.....	23
รูปที่ 4.2 ภาพหน้าจอแสดงขั้นตอนส่งผลการแข่งขัน.....	24
รูปที่ 4.3 ภาพหน้าจอแจ้งเตือนหมายเลขโต๊ะ (จากภาพอยู่ที่โต๊ะหมายเลข1).....	24
รูปที่ 4.4 ภาพหน้าจอการเลือกผลการประมวล.....	25
รูปที่ 4.5 ภาพหน้าจอการแจ้งเตือนเพื่อตรวจสอบผลการประมวล.....	25
รูปที่ 4.6 ภาพหน้าจอการแจ้งเตือนว่ามีการยกเลิกการส่งข้อมูล.....	26
รูปที่ 4.7 ภาพหน้าจอการแจ้งเตือนว่าการส่งข้อมูลนั้นสำเร็จเรียบร้อยแล้ว.....	26
รูปที่ 4.8 ภาพหน้าจอการเลือกผลการแข่งขัน.....	27
รูปที่ 4.9 ภาพหน้าจอการแจ้งเตือนเพื่อตรวจสอบผลการแข่งขัน.....	27
รูปที่ 4.10 ภาพการเก็บข้อมูลภายใน Firebase Database ของโต๊ะหมายเลข1.....	28
รูปที่ 4.11 ภาพการเชื่อมต่อข้อมูลจาก Firebase ไปยัง Google Sheets.....	29
รูปที่ ก.1 รูปภาพโปสเตอร์.....	35

สารบัญรูป (ต่อ)

หน้า

รูปที่ ข.1 หน้าต่างดาวนโหลดโปรแกรมจาก Website	36
รูปที่ ข.2 เป็นการรอดาวนโหลด โปรแกรม Visual Studio Code	36
รูปที่ ข.3 ขั้นตอนการติดตั้งโปรแกรม	37
รูปที่ ข.4 ขั้นตอนการติดตั้งโปรแกรม	37
รูปที่ ข.5 ขั้นตอนการติดตั้งโปรแกรม	38
รูปที่ ข.6 ขั้นตอนการติดตั้งโปรแกรม	38
รูปที่ ข.7 ขั้นตอนการติดตั้งโปรแกรม	39
รูปที่ ข.8 ขั้นตอนการติดตั้งโปรแกรม	39
รูปที่ ข.9 ขั้นตอนการติดตั้งโปรแกรม	40
รูปที่ ข.10 ติดตั้งโปรแกรมเสร็จสมบูรณ์	40
รูปที่ ข.11 หน้าต่างโปรแกรม	41
รูปที่ ข.12 หน้าต่างดาวนโหลดโปรแกรมจาก Website	41
รูปที่ ข.13 ขั้นตอนการติดตั้งโปรแกรม	42
รูปที่ ข.14 ขั้นตอนการตรวจสอบว่าโปรแกรมได้ทำการติดตั้งเรียบร้อยแล้ว	42
รูปที่ ค.1 ภาพในการโหลดไฟล์ Servicesaccountkey.json เพื่อใช้ในการเชื่อมต่อ	43
รูปที่ ค.2 ตัวอย่างข้อมูลที่อยู่ภายในไฟล์ Servicesaccountkey.json	44
รูปที่ ค.3 การเปิดใช้งาน Command Prompt	44
รูปที่ ค.4 การชี้ Directory ของไฟล์ที่ใช้ในการเปิด	45
รูปที่ ค.5 การบอกถึงชื่อไฟล์และชื่อ port ที่ทำการใช้งาน	45
รูปที่ ค.6 การเปิดใช้งาน node.js	46
รูปที่ ค.7 การเข้าใช้งานผ่านอุปกรณ์	46
รูปที่ ค.8 Command prompt แสดงการเชื่อมต่อของอุปกรณ์	47
รูปที่ ค.9 การตั้งค่า IP ของอุปกรณ์ทำให้สามารถใช้งานระบบได้	47
รูปที่ ค.10 Config Firebase	48
รูปที่ ค.11 เก็บข้อมูลที่ Input เข้าฐานข้อมูล	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ IX การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ ค.12 Cloud Firestone.....	49
รูปที่ ค.13 Source code delete data.....	49
รูปที่ ค.14 Interface delete data	50
รูปที่ ค.15 เก็บข้อมูลตารางการแข่งขัน.....	50
รูปที่ ค.16 เก็บข้อมูลตารางการแข่งขัน(ต่อ).....	51
รูปที่ ค.17 Interface ตารางการแข่งขัน.....	51
รูปที่ ง.1 วันโปรเจคเตย์	52
รูปที่ ง.2 วันโปรเจคเตย์	53
รูปที่ ง.3 วันโปรเจคเตย์	53
รูปที่ ง.4 วันโปรเจคเตย์	53
รูปที่ ง.5 วันโปรเจคเตย์	54



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

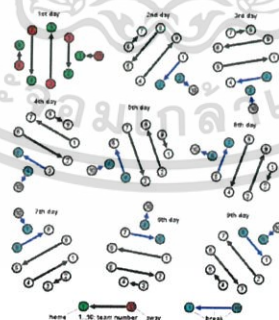
1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันมีการแข่งขันกีฬา “บรีดจ์” อย่างเป็นทางการอยู่อย่างต่อเนื่อง ซึ่งในการแข่งขันนั้นจะมีการจัดบันทึกผลการประมูลและผลการแข่งขัน เพื่อที่จะนำมาคิดผลการแข่งขันอยู่ โดยปกติแล้วจะใช้เครื่องมือเฉพาะในการบันทึกผลการประมูลและผลการแข่งขัน ซึ่งจะมีราคาค่อนข้างสูง และด้วยเทคโนโลยีที่ผ่านเข้ามาทำให้ระบบการทำงานไร้สายนั้นมีเสถียรมากยิ่งขึ้น ทางผู้จัดทำจึงเล็งเห็นว่าทำให้การแข่งขันมีการส่งผลการแข่งขันทางอินเทอร์เน็ตนั้นทำให้มีความรวดเร็วและสามารถดำเนินกิจกรรมต่อไปได้อย่างราบรื่นด้วยเช่นกัน และไม่ต้องสิ้นเปลืองค่าอุปกรณ์บันทึกผลการเล่นบรีดจ์

1.2 ภาพรวม หรือโครงสร้างรวมของโครงการ

ภาพรวมของโครงการจะแบ่งออกเป็น 2 ส่วนนั่นก็คือ

1.2.1 ระบบการลงทะเบียนการแข่งขัน และ ระบบจับคู่การแข่งขัน



รูปที่ 1.1 ภาพจำลองการจับคู่ด้วยระบบ “พบกันหมด” (Round Robin)

(อ้างอิงจาก https://en.wikipedia.org/wiki/Round-robin_tournament)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นระบบลงทะเบียนการแข่งขันและจะนำรายชื่อไปทำการจับคู่โดยใช้การจับคู่แบบ “พบกันหมด” (Round Robin) ที่ได้รับการยอมรับในการแข่งขันมาเป็นเกณฑ์ในการจับคู่ซึ่งผู้เล่นสามารถดูตำแหน่งที่นั่งของตนเองได้ทันทีหลังจากที่กดผลลัพธ์แสดงผลออกมา และสามารถเปิดดู รายละเอียดของจับคู่, ตำแหน่งที่นั่ง (หมายเลขโต๊ะ), ชุดไฟ, ฝั่งที่นั่ง และ รอบที่แข่งได้

1.2.2 ระบบการส่งผลการประมูลและผลการแข่งขัน



รูปที่ 1.2 อุปกรณ์ “Bridgemate”

(อ้างอิงจาก <http://www.bridgemate.com/product/bridgemate-pro/>)

เป็นระบบการส่งผลการแข่งขันโดยที่ผู้เล่นจะสามารถส่งผลการแข่งขันได้โดยที่ใช้โทรศัพท์มือถือ Smart phone ที่ทำการเชื่อมต่อแล้วในการส่งข้อมูลในการแข่งขันแทนการใช้อุปกรณ์ “Bridgemate” ในการส่งข้อมูลซึ่งข้อมูลที่ส่งไปนั้นจะไปเก็บไว้ในส่วนของ Firebase Database และจะทำการเชื่อมต่อข้อมูลจากทาง Firebase Database ส่งไปยัง Google-Sheet ที่ทำการลงทะเบียนไว้ โดยผ่านทาง Firebase Function ซึ่งจะนำข้อมูลที่ได้นั้นไปทำการประมวลผลต่อไปได้ โดยการเชื่อมต่อจะเป็นการเชื่อมต่อโทรศัพท์มือถือ Smart phone และเครื่องคอมพิวเตอร์ผ่านทาง socket.io ใน node.js ซึ่งสามารถส่งข้อมูลหากันและสั่งให้บันทึกผลลงใน Firebase ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วัตถุประสงค์ของการจัดสร้างโครงการ

- 1) เพื่อลดต้นทุนการใช้เครื่องมือเฉพาะที่มีราคาค่อนข้างสูง
- 2) เพื่อให้ผู้ใช้งาน ใช้งานได้ง่ายขึ้น
- 3) เก็บข้อมูลได้ถาวร แก้ไขได้ตลอดเวลา

1.4 ขอบเขตของโครงการ

- 1) การศึกษาในครั้งนี้ มุ่งเน้นไปที่ การเก็บคะแนนของการเล่นเกมสโฟต์แวร์ ทั้งในประเทศและต่างประเทศ ในรูปแบบต่างๆ
- 2) สามารถเข้าสู่ระบบทางเว็บไซต์ได้ทั้งระบบคอมพิวเตอร์และระบบโทรศัพท์ Smart Phone โดยต้องมีการเชื่อมต่ออินเทอร์เน็ต เพื่อรับและส่งข้อมูลได้
- 3) ใช้งานได้เทียบเท่ากับการใช้งานผ่านอุปกรณ์บันทึกผลการเล่นบริดจ์
- 4) ระยะเวลาที่ใช้ในการศึกษา ครั้งนี้ ดำเนินการในภาคเรียนที่ 1-2 ตั้งแต่วันที่ 15 สิงหาคม พ.ศ.2561 – 20 เมษายน พ.ศ. 2562

1.5 ขั้นตอนการดำเนินโครงการ

- 1) ศึกษารายละเอียดโครงการ
- 2) ศึกษาวิธีเล่นและวิธีคิดผลการแข่งขัน “บริดจ์”
- 3) ศึกษาวิธีที่จะเก็บข้อมูลผลการแข่งและหาวิธีที่จะรับและส่งข้อมูลได้
- 4) เก็บรวบรวมข้อมูลที่ศึกษา
- 5) ออกแบบระบบการทำงาน
- 6) สร้างระบบเก็บข้อมูลที่ต้องใช้ในการทำโครงการ
- 7) เริ่มต้นเขียนโปรแกรมการทำงาน
- 8) ทดสอบและแก้ไขข้อผิดพลาด
- 9) วิเคราะห์ข้อมูลของระบบ
- 10) จัดทำรูปเล่ม

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เพื่อลดต้นทุนการนำเข้าอุปกรณ์เฉพาะ ซึ่งมีราคาค่อนข้างสูง ประหยัดค่าใช้จ่าย
- 2) เพื่ออำนวยความสะดวกและลดระยะเวลาการแข่งขัน
- 3) จัดเก็บข้อมูลได้เป็นจำนวนมากใน database
- 4) เพื่อให้ ผู้ใช้งาน ใช้งานได้ง่าย
- 5) กลุ่มเป้าหมาย ใช้งานได้ในวงกว้าง

1.7 ตารางการดำเนินการโครงการ

No.	ขั้นตอนการทำงาน (Task)	สิงหาคม 2018				กันยายน 2018				ตุลาคม 2018				พฤศจิกายน 2018			
		Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4
1	ศึกษาการเล่นและการคิดคะแนน บอร์ด																
2	ศึกษาค้นคว้าวิธีการใช้งาน Firebase																
3	ศึกษาค้นคว้าการใช้งาน HTML และ Nodejs																
4	ออกแบบระบบการทำงาน																
5	เชื่อมต่อ ทดลองการใช้ Firebase database																
6	นำ Firebase Database เชื่อมต่อกับ Google Sheet																
7	ทดสอบและแก้ไขข้อมูล																
8	จัดทำเอกสารประกอบการนำเสนอ																

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้

2.1 การแข่งขันกีฬา บริดจ์ (Bridge)



รูปที่ 2.1 ภาพจำลองการเล่น “บริดจ์”

(อ้างอิงจาก <http://phys.org/news/2015-04-bridge-sport-brain-isnt-muscle.html>)

2.1.1 ความหมายของ “บริดจ์”

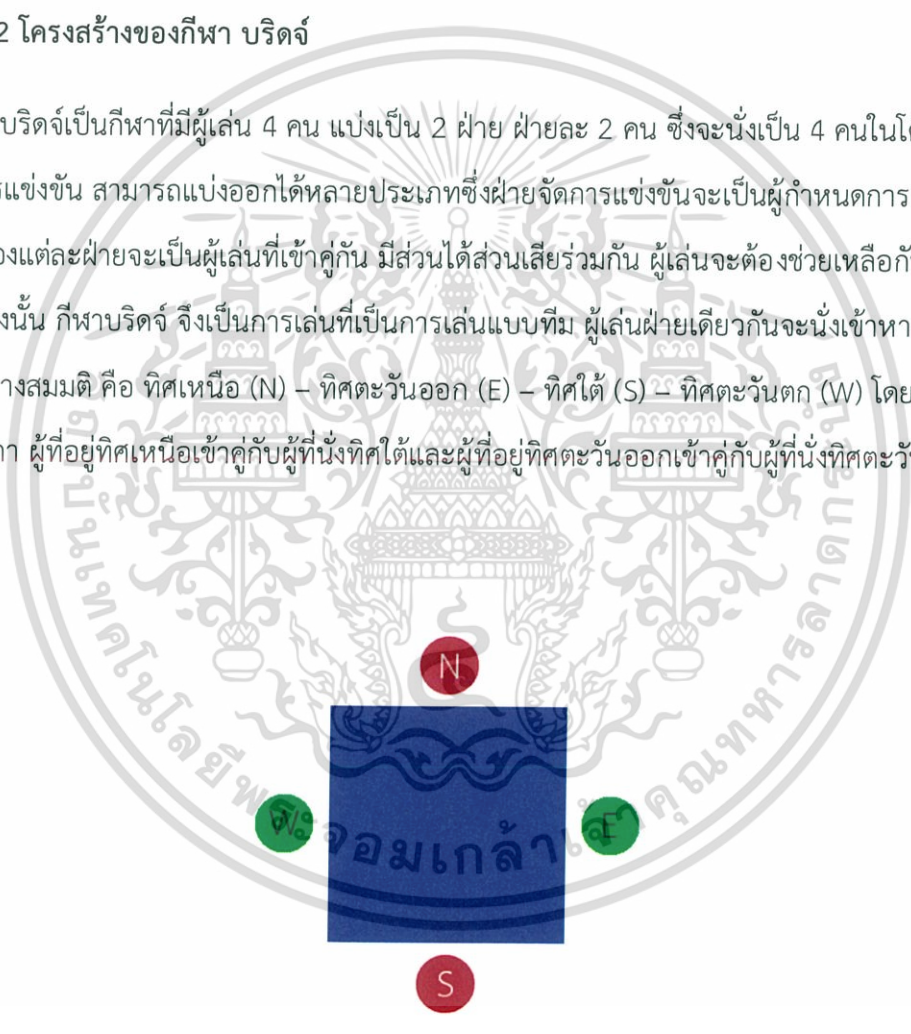
‘บริดจ์’ มาจากคำภาษาอังกฤษ คือ ‘Bridge’ แปลว่า “สะพาน” เป็นชื่อที่สอดคล้องกับวิธีการเล่น ซึ่งจะต้องใช้การเชื่อมโยงของไพ่ของแต่ละมือ เหมือนกับการเชื่อมโยงของสะพาน นั่นคือ เมื่อมีการลงไพ่ใบแรกแล้ว มือใดกินไพ่นั้นก็ต้องออกไพ่ถัดไป และลงไพ่เช่นนี้ไปเรื่อยๆ จนหมดจำนวนไพ่ที่ถือในมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กีฬาบรีดจ์เป็นกีฬาสากล มีการเล่นอย่างแพร่หลายในทุกที่บนโลก และการแข่งขันกีฬาบรีดจ์ก็มีหลายระดับ ตั้งแต่ระดับนักเรียนนักศึกษา เรียกว่า “บรีดจ์อุดมศึกษา” และ บรีดจ์ระดับมหาวิทยาลัย ระดับสโมสร ระดับท้องถิ่น ระดับกระทรวง และ ระดับชาติ โดยเป็นกีฬาชนิดหนึ่งซึ่งจัดการแข่งขันในกีฬาแห่งชาติ ไปจนถึงการแข่งขันระดับนานาชาติ การเล่นเกมกีฬาบรีดจ์มีวัตถุประสงค์หลายอย่าง เช่น เล่นเพื่อการพักผ่อน เล่นเพื่อการผ่อนคลาย เล่นเพื่อความสนุกสนาน เล่นเพื่อฝึกฝนการใช้เหตุผลและการวิเคราะห์ เล่นเพื่อพัฒนาสมอง และ ป้องกันไม่ให้สมองฝ่อ กีฬาบรีดจ์มีประโยชน์แก่สมองมาก เพราะเป็นการใช้สมองคิด เพื่อวิเคราะห์อย่างมีเหตุผลในทุกขั้นตอนของการเล่น

2.1.2 โครงสร้างของกีฬา บรีดจ์

กีฬาบรีดจ์เป็นกีฬาที่มีผู้เล่น 4 คน แบ่งเป็น 2 ฝ่าย ฝ่ายละ 2 คน ซึ่งจะนั่งเป็น 4 คนในโต๊ะเดียวกัน แต่ถ้าเป็นการแข่งขัน สามารถแบ่งออกได้หลายประเภทซึ่งฝ่ายจัดการแข่งขันจะเป็นผู้กำหนดการแข่งขัน ผู้เล่น 2 คน ของแต่ละฝ่ายจะเป็นผู้เล่นที่เข้าคู่กัน มีส่วนได้ส่วนเสียร่วมกัน ผู้เล่นจะต้องช่วยเหลือกัน ไม่ใช่ต่างคนต่างเล่น ดังนั้น กีฬาบรีดจ์ จึงเป็นการเล่นที่เป็นการเล่นแบบทีม ผู้เล่นฝ่ายเดียวกันจะนั่งเข้าหากัน นั่งในตำแหน่งทิศทางสมมติ คือ ทิศเหนือ (N) – ทิศตะวันออก (E) – ทิศใต้ (S) – ทิศตะวันตก (W) โดยที่จะนั่งวนตามเข็มนาฬิกา ผู้ที่อยู่ทิศเหนือเข้าคู่กับผู้ที่นั่งทิศใต้และผู้ที่อยู่ทิศตะวันออกเข้าคู่กับผู้ที่นั่งทิศตะวันตก



รูปที่ 2.2 โครงสร้างตำแหน่งที่นั่งในการแข่งขันของกีฬา “บรีดจ์”

2.1.3 นิยามศัพท์

1. **ทรัมป์ (Trump)** หมายถึง ชุดไพ่ซึ่งมาจากการประมูลกำหนดให้เป็นไพ่ชุดพิเศษที่เรียกว่า “ชุดทรัมป์” โดยไพ่ชุดนี้มีอำนาจพิเศษคือใช้กินไพ่ใบใหญ่ชุดอื่นได้เมื่อในมือไม่มีไพ่ชุดนั้นแล้วและเขียนประมูลด้วยอักษรย่อคือ "C-D-H-S 'แทนคำ' Club (จิก) - Diamond (เหลี่ยม) - Heart (โพแดง) - โพดำ (Spade) ซึ่งย่อจากอักษรตัวหน้าของแต่ละคำ

♣ Club (จิก)	อักษรย่อ C
♦ Diamond (เหลี่ยม)	อักษรย่อ D
♥ Heart (โพแดง)	อักษรย่อ H
♠ Spade (โพดำ)	อักษรย่อ S

รูปที่ 2.3 คำเรียกหน้าไพ่ สัญลักษณ์ และ อักษรย่อ ของทรัมป์

2. **โนทรัมป์ (No trump)** หมายถึงการประมูลซึ่งไม่กำหนดให้มีไพ่ชุดทรัมป์จึงไม่มีไพ่ชุดพิเศษที่มีอำนาจพิเศษการประมูลด้วยการเขียนอักษรย่อคือ “NT” สำหรับการประมูล "NT" ก็จะมีอยู่ในลำดับสูงกว่าไพ่ชุดโพดำการประมูลเรียงจากลำดับต่ำไปถึงลำดับสูงได้ดังนี้คือ "1จิก - 1เหลี่ยม - 1โพแดง - 1โพดำ - 1โนทรัมป์" (1C-1D-1H-1S-1NT) และใช้ตัวเลขกำกับเมื่อขานเพิ่มระดับสูงขึ้นคือ 2-3-4-5-6-7 เช่น 1C-2D ไปตามลำดับจนถึงระดับ 7C ซึ่งเป็นระดับสุดท้ายของไพ่ชุดจิก

3. **ดับเบิล (Double)** มีความหมาย 2 ประการ คือ หมายถึงการขานดับเบิลบังคับให้คู่ขานไปอย่างหนึ่งและอีกอย่างหนึ่งหมายถึงการลงโทษกรณีหลังนี้จะได้หรือเสียคะแนนคิดเป็นสองเท่าของคะแนนปกติ

4. **รีดับเบิล (Redouble)** มีความหมาย 2 ประการเช่นเดียวกัน คือ หมายถึงการบอกคู่ว่าถือไพ่มากเมื่ออีกฝ่ายดับเบิลอย่างหนึ่งไพ่มากหมายถึงแต้มมากและอีกอย่างหนึ่งหมายถึงลงโทษฝ่ายที่ขานลงโทษกรณีหลังนี้จะได้หรือเสียคะแนนคิดเป็นสองเท่าของคะแนนดับเบิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ผ่าน (pass) หมายถึงการชานเมื่อไม่ประสงค์จะประมูล ส่วนการเขียนคือทำเครื่องหมาย "/" ลงในกระดาษเขียนประมูล

6. ประมูลเกม หมายถึงการประมูลถึงระดับเกมและกินไฟได้จำนวนต้องครบตามที่ประมูลได้กรณี "เกม" นี้ใช้เกณฑ์ 100 คะแนนคิดจากการกินทองที่ประมูลได้คือใช้ระดับการประมูลบวกกับฐาน 6 ตัวอย่างเช่นประมูลระดับ 2 ต้องกินไฟให้ได้ 8 ทอง (6 + 2) เป็นต้น

7. สวมอลสะแลม (Small slam) หมายถึงการประมูลถึงระดับ 6 และกินไฟได้ 12 ทอง

8. แกรนด์สะแลม (Grand slam) หมายถึงการประมูลถึงระดับ 7 และกินไฟได้ถึง 13 ทอง

9. ประมูลไม่ถึงเกม หมายถึงการประมูลไม่ถึงระดับเกมและกินไฟไม่ได้จำนวนต้องครบตามที่ประมูล กรณีนี้แม้จะกินไฟได้ทองเกินก็ไม่ถือว่าเกมส่วนคะแนนได้จากจำนวนทองที่กินได้เพิ่มขึ้น

10. คะแนน หมายถึงการคิดคำนวณจากจำนวนทองที่กินได้โดยนับหนึ่งตั้งแต่ทองที่เจ็ดเป็นต้นไปและถ้ากินไม่ได้ต้องครบและเกินตามที่ประมูลก็ได้คะแนนแต่ถ้ากินไฟไม่ครบจำนวนตามที่ประมูลก็จะเสียคะแนนและในการแข่งขันนั้นจะมีตารางคะแนนวางไว้บนโต๊ะผู้เล่นไม่จำเป็นต้องคิดคำนวณเองเพียงแต่จดจำนวนทองที่กินได้หรือไม่ได้ก็พอแล้วตรวจดูจากตารางคะแนนที่ใช้เป็นคู่มือสำหรับการคิดคะแนน

ส่วนการคิดคะแนนการเล่นได้หรือเสียจะต้องเป็นไปตามที่ประมูลซึ่งมีรายละเอียดใช้ในการคิดคำนวณดังต่อไปนี้

(1) สัญญาในทริมป์ทองแรก (นับทองที่เจ็ด) ได้ 40 คะแนนทองถัดไปทองละ 30 คะแนนประมูลเกมคือระดับ 3 และต้องกินไฟได้ 9 ทองเท่ากับ 100 คะแนน (40 + 30 + 30)

(2) สัญญาชุด Major เป็นทริมป์คิดคำนวณทองละ 30 คะแนนประมูลเกมคือระดับ 4 และต้องกินไฟให้ได้ 10 ทองเท่ากับ 120 คะแนน

(3) สัญญาชุด Minor เป็นทริมป์คิดคำนวณทองละ 20 คะแนนประมูลเกมคือระดับ 5 และต้องกินไฟให้ได้ 11 ทองเท่ากับ 100 คะแนน

ส่วนการคิดคะแนนการเล่นไม่ได้หรือเล่นเสียจะต้องขึ้นอยู่กับกรณีที่มีเกมหรือไม่มีเกมและมีรายละเอียดใช้คิดคำนวณดังต่อไปนี้

(1) ไม่มีเกมเล่นตกเสียคะแนนต้องละ 100 คะแนน

(2) ไฟไม่มีเกมเล่นตกเสียคะแนนต้องละคะแนน

(3) กรณีดับเบิ้ลหรือรีดับเบิ้ลและเล่นได้คะแนนเป็นสองเท่าหรือสี่เท่าของคะแนนปกติและค่าปรับแต่ถ้าหากเล่นตกต้องเสียคะแนนเพิ่มเป็นสองเท่าหรือสี่เท่าและเพิ่มเป็นทวีคูณเมื่อเล่นตกเกินหนึ่งตอง

11. ตอง (Trick) คือไฟในมือของแต่ละคนที่วางลงบนโต๊ะคนละใบต่อครั้งครั้งละรวมเป็น 4 ใบ เรียกว่า 1 ตองเมื่อไฟในมือของแต่ละคนซึ่งวางลงบนโต๊ะจนไฟในมือหมดรวม 13 ตอง

12. ผู้แจกไฟ (Dealer) ในที่นี้หมายถึงการแจกไฟจากสำรับไฟตามปกติแต่ถ้าเป็นการแข่งขันเขาจะจัดบอร์ดไฟให้เป็นบอร์ดมิลี่ช่องใช้เสียบไฟทางด้านข้างส่วนด้านบนเขียนทิศสมมติ “N-E-S-W” ตำแหน่ง Dealer ตำแหน่ง “Vul (Vulnerable)” หมายถึงตำแหน่งไม่มีเกม ส่วนตำแหน่งไฟไม่มีเกม (Not Vulnerable) ไม่เขียนไว้ โดยให้ถือว่า เมื่อไม่เขียนก็คือ “ไม่มีเกม”

13. ผู้เดินไฟ (Declarer) คือผู้ประมูลได้และเป็นผู้เดินไฟ

14. ฝ่ายป้องกัน (Defenders) คือฝ่ายตรงข้ามกับผู้เดินไฟ

15. มือตัมมี คือไฟของคู่ของผู้เดินไฟซึ่งวางลงบนโต๊ะ

2.1.4 แนวคิดในการเล่นกีฬาบริดจ์

1. เนื่องจาก “ไฟป็อก” เป็นอุปกรณ์หลักที่ใช้เล่นกีฬาบริดจ์ การรู้จักและเข้าใจไฟทั้ง 52 ใบ มีความสำคัญและเป็นหัวใจหลักของกีฬาชนิดนี้

2. วิธีเข้าถึงแก่นของไฟตามข้อ 1. ก็คือ “คิด” โดยใช้สามัญสำนึกและใช้เหตุผลสนับสนุนสามารถวิเคราะห์ข้อดีข้อเสียตามหลักความน่าจะเป็น อย่างรอบคอบและรอบด้านแล้วนำมาปรับใช้กับการประมูล การเดินไฟและการป้องกันและจะต้องไม่ทำตามอำเภอใจในการประมูล การเดินไฟและการป้องกันโดยจะต้องคิดและวิเคราะห์อย่างเป็นระบบ วิธีคิดก็ไม่ยุ่งยากเพราะเป็นการคิดในกรอบของไฟจำนวน 52 ใบนั่นเอง

3. การจำแนกลำดับไฟ ชุดจิก ชุดเหลี่ยม ชุดโพแดง ชุดโพดำ เป็นการเรียงไฟจากลำดับต่ำสุดไปยังไฟลำดับสูงสุด เพื่อใช้ประมูลเรียงตามลำดับ ถ้าจะประมูลไฟลำดับต่ำภายหลังไฟลำดับสูงทำได้ด้วยการเพิ่มระดับ

เช่นมีผู้ประมูล 1 เหลี่ยมแล้วหากจะประมูลชุดจิกซึ่งลำดับต่ำกว่า ก็ต้องขานเพิ่มระดับคือ 2 จิก (1 เหลี่ยม – 2 จิก) ถ้าประมูลไพ่ตามลำดับปกติก็ขานประมูลได้เลยไม่ต้องเพิ่มระดับ (1 จิก - 1 เหลี่ยม)

4. การรู้จักอักษรย่อ “C-D-H-S” เพื่อนำไปใช้เขียนประมูล

5. การจำแนกไพ่ใบใหญ่สุดลงมาถึงไพ่ใบเล็กสุดตามลำดับก็คือ A-K-2-3-10-9-8-7-6-5-4-3-2 เพื่อนำไปใช้ในการเดินไพ่การป้องกันการให้สัญญาณ (Signal) แบบต่าง ๆ และการลงไพ่ก็ต้องลงตรงตามชุดวันแต่ไพ่ชุดนั้นไม่มีแล้วจึงจะลงไพ่ชุดอื่นได้ส่วนการกินไพ่นั้นคือไพ่ใบใหญ่กินไพ่ใบเล็กยกเว้นกินด้วยทรัมป์เมื่อไพ่ชุดนั้นไม่มีแล้วเมื่อไพ่ลงครบ 13 ใบการเล่นจบหนึ่งกระดาน

6. ต้องแยกออกได้และรู้ว่าผู้ที่ขานไฟคนแรกคือผู้เปิดประมูล (Opener) หลังจากนั้นถ้าหากคู่ของเขา ขานไฟคู่ของเขาเป็นผู้ตอบ (Responder) เหตุที่ต้องแยกและรู้ว่าใครเป็นผู้เปิดใครเป็นผู้ตอบเพราะมีความแตกต่างระหว่างผู้เปิดกับผู้ตอบเกี่ยวกับแต้มของไฟในมือหมายถึงผู้เปิดประมูลจะต้องมีแต้มเท่าใดจึงเปิดประมูลได้ส่วนผู้ตอบไม่จำเป็นจะต้องมีแต้มเท่ากับผู้เปิดถ้าหากผู้ตอบถือไพ่มีแต้มเข้าเกณฑ์การขานตอบผู้ตอบก็สามารถขานได้

7. เมื่อถือไพ่ต่ำกว่าเกณฑ์ก็ต้องขาน “ผ่าน” อีกฝ่ายซึ่งนั่งถัดไป (ตามเข็มนาฬิกา) ก็จะเปิดประมูลถ้าไม่มีแต้มพอแต่ถ้าแต้มไม่พอก็ต้องขานผ่านเช่นกันคู่ของเราซึ่งนั่งถัดไปก็จะเป็นผู้เปิดประมูลถ้าถือไม่มีแต้มพอ (ดูข้อยกเว้นเรื่องการประมูลชุด) และถ้าถือไพ่แต้มไม่พอก็ต้องขานผ่านใช้วิธีการขานหมุนเวียนแบบนี้เรื่อยไปซึ่งการประมูลสิ้นสุดลงเมื่อการขาน “ผ่าน” ครบทุกคนโดยไม่มีผู้ใดประมูลอีก

2.1.5 เรื่องของการคิดคะแนน

2.1.5.1 คะแนนของผู้เดินไฟ

1. เมื่อจบการประมูลและการเดินไพ่จบลงโดยผู้เดินไฟไม่ได้คะแนนนั้นผู้เดินไฟจะต้องเดินไม่ได้จำนวนตองครบหรือเกินกว่าสัญญาณที่ประมูลถ้าหากประมูลไม่ถึงเกมและทำได้ก็จะได้คะแนนดังนี้

(1) สัญญาทรัมป์ตองแรกได้ 40 คะแนน (โดยนับตองที่เจ็ดเป็นตองแรก) ตองต่อจากนั้นไปตองละ 30 คะแนน

(2) สัญญาชุด Major ได้ตองละ 30 คะแนน

(3) สัญญาชุด Minor ได้ตองละ 20 คะแนน

(4) ทุกสัญญาข้างต้นมีคะแนนเพิ่มอีกต่างหาก 50 คะแนนคะแนนเพิ่มตาม (4) ใช้เฉพาะการเล่นบริดจ์ซึ่งจัดไฟใส่บอร์ดเท่านั้นถ้าหากเป็นการเล่นแบบแจกจากสำหรับไฟที่เรียกว่า "รับเบอร์บริดจ์ Rubber Bridge)" จะมีวิธีคิดคะแนนสำหรับคะแนนเพิ่มต่างหากซึ่งจะกล่าวถึงต่อไป

2. ถ้าหากประมูลถึงเกมและเดินไฟได้ตามสัญญาที่ประมูลได้นอกจากได้คะแนนอย่างใดอย่างหนึ่งในข้อ 1. (1) (2) (3) แล้วยังได้คะแนนเพิ่มอีกแต่ไม่มีคะแนนตามข้อ 1. (4) คะแนนที่ได้เพิ่มมีดังนี้ (1) กรณีไม่มีเกมได้เพิ่มอีก 300 คะแนน (2) กรณีมีเกมได้เพิ่มอีก 500 คะแนน

3. ถ้าหากประมูลถึงระดับสะสมอละสลัมและเดินไฟได้ตามสัญญาที่ประมูลได้ (กินได้ 12 ทอง) นอกจากได้คะแนนเพิ่มตามข้อ 2. แล้วยังได้คะแนนเพิ่มอีกดังนี้ (1) กรณีไม่มีเกมได้เพิ่มอีก 500 คะแนน (2) กรณีมีเกมได้เพิ่มอีก 750 คะแนน

4. ถ้าหากประมูลถึงระดับแกรนด์สะสมและเดินไฟได้ตามสัญญาที่ประมูลได้ (กินได้ 13 ทอง) นอกจากได้คะแนนเพิ่มตามข้อ 2. แล้วยังได้คะแนนเพิ่มอีกดังนี้ (คะแนนข้อ 3. ไม่เกี่ยวข้อง)

(1) กรณีไม่มีเกมได้เพิ่มอีก 1,000 คะแนน

(2) กรณีมีเกมได้เพิ่มอีก 1,500 คะแนน

5. กรณีดับเบิลและผู้เดินไฟทำได้ตามสัญญาผู้เดินไม่ได้คะแนนเพิ่มเป็น 2 เท่าของคะแนนตามข้อ 1. และได้ค่าปรับอีก 50 คะแนน

6. กรณีดับเบิลและผู้เดินไฟทำได้ตามสัญญาผู้เดินไม่ได้คะแนนเพิ่มเป็น 2 เท่าของคะแนนตามข้อ 5. เมื่อฝ่ายเดินไม่ได้คะแนนฝ่ายป้องกันก็เสียคะแนน

2.1.5.2 คะแนนของฝ่ายป้องกัน

การที่ฝ่ายป้องกันจะเป็นฝ่ายได้คะแนนนั้นฝ่ายป้องกันจะต้องเดินไฟป้องกันจนผู้เดินไฟทำไม่ได้ตามสัญญาหรือที่เรียกกันว่าเดินตรายละเอียดการได้คะแนนของฝ่ายป้องกันมีดังนี้

1. กรณีไฟไม่มีเกมผู้เดินไฟเดินไฟตกฝ่ายป้องกันได้คะแนนต่อละ 50 คะแนน

2. กรณีไฟมีเกมผู้เดินไฟเดินไฟตกฝ่ายป้องกันได้คะแนนต่อละ 100 คะแนน

3. กรณีดับเบิลไฟไม่มีเกมผู้เดินไฟเดินตกฝ่ายป้องกันได้คะแนนตองแรก 100 คะแนนตองที่สองกับตองที่สามตองละ 200 คะแนนตั้งแต่ตองที่สี่เป็นต้นไปตองละ 300 คะแนน

4. กรณีดับเบิลไฟมีเกมผู้เดินไฟเดินตกฝ่ายป้องกันได้คะแนนตองแรก 200 คะแนนตั้งแต่ตองที่สองตองละ 300 คะแนน

5. ถ้าหากมีการรีดับเบิลผู้เดินไฟเดินตกฝ่ายป้องกันจะได้คะแนนเป็นสองเท่าของข้อ 3. หรือข้อ 4. แล้วแต่กรณี

เมื่อฝ่ายป้องกันได้คะแนน ฝ่ายเดินไฟก็เสียคะแนน

2.1.5.3 คะแนนการแข่งขันกีฬาบริดจ์

1. การแข่งขันประเภททีมคิดคะแนนเปรียบเทียบการได้คะแนนระหว่างสองทีมนำมาหักกลับกัน
2. การแข่งขันประเภทคู่ใช้คะแนนลำดับสูงต่ำของฝ่าย N-S กับฝ่าย E-W นำมาคิดเป็นคะแนนเปรียบเทียบ

คะแนนการแข่งขันประเภททีมหรือประเภทคู่อีกตาม ผู้เล่นไม่จำเป็นต้องคิดคะแนนเป็น เพราะเป็นภารกิจของกรรมการจัดการแข่งขัน เพียงแต่ผู้เล่นบริดจ์ควรจะต้องรู้บ้างเท่านั้น

2.2 Firebase Platform

2.2.1 Firebase

“Firebase” เป็นบริการ Platform จากทาง Google ซึ่งจะเป็นบริการที่ดูแลในส่วนของระบบหลังบ้านหรือเบื้องหลังการทำงานของระบบ Applications ซึ่งจะคอยช่วยเหลือด้านวิเคราะห์ข้อมูล การเก็บข้อมูล และการให้บริการด้าน Service ของตัว Applications (API) อีกด้วย ซึ่งบริการ Firebase จะรองรับ Applications ได้หลายระบบปฏิบัติการ ตัวอย่างเช่น iOS, Android, Web Application โดยมีทั้งเครื่องมือที่ฟรี และเครื่องมือที่มีค่าใช้จ่าย (เหมาะสำหรับระดับงานที่ใหญ่มากขึ้น)



Firebase

รูปที่ 2.4 โลโก้บริการ “Firebase”

(อ้างอิงจาก <https://firebase.google.com/images/social.png>)

2.2.2 บริการแต่ละส่วนของ Firebase

บริการของ Firebase สามารถแบ่งได้เป็น 3 ส่วนด้วยกัน

2.2.2.1 Build better apps ส่วนช่วยเหลือของบริการ Applications

- Authentication: บริการดูแลและจัดการเข้าถึงของข้อมูล โดยการ Sign-in ข้อมูล ซึ่งรองรับ API ได้หลายอย่างเช่น Email, Facebook, Google ID, เบอร์โทรศัพท์, หรือแม้กระทั่งการกำหนด ID และ Password ได้ด้วยครับ
- Hosting: บริการ Hosting หน้า Web App หรือ Mobile App ซึ่งจะทำให้การ Upload ข้อมูล Application ของเราเผยแพร่ขึ้นไปยังบนอินเทอร์เน็ต
- Real-time Database: บริการด้านการเก็บค่าของข้อมูลและทำการประสานข้อมูลระหว่างผู้ใช้งานและอุปกรณ์ โดยใช้ระบบของ Cloud-hosted และเป็นระบบ NoSQL Database ซึ่งจะสามารถรับค่าข้อมูลได้หลายระบบปฏิบัติการ iOS, Android, Web Application และสามารถเชื่อมต่อข้อมูลภายหลังได้ ในขณะที่อุปกรณ์ไม่ได้เชื่อมต่อกับระบบอยู่ซึ่งจะทำการเชื่อมต่อได้ภายหลัง
- Cloud Storage: บริการเก็บข้อมูลไม่ว่าจะเป็นภาพ, เสียง, วิดีโอ ด้วยวิธีการที่ง่าย และเก็บบนพื้นที่ให้บริการของ Google

- Cloud Function: บริการคล้ายในส่วนของ Backend ของระบบ Application โดยจะช่วยเหลือการทำงานของระบบโดยไม่ต้องเขียนโปรแกรมทุกส่วน และใช้งานง่ายขึ้น
- Firestore: โดย SDKs ที่ทาง Firebase เตรียมไว้ให้เราก็มีอย่างครบครัน เช่น iOS, Android, Web, Node.js, Java, Python, Go, REST และ RPC APIs. โดยโครงสร้างจะเป็นแบบ NoSQL ที่เราสามารถจัดเก็บข้อมูลในรูปแบบ Document ที่จะผูก Fields กับ Values เข้าด้วยกัน ซึ่ง Document ก็จะถูกจัดเก็บใน Collections อีกที่หนึ่ง ซึ่งเราจะสามารถสร้าง Query From ไปจัดการเอาข้อมูลที่เราต้องการได้ในแต่ละ Document โดยในบริการ Cloud Firestore สามารถระบุชนิดของข้อมูลได้ด้วย ไม่ว่าจะเป็น ข้อความ, ตัวเลข และ ในส่วนของข้อมูลที่มีความซับซ้อนมีการซ้อนกันของข้อมูลหลายๆ เราก็สามารถสร้างเป็น collections รองภายใน Document และแบ่งข้อมูลเป็นลำดับชั้นเพื่อที่จะรองรับการเติบโตของข้อมูลในอนาคตได้ โดยเราสามารถออกแบบโครงสร้างได้ทุกรูปแบบที่จะสามารถทำงานได้อย่างดีที่สุดใน Application ของเรา

2.2.2.2 Improve app quality ส่วนช่วยเหลือภายใน Application ของทาง Firebase

- Crashlytics: ตรวจสอบปัญหาและข้อผิดพลาดต่างๆในส่วนของ Mobile Application และแบ่งแยกประเภทของปัญหาหรือข้อผิดพลาดที่พบเจอได้
- Performance Monitoring: สามารถแปลงคำสั่ง Code ต่างๆให้เป็นรูปแบบภาษาที่เข้าใจง่ายและเห็นภาพได้มากขึ้นผ่านระบบ Monitoring
- Test Lab: ช่วยทดสอบการใช้งาน Mobile Application ให้

2.2.2.3 Grow your business ส่วนช่วยเหลือระบบหลังบ้านเพื่อโอกาสทางธุรกิจ

- Google Analytics: ส่วนช่วยเก็บสถิติ พฤติกรรมของผู้ใช้งานจากทาง Google โดยจะแบ่งหมวดหมู่อย่างชัดเจน
- Remote Config: สามารถปรับเปลี่ยนหน้าตาของ Application ได้เลยผ่านทาง Remote โดยไม่ต้องไปแก้ที่ Code ของตัว App ได้เลย
- Cloud Message: สามารถเพิ่มความสามารถของ Mobile Application ให้รับการแจ้งเตือนได้โดยส่ง Message ไปยังทุก Platform ได้เลย

2.3 Web Application

“Web Application” เป็นเทคโนโลยีในการใช้ตัว Website ให้รองรับการทำงานแบบ Real-time มีการตอบสนองตลอดเวลาคล้ายคลึง Application หรือก็คือการผสมผสานระหว่าง Website + Application นั้นเอง ซึ่งระบบนี้ทางผู้พัฒนาจะเป็นตัวกำหนด Element ต่างๆภายใน Website ได้ด้วยตัวเอง ซึ่งจะทำให้มีแต่ส่วนที่จำเป็นต่อการใช้งานเท่านั้น เลยทำให้ไม่กินทรัพยากรตัวเครื่อง และสามารถรองรับการเปิดตัว Website ได้หลายระบบปฏิบัติการ ทั้งตัวของ Windows Mac รวมไปถึงระบบปฏิบัติการมือถืออย่าง iOS และ Android เพราะเป็นการเปิดตัวโปรแกรมผ่าน Website นั้นเอง ทำให้อุปกรณ์ที่มีโปรแกรม Browser นั้นสามารถเข้าใช้งานได้นั้นเอง

ในส่วนของ Client ของทาง Web Application นั้นจะเป็นการติดตั้ง Client ทาง Server ที่ให้บริการ ซึ่งการเชื่อมต่อไปยัง Client นั้นจะไม่จำเป็นต้องติดตั้งตัว Client ไว้ที่ต้นทาง ซึ่งต่างจากตัวของ Application ที่จำเป็นต้องติดตั้งบน Desktop หรือ บนมือถือเพื่อเรียกใช้งานตัว Client จุดเด่นอีกอย่างหนึ่ง คือข้อมูลที่ส่งหากัน ระหว่าง Client กับ Server มีปริมาณที่น้อยมากๆ ทำให้เราสามารถย้ายเซิร์ฟเวอร์ไปอยู่บนเครือข่าย Internet ได้ และสามารถใช้งานผ่าน Internet Connection ที่มีความเร็วต่างๆได้ ซึ่งทำให้สามารถใช้ Application เหล่านี้ได้ทุกที่บนโลก

2.4 JavaScript



รูปที่ 2.5 Javascript

(อ้างอิงจาก <https://www.javascript.com/>)

“JavaScript” เป็นภาษาคอมพิวเตอร์ที่ใช้ในการเขียนหน้าเว็บไซต์ ทำให้เว็บไซต์ตอบสนอง ซึ่งเป็นเพียงตัวช่วยในการอ่านข้อมูลบนภาษา HTML และ Java ภายในระบบเว็บไซต์ ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) โดยใช้การประมวลผลแบบ Object Oriented Programming ซึ่งได้รับความนิยมเป็นอย่างมากเพราะว่าเป็นโปรแกรมภาษาแบบเปิด (Open Source) และเป็นภาษาที่ทำให้ผู้พัฒนาสามารถออกแบบและสร้าง Web Page ตามความต้องการได้ รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA หรือ European Computer Manufacturers Association

JavaScript ถูกพัฒนาขึ้นโดย Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator2.0 เพื่อใช้สร้าง Web Page โดยติดต่อกับ Server แบบ Live Wire

ซึ่งประกอบด้วย Standard Library ของ Object ต่างๆ เช่น Array, Date, และ Math, รวมทั้งแกนหลักของภาษา โดยแกนหลักของจาวาสคริปต์สามารถขยายให้รองรับการใช้งานได้หลากหลายวัตถุประสงค์

2.5 Node.js



รูปที่ 2.6 Node.js

(อ้างอิงจาก <https://nodejs.org/en/>)

“Node.js” เป็นบริการ Open Source และ Cross Platform ของ Javascript runtime environment เป็นเครื่องมือที่ได้รับความนิยมมากที่สุดตัวหนึ่ง

Node.js ทำงานในระบบ V8 Javascript โดยใช้แกนกลางของ Google Chrome ในการทำงาน นอกเหนือจากบราวเซอร์ ซึ่งจะช่วยยกระดับการทำงาน (และทำงานอย่างต่อเนื่อง) ของระบบ ได้อย่างรวดเร็ว ยิ่งขึ้น และสามารถเข้าใจ ปรับปรุง และยกระดับการทำงานของ V8 Javascript เพราะฉะนั้นการทำงานของ Javascript โดยใช้ Node.js จึงมีประสิทธิภาพมาก

Node.js ทำงานได้โดยไม่ต้องสร้างการร้องขอใหม่ทุกครั้ง เมื่อ Node.js ต้องการที่จะร้องขอการทำงาน I/O (Input และ Output) จะคล้ายๆกับการอ่านข้อมูลของระบบ Network เข้าถึงฐานข้อมูลและไฟล์ของระบบ หากการเชื่อมต่อหลุดไป จะไม่มีการหลุดออกจากระบบ Node.js จะกลับมาทำงานอีกครั้งหากมีการเชื่อมต่อหรือตอบสนองอีกครั้ง

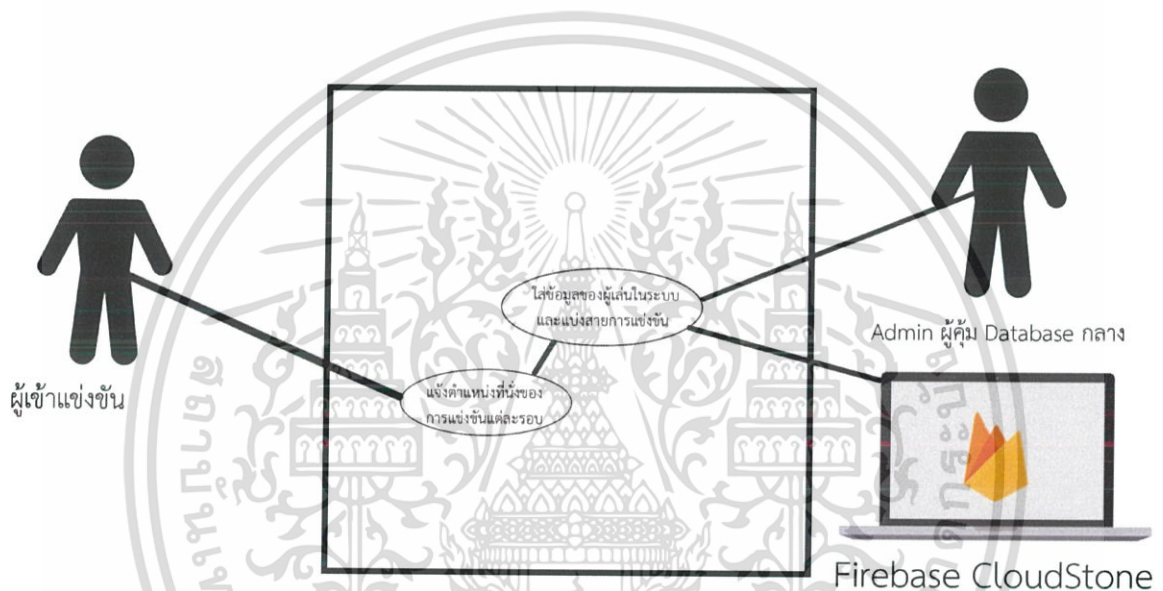


บทที่ 3

การออกแบบ

3.1 ยูสเคสไดอะแกรม (Use case diagram)

3.1.1 ยูสเคสไดอะแกรม ระบบการลงทะเบียนและการจับคู่

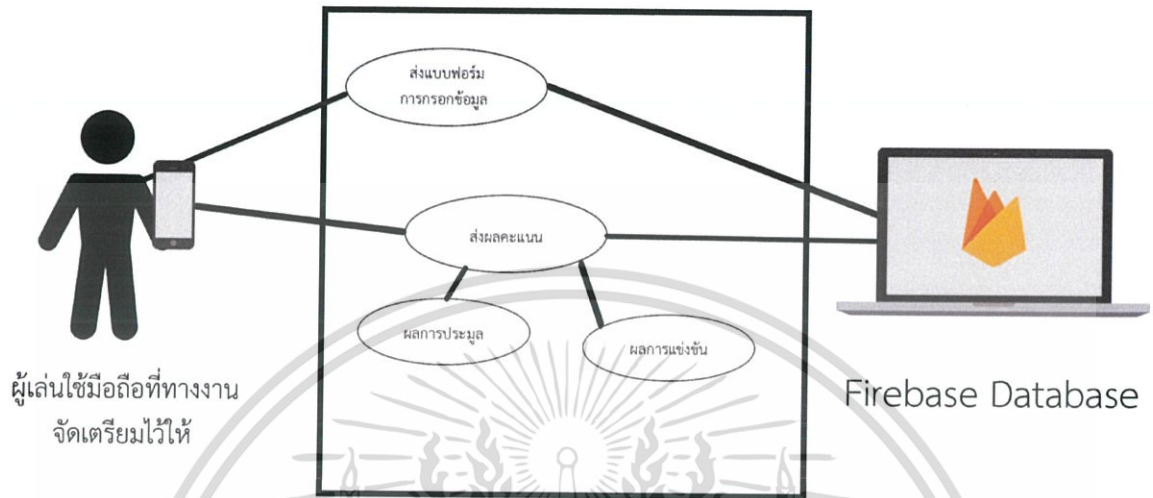


รูปที่ 3.1 ยูสเคสไดอะแกรม ระบบการลงทะเบียนและการจับคู่

จากภาพที่ 3.1 แสดงยูสเคสไดอะแกรมของ ระบบการลงทะเบียนและการจับคู่แอปพลิเคชันระบบจัดการแข่งขันและการคิดผลการแข่งขันเกม “บริดจ์” โดยการแข่งขันทะจะมีการเปิดรับสมัครผู้เข้าแข่งขัน

โดยจะมีการแสดงชื่อผู้เล่นและชื่อทีม อีกทั้งรวมไปถึงเบอร์ที่ติดตัวแต่ละคนในการแข่งขัน นำมาแสดงที่หน้าเว็บไซต์ เพื่อให้ผู้เล่นสามารถเข้าสู่โปรแกรมการแข่งขันได้ในการแข่งขัน โดยข้อมูลที่คีย์เข้าไปจากส่วนกลาง จะถูกเก็บบันทึกไว้ใน Cloud Firebase สามารถลบและแก้ไขได้ที่หน้า Web Application

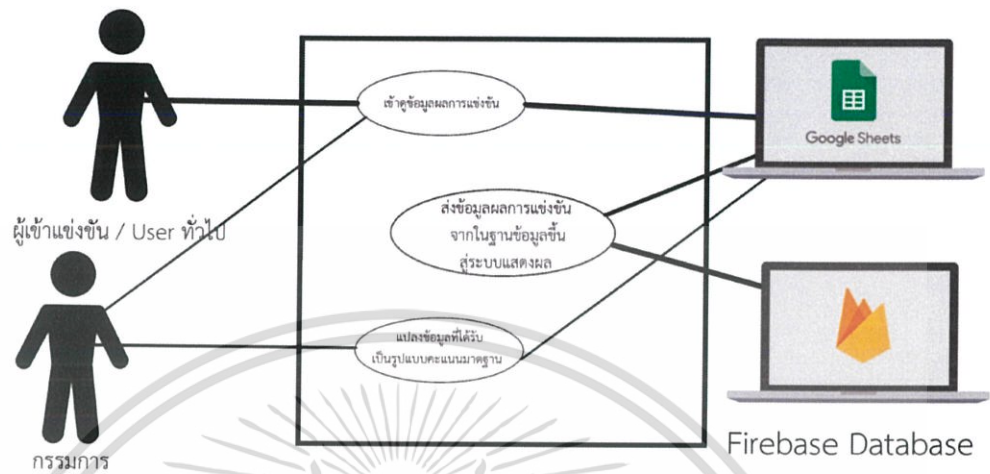
3.1.2 ยูสเคสไดอะแกรม ระบบการส่งคะแนน



รูปที่ 3.2 ยูสเคสไดอะแกรม ระบบการส่งคะแนน

จากภาพที่ 3.2 หลังจากที่ทำกรการจับคู่เป็นที่เรียบร้อยตามยูสเคสไดอะแกรมที่ 3.1 แล้ว เมื่อเข้าสู่เกมการแข่งขันให้ทำการแข่งขันตามปกติและจะใช้เครื่องโทรศัพท์มือถือในการเชื่อมต่อกับตัวของ Server กลางซึ่งเป็นเครื่องคอมพิวเตอร์ โดยจะส่งข้อมูลของผลการแข่งขันแต่ละรอบ คือหน้าการ์ดที่ประมวลได้, จำนวนทองที่ประมวลได้ ทิศของผู้ที่ประมวลได้ และค่าDoubleหรือRedouble ซึ่งจะทำการส่งไปยังเครื่องคอมพิวเตอร์และทำการอัปเดตเข้าสู่ Firebase Database เพื่อที่จะเก็บข้อมูลต่อไป หลังจากนั้นจะใช้วิธีเดียวกันนี้ในการรับข้อมูล จำนวนทองทำได้จริงในเกมนั้นๆ เช่นกัน

3.1.3 ยูสเคสไดอะแกรม ระบบการแสดงผล

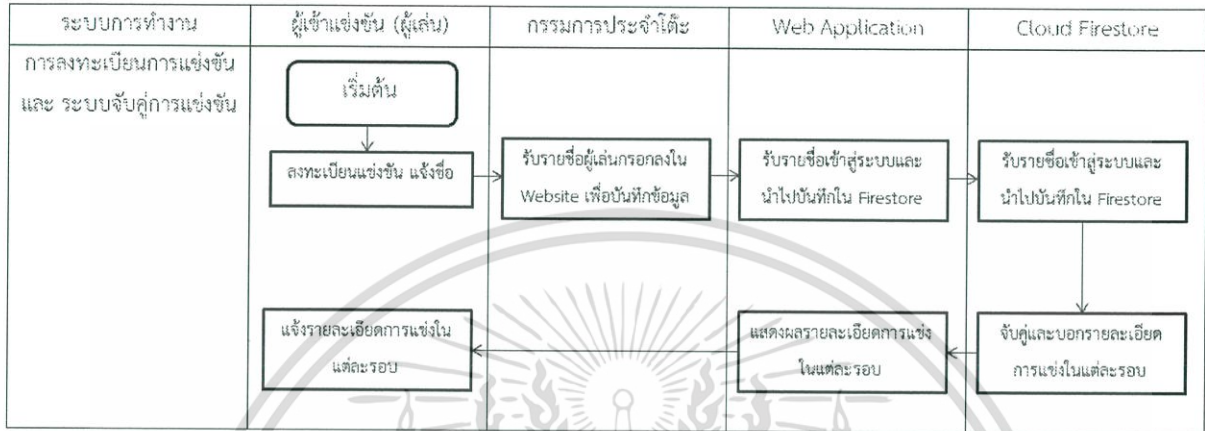


รูปที่ 3.3 ยูสเคสไดอะแกรม ระบบการแสดงผล

จากภาพที่ 3.3 หลังจากที่เราส่งข้อมูลเสร็จเรียบร้อยแล้วตามยูสเคสที่ 3.2 ผู้จะทำการเรียกข้อมูลจากฐานข้อมูล Firebase ออกมาและอัปโหลดไปยัง Google Sheet อัตโนมัติ ซึ่งเป็นข้อมูลที่ได้จากยูสเคสไดอะแกรมที่ 3.2 ที่ส่งมา หลังจากนั้นกรรมการจะเป็นผู้ที่นำผลลัพธ์ที่ได้นั้นมากรอกใส่ในตารางเพื่อคิดผลคะแนนจากข้อมูลที่ได้ในแต่ละรอบ หลังจากนั้นเข้าแข่งขันหรือผู้ใช้งานที่ต้องการทราบผลการแข่งขัน จะสามารถเปิดดูหน้า Website ได้ โดยจะทำการใส่ตาราง Google Sheet ลงบนหน้าเว็บและสามารถดูผลการแข่งขันได้ในแต่ละรอบหรือหลังจบการแข่งขันเรียบร้อยแล้ว

3.2 การออกแบบการทำงานของระบบ

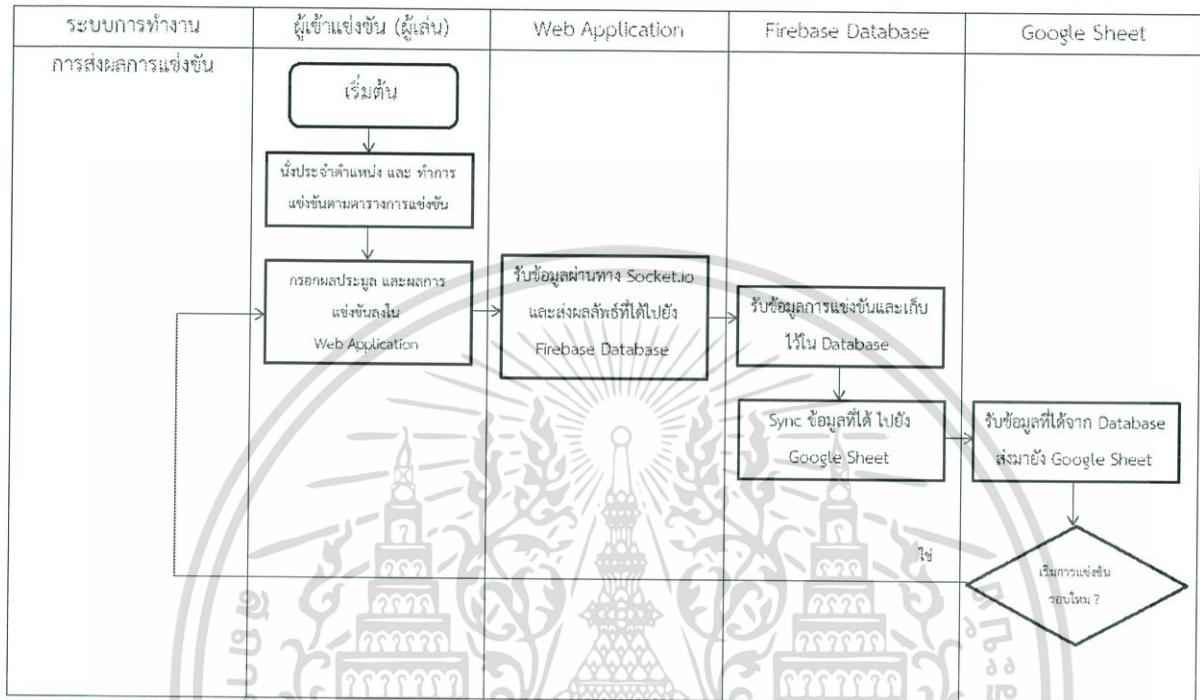
3.2.1 ระบบการลงทะเบียนการแข่งขัน และ ระบบจับคู่การแข่งขัน



รูปที่ 3.4 การออกแบบการทำงานของระบบการลงทะเบียนการแข่งขัน และ ระบบจับคู่การแข่งขัน

จากภาพที่ 3.4 เป็นระบบที่ใช้สำหรับการดูโปรแกรมการแข่งขัน ผลการจับคู่ในแต่ละรอบ เริ่มต้นเป็นการกรอกข้อมูลชื่อลงใน Web application เมื่อกด Submit ข้อมูลชื่อจะถูกบันทึกลงใน firebase Cloud Firestone ซึ่งเมื่อเก็บเสร็จจะดึงข้อมูลจาก Firestone มาแสดงผลบนหน้าจอหลัก ซึ่งเราสามารถแก้ไขข้อมูลที่หน้าเว็บได้ แต่เมื่อเรากดปุ่ม finish เราจะพบหน้าการจับคู่ของผู้เล่นในการแข่งขัน ซึ่งจะอยู่ในรูปแบบ “พบกันหมด” (Round Robin) และเราสามารถนำข้อมูลไปพิมพ์ออกมาได้ เพื่อใช้ในการแข่งขันต่อไป

3.2.2 ระบบการส่งผลการแข่งขัน



รูปที่ 3.5 การออกแบบการทำงานของระบบการส่งผลการแข่งขัน

จากภาพที่ 3.5 หลังจากเข้าสู่การแข่งขัน จะมีการประมวลตามกติกาของเกมปริศนาคณิตศาสตร์ตามปกติ หลังจากนั้นเมื่อจบผลการประมวลจะมีการส่งผลประมวลโดยใช้ระบบ Web application ผ่านทางโทรศัพท์มือถือส่วนกลางที่มีการเชื่อมต่อไว้แล้ว ผู้เข้าแข่งขันจะมีการกรอกผลการประมวลและยืนยันทั้ง 2 ฝ่าย หลังจากยืนยันแล้วจะทำการส่งข้อมูลไปยังฝั่งเครื่องรับผ่านทาง Socket.io ซึ่งหลังจากรับข้อมูลแล้ว จะมีการอัปเดตข้อมูลไปยัง Firebase Database ซึ่งจะเก็บข้อมูลไว้แยกกันหลังจากนั้นจะมีการเชื่อมต่อข้อมูลจาก Firebase Database ไปยัง Google-Sheet ซึ่งจะทำให้ข้อมูลที่ได้นั้นส่งมายังส่วนกลางและสามารถนำไปประมวลผลเป็นคะแนนอย่างเป็นทางการได้ต่อไป หลังจากประมวลเสร็จก็จะเริ่มการแข่งขัน และเมื่อจบการแข่งขันก็จะมีผลการแข่งขันว่า ทิศที่ประมวลไปนั้นทำการแข่งขันได้เท่าไร โดยจะมีการส่งผลโดยใช้ระบบ Web application ผ่าน Socket.io และเชื่อมต่อข้อมูลไปยัง Firebase Database และ Google-Sheet ได้เช่นเดียวกัน

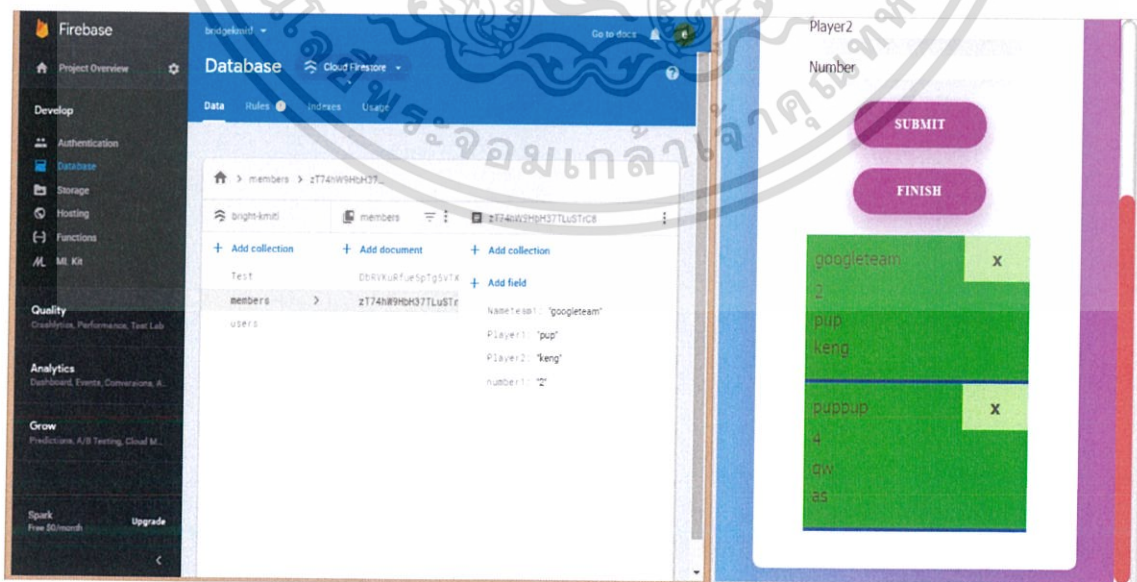
บทที่ 4

ผลการทดลอง

จากการพัฒนาระบบและปรับปรุงแก้ไขระบบการทำงานของระบบ แอปพลิเคชันระบบจัดการแข่งขันและการคิดผลการแข่งขันเกม “บริดจ์” นั้น

4.1 ระบบการลงทะเบียนการแข่งขัน และ ระบบจับคู่การแข่งขัน

เมื่อเราได้รายชื่อของผู้เข้าแข่งขันแล้ว จากนั้นส่วนกลางทำการบันทึกข้อมูลเข้าสู่ระบบเพื่อบันทึกไว้ใน Firebase Cloud Firestone หลังจากนั้นข้อมูลที่อยู่ใน Firestone จะถูกดึงขึ้นมาที่หน้าเว็บ เพื่อแสดงผลการจับคู่การแข่งขันในแต่ละรอบ ซึ่งผู้เข้าแข่งขัน สามารถเปิดดู รายละเอียดของจับคู่, ตำแหน่งที่นั่ง (หมายเลขโต๊ะ), ชุดไฟ, ฝั่งที่นั่ง และ รอบที่แข่งได้ ข้อมูลเหล่านี้เป็นประโยชน์อย่างมากในการแข่งขัน ถ้าข้อมูลที่บันทึกไปเกิดการผิดพลาด เราสามารถแก้ไขข้อมูลได้ และเมื่อแข่งขันเสร็จ ส่วนกลางสามารถลบข้อมูลการแข่งขันได้ทันที



รูปที่ 4.1 ภาพหน้าจอการแสดงผลจับคู่การแข่งขัน และการบันทึกลงใน Cloud firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ระบบการส่งผลการประมวลและผลการแข่งขัน

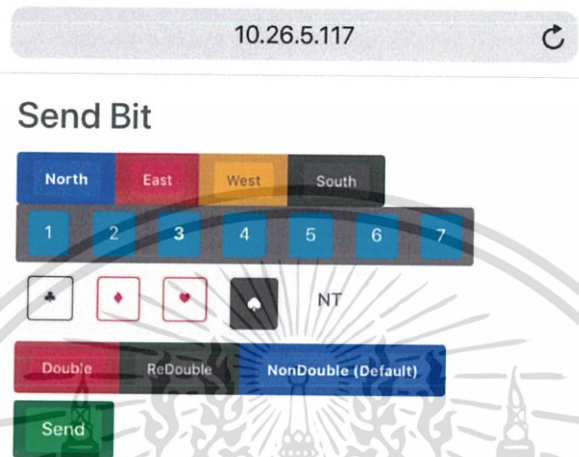
หลังจากที่เข้าสู่ตำแหน่งที่นั่งการแข่งขันแล้ว บนพื้นที่ในการเล่น (โต๊ะ) นั้นจะมีการวางเครื่องโทรศัพท์มือถือที่สามารถเข้าสู่เว็บไซต์ได้ และเป็นเครื่องภายในระบบการแข่งขันไม่ใช่เครื่องส่วนตัวของผู้เข้าแข่งขัน ซึ่งจะเปิดเข้าสู่ระบบไว้และแสดงผลว่าโทรศัพท์เชื่อมต่ออยู่นั้นอยู่ในหมายเลขตำแหน่งที่เท่าไรซึ่งจะแจ้งเตือนก่อนเข้าสู่การใส่ผล



รูปที่ 4.2 ภาพหน้าจอแสดงขั้นตอนส่งผลการแข่งขัน

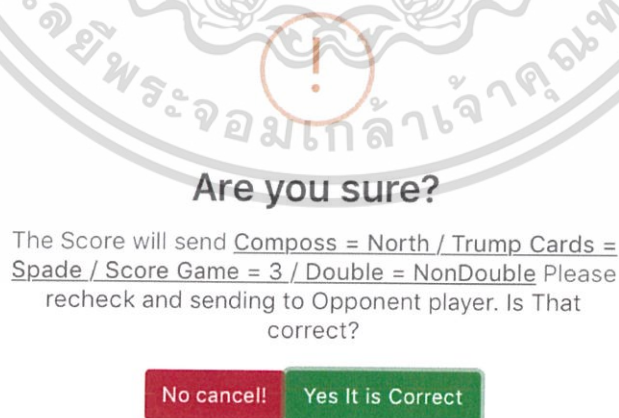
รูปที่ 4.3 ภาพหน้าจอแจ้งเตือนหมายเลขตำแหน่งที่นั่ง (จากภาพอยู่ที่ตำแหน่งโต๊ะหมายเลข 1)

หลังจากที่ทำการประมูลเสร็จสิ้นจะมีการใส่ผลคะแนน โดยจะต้องมีข้อมูลดังต่อไปนี้ “ทิศของผู้ที่สามารถประมูลได้, หน้าตองและจำนวนตองที่ทำการประมูล และ การผูกมัด ดับเบิ้ล/รีดับเบิ้ล (ถ้ามี)” ตามภาพที่ 4.4



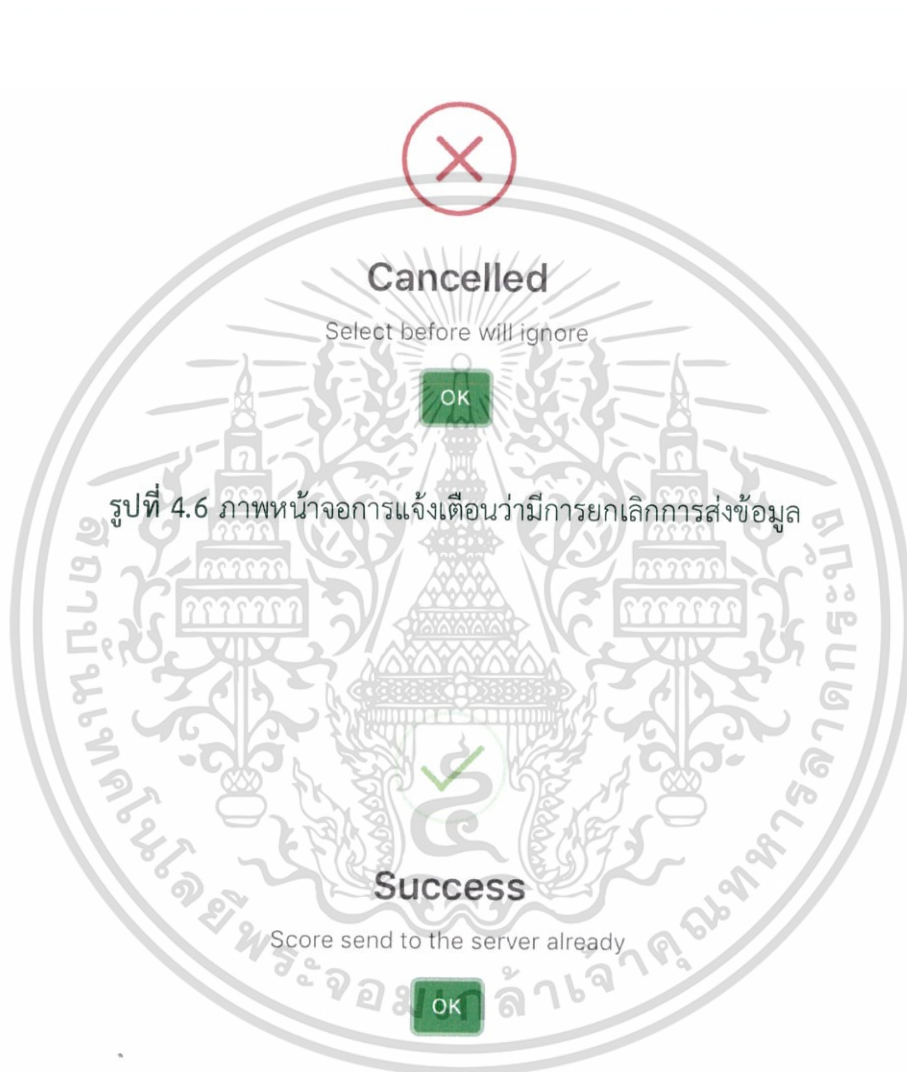
รูปที่ 4.4 ภาพหน้าจอการเลือกผลการประมูล

หลังจากที่ใส่ผลข้อมูลแล้วนั้น จะต้องทำการกด Send เพื่อทำการตรวจสอบและส่งให้ฝ่ายตรงข้ามดูผล พร้อมกับกดยืนยันข้อมูลว่า ข้อมูลที่ทำการกรอกไปนั้นตรงกันหรือไม่ ตามภาพที่ 4.5



รูปที่ 4.5 ภาพหน้าจอการแจ้งเตือนเพื่อตรวจสอบผลการประมูล

หากกดปุ่ม ยืนยัน (ปุ่มสีเขียว) จะแจ้งเตือนว่าส่งผลสำเร็จ ตามภาพที่ 4.7 และข้อมูลนั้นจะถูกส่งมายังเครื่องรับServer ที่ทำการเชื่อมต่อกันผ่าน Socket.io และทำการส่งข้อมูลไปยัง Firebase Database เพื่อเก็บข้อมูลต่อไป แต่หากกดปุ่ม ปฏิเสธ (ปุ่มสีแดง) จะมีการแจ้งเตือนว่ายกเลิกการส่งผล ตามภาพที่ 4.6 ซึ่งจะไม่มีการส่งผลกลับไปซึ่งสามารถกลับไปแก้ไขและส่งผลอีกครั้งได้



รูปที่ 4.6 ภาพหน้าจอการแจ้งเตือนว่ามีการยกเลิกการส่งข้อมูล

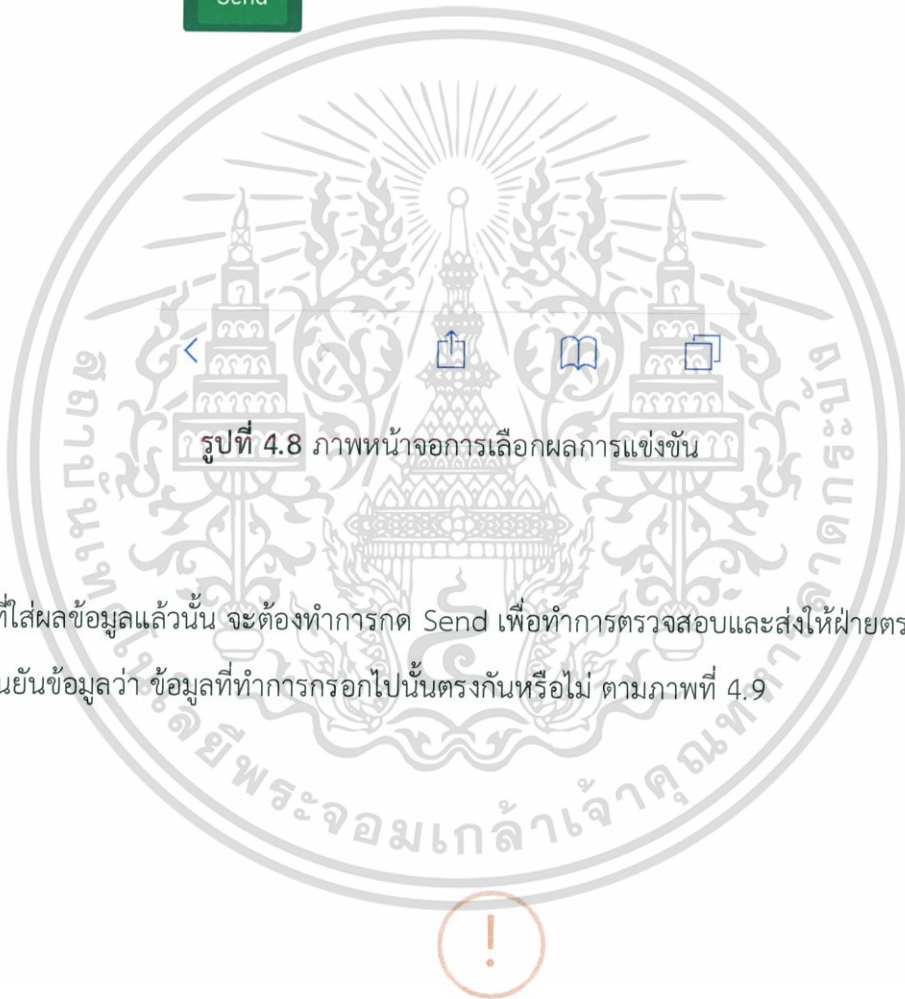
รูปที่ 4.7 ภาพหน้าจอการแจ้งเตือนว่าการส่งข้อมูลนั้นสำเร็จเรียบร้อย

หลังจากนั้นจะเริ่มการแข่งขันในบอร์ดนั้นๆ โดยเมื่อทำการแข่งขันเสร็จเรียบร้อยแล้ว ผู้เข้าแข่งขันจะต้องส่งผลการแข่งขันว่า “ผู้ที่ประมวลได้นั้นสามารถทำได้ทั้งหมดก็ต่อ” โดยจะต้องเลือกจำนวนตอนที่ทำได้ (1-13) ตามภาพที่ 4.8

Send Game Score

1	2	3	4	5	6	7
8	9	10	11	12	13	

Send



รูปที่ 4.8 ภาพหน้าจอการเลือกผลการแข่งขัน

หลังจากที่ใส่ผลข้อมูลแล้วนั้น จะต้องทำการกด Send เพื่อทำการตรวจสอบและส่งให้ฝ่ายตรงข้ามดูผล พร้อมกับกดยืนยันข้อมูลว่า ข้อมูลที่ทำการกรอกไปนั้นตรงกันหรือไม่ ตามภาพที่ 4.9

Are you sure?

The Score will send Gamescore is = 1 Please recheck and sending to Opponent player. Is That correct?

No cancel! Yes It is Correct

รูปที่ 4.9 ภาพหน้าจอการแจ้งเตือนเพื่อตรวจสอบผลการแข่งขันและส่งให้ผู้เล่นฝ่ายตรงข้ามเพื่อยืนยัน

หากกดปุ่ม ยืนยัน (ปุ่มสีเขียว) จะแจ้งเตือนว่าส่งผลสำเร็จ ตามภาพที่ 4.7 และข้อมูลนั้นจะถูกส่งมายังเครื่องรับServer ที่ทำการเชื่อมต่อกันผ่าน Socket.io และทำการส่งข้อมูลไปยัง Firebase Database เพื่อเก็บข้อมูลต่อไป แต่หากกดปุ่ม ปฏิเสธ (ปุ่มสีแดง) จะมีการแจ้งเตือนว่ายกเลิกการส่งผล ตามภาพที่ 4.6 ซึ่งจะไม่มีการส่งผลกลับไปซึ่งสามารถกลับไปแก้ไขและส่งผลอีกครั้งได้เช่นเดียวกับผลการประมวล

หลังจากที่มีการกดปุ่มยืนยันการส่งแล้ว ผลที่ได้จะถูกส่งมาจาก Firebase Database โดยมีการกำหนดเลขโต๊ะไว้ และผลจะอยู่ในข้อมูลของใน Node ประจำโต๊ะซึ่งจะเก็บข้อมูลที่ได้รับมาจากการส่ง Socket.io ตามภาพที่ 4.10



รูปที่ 4.10 ภาพการเก็บข้อมูลภายใน Firebase Database ของตำแหน่งโต๊ะหมายเลข 1

4.3 การส่งข้อมูลจาก Firebase ไปยัง GoogleSheet

หลังจากที่ข้อมูลถูกส่งไปยัง Firebase Database แล้วนั้น จะมีการนำเอา Firebase Function ซึ่งเป็นส่วนหนึ่งของระบบ Firebase ทำให้สามารถเชื่อมต่อข้อมูลจากทาง Firebase และส่งไปยัง Google-Sheetได้ตามภาพที่ 4.11

User : 01		A	B	C	D	E	F	G
AUsername: "01	ID (Table)	1	2	3	4	5	6	
ComposBit: "West	Compos	West	-	-	-	-	-	
ScoreBit: "3	Score Bit	3	-	-	-	-	-	
TrumpBit: "Clut	Trump	Club	-	-	-	-	-	
ZDouble: "NonDoubl	Double	NonDouble	-	-	-	-	-	
User : 01 Game								
AUsername: "01	ID (Table)	1	2	3	4	5	6	
Gamescore: "11	Result Score	11	-	-	-	-	-	

รูปที่ 4.11 ภาพการเชื่อมต่อข้อมูลจาก Firebase ไปยัง Google Sheet

ซึ่งเมื่อได้รับผลมาจากทาง Google-Sheet แล้วผู้คุมหรือกรรมการการแข่งขันสามารถนำข้อมูลไปประมวลผลและคิดผลคะแนนได้ต่อไป



บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

แอปพลิเคชันระบบจัดการแข่งขันและการคิดผลการแข่งขันเกม “บริดจ์” สามารถที่จะจัดการระบบการแข่งขันและใช้สิ่งทดแทนเครื่องมือการแข่งขัน (BridgeMate) ได้และมีการส่งผลมายัง Google-Sheet เพื่อถ่ายทอดการนำข้อมูลไปใช้ต่อ ซึ่งสามารถแบ่งการทำงานออกเป็นช่วงดังต่อไปนี้

5.1.1. ระบบการจับคู่การแข่งขัน สามารถใช้จัดโปรแกรมการแข่งขันได้จริง เปิดดูง่าย ใช้เวลาในการบันทึกข้อมูลน้อยและถูกต้อง สามารถเปิดดูในโทรศัพท์มือถือได้ โปรแกรมการแข่งขันสามารถอ้างอิงในการคิดคะแนนได้ เพื่อเช็คความถูกต้องว่าถูกคู่ ถูกตำแหน่งที่นั่งหรือไม่ ชุดไฟเดียวกันหรือไม่

5.1.2. ระบบการส่งคะแนน เมื่อทำการประมวลเสร็จสิ้นแล้วทำการกรอกผลการประมวลไปยังโทรศัพท์มือถือที่มีการเปิดหน้าเว็บแอปพลิเคชันไว้ ซึ่งสามารถทดแทนการใช้เครื่องมือการแข่งขัน (BridgeMate) ได้ และส่งผลไปยัง Database ผ่านทาง Socket.io เพื่อบันทึกผลที่ส่งมาได้ หลังจากนั้น ผลที่ออกมาจะถูกส่งไปยัง Google-Sheet ผ่านทาง Firebase Function เพื่อที่จะนำผลค่าที่ได้นั้นไปทำการประมวลผลและคำนวณเป็นคะแนนอย่างเป็นทางการของทางกีฬา เพื่อที่จะวัดผลการแข่งขันได้

5.2 ปัญหาที่พบระหว่างการทดลอง

- 1) - ในช่วงแรกที่ได้ทำโปรเจก การดำเนินงานไม่ได้เป็นไปตามจุดประสงค์ที่วางไว้
- 2) - ในระหว่างการทำงาน มีความคิดเห็นที่แตกต่างกัน ทำให้การทำงานล่าช้า

5.3 แนวทางการปรับปรุงและพัฒนาต่อยอดในอนาคต

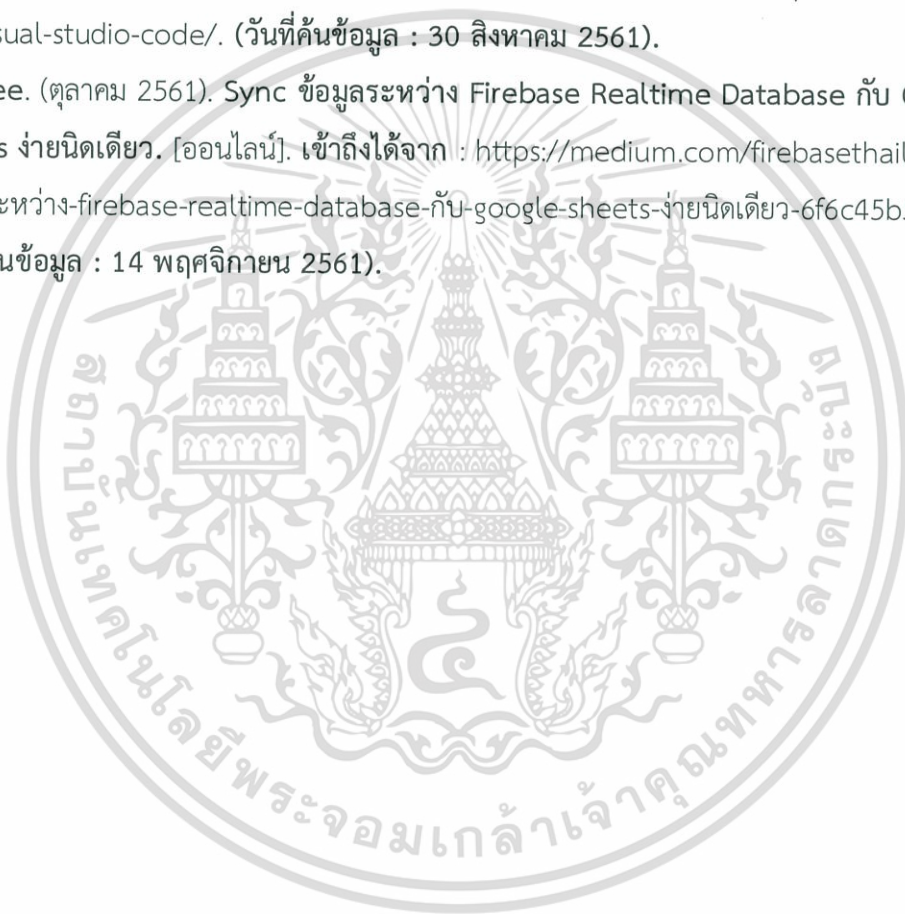
- 1) - ปรึกษาอาจารย์ที่ปรึกษาโปรเจกต์ถึงจุดประสงค์ที่ต้องการมากขึ้นกว่าเดิม และเร่งเวลาในการทำโครงการเพิ่มมากขึ้น
- 2) - สื่อสารกันมากขึ้นระหว่างการทำงาน ตลอดการทำโครงการ
- 3) - ปรับปรุงให้สามารถโชว์ตารางการจับคู่ในทุกๆรอบได้โดยเปลี่ยนไปในแต่ละรอบ
- 4) - ปรับรูปแบบในการใช้งานให้นำใช้งานมากยิ่งขึ้น



บรรณานุกรม

- [1] Chai Phonbopit (กุมภาพันธ์ 2548). สร้างแอป Chat ด้วย Node.js และ Socket.IO. [ออนไลน์]. เข้าถึงได้จาก : <http://www.siamhtml.com/real-time-chat-with-node-js-and-socket-io/>. (วันที่ค้นข้อมูล : 8 มกราคม 2562).
- [2] Google. (ม.ป.ป). Cloud Firestone. [ออนไลน์]. เข้าถึงได้จาก : <https://firebase.google.com/docs/firestore>. (วันที่ค้นข้อมูล : 16 กันยายน 2561).
- [3] Google. (ม.ป.ป). Firebase Realtime Database. [ออนไลน์]. เข้าถึงได้จาก : <https://firebase.google.com/docs/database>. (วันที่ค้นข้อมูล : 16 กันยายน 2561).
- [4] Chonlatit (Karn). (กุมภาพันธ์ 2561). ทำเว็บแบบ Realtime ง่ายๆ ด้วย socket.io + node.js + react. [ออนไลน์]. เข้าถึงได้จาก : <https://engineering.thinknet.co.th/ทำเว็บแบบ-realtime-ง่ายๆด้วย-socket-io-node-js-react-f299bfa4e78a>. (วันที่ค้นข้อมูล : 8 มกราคม 2562).
- [5] Jirawatee. (พฤษภาคม 2560). รู้จัก Cloud Functions for Firebase ตั้งแต่ Zero จนเป็น Hero. [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/firebaseethailand/รู้จัก-cloud-functions-for-firebase-ตั้งแต่-zero-จนเป็น-hero-1c94acbb55af> (วันที่ค้นข้อมูล : 20 กันยายน 2561).
- [6] Jedsada Saengow. (พฤษภาคม 2561). [Firebase] คืออะไร มาดูวิธีสร้าง Project และทำความเข้าใจกับ Firebase. [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/jed-ng/firebase-คืออะไร-มาดูวิธีสร้าง-project-และทำความเข้าใจกับ-firebase-d48bfac67b14>. (วันที่ค้นข้อมูล : 20 กันยายน 2561).
- [7] Jedsada Saengow. (สิงหาคม 2561). รู้จัก Firebase Realtime Database ตั้งแต่ Zero จนเป็น Hero. [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/firebaseethailand/รู้จัก-firebase-realtime-database-ตั้งแต่-zero-จนเป็น-hero-5d09210e6fd6>. (วันที่ค้นข้อมูล : 20 กันยายน 2561).
- [8] นายมงคล เกตุมาน. (ม.ป.ป). กฎ กติกา การแข่งขันกีฬาบริดจ์ : The Laws of Duplicate Contract Bridge. [ออนไลน์]. เข้าถึงได้จาก : http://www.thailandbridgeleague.com/data/content/34/pdf_th/11-25-02.pdf . (วันที่ค้นข้อมูล : 18 สิงหาคม 2561).
- [9] (2554). Firebase คืออะไร และมีข้อดีอย่างไรบ้าง ?. [ออนไลน์]. เข้าถึงได้จาก <https://www.softmelt.com/article.php?id=588>. (วันที่ค้นข้อมูล : 30 สิงหาคม 2561).
- [10] Akkachai Chusang. (2548). รู้จักกับ Visual Studio Code (วิชวล สตูดิโอ โค้ด). [ออนไลน์]. เข้าถึงได้จาก : <http://mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>. (วันที่ค้นข้อมูล : 30 สิงหาคม 2561).

- [11] มาสเตอร์ อี๋ง. (พฤศจิกายน 2560). การ Save ข้อมูล ลง Firebase. [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/มาสเตอร์-อี๋ง/การ-save-ข้อมูล-ลง-firebase-10d14e1a056e>. (วันที่ค้นข้อมูล : 20 กันยายน 2561).
- [12] Thanongkiat Tamtai. (ตุลาคม 2560). เชื่อมชั้นกับ Cloud Firestore ระบบฐานข้อมูลที่เปิดตัวใหม่ล่าสุดจาก Firebase แบบจัดเต็ม. [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/firebasethailand/เชื่อมชั้นกับ-firebase-cloud-firestore-ระบบฐานข้อมูลที่เปิดตัวใหม่ล่าสุดจาก-firebase-แบบจัดเต็ม-d001e43e2be7>. (วันที่ค้นข้อมูล : 10 มกราคม 2562).
- [13] โค้ชเอก. (พฤศจิกายน 2560). สรุป 20 เทคนิคการใช้งาน Visual Studio Code (Editor) ที่น่าสนใจ. [ออนไลน์]. เข้าถึงได้จาก : <https://codingthailand.com/blog/สรุป-20-เทคนิคการใช้งาน-visual-studio-code/>. (วันที่ค้นข้อมูล : 30 สิงหาคม 2561).
- [14] Jirawatee. (ตุลาคม 2561). Sync ข้อมูลระหว่าง Firebase Realtime Database กับ Google Sheets ง่ายนิดเดียว. [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/firebasethailand/sync-ข้อมูลระหว่าง-firebase-realtime-database-กับ-google-sheets-ง่ายนิดเดียว-6f6c45b54080>. (วันที่ค้นข้อมูล : 14 พฤศจิกายน 2561).





ภาคผนวก

Bridge Tournament Management Application

Mr. Ekanant Chanchuey¹, Mr. Siripob Laurujisawat²
Asst.Prof. Nitjaree Satayarak³

Abstract

This research aims to Solve the problem of using expensive Bridge Equipments (BridgeMate). And in each competition Must use a lot Therefore making the application to use to calculate the bridge card score, equivalent to the performance of the Bridge Equipments It is a lot of cost reduction. And effective.

Introduction

At present, the bridge card game has become well known in the new form of playing cards with love and flair in playing for not many. There are two main things: cards and scoring machines. With the scoreboard that has a high price if buying to practice Yourself at home may be too expensive But scoring can be done by hand without using a scoreboard. But for use in the competition, there must be a scoring tool that meets the standards, preventing the exploitation of the team and For confidence in the competition, however, we are getting more and more popular and everyone without having to spend money on buying a calculator, but can use the same score, can collect the player's information and the results can get a lot of points. In the firestore, we can retrieve the information in the desired section.

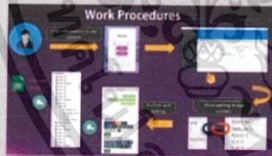


Fig. 1 The system of Application

Methodology

The application consists of 4 technologies

- 1) **Firestore** : Firestore Platform is a service from Google which will take care of the Backend the system works Applications, which will assist in analyzing the data. Data Collection And providing the Service Applications or Application Programming Interface (API). Will serve Firestore Applications supports multiple operating systems, for example iOS, Android, Web Application with the free tools and tool costs (Suitable for Bigger scale).
- 2) **Google sheet** : Google Sheet is the way Google intended to create or open a document. It is available in the form of an Online Service, which does not need to install anything extension.
- 3) **Node.js** : Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a browser. JavaScript is used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser.
- 4) **Socket.io** : is a library that enables real-time, bidirectional and event-based communication between the browser and the server.

Results

Fig. 1 The page of adding contestants' information.



Fig. 2 Page of submitting bid results and submitting points.



Fig. 3 Data sheet of scoring From the database.

Team	Score	Score	Score	Score	Score	Score	Score	Score	Score
Team 1	100	100	100	100	100	100	100	100	100
Team 2	100	100	100	100	100	100	100	100	100
Team 3	100	100	100	100	100	100	100	100	100
Team 4	100	100	100	100	100	100	100	100	100

Conclusion

From the result above, we can explain that :

- Fig.1 Adding contestants' information to the firebase database
- Fig.2 Press the button to send the value. Socket.io will receive the result of pressing the button to send the value to nodejs (server). Then nodejs accepts and then updates up Firestore.
- Fig.3 Then send the function from Firestore to the google sheet.

References

- [1] "Firestore technology." found on website : <https://firebase.google.com/products/#develop-features>
- [2] "getting-started-with-nodejs." found on website : <https://devahoy.com/posts/getting-started-with-nodejs/>
- [3] "Google sheet is ?" found on website : https://support.google.com/docs/topic/9054603?hl=th&ref_topic=1382883&authuser=0
- [4] "Socket.io." found on website : <https://socket.io/docs/>



ภาคผนวก ข

การติดตั้งโปรแกรมสำหรับการใช้งาน

ข.1 การติดตั้งโปรแกรม Visual Studio Code

1.ดาวน์โหลดตัวติดตั้ง VS Code ได้ที่ <https://code.visualstudio.com> เมื่อเข้าเว็บแล้วให้กดปุ่มดาวน์โหลด



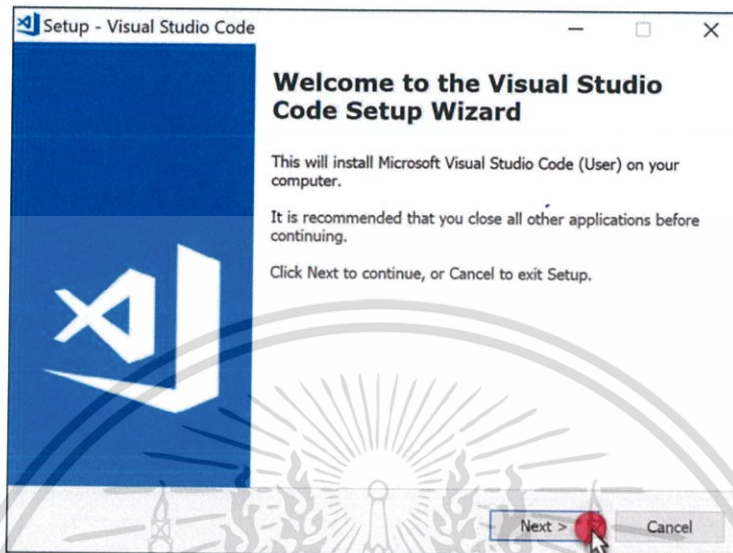
รูปที่ ข.1 หน้าต่างดาวน์โหลดโปรแกรมจาก Website

2.รอกจนกว่าจะดาวน์โหลดตัวติดตั้งเสร็จ และเมื่อดาวน์โหลดเสร็จแล้วให้กดเปิดตัวติดตั้งขึ้นมา



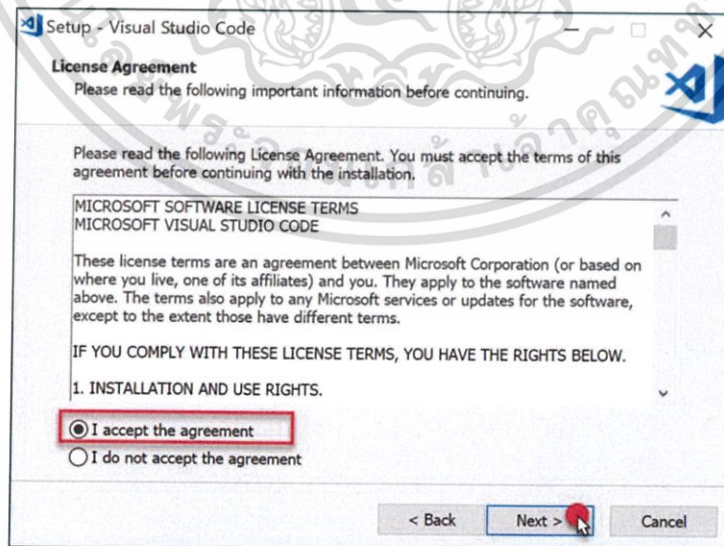
รูปที่ ข.2 เป็นการรอดาวน์โหลด โปรแกรม Visual Studio Code

3.เมื่อเปิดตัวติดตั้งขึ้นมาแล้ว จะแสดงหน้าจอ Welcome to the Visual Studio Code Setup Wizard ให้กดปุ่ม Next >



รูปที่ ข.3 ขั้นตอนการติดตั้งโปรแกรม

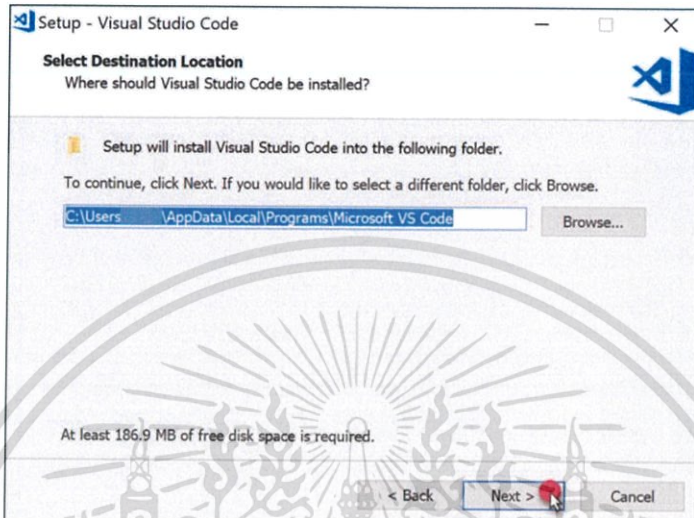
4.หลังจากนั้นจะแสดงหน้าจอ License Agreement ให้เลือก I accept the agreement แล้วกด Next >



รูปที่ ข.4 ขั้นตอนการติดตั้งโปรแกรม

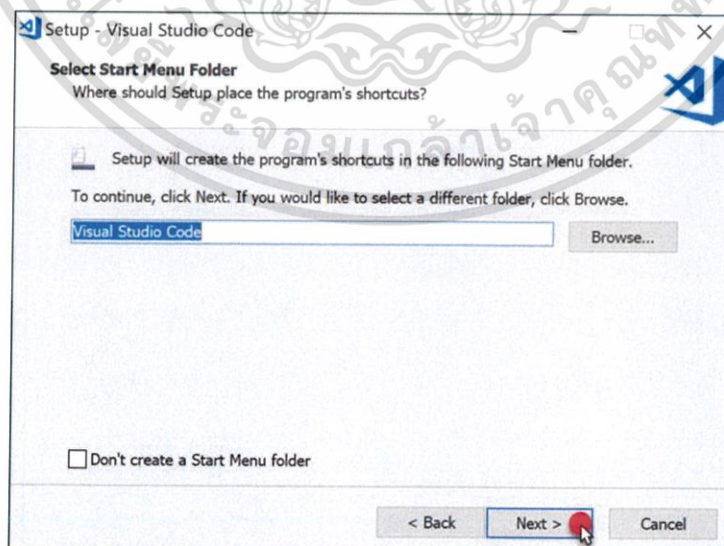
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ³⁷ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้นจะแสดงหน้าจอ Select Destination Location ให้กดปุ่ม Next >



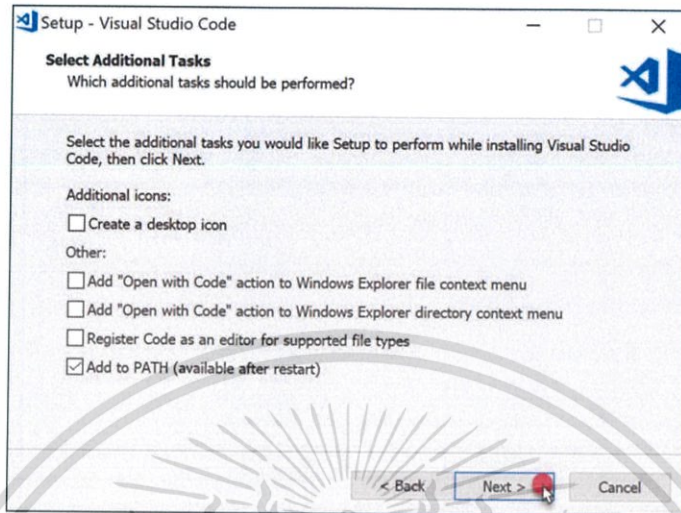
รูปที่ ข.5 ขั้นตอนการติดตั้งโปรแกรม

6. จากนั้นจะแสดงหน้าจอ Select Start Menu Folder ให้กดปุ่ม Next >



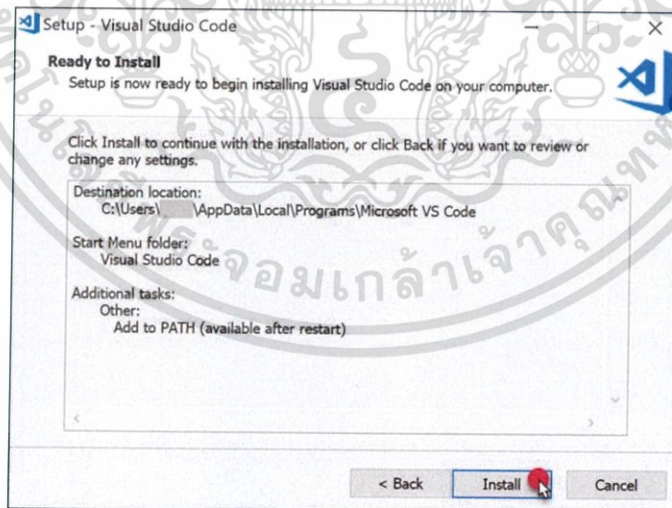
รูปที่ ข.6 ขั้นตอนการติดตั้งโปรแกรม

7. จากนั้นจะแสดงหน้าจอ Select Additional Tasks ให้กดปุ่ม Next >



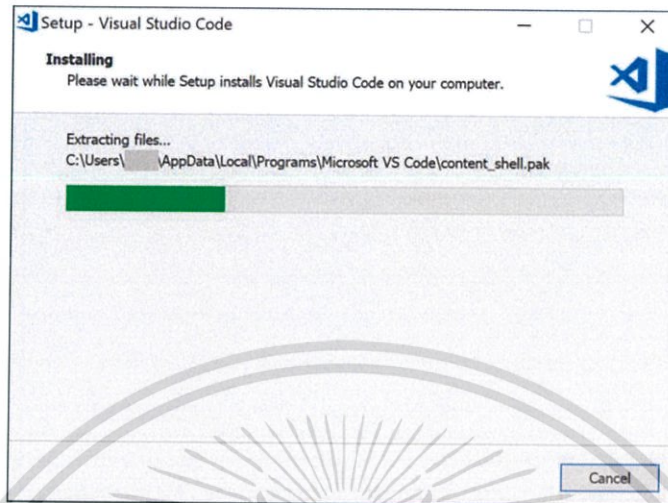
รูปที่ ข.7 ขั้นตอนการติดตั้งโปรแกรม

8. จากนั้นจะแสดงหน้าจอ Ready to Install ให้กดปุ่ม Install



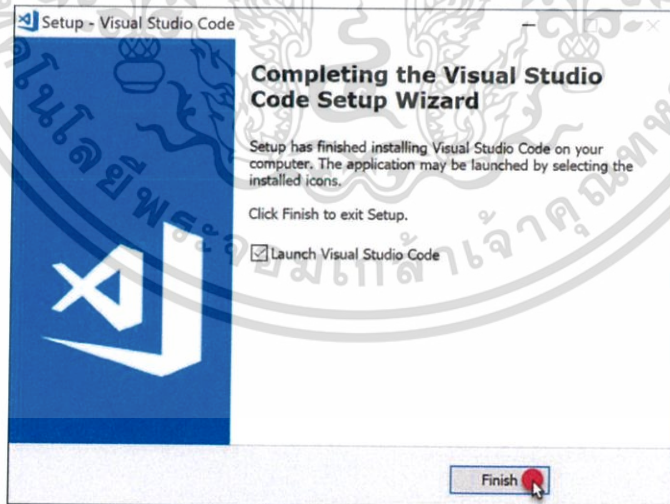
รูปที่ ข.8 ขั้นตอนการติดตั้งโปรแกรม

9. จากนั้นรอโปรแกรมติดตั้งสักครู่...



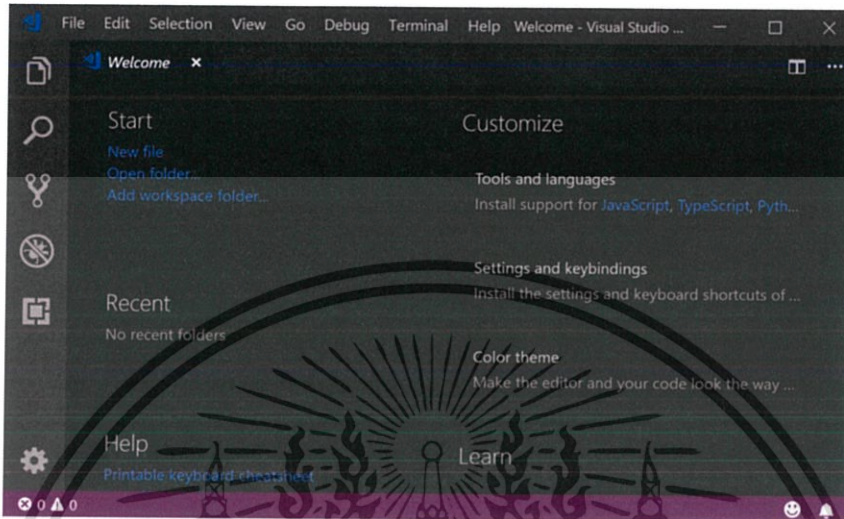
รูปที่ ข.9 ขั้นตอนการติดตั้งโปรแกรม

10. เมื่อติดตั้งเสร็จแล้วจะแสดงหน้าจอ Completing the Visual Studio Code Setup Wizard ให้กดปุ่ม Finish >



รูปที่ ข.10 ติดตั้งโปรแกรมเสร็จสมบูรณ์

11. จากนั้นโปรแกรม Visual Studio Code เปิดขึ้นมาก็เป็นการเสร็จสิ้นขั้นตอนในการติดตั้งโปรแกรม Visual Studio Code



รูปที่ ข.11 หน้าต่างโปรแกรม

ข.2 การติดตั้งโปรแกรม Node.js

การจะนำ application ที่เขียนด้วย Node.js มาใช้งาน เราจะต้องติดตั้ง Node.js ซึ่งสามารถดาวน์โหลดได้ที่ <https://nodejs.org/en> จะปรากฏหน้าต่างดังภาพ ให้ทำการดาวน์โหลดปุ่มทางด้านซ้ายเพื่อติดตั้งโปรแกรม

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)



[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

รูปที่ ข.12 หน้าต่างดาวน์โหลดโปรแกรมจาก Website

เมื่อทำการดาวน์โหลดเสร็จแล้วให้ทำการติดตั้งผ่านตัว installer เพื่อติดตั้งลงในคอมพิวเตอร์

Welcome to the Node.js Setup Wizard

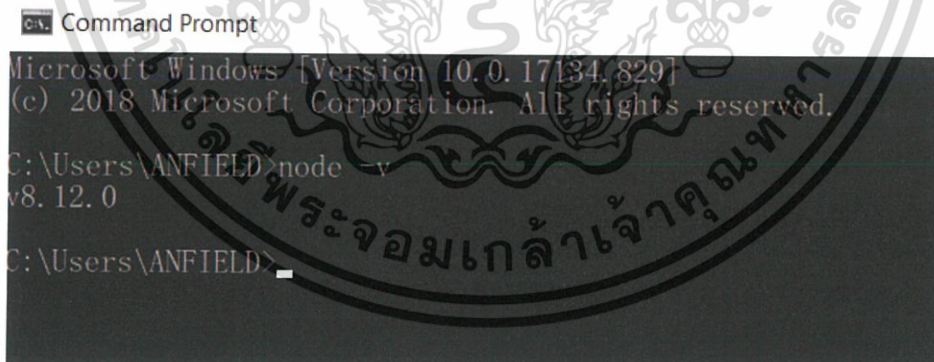


The Setup Wizard will install Node.js on your computer. Click Next to continue or Cancel to exit the Setup Wizard.



รูปที่ ข.13 ขั้นตอนการติดตั้งโปรแกรม

เมื่อติดตั้งเสร็จแล้ว ให้เราเปิด Command-line Interface ขึ้นมา โดยใช้ Command Prompt แล้วพิมพ์คำสั่ง “node -v” เพื่อเช็คว่าคุณได้ทำการติดตั้งเป็นที่เรียบร้อยแล้ว ในกรณีที่ทำการติดตั้งเป็นที่เรียบร้อยแล้ว จะทำการขึ้นข้อความ version ของ node.js ที่ทำการติดตั้ง



รูปที่ ข.14 ขั้นตอนการตรวจสอบว่าโปรแกรมได้ทำการติดตั้งเรียบร้อยแล้ว

ภาคผนวก ค

คำอธิบายส่วนประกอบของโปรแกรม

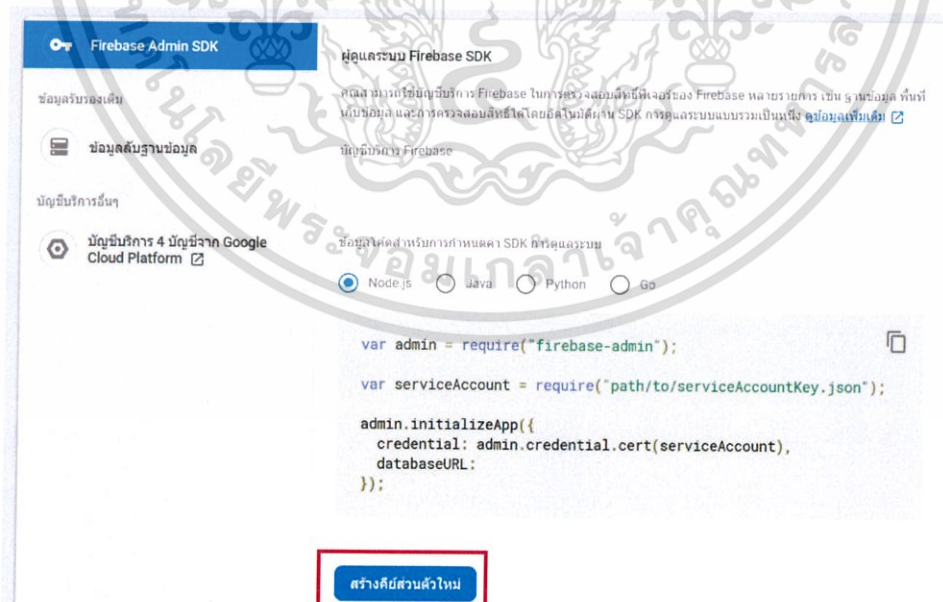
1. ระบบการส่งผลการประมวลผลและผลการแข่งขัน

ในการใช้งานโปรแกรมความสามารถในการส่งข้อมูลจากมือถือเข้าสู่ระบบจะต้องมีการตั้งค่าข้อมูลก่อนที่จะใช้งานซึ่งสามารถตั้งค่าได้ดังนี้

1.1 การสร้างไฟล์ Servicesaccountkey.json

ในการสร้างไฟล์ Servicesaccountkey.json เพื่อให้สามารถส่งข้อมูลที่ได้นั้นไปยัง Firebase Database ได้ ซึ่งจำเป็นมากต่อการส่งข้อมูล โดยในส่วนนี้ทาง Google ได้ทำลิงสำหรับโหลดไฟล์ให้แล้ว ซึ่งสามารถโหลดได้ที่ Link URL “console.firebase.google.com/u/0/project/ ‘ชื่อ Project ของคุณ’ /settings/serviceaccounts/adminsdk.”

ซึ่งจะสามารถกดโหลดตัวไฟล์ได้ตามภาพที่ ค.1



รูปที่ ค.1 ภาพในการโหลดไฟล์ Servicesaccountkey.json เพื่อใช้ในการเชื่อมต่อ

หลังจากโหลดตัว Servicesaccountkey.json มาแล้วซึ่งข้างในจะมีข้อมูลรวมไปถึงคีย์ลับในการใช้ยืนยันตัวตนและใช้ในการเชื่อมต่อโดยอัตโนมัติโดยสามารถนำไปใส่ในตัว source code เพื่ออ้างอิงได้ต่อไป

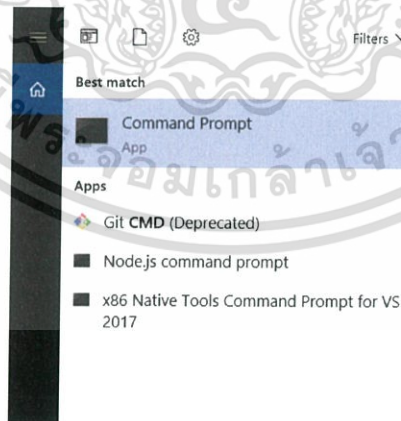
```
1 {  
2   "type":  
3   "project_id": "  
4   "private_key_id":  
5   "private_key": "--  
6   "client_email": "f  
7   "client_id":  
8   "auth_uri": "  
9   "token_uri":  
10  "auth_provider_x509_cert_url":  
11  "client_x509_cert_url":  
12 }
```

รูปที่ ค.2 ตัวอย่างข้อมูลที่อยู่ในไฟล์ Servicesaccountkey.json

1.2 การเปิดใช้งานPort node.js และการเข้าใช้งาน

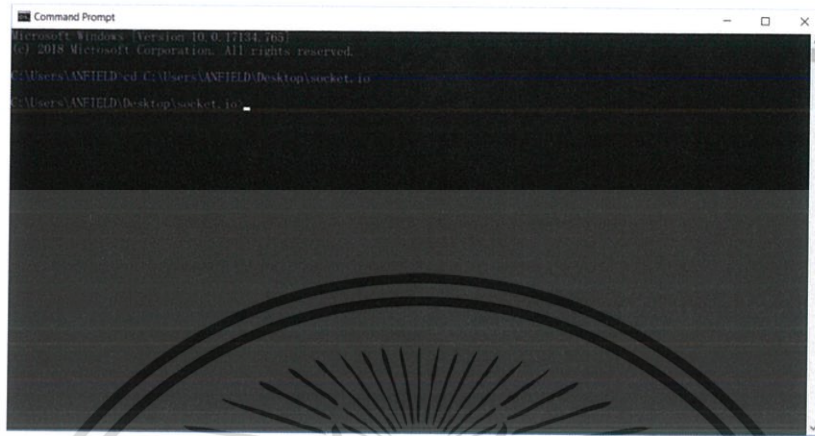
ในการเปิดใช้งาน Port node.js เพื่อเปิดให้ให้อุปกรณ์เข้าสู่ระบบของ socket.io ได้ซึ่งทำได้โดย

- 1) เปิด Command prompt (CMD) ภายในเครื่องคอมพิวเตอร์ที่จะเป็นตัวรับสัญญาณ โดยการเลือกค้นหา แล้วพิมพ์ "CMD" ในช่องค้นหา ก็จะสามารถเปิดใช้งาน Command prompt ได้



รูปที่ ค.3 การเปิดใช้งาน Command Prompt

- 2) หลังจากเปิดใช้งานแล้วให้พิมพ์คำสั่ง cd ตามด้วยชื่อ folder ที่เก็บไฟล์ไว้เพื่อที่จะชี้ Directory ของ Folder นั้นๆ (ตามภาพคือเก็บไว้ที่ “C:\Users\ANFIELD\Desktop\socket.io”)



รูปที่ ค.4 การชี้ Directory ของไฟล์ที่ใช้ในการเปิด

- 3) หลังจากนั้นให้เปิดใช้งาน node.js โดยการพิมพ์ “node app.js” ซึ่งจะเปิดการเปิดใช้งานไฟล์ “app.js” ที่อยู่ใน Folder socket.io และจะเปิดใช้งานใน port ที่กำหนดไว้ (ภาพตัวอย่างอยู่ที่ port “8081”)



รูปที่ ค.5 การบอกถึงชื่อไฟล์และชื่อ port ที่ทำการใช้งาน

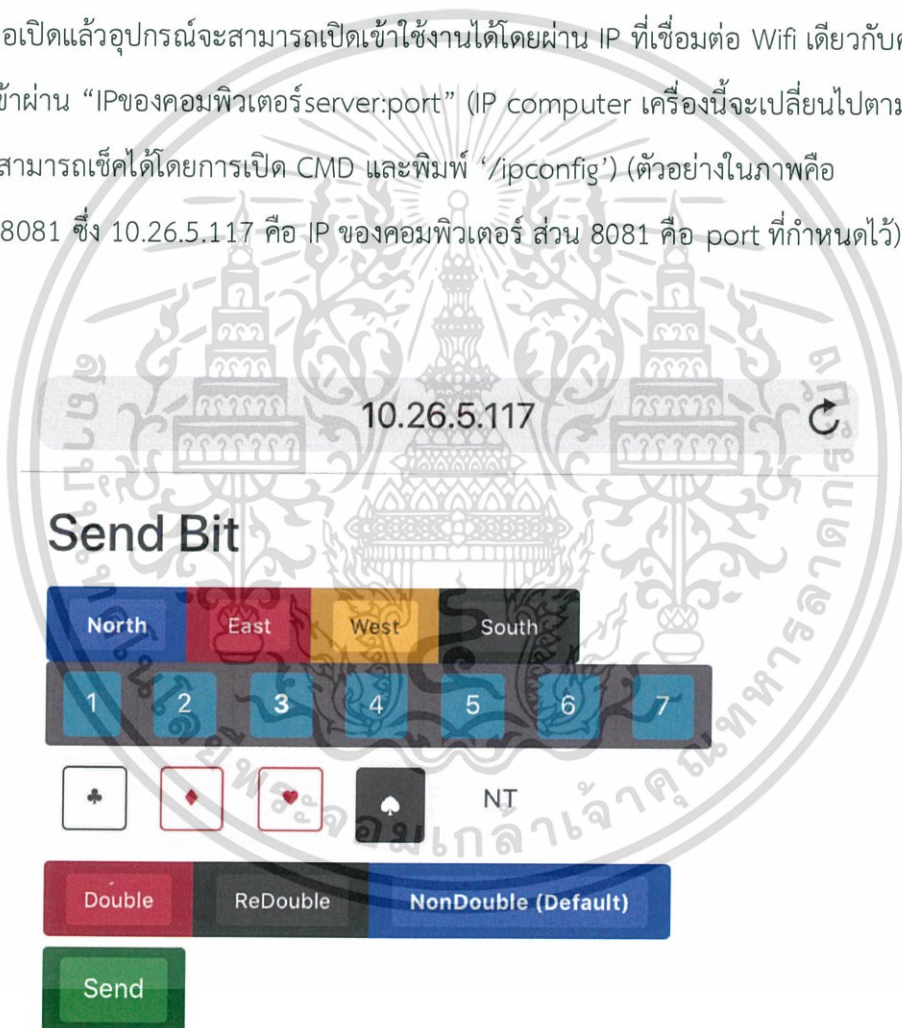

```
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ANFIELD>cd C:\Users\ANFIELD\Desktop\socket.io

C:\Users\ANFIELD\Desktop\socket.io>node app.js
Listening on port: 8081
```

รูปที่ ค.6 การเปิดใช้งาน node.js

ซึ่งเมื่อเปิดแล้วอุปกรณ์จะสามารถเปิดเข้าใช้งานได้โดยผ่าน IP ที่เชื่อมต่อ Wifi เดียวกับคอมพิวเตอร์เครื่องนี้และเข้าผ่าน “IPของคอมพิวเตอร์server:port” (IP computer เครื่องนี้จะเปลี่ยนไปตามเชื่อมต่อ Wifi แต่ละที่ สามารถเช็คได้โดยการเปิด CMD และพิมพ์ ‘ipconfig’) (ตัวอย่างในภาพคือ 10.26.5.117:8081 ซึ่ง 10.26.5.117 คือ IP ของคอมพิวเตอร์ ส่วน 8081 คือ port ที่กำหนดไว้)



รูปที่ ค.7 การเข้าใช้งานผ่านอุปกรณ์

1.3 การกำหนด IP ของอุปกรณ์ในการเชื่อมต่อ

เนื่องจากระบบการส่งผลคะแนนต้องมีการกำหนดอุปกรณ์ที่ใช้งานเนื่องจากป้องกันข้อมูลที่ไม่ถูกต้อง และข้อมูลจากอุปกรณ์ภายนอกทำให้ผลการแข่งขันเปลี่ยนไป ซึ่งการกำหนดอุปกรณ์จะต้องกำหนดที่หน้า source code .js ทุกครั้งก่อนการแข่งขัน โดยเช็คได้ดังนี้

- 1) ให้อุปกรณ์เชื่อมต่อกับ node.js ตามข้อที่ ข.2.2 หลังจากเชื่อมต่อแล้วจะแสดงผล IP ของอุปกรณ์มายังหน้า Command prompt ที่เปิดไว้ (จากภาพ IP ของอุปกรณ์ที่เชื่อมต่อคือ 10.20.1.148)



รูปที่ ค.8 หน้า Command prompt แสดงการเชื่อมต่อของอุปกรณ์

- 2) หลังจากนั้นให้ไปกำหนดที่หน้า .js ว่าจะให้อุปกรณ์นี้เข้าใช้งานในฐานะอุปกรณ์ของโต๊ะไหน โดยการใส่ค่า IP ไว้ที่ clientIp == "ค่า IP" และกำหนด data = "เบอร์โต๊ะของผู้เข้าแข่ง" ซึ่งจะขึ้นแสดงผลบนหน้าจออุปกรณ์หากมีการ setting เป็นที่เรียบร้อย ซึ่งจะต้องแจก พนักงานให้ทำการแก้ไขก่อนเริ่มการแข่งขัน

```
// กำหนด ip client ดังต่อไปนี้
if(clientIp == '::ffff:127.0.0.1'/'แก้ไข IP ตรงนี้' ) /'เมื่อ IP ตรงกับสามารถใช้งานได้' / {
  var data = "5"; // กำหนดเบอร์โต๊ะของผู้แข่งเครื่องนี้ได้
  io.to(socketId).emit('CreateUser', data);
}
if(clientIp == '::ffff:10.20.1.148'/'แก้ไข IP ตรงนี้' ) /'เมื่อ IP ตรงกับสามารถใช้งานได้' / {
  var data = "8"; // กำหนดเบอร์โต๊ะของผู้แข่งเครื่องนี้ได้
  io.to(socketId).emit('CreateUser', data);
}
else if(clientIp == ':::1'/'แก้ไข IP ตรงนี้' ) /'เมื่อ IP ตรงกับสามารถใช้งานได้' / {
  var data = "5"; // กำหนดเบอร์โต๊ะของผู้แข่งเครื่องนี้ได้
  io.to(socketId).emit('CreateUser', data);
}
else if (clientIp == '::ffff:10.111.3.186'/'แก้ไข IP ตรงนี้' ) /'เมื่อ IP ตรงกับสามารถใช้งานได้' / {
  var data = "1"; // กำหนดเบอร์โต๊ะของผู้แข่งเครื่องนี้ได้
  io.to(socketId).emit('CreateUser', data);
}
// ถ้าไม่ตรงตามที่กำหนด ให้อื่น Error และไม่สามารถใช้งานได้
else {
  var data = "error";
  io.to(socketId).emit('CreateUser', data);
}
```

รูปที่ ค.9 การตั้งค่า IP ของอุปกรณ์ทำให้สามารถใช้งานระบบได้

2. ระบบการจับคู่การแข่งขัน

เป็นการจับคู่ของผู้เข้าแข่งขันในแต่ละรอบการแข่งขันจนจบเกมส์การแข่งขัน

2.1 เป็นการ config firebase เข้ากับ visual studio เชื่อมต่อกันเพื่อไว้ใช้ในการเก็บข้อมูลของผู้เข้าแข่งขัน ที่มาแข่งขันในแต่ละรอบ

```
.html < finish.html JS config_firebase.js x < index.html
config_firebase.js > errData
// Initialize Firebase
var config = {
  apiKey: "AIzaSyDDLXnAkmz56IKBvXPuIUFwPAGqMUBN6-M",
  authDomain: "yuyuyu-9473a.firebaseio.com",
  databaseURL: "https://yuyuyu-9473a.firebaseio.com",
  projectId: "yuyuyu-9473a",
  storageBucket: "yuyuyu-9473a.appspot.com",
  messagingSenderId: "1087225218125"
};
firebase.initializeApp(config);
const db = firebase.firestore();
function errData(data) {
  console.log("Error!");
  console.log(err);
}
```

รูปที่ ค.10 Config firebase

2.2 เป็นส่วนของการ Input ข้อมูลผู้เข้าแข่งขันเข้าไปเก็บไว้ในฐานข้อมูล Cloud firestore

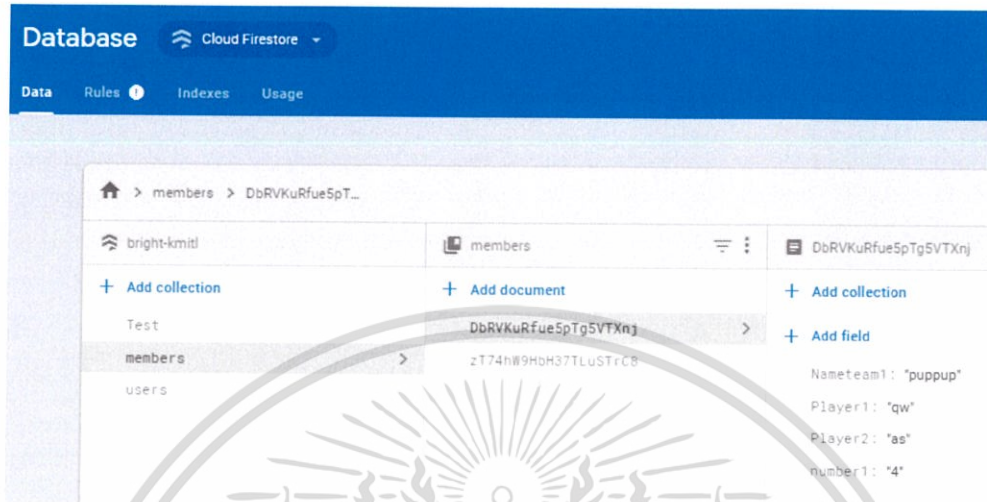
```
finish.html < finish.html JS config_firebase.js JS dbfirestore.js x JS
dbfirestore.js > ...
const datastorageList = document.querySelector('#cafe-list');
function renderCafe(doc) {

  let N_SE_W = document.createElement('N_SE_W');
  let Board_no = document.createElement('Board_no');
  let Table_no = document.createElement('Table_no');

  li.setAttribute('data-id', doc.id)
  name.textContent = doc.data().N_SE_W;
  name.textContent = doc.data().Board_no;
  type.textContent = doc.data().Table_no;
  li.appendChild(N_SE_W);
  li.appendChild(Board_no);
  li.appendChild(Table_no);
  datastorageList.appendChild(li);
}
db.collection('users').get().then((snapshot) => {
  snapshot.docs.forEach(doc => {
    renderCafe(doc);
  })
})
})
```

รูปที่ ค.11 เก็บข้อมูลที่ Input เข้าฐานข้อมูล

2.3 หน้าfirebase เมื่อเรา Input ข้อมูลเข้าไป ข้อมูลจะไปปรากฏในฐานข้อมูลของเรา



รูปที่ ค.12 Cloud Firestore

2.4)เป็นส่วนของการลบข้อมูลที่ input เข้าไปออกจากระบบฐานข้อมูล

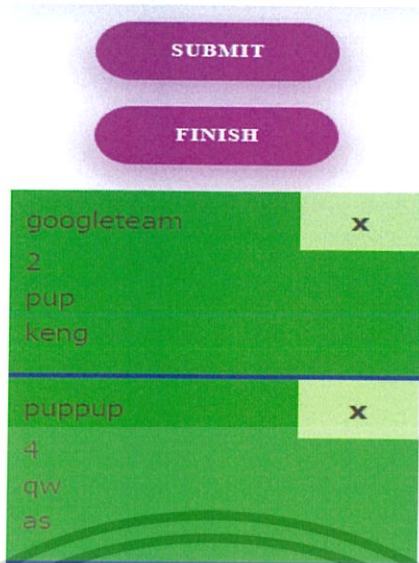
```

//deleting data
cross.addEventListener('click', (e) => {
  e.stopPropagation();
  let id = e.target.parentElement.getAttribute('data-id');
  db.collection('cafes').doc(id).delete();
});

//saving data
form.addEventListener('submit', (e) => {
  e.preventDefault();
  db.collection('cafes').add({
    name: form.name.value,
    city: form.city.value
  });
  form.name.value = '';
  form.city.value = '';
});

// real-time listener
db.collection('cafes').onSnapshot(snapshot => {
  let changes = snapshot.docChanges();
  changes.forEach(change => {
    if(change.type == 'added'){
      renderCafe(change.doc);
    } else if (change.type == 'removed'){
      let li = cafeList.querySelector('[data-id=' + change.doc.id + ']');
      cafeList.removeChild(li);
    }
  })
})
})
```

รูปที่ ค.13 Source code delete data



รูปที่ ค.14 Interface delete data

2.5 หน้าfinish ดูการจับคู่การแข่งขันใน1 รอบการแข่งขัน เป็นการโชว์ผลการจับคู่การแข่งขันในเกมส์การแข่งขัน

```

html
  finish.html
  JS config_firestore.js
  JS app.js
  JS app.js
  index.html
  finish.html
  <body>
  <div>
    <ul id="cafe-list"></ul>
  </div>
  <div id="left">
    <ul class="list-group" id="listgroup">
    </div>
    <div id="right">
    <ul class="list-group" id="groups">
    </div>
  </ul>
  <script>
  db.collection("Register")
    .onSnapshot(function(querySnapshot) {
      $("#listgroup").html("");
      querySnapshot.forEach(function(doc) {
        $("#listgroup").append('<li class="list-group-item">'+ 'User :'+ doc.data().Name +'<br>' + 'Number :'+ doc.data(
          + '</li>');
      });
    });

  db.collection("users")
    .onSnapshot(function(querySnapshot) {
      $("#groups").html("");
      querySnapshot.forEach(function(doc) {
        $("#groups").append('<li class="list-group-item">' + '<br>' + 'NS :'+ doc.data().N S+'EW :'+ doc.data().E_
          + '<br>' + 'Board No :'+doc.data().Board_no +'<br>' + 'Table No :'+ doc.data().Table_no +
          '<br>' + 'Round: '+doc.data().Round+ '<button type="submit" onclick=test()>Click Me!</button>' + '</li>');
      });
    });
  </script>
  </body>
  </html>
  
```

รูปที่ ค.15 เก็บข้อมูลตารางการแข่งขัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
html <> finish.html • JS config_firebase.js JS app2.js JS app.js <> index.html
ish.html > html > body > div.container-login100-form-btn > br

</script><div class="container-login100-form-btn">
  <button type="submit" class="login100-form-btn" onclick="sendData()">
    Submit
  </button><br/><br/><br/>
</div>
<div class="container-login100-form-btn">
  <a href="Finish.html"><button type="" class="login100-form-btn">
    Finish
  </button>
</a> <br/><br/>
</div>
<script src="app2.js"></script>
</body>
</html>
```

รูปที่ ค.16 เก็บข้อมูลตารางการแข่งขัน(ต่อ)

NS:10EW:2
Board No: 1
Table No : 2
Round:8Click Me!

NS:3EW:10
Board No: 3
Table No : 3
Round:1Click Me!

รูปที่ ค.17 Interface ตารางการแข่งขัน

ภาคผนวก ง.

รูปภาพวันโปรเจคเดย์



รูปที่ ง.1 วันโปรเจคเดย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.2 วันโปรเจคเดย์



รูปที่ ง.3 วันโปรเจคเดย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.4 วันโปรเจคเดย์



รูปที่ ง.5 วันโปรเจคเดย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ⁵⁴ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้