

การพัฒนาโมบายแอปพลิเคชันเพื่อเป็นสังคมออนไลน์สำหรับเกมเมอร์

Mobile Application Development of an Online Community for Gamers

นางสาว พัศราพร	เสื่อเหลือง	6104800005
นาย ณิชวีร์	ฉันท์แต่ง	6104800008

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2564

หัวข้อปริญญานิพนธ์

การพัฒนาโมบายแอปพลิเคชันเพื่อเป็นสังคมออนไลน์
สำหรับเกมเมอร์

Mobile Application Development of an Online
Community for Gamers

หน่วยกิตของปริญญานิพนธ์

3 หน่วยกิต

รายชื่อคณะผู้จัดทำ

นางสาวพัศราพร เสือเหลือง 6104800005

นายณัฐธีร์ นันทน์แต่ง 6104800008

อาจารย์ที่ปรึกษา

อาจารย์จรรยา แหยมเจริญ

ระดับการศึกษา

วิทยาศาสตรบัณฑิต

ภาควิชา

วิทยาการคอมพิวเตอร์

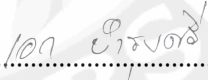
ปีการศึกษา


2564

อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบปริญญานิพนธ์


.....ประธานกรรมการ
(พล.อ.ท.ศ.ดร. พันธ์ธร สงวนโภคัย)


.....กรรมการ
(อาจารย์เอก บำรุงศรี)


.....อาจารย์ที่ปรึกษา
(อาจารย์จรรยา แหยมเจริญ)

หัวข้อปริญญานิพนธ์	การพัฒนาโมบายแอปพลิเคชันเพื่อเป็นสังคมออนไลน์สำหรับเกมเมอร์		
หน่วยกิตของปริญญานิพนธ์	3 หน่วยกิต		
รายชื่อคณะผู้จัดทำ	นางสาวพัศราพร	เสื่อเหลือง	6104800005
	นายณัฐธีร์	ฉันท์แต่ง	6104800008
อาจารย์ที่ปรึกษา	อาจารย์จรรยา	แหยมเจริญ	
ระดับการศึกษา	วิทยาศาสตรบัณฑิต		
ภาควิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2564		

บทคัดย่อ

วัตถุประสงค์ในการจัดทำปริญญานิพนธ์นี้เพื่อพัฒนาสื่อกลางออนไลน์สำหรับเกมเมอร์ในการค้นหาเพื่อนร่วมเล่นเกมที่มีความชอบเหมือนกันและแลกเปลี่ยนประสบการณ์ในการเล่นเกมน โดยพัฒนาเป็นโมบายแอปพลิเคชันแบบครอสแพลตฟอร์ม ทำงานได้ทั้งบนแพลตฟอร์มของแอนดรอยด์และไอโอเอส มีการประยุกต์ใช้เทคโนโลยีเครือข่ายสังคมออนไลน์ คลาวด์คอมพิวติ้ง และสถาปัตยกรรมฐานข้อมูล NoSQL ผู้ใช้งานเป็นได้ทั้งผู้สร้างปาร์ตี้เกมและผู้เข้าร่วมปาร์ตี้เกม ระบบสามารถแนะนำเกมตามความชื่นชอบแพลตฟอร์มของเกมตามที่ใช้กำหนด โดยสุ่มเลือกจำนวน 10 รายการ แนะนำเกมตามความนิยมโดยนับจากความถี่ของปาร์ตี้เกมนั้นสูงสุด และแนะนำเกมใหม่ การออกแบบส่วนติดต่อกับผู้ใช้ออกแบบตามหลักการของ UX และ UI และเครื่องมือในการพัฒนาระบบ ได้แก่ Flutter Framework และภาษา Dart จัดการฐานข้อมูลด้วย Firebase ผลจากการทดสอบระบบโดยคณะผู้จัดทำ ฟังก์ชันต่างๆ สามารถทำงานได้ตามที่กำหนด ข้อมูลมีความถูกต้อง

คำสำคัญ: สังคมออนไลน์, เกมเมอร์, เกมออนไลน์, ฐานข้อมูล โนเอสคิวแอล, ครอสแพลตฟอร์ม

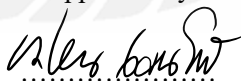
Project Title	Mobile Application Development of an Online Community for Gamers		
Project Credit	3 Units		
By	Miss Phatsaraporn Suealueang	6104800005	
	Mr. Nuttee Chanteang	6104800008	
Advisor	Miss Janya Yamcharoen		
Degree	Bachelor of Science		
Major	Computer Science		
Faculty	Science		
Academic year	2021		

Abstract

The objective of this project was to develop online media for gamers to seek out like-minded people and exchange online experiences. Using the cross-platform mobile application that works on both Android and iOS platforms, the application utilizes social media, and cloud computing. The database architecture of this application was NoSQL, in which the users can be both the party host and attendant. The application, therefore, can recommend games that fit the profile of each player. The user interface was designed according to UX/UI principles. Flutter Framework and Dart language were the developmental tools. After the implementation, the application works according to the specification and the data is accurate.

Keywords: social media, gamers, online game, NoSQL Database, cross-platform

Approved by


.....

Approved by


.....

กิตติกรรมประกาศ (Acknowledgment)

การจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ผู้จัดทำได้รับความกรุณาจากอาจารย์ผู้สอนทุกท่านที่ได้รับความกรุณาชี้แนะและช่วยเหลือในทุกๆ ด้าน เพื่อจัดทำวิทยานิพนธ์นี้ ส่งผลให้คณะผู้จัดทำได้รับความรู้และประสบการณ์ต่างๆ ที่มีค่ามากมาย สำหรับวิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนดังนี้

1. อาจารย์ จรรยา แหยมเจริญ อาจารย์ที่ปรึกษา

คณะผู้จัดทำขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ ที่ได้ให้คำแนะนำสำคัญในการสอบวิทยานิพนธ์ฉบับนี้ และผู้ที่มีส่วนร่วมทุกท่านรวมทั้งบุคคลอื่นๆ ที่ไม่ได้กล่าวนามไว้ที่มีส่วนร่วมในการให้ข้อมูล และให้ความช่วยเหลือในด้านต่างๆ และเป็นທີ່ปรึกษาให้คำแนะนำต่างๆ จนทำให้งานทุกอย่างประสบความสำเร็จไปได้ด้วยดี และจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ซึ่งคณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

นางสาวพัศราพร เสือเหลือง

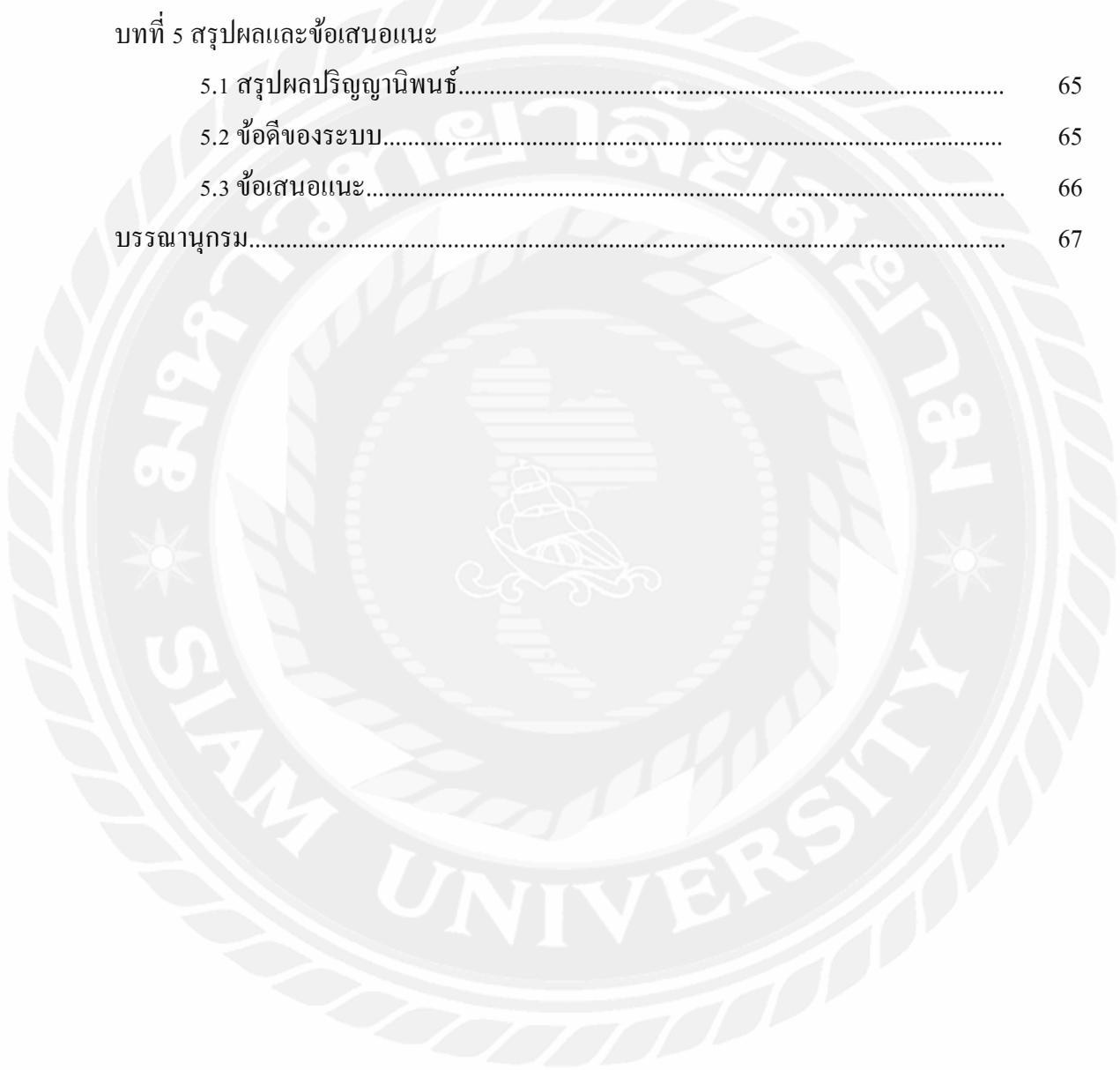
นายณัฐธีร์ ฉันท์แดง

สารบัญ

	หน้า
บทคัดย่อ.....	ก
Abstract.....	ข
กิตติกรรมประกาศ.....	ค
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์.....	2
1.3 ขอบเขตปริญญาานิพนธ์.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 ขั้นตอนและวิธีการดำเนินงานปริญญาานิพนธ์.....	4
1.6 แผนและระยะเวลาในการดำเนินงานปริญญาานิพนธ์.....	6
1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนา.....	6
1.8 อุปกรณ์และเครื่องมือที่รองรับระบบ.....	7
บทที่ 2 การทบทวนวรรณกรรมที่เกี่ยวข้อง	
2.1 เกมออนไลน์ (Game Online).....	8
2.2 การพัฒนาแอปพลิเคชันแบบครอสแพลตฟอร์ม (Cross Platform Application)	8
2.3 ระบบฐานข้อมูล (NoSQL).....	9
2.4 Development Tools.....	10
2.5 Firebase.....	11
บทที่ 3 การวิเคราะห์ระบบ	
3.1 รายละเอียดของปริญญาานิพนธ์.....	13
3.2 แสดงฟังก์ชันการทำงานของระบบด้วย Use Case Diagram.....	14
3.3 คำอธิบายรายละเอียดของยูสเคส Use Case Description.....	15
3.4 แสดงขั้นตอนการทำงานของฟังก์ชันด้วย Sequence Diagram.....	22
3.5 แผนภาพแสดงองค์ประกอบคลาส (Class Diagram).....	28
3.6 รายละเอียดของคลาส.....	29
3.7 โครงสร้างของข้อมูล (Data Structure).....	42
3.8 รายละเอียดของโครงสร้างข้อมูล.....	43

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การออกแบบทางกายภาพ	
4.1 โครงสร้างของแอปพลิเคชัน (Application Structure Map).....	46
4.2 ระบบแนะนำกิจกรรมตามความชอบของแต่ละบุคคล.....	49
4.3 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface Design).....	49
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 สรุปผลปริญาานิพนธ์.....	65
5.2 ข้อดีของระบบ.....	65
5.3 ข้อเสนอแนะ.....	66
บรรณานุกรม.....	67

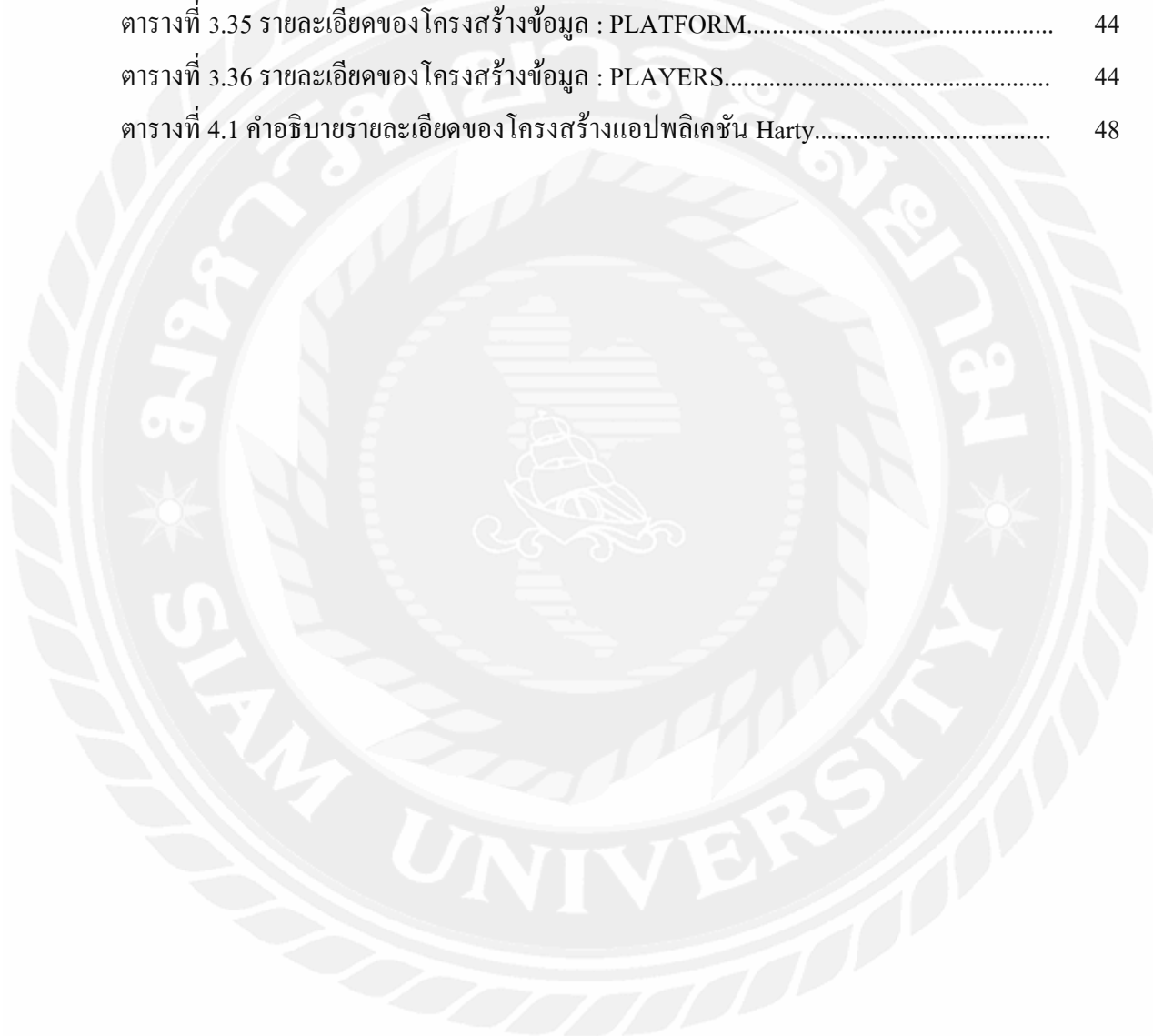


สารบัญตาราง

	หน้า
ตารางที่ 1.1 ระยะเวลาในการดำเนินงานปริญญาโท.....	6
ตารางที่ 3.1 แสดงรายละเอียดของ Use case Login.....	14
ตารางที่ 3.2 แสดงรายละเอียดของ Use case Register.....	15
ตารางที่ 3.3 แสดงรายละเอียดของ Use case Edit Profile.....	16
ตารางที่ 3.4 แสดงรายละเอียดของ Use case Search Game.....	16
ตารางที่ 3.5 แสดงรายละเอียดของ Use case Recommend Game.....	17
ตารางที่ 3.6 แสดงรายละเอียดของ Use case Create Party.....	18
ตารางที่ 3.7 แสดงรายละเอียดของ Use case Remove Player.....	18
ตารางที่ 3.8 แสดงรายละเอียดของ Use case Choose Join Party.....	19
ตารางที่ 3.9 แสดงรายละเอียดของ Use case Leave Party.....	19
ตารางที่ 3.10 แสดงรายละเอียดของ Use case Delete Party.....	20
ตารางที่ 3.11 แสดงรายละเอียดของ Use case Chat.....	20
ตารางที่ 3.12 รายละเอียดของ Class Diagram Detail : PartyList.....	28
ตารางที่ 3.13 รายละเอียดของ Class Diagram Detail : Profile.....	28
ตารางที่ 3.14 รายละเอียดของ Class Diagram Detail : editProfile.....	29
ตารางที่ 3.15 รายละเอียดของ Class Diagram Detail : CreateParty.....	30
ตารางที่ 3.16 รายละเอียดของ Class Diagram Detail : Platform.....	31
ตารางที่ 3.17 รายละเอียดของ Class Diagram Detail : Follow.....	31
ตารางที่ 3.18 รายละเอียดของ Class Diagram Detail : login.....	31
ตารางที่ 3.19 รายละเอียดของ Class Diagram Detail : register.....	32
ตารางที่ 3.20 รายละเอียดของ Class Diagram Detail : joinParty.....	33
ตารางที่ 3.21 รายละเอียดของ Class Diagram Detail : HomeScreen.....	34
ตารางที่ 3.22 รายละเอียดของ Class Diagram Detail : Game Lobby.....	36
ตารางที่ 3.23 รายละเอียดของ Class Diagram Detail : Chat List.....	37
ตารางที่ 3.24 รายละเอียดของ Class Diagram Detail : Chat.....	38
ตารางที่ 3.25 รายละเอียดของ Class Diagram Detail : SendNoti.....	39
ตารางที่ 3.26 รายละเอียดของ Class Diagram Detail : NotificationList.....	39
ตารางที่ 3.27 รายละเอียดของ Class Diagram Detail : AuthService.....	40
ตารางที่ 3.28 รายละเอียดของโครงสร้างข้อมูล : CHAT.....	42
ตารางที่ 3.29 รายละเอียดของโครงสร้างข้อมูล : CONVERSATION.....	42

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 3.30 รายละเอียดของโครงสร้างข้อมูล : FOLLOWING.....	42
ตารางที่ 3.31 รายละเอียดของโครงสร้างข้อมูล : GAMES.....	43
ตารางที่ 3.32 รายละเอียดของโครงสร้างข้อมูล : NOTIFICATION.....	43
ตารางที่ 3.33 รายละเอียดของโครงสร้างข้อมูล : PARTY.....	43
ตารางที่ 3.34 รายละเอียดของโครงสร้างข้อมูล : PARTY_DETAIL.....	44
ตารางที่ 3.35 รายละเอียดของโครงสร้างข้อมูล : PLATFORM.....	44
ตารางที่ 3.36 รายละเอียดของโครงสร้างข้อมูล : PLAYERS.....	44
ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Harty.....	48



สารบัญรูปภาพ

	หน้า
รูปที่ 3.1 Use Case Diagram ของแอปพลิเคชัน Harty.....	13
รูปที่ 3.2 Sequence diagram : Login.....	21
รูปที่ 3.3 Sequence diagram : Register.....	21
รูปที่ 3.4 Sequence diagram : Edit Profile.....	22
รูปที่ 3.5 Sequence diagram : Search Game.....	22
รูปที่ 3.6 Sequence diagram : Recommend Game.....	23
รูปที่ 3.7 Sequence diagram : Create Party.....	23
รูปที่ 3.8 Sequence diagram : Remove Player.....	24
รูปที่ 3.9 Sequence diagram : Join Party.....	24
รูปที่ 3.10 Sequence diagram : Leave Party.....	25
รูปที่ 3.11 Sequence diagram : Delete Party.....	25
รูปที่ 3.12 Sequence diagram : Chat.....	26
รูปที่ 3.13 Class Diagram ของแอปพลิเคชัน Harty.....	27
รูปที่ 3.14 โครงสร้างของข้อมูล (NoSQL Database Structure).....	41
รูปที่ 4.1 โครงสร้างของแอปพลิเคชัน Harty.....	47
รูปที่ 4.2 ปุ่มก่อนของแอปพลิเคชัน Harty.....	51
รูปที่ 4.3 หน้า Slash Screen.....	51
รูปที่ 4.4 หน้าเข้าสู่ระบบ (Login).....	52
รูปที่ 4.5 หน้าตั้งค่าโปรไฟล์ผู้ใช้งานใหม่.....	53
รูปที่ 4.6 หน้าตั้งค่ารูปโปรไฟล์.....	54
รูปที่ 4.7 หน้า Home Feed แสดงหน้าจอหลักของแอปพลิเคชัน.....	55
รูปที่ 4.8 หน้า Profile.....	56
รูปที่ 4.9 หน้า Game Lobby.....	57
รูปที่ 4.10 หน้า Game Room.....	58
รูปที่ 4.11 หน้า Party's Detail.....	59
รูปที่ 4.12 หน้า Notification.....	60
รูปที่ 4.13 หน้า Chat List.....	61
รูปที่ 4.14 หน้า Chat.....	62
รูปที่ 4.15 หน้า Settings.....	63
รูปที่ 4.16 หน้า Theme Setting.....	64

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.17 หน้า Edit Profile.....

65



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

เกมออนไลน์ หมายถึง เกมที่ต้องเล่นผ่านระบบอินเทอร์เน็ต (internet) เช่น League of Legends, ROV, Valorant เป็นต้น ปัจจุบันเกมออนไลน์ได้รับความนิยมจากผู้เล่นทั่วโลกโดยประเทศไทยมีผู้เล่นเกมออนไลน์ประมาณ 32 ล้านคน ภายในสิ้นปี 2565 มีจำนวนผู้ชมกีฬาอีสปอร์ตจำนวน 4.3 ล้านคน มีอัตราการเติบโตทางด้านเศรษฐกิจที่เพิ่มมากขึ้นเมื่อเทียบกับต่างประเทศ (มติชนออนไลน์, 2565) และเกมออนไลน์จัดเป็นกีฬาอีสปอร์ต (E-Sport) สามารถช่วยคลายเครียดและสร้างรายได้ ส่วนใหญ่เกมออนไลน์เป็นระบบผู้เล่นหลายคน (Multiplayer) และมีการพัฒนาบนหลากหลายแพลตฟอร์มให้ผู้เล่นได้เลือกเล่นได้ตามที่ชอบและถนัด เช่น คอมพิวเตอร์ส่วนตัว, โทรศัพท์มือถือ, หรือเครื่องเล่นเกมประเภทต่างๆ ทั้งในบ้านและนอกบ้าน จึงทำให้ปัจจุบันมีผู้คนหันมาเล่นเกมเป็นจำนวนมาก และธุรกิจเกมก็กำลังเติบโตอย่างก้าวกระโดด

แต่เนื่องจากเกมออนไลน์ส่วนใหญ่ในปัจจุบันจะนิยมเล่นเป็นทีม หรือระบบผู้เล่นหลายคน และในทุกๆ เกมมีระบบสุ่มผู้เล่นมาให้ในทีม ทำให้ผู้เล่นต้องไปร่วมทีมกับคนอื่นที่ยังไม่เคยพูดคุยหรือพบเจอเป็นเพื่อนกันในชีวิตจริง ส่งผลให้ผู้เล่นที่ต้องการเล่นเกมๆ นั้นอาจจะพบกับทีมที่ไม่ตรงตามความต้องการหรือพบผู้เล่นที่ไม่ให้ความสนุกในการเล่นเป็นทีม (Teamwork) ละทิ้งทีมในเกม ทำให้การเล่นเกมนั้นไม่มีประสิทธิภาพ ความสนุกลดลง และนำไปสู่ความพ่ายแพ้ในเกมๆ นั้น ซึ่งอาจส่งผลให้ผู้เล่นเกิดความรู้สึกไม่อยากเล่นเกมที่ต้องเล่นเป็นทีมกับผู้อื่น แต่สำหรับผู้เล่นบางคนอาจจะไม่มีเพื่อนในชีวิตจริงที่เล่นเกมเดียวกันหรือมีช่วงเวลาในการเล่นไม่ตรงกัน หรืออาจจะมีความชอบแนวเกมคนละประเภท หรือมีสมาชิกในทีมไม่ครบหรือไม่พร้อมก่อนการเริ่มแข่งขันในเกม ผู้เล่นจึงอาจจะไปใช้แพลตฟอร์มที่เป็นโซเชียลต่างๆ ที่มีอยู่ในปัจจุบัน เช่น Facebook, Discord หรือสังคมออนไลน์ (Online Community) อื่นๆ แต่สังคมออนไลน์เหล่านั้นก็ไม่ได้พัฒนาขึ้นมาเพื่อตอบโจทย์ในการหาเพื่อนเล่นเกมที่ตรงกับความต้องการ และยังมีความซับซ้อนในการใช้งานและไม่ตอบโจทย์ผู้เล่นเกมโดยตรง

จากปัญหาที่กล่าวมาข้างต้นคณะผู้จัดทำจึงได้ทำการศึกษาและพัฒนาโมบายแอปพลิเคชัน “ฮार्टี้ (Harty)” สำหรับเป็นเครื่องมือและสื่อกลางในการช่วยค้นหาผู้เล่นเกมที่มีความชอบเกมประเภทเดียวกันหรือมีช่วงเวลาในการเล่นเกมที่ตรงกัน และสามารถสื่อสารกันได้อย่างมีประสิทธิภาพในทีมเพื่อสร้างสังคมผู้เล่นเกมที่มีคุณภาพ และได้พบปะกับผู้เล่นใหม่ๆ ที่มีความชื่นชอบในเกมประเภทเดียวกันมาร่วมทีมเพื่อเพิ่มความสนุกและเพิ่มประสิทธิภาพในการเล่นเกม โดยแอปพลิเคชันที่พัฒนานี้เป็นโมบายแอปพลิเคชันที่ทำงานข้ามแพลตฟอร์ม (Cross-Platform Mobile

Application) สามารถใช้งานได้ทั้งบนแพลตฟอร์มของแอนดรอยด์ (Android) และไอโอเอส (iOS) พัฒนาด้วยภาษา Dart และ Flutter Framework จัดเก็บข้อมูลด้วยสถาปัตยกรรมฐานข้อมูล NoSQL ของ Firebase

1.2 วัตถุประสงค์ของปริญญาานิพนธ์

เพื่อพัฒนาโมบายแอปพลิเคชันสำหรับเป็นสื่อกลางสำหรับเกมเมอร์ ในการค้นหาผู้เล่นเกม และเป็นสังคมออนไลน์สำหรับผู้ที่ชื่นชอบการเล่นเกมออนไลน์

1.3 ขอบเขตของปริญญาานิพนธ์

1.3.1 พัฒนาเป็นโมบายแอปพลิเคชันที่สามารถทำงานข้ามแพลตฟอร์ม (Cross-Platform Mobile Application)

1.3.2 ประยุกต์ใช้สถาปัตยกรรมฐานข้อมูล NoSQL ในการบริหารจัดการข้อมูลของระบบ

1.3.3 ฟังก์ชันการทำงานของระบบ ประกอบด้วย

1.3.3.1 การยืนยันตัวตนเพื่อเข้าใช้งานระบบ (Authentication) ผู้ใช้สามารถลงทะเบียนเพื่อเข้าใช้ระบบได้โดยผ่านบัญชีของ Facebook และ Google ระบบจะทำการดึงข้อมูลอีเมลล์มาเพื่อเป็นการลงชื่อเข้าใช้งานแอปพลิเคชันแล้วทำการป้อนข้อมูลของผู้ใช้ เช่น ชื่อผู้ใช้ เวลาในการสะดวกที่จะเล่นเกม และแพลตฟอร์มของเกมที่ใช้เล่น เป็นต้น

1.3.3.2 การสร้างปาร์ตี้เพื่อค้นหาผู้ร่วมเล่นเกม

1.3.2.2.1 ผู้ใช้สามารถเลือกเกมที่ต้องการสร้างปาร์ตี้ได้

1.3.2.2.2 ผู้ใช้สามารถเข้าร่วมปาร์ตี้ที่มีอยู่แล้วได้

1.3.2.2.3 ผู้ใช้สามารถตรวจสอบปาร์ตี้ที่มีผู้เล่นอยู่แล้วได้

1.3.2.2.4 ผู้ใช้สามารถสร้างปาร์ตี้ใหม่ได้

1.3.2.2.5 ผู้ใช้ที่สร้างปาร์ตี้สามารถลบผู้เล่นคนอื่นออกจากปาร์ตี้ของตนได้

1.3.2.2.6 ผู้ใช้ที่สร้างปาร์ตี้สามารถลบปาร์ตี้ที่ตนสร้างได้

1.3.2.2.7 ผู้ใช้สามารถส่งข้อความส่วนตัวไปยังผู้เล่นคนอื่นได้

1.3.3.3 การติดตามผู้เล่น

1.3.2.3.1 ผู้ใช้สามารถติดตามผู้เล่นที่ตนสนใจได้

1.3.2.3.2 ผู้ใช้สามารถยกเลิกการติดตามผู้เล่นได้

1.3.3.4 ระบบการแนะนำเกม

- 1.3.2.4.1 โดยระบบจะแนะนำเกมตามแพลตฟอร์มของเกมที่ใช้แต่ละคนกำหนดไว้ได้ ดังนั้นผู้ใช้แต่ละคนจะได้รับการแนะนำเกมที่แตกต่างกัน (Personalized Recommendation)
- 1.3.2.4.2 เกมที่กำลังเป็นที่นิยม (Popularity) โดยดูจากเกมที่มีจำนวนผู้สร้างปาร์ตี้เป็นจำนวนมาก
- 1.3.2.4.2 เกมใหม่บนแพลตฟอร์มต่างๆ
- 1.3.3.5 การตั้งค่า
 - 1.3.2.5.1 ผู้ใช้สามารถตั้งค่าธีมสีของแอปพลิเคชัน โดยมีให้เลือก 5 ธีมสี
 - 1.3.2.5.2 ผู้ใช้สามารถออกจากระบบและเปลี่ยนบัญชีผู้ใช้ได้
 - 1.3.2.5.3 ผู้ใช้สามารถลบบัญชีผู้ใช้ของตนเองได้
- 1.3.4 แหล่งข้อมูลที่น่ามาใช้และแหล่งจัดเก็บข้อมูล

ระบบได้ข้อมูลรายชื่อเกมและรูปภาพของเกมมาจากร้านค้าเกมออนไลน์ บนแพลตฟอร์มต่างๆ ได้แก่

 - Steam รายชื่อและรูปภาพของเกมบนแพลตฟอร์มสำหรับคอมพิวเตอร์ส่วนบุคคล (PC)
 - Appstore, Playstore รายชื่อและรูปภาพของเกมบนแพลตฟอร์มมือถือ (Mobile)
 - Nintendo Store รายชื่อและรูปภาพของเกมบนแพลตฟอร์มสำหรับเครื่องนินเทนโด
 - Playstation Store รายชื่อและรูปภาพของเกมบนแพลตฟอร์มสำหรับเครื่องเพลย์สเตชัน
 - Xbox Game Store รายชื่อและรูปภาพของเกมบนแพลตฟอร์มสำหรับเครื่องเอ็กซ์บ็อกซ์
 - ฐานข้อมูลผู้ใช้ได้มาจากการล็อกอินด้วยบัญชีรายชื่อจาก Google และ Facebook เพื่อจัดเก็บอีเมลล์และรหัสผ่านในการยืนยันตัวตนสำหรับการเข้าสู่ระบบ และการจัดเก็บข้อมูลส่วนบุคคลของผู้ใช้ เช่น วันที่เกิด, อายุ, เพศ, ช่วงเวลาที่สะดวกในการเล่นเกม, แพลตฟอร์มของเกมที่ใช้เล่น เป็นต้น โดยที่ข้อมูลต่างๆ จะถูกจัดเก็บลงบน Firebase

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 เป็นสังคมออนไลน์สำหรับเกมเมอร์อีกแหล่งหนึ่ง
- 1.4.2 เพิ่มช่องทางในการสร้างมิตรภาพใหม่ๆ สำหรับในยุคปัจจุบันแบบ New Normal
- 1.4.3 เกิดการแลกเปลี่ยนข้อมูล ความรู้ ทักษะ และการร่วมเล่นเกม
- 1.4.4 ผู้ใช้มีโอกาสในการพบเกมใหม่ๆ และสังคมคนเล่นเกมที่จะเป็นประสบการณ์ใหม่

1.4.5 พัฒนาสังคมของเกม ความมีน้ำใจ และการแบ่งปันในหมู่ผู้เล่นเกม

1.5 ขั้นตอนและวิธีการดำเนินงานปริญญานิพนธ์

1.5.1 การรวบรวมความต้องการและการศึกษาข้อมูล (Detailed Study)

รวบรวมความต้องการโดยมีการประชุมร่วมกันระหว่างคณะผู้จัดทำ และอาจารย์ที่ปรึกษา และกลุ่มเป้าหมายที่เป็นนักเล่นเกมหรือเกมเมอร์ ซึ่งได้ทำการสืบค้นและศึกษาข้อมูลรวบรวมปัญหาต่างๆ เพื่อนำมาปรับใช้กับระบบ และคณะผู้จัดทำได้ทำการศึกษาเกี่ยวกับเทคโนโลยี เครื่องมือที่จะนำมาประยุกต์ใช้ในการสร้างแอปพลิเคชัน ได้แก่ การพัฒนาโปรแกรมแบบข้ามแพลตฟอร์ม (Cross-Platform Development) โดยเขียนชุดคำสั่งเพียงครั้งเดียวแต่สามารถทำงานได้หลากหลายอุปกรณ์หรือแพลตฟอร์ม สถาปัตยกรรมฐานข้อมูลที่สามารถรองรับการทำงานแบบออนไลน์และรองรับขนาดของข้อมูลที่ขยายได้ในอนาคต รวมถึงศึกษาระบบแนะนำส่วนบุคคล (Recommendation System) เพื่อนำมาประยุกต์ใช้ในการแนะนำเกมให้ตรงใจผู้เล่น

1.5.2 การวิเคราะห์ระบบ (System Analysis)

นำข้อมูลต่างๆ ที่ได้รวบรวมมาทำการวิเคราะห์และวางแผนการปฏิบัติงาน เพื่อกำหนดขอบเขตของระบบและฟังก์ชันการทำงาน โดยนำเสนอผ่านแผนภาพไดอะแกรมต่างๆ ได้แก่ Use Case Diagram แสดงภาพรวมฟังก์ชันการทำงานทั้งหมด Sequence Diagram แสดงขั้นตอนการทำงานของฟังก์ชัน Class Diagram แสดงองค์ประกอบคลาส และ Data Structure Tree แสดงโครงสร้างของฐานข้อมูล NoSQL

วิเคราะห์อัลกอริทึมที่จะนำมาใช้ในระบบแนะนำตามความชอบของแต่ละบุคคล โดยได้กำหนดให้ผู้ใช้เลือกแพลตฟอร์มของเกมที่เล่นตั้งแต่ขั้นตอนการลงทะเบียนเข้าใช้งานครั้งแรก ซึ่งจะใช้ความสามารถของภาษา SQL ในการดึงข้อมูลเกมที่มีแพลตฟอร์มตรงกับที่ผู้ใช้กำหนดจากฐานข้อมูล การนับจำนวนปาร์ตี้ของเกมที่มีจำนวนการสร้างสูงสุดเพื่อแนะนำเกมยอดนิยม และการแนะนำเกมใหม่ล่าสุดที่ได้ข้อมูลแหล่งข้อมูลต่างๆ

1.5.3 การออกแบบระบบ (System Design)

ออกแบบระบบตามที่ได้วิเคราะห์ไว้ในขั้นตอนก่อนหน้า โดย

1.5.3.1 ออกแบบสถาปัตยกรรมที่ใช้ในการพัฒนาโครงการเป็นแบบ Cross-Platform Mobile Application

โดยเป็นแอปพลิเคชันที่ทำงานบนอุปกรณ์เคลื่อนที่ เช่น โทรศัพท์มือถือ หรือ แท็บเล็ต และเขียนชุดคำสั่งเพียงครั้งเดียวแต่สามารถทำงานได้ทั้งบนระบบปฏิบัติการแอนดรอยด์

(Android) และ ไอโอเอส (iOS) โดยเครื่องมือและภาษาที่ใช้ในการพัฒนา ได้แก่ Flutter Framework และภาษา Dart พัฒนาซอฟต์แวร์ตามหลักการของ MVC (Model View Controller)

1.5.3.2 ออกแบบโครงสร้างของข้อมูล (Data Structure Design)

เพื่อให้ง่ายต่อการแลกเปลี่ยนข้อมูลข้ามระบบหรือข้ามแพลตฟอร์ม ได้ทำการออกแบบโครงสร้างข้อมูลตามหลักการของ NoSQL Database ที่ทำงานบนคลาวด์ โดยเลือกใช้โปรแกรม Firebase

1.5.3.3 ออกแบบส่วนติดต่อกับผู้ใช้งาน (User Interface Design)

การออกแบบหน้าจอที่ใช้ในการติดต่อกับผู้ใช้ (User Interface (UI)) มุ่งเน้นออกแบบให้ผู้ใช้ใช้งานง่ายและสามารถใช้งานได้จริง โดยคำนึงถึง User Experience (UX) เป็นหลัก เพื่อให้ผู้ใช้ใช้งานได้ง่าย สามารถใช้งานได้ด้วยตนเอง และไม่ว่าผู้ใช้จะใช้แอปพลิเคชันบนแพลตฟอร์มใดก็จะเห็นหน้าจอเป็นแบบเดียวกัน

1.5.3.4 ออกแบบอัลกอริทึมในการแนะนำเกมตามความชอบของแต่ละบุคคล

ในการแนะนำเกมตามความชอบส่วนบุคคล จะยึดตามแพลตฟอร์มที่ผู้ใช้กำหนดเป็นหลักในการแนะนำเกม โดยใช้ภาษา SQL ในการค้นหาข้อมูลเกมที่มีแพลตฟอร์มตรงกับที่ผู้ใช้กำหนดไว้ตั้งแต่การลงทะเบียนใช้งานระบบในครั้งแรก โดยข้อมูลที่แนะนำแต่ละครั้งจะไม่ซ้ำกัน เนื่องจากจะดึงข้อมูลใหม่ล่าสุดเสมอ นอกจากนี้ยังใช้จำนวนความถี่ (Frequency) ของเกมที่มีการสร้างปาร์ตี้มากที่สุด เพื่อแนะนำเกมใหม่ๆ แก่ผู้ใช้

1.5.4 การพัฒนาระบบ (System Development)

นำข้อมูลที่ได้ทำการวิเคราะห์และออกแบบไว้มาเขียนเป็นชุดคำสั่งด้วย ภาษา Dart และ Flutter Framework เขียนชุดคำสั่งด้วยโปรแกรม Microsoft Visual Studio Code และเรียกใช้ API ต่างๆ ของ Flutter ทั้งในการสร้างส่วนติดต่อกับผู้ใช้ หรือ UI และการสร้างฟังก์ชันต่างๆ รวมถึงการติดต่อกับฐานข้อมูล Firebase และระบบการแนะนำตามแนะนำเกมตามความชอบส่วนบุคคล

1.5.5 การทดสอบระบบ (System Testing)

ในการทดสอบระบบคณะผู้จัดทำได้ทำการทดสอบพร้อมกับขั้นตอนของการเขียนชุดคำสั่ง โดยทำการทดสอบการทำงานฟังก์ชันย่อย (Unit Testing) โดยถ้าฟังก์ชันไม่สามารถทำงานได้ตามที่กำหนดจะทำการปรับแก้ทันที และทดสอบการทำงานร่วมกันระหว่างฟังก์ชันแต่ละฟังก์ชัน (Integration Testing) ว่าสามารถทำงานร่วมกัน สามารถส่งข้อความ (Message) ถึงกันได้ ถูกต้องหรือไม่ ถ้าพบข้อผิดพลาดจะทำการปรับแก้ทันที และได้ทำการทดสอบระบบทั้งหมด (System Testing) เมื่อพัฒนาทุกฟังก์ชันเสร็จแล้ว เพื่อทดสอบการทำงานของระบบทั้งหมดว่าสามารถทำงานร่วมกันและเป็นไปตามหน้าที่ที่กำหนดหรือไม่

1.5.6 การจัดทำเอกสารประกอบปริญญาโท (Documentation)

เป็นการจัดทำเอกสารเพื่อนำเสนอรายละเอียดของปริญญาโทสำหรับเป็นเอกสารอ้างอิง และแนวทางในการศึกษาต่อไป

1.6 แผนและระยะเวลาในการดำเนินงานปริญญาโท

ตารางที่ 1.1 ระยะเวลาในการดำเนินงานปริญญาโท

กิจกรรม	2564					2565				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. รวบรวมความต้องการและศึกษาข้อมูล	←→									
2. วิเคราะห์ระบบ		←→								
3. ออกแบบระบบ			←→							
4. พัฒนาระบบ			←→							
5. ทดสอบระบบ							←→			
6. จัดทำเอกสารประกอบปริญญาโท							←→			

1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนา

1.7.1 ฮาร์ดแวร์ (Hardware)

1.7.1.1 เครื่องคอมพิวเตอร์ Acer Nitro 5

1.7.1.2 โทรศัพท์มือถือ Android

1.7.2 ซอฟต์แวร์ (Software)

1.7.2.1 ระบบปฏิบัติการ Microsoft Windows 11

1.7.2.2 โปรแกรม Visual Studio Code

1.7.2.3 Flutter Framework

1.7.2.4 Firebase API

1.7.2.5 Version Control

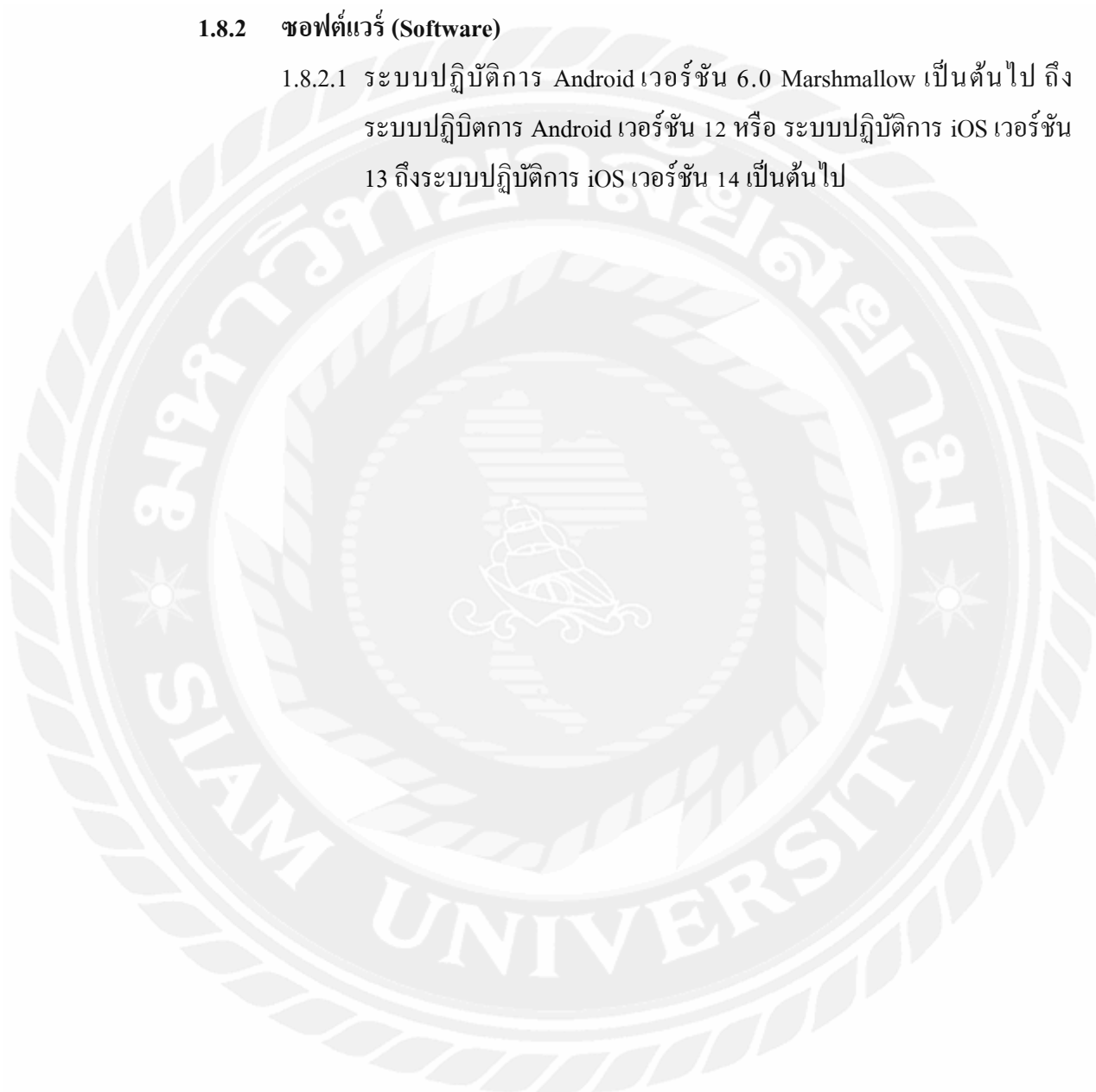
1.8 อุปกรณ์และเครื่องมือที่รองรับระบบ

1.8.1 ฮาร์ดแวร์ (Hardware)

1.8.1.1 โทรศัพท์มือถือหรือแท็บเล็ตที่สามารถเชื่อมต่อเครือข่ายอินเทอร์เน็ตได้

1.8.2 ซอฟต์แวร์ (Software)

1.8.2.1 ระบบปฏิบัติการ Android เวอร์ชัน 6.0 Marshmallow เป็นต้นไป ถึงระบบปฏิบัติการ Android เวอร์ชัน 12 หรือระบบปฏิบัติการ iOS เวอร์ชัน 13 ถึงระบบปฏิบัติการ iOS เวอร์ชัน 14 เป็นต้นไป



บทที่ 2

การทบทวนวรรณกรรมที่เกี่ยวข้อง

การพัฒนาปริญญาบัตรนี้ คณะผู้จัดทำได้ทำการศึกษาข้อมูล แนวคิด และทฤษฎีที่เกี่ยวข้อง เพื่อเป็นกรอบแนวทางในการพัฒนาระบบ ประกอบด้วย

2.1 เกมออนไลน์(Game Online)¹

เกมออนไลน์ (Online Game) หมายถึง วิดีโอเกมที่เล่นบนเครือข่ายคอมพิวเตอร์ โดยเฉพาะบนอินเทอร์เน็ต เกมออนไลน์มีส่วนที่คล้ายคลึงกับเกมหลายผู้เล่นผ่านระบบอินเทอร์เน็ต (Multiplayer) เกมออนไลน์ส่วนมากจะเป็นเกมสแบบ MMO (Massive Multiplayer Online) หรือก็คือเกมสหลายผู้เล่นที่รับจำนวนผู้เล่นได้มหาศาลในพื้นที่ๆ หนึ่ง เกมออนไลน์ได้รับความนิยมมากเนื่องจาก ผู้เล่นได้เข้าสังคมจึงรู้สึกสนุกที่จะมีเพื่อนเล่นเกมสไปด้วยกันมากกว่าการเล่นเกมสคนเดียว และยังมีกราฟิกที่สวยงามมาก จึงเป็นสิ่งที่ดึงดูดให้คนหันมาเล่นเกมออนไลน์ มีกิจกรรมต่างๆ รวมทั้งมีการเพิ่มแผนที่ในเกม อาวุธ เครื่องแต่งกาย มอนสเตอร์ และอื่นๆ

2.2 การพัฒนาแอปพลิเคชันแบบครอสแพลตฟอร์ม (Cross-Platform Application)²

ครอสแพลตฟอร์ม (Cross-Platforms) คือการพัฒนาระบบเพียงครั้งเดียว แต่สามารถนำไปติดตั้งในหลายแพลตฟอร์มหรือหลายอุปกรณ์ได้ ซึ่งครอสแพลตฟอร์มสามารถเรียกได้อีกอย่างว่า Hybrid Mobile Application เช่น ซอฟต์แวร์สามารถทำงานได้ในระบบวินโดวส์ (Windows Platform) บนสถาปัตยกรรม x86 และ Mac OS X บน PowerPC Platforms เป็นต้น

2.2.1 Hybrid Application

Hybrid Application มีกระบวนการพัฒนาเช่นเดียวกับเว็บแอปพลิเคชัน ใช้ภาษา HTML CSS และ JavaScript ในการพัฒนา ร่วมกับคอมโพเนนท์ที่ทำให้แอปพลิเคชันสามารถเรียกใช้ความสามารถของฮาร์ดแวร์และระบบปฏิบัติการของอุปกรณ์ได้ โดยในปัจจุบันมี Framework ในการพัฒนาแอปพลิเคชัน Hybrid Application อยู่เป็นจำนวนมาก เช่น PhoneGap, Ionic, Framwork7 หรือ React Native เป็นต้น

¹ <http://wananurat02472.blogspot.com/2013/11/game-online.html>

² <https://imd.co.th/category/blog/>

ข้อดีของ Hybrid Application

- พัฒนาด้วยภาษา HTML CSS และ JavaScript ทำให้ง่ายและเรียนรู้ได้อย่างรวดเร็ว
- พัฒนาครั้งเดียวสามารถใช้ได้หลายแพลตฟอร์มทั้ง iOS และ Android
- ใช้ต้นทุนในการพัฒนาน้อยกว่า Native Application

ข้อด้อยของ Hybrid Application

- ประสิทธิภาพการทำงานจะด้อยกว่าเนทีฟแอปพลิเคชัน (Native Application)
- บางกรณีใช้ความสามารถของอุปกรณ์ได้ไม่เต็มที่ เนื่องจากขึ้นอยู่กับเฟรมเวิร์ค (Framework) ที่เลือกใช้ในการพัฒนานั้นมีคอมโพเนนต์ที่ต้องการหรือไม่

2.3 ระบบฐานข้อมูลแบบ NoSQL³

ระบบฐานข้อมูลแบบ NoSQL เป็นระบบที่สร้างตามวัตถุประสงค์สำหรับโมเดลข้อมูลแบบเฉพาะเจาะจงและมีแบบแผนที่ยืดหยุ่นสำหรับการสร้างแอปพลิเคชันที่ทันสมัย ระบบฐานข้อมูลแบบ NoSQL เป็นระบบที่รู้จักกันในด้านความง่ายต่อการพัฒนา การทำงาน และ ประสิทธิภาพตามขนาดที่ต้องการ

ระบบฐานข้อมูลแบบ NoSQL เหมาะสมสำหรับแอปพลิเคชันสมัยใหม่ เช่น อุปกรณ์เคลื่อนที่ เว็บ และเกมที่ต้องมีฐานข้อมูลที่ยืดหยุ่น สามารถปรับขนาดได้ มีประสิทธิภาพสูง และทำงานได้ดีมีประสิทธิภาพ โดยระบบฐานข้อมูลแบบ NoSQL มีจุดเด่น ดังนี้

- ความยืดหยุ่น โดยทั่วไป ระบบฐานข้อมูลแบบ NoSQL จะมีแบบแผนยืดหยุ่นที่ทำให้การพัฒนาทำได้รวดเร็วและสามารถทำซ้ำคำสั่งได้มีประสิทธิภาพกว่าเดิม โดยโมเดลข้อมูลที่ยืดหยุ่นทำให้ฐานข้อมูล NoSQL เหมาะสมที่สุด สำหรับข้อมูลแบบกึ่งมีโครงสร้างและไม่มีโครงสร้าง
- ความสามารถในการปรับขนาด โดยทั่วไป ระบบฐานข้อมูลแบบ NoSQL ถูกออกแบบมาให้ปรับขนาดออกได้โดยใช้คลัสเตอร์แบบกระจายของฮาร์ดแวร์แทนการปรับขนาดขึ้น โดยเพิ่มเซิร์ฟเวอร์ที่มีราคาแพงและมีประสิทธิภาพสูง
- มีประสิทธิภาพสูง ระบบฐานข้อมูลแบบ NoSQL ได้รับการปรับปรุงประสิทธิภาพสำหรับโมเดลข้อมูลบางโมเดล และเข้าถึงรูปแบบที่เปิดใช้งานประสิทธิภาพที่สูงกว่า
- มี API รองรับ ระบบฐานข้อมูลแบบ NoSQL มี API การทำงานและประเภทข้อมูลที่สร้างตามวัตถุประสงค์สำหรับโมเดลข้อมูลแต่ละ โมเดลที่สอดคล้องกัน

³ <https://aws.amazon.com/th/nosql/>

ข้อดีของ NoSQL

- ระบบฐานข้อมูลแบบ NoSQL โดยถูกออกแบบมาให้มีความสามารถที่สูงมาก และขนาดระบบเพื่อรองรับผู้ใช้งานจำนวนมากได้ง่าย
- ระบบฐานข้อมูลแบบ NoSQL ในหลายๆ ระบบมักจะถูกออกแบบมาสำหรับ Unstructured Data โดยเฉพาะ

ข้อเสียของ NoSQL

- ระบบฐานข้อมูลแบบ NoSQL จะทำงานแบบ Non-transactional ดังนั้นถ้าข้อมูลมีความละเอียดสูงและไม่สามารถผิดพลาดได้ ระบบฐานข้อมูลนี้ก็อาจจะไม่เหมาะในหลายๆกรณี
- การเรียกอ่านข้อมูลขึ้นมาใช้ส่วนใหญ่มักจะมีต้นทุนที่สูงกว่าการใช้ SQL เพราะจะไม่สามารถเลือกเจาะจงได้อย่างง่าย แต่การบันทึกข้อมูลลงไปส่วนใหญ่จะง่ายกว่า SQL
- เทคโนโลยีส่วนใหญ่ไม่มีความเป็นมาตรฐานกลาง ดังนั้นการเปรียบเทียบแต่ละเทคโนโลยีค่อนข้างทำได้ยาก โดยผู้ใช้งานจะต้องมีความคุ้นเคยกับการจัดการซอฟต์แวร์เหล่านี้ให้ได้ด้วยตัวเอง
- ผู้เชี่ยวชาญที่สามารถสนับสนุนเทคโนโลยีเหล่านี้ในระดับองค์กรได้นั้นยังมีไม่มากนัก แต่เทคโนโลยี NoSQL นี้กลับมีความจำเป็นมากในการที่องค์กรจะสร้างความแตกต่างในเชิงเทคโนโลยีจากคู่แข่ง

2.4 Development Tools⁴

2.5.1 Flutter

Flutter เป็น Framework ที่ใช้สร้าง UI (User Interface) สำหรับ Mobile Application ที่สามารถทำงานข้ามแพลตฟอร์มได้ทั้ง iOS และ Android ในเวลาเดียวกัน โดยภาษาที่ใช้ใน Flutter นั้นก็คือภาษา Dart ซึ่งเป็นภาษาที่ถูกพัฒนาโดย Google และที่สำคัญภาษา Dart นี้เป็น Open Source ที่สามารถใช้งานได้ฟรี

จุดเด่นของ Flutter

- ระบบ Hot Reload โดยเมื่อมีการสร้างพีเจอร์หรือการกระทำต่างๆ กับ UI จะต้องมีการ Reload เพื่อให้หน้า UI Update ซึ่งระบบ Hot Reload จะช่วยย่นระยะเวลาในการ Reload ให้เหลือเพียงเสี้ยววินาทีเท่านั้น ทำให้การพัฒนา UI ของแอปพลิเคชัน มีความรวดเร็วขึ้นอย่างมาก
- Build-In ที่ช่วยในการออกแบบ UI ให้มีความสวยงามยิ่งขึ้นอย่าง Material Design และ Cupertino (iOS-flavor) มี Framework ที่ช่วยให้การทำแอนิเมชันต่างๆ หรือ Gesture

⁴ <https://medium.com/@hizokaz/มาทำความรู้จักกับ-flutter-กันเถอะ>

ของ UI เป็นเรื่องง่ายยิ่งขึ้น และยังสามารถใช้งานร่วมกับ IDE ที่กำลังเป็นที่นิยมในปัจจุบันอย่าง Visual Studio Code และ Android Studio ได้

ข้อเสียของ Flutter

- การใช้ภาษา Dart ในการพัฒนา ซึ่งผู้พัฒนาส่วนใหญ่อาจจะยังไม่คุ้นเคยกับรูปแบบของคำสั่งของภาษานี้ เมื่อเกิดปัญหาเกี่ยวกับการใช้งานอาจจะทำให้ผู้พัฒนาต้องใช้เวลาในการแก้ไข

2.6 Firebase ⁵

Firebase เป็นหนึ่งในผลิตภัณฑ์ของ Google โดย Firebase คือ แพลตฟอร์มที่รวบรวมเครื่องมือต่างๆ สำหรับการจัดการในส่วน Backend หรือ Service side ซึ่งทำให้สามารถ สร้าง Mobile Application ได้อย่างมีประสิทธิภาพ และยังคงเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูล โดยมีบริการต่างๆ ดังนี้

- Cloud Firestore สามารถจัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้และอุปกรณ์ในระดับโลก โดยใช้ฐานข้อมูล NoSQL ที่โฮสต์บนคลาวด์ Cloud Firestore ให้การซิงโครไนซ์แบบเรียลไทม์และการสนับสนุนออฟไลน์พร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ การผสมผสานร่วมกับผลิตภัณฑ์ Firebase อื่นๆ จะช่วยให้การสร้างแอปพลิเคชันแบบไร้เซิร์ฟเวอร์ได้อย่างแท้จริง
- Authentication การจัดการผู้ใช้ของคุณด้วยวิธีที่ง่ายและปลอดภัย Firebase Auth มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึงอีเมลและรหัสผ่านผู้ให้บริการบุคคลที่สามเช่น Google หรือ Facebook และใช้ระบบบัญชีที่มีอยู่ได้โดยตรง สามารถสร้างอินเทอร์เฟซหรือใช้ประโยชน์จากโอเพ่นซอร์ส UI ที่ปรับแต่งได้
- Hosting ช่วยลดความซับซ้อนของเว็บโฮสติ้งของคุณด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปพลิเคชันที่ทันสมัย เมื่ออัปเดตเนื้อหาของเว็บจะส่งเนื้อหาของเว็บเหล่านั้นไปยัง CDN ทั่วโลกของเราโดยอัตโนมัติและมอบใบรับรอง SSL เพื่อให้ผู้ใช้ได้รับประสบการณ์ที่ปลอดภัยเชื่อถือได้
- Realtime Database คือฐานข้อมูลดั้งเดิมของ Firebase เป็นโซลูชันที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ ที่ต้องการสถานะการซิงค์ระหว่างไคลเอนต์แบบเรียลไทม์
- Cloud Messaging สามารถส่งข้อความและการแจ้งเตือนไปยังผู้เข้าชมแพลตฟอร์มทั้ง Android, iOS และเว็บได้ สามารถส่งข้อความไปยังอุปกรณ์เดี่ยว กลุ่มอุปกรณ์ หรือ

⁵ <https://www.4xtreme.com/2020/11/20/firebase-คืออะไร/>

หัวข้อเฉพาะหรือกลุ่มผู้ใช้ Firebase Cloud Messaging (FCM) สามารถปรับขนาดเป็นแอปพลิเคชันที่มีขนาดใหญ่ได้ สามารถส่งข้อความหลายแสนล้านข้อความต่อวันได้



บทที่ 3

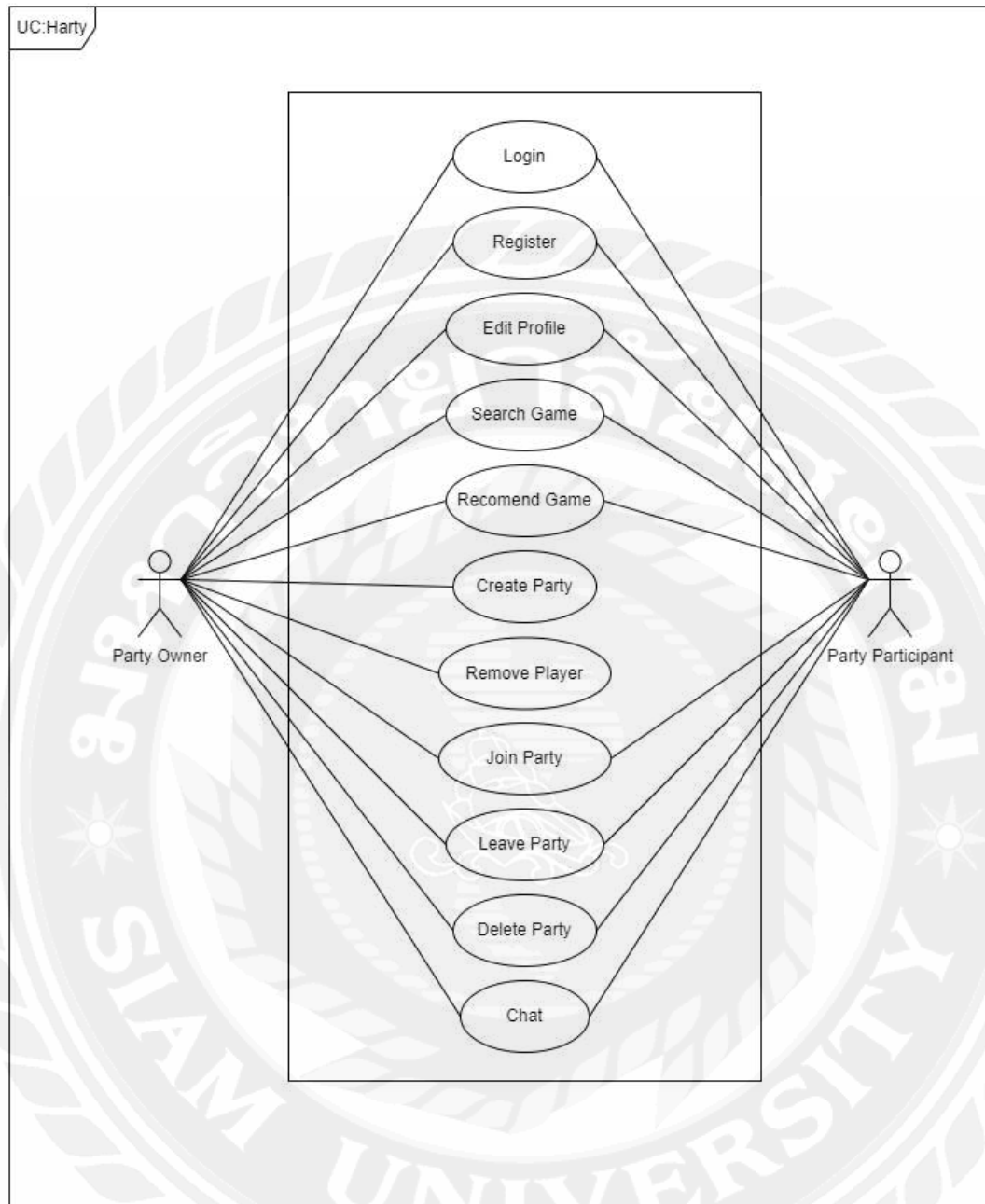
การวิเคราะห์ระบบ

3.1 รายละเอียดของปฏิยานิพนธ์

คณะผู้จัดทำได้พัฒนาโมบายแอปพลิเคชัน Harty (ฮาตี้) เป็นแอปพลิเคชันสื่อสังคมออนไลน์สำหรับเกมเมอร์ โดยสามารถค้นหาเพื่อนที่ชอบเล่นเกมเดียวกัน แอปพลิเคชันจะทำหน้าที่เป็นสื่อกลางให้ผู้ใช้ได้พบปะ พูดคุยผ่านแชท และแลกเปลี่ยนประสบการณ์ระหว่างผู้ใช้ โดยผู้ใช้งานจะต้องลงทะเบียนใช้งานแอปพลิเคชันเพื่อสร้างบัญชีผู้ใช้ และผู้ใช้แต่ละคนสามารถเป็นได้ทั้งผู้สร้างปาร์ตี้เกมและผู้เข้าร่วมปาร์ตี้เกมที่ผู้ใช้คนอื่นสร้าง สามารถติดตามผู้ใช้คนอื่นๆ รวมถึงกดถูกใจปาร์ตี้เกม และจะได้รับแนะนำเกมที่ตรงกับแพลตฟอร์มที่ชอบเล่นจากระบบได้

ดังนั้นในการพัฒนาระบบ จะประกอบด้วยผู้ใช้งานจะต้องลงทะเบียนเพื่อเข้าใช้งานก่อนเป็นลำดับแรกเมื่อเข้าสู่แอปพลิเคชันในครั้งแรก โดยผู้ใช้สามารถลงทะเบียนผ่านบัญชีของ Google หรือ Facebook เพื่อขอใช้ข้อมูลอีเมล (Email Address) และในครั้งแรกของการเข้าใช้งานแอปพลิเคชันผู้ใช้จะต้องทำการเลือกแพลตฟอร์มของเกมที่ตนเองชอบหรือเล่นเป็นประจำ เพื่อใช้เป็นตัวแปรในการแนะนำเกมตามความชอบส่วนบุคคล (Personalized Recommendation) โดยระบบจะแสดงเกมที่ตรงกับแพลตฟอร์มที่เลือกไว้ด้วยวิธีการสุ่ม (Random) จำนวน 10 รายการ และแนะนำเกมที่ได้รับคามนิยม (Popularity) โดยนับจากจำนวนปาร์ตี้ของเกมนั้นๆ เกมใดที่มีจำนวนการสร้างปาร์ตี้มากจะถูกแนะนำไปยังผู้ใช้ของระบบ เมื่อลงทะเบียนเรียบร้อยแล้วผู้ใช้จะสามารถเข้าใช้งานฟังก์ชันต่างๆ ได้ แอปพลิเคชันพัฒนาโดยใช้ Flutter Framework ในการสร้าง Native Interface สำหรับแอนดรอยด์และไอโอเอส ข้อมูลของผู้ใช้งานจะถูกจัดเก็บลงใน Firebase และเขียนชุดคำสั่งด้วยภาษา Dart เครื่องมือที่ใช้เขียนชุดคำสั่งคือ MS Visual Studio Code

3.2 แสดงฟังก์ชันการทำงานของระบบด้วย Use Case Diagram



3.3 รูปที่ 3.1 Use Case Diagram ของแอปพลิเคชัน Harty

3.4 คำอธิบายรายละเอียดของยูสเคส (Use Case Description)

ตารางที่ 3.1 แสดงรายละเอียดของ Use Case Login

Use Case Name	LOGIN
Use Case ID	UC1
Brief Description	สำหรับยืนยันตัวตน โดยใช้อีเมลและรหัสผ่าน เพื่อเข้าใช้งานแอปพลิเคชัน Harty
Primary Actors	Party Owner และ Party Participant
Secondary Actors	-
Preconditions	จะต้องทำการลงทะเบียนก่อนเข้าใช้งานระบบ
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้เข้าสู่ระบบแอปพลิเคชัน 2. ระบบแสดง Field สำหรับป้อนข้อมูลเพื่อเข้าสู่ระบบ โดยใช้อีเมลและรหัสผ่าน <ol style="list-style-type: none"> 2.1 ถ้าผู้ใช้ป้อนข้อมูล ชื่อผู้ใช้ และ รหัสผ่าน ถูกต้อง <ol style="list-style-type: none"> 2.1.1 ระบบจะแสดงหน้าหลักของแอปพลิเคชัน 2.2 ถ้าผู้ใช้ป้อนข้อมูล ชื่อผู้ใช้ และ รหัสผ่าน ไม่ถูกต้อง <ol style="list-style-type: none"> 3.2.1 ระบบจะแสดง ข้อความ ไม่สามารถเข้าสู่ระบบได้ 3.2.2 ผู้ใช้ต้องป้อนข้อมูล เพื่อเข้าสู่ระบบอีกครั้ง
Post Condition	เมื่อเข้าสู่ระบบสำเร็จ จะสามารถใช้งานฟังก์ชันต่างๆ ได้
Alternative Flows	-

ตารางที่ 3.2 แสดงรายละเอียดของ Use Case Register

Use Case Name	REGISTER
Use Case ID	UC2
Brief Description	สำหรับลงทะเบียน เพื่อเข้าใช้งานแอปพลิเคชัน Harty
Primary Actors	Party Owner และ Party Participant
Secondary Actors	-
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้ทำการลงทะเบียนข้อมูลผู้ใช้ 2. ผู้ใช้สามารถเลือกลงทะเบียนผู้ใช้ได้ผ่านทางบัญชีผู้ใช้จาก Google หรือ บัญชีผู้ใช้จาก Facebook ในการลงทะเบียนใช้งานแอปพลิเคชัน 3. ผู้ใช้งานป้อนข้อมูลสำหรับลงทะเบียนผู้ใช้งาน ดังนี้ ชื่อผู้ใช้งาน, อายุ, แพลตฟอร์มที่เล่น, ช่วงเวลาที่เล่น, เพศ, ข้อมูลการติดต่อของแพลตฟอร์มต่างๆ <ol style="list-style-type: none"> 3.1 ถ้าผู้ใช้งานป้อนข้อมูลครบตามที่กำหนด <ol style="list-style-type: none"> 3.1.1 ระบบจะนำไปสู่หน้าตั้งค่ารูปโปรไฟล์ โดยผู้ใช้สามารถเลือกรูปโปรไฟล์จากอัลบั้มรูปภาพหรือทำการถ่ายภาพใหม่ได้ 4. ผู้ใช้งานทำการลงทะเบียนข้อมูลผู้ใช้งาน จัดเก็บข้อมูลใน Firebase ที่มีชื่อว่า Player โดยจัดเก็บข้อมูล email, name, profilephoto
Post Condition	เมื่อลงทะเบียนสำเร็จระบบจะเข้าสู่หน้า Home หรือหน้าหลักของแอปพลิเคชัน Harty
Alternative Flows	-

ตารางที่ 3.3 แสดงรายละเอียดของ Use Case Edit Profile

Use Case Name	EDIT PROFILE
Use Case ID	UC3
Brief Description	สำหรับจัดการข้อมูลผู้ใช้งาน
Primary Actors	Party Owner และ Party Participant
Secondary Actors	-
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้งานทำการคลิกรูปโปรไฟล์ 2. ผู้ใช้งานแก้ไขข้อมูลผู้ใช้งาน ดังนี้ รูปภาพผู้ใช้งาน เพศ ข้อมูลช่องทางการติดต่อ 3. ผู้ใช้งานจัดการข้อมูลสำเร็จ โดยนำข้อมูลที่ผู้ใช้ทำการเปลี่ยนแปลง ไปทำการปรับปรุงแก้ไขข้อมูลใน Firebase ที่มีชื่อว่า Player
Post Condition	-
Alternative Flows	-

ตารางที่ 3.4 แสดงรายละเอียดของ Use Case Search Game

Use Case Name	SEARCH GAME
Use Case ID	UC4
Brief Description	สำหรับค้นหาเกมที่ต้องการ
Primary Actors	Party Owner
Secondary Actors	-
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นผู้ใช้ทำการค้นหาชื่อเกมที่ต้องการ 2. ผู้ใช้งานป้อนชื่อเกมที่ต้องการค้นหา 3. ระบบนำชื่อเกมมาค้นหาใน Firebase ที่มีชื่อว่า Games และทำการค้นหาชื่อเกมที่ตรงกับ Name
Post Condition	-
Alternative Flows	-

ตารางที่ 3.5 แสดงรายละเอียดของ Use Case Recommend Game

Use Case Name	RECOMMEND GAME
Use Case ID	UC5
Brief Description	ระบบแนะนำเกมตามความชอบส่วนบุคคล (Personalized Recommendation) โดยแบ่งเป็น 1. แนะนำเกมตามแพลตฟอร์มที่ชอบเล่น 2. แนะนำเกมที่ได้รับคามนิยมมีผู้สร้างปาร์ตี้มากที่สุด และ 3. แนะนำเกมใหม่ล่าสุด
Primary Actors	Party Owner และ Party Participant
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องทำการกำหนดแพลตฟอร์มของเกมที่ชอบเล่นในขั้นตอนการลงทะเบียนเสียก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นผู้ใช้เข้าสู่หน้าหลัก 2. ในหน้าหลัก ระบบจะทำการเลือกเกมที่จะแนะนำตามเงื่อนไข โดยแบ่งเป็น <ol style="list-style-type: none"> 2.1 ตามแพลตฟอร์มที่ตรงกับที่ผู้ใช้กำหนดไว้ในขั้นตอนการลงทะเบียน โดยจะใช้วิธีการสุ่มเลือกจำนวน 10 เกม ที่ตรงกับแพลตฟอร์ม 2.2 เกมที่มีผู้ใช้สร้างปาร์ตี๊ไว้เป็นจำนวนมากที่สุด 10 อันดับแรก 2.3 เกมที่มีวันวางจำหน่ายใกล้เคียงกับปัจจุบันมากที่สุดเป็นจำนวน 10 อันดับแรก
Post Condition	ผู้ใช้งานได้รับการแนะนำเกมได้ตรงตามความชอบของผู้ใช้แต่ละคน
Alternative Flows	-

ตารางที่ 3.6 แสดงรายละเอียดของ Use Case Create Party

Use Case Name	CREATE PARTY
Use Case ID	UC6
Brief Description	สำหรับสร้างห้องปาร์ตี้ที่ต้องการหาผู้เล่นเกม
Primary Actors	Party Owner
Secondary Actors	-
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้สร้างปาร์ตี้ทำการสร้างปาร์ตี้ 2. ผู้สร้างกิจกรรมป้อนข้อมูลในการสร้างปาร์ตี้ดังนี้ จำนวนผู้เล่น ช่วงเวลาที่จะเริ่มเกม แพลตฟอร์มของเกมที่จะเล่น ชื่อห้อง 3. เมื่อผู้สร้างกิจกรรมทำการสร้างกิจกรรมสำเร็จ จัดเก็บข้อมูลใน Firebase ที่มีชื่อว่า Party และ Partydetail โดยจัดเก็บข้อมูล create_date, delete, game_id, name, note, platform, start_time, user_id, user_img, user_qty
Post Condition	ระบบจะแสดงปาร์ตี้ที่สร้างใหม่ในหน้า Home และหน้า Partylist
Alternative Flows	-

ตารางที่ 3.7 แสดงรายละเอียดของ Use Case Remove Player

Use Case Name	Remove Player
Use Case ID	UC7
Brief Description	สำหรับจัดการสมาชิกปาร์ตี้
Primary Actors	Party Owner
Secondary Actors	-
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้สร้างปาร์ตี้และมีผู้เข้าร่วมปาร์ตี้ 2. ผู้สร้างปาร์ตี้คลิกปุ่มลบผู้เล่น จากนั้นระบบจะทำการ Alert Dialog ยืนยันว่าผู้ใช้ต้องการลบผู้เล่นออกจากปาร์ตี้จริงหรือไม่ เมื่อผู้ใช้ลบผู้เล่นสำเร็จ ผู้เล่นคนนั้นจะถูกลบออกจากรายชื่อสมาชิกในปาร์ตี้ทันที
Post Condition	-
Alternative Flows	-

ตารางที่ 3.8 แสดงรายละเอียดของ Use Case Choose Join Party

Use Case Name	JOIN PARTY
Use Case ID	UC8
Brief Description	สำหรับผู้ใช้ขอเข้าการร่วมปาร์ตี้ที่ผู้ใช้คนอื่นสร้าง
Primary Actors	Party Participant
Secondary Actors	Party Owner
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นผู้ใช้กดเข้าร่วมปาร์ตี้ที่ต้องการ 2. ระบบแสดงรายการปาร์ตี้ทั้งหมดในหน้า Partylist <ol style="list-style-type: none"> 2.1 ผู้ใช้ระบบทำการกดเข้าร่วมปาร์ตี้ของเกมที่ต้องการ 3. เมื่อมีผู้ใช้ทำการเข้าร่วมปาร์ตี้ ระบบจะส่ง Notification ไปยังผู้สร้างปาร์ตี้ต่างๆ
Post Condition	-
Alternative Flows	-

ตารางที่ 3.9 แสดงรายละเอียดของ Use Case Leave Party

Use Case Name	LEAVE PARTY
Use Case ID	UC9
Brief Description	สำหรับค้นหากิจกรรม
Primary Actors	Party Participant
Secondary Actors	Party Owner
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้ทำการเข้าร่วมปาร์ตี้ 2. ผู้ใช้งานคลิกปุ่ม Leave เพื่อทำการออกจากปาร์ตี้ที่เข้าร่วมอยู่ 3. ระบบแสดง Alert Dialog เพื่อยืนยันการออกจากปาร์ตี้ 4. ระบบจะส่ง Notification ไปยังผู้สร้างปาร์ตี้
Post Condition	ผู้ใช้ออกจากปาร์ตี้
Alternative Flows	-

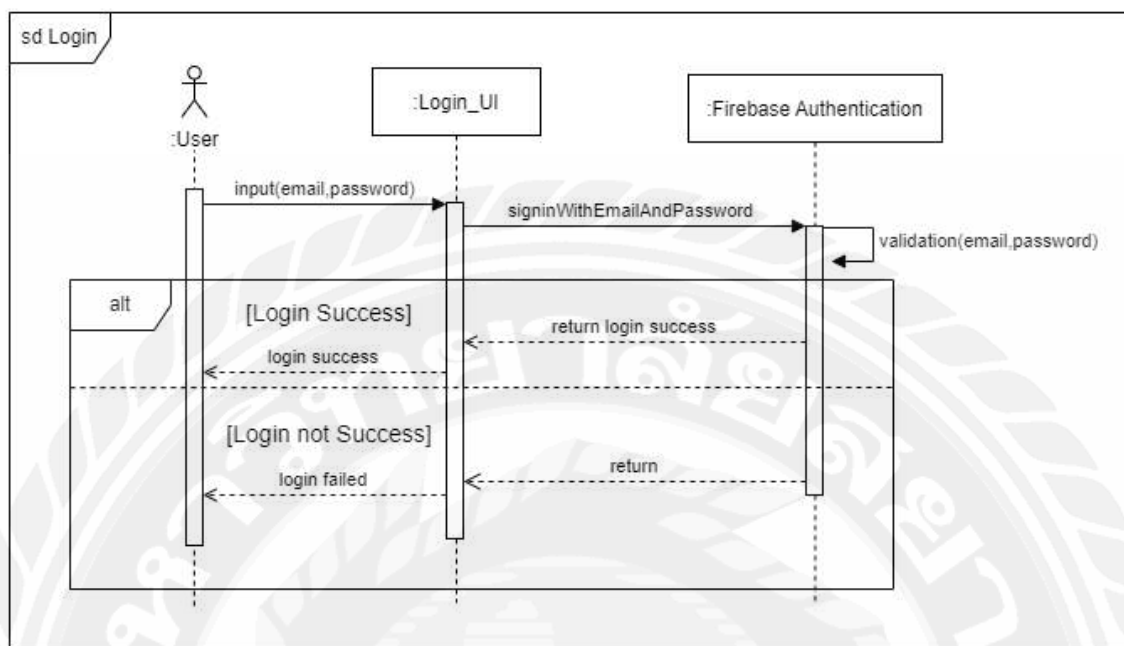
ตารางที่ 3.10 แสดงรายละเอียดของ Use Case Delete Party

Use Case Name	DELETE PARTY
Use Case ID	UC10
Brief Description	สำหรับลบปาร์ตี้ที่ตนเองสร้างออกจาก Partylist
Primary Actors	Party Owner
Secondary Actors	-
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้สร้างกิจกรรมทำการลบปาร์ตี้ที่ตนเองสร้างเท่านั้น 2. ผู้สร้างปาร์ตี้คลิกปุ่มลบ จากนั้นระบบจะทำการ Alert Dialog ยืนยันว่าผู้ใช้ต้องการลบปาร์ตี้จริงหรือไม่ เมื่อผู้ใช้ลบปาร์ตี้สำเร็จ ปาร์ตี้จะถูกลบ และผู้เข้าร่วมปาร์ตี้จะถูกลบออกจากปาร์ตี้เกมที่ถูกลบ โดยอัตโนมัติ
Post Condition	-
Alternative Flows	-

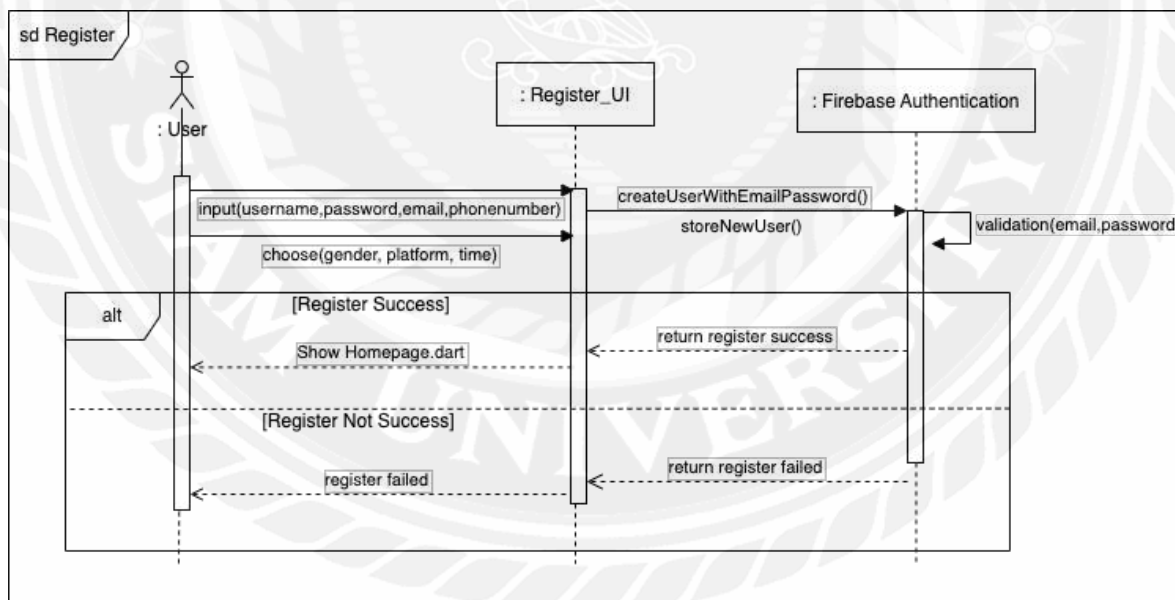
ตารางที่ 3.11 แสดงรายละเอียดของ Use Case Chat

Use Case Name	CHAT
Use Case ID	UC11
Brief Description	สำหรับส่งข้อความสื่อสารกันระหว่างผู้ใช้ที่ได้มีการกดติดตามกันแล้ว
Primary Actors	Party Owner
Secondary Actors	Party Participant
Preconditions	-
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้ทำ Follow ผู้ใช้อื่น <ol style="list-style-type: none"> 1.1 เมื่อทำการ Follow ผู้ใช้อื่น <ol style="list-style-type: none"> 1.1.1 จะสามารถส่งข้อความส่วนตัวถึงกันได้
Post Condition	-
Alternative Flows	-

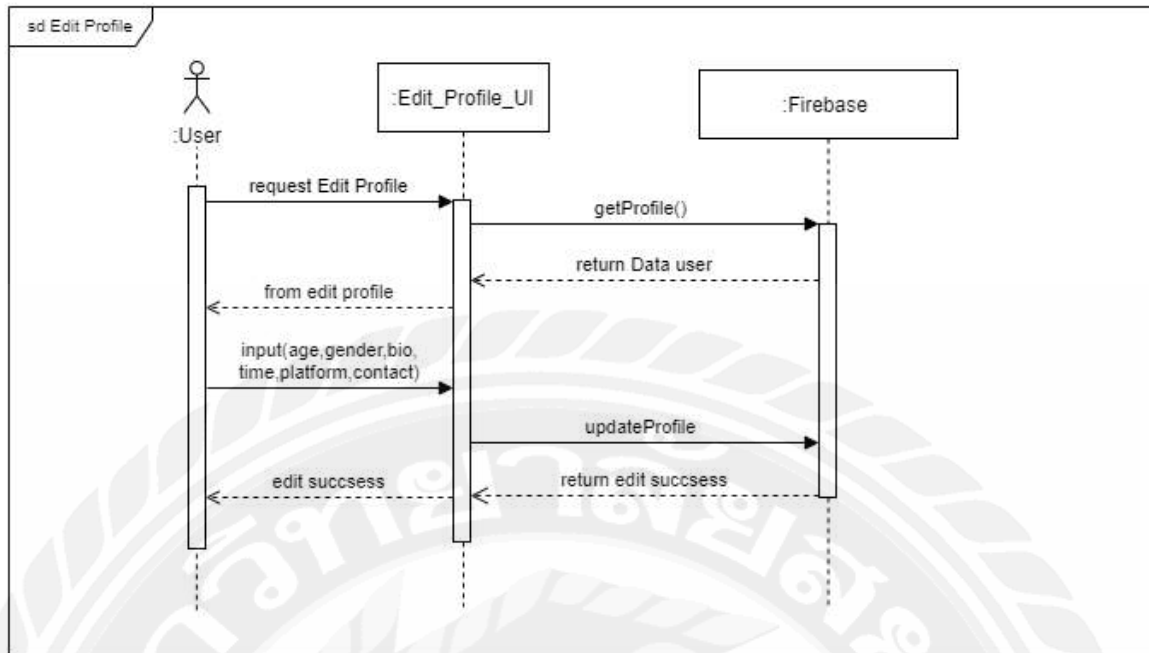
3.4 แสดงขั้นตอนการทำงานของฟังก์ชันด้วย Sequence Diagram



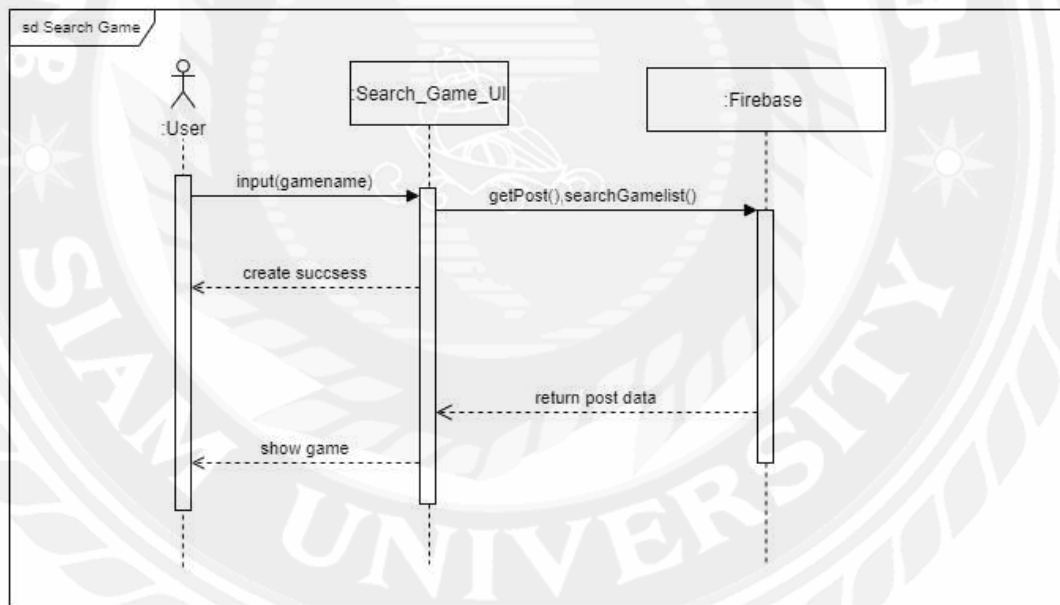
รูปที่ 3.2 Sequence Diagram : Login



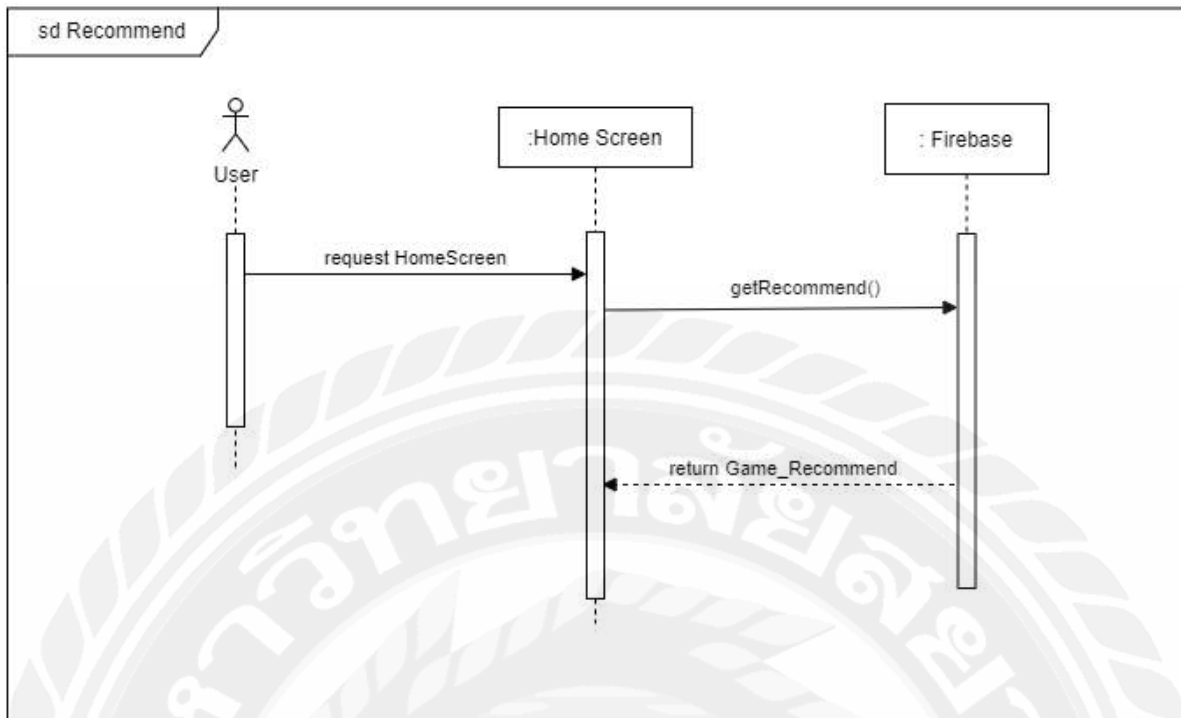
รูปที่ 3.3 Sequence Diagram : Register



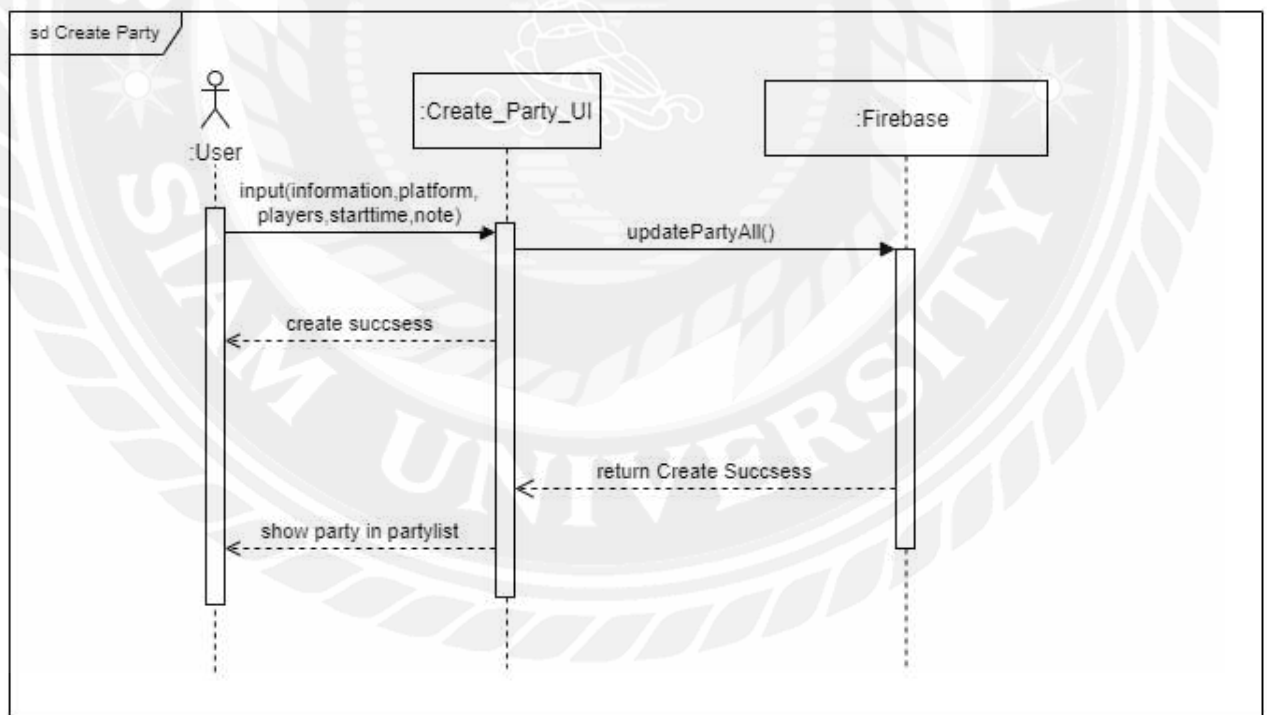
รูปที่ 3.4 Sequence Diagram : Edit Profile



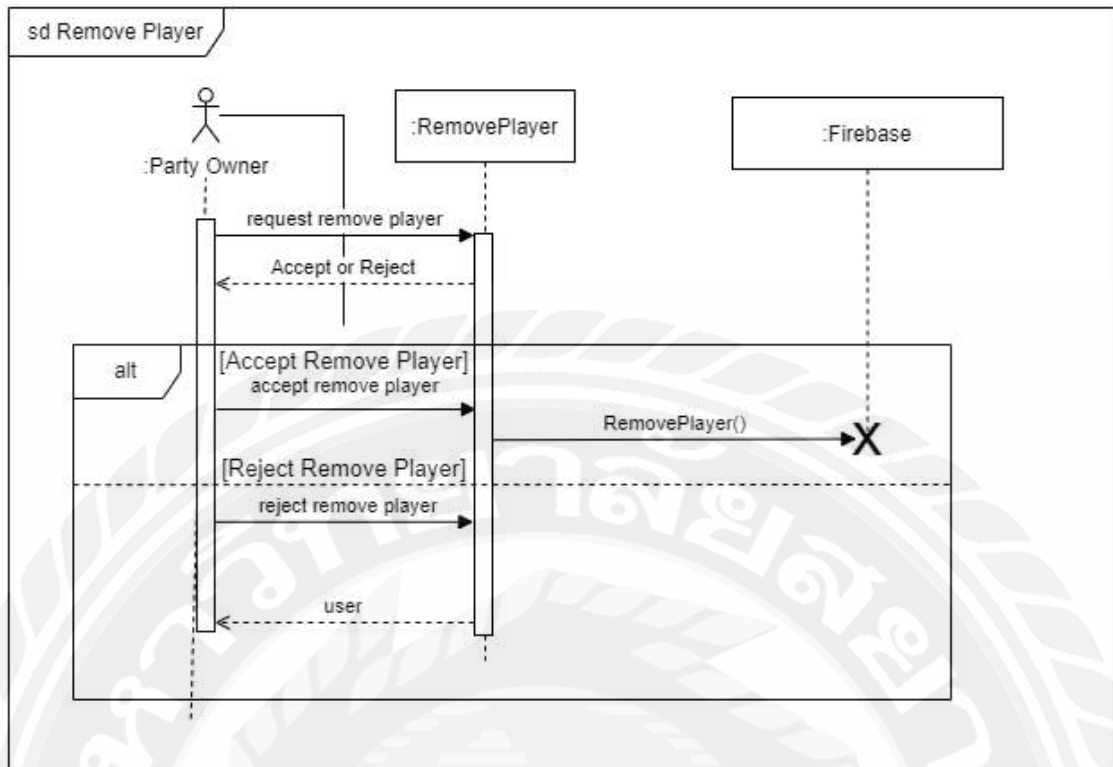
รูปที่ 3.5 Sequence Diagram : Search Game



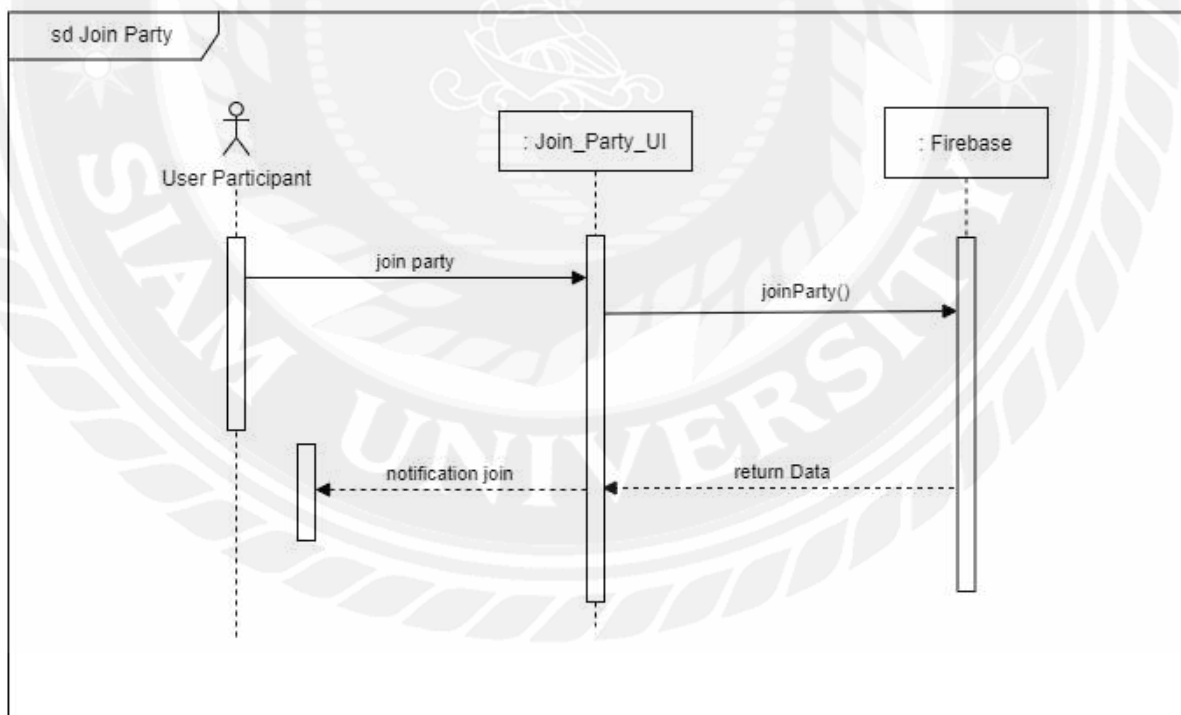
รูปที่ 3.6 Sequence Diagram : Recommend Game



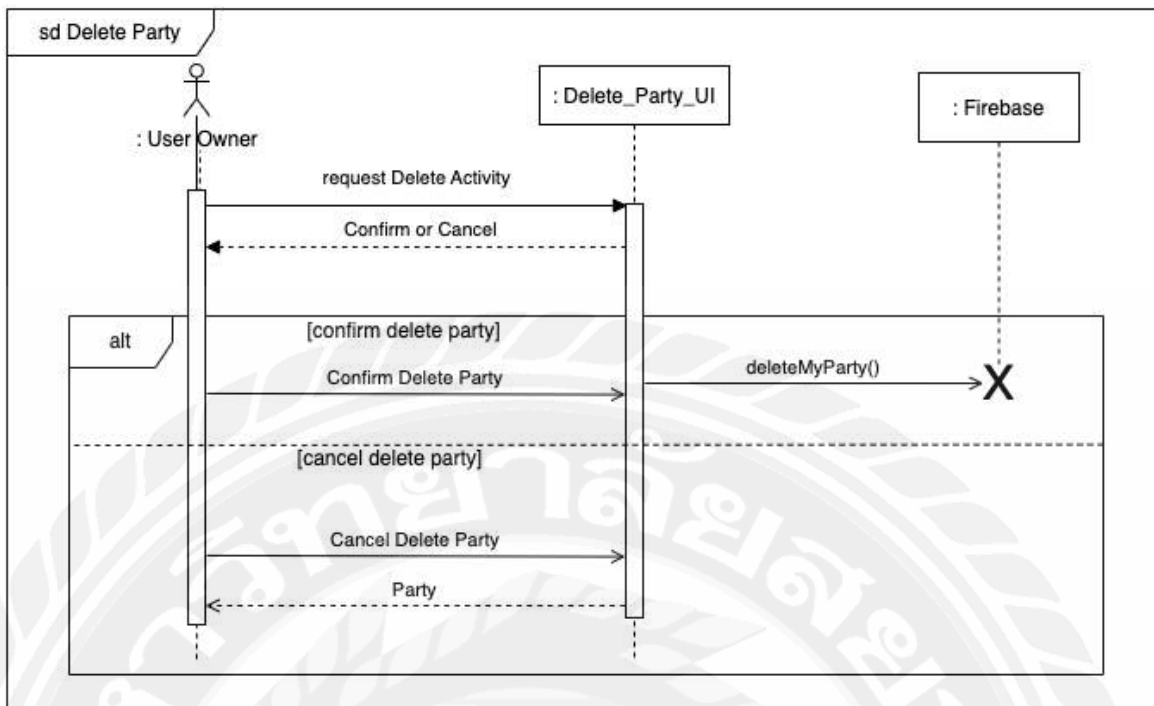
รูปที่ 3.7 Sequence Diagram : Create Party



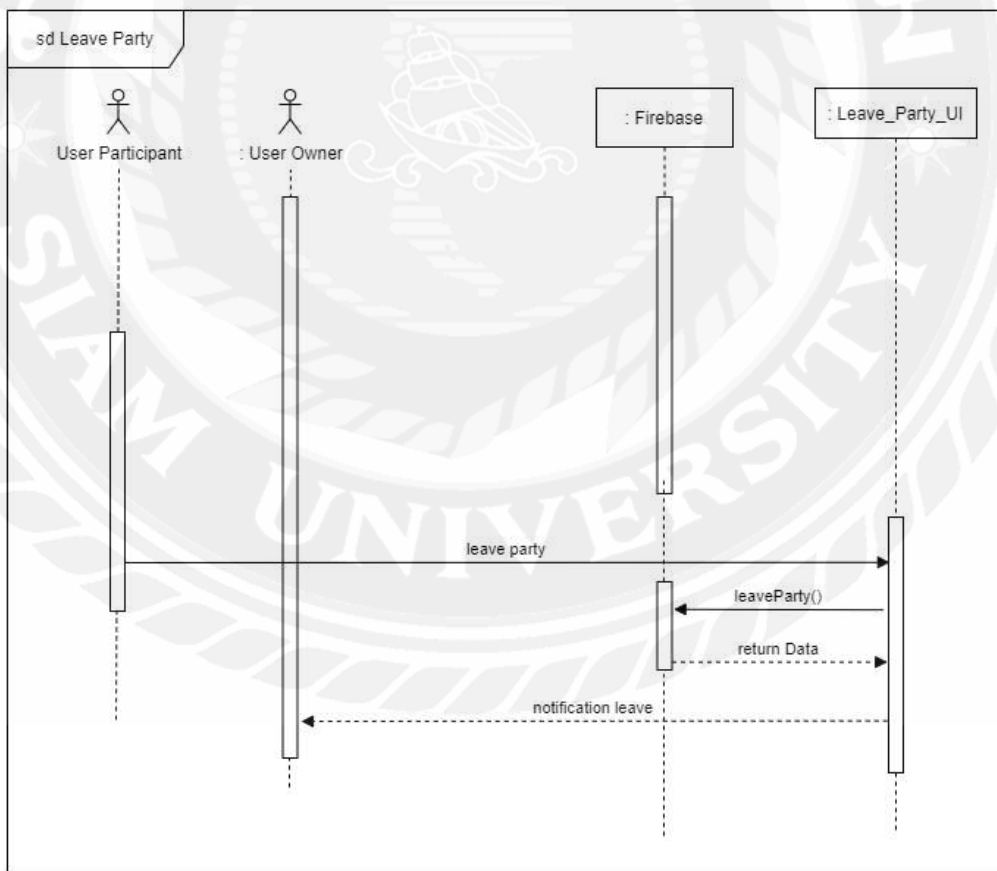
รูปที่ 3.8 Sequence Diagram : Remove Player



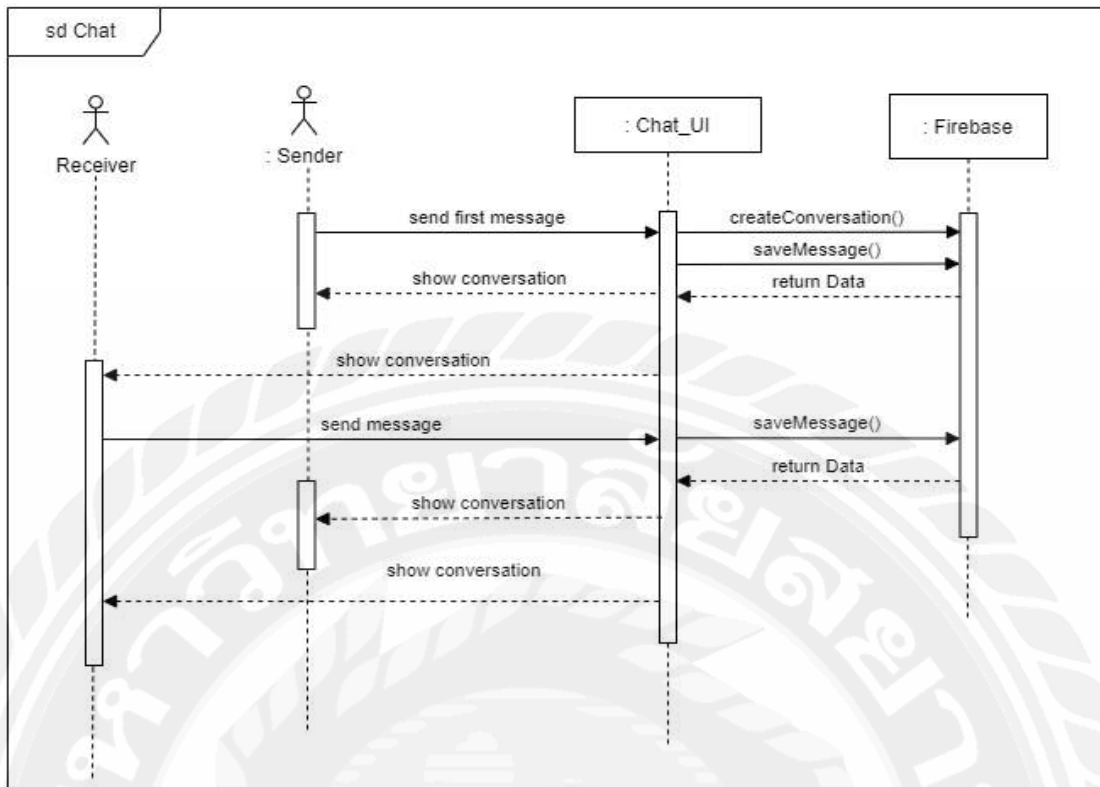
รูปที่ 3.9 Sequence Diagram : Join Party



รูปที่ 3.10 Sequence Diagram : Leave Party

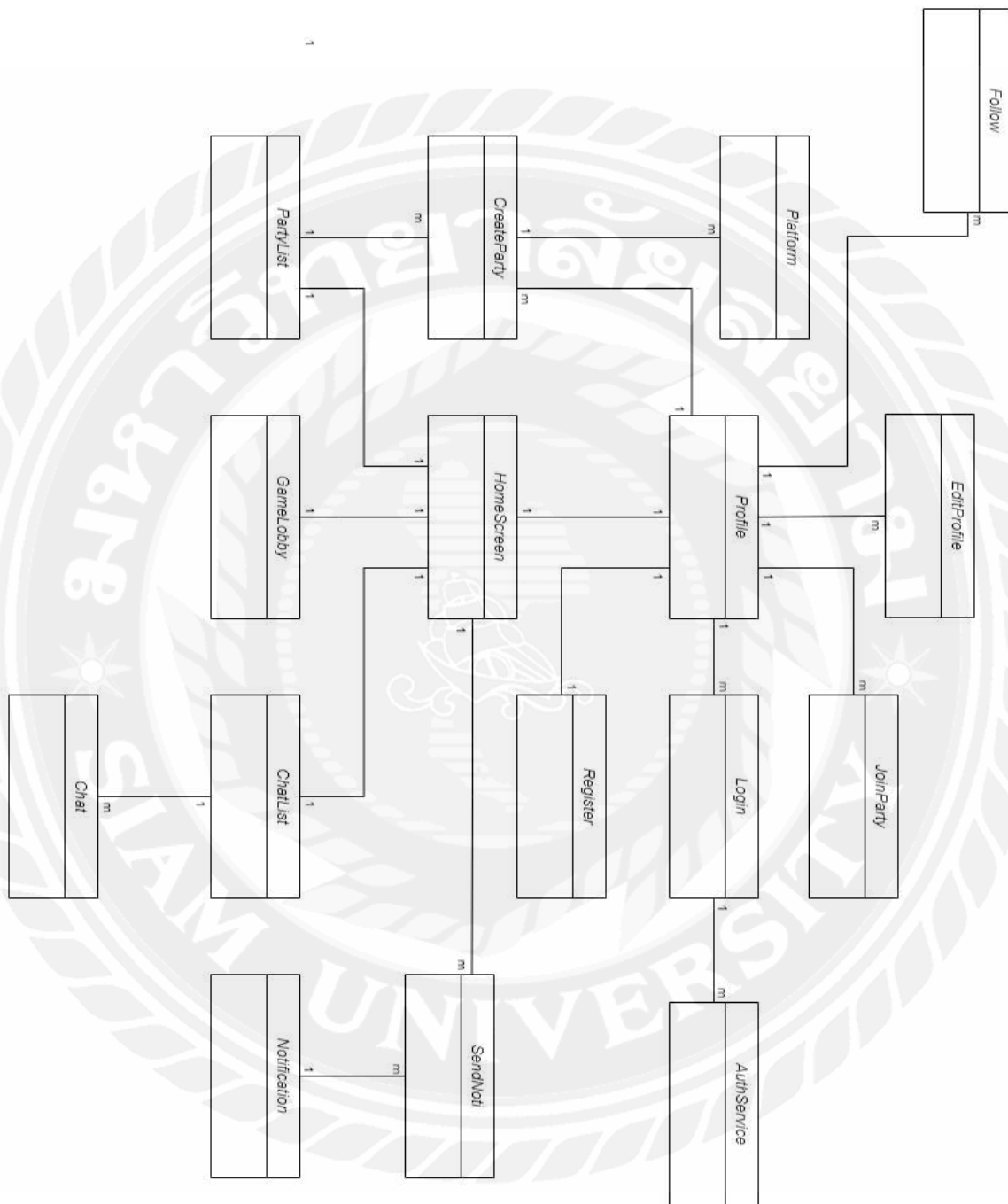


รูปที่ 3.11 Sequence Diagram : Delete Party



รูปที่ 3.12 Sequence Diagram : Chat

3.5 แผนภาพแสดงองค์ประกอบคลาส (Class Diagram)



รูปที่ 3.13 Class Diagram ของแอปพลิเคชัน Harty

3.6 รายละเอียดของคลาส

ตารางที่ 3.12 รายละเอียดของคลาส PartyList

PartyList
<ul style="list-style-type: none"> - user: object - snapshotsPartyDetail: object - mainWidget: object - widget: object - btnJoin: String - countPlayer: int - snapshotsPartyDetail: object - snapshotsGame: object - snapshots: object
<ul style="list-style-type: none"> + initFirebaseMessaging()

ตารางที่ 3.13 รายละเอียดของคลาส Profile

Profile
<ul style="list-style-type: none"> - Following: int - Follower: int - getDataStatus: bool - snapshotFollowing: object - snapshotFollower: object - user: object - arguments: object - _myFollowCollection: object - button: String - contact: Array - platforms: Array - snapshotsProfile: object - snapshotsFollowing: object
<ul style="list-style-type: none"> + initFirebaseMessaging() + getFollow()

ตารางที่ 3.14 รายละเอียดของคลาส editProfile

editProfile
<ul style="list-style-type: none"> - arrPlatform : Array - arrContacts : Array - formKey : object - myPlayer : object - setDataStatus : bool - userID : String - name : String - description : String - age : String - startTime : String - endTime : String - platforms : Array - contacts : Array - _myPlayersCollection : object - genderController : String - isPCSelected : bool - isPlaystationSelected : bool - isNintendoSelected : bool - isMobileSelected : bool - isXboxSelected : bool - isArcadeSelected : bool - isXboxSelected : bool - avaTime2Changed : String - avaTime2ToValidate : String - avaTime2Saved : String
<ul style="list-style-type: none"> + _showDiscardDialog() + initFirebaseMessaging()

ตารางที่ 3.15 รายละเอียดของคลาส CreateParty

CreateParty
<ul style="list-style-type: none"> - getDataStatus : bool - arrPlatform : array - platformController : String - formKey : object - myParty : Array - firebase : object - snapshot : object - docs : object - radio : int - user : object - maxPlayers : int - start_time : int - startTime : String - snapshot : object - _myPartyCollection : object - _myPartyDetailCollection : object - arguments : object - now : String - formattedDate : String - docRef : object - docref : String - docRefDetail : object
<ul style="list-style-type: none"> + getData() + initFirebaseMessaging()

ตารางที่ 3.16 รายละเอียดของคลาส PlatForm

PlatForm
- platform_id: String
- platform: String

ตารางที่ 3.17 รายละเอียดของคลาส Follow

Follow
- follow_date: Timestamp
- following: String
- status: String
- uid: String
- unfollow_date: String

ตารางที่ 3.18 รายละเอียดของคลาส login

login
- timeBackPressed : DateTime
- difference : DateTime
- isExitWarning : DateTime
- timeBackPressed : DateTime
- message : String
- isRedTheme : bool
- isPinkTheme : bool
- isBlueTheme : bool
- isGreenTheme : bool
- isYellowTheme : bool
- isEngLanguage : bool
- authBloc : object

ตารางที่ 3.19 รายละเอียดของคลาส register

register
- age: String
- contact: String
- deleted: number
- descriptions: String
- email: String
- start_time: String
- end_time: String
- gender: String
- picture: String
- platform: String
- token: String
- uid: String
- username: String
+ _showWarningDialog()
+ initFirebaseMessaging()
+ getData()

ตารางที่ 3.20 รายละเอียดของคลาส joinParty

joinParty
mounted: bool
userId: String
joinedParty: bool
btnJoin: String
players: int
snapshotPlayer: object
token: String
endpointUrl: String
title: String
body: String
bodyData: object
arrUserIDinParty: Array
arrUserKickIDinParty: Array

ตารางที่ 3.21 รายละเอียดของคลาส HomeScreen

HomeScreen
- gameId : String
- imageURL : String
- name : String
- platform : String
- firebaseMessaging : object
- getDataStatus : bool
- getDataGamesInterested : bool
- user : object
- allData : Array
- playerData : Array
- platformID : Array
- gameInterested : Array
- countData : int
- currentads : int
- beginningDate : DateTime
- newDate : DateTime
- snapshot : object
- docs : object
- userImage : String
- username : String
- countGamesInterest : int
- countGames : int
- round : int
- maxGame : int
- keepGameToCheck : Array
- random : object
- randomNumber : int
- gameId : String
- snapshotGame : object
- game : object

ตารางที่ 3.21 รายละเอียดของคลาส HomeScreen (ต่อ)

GameLobby
<ul style="list-style-type: none"> - mainWidget : object - button : String - countPlayer : int - widget : object - platform : String - gameId : String - gameImg : String - gamename : String - gamePlatform : String - _hotGame : Array - hotGame : Array - count : int - screens : Array - _selectedIndex : int - argumentIndex : int - checkArguments : object
<ul style="list-style-type: none"> + getData() + getGame() + initFirebaseMessaging() + DeletePartyDetail() + _onItemTapped()

ตารางที่ 3.22 รายละเอียดของคลาส GameLobby

GameLobby
<ul style="list-style-type: none"> - games : Array - Query : String - txtQuery : String - debouncer : object - getDataStatus : bool - allGames : Array - snapshot : object - queryLower : string - docs : object - data : object - platform : String - game : object - duration : object
<ul style="list-style-type: none"> + getData() + initFirebaseMessaging() + debounce() + searchGame()

ตารางที่ 3.23 รายละเอียดของคลาส ChatList

ChatList
- user : object
- DateDiff : object
- clearChatStatus : bool
- snapshotConversation : object
- snapshotChat : object
- dateDiff : int
- chatuserid : String
- snapshotsChatlist : object
- snapshotsPlayers : object
- playerImage : String
- snapshotsList : object
- following : String
- i : int
- snapshotsPlayer : object
+ _timeToClearChat()
+ initFirebaseMessaging()

ตารางที่ 3.24 รายละเอียดของคลาส Chat

Chat
<ul style="list-style-type: none"> - _myChatCollection : object - _myConversationCollection : object - _myNotiCollection : object - messageController : String - _scrollController : object - saveDataStatus : bool - _needsScroll : bool - ConversationDocID : String - saveDataStatus : bool - snapshot : object - docRef : object - messages : Array - arguments : object - token : String - hashCodeUID : String - hashCodeFollowID : String - chatuserid : String - keyID : String - title : String - body : String - dt : String - newFormat : String - date : String - endpointUrl : String - bodyData : object
<ul style="list-style-type: none"> + saveConversation() + initFirebaseMessaging()+ getEvenReqPosts() + getCloudFirestore()

ตารางที่ 3.25 รายละเอียดของคลาส SendNoti

SendNoti
sendToUserId: String
title: String
body: String
uid: String
menu: String
token: String
endpointUrl: String
bodyData: object

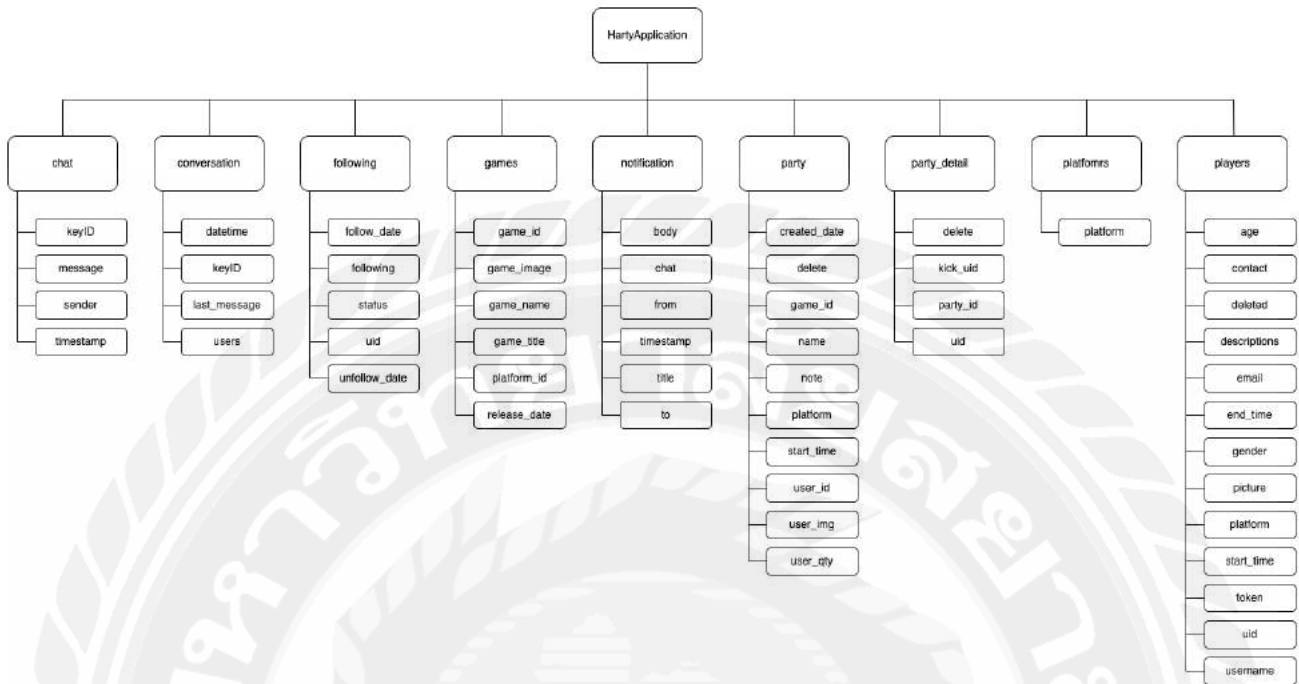
ตารางที่ 3.26 รายละเอียดของคลาส NotificationList

NotificationList
- user : object
- now : String
- today : String
- snapshotsNoti : object
- t : String
- dt : String
- newFormat : String
- dateToCheck : String
- aDate : String
- date : String
- snapshotsPlayers : object
- playerImage : String
- i : int
- snapshotsPlayer : object
+ initFirebaseMessaging()

ตารางที่ 3.27 รายละเอียดของคลาสAuthService

AuthService
<ul style="list-style-type: none"> - FirebaseAuth: _firebaseAuth - GoogleSignIn: googleSignIn - currentUser: object - googleUser: object - googleAuth: object - credential: object - statusNewUser: bool - result: object - user: object - snapshotPlayer: object - res: object - fbToken: objec
<ul style="list-style-type: none"> + loginGoogle() + loginFacebook() + logout()

3.7 โครงสร้างข้อมูล (Data Structure)



รูปที่ 3.14 โครงสร้างของข้อมูล (NoSQL Database Structure)

3.8 รายละเอียดของโครงสร้างข้อมูล

ตารางที่ 3.28 รายละเอียดของข้อมูล CHAT

Element Name	Description	Data Type
keyID	รหัสสำหรับระบุหน้าจอกการใช้งาน	String
message	ข้อความ	String
sender	รหัสยืนยันตัวตนของผู้ใช้	String
timestamp	เวลาที่ส่งข้อความ	Timestamp

ตารางที่ 3.29 รายละเอียดของข้อมูล CONVERSATION

Element Name	Description	Data Type
datetime	เวลาที่ส่งข้อความล่าสุด	String
keyID	รหัสสำหรับระบุหน้าจอกการใช้งาน	String
Last_message	ข้อความที่ส่งล่าสุด	String
users	รหัสยืนยันตัวตนของผู้ใช้	Array

ตารางที่ 3.30 รายละเอียดของข้อมูล FOLLOWING

Element Name	Description	Data Type
follow_date	วันที่กดติดตาม	Timestamp
following	รหัสยืนยันตัวตนของผู้ใช้ผู้ใช้ที่ติดตาม	String
status	สถานะการติดตาม	String
uid	รหัสยืนยันตัวตนของผู้ใช้	String
Unfollow_date	วันที่กดยกเลิกติดตาม	Timestamp

ตารางที่ 3.31 รายละเอียดของข้อมูล GAMES

Element Name	Description	Data Type
game_id	รหัสเฉพาะของเกม	String
game_image	ที่อยู่ของรูปปกเกม	String
game_name	ชื่อเกม	String
game_title	ชื่อเกมที่โชว์บนระบบ	String
platform_id	แพลตฟอร์มของเกม	Array
release_date	วันที่ปล่อย	Timestamp

ตารางที่ 3.32 รายละเอียดของข้อมูล NOTIFICATION

Element Name	Description	Data Type
body	ข้อความแจ้งเตือน	String
chat	ประเภทการแจ้งเตือน	bool
from	รหัสยืนยันตัวตนของผู้ส่ง	String
timestamp	ชื่อเกมที่แสดงบนระบบ	Timestamp
title	หัวข้อการแจ้งเตือน	String
to	รหัสยืนยันตัวตนของผู้รับ	String

ตารางที่ 3.33 รายละเอียดของข้อมูล PARTY

Element Name	Description	Data Type
created_date	เวลาที่สร้างปาร์ตี้	Timestamp
delete	สถานะการลบปาร์ตี้	String
game_id	รหัสเฉพาะของเกม	String
Element Name	ชื่อปาร์ตี้	String
note	โน้ตปาร์ตี้	String
platform	แพลตฟอร์มของเกม	String
start_time	เวลาเริ่มปาร์ตี้	String
user_id	DocID ผู้ใช้ที่สร้างปาร์ตี้	String
user_img	รูปของผู้สร้างปาร์ตี้	String
user_qty	จำนวนสมาชิกสูงสุดของปาร์ตี้	String

ตารางที่ 3.34 รายละเอียดของข้อมูล PARTY_DETAIL

Element Name	Description	Data Type
delete	สถานะการลบปาร์ตี้	String
kick_uid	รหัสยืนยันตัวตนผู้ใช้ที่ถูกขับออกในปาร์ตี้	Array
party_id	รหัสของปาร์ตี้	String
uid	รหัสยืนยันตัวตนสมาชิกในปาร์ตี้	Array

ตารางที่ 3.35 รายละเอียดของข้อมูล PLATFORM

Element Name	Description	Data Type
platform	ชื่อแพลตฟอร์ม	String

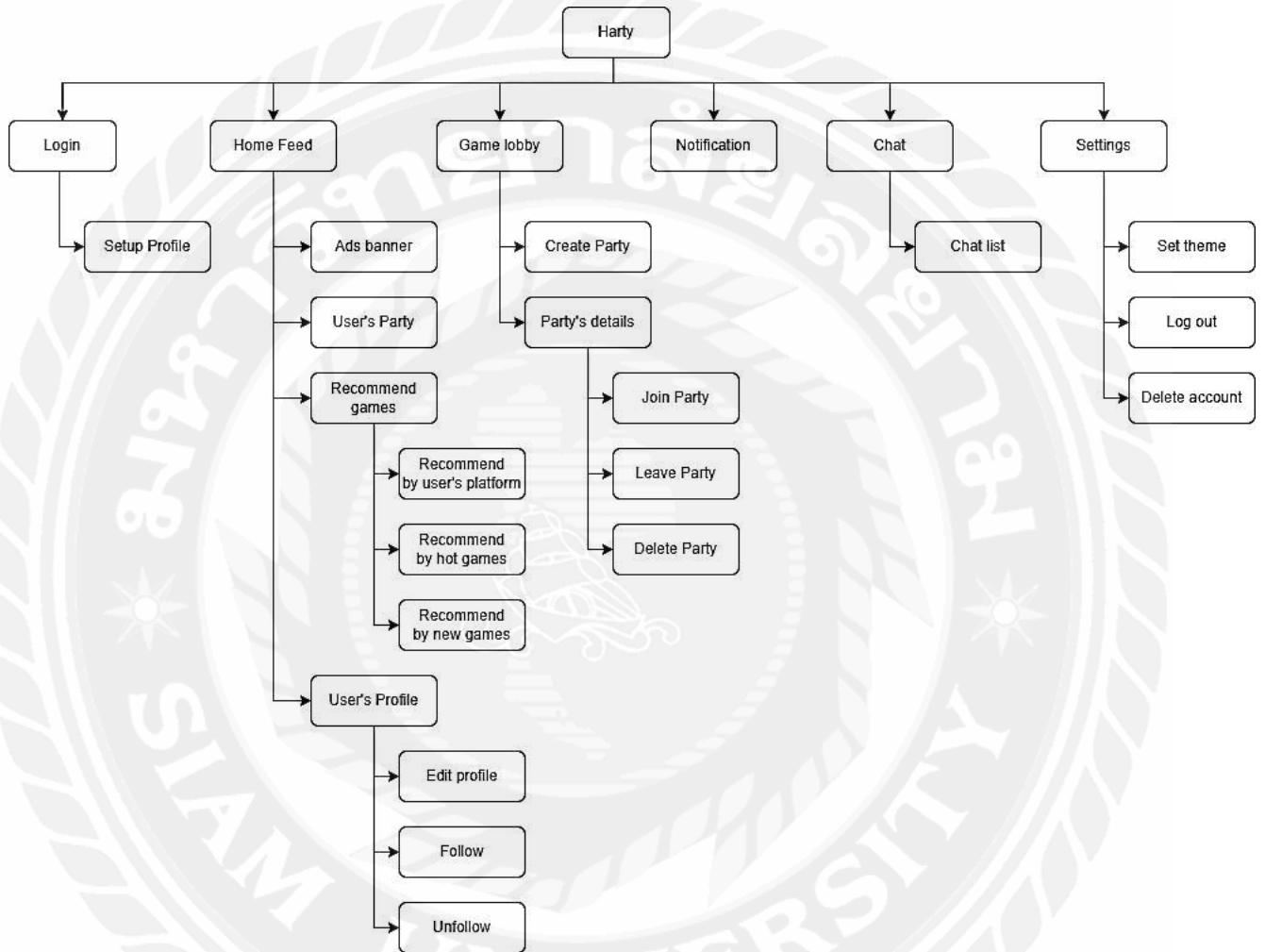
ตารางที่ 3.36 รายละเอียดของข้อมูล PLAYERS

Element Name	Description	Data Type
age	ชื่อแพลตฟอร์ม	String
contact	ข้อมูลการติดต่อ	String
deleted	สถานะการลบผู้ใช้	number
description	รายละเอียดผู้ใช้	String
email	อีเมลของผู้ใช้	String
end_time	เวลาที่เลิกเล่นเกม	String
gender	เพศของผู้ใช้	String
picture	รูปภาพของผู้ใช้	String
platform	แพลตฟอร์มของผู้ใช้เล่น	String
Start_time	เวลาที่เริ่มเล่นเกม	String
token	รหัสของอุปกรณ์ที่ผู้ใช้ใช้งาน	String
uid	รหัสยืนยันตัวตนของผู้ใช้	String
username	ชื่อบัญชีของผู้ใช้	String

บทที่ 4

การออกแบบทางกายภาพ

4.1 โครงสร้างของแอปพลิเคชัน (Application Structure Map)



รูปที่ 4.1 โครงสร้างของแอปพลิเคชัน Harty

ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Harty

หน้าจอ	คำอธิบายรายละเอียด
Login (loginscreen.dart)	การเข้าสู่ระบบเพื่อยืนยันตัวตน
Setup Profile (setup_profile1.dart) (setup_profile2.dart) (setup_profile3.dart)	<p>ผู้ใช้ใหม่รายจะต้องทำการตั้งค่าโปรไฟล์ เพื่อใช้เป็นข้อมูลแสดงตัวตนในระบบ</p> <ul style="list-style-type: none"> - ตั้งค่าชื่อ - ตั้งค่าข้อมูลเบื้องต้น (เช่น เพศ , อายุ) - ตั้งค่ารูปโปรไฟล์ - ตั้งค่าแพลตฟอร์มเกมที่ใช้เล่น - ตั้งค่าเวลาที่สะดวกเล่นเกม
Home Feed (homescreen.dart)	<p>หน้าแรก (Home) ของแอปพลิเคชัน ระบบจะทำการแสดงโฆษณาโดยเลื่อนอัตโนมัติ และระบบจะแสดงข้อมูลปาร์ตี้เกมทั้งหมดของผู้ใช้ที่ผู้ใช้เป็นคนสร้างปาร์ตี้ และมีระบบแนะนำเกมโดยจะมี 3 รูปแบบได้แก่</p> <ul style="list-style-type: none"> - การสุ่มเลือก 10 เกม ที่ตรงตามแพลตฟอร์มของเกมที่ใช้กำหนดไว้ในขั้นตอนการลงทะเบียน - เกมที่มีผู้ใช้สร้างปาร์ตี้เป็นจำนวนมาก - เกมที่มีวันวางจำหน่ายล่าสุดไม่เกินปีปัจจุบันในระบบ
Profile (profilescreen.dart)	<p>แสดงข้อมูลผู้ใช้ในระบบ และสามารถแก้ไขข้อมูลส่วนตัวได้ดังนี้</p> <ul style="list-style-type: none"> - รูปภาพ - ชื่อผู้ใช้งาน - เพศ - อายุ - เวลาที่สะดวกเล่น - แพลตฟอร์มเกมที่ใช้เล่น <p>ผู้ใช้สามารถกดติดตามผู้ใช้คนอื่นในระบบ หรือยกเลิกการติดตามได้</p>
Game Lobby (gamelobbysceen.dart)	<p>แสดงเกมทั้งหมดในระบบ</p> <p>โดยผู้ใช้สามารถกดเลือกเกมที่ต้องการจะสร้างปาร์ตี้เกมได้</p>

ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Harty (ต่อ)

หน้าจอ	คำอธิบายรายละเอียด
Game Room (gameroomscreen.dart)	แสดงรายชื่อห้องทั้งหมดที่มีผู้ใช้สร้างไว้ เป็นการแสดงผลแบบคอลลิมน์สามารถเลื่อนดูได้
Create party (createpartyscreen.dart)	การสร้างปาร์ตี้เกม โดยมีข้อมูลที่ต้องระบุดังนี้ <ul style="list-style-type: none"> - ชื่อห้องปาร์ตี้ - เวลาที่เริ่มเล่น - แพลตฟอร์มที่เล่น - จำนวนผู้เล่นที่ต้องการ
Party's Detail (gameroommembersceen.dart)	แสดงรายละเอียดของปาร์ตี้ในเกมนั้นๆ โดยมีรายละเอียดดังนี้ <ul style="list-style-type: none"> - ชื่อห้องปาร์ตี้ - เวลาที่เริ่มเล่น - แพลตฟอร์มที่เล่น - จำนวนผู้เล่นในห้อง - รายชื่อผู้เล่นที่เข้าร่วมแล้ว <p>โดยผู้ใช้สามารถเลือกเข้าร่วมปาร์ตี้เกมได้ หรือ ออกจากปาร์ตี้เกมนั้นๆ ได้</p> <p>ถ้าผู้ใช้เป็นผู้สร้างสร้างปาร์ตี้เกมนั้น จะสามารถลบปาร์ตี้เกมนั้นๆ ออกจากระบบได้</p>
Notification (notificationlistsceen.dart)	แสดงการแจ้งเตือน โดยมีรายละเอียดดังนี้ <ul style="list-style-type: none"> - มีผู้เข้าร่วมปาร์ตี้เกมที่มีผู้ใช้สร้างไว้ - มีผู้เล่นออกจากปาร์ตี้เกม - เจ้าของห้องทำการลบผู้ใช้ออกจากปาร์ตี้เกมได้
Chat (chatscreen.dart)	ระบบการส่งข้อความถึงผู้ใช้คนอื่น โดยจะแสดงผลข้อความที่ส่ง และข้อความที่ผู้ใช้ตอบกลับ เป็นการแสดงผลแบบเรียลไทม์ (Real Time)
Chat List (chatlistsceen.dart)	แสดงรายชื่อที่ผู้ใช้ติดตามผู้ใช้คนอื่นไว้ สำหรับการส่งข้อความ และแสดงรายการประวัติการแชทที่ผู้ใช้เคยส่งข้อความถึงผู้ใช้คนอื่นในระบบ
Settings (settingsceen.dart)	แสดงรายการตั้งค่าสำหรับผู้ใช้โดยมีการตั้งค่าดังนี้ <ul style="list-style-type: none"> - การตั้งค่าธีมสีในระบบ

ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Harty (ต่อ)

หน้าจอ	คำอธิบายรายละเอียด
	<ul style="list-style-type: none"> - การออกจากระบบ - การลบบัญชีผู้ใช้จากระบบ
Edit Profile (editprofile.dart)	แก้ไขข้อมูลรายละเอียดใน โปรไฟล์ของผู้ใช้

4.2 ระบบแนะนำเกมตามความชอบส่วนบุคคล (Personalized Recommendation)

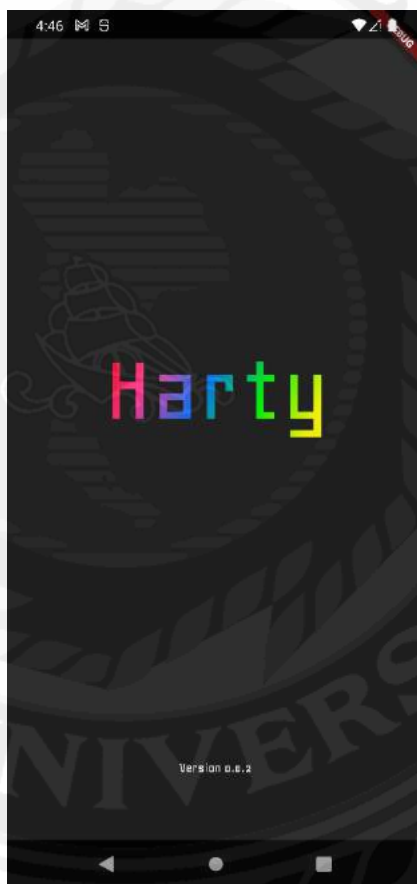
แอปพลิเคชัน Harty มีระบบแนะนำเกมเพื่อให้ผู้ใช้มีโอกาสค้นพบเกมและเพื่อนใหม่ๆ โดยระบบแนะนำเกมจะมีอยู่ 3 รูปแบบหลักๆ ดังนี้ 1. แนะนำโดยการสุ่มเลือก 10 เกม โดยระบบจะสุ่มดึงรายชื่อเกมที่มีในฐานข้อมูลและมีแพลตฟอร์มตรงตามที่ใช้กำหนดไว้ในขั้นตอนการลงทะเบียนเข้าใช้งานระบบ เพื่อให้ผู้ใช้ได้ค้นพบเกมอื่นๆ บนแพลตฟอร์มที่ผู้ใช้ชื่นชอบอยู่แล้ว 2. แนะนำเกมที่มีผู้สร้างปาร์ตี้ไว้เป็นจำนวนมาก ซึ่งจะทำให้ผู้ใช้มีโอกาสได้ลองประสบการณ์เกมที่มีผู้เล่นคนอื่นเล่นเกมนี้อยู่แล้วเป็นจำนวนมากจาก 10 อันดับแรกจากยอดผู้เล่นได้ทำการสร้างปาร์ตี้ไว้ และอาจทำให้ได้พบปะเพื่อนใหม่ๆเป็นจำนวนมากเช่นกันและ 3. แนะนำใหม่ โดยแสดงเกมที่มีกำหนดวางจำหน่ายล่าสุดตามวันที่ในระบบของผู้ใช้ เพื่อให้ผู้ใช้ค้นพบเกมที่ใหม่ๆ และมีโอกาสจะได้พบปะเพื่อนและสังคมเกมใหม่ๆ

4.3 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface Design)

แอปพลิเคชัน Harty ทำหน้าที่เป็นตัวกลางสำหรับกลุ่มคนที่ต้องการแบ่งปันประสบการณ์และค้นหาเพื่อนที่ชื่นชอบเกมเดียวกันและประเภทใกล้เคียงกัน ในการออกแบบส่วนติดต่อกับผู้นั้นจะยึดหลักการของ UX/UI เป็นหลัก โดยคำนึงถึงความพึงพอใจและการใช้งานที่ง่ายต่อผู้ใช้ ด้วยการออกแบบที่เรียบง่าย และใช้โทนสีที่เป็นที่ชื่นชอบสำหรับเกมเมอร์โดยเฉพาะซึ่งจะเป็นโทนมืด และสีนีออนเพื่อให้มีความทันสมัย โดยผู้ใช้สามารถเรียนรู้การใช้งานระบบได้ด้วยตนเอง ดังนั้นรูปแบบ ตัวอักษร การจัดวางตำแหน่งวัตถุต่างๆ จะเป็นที่คุ้นเคยของผู้ใช้ โดยมีรายละเอียดดังนี้

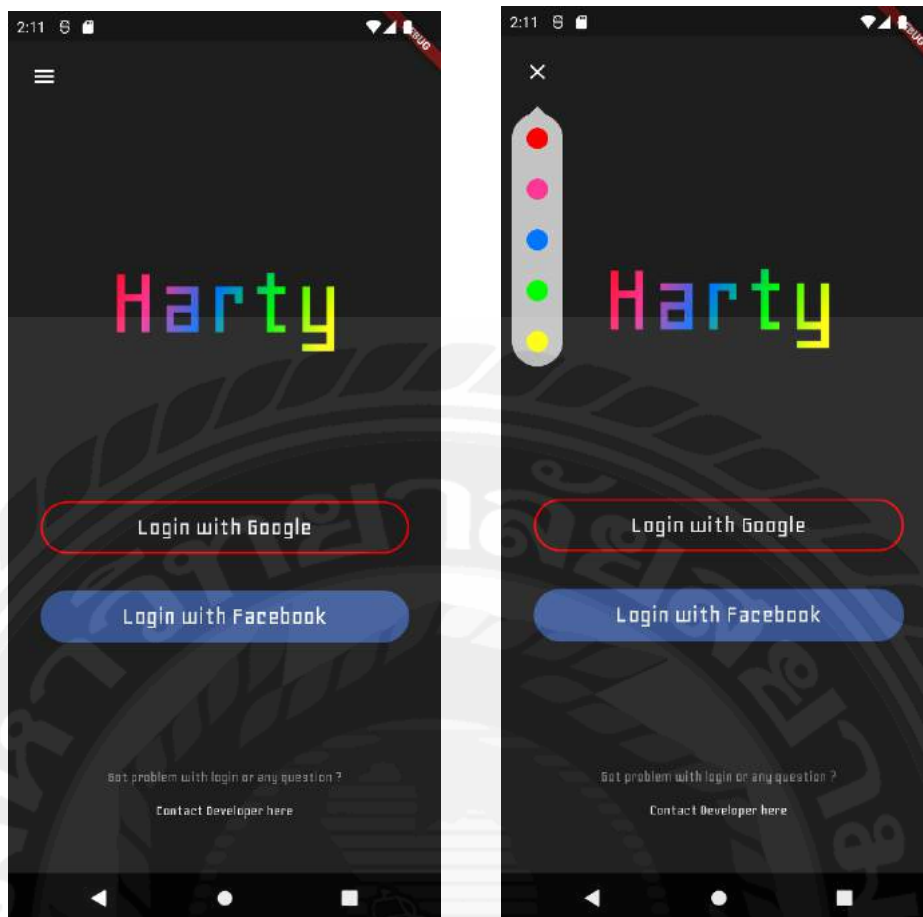


รูปที่ 4.2 ไอคอนของแอปพลิเคชัน Harty



รูปที่ 4.3 หน้า Splash Screen

จากรูปที่ 4.3 หน้า Splash Screen เป็นหน้าแรกเมื่อผู้ใช้เข้าใช้งานแอปพลิเคชัน ในหน้าจอนี้จะแสดงโลโก้ของแอปพลิเคชัน Harty และบอกเวอร์ชันในการพัฒนาปัจจุบันของแอปพลิเคชัน โดยจะแสดงผลให้ผู้ใช้เห็นเป็นเวลา 2 วินาที ก่อนเข้าสู่หน้าการสู่ระบบ (Login)

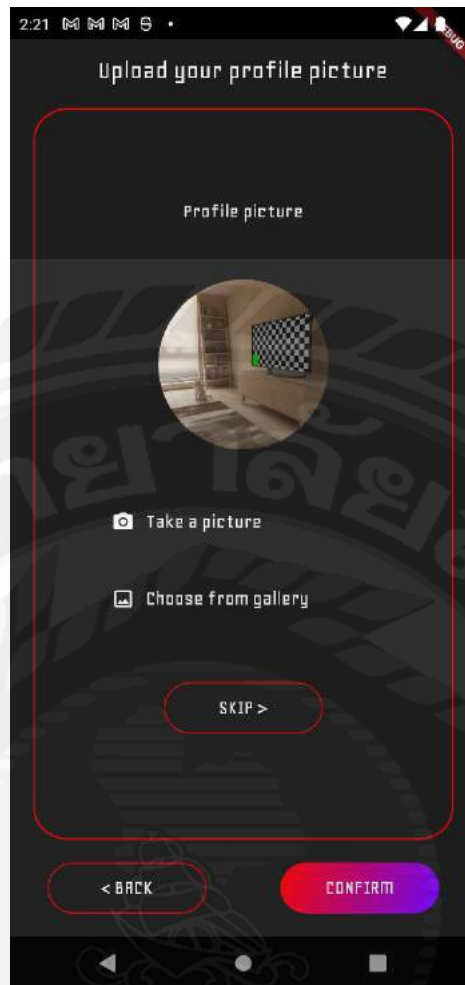


รูปที่ 4.4 หน้าเข้าสู่ระบบ (Login)

จากรูปที่ 4.4 หน้าเข้าสู่ระบบ (Login) ในหน้าจอนี้จะแสดงโลโก้ของแอปพลิเคชัน และช่องทางการเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชันโดยให้ผู้ใช้เลือกระหว่างการเข้าสู่ระบบผ่านทางบัญชีของ Google หรือ Facebook ซึ่งจะไม่เกิดความยุ่งยากแก่ผู้ใช้งาน ทำให้สามารถล็อกอินได้สะดวก และไม่ยุ่งยากในการเข้าใช้งานระบบ โดยถ้าเป็นผู้ใช้ที่เคยลงทะเบียนชื่อในระบบเอาไว้แล้ว จะไม่แสดงผลหน้านี้ให้ผู้ใช้เห็น แต่จะทำการเข้าสู่ระบบให้โดยอัตโนมัติ และเมนูบนด้านซ้ายจะเป็นปุ่มเมนูเพื่อให้ผู้ใช้สามารถเลือกสีธีมของแอปพลิเคชันได้

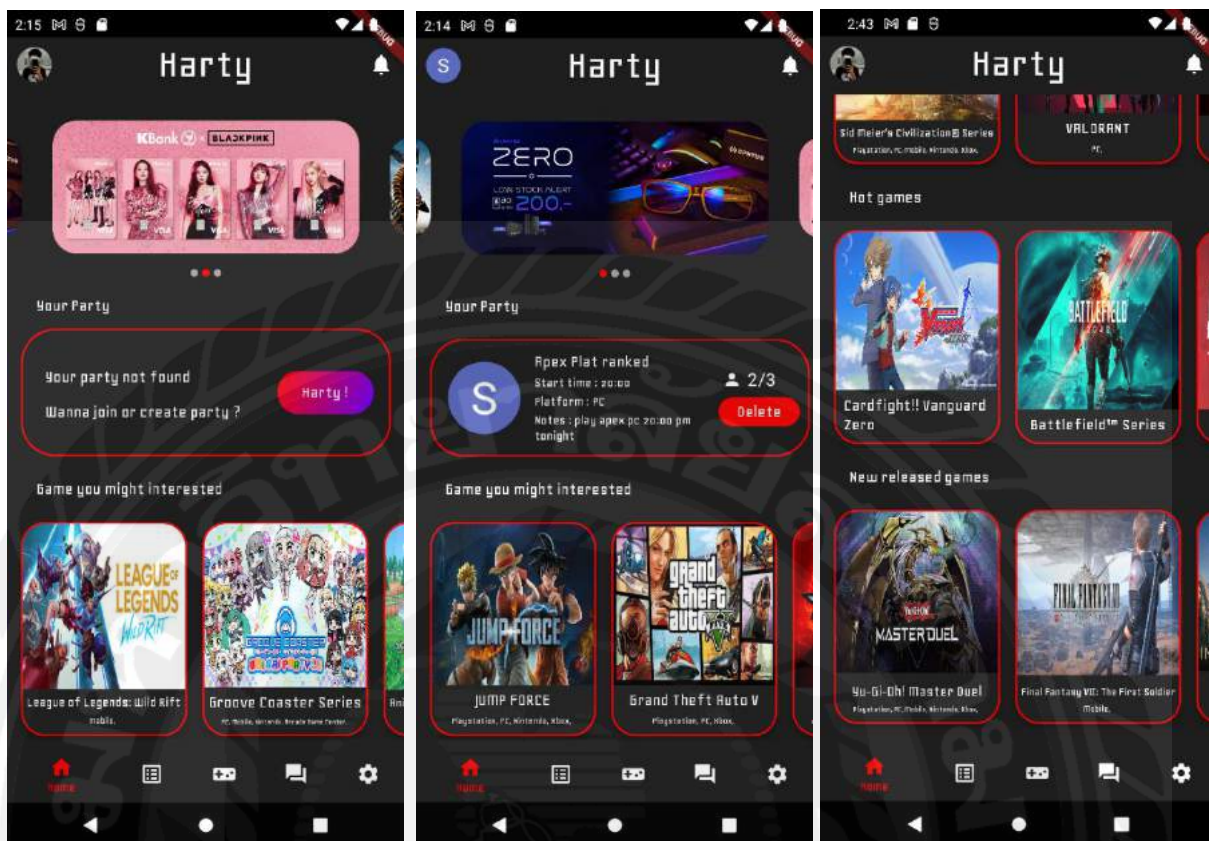
รูปที่ 4.5 หน้าตั้งค่าโปรไฟล์ผู้ใช้งานใหม่

จากรูปที่ 4.5 หน้าลงทะเบียนสำหรับผู้ใช้งานใหม่ เพื่อเป็นข้อมูลของผู้ใช้เพื่อระบุถึงตัวตน สำหรับการใช้งานในระบบ โดยจะมีช่องสำหรับกรอกข้อมูลผู้ใช้ ดังนี้ ชื่อผู้ใช้ อายุ เพศ ข้อความแนะนำตัว เวลาที่สะดวกเล่นเกม และช่องทางการติดต่อตามแพลตฟอร์มต่างๆ โดยมีชื่อและเงื่อนไขกำกับช่องสำหรับกรอกข้อมูลตามลำดับ และปุ่ม Next สำหรับการเข้าสู่ขั้นตอนการตั้งค่าถัดไป



รูปที่ 4.6 หน้าตั้งค่ารูปโปรไฟล์

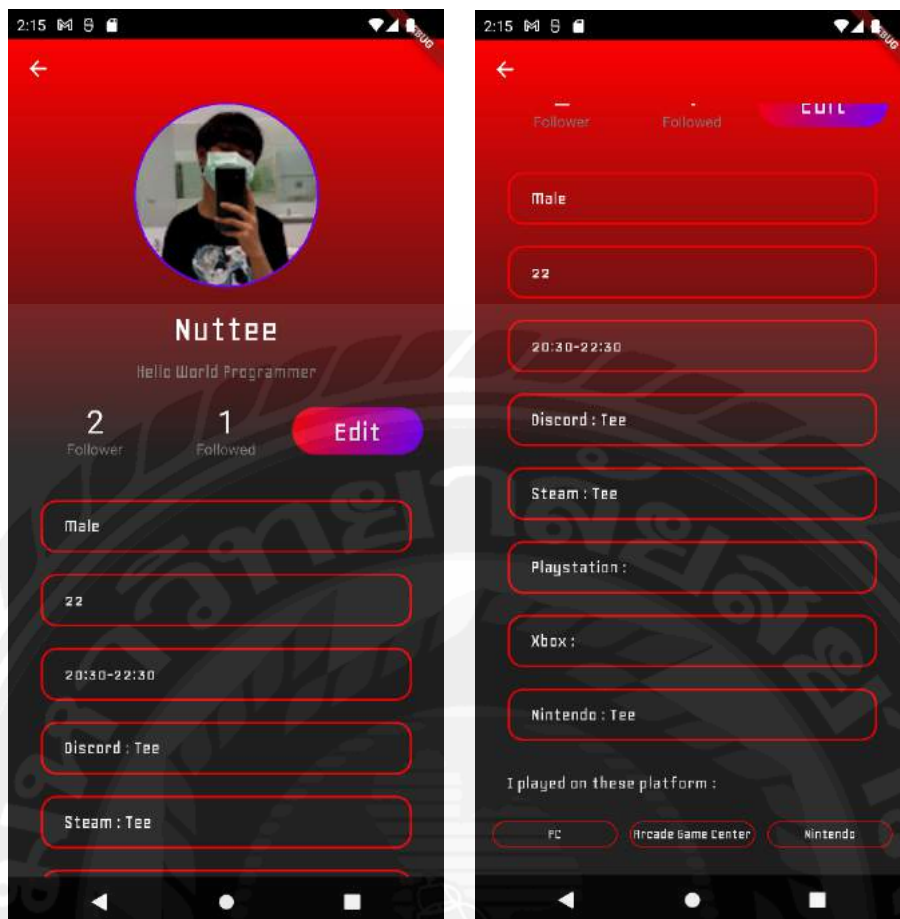
จากรูปที่ 4.6 หน้าตั้งค่ารูปโปรไฟล์ โดยในหน้าจอนี้จะให้ผู้ใช้เลือกวิธีการอัปโหลดรูปโปรไฟล์ได้ 2 วิธี โดยสามารถถ่ายรูปจากกล้องมือถือ หรือ อัปโหลดรูปจากแกลลอรี่มือถือของผู้ใช้ และอีกทั้งผู้ใช้อังสามารถเลือกที่จะข้ามขั้นตอนนี้ได้ผ่านปุ่ม Skip โดยที่ระบบจะทำการดึงรูปของผู้ใช้มาโดยอัตโนมัติผ่านบัญชีที่ผู้ใช้ใช้ในการเข้าสู่ระบบ (Google หรือ Facebook)



รูปที่ 4.7 หน้า Home Feed แสดงหน้าจอหลักของแอปพลิเคชัน

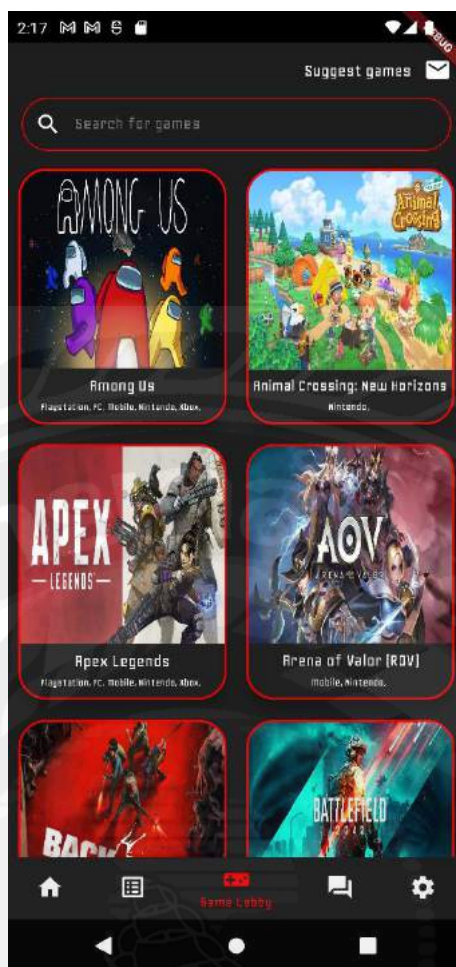
จากรูปที่ 4.7 หน้า Home หรือหน้าจอหลัก โดยแสดงเนื้อหา (Contents) ต่างๆ ประกอบด้วย

- รูปโปรไฟล์ผู้ใช้ (สามารถกดเพื่อเข้าไปดูโปรไฟล์ของผู้ใช้ได้)
- ชื่อของแอปพลิเคชัน Harty
- ปุ่มระฆังสำหรับการแจ้งเตือน (สามารถกดไปยังหน้า Notification ได้)
- Banner ในอนาคตจะเป็นพื้นที่สำหรับการโฆษณา
- Row แสดงห้องปาร์ตี้ต่างๆ ของผู้ใช้ (ถ้าผู้ใช้ยังไม่เคยสร้างห้องปาร์ตี้ด้วยตัวเอง ระบบจะแนะนำให้ผู้ใช้งานสามารถกดปุ่มเพื่อนำไปหน้า Game Lobby ได้)
- เกมที่ผู้ใช้อาจจะสนใจ (ระบบแนะนำตามแพลตฟอร์มที่ผู้ใช้เป็นคนเลือก)
- เกมที่เป็นที่นิยม (ระบบแนะนำเกมที่มีผู้ใช้สร้างห้องปาร์ตี้ไว้เป็นจำนวนมาก)
- เกมที่เพิ่งวางจำหน่าย (ระบบแนะนำเกมที่มีวันวางจำหน่ายล่าสุดตามวันที่ในระบบของผู้ใช้)
- แถบ Bottom Bar ด้านล่าง จะเป็นปุ่มนำทางไปหน้าจอต่างๆ ถัดไป



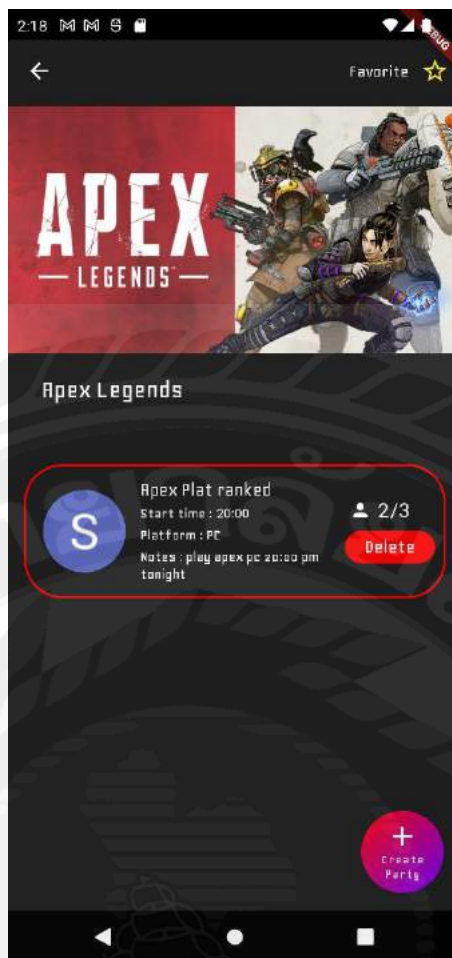
รูปที่ 4.8 หน้า Profile

จากรูปที่ 4.8 หน้า Profile แสดงข้อมูลของผู้ใช้ ในหน้าจอนี้จะแสดงข้อมูลของผู้ใช้ใน ระบบ และสามารถกดปุ่มแก้ไขข้อมูลของผู้ใช้ได้ แต่ถ้าเป็นหน้าโปรไฟล์ของผู้ใช้คนอื่นปุ่มนี้จะ เป็นปุ่ม Follow / Unfollow เพื่อกดติดตามผู้ใช้นั้น สำหรับรับการแจ้งเตือนและแชทกับผู้ใช้นั้น ได้ โดยในหน้าจอนี้จะแสดงข้อมูลประกอบด้วย รูปโปรไฟล์ผู้ใช้ ชื่อความแนะนำตัว จำนวน ผู้ติดตาม และผู้ที่ติดตาม ชื่อผู้ใช้ เพศ อายุ เวลาที่สะดวก ข้อมูลการติดต่อตามแพลตฟอร์มต่างๆ และ ข้อมูลแพลตฟอร์มที่ผู้ใช้เล่น



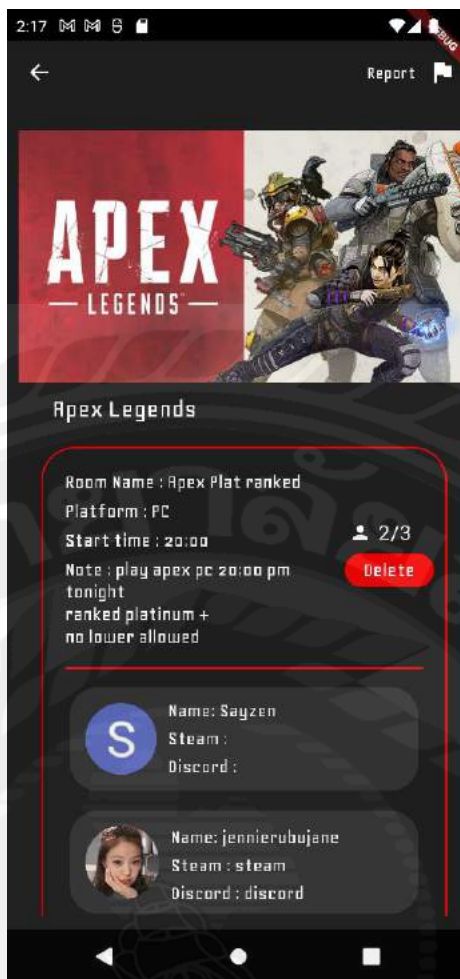
รูปที่ 4.9 หน้า Game Lobby

จากรูปที่ 4.9 หน้า Activity Detail สำหรับแสดงรายละเอียดกิจกรรม ในหน้าจอนี้จะแสดงปุ่มย้อนกลับบนด้านซ้ายของหน้าจอ ส่วนด้านขวาคือสัญลักษณ์ตัวเลือกเพิ่มเติมซึ่งประกอบไปด้วย 2 ตัวเลือก คือ 1. แก้ไขกิจกรรมและลบกิจกรรม และ 2. แสดงรูปภาพของกิจกรรม ชื่อของกิจกรรม ปุ่มสัญลักษณ์หัวใจ สำหรับให้ผู้ใช้งานกดถูกใจกิจกรรมนั้นได้ และจะเป็นรายละเอียดกิจกรรมต่างๆ มีจำนวน 9 รายการ ดังนี้ วันที่กิจกรรมสิ้นสุด รายละเอียดกิจกรรม รูปโปรไฟล์ผู้สร้างกิจกรรม ชื่อผู้สร้างกิจกรรม เพศของผู้เข้าร่วมกิจกรรม จำนวนคนที่ต้องการเข้าร่วมกิจกรรม ช่วงอายุที่เข้าร่วมกิจกรรม สถานที่ทำกิจกรรม ประเภทของกิจกรรม และปุ่ม Join สำหรับส่งคำขอเข้าร่วมกิจกรรมไปยังผู้สร้างกิจกรรม



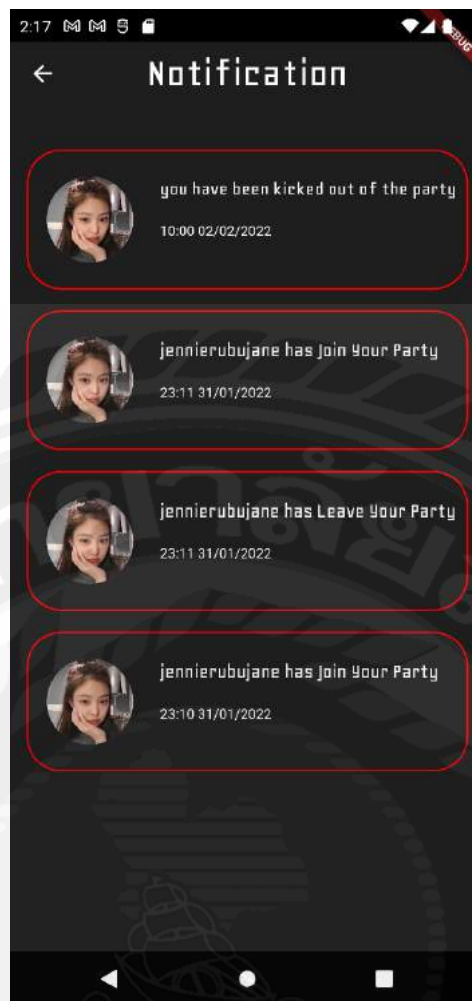
รูปที่ 4.10 หน้า Game Room

จากรูปที่ 4.10 หน้า Edit Event สำหรับแก้ไขกิจกรรมโดยผู้สร้างกิจกรรมนั้นๆ ในหน้าจอนี้ จะแสดงข้อความหัวข้อเพื่อบ่งบอกว่าเป็นหน้าแก้ไขกิจกรรม จะมีสัญลักษณ์ก่อกองสำหรับใช้งาน นำรูปกิจกรรมมาใส่ ต่อมาจะเป็นการแสดงรูปภาพกิจกรรม ปุ่ม Change Image สำหรับผู้ใช้งานที่ต้องการเปลี่ยนรูปภาพกิจกรรม และรายละเอียดต่างๆ ดังนี้ ชื่อกิจกรรม คำอธิบายกิจกรรม จำนวน คนที่ต้องการเข้าร่วมกิจกรรม ช่วงอายุที่เข้าร่วมกิจกรรม วันที่กิจกรรมเริ่มต้น ช่วงเวลา สถานที่ทำ กิจกรรม เลือกวันที่กิจกรรมสิ้นสุด เลือกเพศที่ต้องการเข้าร่วมกิจกรรม เลือกประเภทกิจกรรม และ ปุ่ม Edit สำหรับยืนยันการแก้ไขกิจกรรม



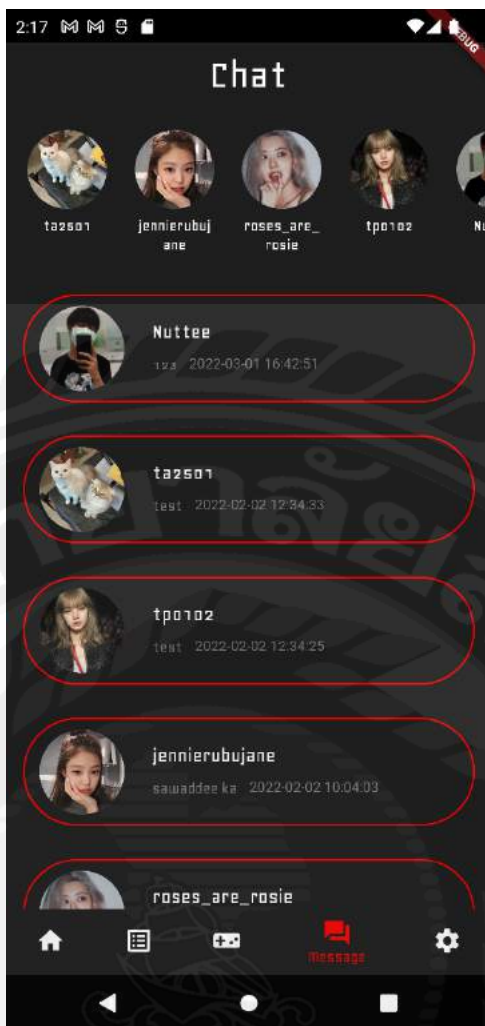
รูปที่ 4.11 หน้า Party's Detail

จากรูปที่ 4.11 หน้า Party's Detail โดยจะแสดงข้อมูลรายละเอียดในกิจกรรมนั้นๆ ไม่ว่าจะเป็นเจ้าของปาร์ตี้สร้างไว้เอง หรือเป็นกิจกรรมที่ผู้ใช้คนอื่นสร้างไว้ ในหน้านี้จะแสดงรายละเอียดต่างๆ ได้แก่ ชื่อห้อง แพลตฟอร์มที่เลือก เวลาเริ่มกิจกรรม รายละเอียดอื่นๆ รายชื่อสมาชิกของคนที่ร่วมกิจกรรม และจำนวนสมาชิกในกิจกรรม โดยถ้าผู้ใช้เป็นผู้สร้างกิจกรรมนั้นจะมีปุ่ม Delete โดยเมื่อกดปุ่มแล้วจะสามารถทำการลบกิจกรรมที่สร้างไว้ได้ กิจกรรมที่ถูกลบจะหายไปจากรายชื่อกิจกรรมในหน้า Game Room ทันที



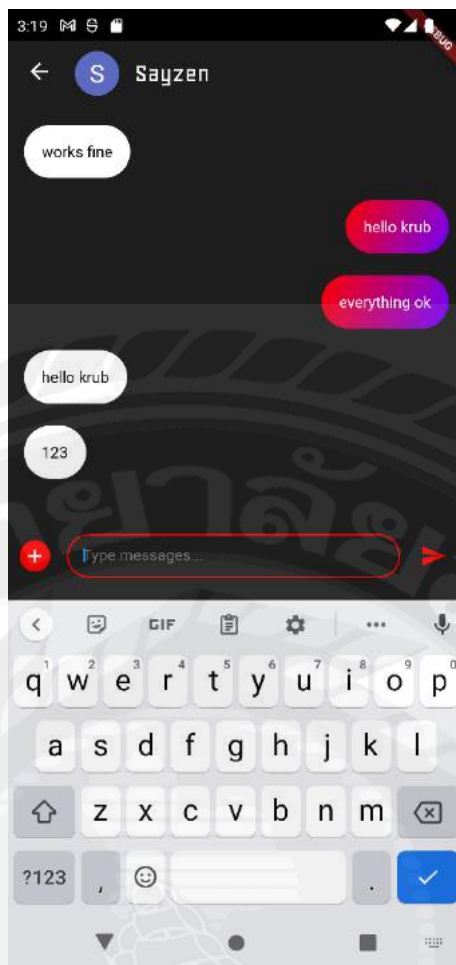
รูปที่ 4.12 หน้า Notification

จากรูปที่ 4.12 หน้า Notification แสดงการแจ้งเตือนต่างๆ ให้ผู้ใช้ในระบบได้รับรู้ถึงการเปลี่ยนแปลงที่เกิดขึ้นในระบบ โดยจะแสดงการแจ้งเตือนเมื่อ มีผู้เข้าร่วมปาร์ตี้ที่ผู้ใช้สร้างไว้ มีผู้เล่นออกจากปาร์ตี้ และ เจ้าของห้องทำการลบผู้ใช้ออกจากปาร์ตี้



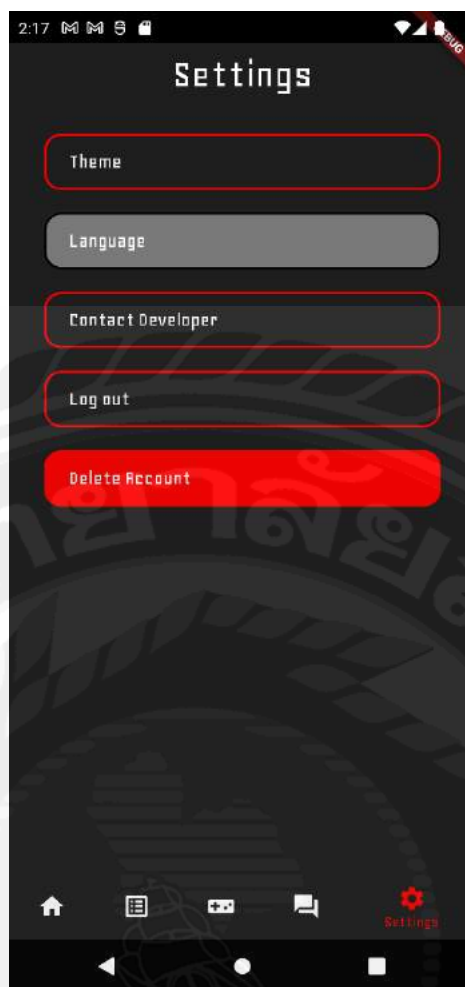
รูปที่ 4.13 หน้า Chat List

จากรูปที่ 4.13 หน้า Chat List เป็นหน้าจอแสดงรายชื่อผู้ติดตามที่ผู้ใช้ได้กด Follow ไว้ในระบบในหน้าโปรไฟล์ของผู้ใช้คนอื่น และแสดงรายการแชทข้อความที่ผู้ใช้เคยส่งถึงผู้ใช้คนอื่นโดยเรียงลำดับตามวันที่และเวลาจากใหม่ล่าสุดอยู่ด้านบนและเก่าสุดอยู่ด้านล่าง



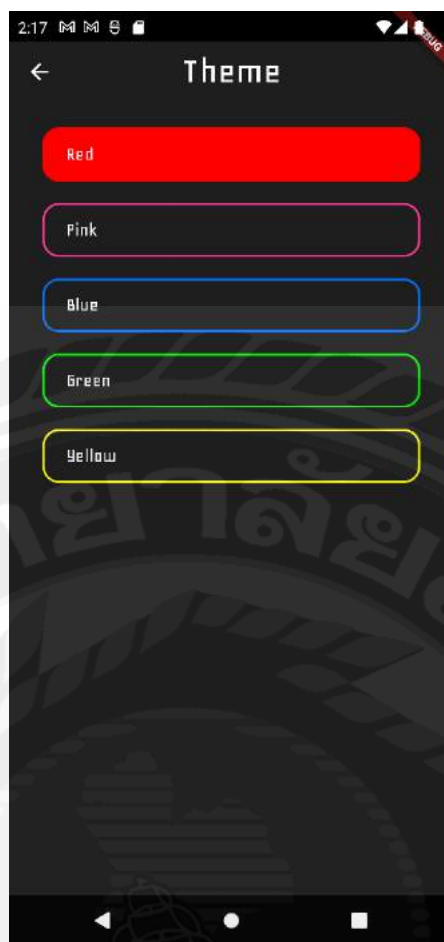
รูปที่ 4.14 หน้า Chat

จากรูปที่ 4.14 หน้า Chat เป็นหน้าจอไว้สำหรับให้ผู้ใช้ส่งข้อความถึงผู้ใช้คนอื่นในระบบ โดยจะแบ่งเป็นฝั่งด้านซ้ายจะแสดงข้อความของผู้ใช้คนที่กำลังคุยด้วย และฝั่งด้านขวาจะเป็นข้อความที่ผู้ใช้เป็นคนพิมพ์ข้อความส่งไป เมื่อพิมพ์ข้อความในช่องกรอกข้อความเสร็จแล้ว ต้องทำการกดปุ่ม Send ที่อยู่ทางด้านล่างขวาเพื่อเป็นการยืนยันการส่งข้อความ



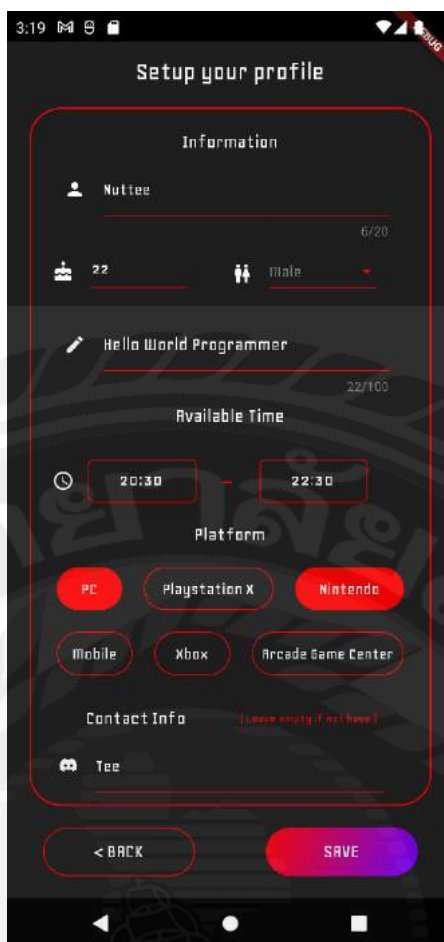
รูปที่ 4.15 หน้า Settings

จากรูปที่ 4.15 หน้า Settings เป็นหน้าจอแสดงถึงการตั้งค่าต่างๆ โดยจะแบ่งเป็นหัวข้อให้ผู้ใช้ทำการเลือกเข้าไปหน้าตั้งค่าตามหัวข้อต่างๆ โดยในที่นี้จะมีการตั้งค่าดังนี้ ธีม ภาษา (ยังไม่รองรับ ณ ขณะนี้) การติดต่อผู้พัฒนา การออกจากระบบ และการลบบัญชีผู้ใช้



รูปที่ 4.16 หน้า Theme Setting

จากรูปที่ 4.16 หน้า Theme Setting เป็นหน้าจอแสดงผลการตั้งค่าธีมสี โดย ณ ขณะนี้ผู้ใช้สามารถเลือกการตั้งค่าได้ 5 สีตามหัวข้อ โดยมีสีดังนี้ คือ สีแดง สีชมพู สีฟ้า สีเขียว และ สีเหลือง



รูปที่ 4.17 หน้า Edit Profile

จากรูปที่ 4.17 หน้า Edit Profile สำหรับให้ผู้ใช้แก้ไขข้อมูลของตนเอง ในหน้าจอนี้จะแสดงข้อความเพื่อบ่งบอกว่าเป็นหน้าแก้ไขโปรไฟล์ผู้ใช้งาน ต่อมาจะมีปุ่ม Save สำหรับบันทึกการแก้ไขข้อมูล รายละเอียดที่แสดงบนหน้าจอ ประกอบด้วย ชื่อผู้ใช้งาน อายุ เพศ คำแนะนำตัว เวลาที่สะดวกแพลตฟอร์ม และ ข้อมูลการติดต่อ และผู้ใช้สามารถเลือกแก้ไขข้อมูลต่างๆ ได้ และมีปุ่ม Back สำหรับการยกเลิกการแก้ไข และกลับไปสู่หน้าโปรไฟล์

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลปริญญานิพนธ์

จากการพัฒนาแอปพลิเคชัน Harty เพื่อเป็นสื่อกลางสำหรับสังคมออนไลน์สำหรับเกมเมอร์อีกแหล่งหนึ่งที่จะช่วยเพิ่มโอกาสในการนำพากลุ่มคนที่มีความชื่นชอบเกมเหมือนกันมาพบปะและรวมตัวกันในการเล่นและแลกเปลี่ยนความรู้ทางด้านเกม โดยแอปพลิเคชัน Harty จะเป็นโมบายแอปพลิเคชันที่ผู้ใช้งานสามารถสร้างปาร์ตี้เกม และนัดหมายผู้เล่นที่มีความชื่นชอบเหมือนกัน มีเวลาว่างตรงกัน มาร่วมกันเล่นเกม ก่อให้เกิดมิตรภาพใหม่ๆ และสามารถค้นหาประสบการณ์ผ่านสังคมออนไลน์ได้ง่ายและสะดวก โดยเฉพาะอย่างยิ่งในยุคปัจจุบันที่มีสถานการณ์การแพร่ระบาดของเชื้อไวรัส Covid-19 ทำให้ผู้คนที่ชื่นชอบเล่นเกม อาจจะไม่ค่อยพบปะสังสรรค์ในชีวิตจริง แต่แอปพลิเคชัน Harty สามารถทำให้ผู้ใช้สามารถหาเพื่อนมาเล่นเกมด้วยกันได้ หรือค้นหาเกมใหม่ๆ เล่นกับเพื่อนคนใหม่ๆ ได้ เป็นต้น โดยฟังก์ชันการทำงานต่างๆ พัฒนาเพื่อรองรับการใช้งานให้ตรงตามความต้องการของผู้ใช้ให้ได้มากที่สุด และระบบยังรองรับการเติบโตของผู้เล่น ลดภาระการดูแลรักษาระบบ โดยเลือกใช้เทคโนโลยีคลาวด์คอมพิวเตอร์และฐานข้อมูล NoSQL นอกจากนี้ยังรองรับการใช้งานทั้งแพลตฟอร์มแอนดรอยด์และไอโอเอส โดยการเขียนชุดคำสั่งเพียงครั้งเดียวทำให้ทั้ง 2 แพลตฟอร์มมีหน้าจอรูปร่างและรูปแบบการใช้งานเหมือนกัน ผู้ใช้จะไม่รู้สึกแตกต่างเมื่อมีการใช้งานอีกแพลตฟอร์มหนึ่ง

5.2 ข้อดีของระบบ

- 5.2.1 ผู้ใช้สามารถสร้างปาร์ตี้เกมเพื่อหาเพื่อนร่วมเล่นเกมได้
- 5.2.2 ผู้ใช้สามารถเข้าร่วมเกมที่ตนเองสนใจหรือชื่นชอบได้ตลอดเวลา
- 5.2.3 ผู้ใช้งานสามารถพูดคุยและแลกเปลี่ยนข้อมูลที่สนใจร่วมกันเกี่ยวกับเกมนั้นๆ เพื่อสร้างมิตรภาพใหม่ๆ ได้
- 5.2.4 ผู้ใช้งานมีโอกาสได้ร่วมเล่นเกมใหม่ๆ ที่ยังไม่เคยเล่นมาก่อน
- 5.2.5 เป็นสังคมออนไลน์ที่มีความเฉพาะตัว มุ่งเน้นการเล่นโดยเฉพาะ
- 5.2.6 ทำให้คนที่ไม่มีเพื่อนสามารถหาเพื่อนในแอปพลิเคชัน Harty ได้ ซึ่งอาจจะเป็นหนทางหนึ่งที่จะลดจำนวนผู้ป่วยด้วยโรคซึมเศร้าได้

5.3 ข้อเสนอแนะ

เพื่อให้ระบบมีประสิทธิภาพมากยิ่งขึ้น ควรมีการพัฒนาฟังก์ชันการทำงานเพิ่มเติม ดังนี้

- 5.3.1 พัฒนาระบบการแนะนำเกมที่ตรงกับความชอบหรือความต้องการผู้ใช้ให้มีประสิทธิภาพมากขึ้น โดยเลือกใช้อัลกอริทึมใหม่ๆ หรือมีการเก็บข้อมูลที่จำเป็นเพิ่มเติม เป็นต้น
- 5.3.2 พัฒนาการปรับปรุงแก้ไขข้อมูลมีความเรียลไทม์ (Real Time) ที่รวดเร็วยิ่งขึ้น ระหว่าง Firebase และฝั่งแอปพลิเคชันบนอุปกรณ์โทรศัพท์มือถือ
- 5.3.3 มีระบบกรองคำพูด หรือผู้เล่นที่ประพฤติไม่เหมาะสมในสังคมเกม



บรรณานุกรม

เน็ตโฟลว์. (2559). 10 ปีผ่านมา การพัฒนาโปรแกรมแบบ *cross platform* ถึงไหนแล้ว ?. เข้าถึงได้

จาก <https://nextflow.in.th/2016/10-year-cross-platform-application-development-thai/>

บล็อกสปอต (2556, 18 พฤศจิกายน). เกมออนไลน์คืออะไร ? [เว็บไซต์]. เข้าถึงได้จาก

<http://wananurat02472.blogspot.com/2013/11/game-online.html>

โฟว์เอ็กซ์ตรีม (2563, 20 พฤศจิกายน). Firebase คืออะไร [เว็บไซต์]. เข้าถึงได้จาก

<https://www.4xtreme.com/2020/11/20/firebase-คืออะไร/>

รัชกร สุรพัชรเดชา (2563, 11 เมษายน). Flutter : Google Signin ด้วย Firebase Auth และจัดเก็บ

ข้อมูล [เว็บไซต์]. เข้าถึงได้จาก <https://medium.com/@ratchakorn.s/flutter-google-signin->

[ด้วย-firebase-authและจัดเก็บข้อมูลพื้นฐาน-user-ลง-cloud-firestore-d37558b792ad](https://medium.com/@ratchakorn.s/flutter-google-signin-)

อิโซกะ. (2561, 22 มิถุนายน). มาทำความรู้จักกับ *Flutter* กันเถอะ [เว็บไซต์]. เข้าถึงได้จาก

<https://medium.com/@hizokaz/มาทำความรู้จักกับ-flutter-กันเถอะ-4dca2ad634bd>