



การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวตีฟ
ด้วยไอโอไนคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส

ศศิธร ประสงค์

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2560
ลิขสิทธิ์เป็นของมหาวิทยาลัยอุบลราชธานี



THE DEVELOPMENT OF INFORMATION SEARCH SYSTEM
IN WIKIPEDIA AND CONSERVAPEDIA USING IONIC FRAMEWORK
WITH JSON AND CACHE ON IOS OPERATING SYSTEM

SASITON PRASONG

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE
MAJOR IN INFORMATION TECHNOLOGY
FACULTY OF SCIENCE
UBON RATCHATHANI UNIVERSITY
ACADEMIC YEAR 2017
COPYRIGHT OF UBON RATCHATHANI UNIVERSITY



ใบรับรองการค้นคว้าอิสระ
มหาวิทยาลัยอุบลราชธานี
ปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

เรื่อง การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวตฟายโอเอไอเคเฟรมเวิร์ก
เจสันและแคชบนระบบปฏิบัติการไอโอเอส

ผู้วิจัย นายศศิธร ประสงค์

คณะกรรมการสอบ

ดร.ณัฐ ดิษเจริญ

ดร.สมปอง เวฬุวนาธร

ดร.ชัชวิน นามมัน

ประธานกรรมการ

กรรมการ

กรรมการ

อาจารย์ที่ปรึกษา

(ดร.สมปอง เวฬุวนาธร)

(ผู้ช่วยศาสตราจารย์ ดร.ชริตา ปุกหุด)

คณบดีคณะวิทยาศาสตร์

(รองศาสตราจารย์ ดร.อริยาภรณ์ พงษ์รัตน์)

รองอธิการบดีฝ่ายวิชาการ

ลิขสิทธิ์เป็นของมหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2560

กิตติกรรมประกาศ

การศึกษาคำว่าอิสระนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับคำแนะนำในการศึกษาวิจัยและออกแบบการทดลองจากอาจารย์ที่ปรึกษาและคณาจารย์ในสาขาเทคโนโลยีสารสนเทศทุกท่าน รวมทั้งบทความวิจัยที่เกี่ยวข้อง ขอกราบขอบพระคุณ ดร.สมปอง เวฬุวนาธร อาจารย์ที่ปรึกษา ดร.ณัฐ ดิษเจริญ ประธานหลักสูตรวิทยาศาสตรมหาบัณฑิตสาขาวิชาเทคโนโลยีสารสนเทศ คณะอาจารย์ประจำหลักสูตรเทคโนโลยีสารสนเทศที่ถ่ายทอดความรู้และให้คำแนะนำตลอดหลักสูตรของการเรียน และการให้แนวคิดต่างๆ ในการศึกษา คอยอบรมแนะนำสั่งสอนทั้งในเรื่องการเรียน และเรื่องคุณธรรมจริยธรรม

ขอกราบขอบพระคุณบิดา-มารดา ที่คอยให้กำลังใจในการศึกษาอยู่เสมอและทำดีที่สุดขอขอบคุณเพื่อนนักศึกษาปริญญาโทสาขาเทคโนโลยีสารสนเทศ ที่ช่วยเหลือ แนะนำ ให้ประสบการณ์ ความรู้ และกำลังใจจนสามารถทำให้การศึกษาค้นคว้าอิสระนี้สำเร็จลุล่วงไปได้ด้วยดี

ศศิธร ประสงค์

ผู้วิจัย

บทคัดย่อ

เรื่อง : การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีเดีย
ด้วยไอโอไนคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส

ผู้วิจัย : ศศิธร ประสงค์

ชื่อปริญญา : วิทยาศาสตรมหาบัณฑิต

สาขาวิชา : เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษา: ดร.สมปอง เวฬุวนาร

คำสำคัญ : การสืบค้นข้อมูล, วิกิพีเดีย, คอนเซอร์เวทีเดีย, เจสัน, แคช

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาและประเมินประสิทธิภาพระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีเดียบนระบบปฏิบัติการไอโอเอส (iOS) ซึ่งพัฒนาโดยใช้ไอโอไนคเฟรมเวิร์ก (Ionic Framework) และฐานข้อมูลเอสคิวไลต์ (SQLite) ใช้เทคนิคการทำแคชบนมือถือ (Cache Mobile) และแคชบนเซิร์ฟเวอร์ (Cache Server) เข้ามาช่วยเพิ่มประสิทธิภาพในการสืบค้น การทำแคชจะเก็บเป็นไฟล์รูปแบบเจสัน (JSON) เพื่อช่วยเพิ่มประสิทธิภาพความเร็วของการแลกเปลี่ยนข้อมูลระหว่างไคลเอนต์ (Client) และเซิร์ฟเวอร์ (Server) โดยในการวัดประสิทธิภาพความเร็วในการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีเดียพบว่าการสืบค้นในวิกิพีเดียและคอนเซอร์เวทีเดียแบบผ่านเว็บไซต์ของคอนเซอร์เวทีเดียโดยตรงไม่ผ่านแคชใช้เวลาตอบสนองช้าที่สุดเฉลี่ย 826 มิลลิวินาทีและการสืบค้นแบบผ่านแคชบนมือถือใช้เวลาตอบสนองเร็วที่สุดเฉลี่ย 9 มิลลิวินาที ซึ่งมีความเร็วเพิ่มขึ้นเฉลี่ยร้อยละ 98.91 ของการสืบค้น ในส่วนของการวัดประสิทธิภาพการแสดงผลหน้ารายละเอียดพบว่าการแสดงผลข้อมูลวิกิพีเดียแบบผ่านเว็บไซต์ของคอนเซอร์เวทีเดียโดยตรงไม่ผ่านแคชใช้เวลาตอบสนองช้าที่สุดเฉลี่ย 2204 มิลลิวินาทีและการแสดงผลหน้ารายละเอียดแบบผ่านแคชบนมือถือใช้เวลาตอบสนองเร็วที่สุดเฉลี่ย 30 มิลลิวินาที ซึ่งมีความเร็วเพิ่มขึ้นเฉลี่ยร้อยละ 98.64 ของการแสดงผลหน้ารายละเอียด การประเมินความพึงพอใจของผู้ใช้ทั่วไปจำนวน 15 คนพบว่าอยู่ในระดับดี โดยได้ค่าเฉลี่ยเฉลี่ย 4.12 และส่วนเบี่ยงเบนมาตรฐานเฉลี่ย 0.63 การประเมินความพึงพอใจของผู้เชี่ยวชาญจำนวน 5 คน พบว่าอยู่ในระดับดี โดยได้ค่าเฉลี่ยเท่ากับ 3.90 และส่วนเบี่ยงเบนมาตรฐานเฉลี่ย 0.56 ดังนั้นจึงสามารถสรุปได้ว่าระบบที่พัฒนาขึ้นสามารถนำไปใช้ในการสืบค้นข้อมูลบนวิกิพีเดียและคอนเซอร์เวทีเดียได้จริง การใช้รูปแบบการแลกเปลี่ยนข้อมูลแบบเจสันและเทคนิคการทำแคชมาช่วยทำให้การสืบค้นข้อมูลมีประสิทธิภาพเพิ่มมากขึ้น

ABSTRACT

TITLE : THE DEVELOPMENT OF INFORMATION SEARCH SYSTEM
IN WIKIPEDIA AND CONSERVAPEDIA USING IONIC FRAMEWORK
WITH JSON AND CACHE ON IOS OPERATING SYSTEM

AUTHOR : SASITON PRASONG

DEGREE : MASTER OF SCIENCE

MAJOR : INFORMATION TECHNOLOGY

ADVISOR : SOMPONG VALUVANATHORN, Ph.D.

KEYWORDS : INFORMATION SEARCH, WIKIPEDIA, CONSERVAPEDIA, JSON, CACHING

This research aims to develop and evaluate the efficiency of the information search system in Wikipedia and Conservapedia on iOS operating system. This system is developed using the Ionic Framework and SQLite database. There are caching techniques both cache on mobile and cache on server to improve search efficiency. Caching is stored as a JSON format to enhance the speed of data exchange between the client and server. The measuring a response time of search in Wikipedia and Conservapedia shows that the response time of search in Conservapedia without cache used most is equal to 826 millisecond and the response time of search in Wikipedia and Conservapedia via cache on mobile used least is equal to 9 millisecond, the results show that the response time speed up to 98.91 percentage. In the part of measuring a response time of details page in Wikipedia and Conservapedia, the result indicted that the response time of details page in Conservapedia without cache used most is equal to 2204 millisecond and the response time of details page in Wikipedia via cache on mobile used least is equal to 30 millisecond, The results show that the response time speed up to 98.64 percentage. Furthermore, the five-point Likert scales questionnaire is used to evaluate satisfaction from 15 users. The results reveal that is good level (the average is equal to 4.12 and SD is equal to 0.63). And evaluate satisfaction from 5 experts the results is good level (the average is equal to 3.90 and SD is equal to 0.56). Overall, this shows that the proposed system can be used effectively.

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพ	ช
บทที่ 1 บทนำ	
1.1 ความเป็นมาของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของการค้นคว้าอิสระ	2
1.4 เครื่องมือที่ใช้ในการพัฒนา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	
2.1 ความรู้พื้นฐานเกี่ยวกับวิกิพีเดีย	4
2.2 ความรู้พื้นฐานเกี่ยวกับคอนเซอร์เวทีฟ	5
2.3 ความรู้พื้นฐานเกี่ยวกับเจสัน	5
2.4 ความรู้พื้นฐานเกี่ยวกับแคช	9
2.5 ความรู้พื้นฐานเกี่ยวกับไอโอไนคเฟรมเวิร์ก	10
2.6 งานวิจัยที่เกี่ยวข้อง	13
บทที่ 3 วิธีดำเนินการวิจัย	
3.1 การออกแบบสถาปัตยกรรมแคชบนมือถือและแคชบนเซิร์ฟเวอร์	15
3.2 การออกแบบระบบด้วยแผนภาพการไหลของกระแสข้อมูล (Data Flow Diagram)	19
3.3 การออกแบบฐานข้อมูลด้วยแผนภาพความสัมพันธ์ของข้อมูล (E-R Diagram)	24
3.4 การออกแบบพจนานุกรมข้อมูล (Data Dictionary)	26
3.5 การออกแบบหน้าจอ (Graphic User Interface)	28
3.6 การออกแบบการวิเคราะห์และประเมินผล	30

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการดำเนินการวิจัย

4.1 ผลการพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวียมพีเดีย	32
4.2 ผลการทดสอบระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวียมพีเดีย	45
4.3 ผลการวัดประสิทธิภาพความเร็วในการสืบค้นข้อมูล	55
4.4 ผลการวัดค่าความถูกต้องของระบบ	59
4.5 ผลการประเมินความพึงพอใจ	59

บทที่ 5 สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการวิจัย	62
5.2 ข้อเสนอแนะ	63

เอกสารอ้างอิง

64

ภาคผนวก

ก การติดตั้งและใช้งานแอปพลิเคชัน	68
ข แบบสอบถามความพึงพอใจ	83
ค บทความวิจัย	88

ประวัติผู้วิจัย

98

สารบัญตาราง

ตารางที่		หน้า
3.1	สัญลักษณ์แผนภาพกระแสข้อมูลแบบ Gane & Sarson	19
3.2	สัญลักษณ์ในการออกแบบ ER Diagram ในรูปแบบ Crow’s Foot Model	24
3.3	พจนานุกรมแคชบนมือถือ (Cache_Mobile)	26
3.4	พจนานุกรมประวัติการสืบค้น (History_Search)	26
3.5	พจนานุกรมแคชบนเซิร์ฟเวอร์ (Cache_Server)	26
3.6	พจนานุกรมตั้งค่า (Settings)	27
3.7	พจนานุกรมข้อมูลผู้ใช้ (User_Profile)	27
4.1	การทดสอบการสืบค้นข้อมูลจากวิกิพีเดีย	45
4.2	การทดสอบการเก็บแคชบนมือถือ	47
4.3	การทดสอบลบแคชบนมือถือ	50
4.4	การทดสอบเก็บแคชบนเซิร์ฟเวอร์	51
4.5	การทดสอบลบแคชจากเซิร์ฟเวอร์	54
4.6	ผลการหาค่าเฉลี่ยการสืบค้นบนวิกิพีเดีย	56
4.7	ผลการหาค่าเฉลี่ยการสืบค้นบนคอนเซอร์เวพีเดีย	56
4.8	ผลสรุปการหาค่าเฉลี่ยการสืบค้นบนวิกิพีเดียและคอนเซอร์เวพีเดีย	56
4.9	ผลการหาค่าเฉลี่ยการแสดงรายละเอียดชื่อเรื่องบนวิกิพีเดีย	57
4.10	ผลการหาค่าเฉลี่ยการแสดงรายละเอียดชื่อเรื่องบนคอนเซอร์เวพีเดีย	58
4.11	สรุปการหาค่าเฉลี่ยการแสดงรายละเอียดชื่อเรื่องบนวิกิพีเดียและคอนเซอร์เวพีเดีย	58
4.12	ผลสรุปค่าความถูกต้องของผลการสืบค้นข้อมูลในกับวิกิพีเดียและคอนเซอร์เวพีเดีย	59
4.13	เกณฑ์คะแนนเฉลี่ยของแบบประเมิน	60
4.14	ผลการประเมินความพึงพอใจต่อการใช้งานระบบของผู้ใช้งานทั่วไป	60
4.15	ผลการประเมินความพึงพอใจต่อการใช้งานระบบของผู้เชี่ยวชาญ	61

สารบัญภาพ

ภาพที่		หน้า
2.1	รูปแบบข้อมูลเจสัน	5
2.2	โครงสร้างเจสันแบบออบเจกต์	6
2.3	โครงสร้างเจสันแบบอาร์เรย์	6
2.4	โครงสร้างเจสันแบบเวลู	7
2.5	โครงสร้างเจสันแบบสตริง	7
2.6	โครงสร้างเจสันแบบตัวเลข	8
2.7	โครงสร้างไฟล์เจสันของวิกิพีเดีย	8
2.8	โครงสร้างไฟล์เจสันของคอนเซอร์เวทีฟ	9
2.9	หน่วยความจำแคชและหน่วยความจำหลัก	10
2.10	สถาปัตยกรรมของ Cordova Application	11
2.11	โครงสร้างการทำงานแบบ MVC Pattern	12
3.1	สถาปัตยกรรมแคชบนมือถือ	16
3.2	สถาปัตยกรรมการทำแคชบนเซิร์ฟเวอร์	17
3.3	การออกแบบผังงานของการพัฒนาระบบ	18
3.4	แผนภาพ Context Diagram ของระบบ	20
3.5	แผนภาพ DFD Level 0 ของระบบ	21
3.6	แผนภาพ DFD Level 1 ของกระบวนการสืบค้นข้อมูล	22
3.7	แผนภาพ DFD Level 1 ของกระบวนการจัดการแคช	23
3.8	แผนภาพ DFD Level 1 ของกระบวนการจัดการประวัติการสืบค้น	23
3.9	แผนภาพ DFD Level 1 ของกระบวนการจัดการโปรไฟล์	24
3.10	แผนภาพ E-R Diagram ของระบบ	25
3.11	การออกแบบหน้าจอการสืบค้นข้อมูล	28
3.12	การออกแบบหน้าจอรายละเอียดข้อมูล	29
3.13	การออกแบบหน้าจอการตั้งค่า	30
4.1	หน้าจอการตั้งค่าแคชสำหรับการสืบค้นข้อมูล	32
4.2	ส่วนของโปรแกรมในการตั้งค่าแคชสำหรับการสืบค้นข้อมูล	33
4.3	หน้าจอการตั้งค่าการสืบค้นแบบเพิ่มประสิทธิภาพ	34
4.4	ส่วนของโปรแกรมในการตั้งค่าการสืบค้นแบบเพิ่มประสิทธิภาพ	35

สารบัญญภาพ (ต่อ)

ภาพที่		หน้า
4.5	หน้าจอการตั้งค่าภาษาสำหรับการสืบทอด	36
4.6	ส่วนของโปรแกรมในการตั้งค่าภาษาสำหรับการสืบทอด	36
4.7	หน้าจอการตั้งค่าพื้นที่เก็บและเคลียร์แคชอัตโนมัติ	37
4.8	ส่วนของโปรแกรมในการตั้งค่าเคลียร์แคชอัตโนมัติ	38
4.9	หน้าจอการสืบทอดข้อมูลในวิกิพีเดีย และคอนเซอร์เวทีฟ	39
4.10	ส่วนของโปรแกรมการสืบทอดแบบไม่มีแคช	40
4.11	ส่วนของโปรแกรมการสืบทอดแบบผ่านแคชบนมือถือ	41
4.12	ส่วนของโปรแกรมการสืบทอดแบบผ่านแคชบนเซิร์ฟเวอร์	42
4.13	หน้าจอการจัดเก็บแคชบนเซิร์ฟเวอร์	43
4.14	ส่วนของโปรแกรมการจัดเก็บแคชบนเซิร์ฟเวอร์	44
4.15	หน้าจอการแจ้งเตือนให้กรอกข้อมูลคำค้นหา	45
4.16	หน้าจอการแจ้งเตือนการเชื่อมต่ออินเทอร์เน็ต	46
4.17	หน้าจอแสดงผลการสืบทอด	46
4.18	หน้าจอแสดงผลหน้ารายละเอียดของชื่อเรื่อง	47
4.19	หน้าจอการตั้งค่าเปิดใช้งานระบบแคชบนมือถือ	48
4.20	หน้าจอการสืบทอดและเก็บแคชบนมือถือ	48
4.21	หน้าจอการแสดงรายละเอียดและเก็บแคชบนมือถือ	49
4.22	หน้าจอการแสดงรายการแคชไฟล์ที่ถูกจัดเก็บ	49
4.23	หน้าจอการตั้งค่าลบแคชแบบอัตโนมัติ	50
4.24	หน้าจอการลบแคชด้วยตัวเองตามช่วงเวลา	51
4.25	หน้าจอการตั้งค่าเปิดใช้งานระบบแคชบนเซิร์ฟเวอร์	52
4.26	หน้าจอการสืบทอดและเก็บแคชบนเซิร์ฟเวอร์	52
4.27	หน้าจอการแสดงรายละเอียดและเก็บแคชบนเซิร์ฟเวอร์	53
4.28	หน้าจอการแสดงรายการแคชไฟล์ที่ถูกจัดเก็บ	53
4.29	หน้าจอการลบแคชบนเซิร์ฟเวอร์	54
4.30	หน้าจอแสดงแคชบนเซิร์ฟเวอร์ถูกลบออกจากระบบ	55
ผ.1	เมนู Preferences ใน Xcode	69
ผ.2	เพิ่มบัญชี Apple ID ลงใน Xcode	70

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
ผ.3	ล็อกอินบัญชี Apple ID ใน Xcode	71
ผ.4	ทำการ Build Project ในแพลตฟอร์มของ iOS	72
ผ.5	เปิดไฟล์ .xcodeproj ใน Xcode	72
ผ.6	เชื่อมต่อ iPhone เข้ากับ Mac OS	72
ผ.7	รันแอปพลิเคชันเพื่อแสดงผลบน iPhone	73
ผ.8	หน้าจอแจ้งเตือนการลงนามแอปพลิเคชันบน Xcode 7	73
ผ.9	หน้าจอเลือกโปรไฟล์เป็น Personal Team	74
ผ.10	หน้าจอ Error การลงนามแอปพลิเคชันบน Xcode 8	74
ผ.11	หน้าจอการลงนามแอปพลิเคชันบน Xcode 8	75
ผ.12	หน้าจอแจ้งเตือนความน่าเชื่อถือของ Certificate	75
ผ.13	หน้าจอการรับรองความน่าเชื่อถือของแอปพลิเคชัน	76
ผ.14	หน้าจอการสืบค้นข้อมูลสารานุกรมในวิกิพีเดียและคอนเซอร์เวทีvely	77
ผ.15	หน้าจอแสดงรายละเอียดข้อมูล	78
ผ.16	หน้าจอการตั้งค่าระบบสำหรับสืบค้นข้อมูล	79
ผ.17	หน้าจอการตั้งค่าแคชบนมือถือ	80
ผ.18	หน้าจอการเคลียร์แคชแบบแมนนวล	81
ผ.19	หน้าจอข้อมูลประวัติการสืบค้น	82
ผ.20	บทความวิจัยที่ได้รับการเผยแพร่	89
ผ.21	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	90
ผ.22	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	91
ผ.23	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	92
ผ.24	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	93
ผ.25	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	94
ผ.26	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	95
ผ.27	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	96
ผ.28	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	97
ผ.29	บทความวิจัยที่ได้รับการเผยแพร่ (ต่อ)	98

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันมีข้อมูลข่าวสารเกิดขึ้นทุก ๆ วัน ข้อมูลในเรื่องเดียวกันอาจถูกสร้างและจัดเก็บไว้ในหลาย ๆ ที่ ซึ่งแต่ละที่อาจจะมีเนื้อหาที่เหมือนกันหรือแตกต่างกัน ทำให้การเข้าถึงข้อมูลเหล่านั้นยากต่อการตรวจสอบความถูกต้องและการสืบค้นข้อมูลด้วยเสิร์จเอนจินเป็นเพียงเครื่องมือที่ใช้สำหรับการค้นหาข้อมูลหรือบทความบนเว็บไซต์ที่มีความสอดคล้องกับ "คำค้น" เท่านั้น โดยไม่ได้มีการตรวจสอบว่าบทความที่ค้นเจอนั้นถูกต้องและมีความน่าเชื่อถือหรือไม่ จากปัญหาดังกล่าวจะอย่างไรให้สามารถค้นหาข้อมูลจากแหล่งที่น่าเชื่อถือได้โดยตรงและแสดงผลข้อมูลออกมาได้ทันที รวมทั้งมีประสิทธิภาพในการสืบค้นที่รวดเร็วและทำงานบนโทรศัพท์มือถือระบบปฏิบัติการไอโอเอสได้

วิกิพีเดีย (Wikipedia) เป็นเว็บไซต์สารานุกรมเสรีที่ประกอบด้วยฐานข้อมูลเชิงสารานุกรมขนาดใหญ่และมีการอ้างอิงแหล่งที่มาที่มีความน่าเชื่อถือ รวมทั้งมีการปรับปรุงแก้ไขข้อมูลให้มีความทันสมัยอยู่ตลอดเวลา วิกิพีเดียจึงเป็นเว็บไซต์ที่ได้รับการยอมรับในด้านความถูกต้องของข้อมูลและมีจำนวนข้อมูลมหาศาลที่ครอบคลุมหลายภาษาทั่วโลก ผู้วิจัยจึงได้นำวิกิพีเดียมาเป็นฐานข้อมูลที่ใช้สำหรับการสืบค้น ทั้งนี้ผู้วิจัยเห็นว่าการสืบค้นข้อมูลสารานุกรมเพียงแหล่งเดียวอาจไม่ครอบคลุมเนื้อหาและข้อเท็จจริงทั้งหมด จึงมีแนวคิดในการใช้ข้อมูลจากเว็บไซต์สารานุกรมอื่นเพิ่มเติมเพื่อช่วยเพิ่มความชัดเจนของข้อมูลที่ได้จากการสืบค้นมากขึ้น

คอนเซอร์เวพีเดีย (Conservapedia) เป็นเว็บไซต์สารานุกรมเสรีที่มีการจัดเก็บฐานข้อมูลเชิงสารานุกรม และสามารถแก้ไขปรับปรุงข้อมูลได้เช่นเดียวกับวิกิพีเดีย ผู้วิจัยจึงได้นำคอนเซอร์เวพีเดียมาใช้เป็นฐานข้อมูลในการสืบค้นอีกหนึ่งแหล่ง

อย่างไรก็ตามในการสืบค้นข้อมูลโดยตรงจากเว็บไซต์วิกิพีเดียและคอนเซอร์เวพีเดียนั้นมีการเชื่อมต่ออินเทอร์เน็ตตลอดเวลาเนื่องจากมีการรับ-ส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ทุกครั้งที่มีการสืบค้นทำให้การแสดงผลช้าและเสียแบนด์วิดท์เป็นจำนวนมาก

เพื่อให้การสืบค้นข้อมูลมีประสิทธิภาพและเกิดความรวดเร็วในการแสดงผล ผู้วิจัยจึงได้นำเทคนิคการทำแคช (Caching) มาใช้สำหรับการจัดสรรทรัพยากรให้มีประสิทธิภาพมากขึ้น โดยลดการใช้แบนด์วิดท์ให้น้อยที่สุด นอกจากนี้ผู้วิจัยได้ใช้รูปแบบข้อมูลเจสัน (JSON) มาใช้ในการโอนถ่ายข้อมูลระหว่างไคลเอนต์กับเซิร์ฟเวอร์ ซึ่งจากการศึกษาการเปรียบเทียบประสิทธิภาพการแลกเปลี่ยนข้อมูลระหว่างเอกซ์เอ็มแอล (XML) และเจสัน (JSON) [1] พบว่าเจสันสามารถส่งข้อมูลได้รวดเร็วกว่าและรองรับการทำงานร่วมกับภาษาโปรแกรมหลายภาษา

1.2 วัตถุประสงค์

1.2.1 เพื่อพัฒนาระบบสืบค้นข้อมูลวิกิพีเดีย และ คอนเซอร์เวทีฟิเดียด้วยไอโอนิคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส

1.2.2 เพื่อประเมินความพึงพอใจของผู้ใช้ระบบสืบค้นข้อมูลวิกิพีเดีย และ คอนเซอร์เวทีฟิเดียด้วย ไอโอนิคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส ซึ่งประกอบด้วย

1.2.2.1 ความง่ายต่อการใช้งาน เช่น การออกแบบหน้าจอที่เข้าใจง่าย

1.2.2.2 ประสิทธิภาพของแอปพลิเคชัน เช่น ความเร็วในการสืบค้นและความถูกต้องของ ข้อมูล

1.3 ขอบเขตของการค้นคว้าอิสระ

1.3.1 ขอบเขตของข้อมูล

1.3.1.1 สามารถสืบค้นข้อมูลจากฐานข้อมูลของวิกิพีเดียตามคีย์เวิร์ดที่ค้นหาได้

1.3.1.2 สามารถสืบค้นข้อมูลจากฐานข้อมูลของคอนเซอร์เวทีฟิเดียตามคีย์เวิร์ดที่ค้นหาได้

1.3.2 ขอบเขตของระบบ

1.3.2.1 สามารถแสดงการสืบค้นข้อมูลจากวิกิพีเดียบนมือถือระบบปฏิบัติการไอโอเอสได้

1.3.2.2 สามารถแสดงการสืบค้นข้อมูลจากคอนเซอร์เวทีฟิเดียบนมือถือระบบปฏิบัติการ ไอโอเอสได้

1.3.2.3 สามารถเก็บคำค้นและผลการสืบค้นข้อมูลบนมือถือระบบปฏิบัติการไอโอเอสได้

1.3.2.4 สามารถแสดงคีย์เวิร์ดที่ใกล้เคียงกับคำค้นได้

1.3.2.5 สามารถกำหนดค่าคุณลักษณะเพิ่มเติมเพื่อใช้เป็นค่านำหนักรในการสืบค้นได้

1.3.2.6 สามารถแชร์ผลการสืบค้นไปยังสื่อสังคมออนไลน์ (Social Network) เช่น เฟสบุ๊ก (Facebook) ได้

1.3.2.7 สามารถกำหนดดอปชั่นในการสืบค้นจากวิกิพีเดีย หรือ คอนเซอร์เวทีฟิเดีย หรือ จากทั้ง 2 แหล่งข้อมูลได้

1.4 เครื่องมือที่ใช้ในการพัฒนา

1.4.1 ฮาร์ดแวร์ (Hardware)

1.4.1.1 โทรศัพท์มือถือระบบปฏิบัติการไอโอเอส

1.4.1.2 เครื่องคอมพิวเตอร์แมคบุ๊ก

1.4.1.3 เครื่องคอมพิวเตอร์พีซี

1.4.1.4 โฮสติ้งเซิร์ฟเวอร์

1.4.2 ซอฟต์แวร์ (Software)

1.4.2.1 Xcode

1.4.2.2 XAMPP For Windows

1.4.2.3 SQLite Database

1.4.2.4 MySQL Database

1.4.2.5 Sublime Text

1.4.2.6 Ionic Framework

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้ระบบสืบค้นข้อมูลในวิกิพีเดีย และ คอนเซอร์เวทีเดีย บนระบบปฏิบัติการไอโอเอส

1.5.2 ได้ข้อมูลสารานุกรมอ้างอิงจาก 2 แหล่งเมื่อมีการสืบค้นคำเดียวกัน โดยมีการแสดงผลจากทั้งวิกิพีเดียและคอนเซอร์เวทีเดีย

1.5.3 ลดขั้นตอนการสืบค้นแบบเดิมที่ต้องทำงานผ่านเว็บเบราว์เซอร์ (Web Browser) และ เซิร์ฟเวอร์

1.5.4 ได้รับผลการสืบค้นข้อมูลที่มีความรวดเร็ว เนื่องจากใช้การแลกเปลี่ยนข้อมูลระหว่างเครือข่ายแบบเจสัน

1.5.5 ลดการใช้แบนด์วิดท์เมื่อเปิดใช้ระบบแคชบนมือถือ ในกรณีที่สืบค้นคำเดิมที่เคยค้นมาแล้ว

1.5.6 มีระบบช่วยแนะนำคำศัพท์ที่เกี่ยวข้องในการสืบค้น โดยใช้ข้อมูลจากประวัติการสืบค้นที่ผ่านมา

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาเรื่อง การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยไอโอไนคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการโอโอเอสในครั้งนี้ ผู้วิจัยได้ศึกษากรอบทฤษฎี เอกสารและงานวิจัยที่เกี่ยวข้องโดยนำเสนอตามลำดับดังนี้

- 2.1 ความรู้พื้นฐานเกี่ยวกับวิกิพีเดีย
- 2.2 ความรู้พื้นฐานเกี่ยวกับคอนเซอร์เวทีฟ
- 2.3 ความรู้พื้นฐานเกี่ยวกับเจสัน
- 2.4 ความรู้พื้นฐานเกี่ยวกับแคช
- 2.5 ความรู้พื้นฐานเกี่ยวกับไอโอไนคเฟรมเวิร์ค
- 2.6 งานวิจัยที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานเกี่ยวกับวิกิพีเดีย

วิกิพีเดีย [2] คือ สารานุกรมออนไลน์ที่มีบทความหลายภาษามีเนื้อหาในลักษณะเสรี โดยเนื้อหาของบทความเกิดจากการร่วมกันสร้างของผู้อ่านทั่วโลก คำว่า "วิกิพีเดีย" มีที่มาจากการผสมคำระหว่างคำว่า "วิกิ" และ "เอนไซโคลพีเดีย" ซึ่งคำว่า วิกิ หมายถึง ลักษณะของเว็บไซต์ที่ร่วมกันสร้างและปรับปรุงเนื้อหา ส่วนคำว่า เอนไซโคลพีเดีย หมายถึง สารานุกรม หรือข้อมูลที่เป็นองค์ความรู้ที่ถูกจัดเก็บอย่างเป็นระบบ วิกิพีเดียบริหารงานโดยมูลนิธิวิกิพีเดีย ก่อตั้งเมื่อปี พ.ศ. 2544 โดย จิมมี เวลส์ และ แลร์รี แซงเจอร์ ซึ่งในปัจจุบันวิกิพีเดียเป็นเว็บไซต์สารานุกรมออนไลน์ที่ใหญ่ที่สุดในโลก

การที่วิกิพีเดียเป็นสารานุกรมที่เปิดให้บุคคลทั่วไปสามารถเข้ามาสร้างและแก้ไขบทความได้ทำให้เกิดข้อโต้แย้งกันในเรื่องความน่าเชื่อถือของบทความ ซึ่งวิกิพีเดีย "ไม่มีการรับประกันถึงความถูกต้อง" ของเนื้อหา แต่อย่างไรก็ตามในปี พ.ศ. 2548 วารสารเนเจอร์ ได้ทำการสำรวจบทความวิทยาศาสตร์บนวิกิพีเดียและบริตานิกา ความถูกต้องแม่นยำของสารานุกรมอยู่ในเกณฑ์ที่ใกล้เคียงกัน

วิกิพีเดียทำงานด้วยซอฟต์แวร์พิเศษที่มีชื่อว่า มีเดียวิกิ ที่เป็นลักษณะโอเพนซอร์ซและซอฟต์แวร์เสรีแบบพิเศษ ทำงานผ่านการบริหารเว็บไซต์ที่เรียกว่าวิกิ ตัวซอฟต์แวร์เขียนขึ้นด้วยภาษาพีเอชพีที่ทำงานร่วมกับฐานข้อมูลมายเอสคิวแอล

วิกิพีเดียมีการเรียกใช้งานประมาณ 25,000 ถึง 60,000 หน้าต่อวินาที ขึ้นอยู่กับช่วงเวลาของแต่ละวัน โดยในการเรียกใช้หน้านั้นจะเรียกจากสควิดแคชเซิร์ฟเวอร์ ถ้าพบจะส่งข้อมูลสารานุกรมกลับมา แต่ถ้าไม่พบจะถูกส่งไปยังโหนดบานลานซ์เซิร์ฟเวอร์เพื่อค้นจากคลังข้อมูลหลักเมื่อค้นพบจะทำการสร้างแคชและส่งข้อมูลสารานุกรมกลับมา

2.2 ความรู้พื้นฐานเกี่ยวกับคอนเซอร์เวพีเดีย

คอนเซอร์เวพีเดีย [3] คือ สารานุกรมออนไลน์ที่มีเนื้อหาเสรี ก่อตั้งขึ้นในเดือนพฤศจิกายน พ.ศ. 2549 โดย Andrew Schlafly เริ่มแรกจัดทำขึ้นเพื่อเป็นแหล่งให้ความรู้แก่เด็กนักเรียน ต่อมาเมื่อจำนวนข้อมูลมีเพิ่มมากขึ้นจึงกลายเป็นแหล่งข้อมูลที่เปิดกว้างสำหรับบุคคลทั่วไป ปัจจุบันคอนเซอร์เวพีเดียมีหน้าเพจเนื้อหา 44,910 หน้า และมีสมาชิกผู้ใช้งาน 66,482 คน

คอนเซอร์เวพีเดียทำงานด้วยซอฟต์แวร์ มีเดียวิกิ เช่นเดียวกับวิกิพีเดีย โดยจะมี MediaWiki API ที่มีรูปแบบการทำงานแบบ REST(REpresentational State Transfer) ซึ่งเป็น Web Service ชนิดหนึ่งที่ใช้สื่อสารบนอินเทอร์เน็ตใช้หลักการแบบ stateless คือ ไม่มี session โดยการทำงานจะอาศัย URI/URL ของ Request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูป XML, HTML, JSON ซึ่งจะเรียกใช้ผ่านทาง HTTP Method GET / POST / PUT / DELETE เป็นต้น

2.3 ความรู้พื้นฐานเกี่ยวกับเจสัน

เจสัน [4] คือ รูปแบบของข้อมูลที่ใช้สำหรับแลกเปลี่ยนข้อมูลขนาดเล็ก มีการจัดเก็บในรูปแบบที่สามารถอ่านและเข้าใจได้ง่าย ถูกกำหนดขึ้นภายใต้ภาษา JavaScript (JavaScript Programming Language, Standard ECMA-262 3rd Edition – December 1999.) เจสันเป็นรูปแบบข้อมูลตัวอักษรที่มีความเป็นอิสระอย่างสมบูรณ์ มีหลักการเขียนที่เข้าใจง่ายสำหรับนักเขียนโปรแกรมภาษาต่างๆ ไม่ว่าจะเป็น ภาษา C, C++, C#, Java, Javascript, Perl, Python และอื่นๆ มาตรฐานของฟอร์แมต เจสัน คือ RFC 4627 มี Internet media type เป็น application/json และมีนามสกุลของไฟล์เป็น .json ในการพัฒนาเว็บแอปพลิเคชันด้วยอาแจ็กซ์ (AJAX) นิยมใช้การแลกเปลี่ยนข้อมูลแบบเจสันเนื่องจากมีความกระชับและเข้าใจง่ายกว่า XML คุณสมบัติเหล่านี้ทำให้เจสันเหมาะที่จะนำมาใช้ในการแลกเปลี่ยนข้อมูล

เจสันมีลักษณะของภาษาจาวาสคริปต์ แต่ไม่ถูกมองว่าเป็นภาษาโปรแกรม โดยจะถูกมองว่าเป็นรูปแบบของข้อมูลที่ใช้สำหรับการแลกเปลี่ยนข้อมูล ในปัจจุบันมีไลบรารีของภาษาโปรแกรมต่าง ๆ ที่ใช้ประมวลผลข้อมูลในรูปแบบเจสันมากมาย ตัวอย่างรูปแบบข้อมูลเจสัน แสดงได้ในภาพที่ 2.1

```
{ "users": [
  {
    "firstName": "John",
    "lastName": "Jones",
    "joined": {
      "month": "April",
      "day": 28,
      "year": 2010
    }
  }
]}
```

ภาพที่ 2.1 รูปแบบข้อมูลเจสัน

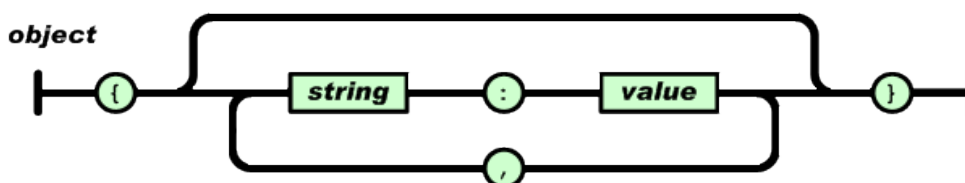
2.3.1 รูปแบบการแลกเปลี่ยนข้อมูลเจสันมี 2 รูปแบบ

2.3.1.1 รูปแบบชุดข้อมูลของชื่อหรือข้อมูลคู่กัน ในภาษาโปรแกรม ข้อมูลจะจัดอยู่ในรูปแบบของ object, record, struct, dictionary, hash table, keyed list หรือ associative array

2.3.1.2 รูปแบบลำดับของค่าข้อมูล ในภาษาโปรแกรมข้อมูลจะจัดอยู่ในรูปแบบของ array, vector, list หรือ sequence

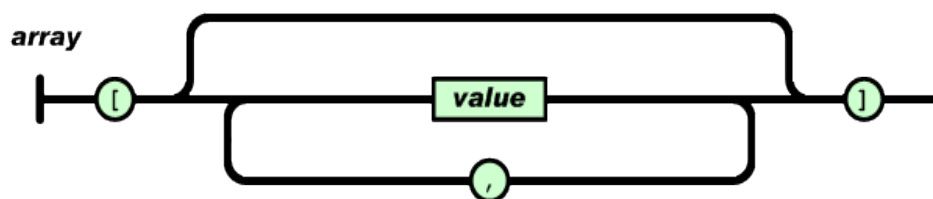
2.3.2 รูปแบบโครงสร้างของเจสัน

2.3.2.1 โครงสร้างแบบออบเจกต์ (Object) เป็นชุดของข้อมูลที่มีชื่อข้อมูลและค่าของข้อมูลนั้นคู่กัน ซึ่งจะถูกริเริ่มด้วยเครื่องหมาย { และจะปิดท้ายข้อมูลด้วยเครื่องหมาย } ข้อมูลแต่ละค่าจะมีเครื่องหมาย: กำกับระหว่างชื่อข้อมูลกับค่าของข้อมูล และแต่ละข้อมูลจะมีเครื่องหมาย , คั่นดังแสดงในภาพที่ 2.2



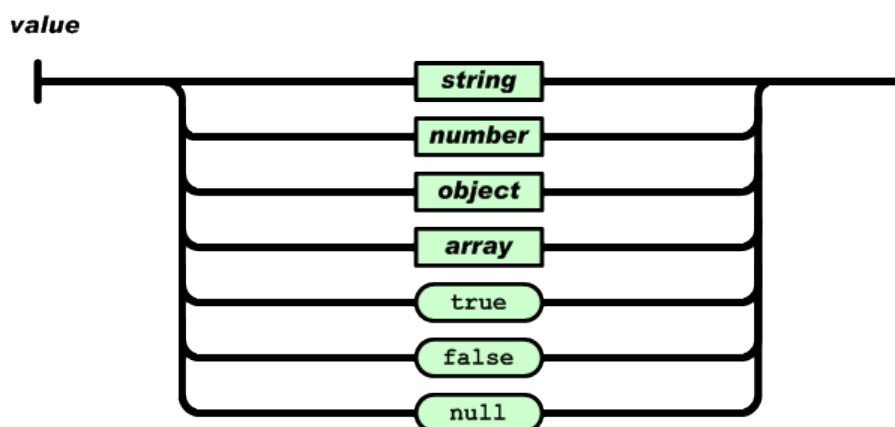
ภาพที่ 2.2 โครงสร้างเจสันแบบออบเจกต์

2.3.2.2 โครงสร้างแบบอาร์เรย์ (Array) เป็นลำดับของข้อมูล ซึ่งจะถูกริเริ่มด้วยเครื่องหมาย [และจะจบด้วยเครื่องหมาย] โดยที่แต่ละค่าของข้อมูลจะถูกคั่นด้วยเครื่องหมาย , ดังแสดงในภาพที่ 2.3



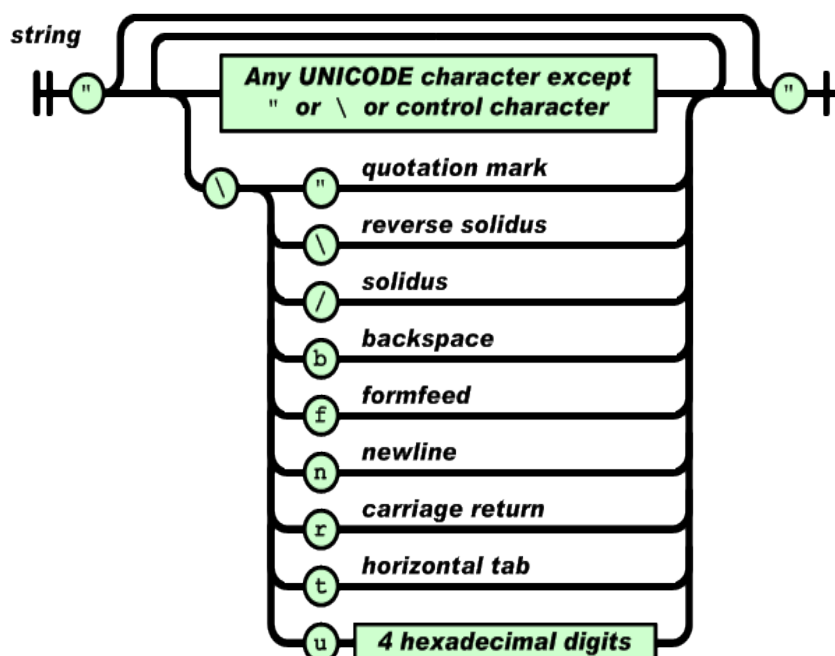
ภาพที่ 2.3 โครงสร้างเจสันแบบอาร์เรย์

2.3.2.3 โครงสร้างแบบแวลู (Value) เป็น String ที่อยู่ในเครื่องหมาย “” หรือตัวเลข หรือค่าทางตรรกศาสตร์ true, false หรือ ค่า null หรือ object หรือ array ซึ่งโครงสร้างสามารถวางซ้อนกันได้ ดังแสดงในภาพที่ 2.4



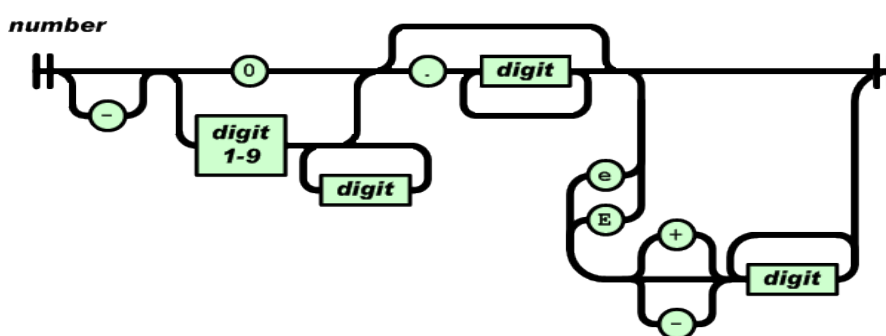
ภาพที่ 2.4 โครงสร้างเจสันแบบแวลู

2.3.2.4 โครงสร้างแบบสตริง (String) เป็นลำดับของตัวอักษรตั้งแต่ 0 ตัวอักษรหรือมากกว่า ซึ่งอยู่ภายใต้เครื่องหมาย “” และจะใช้เครื่องหมาย ในการใส่เครื่องหมายกำกับต่างๆ ซึ่งจะมีลักษณะคล้ายกับ String ในภาษา C หรือภาษา Java ดังแสดงในภาพที่ 2.5



ภาพที่ 2.5 โครงสร้างเจสันแบบสตริง

2.3.2.5 โครงสร้างแบบตัวเลข (Number) นั้นมีความคล้ายคลึงกับ Number ในภาษา C หรือภาษา Java อย่างมาก แต่จะไม่สามารถใช้เลขฐาน 8 กับเลขฐาน 16 ได้และช่องว่าง (Whitespace) สามารถที่จะใส่ไว้ระหว่างสัญลักษณ์ต่างๆ ได้ ยกเว้นรายละเอียดซึ่งเข้ารหัสที่สมบูรณ์ในการบรรยายภาษาต่างๆ ดังแสดงในภาพที่ 2.6



ภาพที่ 2.6 โครงสร้างเจสันแบบตัวเลข

2.3.3 รูปแบบโครงสร้างไฟล์เจสันของวิกิพีเดีย

การสืบค้นข้อมูลในวิกิพีเดียใช้วิธีการเรียกใช้ API จากเซิร์ฟเวอร์ของวิกิพีเดีย โดยการส่งคำร้องในคีย์เวิร์ดที่ต้องการสืบค้นเมื่อทางเซิร์ฟเวอร์ประมวลผลเสร็จจะส่งค่าผลลัพธ์กลับมาเป็นรูปแบบโครงสร้างไฟล์เจสัน ดังแสดงในภาพที่ 2.7

```
"query": {
  "searchinfo": {
    "totalhits": 2828169
  },
  "search": [
    {
      "ns": 0,
      "title": "Wikipedia",
      "pageid": 5043734
    },
    {
      "ns": 0,
      "title": "List of Wikipedias",
      "pageid": 6050087
    },
    {
      "ns": 0,
      "title": "English Wikipedia",
      "pageid": 328159
    },
    {
      "ns": 0,
      "title": "Tamil Wikipedia",
      "pageid": 13627771
    }
  ]
}
```

ภาพที่ 2.7 โครงสร้างไฟล์เจสันของวิกิพีเดีย

2.3.4 รูปแบบโครงสร้างไฟล์เจสันของคอนเซอร์เวเดีย

การสืบค้นข้อมูลในวิกิพีเดียใช้วิธีการเรียกใช้ API จากเซิร์ฟเวอร์ของคอนเซอร์เวเดีย โดยการส่งคำร้องในคีย์เวิร์ดที่ต้องการสืบค้นเมื่อทางเซิร์ฟเวอร์ประมวลผลเสร็จจะส่งค่าผลลัพธ์กลับมาเป็นรูปแบบโครงสร้างไฟล์เจสัน ดังแสดงในภาพที่ 2.8

```
"query": {
  "searchinfo": {
    "totalhits": 174
  },
  "search": [
    {
      "ns": 0,
      "title": "Main Page/Previous Conservapedia Breaking News/Archives/June 2010"
    },
    {
      "ns": 0,
      "title": "Debate:Is Conservapedia fair and balanced?"
    },
    {
      "ns": 0,
      "title": "Debate:How should Conservapedia work to avoid having a conservative bias?"
    },
    {
      "ns": 0,
      "title": "Main Page/Previous Conservapedia Breaking News/Archives/2012"
    },
    {
      "ns": 0,
      "title": "Debate:Can Conservapedia Succeed?"
    },
    {
      "ns": 0,
      "title": "How To Create a Conservapedia Account"
    }
  ]
}
```

ภาพที่ 2.8 โครงสร้างไฟล์เจสันของคอนเซอร์เวเดีย

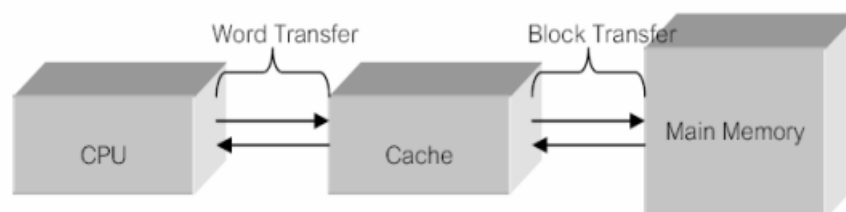
จากภาพที่ 2.8 โครงสร้างไฟล์เจสันของคอนเซอร์เวเดียจะมีความคล้ายกันกับโครงสร้างไฟล์เจสันของวิกิพีเดีย ทั้งนี้เนื่องจากเว็บไซต์สารานุกรมทั้งสองแหล่งใช้ซอฟต์แวร์โอเพนซอร์ซมีเดียวิกิในการพัฒนาเป็นเว็บแอปพลิเคชัน

2.4 ความรู้พื้นฐานเกี่ยวกับแคช

แคช [5] คือ หน่วยความจำขนาดเล็กที่มีความเร็วสูงที่ใช้เก็บข้อมูลหรือคำสั่งที่ถูกเรียกใช้บ่อย ๆ ซึ่งข้อมูลและคำสั่งที่เก็บอยู่ในแคชที่ใช้ SRAM (Static RAM) จะถูกดึงไปใช้งานได้เร็วกว่าการดึงข้อมูลจากหน่วยความจำหลักที่ใช้ DRAM (Dynamic RAM) หลายเท่าตัว แคชถูกนำมาใช้ในระบบคอมพิวเตอร์ตั้งแต่ PCs ไปจนถึง Supercomputer โดยตัวแคชนั้นจะถูกรวมไว้กับโปรเซสเซอร์และ Memory Management Unit (MMU) แต่หากไม่ได้รวมไว้ แคชจะถูกวางไว้บน CPU Board ซึ่งเรียกแคชประเภทนี้ว่า แคชภายนอก (External Cache) การที่แคชถูกวางไว้ใกล้กับโปรเซสเซอร์ทำให้เป็นการลดเวลาในการเข้าถึงข้อมูลระหว่างโปรเซสเซอร์และแคช ทำให้ศักยภาพในการทำงานสูงขึ้น

2.4.1 ลักษณะพื้นฐานของหน่วยความจำแคช (Cache)

หน่วยความจำแคชสร้างมาเพื่อวัตถุประสงค์ที่จะให้เป็นหน่วยความจำที่ทำงานได้เร็วที่สุดเท่าที่เทคโนโลยีในขณะนั้นจะทำได้ แนวคิดนี้แสดงได้ดังภาพที่ 2.9



ภาพที่ 2.9 หน่วยความจำแคชและหน่วยความจำหลัก

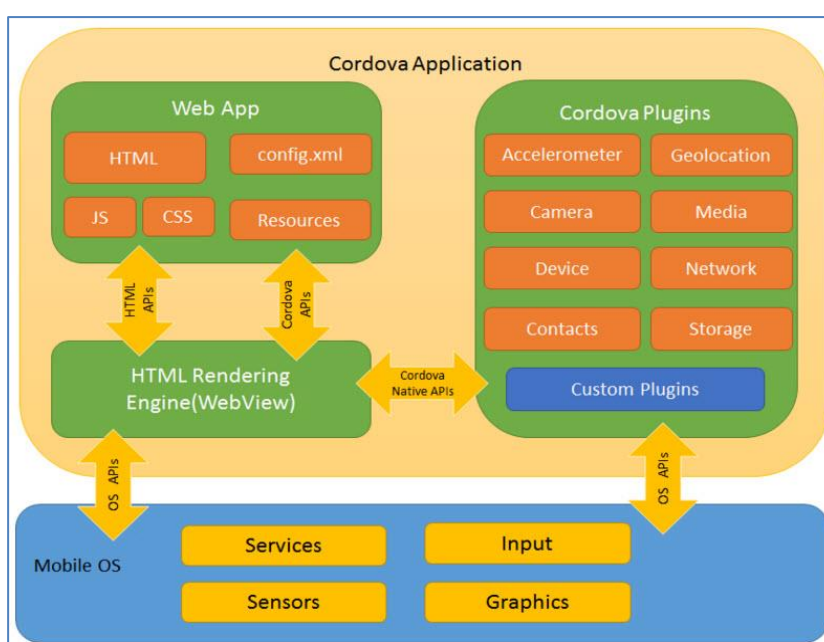
จากภาพที่ 2.9 ในเครื่องคอมพิวเตอร์มีหน่วยความจำหลักที่มีความเร็วต่ำ (เมื่อเปรียบเทียบกับความเร็วของซีพียู) ที่มีปริมาณมากและมีหน่วยความจำแคชที่เร็วกว่าแต่มีขนาดเล็กกว่า โดยปกติหน่วยความจำแคชจะเก็บสำเนาของข้อมูลบางส่วนในหน่วยความจำหลักเอาไว้ เมื่อโปรเซสเซอร์ต้องการอ่านข้อมูลจำนวนหนึ่ง Word ในหน่วยความจำ ข้อมูลส่วนนั้นจะถูกตรวจสอบว่ามีอยู่ในแคชหรือไม่ ถ้ามีอยู่จะนำข้อมูลในแคชไปใช้ ถ้าไม่มีอยู่จะเกิดการคัดลอกสำเนาข้อมูลหนึ่งบล็อกจากหน่วยความจำหลักมายังแคช แล้วจึงนำ Word ที่ต้องการส่งไปให้โปรเซสเซอร์ในภายหลัง เนื่องจากปรากฏการณ์อ้างอิงในพื้นที่เดียวกัน (Locality of Reference) จะทำให้การอ้างอิงข้อมูลในหน่วยความจำครั้งต่อ ๆ ไปเป็นการอ้างอิงที่เดิมหรือตำแหน่งใกล้เคียงจุดเดิม ดังนั้นการคัดลอกข้อมูลหนึ่งบล็อกจากหน่วยความจำหลักมายังแคช จะสามารถนำมาใช้งานได้ระยะหนึ่งก่อนที่จะมีการคัดลอกข้อมูลในครั้งต่อไป

2.5 ความรู้พื้นฐานเกี่ยวกับไอโอนิคเฟรมเวิร์ก

ไอโอนิค [6] เป็นซอฟต์แวร์โอเพนซอสสำหรับพัฒนาแอปพลิเคชันบนมือถือแบบไฮบริด (Hybrid) สร้างขึ้นจากการรวมกันของ Apache Cordova และ AngularJS ซึ่งเป็นแนวคิดในการพัฒนาแอปที่ช่วยลดขั้นตอนในการพัฒนาลง โดยการพัฒนารั้ดเดียวสามารถนำไปใช้ได้หลายแพลตฟอร์ม

2.5.1 Cordova

เกิดขึ้นจากการตกลงระหว่าง Adobe และ Nitobi ต้องการให้เกิดซอฟต์แวร์โอเพนซอสสำหรับการพัฒนาแอปพลิเคชันแบบไฮบริด ซึ่งถูกนำไปใช้ในโครงการ Hybrid Mobile Application หลายโครงการ เช่น AppGyver, Ionicframework เป็นต้น Cordova เป็นแพลตฟอร์มที่ใช้สำหรับสร้างแอปพลิเคชันบนมือถือ โดยใช้ภาษา HTML, CSS และ Javascript เป็นหลักในการพัฒนาสนับสนุนระบบปฏิบัติการมากมายเช่น Android, iOS, Blackberry, Ubuntu, Windows Phone, WebOS, Amazon Fire OS เป็นต้น สถาปัตยกรรม Cordova Application แสดงได้ในภาพที่ 2.10



ภาพที่ 2.10 สถาปัตยกรรมของ Cordova Application

2.5.2 เทคโนโลยีที่ใช้ในการพัฒนา

2.5.2.1 HTML5 [7] คือ ภาษามาร์กอัปที่ใช้สำหรับเขียนเว็บไซต์ ซึ่ง HTML5 นี้เป็นภาษาที่ถูกพัฒนาต่อมาจากภาษา HTML และพัฒนาขึ้นมาโดย WHATWG (The Web Hypertext Application Technology Working Group) โดยได้มีการปรับปรุงเพิ่มเติมคุณสมบัติหลาย ๆ อย่างเข้ามาเพื่อให้ผู้พัฒนาสามารถใช้งานได้ง่ายและมีประสิทธิภาพยิ่งขึ้น

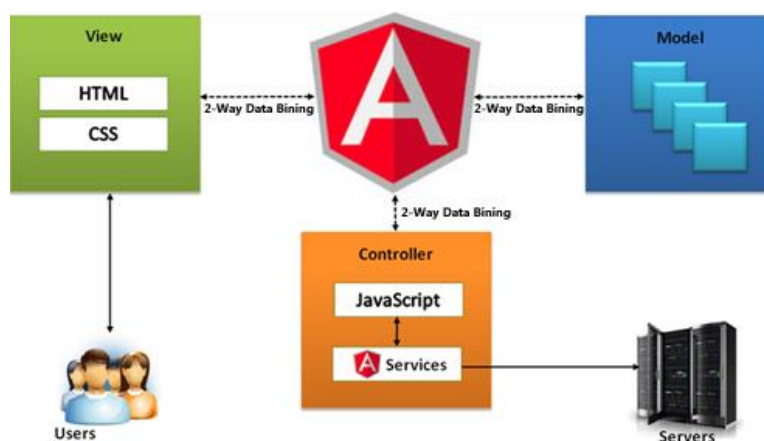
2.5.2.2 CSS3 [8] คือ ภาษาที่ใช้สำหรับจัดรูปแบบการแสดงผลของเอกสาร HTML ซึ่งทำให้การแสดงผลของ HTML มีความสวยงามมากขึ้น โดยใน CSS3 จะแยกเนื้อหาออกเป็นโมดูล (Module) ซึ่งในแต่ละโมดูลจะมี CSS Properties และ Values เพื่อใช้ในการควบคุมการแสดงผลของ HTML โดยสื่อสารผ่าน Selectors ที่เป็นองค์ประกอบหลักของการเขียน CSS

2.5.2.3 AngularJS [9] คือ Javascript Framework ที่พัฒนาโดย Google AngularJS นั้นจะมีลักษณะพิเศษในการเรียกใช้งานโดยจะต้องเพิ่ม Attribute พิเศษให้กับ HTML เพื่อใช้งาน และยังเป็น Framework ที่ใช้จัดการหน้าจอในลักษณะ Single Page Applications (SPAs) หรือการจัดการทุกอย่างในหน้าจอเดียว ซึ่งรูปแบบการทำงานของ Angular จะเป็นแบบ Model View Controller (MVC) มีการแบ่งการทำงานดังนี้

(1) Model เป็นส่วนที่รับผิดชอบในการจัดการข้อมูล เช่น การจัดเก็บข้อมูล ในฐานข้อมูลการรับค่าข้อมูลจากฟอร์มหรือการบำรุงรักษาข้อมูล เป็นต้น โดยมีคุณสมบัติการทำงานแบบ 2-Way-Data Binding นั่นคือการผูกค่าของข้อมูลเข้าด้วยกันใน Model และ AngularJS โดยค่าของข้อมูลจะ Synchronize กันเมื่อมีการเปลี่ยนแปลงค่าจากฝั่งหนึ่งอีกฝั่งก็จะเปลี่ยนค่าตามไป

(2) View เป็นส่วนที่รับผิดชอบในการแสดงผลข้อมูลหรือแสดงผลออกทางหน้าจอผู้ใช้ โดยการทำงานจะมีคุณสมบัติเป็นแบบ 2-Way-Data Binding เช่นเดียวกับ Model ซึ่งภาษาที่ทำงานร่วมกับ AngularJS ในส่วนของ View ก็คือ HTML และ CSS

(3) Controller เป็นส่วนที่ควบคุมการทำงานในส่วน of View และ Model โดยมีคุณสมบัติ 2-Way-Data Binding เช่นเดียวกับ Model และ View ซึ่งในส่วนของ Controller อาจจะมีการเชื่อมต่อกับเซิร์ฟเวอร์ภายนอกในการรับค่ามาแสดงผล ซึ่งรูปแบบการทำงานแบบ MVC แสดงได้ในภาพที่ 2.11



ภาพที่ 2.11 โครงสร้างการทำงานแบบ MVC Pattern

2.6 งานวิจัยที่เกี่ยวข้อง

หลักการและทฤษฎีที่กล่าวมาข้างต้นสามารถสรุปงานวิจัยที่เกี่ยวข้องใช้สำหรับแนวทางการพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยโอโอไนคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการโอโอเอส ดังต่อไปนี้

2.6.1 อัยพรณ เอโกบล (2557) [10] ได้ศึกษาเรื่องการทำเจสันแคชด้วยโนเอสคิวแอล ซึ่งเป็นการนำไปประยุกต์ใช้กับหน่วยงานการประปานครหลวงที่มีข้อมูลเข้าออกจำนวนมาก ทำให้เพิ่มภาระให้กับการจัดการโครงสร้างฮาร์ดแวร์ทั้งระบบ จึงมีความต้องการวิธีการที่จะช่วยลดภาระทางด้านฮาร์ดแวร์และเครือข่าย ซึ่งจากการศึกษาพบว่าเทคนิคการทำแคชข้อมูลเจสันในรูปแบบแท็กไฟล์ (Text File) ช่วยทำให้ระบบมีประสิทธิภาพทางเวลาดีขึ้น โดยมีค่าเฉลี่ยเวลาในการตอบสนองเท่ากับ 142.8 มิลลิวินาที และการจัดเก็บข้อมูลจำนวนข้อมูลมากกว่า 10,000 เรคอร์ด โนเอสคิวแอลใช้เวลาเฉลี่ยเพียงแค่ 3.041 วินาที ในขณะที่มายเอสคิวแอลใช้เวลาเฉลี่ยถึง 140 วินาที ดังนั้นการนำเจสันแคช และ โนเอสคิวแอล มาประยุกต์ใช้งานร่วมกันจึงช่วยให้ระบบมีประสิทธิภาพที่ดียิ่งขึ้น

2.6.2 สัญญา เครือหงษ์ (2556) [11] ได้ศึกษาเรื่องการพัฒนาแอปพลิเคชันติดตามสถานะอีเอ็มเอสผ่านมือถือด้วยข้อมูลรูปแบบเจสัน เพื่อใช้ตรวจสอบสถานะการส่งพัสดุและจดหมายลงทะเบียนของบริษัทไปรษณีย์ไทยบนระบบปฏิบัติการโอโอเอส โดยแอปพลิเคชันสามารถแสดงสำนักงานไปรษณีย์ วัน เวลา และข้อมูลสถานะของพัสดุ ซึ่งชื่อสถานะที่จะถูกแปลงให้อยู่ในรูปแบบเจสันแล้วส่งไปยัง Geocoding Google เพื่อหาพิกัดแสดงบนแผนที่จริงบนมือถือ สามารถบันทึกรายการไว้เพื่อใช้ติดตามสถานะการส่งพัสดุได้ โดยผลการประเมินแอปพลิเคชันพบว่าสามารถช่วยอำนวยความสะดวกในการตรวจสอบสถานะการส่งพัสดุและจดหมายลงทะเบียนให้กับผู้ใช้งานได้อย่างมีประสิทธิภาพ ผลการประเมินความพึงพอใจเท่ากับ 4.23 จากกลุ่มผู้ใช้งาน 45 คน

2.6.3 May Sabai Han (2013) [12] ได้ศึกษาเรื่องการค้นคืนข้อมูลเชิงความหมายบนวิกิพีเดียในงานวิจัยนี้ใช้การวัดความคล้ายของข้อมูลเชิงความหมายด้วยวิธีการใช้วิกิพีเดียออนโทโลยีและกลยุทธ์การกระตุ้นแบบแพร่กระจาย จากผลการประเมินพบว่าการใช้ออนโทโลยีสามารถลดความพยายามของผู้เชี่ยวชาญที่จำเป็นในการสร้างออนโทโลยี และ กลยุทธ์การกระตุ้นแบบแพร่กระจายได้ผลลัพธ์ที่ดีมากโดยเฉพาะระบบสัมพันธ์เชิงความหมาย เช่น การปรับเปลี่ยนคำศัพท์ การคำนวณความคล้ายของข้อมูลเชิงความหมายด้วยออนโทโลยี และ การอธิบายเอกสาร

2.6.4 Tomas Cerny (2010) [13] ได้ศึกษาเรื่องการทำงานร่วมกันของเว็บแคชในเครือข่าย ซึ่งการทำงานระหว่างไคลเอ็นท์กับเซิร์ฟเวอร์โดยตรงที่ไม่ผ่านแคชทำให้มีการร้องขอข้อมูลจากเซิร์ฟเวอร์จะมีการประมวลผลทุกครั้งบนฝั่งเซิร์ฟเวอร์ ทำให้เซิร์ฟเวอร์รับภาระหนัก การใช้เว็บแคชเป็นตัวกลางจะคอยให้บริการข้อมูลที่ถูกระบุเรียกใช้งานซ้ำ ๆ จึงช่วยลดภาระของเซิร์ฟเวอร์และแบนด์วิดท์ที่ไม่จำเป็นซึ่งแนวคิดการใช้เว็บแคชหลาย ๆ เว็บแคชมาทำงานร่วมกันเพื่อให้เกิดประสิทธิภาพมากยิ่งขึ้น ผลที่ได้จากการทดลองพบว่าการออกแบบและนำไปใช้กับเครือข่ายซ้อนทับแบบ Peer-to-Peer ได้ผลลัพธ์ที่ดี

2.6.5 Nurzhan Nurseitov (2009) [14] ได้ศึกษาเรื่องการเปรียบเทียบประสิทธิภาพการแลกเปลี่ยนข้อมูลระหว่างเอ็กเอ็มแอลกับเจสัน โดยในการทดสอบได้แบ่งวิธีการออกเป็น 2 เงื่อนไข คือ 1)ทดสอบโดยการส่งผ่านข้อมูลขนาดใหญ่จำนวน 1,000,000 ออปเจ็กต์ในครั้งเดียว ซึ่งผลปรากฏว่ารูปแบบเจสันใช้เวลาเฉลี่ยเพียงแค่ 0.08 มิลลิวินาที ในขณะที่เอ็กเอ็มแอลใช้เวลาเฉลี่ยมากถึง 4.55 มิลลิวินาที 2)ทดสอบโดยใช้ข้อมูลขนาดเล็กหลาย ๆ ขนาดแบ่งเป็น 20,000 40,000 60,000 80,000 และ 100,000 ออปเจ็กต์ ซึ่งผลปรากฏว่ารูปแบบเจสันใช้เวลาเฉลี่ยเพียงแค่ 0.085 มิลลิวินาที ในขณะที่เอ็กเอ็มแอลใช้เวลามากถึง 3.09 มิลลิวินาที ทำให้สรุปได้ว่าการใช้ข้อมูลรูปแบบเจสันในการแลกเปลี่ยนข้อมูลนั้นสามารถส่งข้อมูลได้รวดเร็วกว่าข้อมูลในรูปแบบเอ็กเอ็มแอล

2.6.6 P. Ravi Kiran (2014) [15] ได้ทำการศึกษาเรื่องการเปรียบเทียบข้อแตกต่างการเข้าถึงข้อมูลเซิร์ฟเวอร์โดยใช้อุปกรณ์มือถือและการทำแคชข้อมูลบนมือถือ ซึ่งเป็นการเปรียบเทียบระหว่างการเข้าถึงแบบผ่านระบบตัวกลาง(Middle ware) ที่คอยให้บริการข้อมูลหรือเว็บเซอร์วิสกับแบบไม่ผ่านระบบตัวกลางหรือเป็นการใช้ J2ME Driver ในการเชื่อมต่อกับฐานข้อมูลของเซิร์ฟเวอร์โดยตรง ผลที่ได้จากการทดลองพบว่าการเข้าถึงแบบไม่ผ่านตัวกลางมีความยุ่งยากและข้อจำกัดมากกว่าเนื่องจากทุกครั้งที่มีการร้องขอข้อมูลต้องมีการสร้างการเชื่อมต่อผ่านไดรเวอร์และเก็บข้อมูลแคชจำนวนมาก

จากงานวิจัยข้างต้น ผู้วิจัยได้นำแนวคิดในการพัฒนาแอปพลิเคชันบนมือถือที่ใช้รูปแบบการแลกเปลี่ยนข้อมูลแบบเจสันมาพัฒนาเพื่อใช้สืบค้นข้อมูลจากเว็บไซต์สารานุกรมวิกิพีเดียและคอนเซอร์เวทีฟเดีย และนำแนวคิดในการสร้างแคชมาประยุกต์ใช้เพื่อแก้ปัญหาเวลาการตอบสนองในการแสดงผลการสืบค้นและลดแบนด์วิดท์ที่ไม่จำเป็น

บทที่ 3

วิธีดำเนินการวิจัย

งานวิจัยนี้ ผู้วิจัยได้นำหลักการวิเคราะห์และออกแบบระบบตามหลักวิศวกรรมซอฟต์แวร์ โดยมีขั้นตอนการดำเนินงานดังนี้

- 3.1 การออกแบบสถาปัตยกรรมแคชบนมือถือและแคชบนเซิร์ฟเวอร์
- 3.2 การออกแบบระบบด้วยแผนภาพกระแสข้อมูล (Data Flow Diagram)
- 3.3 การออกแบบฐานข้อมูลด้วยแผนภาพความสัมพันธ์ของข้อมูล (E-R Diagram)
- 3.4 การออกแบบพจนานุกรมข้อมูล (Data Dictionary)
- 3.5 การออกแบบหน้าจอ (Graphic User Interface)
- 3.6 การออกแบบการวิเคราะห์และประเมินผล

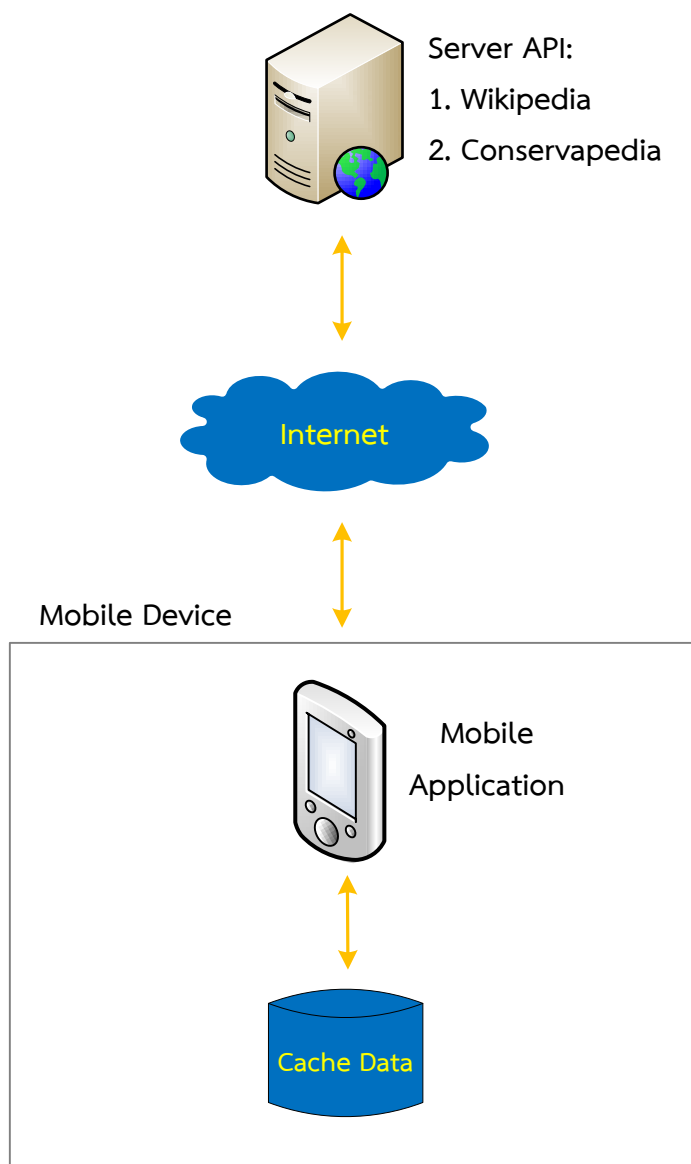
3.1 การออกแบบสถาปัตยกรรมแคชบนมือถือและแคชบนเซิร์ฟเวอร์

ในการส่งรับข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์มักจะมีปัญหาในเรื่องของความเร็วในการโอนถ่ายข้อมูลจากต้นทางไปปลายทางและการประมวลผลบนฝั่งเซิร์ฟเวอร์ทุก ๆ ครั้งที่มีการร้องขอข้อมูลจากไคลเอนต์การสร้างแคชขึ้นมาจึงเป็นแนวคิดที่เข้ามาช่วยในเรื่องของการลดแบนด์วิดท์และการประมวลผลของฝั่งเซิร์ฟเวอร์ โดยแคชจะอยู่ในรูปของข้อมูลที่เคยถูกเรียกใช้มาแล้วมีการจัดเก็บอย่างเป็นระบบและสามารถนำกลับมาใช้ใหม่ได้โดยไม่ต้องเชื่อมต่อไปยังเซิร์ฟเวอร์ทุกครั้ง ทำให้สามารถที่จะแสดงผลข้อมูลออกมาได้อย่างรวดเร็ว และ ลดการใช้แบนด์วิดท์ได้อย่างมีประสิทธิภาพ

สถาปัตยกรรมระบบแคชที่นำมาใช้งานร่วมกับการพัฒนาแอปพลิเคชันบนมือถือระบบปฏิบัติการไอโอเอสในงานวิจัยนี้ แบ่งออกได้ 2 แบบดังนี้

3.1.1 การทำแคชบนมือถือ [16]

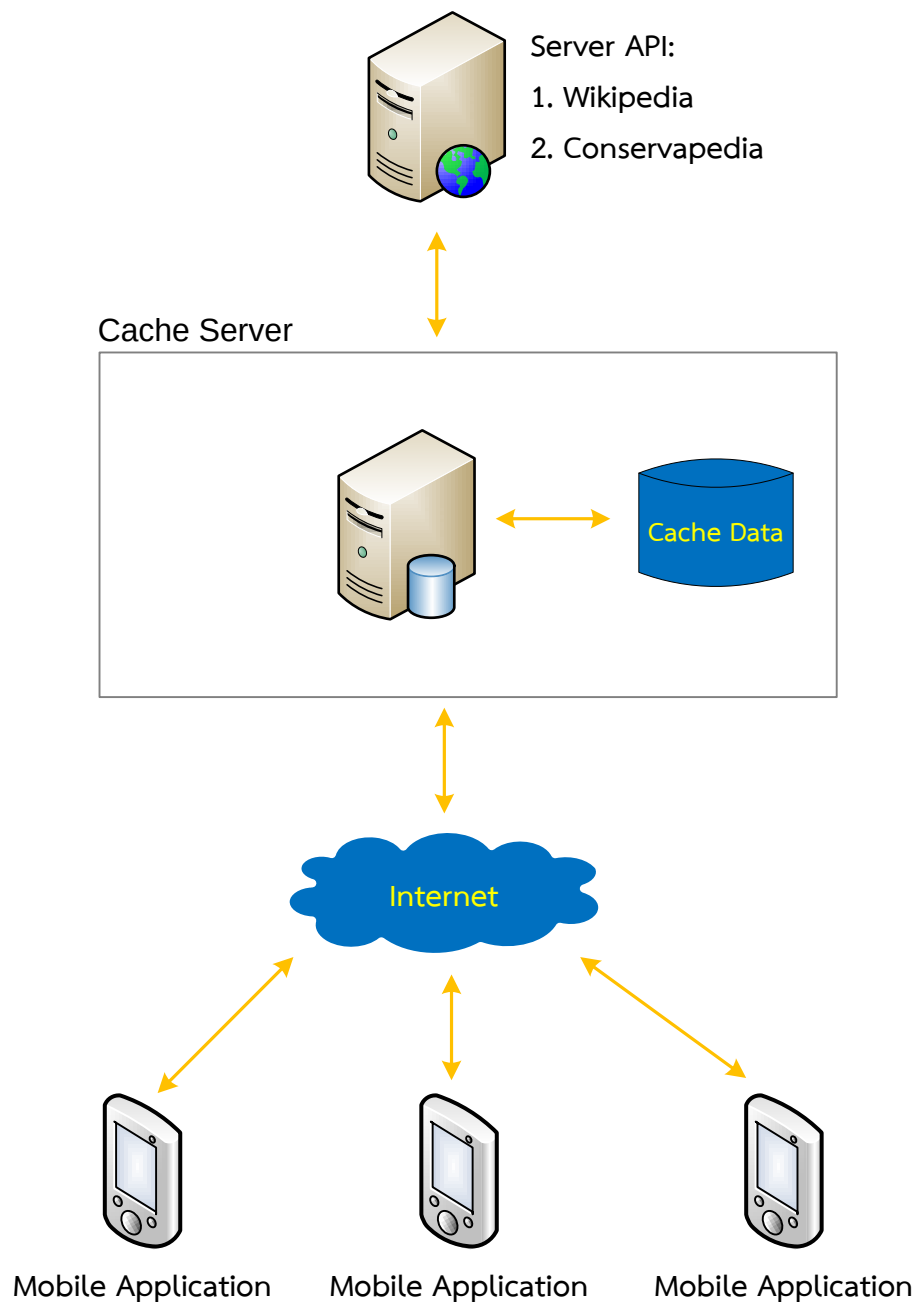
เป็นวิธีการจัดเก็บข้อมูลแคชไว้บนหน่วยความจำภายในของมือถือ เริ่มจากแอปพลิเคชันบนมือถือทำการสืบค้นข้อมูลภายในแคชบนมือถือถ้าพบก็จะนำข้อมูลนั้นมาแสดงผลทันที แต่ถ้าหากไม่พบก็จะทำการร้องขอไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ตอบกลับมาระบบจะทำการสร้างแคชไฟล์ของผลลัพธ์นั้นไว้บนมือถือเพื่อให้สามารถเรียกใช้งานได้ในครั้งต่อไป ดังแสดงในภาพที่ 3.1



ภาพที่ 3.1 สถาปัตยกรรมแคชบนมือถือ

3.1.2 การทำแคชบนเซิร์ฟเวอร์ [17]

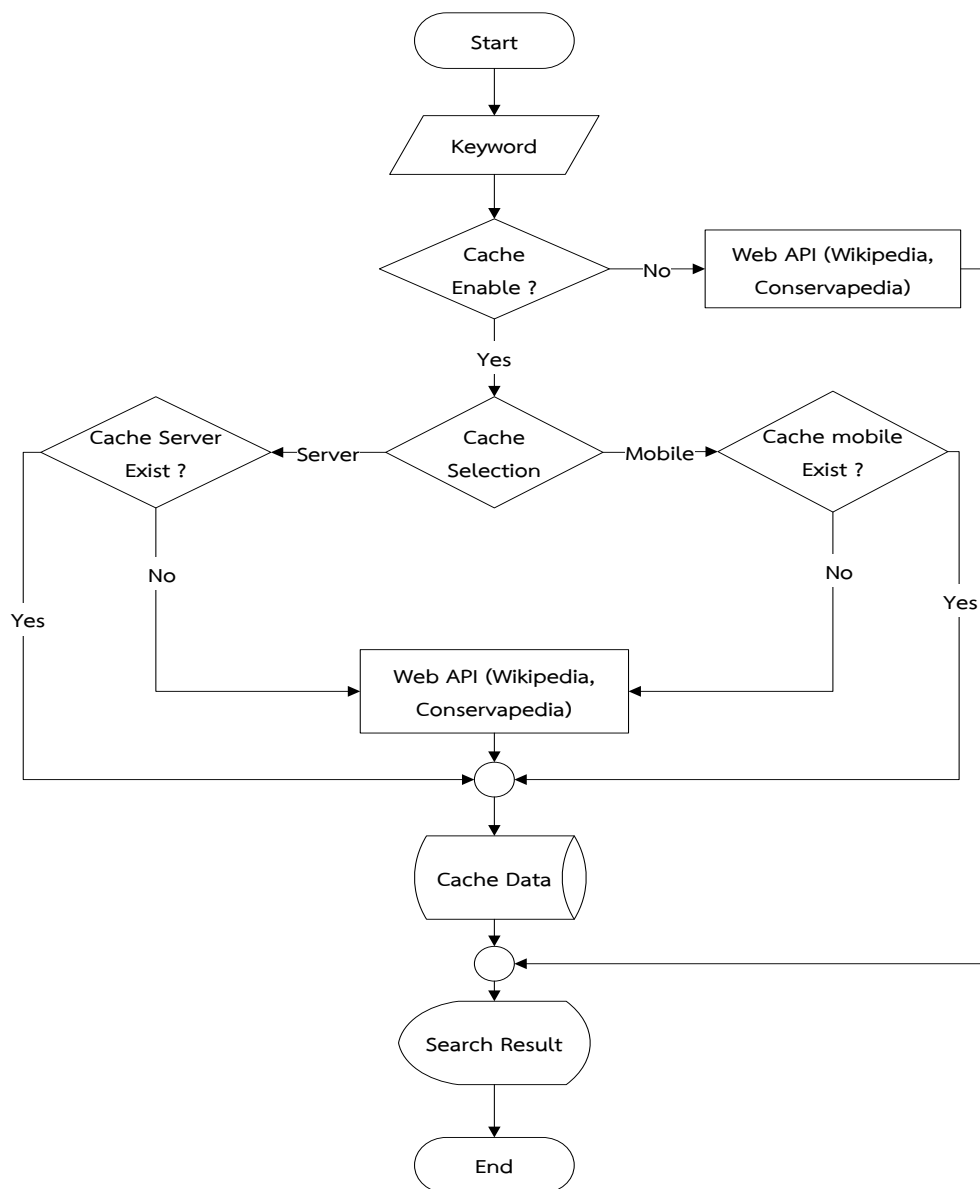
เป็นวิธีการจัดเก็บข้อมูลแคชไว้บนแคชเซิร์ฟเวอร์ เริ่มจากแอปพลิเคชันบนมือถือทำการสืบค้นข้อมูลบนแคชเซิร์ฟเวอร์ถ้าพบว่ามีอยู่ในแคชเซิร์ฟเวอร์ก็จะเรียกข้อมูลจากแคชเซิร์ฟเวอร์ส่งกลับมาทันที แต่ถ้าไม่พบก็จะไปสืบค้นข้อมูลจากเว็บไซต์ของวิกิพีเดียหรือคอนเซอร์เวเดีย เมื่อเว็บไซต์ตอบกลับมาระบบจะทำการสร้างแคชเก็บไว้ที่แคชเซิร์ฟเวอร์และส่งผลลัพธ์กลับมายังแอปพลิเคชันบนมือถือ ในแคชเซิร์ฟเวอร์จะคอยทำหน้าที่สร้างไฟล์แคชและแจกจ่ายไฟล์แคชไปยังแอปพลิเคชันที่เรียกใช้ ดังแสดงในภาพที่ 3.2



ภาพที่ 3.2 สถาปัตยกรรมการทำแคชบนเซิร์ฟเวอร์

3.1.3 การออกแบบผังงานของการพัฒนาระบบ (Flow Chart)

การออกแบบผังงาน (Flow Chart) [18] เป็นแผนภาพที่ใช้อธิบายขั้นตอนหรือกระบวนการทำงานตั้งแต่เริ่มต้นจนถึงสิ้นสุดการทำงาน โดยในงานวิจัยนี้ผู้วิจัยได้ออกแบบผังงานการทำงานของระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวตฟิเดียด้วยไอโฟนิกเฟรมเวิร์ก เจสันและแคชบนระบบปฏิบัติการไอโอเอส แสดงได้ดังภาพที่ 3.3



ภาพที่ 3.3 ออกแบบผังงานของการพัฒนาระบบ

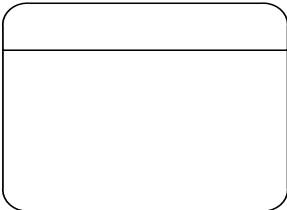
3.2 การออกแบบระบบด้วยแผนภาพกระแสข้อมูล (Data Flow Diagram)

แผนภาพแสดงกระแสข้อมูล (Data Flow Diagram: DFD) [19] เป็นแผนภาพที่มีการวิเคราะห์แบบมีโครงสร้าง โดยแผนภาพกระแสข้อมูลนี้ใช้เป็นเครื่องมือในการพัฒนาระบบงาน แสดงความสัมพันธ์ระหว่างโปรเซสและข้อมูลที่เกี่ยวข้อง ซึ่งข้อมูลในแผนภาพทำให้ทราบถึง

- (1) ข้อมูลมาจากไหน
- (2) ข้อมูลไปที่ไหน
- (3) ข้อมูลเก็บที่ใด
- (4) เกิดเหตุการณ์ใดกับข้อมูลในระหว่างทาง

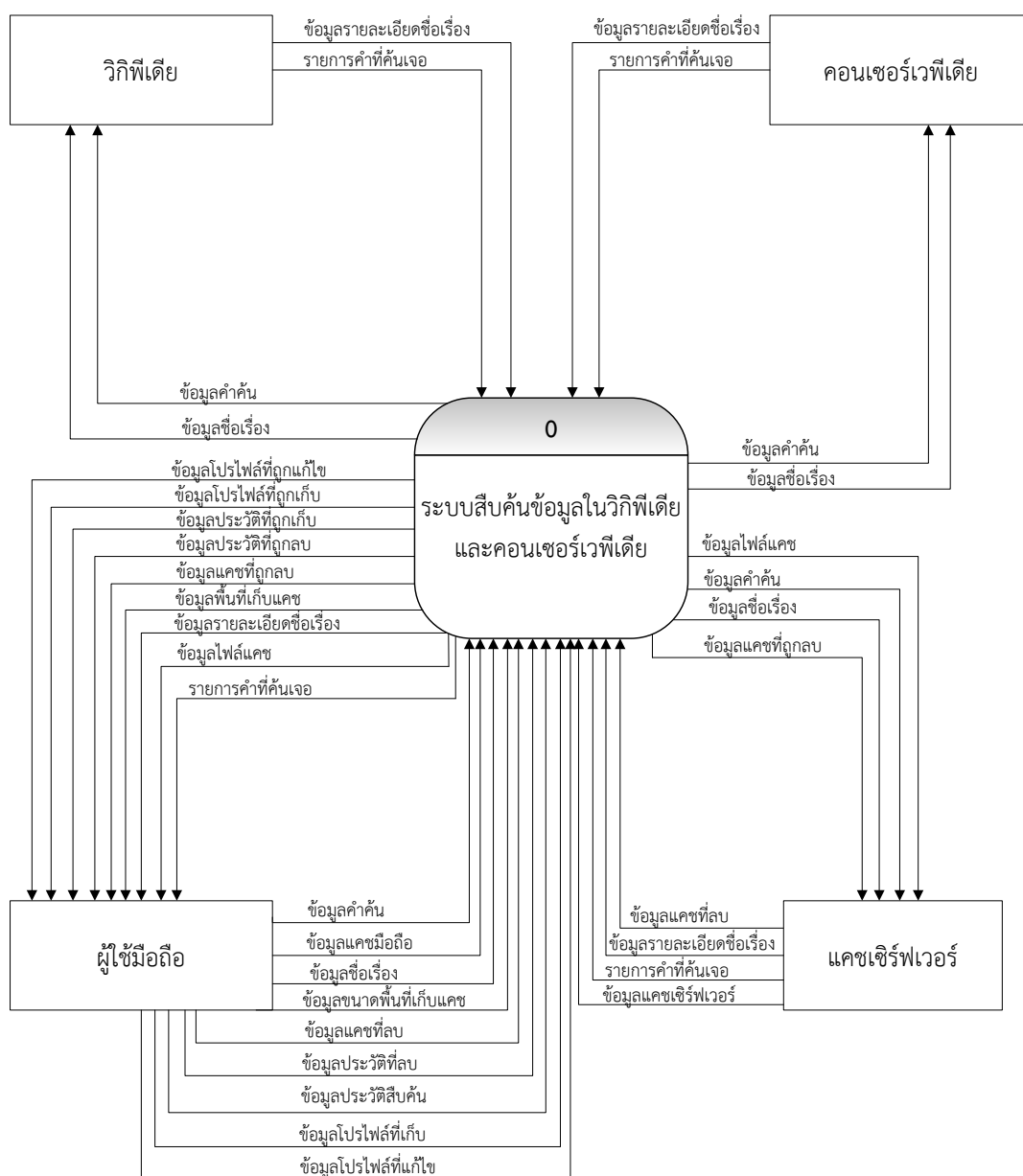
สัญลักษณ์ในการสร้างแผนภาพกระแสข้อมูลที่ใช้ในงานวิจัยนี้จะใช้แบบ Gane & Sarson ซึ่งมีรูปแบบแสดงในตารางที่ 3.1

ตารางที่ 3.1 สัญลักษณ์แผนภาพกระแสข้อมูลแบบ Gane & Sarson

ชื่อสัญลักษณ์	สัญลักษณ์	ความหมาย
Entity		บุคคล, หน่วยงาน หรือระบบงานภายนอก
Process		การประมวลผล
Data Store		การเก็บข้อมูล
Data Flow		เส้นทางการไหลของข้อมูล
Connector		สัญลักษณ์จุดเชื่อมต่อ

3.2.1 ภาพรวมของระบบ (Context Diagram)

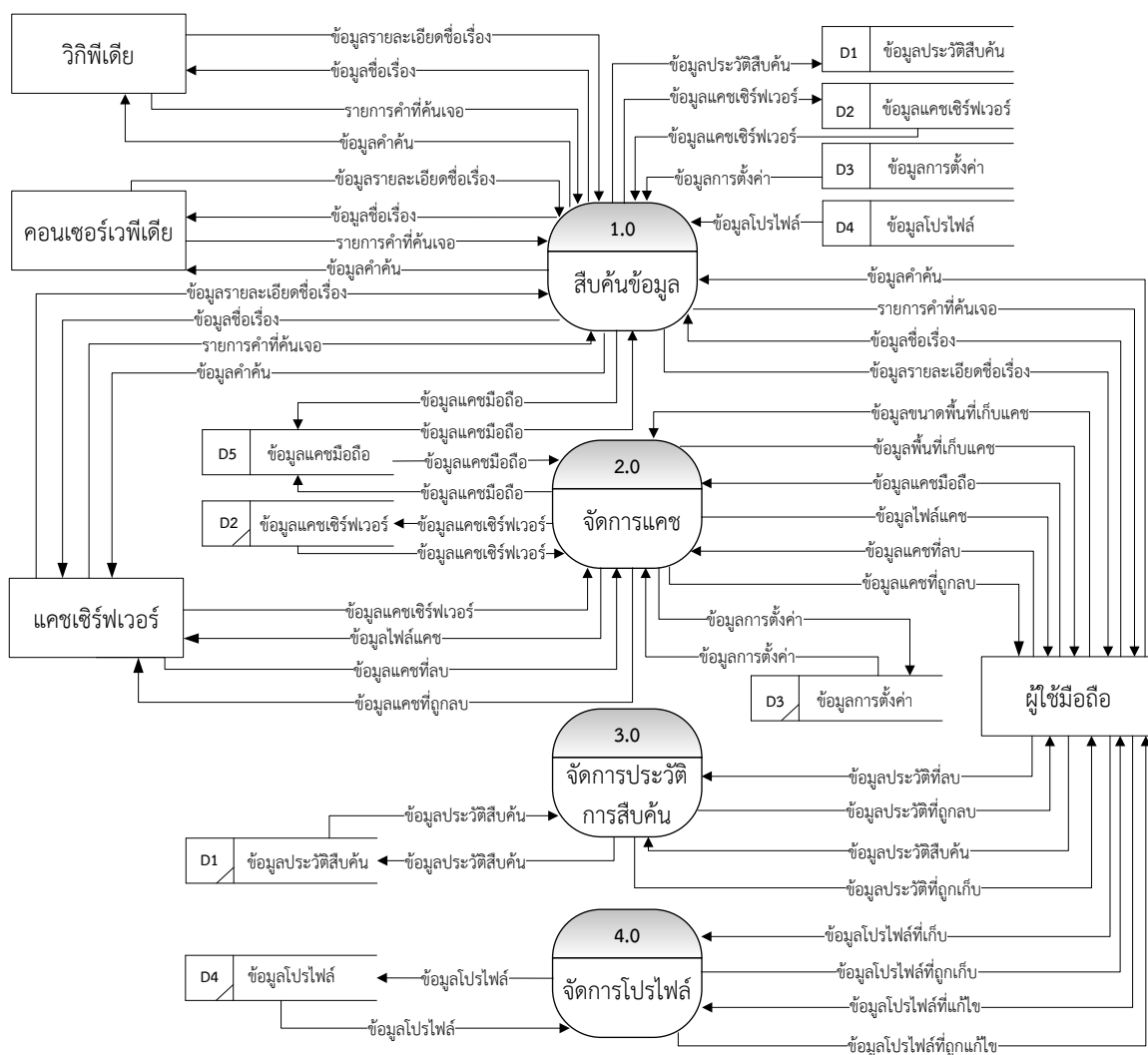
เป็นแผนภาพกระแสข้อมูลชั้นบนสุดที่แสดงภาพรวมการทำงานของระบบที่มีความสัมพันธ์กับสภาพแวดล้อมภายนอก โดยแสดงให้เห็นถึงรายละเอียดของกระบวนการการทำงานทั้งหมดที่มีอยู่ภายในว่ามีขั้นตอนใดบ้าง แผนภาพ Context Diagram แสดงในภาพที่ 3.4



ภาพที่ 3.4 แผนภาพ Context Diagram ของระบบ

3.2.2 แผนภาพกระแสการไหลของข้อมูลระดับ 0 (Data Flow Diagram Level 0)

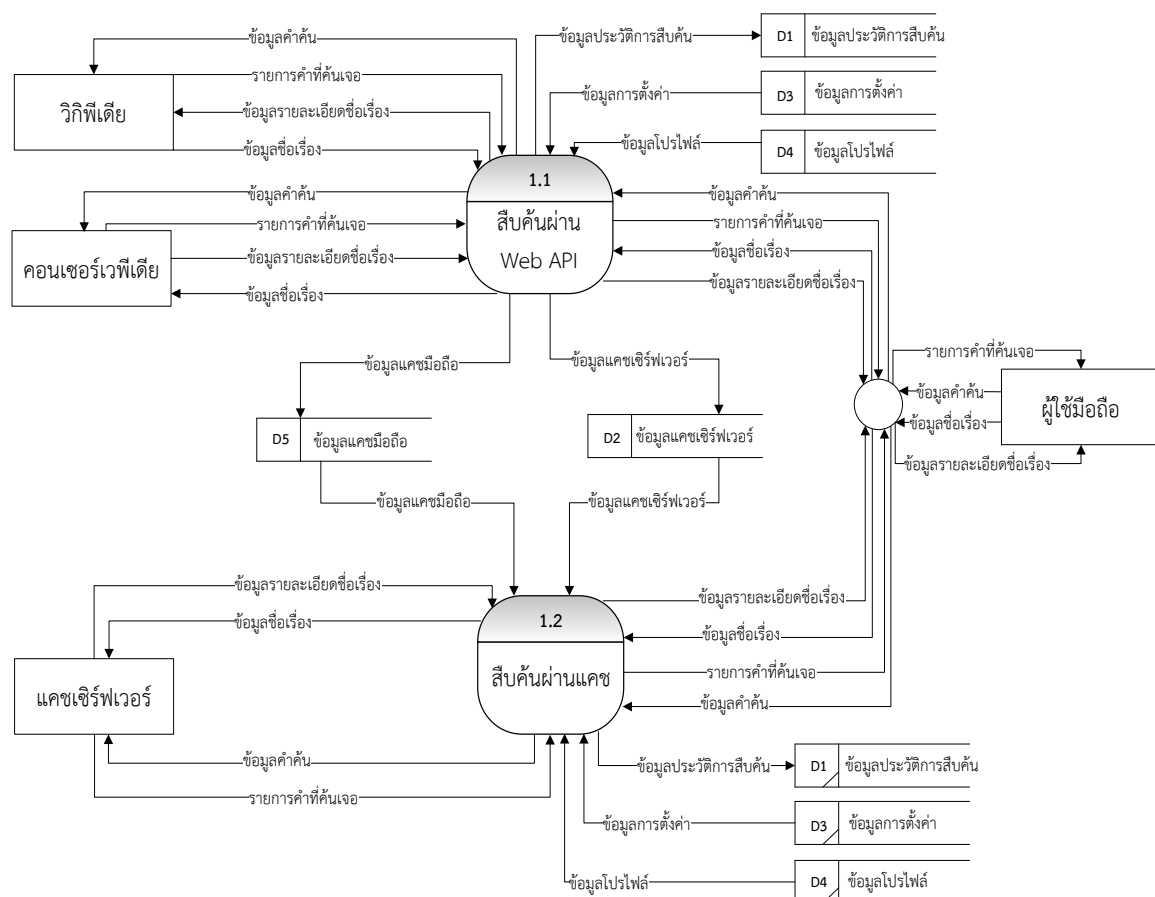
DFD Level 0 เป็นการแตกรายละเอียด โดยจะแสดงถึงโปรเซสหลัก ๆ และผู้ที่เกี่ยวข้องกับระบบ รวมถึงการเก็บข้อมูลต่าง ๆ ในแต่ละโปรเซส โดยแสดงในภาพที่ 3.5



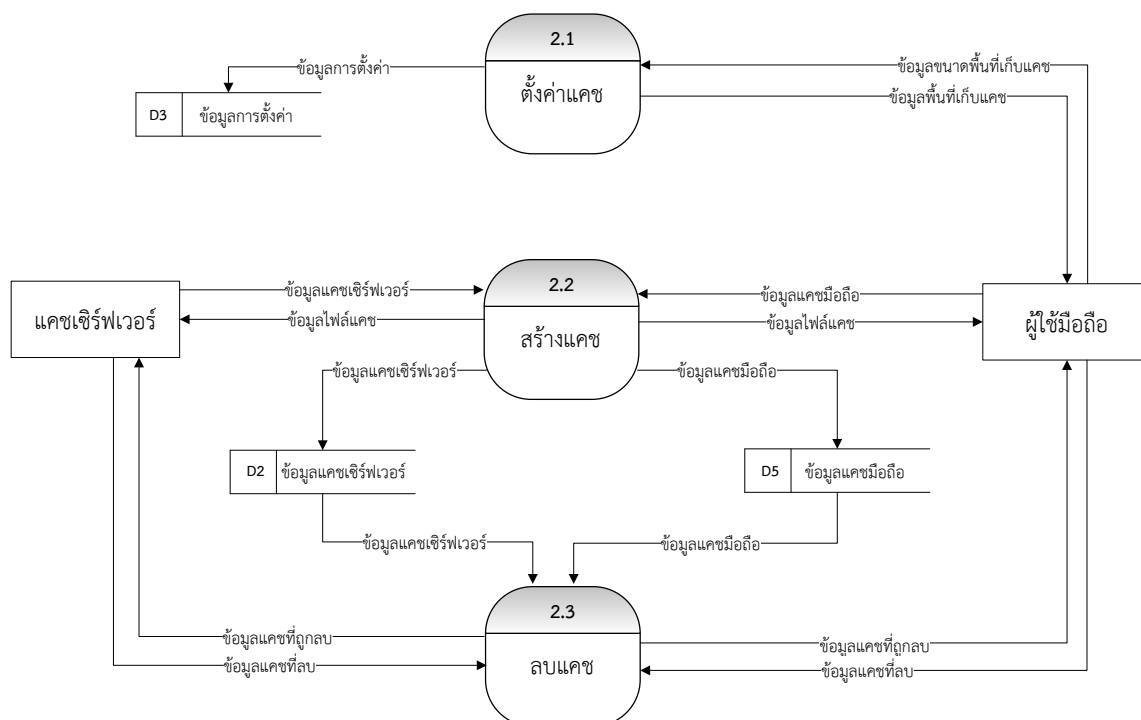
ภาพที่ 3.5 แผนภาพ Data Flow Diagram Level 0 ของระบบ

3.2.3 แผนภาพกระแสการไหลของข้อมูลระดับ 1 (Data Flow Diagram Level 1)

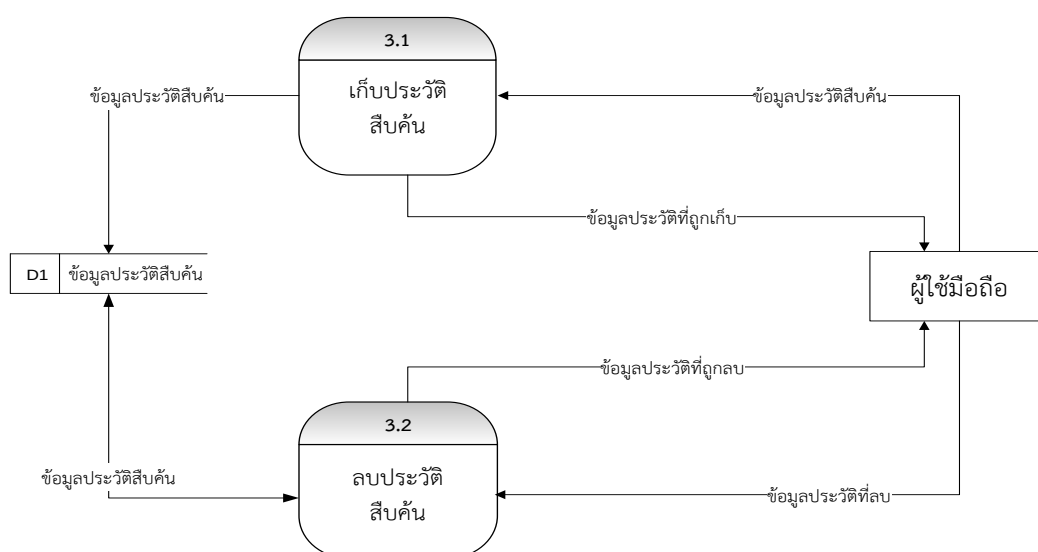
DFD Level 1 เป็นการแบ่งโปรเซสหลักเป็นโปรเซสย่อย ๆ เพื่อให้สามารถอธิบายกระบวนการทำงานให้ละเอียดมากยิ่งขึ้น โดยแสดงในภาพที่ 3.6 และ ภาพที่ 3.9



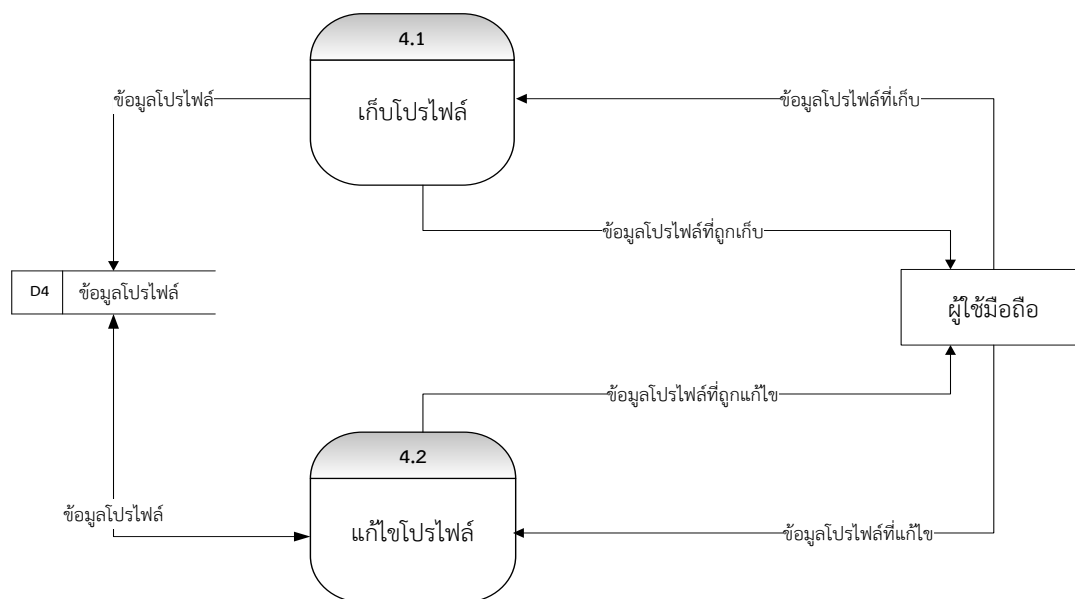
ภาพที่ 3.6 แผนภาพ Data Flow Diagram Level 1 ของกระบวนการสืบค้นข้อมูล



ภาพที่ 3.7 แผนภาพ Data Flow Diagram Level 1 ของกระบวนการจัดการแคช



ภาพที่ 3.8 แผนภาพ Data Flow Diagram Level 1 ของกระบวนการจัดการประวัติการสืบค้น

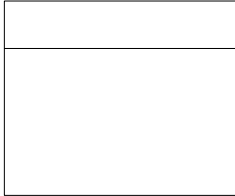


ภาพที่ 3.9 แผนภาพ Data Flow Diagram Level 1 ของกระบวนการจัดการโปรไฟล์

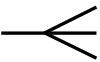
3.3 การออกแบบฐานข้อมูลด้วยแผนภาพความสัมพันธ์ของข้อมูล (E-R Diagram)

การออกแบบฐานข้อมูลเชิงสัมพันธ์โดยใช้ E-R Diagram (Entity Relationship Diagram) ซึ่งเป็นเครื่องมือในการนำเสนอโครงสร้างของฐานข้อมูลในระดับความคิด (Conceptual Level) ให้ออกมาในลักษณะของแผนภาพ (Diagram) เพื่อที่จะสื่อความหมายระหว่างนักออกแบบฐานข้อมูลและนักพัฒนาระบบให้ง่ายต่อความเข้าใจ ในงานวิจัยนี้ได้ใช้รูปแบบ Crow's Foot Model [20] ซึ่งใช้สัญลักษณ์แสดงในตารางที่ 3.2

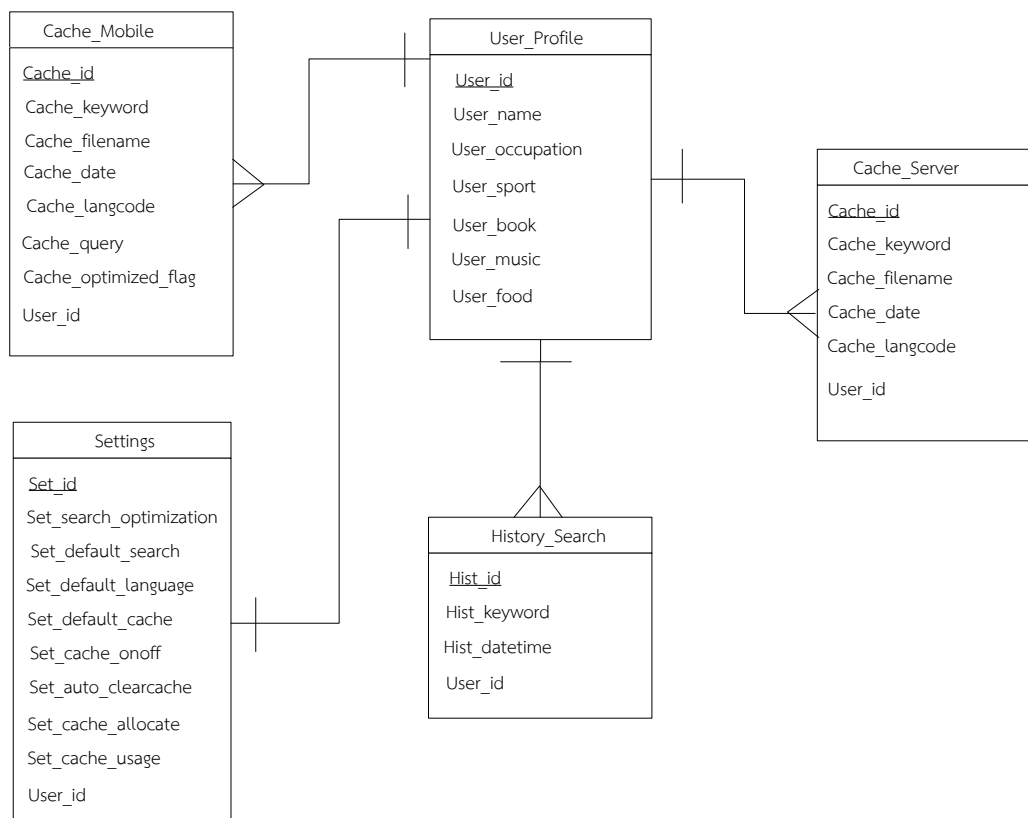
ตารางที่ 3.2 สัญลักษณ์ในการออกแบบ ER Diagram ในรูปแบบ Crow's Foot Model

สัญลักษณ์ Crow's Foot Model	ความหมาย
	เอนทิตี (Entity) หรือ กลุ่มข้อมูล
	Relationship Line เส้นเชื่อมความสัมพันธ์ระหว่างเอนทิตี
	Relationship ใช้แสดงความสัมพันธ์ระหว่างเอนทิตีแบบต่อหนึ่ง

ตารางที่ 3.2 สัญลักษณ์ในการออกแบบ ER Diagram ในรูปแบบ Crow's Foot Model (ต่อ)

สัญลักษณ์ Crow's Foot Model	ความหมาย
	Relationship ใช้แสดงความสัมพันธ์ระหว่างเอนทิตีที่แบบต่อกกลุ่ม
Entity Name <u>Identifier</u> Attribute1	การกำหนดคีย์หลัก (Primary Key) ด้วยการขีดเส้นใต้ ซึ่งเป็นข้อมูลที่ไม่ซ้ำกัน
Entity Name Attribute1 Attribute2	การกำหนดแอตทริบิวต์ให้กับเอนทิตี

3.3.1 แผนภาพ E-R Diagram การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยไอโอไนคเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส



ภาพที่ 3.10 แผนภาพ E-R Diagram ของระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยไอโอเอส

3.4 การออกแบบพจนานุกรมข้อมูล (Data Dictionary)

พจนานุกรมข้อมูลเป็นเครื่องมือที่ช่วยในการจัดเก็บข้อมูลให้เป็นหมวดหมู่ แสดงรายละเอียดตารางข้อมูลต่าง ๆ ในฐานข้อมูล ซึ่งประกอบด้วย ชื่อข้อมูล, คำอธิบายข้อมูล, ชนิดข้อมูล และ ขนาดข้อมูล โดยในงานวิจัยนี้สามารถออกแบบพจนานุกรมข้อมูลได้ดังตารางที่ 3.3 ถึง 3.9

ตารางที่ 3.3 พจนานุกรมแคชบนมือถือ (Cache_Mobile)

Field Name	Field Type	Size(Bytes)	Description
<u>Cache_id(PK)</u>	Integer	8	รหัสแคชมือถือ
Cache_keyword	Varchar	250	คีย์เวิร์ด
Cache_filename	Varchar	250	ไฟล์แคช
Cache_date	Varchar	50	วันที่เก็บแคช
Cache_langcode	Varchar	10	รหัสภาษา
Cache_query	Varchar	250	คิวรีสตริง
Cache_optimized_flag	Boolean	1	ค้นแบบเพิ่มประสิทธิภาพ
User_id(FK)	Integer	8	รหัสผู้ใช้

ตารางที่ 3.4 พจนานุกรมประวัติการสืบค้น (History_Search)

Field Name	Field Type	Size(Bytes)	Description
<u>Hist_id(PK)</u>	Integer	8	รหัสประวัติการสืบค้น
Hist_keyword	Varchar	250	ประวัติคำค้น
Hist_datetime	Varchar	50	วันที่เก็บประวัติ
User_id(FK)	Integer	8	รหัสผู้ใช้

ตารางที่ 3.5 พจนานุกรมแคชบนเซิร์ฟเวอร์ (Cache_Server)

Field Name	Field Type	Size(Bytes)	Description
<u>Cache_id(PK)</u>	Integer	8	รหัสแคชเซิร์ฟเวอร์
Cache_keyword	Varchar	250	คีย์เวิร์ด
Cache_filename	Varchar	250	ไฟล์แคช
Cache_date	Varchar	10	วันที่เก็บแคช
Cache_langcode	Varchar	10	รหัสภาษา
User_id(FK)	Integer	8	รหัสผู้ใช้

ตารางที่ 3.6 พจนานุกรมตั้งค่า (Settings)

Field Name	Field Type	Size(Bytes)	Description
<u>Set_id(PK)</u>	Integer	8	รหัสตั้งค่า
Set_default_cache	Varchar	250	ตั้งค่าการเก็บแคชเริ่มต้น
Set_default_search	Varchar	250	ตั้งค่าเว็บที่สืบค้นเริ่มต้น
Set_default_language	Varchar	250	ตั้งค่าภาษาที่สืบค้นเริ่มต้น
Set_search_optimization	Boolean	1	ตั้งค่าค้นแบบเพิ่มประสิทธิภาพ
Set_cache_onoff	Boolean	1	ตั้งค่าปิด/เปิดแคช
Set_auto_clearcache	Boolean	1	ตั้งค่าเคลียร์แคชอัตโนมัติ
Set_cache_allocate	Integer	8	ตั้งค่าขนาดแคชที่เก็บ
Set_cache_usage	Integer	8	ตั้งค่าแคชที่ถูกใช้
User_id(FK)	Integer	8	รหัสผู้ใช้

ตารางที่ 3.7 พจนานุกรมข้อมูลผู้ใช้ (User_Profile)

Field Name	Field Type	Size(Bytes)	Description
<u>User_id(PK)</u>	Integer	8	รหัสผู้ใช้
User_name	Varchar	250	ชื่อผู้ใช้
User_occupation	Varchar	250	ข้อมูลด้านอาชีพ
User_sport	Varchar	250	ข้อมูลด้านกีฬา
User_book	Varchar	250	ข้อมูลด้านหนังสือ
User_music	Varchar	250	ข้อมูลด้านดนตรี
User_food	Varchar	250	ข้อมูลด้านอาหาร

3.5 การออกแบบหน้าจอของโปรแกรม

ในงานวิจัยนี้ผู้วิจัยได้ออกแบบหน้าจอหลักดังนี้

3.5.1 หน้าจอการสืบค้นข้อมูล

โดยในหน้าจอการสืบค้นนี้ จะค้นจากเว็บไซต์สารานุกรมจากทั้ง 2 แหล่ง จึงออกแบบหน้าจอให้สามารถกดเพื่อสลับดูผลลัพธ์ที่ได้จากการสืบค้น ดังแสดงในภาพที่ 3.11



ภาพที่ 3.11 การออกแบบหน้าจอการสืบค้นข้อมูล

3.5.2 หน้าจอการแสดงผลรายละเอียดข้อมูล

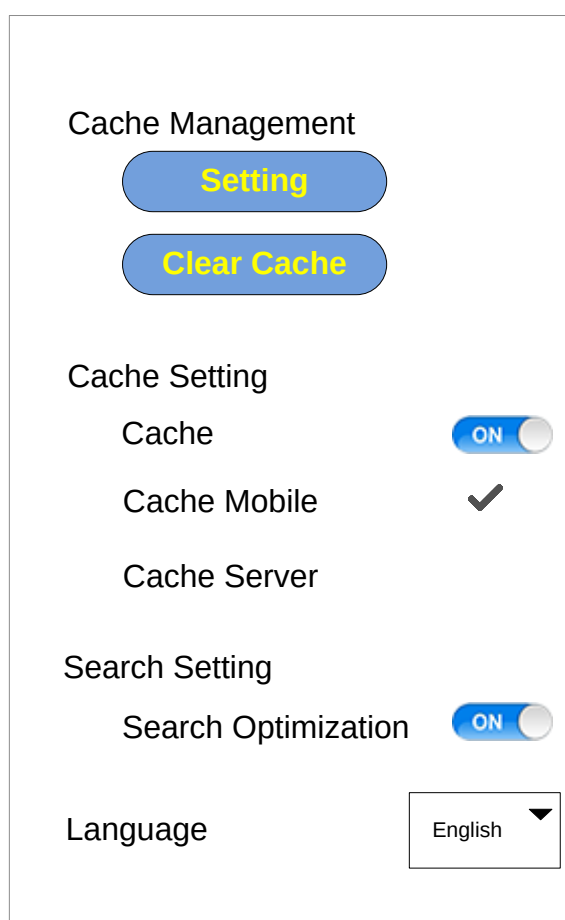
เป็นหน้าจอที่ใช้ในการแสดงรายละเอียดข้อมูลของชื่อเรื่องที่ได้จากวิกิพีเดียและคอนเซอร์เวทีฟเดียว ดังแสดงในภาพที่ 3.12



ภาพที่ 3.12 การออกแบบหน้าจอรายละเอียดข้อมูล

3.5.3 หน้าจอการตั้งค่า

เป็นหน้าจอที่ใช้สำหรับกำหนดค่าต่าง ๆ ในการสืบค้นข้อมูล เช่น การจัดการแคช การสืบค้นผ่านแคชแบบต่าง ๆ การสืบค้นแบบเพิ่มประสิทธิภาพ และการกำหนดภาษาในการสืบค้น ดังแสดงในภาพที่ 3.13



ภาพที่ 3.13 การออกแบบหน้าจอการตั้งค่า

3.6 การออกแบบการวิเคราะห์และประเมินผล

ทำการเปรียบเทียบและวิเคราะห์ผลการทดลองตามเกณฑ์การประเมินผล โดยแบ่งออกเป็น 2 รูปแบบดังนี้

3.6.1 การวัดประสิทธิภาพเวลาตอบสนองของการสืบค้นข้อมูล

การวัดประสิทธิภาพเวลาตอบสนองจะใช้เวลาเปรียบเทียบการสืบค้นในวิกิพีเดียและคอนเซอร์เวทีฟเพียง 3 แบบ คือ เวลาตอบสนองการสืบค้นผ่านแคชมือถือ เวลาตอบสนองการสืบค้นผ่านแคชเซิร์ฟเวอร์ และ เวลาตอบสนองการสืบค้นโดยไม่ผ่านแคช ในการทดลองนี้ทำการรันแอปพลิเคชันบนระบบปฏิบัติการไอโอเอสด้วยโปรแกรมจำลอง (Emulator) และใช้อินเทอร์เน็ตความเร็ว 512 Kbps โดยการใช้การหาค่าเฉลี่ยด้วยเงื่อนไขดังนี้

- 3.6.1.1 สืบค้นผ่านแคชมือถือทั้งหมด 50 คำค้น แต่ละคำค้นคลิกค้น 3 ครั้ง
- 3.6.1.2 สืบค้นผ่านแคชเวิร์ฟเวอร์ทั้งหมด 50 คำค้น แต่ละคำค้นคลิกค้น 3 ครั้ง
- 3.6.1.3 สืบค้นไม่ผ่านแคชด้วยทั้งหมด 50 คำค้น แต่ละคำค้นคลิกค้น 3 ครั้ง

3.6.2 การวัดค่าความถูกต้องของระบบ

การวัดค่าความถูกต้องของระบบจะทำการเปรียบเทียบข้อมูลผลลัพธ์จากเว็บไซต์ สารานุกรมทั้ง 2 แหล่ง คือ วิกิพีเดีย และ คอนเซอร์เวทีฟเดีย มาใช้ในการวัดความถูกต้องของระบบ โดยกำหนดรูปแบบในการทดสอบดังนี้

3.6.2.1 เปรียบเทียบผลลัพธ์ความเที่ยงตรงในหน้าการสืบค้นข้อมูลด้วย 100 คำค้น จาก เว็บไซต์วิกิพีเดีย และระบบสืบค้นข้อมูลในวิกิพีเดียที่พัฒนาขึ้น

3.6.2.2 เปรียบเทียบผลลัพธ์ความเที่ยงตรงในหน้าการสืบค้นข้อมูลด้วย 100 คำค้น จาก เว็บไซต์คอนเซอร์เวทีฟเดีย และระบบสืบค้นข้อมูลในคอนเซอร์เวทีฟเดียที่พัฒนาขึ้น โดยกำหนดเกณฑ์การให้คะแนนค่าความถูกต้อง 5 ระดับ ดังนี้

คะแนน 95 - 100 หมายถึง ดีมาก

คะแนน 85 - 94 หมายถึง ดี

คะแนน 75 - 84 หมายถึง พอใช้

คะแนน 65 - 74 หมายถึง น้อย

คะแนน 0 - 64 หมายถึง น้อยที่สุด

3.6.3 การประเมินความพึงพอใจของผู้ใช้งาน

การประเมินความพึงพอใจของผู้ใช้งานระบบ ผู้วิจัยได้แบ่งประเภทของผู้ใช้งานออกเป็น 2 ประเภท คือ กลุ่มผู้ใช้ทั่วไป จำนวน 15 คน ซึ่งประกอบด้วยคนทำงานด้านไอที 5 คน นักศึกษา ระดับอุดมศึกษา 5 คน และ นักเรียนระดับมัธยมศึกษา 5 คนและ กลุ่มผู้เชี่ยวชาญ จำนวน 5 คน ซึ่งเป็นผู้ที่มีความรู้ด้านไอทีเป็นอย่างดี โดยให้ทดสอบใช้งานระบบแล้วตอบแบบสอบถาม ซึ่งมีเกณฑ์การวัดค่าความพึงพอใจ 5 ระดับของลิเคอร์ท (Likert Scale) [21] ซึ่งมีเกณฑ์การให้คะแนนดังนี้

ระดับ 5 หมายถึง มีความพึงพอใจในระดับมากที่สุด

ระดับ 4 หมายถึง มีความพึงพอใจในระดับมาก

ระดับ 3 หมายถึง มีความพึงพอใจในระดับปานกลาง

ระดับ 2 หมายถึง มีความพึงพอใจในระดับน้อย

ระดับ 1 หมายถึง มีความพึงพอใจในระดับน้อยที่สุด

จากนั้นกำหนดเกณฑ์ในการแปลค่าของคะแนนดังนี้

ระดับคะแนน 4.50 - 5.00 หมายถึง มีความพึงพอใจในระดับดีมาก

ระดับคะแนน 3.50 - 4.49 หมายถึง มีความพึงพอใจในระดับดี

ระดับคะแนน 2.50 - 3.49 หมายถึง มีความพึงพอใจในระดับปานกลาง

ระดับคะแนน 1.50 - 2.49 หมายถึง มีความพึงพอใจในระดับน้อย

ระดับคะแนน 1.00 - 1.49 หมายถึง มีความพึงพอใจในระดับน้อยที่สุด

บทที่ 4

ผลการดำเนินการวิจัย

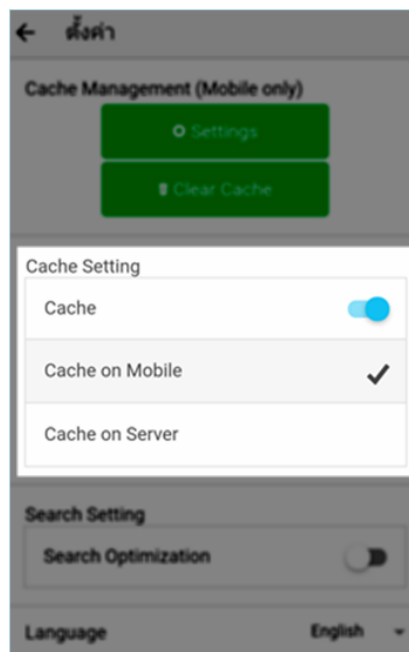
งานวิจัยฉบับนี้มีวัตถุประสงค์เพื่อพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยไอโฟนเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส ซึ่งผู้วิจัยได้ทำการศึกษารวบรวมข้อมูลที่เกี่ยวข้องในการวิเคราะห์และออกแบบระบบตามหลักวิศวกรรมซอฟต์แวร์ โดยพัฒนาเป็นแอปพลิเคชันที่สามารถนำมาใช้ในการทดสอบและประเมินผลได้ดังต่อไปนี้

- 4.1 ผลการพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟ
- 4.2 ผลการทดสอบระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟ
- 4.3 ผลการวัดประสิทธิภาพความเร็วในการสืบค้นข้อมูล
- 4.4 ผลการวัดประสิทธิภาพความถูกต้องของระบบ
- 4.5 ผลการประเมินความพึงพอใจ

4.1 ผลการพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟ

4.1.1 หน้าจอการตั้งค่าแคชสำหรับการสืบค้นข้อมูล

เป็นหน้าจอที่ใช้กำหนดค่าการใช้งานแคช โดยกำหนดค่าดังนี้ คือ ไม่ใช้แคช ใช้แคชมือถือ และใช้แคชเซิร์ฟเวอร์ กรณีที่ไม่ใช้แคชระบบจะทำการสืบค้นข้อมูลจากวิกิพีเดียและคอนเซอร์เวทีฟโดยตรง แสดงการตั้งค่าได้ดังภาพที่ 4.1 ถึง ภาพที่ 4.2



ภาพที่ 4.1 หน้าจอการตั้งค่าแคชสำหรับการสืบค้นข้อมูล

1.	\$scope.cacheSetting = function(item) {
2.	var cacheSelected = "";
3.	if(item.value == "mobile") {
4.	cacheSelected = "mobile";
5.	}else {
6.	cacheSelected = "server";
7.	}
8.	var updateDefaultCacheQuery = "UPDATE settings SET
9.	set_default_cache = ? WHERE set_id = ?";
10.	\$cordovaSQLite.execute(db, updateDefaultCacheQuery, [cacheSelected
11.	, 1]).then(function(res) {
12.	console.log("Update successfully!");
13.	}, function (err) {
14.	console.error(err.message);
15.	});
16.	}

ภาพที่ 4.2 ส่วนของโปรแกรมในการตั้งค่าแคชสำหรับการสืบค้นข้อมูล

จากภาพที่ 4.2 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

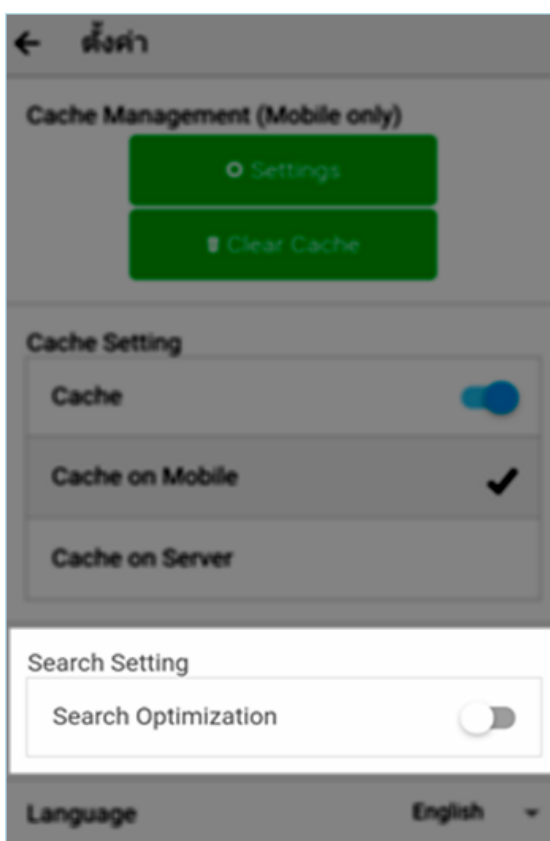
บรรทัดที่ 1 ฟังก์ชันการตั้งค่าดีฟอลต์แคช

บรรทัดที่ 2-7 ตรวจสอบเงื่อนไขและกำหนดค่าแคชที่เลือก

บรรทัดที่ 8-16 อัปเดตค่าดีฟอลต์แคชที่เลือกลงฐานข้อมูลในตารางการตั้งค่า

4.1.2 หน้าจอการตั้งค่าการสืบค้นแบบเพิ่มประสิทธิภาพ

เป็นหน้าจอที่ใช้กำหนดค่าคุณสมบัติการสืบค้นแบบเพิ่มประสิทธิภาพโดยใช้ค่าตัวแปรของข้อมูลโปรไฟล์ผู้ใช้งานมาปรับปรุงควรีให้ได้ผลลัพธ์การสืบค้นที่มีประสิทธิภาพและตรงกับผู้ใช้งานมากขึ้น แสดงการตั้งค่าสืบค้นแบบเพิ่มประสิทธิภาพ ดังภาพที่ 4.3 ถึง ภาพที่ 4.4



ภาพที่ 4.3 หน้าจอการตั้งค่าการสืบค้นแบบเพิ่มประสิทธิภาพ

1.	\$scope.toggleSearchOpt = function() {
2.	var set_opt = "";
3.	if(\$scope.searchOpt == true) {
4.	\$scope.searchOpt = false;
5.	set_opt = "off";
6.	}else {
7.	\$scope.searchOpt = true;
8.	set_opt = "on";
9.	}
10.	var updateSearchQuery = "UPDATE settings SET set_search_optimizing = ?
11.	WHERE set_id = ?";
12.	\$cordovaSQLite.execute(db, updateSearchQuery, [set_opt,
13.	1])).then(function(res) {
14.	console.log("Update successfully!");
15.	}, function (err) {
16.	console.error(err.message);
17.	});
18.	}

ภาพที่ 4.4 ส่วนของโปรแกรมในการตั้งค่าการสับค้นแบบเพิ่มประสิทธิภาพ

จากภาพที่ 4.4 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

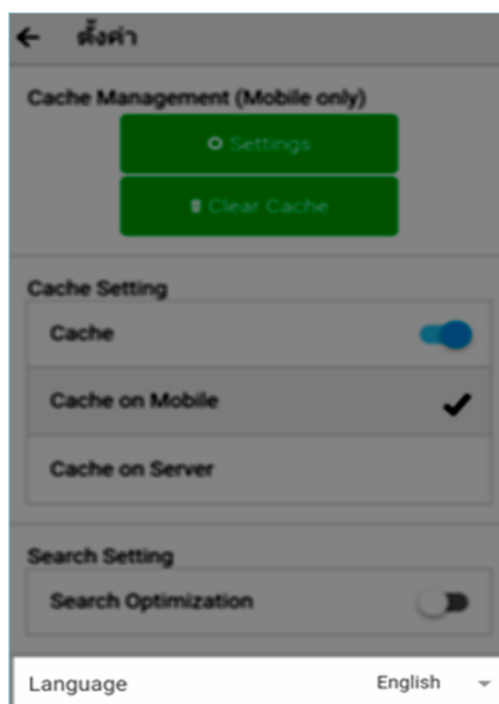
บรรทัดที่ 1 ฟังก์ชันการตั้งค่าการสับค้นแบบเพิ่มประสิทธิภาพ

บรรทัดที่ 2-9 ตรวจสอบเงื่อนไขและกำหนดค่าการสับค้นแบบเพิ่มประสิทธิภาพ

บรรทัดที่ 10-18 อัปเดตค่าการสับค้นแบบเพิ่มประสิทธิภาพลงฐานข้อมูลในตารางการตั้งค่า

4.1.3 หน้าจอการตั้งค่าภาษาสำหรับการสืบค้น

เป็นหน้าจอที่ใช้กำหนดค่าภาษาสำหรับการสืบค้นข้อมูลในวิกิพีเดีย แสดงดังภาพที่ 4.5 ถึงภาพที่ 4.6



ภาพที่ 4.5 หน้าจอการตั้งค่าภาษาสำหรับการสืบค้น

1.	\$scope.updateLang = function(lang) {
2.	var updateSearchQuery = "UPDATE settings SET set_default_language =
3.	? WHERE set_id = ?";
4.	\$cordovaSQLite.execute(db, updateSearchQuery, [lang ,
5.	1]).then(function(res) {
6.	console.log("Update successfully!");
7.	}, function (err) {
8.	console.error(err.message);
9.	});
10.	}

ภาพที่ 4.6 ส่วนของโปรแกรมในการตั้งค่าภาษาสำหรับการสืบค้น

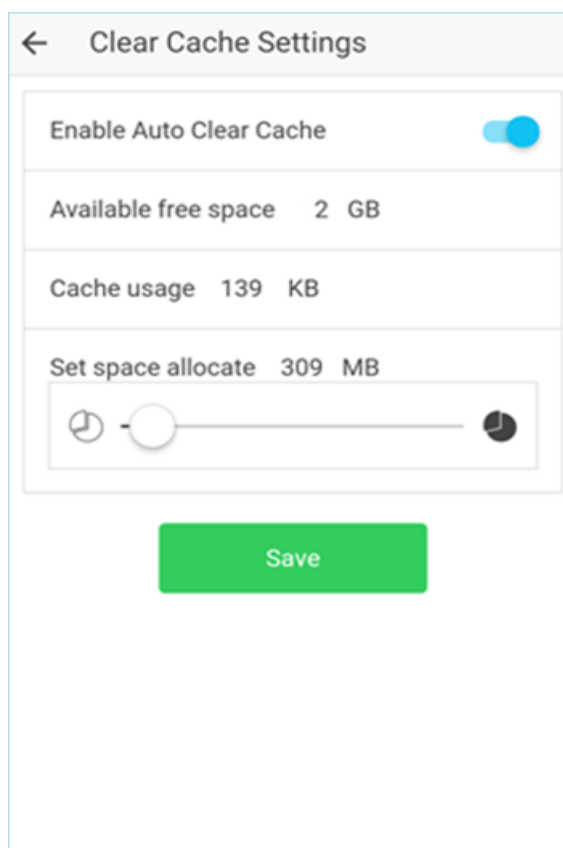
จากภาพที่ 4.6 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

บรรทัดที่ 1 ฟังก์ชันการตั้งค่าภาษาสำหรับการสืบค้น

บรรทัดที่ 2-10 อัปเดตค่าภาษาที่ใช้สำหรับการสืบค้นลงฐานข้อมูลในตารางการตั้งค่า

4.1.4 หน้าจอการตั้งค่าพื้นที่เก็บแคชและเคลียร์แคชอัตโนมัติ

เป็นหน้าจอที่ใช้กำหนดค่าพื้นที่สำหรับเก็บแคช โดยเมื่อแคชถูกเก็บจนเต็มพื้นที่ระบบจะทำการเคลียร์อัตโนมัติ แสดงดังภาพที่ 4.7 ถึง ภาพที่ 4.8



ภาพที่ 4.7 หน้าจอการตั้งค่าพื้นที่เก็บแคชและเคลียร์แคชอัตโนมัติ

1.	\$scope.toggleCacheAuto = function() {
2.	\$q.when().
3.	then(function () {
4.	var deff = \$q.defer();
5.	if(\$scope.enable_cache_auto == true) {
6.	\$scope.enable_cache_auto = false;
7.	deff.resolve(0);
8.	}else {
9.	\$scope.enable_cache_auto = true;
10.	deff.resolve(1);
11.	}
12.	return deff.promise;
13.	}).
14.	then(function (data) {
15.	var updateSettingsQuery = "UPDATE settings SET
16.	set_auto_clear_cache_enabled = ? WHERE set_id = ?";
17.	\$cordovaSQLite.execute(db, updateSettingsQuery , [data,
18.	1]).then(function(res) {
19.	Console.log("Update Successfully!");
20.	}, function (err) {
21.	console.error(err.message);
22.	});
23.	});
24.	}

ภาพที่ 4.8 ส่วนของโปรแกรมในการตั้งค่าเคลียร์แคชอัตโนมัติ

จากภาพที่ 4.8 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

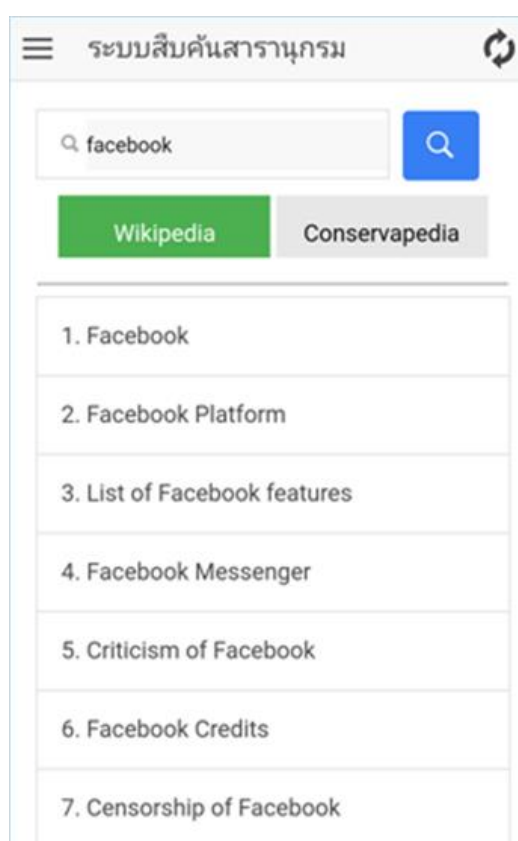
บรรทัดที่ 1 ฟังก์ชันการตั้งค่าเคลียร์แคชอัตโนมัติ

บรรทัดที่ 2-14 ตรวจสอบและกำหนดค่าการเคลียร์แคชอัตโนมัติ

บรรทัดที่ 15-24 อัปเดตค่าเคลียร์แคชอัตโนมัติลงฐานข้อมูลในตารางการตั้งค่า

4.1.5 หน้าจอการสืบค้นข้อมูล

เป็นหน้าจอที่ใช้สำหรับการสืบค้นข้อมูลสารานุกรมจากวิกิพีเดีย และคอนเซอร์เวทีเดีย โดยในการสืบค้นระบบจะทำการตรวจสอบว่าได้มีการตั้งแคชในการสืบค้นหรือไม่ หากมีการตั้งค่าแคช และพบว่าเป็นค่าที่เคยค้นไปแล้ว ระบบจะดึงผลการสืบค้นจากแคชมาแสดง แต่ถ้าหากไม่พบระบบจะทำการสืบค้นจากเว็บไซต์วิกิพีเดีย และคอนเซอร์เวทีเดียมาแสดงผล พร้อมทำการสร้างแคชเก็บไว้ แสดงดังภาพที่ 4.9 ถึง ภาพที่ 4.12



ภาพที่ 4.9 หน้าจอการสืบค้นข้อมูลในวิกิพีเดีย และคอนเซอร์เวทีเดีย

1.	\$scope.noCacheSearch = function(keyword, cachetype) {
2.	if(cachetype == "no") {
3.	\$http.jsonp('https://' + CK.getLangCode().langcode + '.wikipedia.org/w/
4.	api.php?action=query&list=search&srsearch=' + keyword +'
5.	&format=json&srprop=redirecttitle&callback=JSONP_CALLBACK')
6.	.success(function(response) {
7.	\$scope.wikilist = response.query;
8.	})
9.	.then(function (data) {
10.	\$http.jsonp('http://www.conservapedia.com/api.php?action=query
11.	&list=search&srsearch=' + \$scope.myKeyword + '&format=json
12.	&srprop=redirecttitle&callback=JSONP_CALLBACK')
13.	.success(function(response) { // for cons
14.	\$scope.constlist = response.query;
15.	});
16.	}
17.	});

ภาพที่ 4.10 ส่วนของโปรแกรมในการสืบค้นแบบไม่มีแคช

จากภาพที่ 4.10 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

บรรทัดที่ 1 ฟังก์ชันการสืบค้นข้อมูลแบบไม่มีแคช

บรรทัดที่ 2-8 ตรวจสอบถ้าไม่มีกำหนดแคชให้สืบค้นข้อมูลผ่าน API ของ Wikipedia

บรรทัดที่ 9-17 สืบค้นข้อมูลผ่าน API ของ Conservapedia

1.	\$scope.checkRowsCache(\$scope.keyword)
2.	.then(function(data){
3.	var cacheRows = data.chkrows;
4.	if(cacheRows >= 1) {
5.	var wiki_json = data.json_wiki_file;
6.	var cons_json = data.json_cons_file;
7.	\$scope.open_file_wiki =
8.	\$cordovaFile.readAsText(cordova.file.dataDirectory, wiki_json)
9.	.then(function (success) {
10.	\$scope.DetailsWiki = JSON.parse(success);
11.	}, function (error) {
12.	console.log(error.message);
13.	});
14.	\$scope.open_file_cons=
15.	\$cordovaFile.readAsText(cordova.file.dataDirectory, cons_json).
16.	then(function (success) {
17.	\$scope.DetailsCons = JSON.parse(success);
18.	},function (error) {
19.	console.log(error.message);
20.	});
21.	\$q.all([\$scope.open_file_wiki, \$scope.open_file_cons])
22.	.then(function(values) {
23.	console.log("Search finished!!");
24.	});
25.	}
26.	});

ภาพที่ 4.11 ส่วนของโปรแกรมในการสืบค้นแบบผ่านแคชบนมือถือ

จากภาพที่ 4.11 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

บรรทัดที่ 1-2 ฟังก์ชันการสืบค้นข้อมูลผ่านแคชบนมือถือ

บรรทัดที่ 3 รับค่าตัวแปรเพื่อตรวจสอบว่ามีคำค้นนี้อยู่ในแคชบนมือถือหรือไม่

บรรทัดที่ 4 ตรวจสอบว่ามีคำค้นนี้อยู่ในแคชบนมือถือหรือไม่

บรรทัดที่ 5 รับค่าตัวแปรชื่อไฟล์แคชของวิกิพีเดีย

บรรทัดที่ 6 รับค่าตัวแปรชื่อไฟล์แคชของคอนเซอร์เวทีฟ

บรรทัดที่ 7-13 กำหนดตัวแปรรับค่าฟังก์ชันการอ่านข้อมูลเจสันจากไฟล์แคชของวิกิพีเดีย

บรรทัดที่ 14-20 กำหนดตัวแปรรับค่าฟังก์ชันการอ่านข้อมูลเจสันจากไฟล์แคชของคอนเซอร์เวอพีเดีย

บรรทัดที่ 21-26 ดึงข้อมูลจากไฟล์แคชของวิกิพีเดียและคอนเซอร์เวอพีเดียมาแสดงผล

```

1. $http({ url: 'http://infocache.org/cache/checkcache.php',
2.         method: "POST",
3.         data: JSON.stringify({ data_keyword: $scope.keyword}),
4.         headers: {'Content-Type': 'application/x-www-form-urlencoded'}
5.     }).success(function(data, status, header, config) {
6.         $scope.cache_result = data.res;
7.         if($scope.cache_result == "done") {
8.             $scope.get_json_wiki =
9. $http.get('http://infocache.org/cache/getjson_wiki.php?jsonfile='+$scope.
10. keyword).success(function (response) {
11.             $scope.Details = response;
12.             }).error(function (data, status, header, config) {
13.                 console.log("Error Data: " + data);
14.             });
15.             $scope.get_json_cons =
16. $http.get('http://infocache.org/cache/getjson_cons.php?jsonfile='+$scope.
17. keyword).success(function (response) {
18.                 $scope.DetailsCons = response;
19.             }).error(function (data, status, header, config) {
20.                 console.log("Error Data: " + data);
21.             });
22.             $q.all([$scope.get_json_wiki,
23. $scope.get_json_cons]).then(function(values) {
24.                 console.log("Search finished!!");
25.             });
26.         }
27.     });

```

ภาพที่ 4.12 ตัวอย่างโปรแกรมการสืบค้นแบบผ่านแคชบนเซิร์ฟเวอร์

จากภาพที่ 4.12 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

บรรทัดที่ 1-7 ฟังก์ชันการตรวจสอบแคชบนเซิร์ฟเวอร์ว่ามีแคชหรือไม่

บรรทัดที่ 8-14 กำหนดตัวแปรรับค่าฟังก์ชันการอ่านข้อมูลเจสันของวิกิพีเดียจากเซิร์ฟเวอร์

บรรทัดที่ 15-21 กำหนดตัวแปรรับค่าฟังก์ชันการอ่านข้อมูลเจสันของคอนเซอร์เวทีฟจากเซิร์ฟเวอร์

บรรทัดที่ 22-27 ดึงข้อมูลจากแคชเซิร์ฟเวอร์ของวิกิพีเดียและคอนเซอร์เวทีฟมาแสดงผล

4.1.6 หน้าจอการจัดเก็บแคชบนเซิร์ฟเวอร์

เป็นหน้าจอที่แสดงการจัดเก็บแคชบนเซิร์ฟเวอร์ โดยแคชที่อยู่บนเซิร์ฟเวอร์จะสร้างเป็นไฟล์เจสันเก็บเอาไว้ แสดงดังภาพที่ 4.13 ถึง 4.14

No.	Keywords	Filename	Time Created
108	Orange (fruit)	wiki_594ce6a545506.json	2017-06-23 17:00:05
106	orange	wiki_orange.json	2017-06-23 16:59:17
107	orange	cons_orange.json	2017-06-23 16:59:17
105	North Korea	wiki_5944b99a17a1c.json	2017-06-17 12:09:46
104	North Korea	cons_5944b41ab4218.json	2017-06-17 11:46:18
102	korea	wiki_korea.json	2017-05-20 11:23:14
103	korea	cons_korea.json	2017-05-20 11:23:14
101	Swordfish	wiki_591fc0939cc65.json	2017-05-20 11:05:39
100	Football sport	cons_591fc06104383.json	2017-05-20 11:05:21
98	sport	wiki_sport.json	2017-05-20 11:04:36

ภาพที่ 4.13 หน้าจอการจัดเก็บแคชบนเซิร์ฟเวอร์

1.	\$put_json = array();
2.	\$val_keyword = \$_POST['c_keyword'];
3.	\$val_langcode = \$_POST['c_langcode'];
4.	\$val_jsonfile = \$_POST['c_jsonfile'];
5.	if (\$mysqli->connect_errno) {
6.	\$put_json = array ('res' => 'error1');
7.	}else {
8.	\$cache_time = date("Y-m-d H:i:s");
9.	\$cache_filename = \$val_keyword.".json";
10.	\$sql_comm = "INSERT INTO cache_server (cache_keyword,
11.	cache_filename, cache_langcode, cache_type_id,
12.	cache_time_created) VALUES ('\$val_keyword', '\$cache_filename',
13.	'\$val_langcode','1','\$cache_time)";
14.	if (\$mysqli->query(\$sql_comm) === TRUE) {
15.	file_put_contents(\$cache_filename, json_encode
16.	(\$val_jsonfile, JSON_FORCE_OBJECT));
17.	\$put_json = array (
18.	'res' => 'done',
19.	'filename_wiki' => \$c_filename.'_wiki',
20.	'filename_cons' => \$c_filename.'_cons');
21.	}else {
22.	\$put_json = array ('res' => 'error2');
23.	}
24.	}
25.	echo json_encode(\$put_json);

ภาพที่ 4.14 ส่วนของโปรแกรมในการจัดเก็บแคชบนเซิร์ฟเวอร์

จากภาพที่ 4.14 สามารถอธิบายตัวอย่างโค้ดได้ดังนี้

บรรทัดที่ 1 ประกาศตัวแปรอาร์เรย์สำหรับรับค่าข้อมูลเจสัน

บรรทัดที่ 2-4 รับค่าตัวแปรคีย์เวิร์ด รับค่าตัวแปรรหัสภาษา และ รับค่าตัวแปรชื่อไฟล์เจสัน

บรรทัดที่ 5-6 กรณีการเชื่อมต่อฐานข้อมูลมายเอสคิวแอลผิดพลาด

บรรทัดที่ 7 กรณีเชื่อมต่อฐานข้อมูลสำเร็จ

บรรทัดที่ 8-9 กำหนดค่าตัวแปรเวลา และ ค่าตัวแปรชื่อไฟล์แคช

บรรทัดที่ 10-13 คำสั่งเอสคิวแอลในการบันทึกข้อมูลบนเซิร์ฟเวอร์

บรรทัดที่ 14-24 ถ้าบันทึกข้อมูลแคชลงฐานข้อมูลสำเร็จ ให้สร้างไฟล์เจสันเก็บบนเซิร์ฟเวอร์

บรรทัดที่ 25 เข้ารหัสข้อมูลอาร์เรย์เป็นเจสันสำหรับการเรียกใช้ข้อมูลจากไคลเอนต์

4.2 ผลการทดสอบระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทิฟิเดีย

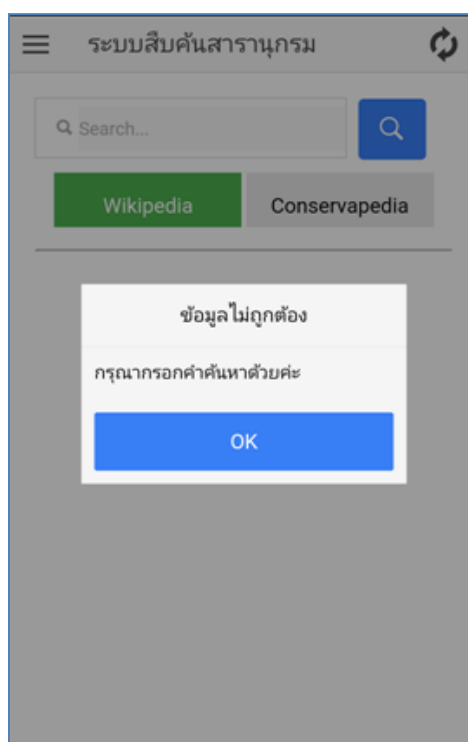
การทดสอบระบบใช้วิธีการทดสอบแบบกล่องดำ (Black Box Testing) [22] โดยให้ความสำคัญกับการนำเข้าข้อมูล (Input) ข้อมูลที่นำออก (Actual Output) และสิ่งที่คาดหวังหลังจากข้อมูลถูกประมวลผล (Expected Output)

4.2.1 การทดสอบการสืบค้นข้อมูล

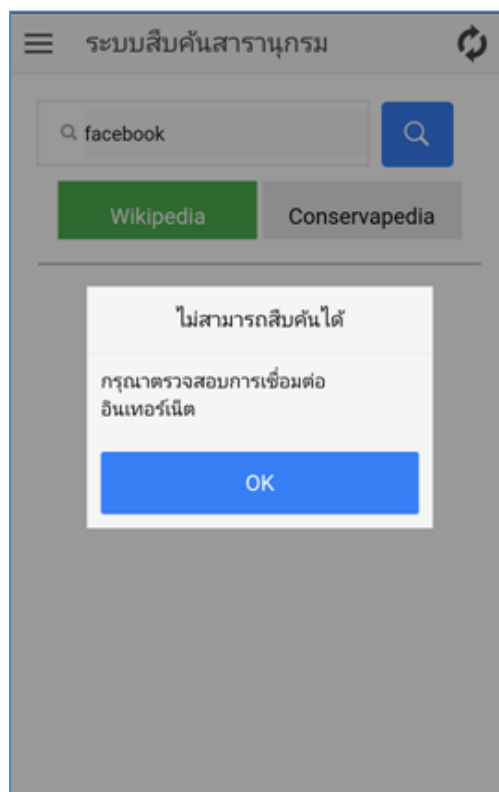
ในการทดสอบนี้จะตรวจสอบการกรอกคำค้นและการเชื่อมต่ออินเทอร์เน็ต ถ้าไม่ได้กรอกคำค้นแล้วคลิกปุ่มค้นหา ระบบจะแจ้งข้อผิดพลาดขึ้นมา หรือถ้าไม่ได้มีการเชื่อมต่ออินเทอร์เน็ตระบบก็จะแจ้งข้อผิดพลาดขึ้นมาเช่นเดียวกัน เมื่อเชื่อมต่ออินเทอร์เน็ตและกรอกคำค้นคลิกปุ่มค้นหา จะแสดงผลการค้นหาขึ้นมา แสดงได้ดังตารางที่ 4.1 และภาพที่ 4.15 ถึงภาพที่ 4.18

ตารางที่ 4.1 การทดสอบการสืบค้นข้อมูลจากวิกิพีเดีย

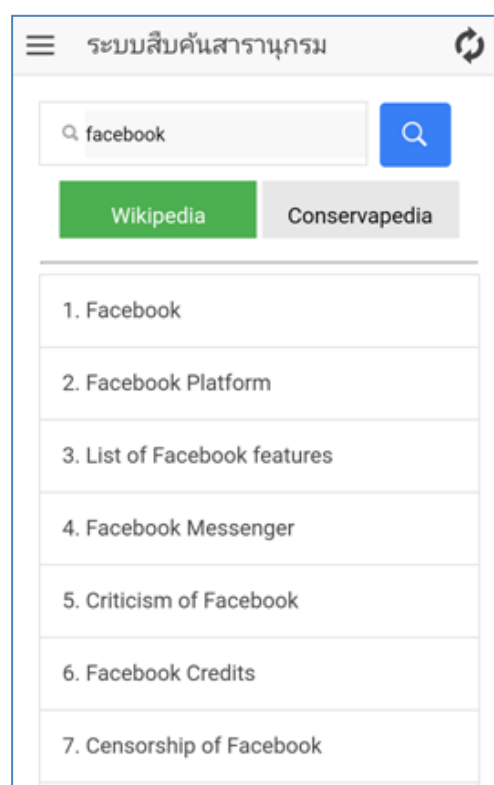
เงื่อนไขการตรวจสอบ	ผลลัพธ์
ไม่ได้กรอกคำค้น	ไม่สามารถสืบค้นได้
กรอกคำค้น ไม่ได้เชื่อมต่ออินเทอร์เน็ต	ไม่สามารถสืบค้นได้
กรอกคำค้น และเชื่อมต่ออินเทอร์เน็ต	สืบค้นข้อมูลได้ และแสดงผลการค้นหา
หน้ารายละเอียด	สามารถแสดงผลรายละเอียดของชื่อเรื่องที่คลิกได้



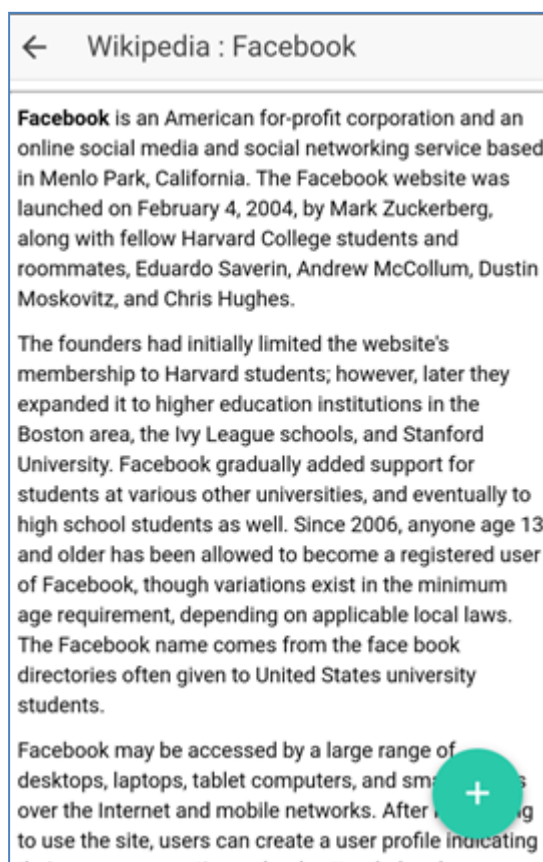
ภาพที่ 4.15 หน้าจอการแจ้งเตือนให้กรอกข้อมูลคำค้นหา



ภาพที่ 4.16 หน้าจอการแจ้งเตือนการเชื่อมต่ออินเทอร์เน็ต



ภาพที่ 4.17 หน้าจอแสดงผลลัพธ์การสืบค้น



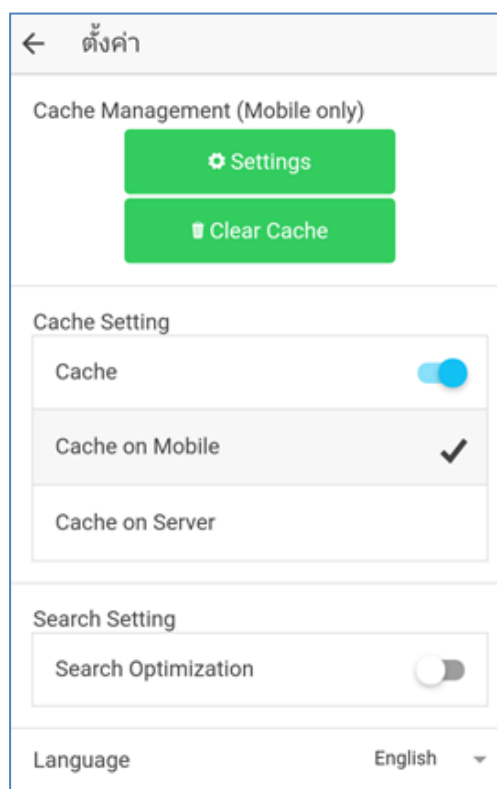
ภาพที่ 4.18 หน้าจอแสดงผลลัพธ์หน้ารายละเอียดของชื่อเรื่อง

4.2.2 การทดสอบเก็บแคชบนมือถือ

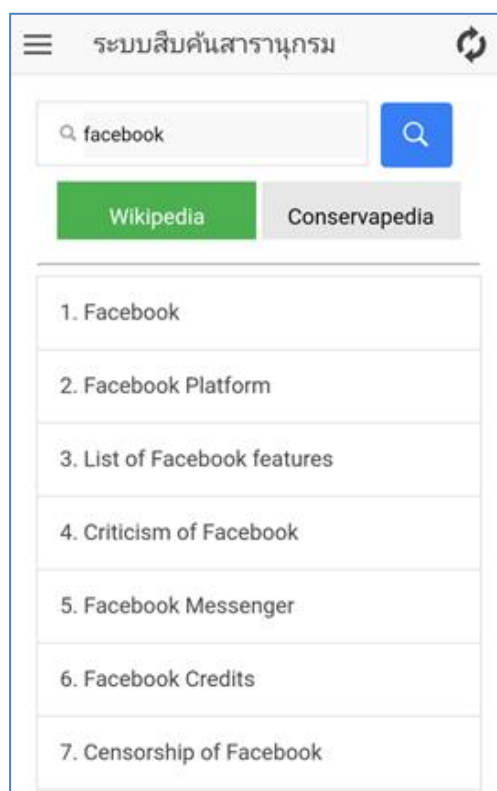
การทดสอบเก็บแคชบนมือถือมีเงื่อนไขการทดสอบและผลลัพธ์ของการทดสอบ แสดงดังตารางที่ 4.2 และภาพที่ 4.19 ถึงภาพที่ 4.22

ตารางที่ 4.2 การทดสอบการเก็บแคชบนมือถือ

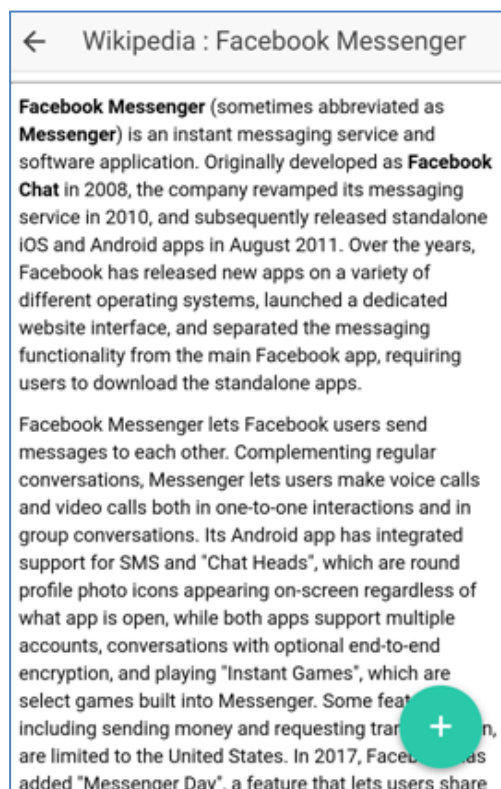
เงื่อนไขการทดสอบ	ผลลัพธ์
ตั้งค่าเก็บแคช	ระบบแคชบนมือถือเปิดใช้งาน
หน้าสืบค้นข้อมูล	เก็บแคชหน้าสืบค้น
หน้ารายละเอียด	เก็บแคชหน้ารายละเอียด



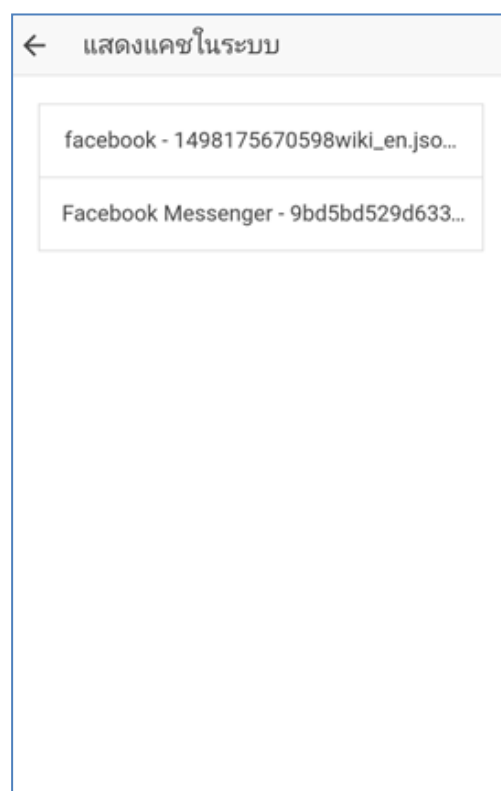
ภาพที่ 4.19 หน้าจอการตั้งค่าเปิดใช้งานระบบแคชบนมือถือ



ภาพที่ 4.20 หน้าจอการสืบค้นและเก็บแคชบนมือถือ



ภาพที่ 4.21 หน้าจอการแสดงผลรายละเอียดและเก็บแคชบนมือถือ



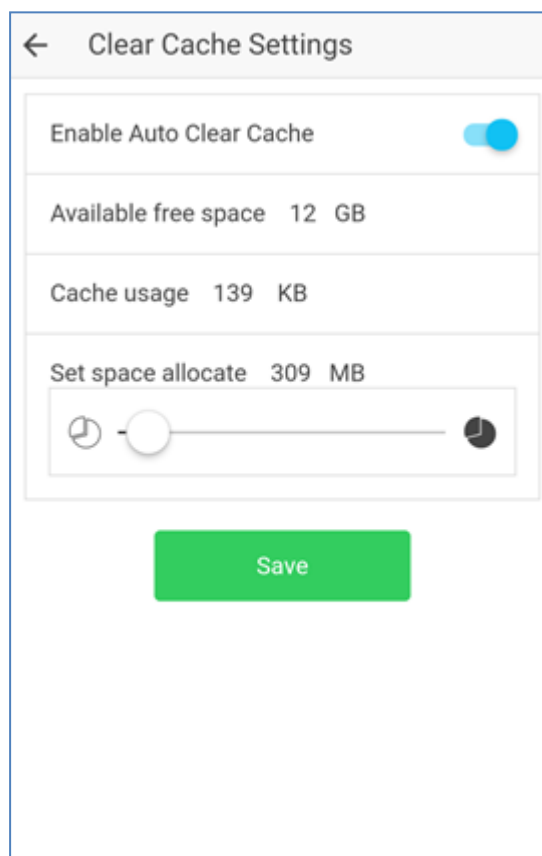
ภาพที่ 4.22 หน้าจอการแสดงผลรายการแคชไฟล์ที่ถูกจัดเก็บ

4.2.3 การทดสอบลบแคชบนมือถือ

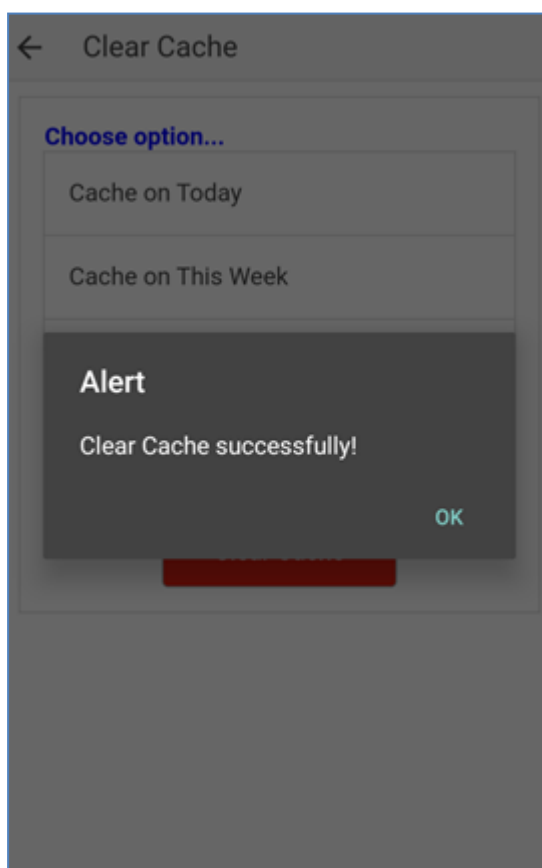
การทดสอบลบแคชบนมือถือมีวิธีการลบ 2 แบบ คือ แบบอัตโนมัติ และแบบลบแคชด้วยตัวเอง โดยมีเงื่อนไขการทดสอบและผลลัพธ์ของการทดสอบ แสดงดังตารางที่ 4.3 และภาพที่ 4.23 ถึงภาพที่ 4.24

ตารางที่ 4.3 การทดสอบลบแคชบนมือถือ

เงื่อนไขการทดสอบ	ผลลัพธ์
ตั้งค่าลบแคชอัตโนมัติ	แคชถูกลบออกจากมือถือ
ลบแคชด้วยตัวเองตามช่วงเวลา	แคชถูกลบออกจากมือถือ



ภาพที่ 4.23 หน้าจอการตั้งค่าลบแคชแบบอัตโนมัติ



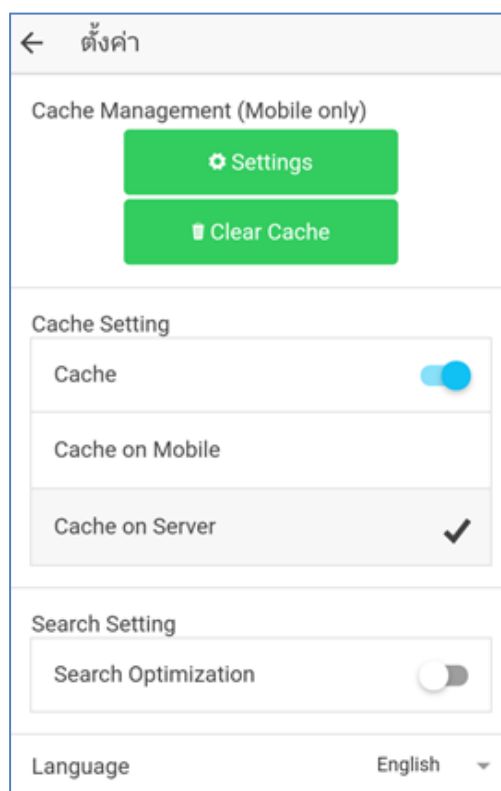
ภาพที่ 4.24 หน้าจอการลบแคชด้วยตัวเองตามช่วงเวลา

4.2.4 การทดสอบเก็บแคชบนเซิร์ฟเวอร์

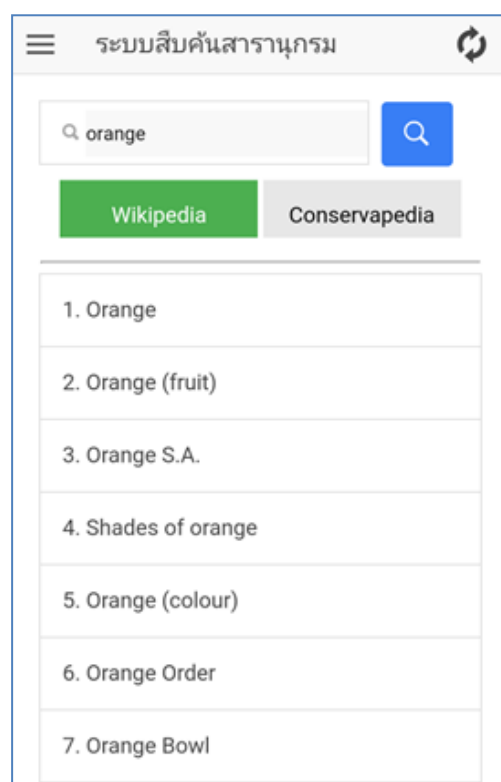
การทดสอบเก็บแคชบนเซิร์ฟเวอร์มีเงื่อนไขการทดสอบและผลลัพธ์ของการทดสอบแสดงดังตารางที่ 4.4 และภาพที่ 4.25 ถึงภาพที่ 4.28

ตารางที่ 4.4 การทดสอบเก็บแคชบนเซิร์ฟเวอร์

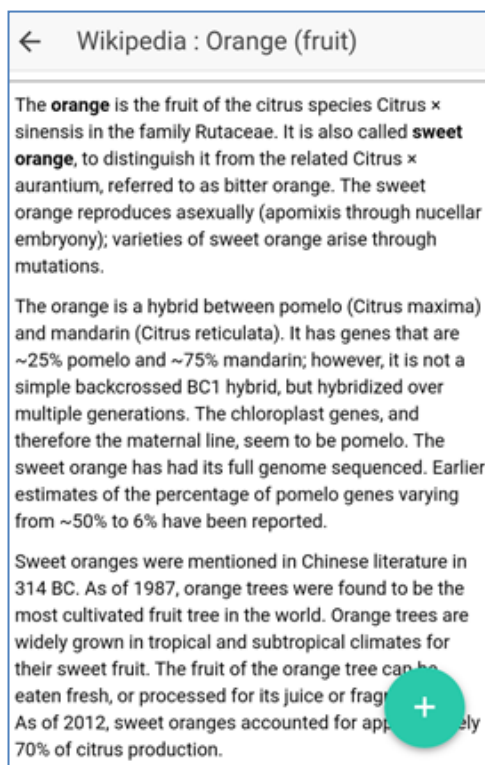
เงื่อนไขการทดสอบ	ผลลัพธ์
ตั้งค่าเก็บแคชบนเซิร์ฟเวอร์	ระบบแคชบนเซิร์ฟเวอร์เปิดใช้งาน
หน้าสืบค้นข้อมูล	เก็บแคชหน้าสืบค้น
หน้ารายละเอียด	เก็บแคชหน้ารายละเอียด



ภาพที่ 4.25 หน้าจอการตั้งค่าเปิดใช้งานระบบแคชบนเซิร์ฟเวอร์



ภาพที่ 4.26 หน้าจอการสืบค้นและเก็บแคชบนเซิร์ฟเวอร์



ภาพที่ 4.27 หน้าจอการแสดงผลรายละเอียดและเก็บแคชบนเซิร์ฟเวอร์



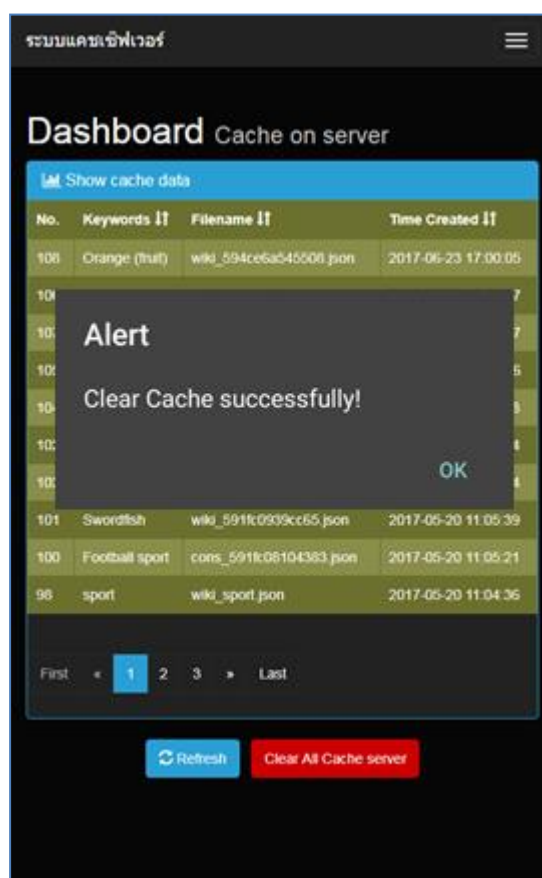
ภาพที่ 4.28 หน้าจอการแสดงผลรายการแคชไฟล์ที่ถูกจัดเก็บ

4.2.5 การทดสอบลบแคชบนเซิร์ฟเวอร์

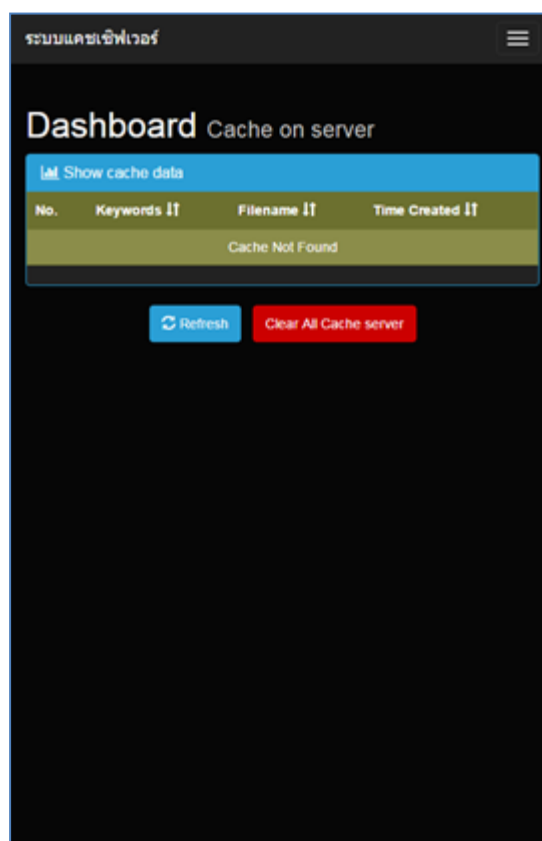
การทดสอบลบแคชบนเซิร์ฟเวอร์มีเงื่อนไขการทดสอบและผลลัพธ์ของการทดสอบ แสดงดังตารางที่ 4.5 และภาพที่ 4.29 ถึงภาพที่ 4.30

ตารางที่ 4.5 การทดสอบลบแคชจากเซิร์ฟเวอร์

เงื่อนไขการทดสอบ	ผลลัพธ์
ลบแคชบนเซิร์ฟเวอร์	แคชถูกลบออกจากเซิร์ฟเวอร์



ภาพที่ 4.29 หน้าจอการลบแคชบนเซิร์ฟเวอร์



ภาพที่ 4.30 หน้าจอแสดงแคชบนเซิร์ฟเวอร์ถูกลบออกจากระบบ

4.3 ผลการวัดประสิทธิภาพความเร็วในการสืบค้นข้อมูล

ในการวัดประสิทธิภาพความเร็วจะวัดจากค่าเวลาตอบสนองในการแสดงผลของการสืบค้นแต่ละรูปแบบ ซึ่งจำแนกออกได้ 3 แบบ คือ สืบค้นแบบผ่านแคชบนมือถือ สืบค้นแบบผ่านแคชบนเซิร์ฟเวอร์ และ สืบค้นแบบไม่ผ่านแคช โดยในงานวิจัยนี้ผู้วิจัยได้กำหนดแนวทางในการทดลองเพื่อวัดประสิทธิภาพความเร็วเฉลี่ยของเวลาการตอบสนองในแต่ละรูปแบบได้ดังนี้

- (1) การหาค่าเฉลี่ยเวลาการตอบสนองของการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดีย
- (2) การหาค่าเฉลี่ยเวลาการตอบสนองการแสดงผลหน้ารายละเอียดของชื่อเรื่องในวิกิพีเดียและคอนเซอร์เวทีฟเดีย

4.3.1 การหาค่าเฉลี่ยเวลาการตอบสนองของการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดีย

ในการทดลองนี้จะทำการรันแอปพลิเคชันบนระบบปฏิบัติการไอโอเอสด้วยโปรแกรมจำลอง (Emulator) และใช้อินเทอร์เน็ตที่มีความเร็ว 512 Kbps โดยกำหนดเงื่อนไขดังนี้

- 4.3.1.1 สืบค้นผ่านแคชมือถือทั้งหมด 50 คำค้น แต่ละคำค้นคลิกค้น 3 ครั้ง
- 4.3.1.2 สืบค้นผ่านแคชเซิร์ฟเวอร์ทั้งหมด 50 คำค้น แต่ละคำค้นคลิกค้น 3 ครั้ง
- 4.3.1.3 สืบค้นไม่ผ่านแคชด้วยทั้งหมด 50 คำค้น แต่ละคำค้นคลิกค้น 3 ครั้ง

ซึ่งจากผลการทดลองครั้งนี้ทำให้ได้ผลค่าเฉลี่ย แสดงได้ดังตารางที่ 4.6 ถึงตารางที่ 4.8

ตารางที่ 4.6 ผลการหาค่าเฉลี่ยการสืบทอดวิกิพีเดีย

การทดสอบ	ค่าเฉลี่ยเวลาตอบสนอง (มิลลิวินาที)
สืบทอดผ่านแคชบนมือถือ	9
สืบทอดผ่านแคชบนเซิร์ฟเวอร์	105
สืบทอดไม่ผ่านแคช	814

จากตารางที่ 4.6 พบว่าการสืบทอดในวิกิพีเดียแบบผ่านแคชบนมือถือใช้เวลาการตอบสนองเร็วที่สุดเฉลี่ย 9 มิลลิวินาที ในขณะที่การสืบทอดในวิกิพีเดียแบบไม่ผ่านแคชใช้เวลาการตอบสนองช้าที่สุดเฉลี่ย 812 มิลลิวินาที จากผลการทดสอบแสดงให้เห็นว่าผลการสืบทอดในวิกิพีเดียแบบผ่านแคชบนมือถือมีความเร็วเพิ่มขึ้น 98.89 เปอร์เซ็นต์ และการสืบทอดในวิกิพีเดียแบบผ่านแคชบนเซิร์ฟเวอร์มีความเร็วเพิ่มขึ้น 87.10 เปอร์เซ็นต์

ตารางที่ 4.7 ผลการหาค่าเฉลี่ยการสืบทอดบนคอนเซอร์เวตฟิเดีย

การทดสอบ	ค่าเฉลี่ยเวลาตอบสนอง (มิลลิวินาที)
สืบทอดผ่านแคชมือถือ	9
สืบทอดผ่านแคชเซิร์ฟเวอร์	108
สืบทอดไม่ผ่านแคช	826

จากตารางที่ 4.7 พบว่าการสืบทอดผ่านแคชบนมือถือในคอนเซอร์เวตฟิเดียมีค่าเฉลี่ยการตอบสนองเร็วที่สุดเฉลี่ย 9 มิลลิวินาที ในขณะที่การสืบทอดในคอนเซอร์เวตฟิเดียแบบไม่ผ่านแคชใช้เวลาการตอบสนองช้าที่สุดเฉลี่ย 826 มิลลิวินาที จากผลการทดสอบแสดงให้เห็นว่าผลการสืบทอดในคอนเซอร์เวตฟิเดียแบบผ่านแคชบนมือถือมีความเร็วเพิ่มขึ้น 98.91 เปอร์เซ็นต์ และการสืบทอดในคอนเซอร์เวตฟิเดียแบบผ่านแคชบนเซิร์ฟเวอร์มีความเร็วเพิ่มขึ้น 86.92 เปอร์เซ็นต์

ตารางที่ 4.8 ผลสรุปการหาค่าเฉลี่ยการสืบทอดในวิกิพีเดียและคอนเซอร์เวตฟิเดีย

การทดสอบ	วิกิพีเดีย (มิลลิวินาที)	คอนเซอร์เวตฟิเดีย (มิลลิวินาที)
สืบทอดผ่านแคชมือถือ	9	9
สืบทอดผ่านแคชเซิร์ฟเวอร์	105	108
สืบทอดไม่ผ่านแคช	814	826

จากตารางที่ 4.8 พบว่าการสื่บค้นผ่านแคชบนมือถือในวิกิพีเดียและคอนเซอร์เวทีเดีย่นั้นมีค่าเฉลี่ยการตอบสนองเร็วที่สุดเฉลี่ย 9 มิลลิวินาที ในขณะที่การสื่บค้นในคอนเซอร์เวทีเดียแบบไม่ผ่านแคชใช้เวลาการตอบสนองช้าที่สุดเฉลี่ย 826 มิลลิวินาที จากผลการทดสอบแสดงให้เห็นว่าผลการสื่บค้นในวิกิพีเดียแบบผ่านแคชบนมือถือมีความเร็วเพิ่มขึ้น 98.91 เปอร์เซ็นต์ และการสื่บค้นในวิกิพีเดียแบบผ่านแคชบนเซิร์ฟเวอร์มีความเร็วเพิ่มขึ้น 87.29 เปอร์เซ็นต์

4.3.2 การหาค่าเฉลี่ยเวลาการตอบสนองการแสดงผลหน้ารายละเอียดของชื่อเรื่องในวิกิพีเดียและคอนเซอร์เวทีเดีย

ในการทดลองนี้จะทำการรันแอปพลิเคชันบนระบบปฏิบัติการไอโอเอสด้วยโปรแกรมจำลอง (Emulator) และใช้อินเทอร์เน็ตที่ความเร็ว 512 Kbps โดยกำหนดเงื่อนไขดังนี้

4.3.2.1 แสดงรายละเอียดชื่อเรื่องผ่านแคชมือถือทั้งหมด 50 ชื่อเรื่อง แต่ละชื่อเรื่องคลิกดูรายละเอียด 3 ครั้ง

4.3.2.2 แสดงรายละเอียดชื่อเรื่องผ่านแคชเซิร์ฟเวอร์ทั้งหมด 50 ชื่อเรื่อง แต่ละชื่อเรื่องคลิกดูรายละเอียด 3 ครั้ง

4.3.2.3 แสดงรายละเอียดชื่อเรื่องไม่ผ่านแคชด้วยทั้งหมด 50 ชื่อเรื่อง แต่ละชื่อเรื่องคลิกดูรายละเอียด 3 ครั้ง

ซึ่งจากผลการทดลองครั้งนี้ทำให้ได้ผลค่าเฉลี่ย แสดงได้ดังตารางที่ 4.9 ถึงตารางที่ 4.11

ตารางที่ 4.9 ผลการหาค่าเฉลี่ยการแสดงผลรายละเอียดชื่อเรื่องบนวิกิพีเดีย

การทดสอบ	ค่าเฉลี่ยเวลาตอบสนอง (มิลลิวินาที)
แสดงรายละเอียดชื่อเรื่องผ่านแคชมือถือ	30
แสดงรายละเอียดชื่อเรื่องผ่านแคชเซิร์ฟเวอร์	189
แสดงรายละเอียดชื่อเรื่องไม่ผ่านแคช	1021

จากตารางที่ 4.9 พบว่าแสดงรายละเอียดชื่อเรื่องในวิกิพีเดียผ่านแคชบนมือถือนั้นมีค่าเฉลี่ยเวลาตอบสนองเร็วที่สุดเฉลี่ย 30 มิลลิวินาที ในขณะที่การแสดงผลรายละเอียดชื่อเรื่องในวิกิพีเดียโดยไม่ผ่านแคชใช้เวลาตอบสนองช้าที่สุดเฉลี่ย 1021 มิลลิวินาที จากผลการทดสอบแสดงให้เห็นว่าการแสดงผลหน้ารายละเอียดชื่อเรื่องในวิกิพีเดียแบบผ่านแคชบนมือถือมีความเร็วเพิ่มขึ้น 97.06 เปอร์เซ็นต์ และการหน้ารายละเอียดชื่อเรื่องในวิกิพีเดียแบบผ่านแคชบนเซิร์ฟเวอร์มีความเร็วเพิ่มขึ้น 81.49 เปอร์เซ็นต์

ตารางที่ 4.10 ผลการหาค่าเฉลี่ยการแสดงผลละเอียดชื่อเรื่องบนคอนเซอร์เวทีเดีย

การทดสอบ	ค่าเฉลี่ยเวลาตอบสนอง (มิลลิวินาที)
แสดงผลละเอียดชื่อเรื่องผ่านแคชมือถือ	33
แสดงผลละเอียดชื่อเรื่องผ่านแคชเซิร์ฟเวอร์	205
แสดงผลละเอียดชื่อเรื่องไม่ผ่านแคช	2204

จากตารางที่ 4.10 พบว่าการแสดงผลละเอียดชื่อเรื่องในคอนเซอร์เวทีเดียผ่านแคชบนมือถือนั้นมีค่าเฉลี่ยการตอบสนองเร็วที่สุดเฉลี่ย 33 มิลลิวินาที ในขณะที่การแสดงผลละเอียดชื่อเรื่องในคอนเซอร์เวทีเดียแบบไม่ผ่านแคชใช้เวลาตอบสนองช้าที่สุดเฉลี่ย 2204 มิลลิวินาที จากผลการทดสอบแสดงให้เห็นว่าการแสดงผลหน้ารายละเอียดชื่อเรื่องในคอนเซอร์เวทีเดียแบบผ่านแคชบนมือถือมีความเร็วเพิ่มขึ้น 98.50 เปอร์เซ็นต์ และการแสดงผลหน้ารายละเอียดชื่อเรื่องในคอนเซอร์เวทีเดียแบบผ่านแคชบนเซิร์ฟเวอร์มีความเร็วเพิ่มขึ้น 90.70 เปอร์เซ็นต์

ตารางที่ 4.11 สรุปการหาค่าเฉลี่ยการแสดงผลละเอียดชื่อเรื่องบนวิกิพีเดียและคอนเซอร์เวทีเดีย

การทดสอบ	วิกิพีเดีย (มิลลิวินาที)	คอนเซอร์เวทีเดีย (มิลลิวินาที)
แสดงผลละเอียดชื่อเรื่องผ่านแคชมือถือ	30	33
แสดงผลละเอียดชื่อเรื่องผ่านแคชเซิร์ฟเวอร์	189	205
แสดงผลละเอียดชื่อเรื่องไม่ผ่านแคช	1021	2204

จากตารางที่ 4.11 พบว่าการแสดงผลละเอียดชื่อเรื่องในวิกิพีเดียผ่านแคชบนมือถือนั้นมีค่าเฉลี่ยการตอบสนองเร็วที่สุดเฉลี่ย 30 มิลลิวินาที ในขณะที่การแสดงผลละเอียดชื่อเรื่องโดยไม่ผ่านแคชในคอนเซอร์เวทีเดียใช้เวลาตอบสนองช้าที่สุดเฉลี่ย 2204 มิลลิวินาที จากผลการทดสอบแสดงให้เห็นว่าการแสดงผลหน้ารายละเอียดชื่อเรื่องในวิกิพีเดียแบบผ่านแคชบนมือถือมีความเร็วเพิ่มขึ้น 98.64 เปอร์เซ็นต์ และการแสดงผลหน้ารายละเอียดชื่อเรื่องในวิกิพีเดียแบบผ่านแคชบนเซิร์ฟเวอร์มีความเร็วเพิ่มขึ้น 91.42 เปอร์เซ็นต์

4.4 ผลการวัดค่าความถูกต้องของระบบ

การวัดค่าความถูกต้องของระบบโดยใช้การเปรียบเทียบผลลัพธ์ที่ได้จากการสืบค้นข้อมูลของระบบที่พัฒนาขึ้นกับเว็บไซต์สารานุกรมวิกิพีเดีย และคอนเซอร์เวทีฟเดีย โดยใช้การหาค่าเฉลี่ยเปอร์เซ็นต์ความถูกต้องในการแสดงผลการสืบค้น โดยกำหนดเงื่อนไขดังนี้

4.4.1 ทำการเปรียบเทียบผลลัพธ์ความเที่ยงตรงจากการสืบค้นข้อมูลในเว็บไซต์วิกิพีเดีย และระบบสืบค้นข้อมูลในวิกิพีเดียที่พัฒนาขึ้นโดยใช้คำค้นเดียวกันจำนวน 100 คำค้น

4.4.2 ทำการเปรียบเทียบผลลัพธ์ความเที่ยงตรงจากการสืบค้นข้อมูลเว็บไซต์คอนเซอร์เวทีฟเดีย และระบบสืบค้นข้อมูลในคอนเซอร์เวทีฟเดียที่พัฒนาขึ้นโดยใช้คำค้นเดียวกันจำนวน 100 คำค้น กำหนดเกณฑ์การให้คะแนน 5 ระดับ ดังนี้

คะแนน 95 - 100 หมายถึง ดีมาก

คะแนน 85 - 94 หมายถึง ดี

คะแนน 75 - 84 หมายถึง พอใช้

คะแนน 65 - 74 หมายถึง น้อย

คะแนน 0 - 64 หมายถึง น้อยที่สุด

ซึ่งจากผลการทดลองครั้งนี้ทำให้ได้ผลค่าเฉลี่ย แสดงได้ดังตารางที่ 4.12

ตารางที่ 4.12 ผลสรุปค่าความถูกต้องของผลการสืบค้นข้อมูลในกับวิกิพีเดียและคอนเซอร์เวทีฟเดีย

การทดสอบ	วิกิพีเดีย (เปอร์เซ็นต์)	คอนเซอร์เวทีฟเดีย (เปอร์เซ็นต์)
ค่าความถูกต้องในการแสดงผลเฉลี่ย	92	98

จากตารางที่ 4.12 พบว่าการแสดงค่าความถูกต้องจากผลการสืบค้นข้อมูลในวิกิพีเดียเฉลี่ย 92 เปอร์เซ็นต์ ซึ่งอยู่ในเกณฑ์ที่ดี และค่าความถูกต้องจากผลการสืบค้นข้อมูลในคอนเซอร์เวทีฟเดียเฉลี่ย 98 เปอร์เซ็นต์ ซึ่งอยู่ในเกณฑ์ดีมาก

4.5 ผลการประเมินความพึงพอใจ

จากการพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดียด้วยโอโคโนนิเฟรมเวิร์ก เจสัน และแคช บนระบบปฏิบัติการโอโอเอส ผู้วิจัยได้ออกแบบการประเมินผลวัดความพึงพอใจต่อการใช้ระบบ โดยแบ่งเป็นกลุ่มผู้ใช้งานทั่วไป และ กลุ่มผู้เชี่ยวชาญ ในการให้คะแนนแบบสอบถามในแต่ละด้าน โดยมีเกณฑ์การให้คะแนน 5 ระดับ ดังนี้

ระดับ 5 หมายถึง มีความพึงพอใจในระดับมากที่สุด

ระดับ 4 หมายถึง มีความพึงพอใจในระดับมาก

ระดับ 3 หมายถึง มีความพึงพอใจในระดับปานกลาง

ระดับ 2 หมายถึง มีความพึงพอใจในระดับน้อย

ระดับ 1 หมายถึง มีความพึงพอใจในระดับน้อยที่สุด

เกณฑ์คะแนนเฉลี่ยของแบบประเมินความพึงพอใจของผู้ใช้งานระบบ แสดงดังตารางที่ 4.13

ตารางที่ 4.13 เกณฑ์คะแนนเฉลี่ยของแบบประเมิน

ระดับคะแนน	ความหมาย
4.50-5.00	มีความพึงพอใจในระดับดีมาก
3.50-4.49	มีความพึงพอใจในระดับดี
2.50-3.49	มีความพึงพอใจในระดับพอใช้
1.50-2.49	มีความพึงพอใจในระดับน้อย
1.00-1.49	มีความพึงพอใจในระดับน้อยที่สุด

4.5.1 การประเมินความพึงพอใจกลุ่มผู้ใช้งานทั่วไป

ในการประเมินความพึงพอใจจะใช้กลุ่มตัวอย่างที่เป็นผู้ใช้งานทั่วไปจำนวน 15 คน ซึ่งประกอบด้วยคนทำงานด้านไอที 5 คน นักศึกษาระดับอุดมศึกษา 5 คนและนักเรียนระดับมัธยมศึกษา 5 คนโดยให้ทดลองใช้งานแอปพลิเคชันบนระบบปฏิบัติการไอโอเอส แล้วใช้แบบสอบถามในการให้คะแนนในแต่ละด้าน ซึ่งมีทั้งหมด 5 ด้านประกอบด้วย ด้านความสวยงาม ด้านความถูกต้องของข้อมูล ด้านความง่ายต่อการใช้งาน ด้านความเร็วในการประมวลผล และด้านฟังก์ชันการใช้งานตรงกับความต้องการ โดยผลที่ได้จากการประเมินแสดงดังตารางที่ 4.14

ตารางที่ 4.14 ผลการประเมินความพึงพอใจต่อการใช้งานระบบของผู้ใช้งานทั่วไป

รายการประเมิน	ระดับความพึงพอใจ		
	ค่าเฉลี่ย	ค่าเบี่ยงเบนมาตรฐาน	คุณภาพ
1. ด้านความสวยงาม	4.20	0.56	ดี
2. ด้านความถูกต้องของข้อมูล	4.07	0.70	ดี
3. ด้านความง่ายต่อการใช้งาน	4.27	0.59	ดี
4. ด้านความเร็วในการประมวลผล	4.14	0.64	ดี
5. ด้านฟังก์ชันการใช้งานตรงกับความต้องการ	3.94	0.70	ดี
สรุปผล	4.12	0.63	ดี

จากตารางที่ 4.14 พบว่าการประเมินความพึงพอใจต่อการใช้งานระบบของผู้ใช้งานทั่วไปในด้านความสวยงาม ด้านความถูกต้องของข้อมูล ด้านความง่ายต่อการใช้งาน ด้านความเร็วในการประมวลผล และ ด้านฟังก์ชันการใช้งานตรงกับความต้องการ มีค่าเฉลี่ยเท่ากับ 4.12 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.63 ซึ่งอยู่ในเกณฑ์ที่ดี

4.5.2 การประเมินความพึงพอใจกลุ่มผู้เชี่ยวชาญ

ในการประเมินความพึงพอใจจะใช้กลุ่มตัวอย่างที่เป็นผู้เชี่ยวชาญจำนวน 5 คน โดยผู้เชี่ยวชาญเป็นผู้ที่มีความรู้ด้านไอทีเป็นอย่างดี ซึ่งในการทดลองจะใช้แบบสอบถามในการให้คะแนนในแต่ละด้าน ซึ่งมีทั้งหมด 2 ด้าน ประกอบด้วย ด้านความถูกต้องของข้อมูล และ ด้านความเร็วในการประมวลผล โดยผลที่ได้จากการประเมินแสดงดังตารางที่ 4.15

ตารางที่ 4.15 ผลการประเมินความพึงพอใจต่อการใช้งานระบบของผู้เชี่ยวชาญ

รายการประเมิน	ระดับความพึงพอใจ		
	ค่าเฉลี่ย	ค่าเบี่ยงเบนมาตรฐาน	คุณภาพ
1. ด้านความถูกต้องของข้อมูล	3.80	0.45	ดี
2. ด้านความเร็วในการประมวลผล	4.00	0.71	ดี
สรุปผล	3.90	0.56	ดี

จากตารางที่ 4.15 พบว่าการประเมินความพึงพอใจต่อการใช้งานระบบของผู้เชี่ยวชาญในด้านด้านความถูกต้องของข้อมูล และ ด้านความเร็วในการประมวลผล มีค่าเฉลี่ยเท่ากับ 3.90 และ ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.56 ซึ่งอยู่ในเกณฑ์ที่ดี

บทที่ 5

สรุปผล และข้อเสนอแนะ

ผู้วิจัยได้สรุปผลการดำเนินการวิจัยการพัฒนากระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยไอโอดีนเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส รวมทั้งข้อเสนอแนะสำหรับการประยุกต์ใช้งาน ซึ่งมีรายละเอียดของเนื้อหาประกอบด้วยหัวข้อ ดังต่อไปนี้

5.1 สรุปผลการวิจัย

5.2 ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟด้วยไอโอดีนเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส ซึ่งผู้วิจัยได้คำนึงถึงประโยชน์ที่ได้จากสืบค้นข้อมูลจากเว็บไซต์สารานุกรมวิกิพีเดียและคอนเซอร์เวทีฟ ซึ่งถือเป็นแหล่งสารสนเทศที่สำคัญสำหรับผู้ใช้งานทั่วโลกทั้งในปัจจุบันและอนาคต จึงได้พัฒนาแอปพลิเคชันนี้ขึ้นมาเพื่อเป็นเครื่องมือในการสนับสนุนและอำนวยความสะดวกในการสืบค้นข้อมูลสารานุกรมให้มีประสิทธิภาพมากขึ้น

จากการทดสอบวัดประสิทธิภาพความเร็วในการตอบสนองการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟพบว่าการสืบค้นข้อมูลในคอนเซอร์เวทีฟแบบไม่ใช้แคชนั้นใช้เวลาการตอบสนองช้าที่สุดเฉลี่ย 826 มิลลิวินาที ในขณะที่การสืบค้นผ่านแคชบนมือถือใช้เวลาการตอบสนองเร็วที่สุดเฉลี่ย 9 มิลลิวินาที ผลการทดสอบแสดงให้เห็นว่าการนำแคชมาใช้สืบค้นข้อมูลช่วยให้ประสิทธิภาพความเร็วในการแสดงผลเพิ่มขึ้น 98.91 เปอร์เซ็นต์ ในส่วนของการแสดงผลหน้ารายละเอียดของชื่อเรื่องพบว่าการแสดงผลหน้ารายละเอียดในคอนเซอร์เวทีฟแบบไม่ใช้แคชนั้นใช้เวลาการตอบสนองช้าที่สุดเฉลี่ย 2204 มิลลิวินาที ในขณะที่การแสดงผลหน้ารายละเอียดของชื่อเรื่องในวิกิพีเดียผ่านแคชบนมือถือใช้เวลาตอบสนองเร็วที่สุดเฉลี่ย 30 มิลลิวินาที ผลการทดสอบแสดงให้เห็นว่าการนำแคชมาใช้ในการแสดงผลหน้ารายละเอียดของชื่อเรื่องช่วยให้ประสิทธิภาพความเร็วในการแสดงผลเพิ่มขึ้น 98.64 เปอร์เซ็นต์ ดังนั้นจึงสรุปได้ว่าการสืบค้นข้อมูลผ่านแคชที่ใช้รูปแบบการแลกเปลี่ยนข้อมูลแบบเจสันช่วยเพิ่มประสิทธิภาพความเร็วในการตอบสนองรวดเร็วมากขึ้นและช่วยลดแบนด์วิดท์ของการใช้งานอินเทอร์เน็ตอีกด้วย

การประเมินความพึงพอใจต่อการใช้งานระบบของกลุ่มผู้ใช้งานในแต่ละด้านรวมกันค่าเฉลี่ยเท่ากับ 4.12 ค่าความเบี่ยงเบนมาตรฐานเท่ากับ 0.63 ซึ่งถือว่าอยู่ในระดับดี และความพึงพอใจของกลุ่มผู้เชี่ยวชาญในแต่ละด้านรวมกันค่าเฉลี่ยเท่ากับ 3.90 ค่าความเบี่ยงเบนมาตรฐานเท่ากับ 0.56 ซึ่งถือว่าอยู่ในระดับดี จึงสรุปได้ว่าระบบที่พัฒนาขึ้นมาสามารถนำไปใช้ในการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟได้จริง

5.2 ข้อเสนอแนะ

การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวตีฟด้วยไอออนิกเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการโอโอเอส ผู้วิจัยมีข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาและการประยุกต์ใช้ต่อไป ดังต่อไปนี้

5.2.1 งานวิจัยนี้พัฒนาขึ้นเพื่อใช้งานบนระบบปฏิบัติการโอโอเอสโดยเฉพาะ ระบบจึงยังต้องพัฒนาเพื่อให้รองรับปฏิบัติการอื่น ๆ ด้วย ซึ่งมีอยู่ในปัจจุบันนั้นใช้ระบบปฏิบัติการที่หลากหลาย เช่น แอนดรอยด์ วินโดวส์โฟน ซิมเบียน แบล็คเบอร์รี่ เป็นต้น

5.2.2 งานวิจัยนี้พัฒนาขึ้นเพื่อการสืบค้นข้อมูลสารานุกรมในเว็บไซต์วิกิพีเดียและคอนเซอร์เวตีฟ โดยทั้งนี้ผู้วิจัยมองว่าเป็นสารานุกรมเสรีที่ได้รับความนิยมและเข้าถึงได้ง่าย อย่างไรก็ตามในปัจจุบันมีเว็บไซต์สารานุกรมเกิดขึ้นจำนวนมากที่มีคุณภาพและเปิดให้เข้าถึงแบบเสรีได้ ซึ่งสามารถนำแอปพลิเคชันนี้ไปพัฒนาต่อยอดเพื่อให้สืบค้นจากเว็บไซต์สารานุกรมจากแหล่งอื่น ๆ เพิ่มเติมได้ ช่วยให้ผู้ใช้งานได้รับข้อมูลที่เป็นประโยชน์มากขึ้นด้วย

5.2.3 ในส่วนของการสืบค้นโดยใช้คุณลักษณะเพิ่มเติมเพื่อใช้เป็นค่านำหนักในการสืบค้นนั้น งานวิจัยนี้ผู้วิจัยได้ใช้ข้อมูลโปรไฟล์ผู้ใช้ในการปรับปรุงคิวรีเพื่อเพิ่มประสิทธิภาพในการสืบค้น แต่ยังไม่แสดงผลการสืบค้นได้ไม่ค่อยดี ซึ่งจุดนี้สามารถนำไปพัฒนาเพิ่มเติมด้วยการใช้อัลกอริทึมในการกำหนดค่านำหนักที่เหมาะสมเพื่อให้ได้ผลลัพธ์ของการสืบค้นมีความเที่ยงตรงและแม่นยำมากขึ้น

เอกสารอ้างอิง

เอกสารอ้างอิง

- [1] Alen Šimec. “Comparison of JSON and XML Data Formats”, In **Central European Conference on Information and Intelligent Systems**. p.272-275. Varazdin: University of Zagreb, 2014.
- [2] Wikipedia. “Wikipedia”, **ลักษณะสารานุกรม**. <https://th.wikipedia.org/wiki/วิกิพีเดีย>. 10 กรกฎาคม, 2557.
- [3] Wikipedia. “Conservapedia”, **Comparison to Wikipedia**. <https://en.wikipedia.org/wiki/Conservapedia>. 7 พฤศจิกายน, 2559.
- [4] Box Single. “JSON”, **ทำความเข้าใจกับ JSON คืออะไร**. <http://www.boxsingle.com/?page=Blog.ShowBlogDetail&blogID=13>. 10 กรกฎาคม, 2557.
- [5] cs@psu. “Cache”, **หน่วยความจำแคช**. <http://staff.cs.psu.ac.th/ladda/subjects/344-521/hw2551/L1cache.pdf>. 10 กรกฎาคม, 2557.
- [6] Ionic Framework. “Ionic Framework”, **get started Ionic Framework**. <http://ionicframework.com>. 7 พฤศจิกายน, 2559.
- [7] softmelt. “HTML5”, **HTML5 คืออะไร**. <http://www.softmelt.com/article.php?id=404>. 7 พฤศจิกายน, 2559.
- [8] kanistakan. “CSS3”, **CSS คืออะไร**. <https://sites.google.com/site/ninknitkann/-css3-khux-xari>. 7 พฤศจิกายน, 2559.
- [9] Nopphanan. “AngularJS”, **AngularJS คืออะไร**. <http://programminghunter.blogspot.com/2015/10/what-is-angularjs.html>. 7 พฤศจิกายน, 2559.
- [10] อัยพรณ เอโกบล. “การทำเจชันแคชด้วยโนเอสคิวแอล”, ใน **การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศครั้งที่ 11 (NCCIT)**. น. 247-253. กรุงเทพฯ: โรงแมมโนมา, 2558.
- [11] สัณญา เครือหงษ์. “การพัฒนาแอปพลิเคชันติดตามสถานะอีเอ็มएसผ่านมือถือด้วยข้อมูลรูปแบบเจชัน”, ใน **การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศครั้งที่ 9 (NCCIT)**. น. 596-600. กรุงเทพฯ: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2556.
- [12] May Sabai Han, Sebastian Wendland and Essam Mansour. “Semantic Information Retrieval based on Wikipedia Taxonomy”, **International Journal of Computer Applications Technology and Research**. 2(1): 77-80, 2013.

เอกสารอ้างอิง (ต่อ)

- [13] Tomas Cerny. “Cooperative Web Cache”, In **the 18th International Conference on Systems, Signals and Image Processing**. p.85-88. Sarajevo: University of Sarajevo, 2011.
- [14] Nurzhan Nurseitov. “Comparison of JSON and XML Data Interchange Formats: A Case Study”, In **the 22nd International Conference on Computer Applications in Industry and Engineering**. p.157-162. California: San Francisco, 2009.
- [15] P. Ravi Kiran. “A Comparative Study on Java Mobile Clients with Data Caching in Accessing Information Systems”. **International Journal of P2P Network Trends and Technology (IJPTT)**. 8(1): 17-20; May 2014.
- [16] Hagen H'opfner. “Data Caching on Mobile Devices”, In **the 4th International Conference on Software and Data Technologies**. p.371-374. Sofia: Delft University of Technology, 2009.
- [17] Ali Raza. “Extreme Web Caching for Faster Web Browsing”, In **ACM Conference on Special Interest Group on Data Communication**. p.111-112. Brisbane: Brisbane Convention and Exhibition Centre, 2015.
- [18] สุนิสา रिमเจริญ. “Programming Fundamental 1”, **การเขียนโปรแกรมเบื้องต้น**. <http://angsila.cs.buu.ac.th/~sunisa/886201/lab01.pdf>. 5 ตุลาคม, 2560.
- [19] นภัทร รัตนาคินทร์. “การวิเคราะห์ออกแบบระบบ”, **การวิเคราะห์ระบบ**. <http://www.macare.net/analysis/index.php?id=-3>. 10 กรกฎาคม, 2557.
- [20] สุรัตน์ โคอินทรางกูร. “Entity-Relationship Diagram”, **มาตรฐานของ E-R Diagram**. <http://www.bus.tu.ac.th/usr/surat/is314/pdf/er.pdf>. 10 กรกฎาคม, 2557.
- [21] บุรินทร์ รุจจนพันธุ์. “เกณฑ์การแปลความหมาย”, **ความกว้างของอันตรภาคชั้น**. <http://www.thaiall.com/blog/burin/4967>. 10 กรกฎาคม, 2557.
- [22] Mohd. Ehmer Khan. “A Comparative Study of White Box, Black Box and Grey Box Testing Techniques”, In **the International Journal of Advanced Computer Science and Applications**. p.12-15. Tamil Nadu: Anna University, 2012.

ภาคผนวก

ภาคผนวก ก
การติดตั้งและใช้งานแอปพลิเคชัน

การติดตั้งและใช้งานแอปพลิเคชัน

1. ความต้องการของระบบ

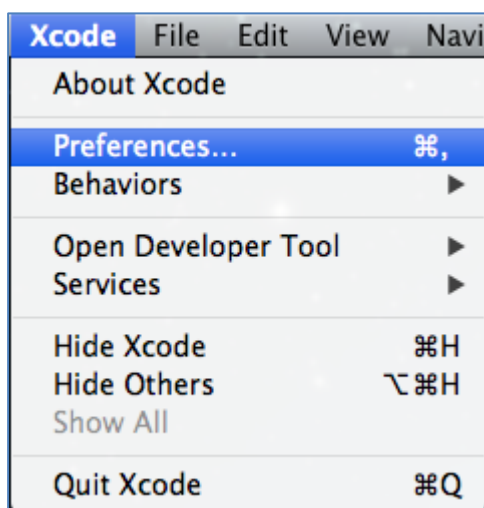
ความต้องการของระบบสำหรับการติดตั้งแอปพลิเคชันเพื่อทดสอบใช้งานบนมือถือระบบปฏิบัติการไอโอเอส ประกอบด้วย

- 1.1 โปรแกรม Xcode เวอร์ชัน 7 ขึ้นไป
- 1.2 iOS 9
- 1.3 Apple ID แบบทั่วไป หรือ แบบบัญชีนักพัฒนา

2. สร้าง Provisioning Profile

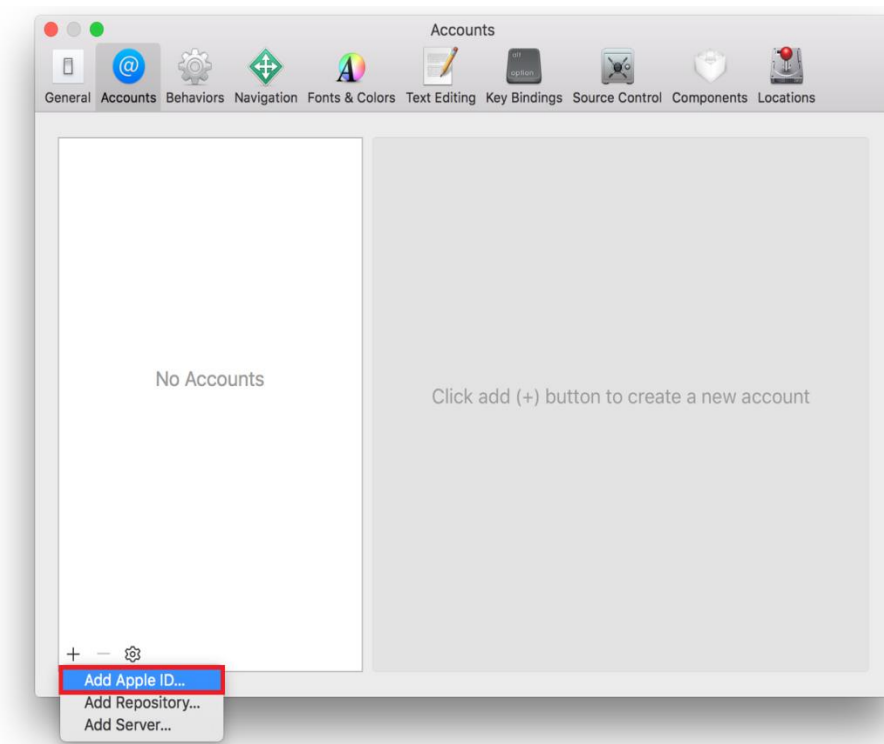
การสร้าง Provision Profile จะใช้บัญชี Apple ID ในการสร้าง ดังนี้

- 2.1 เปิดโปรแกรม Xcode แล้วเข้าไปที่เมนู Preferences



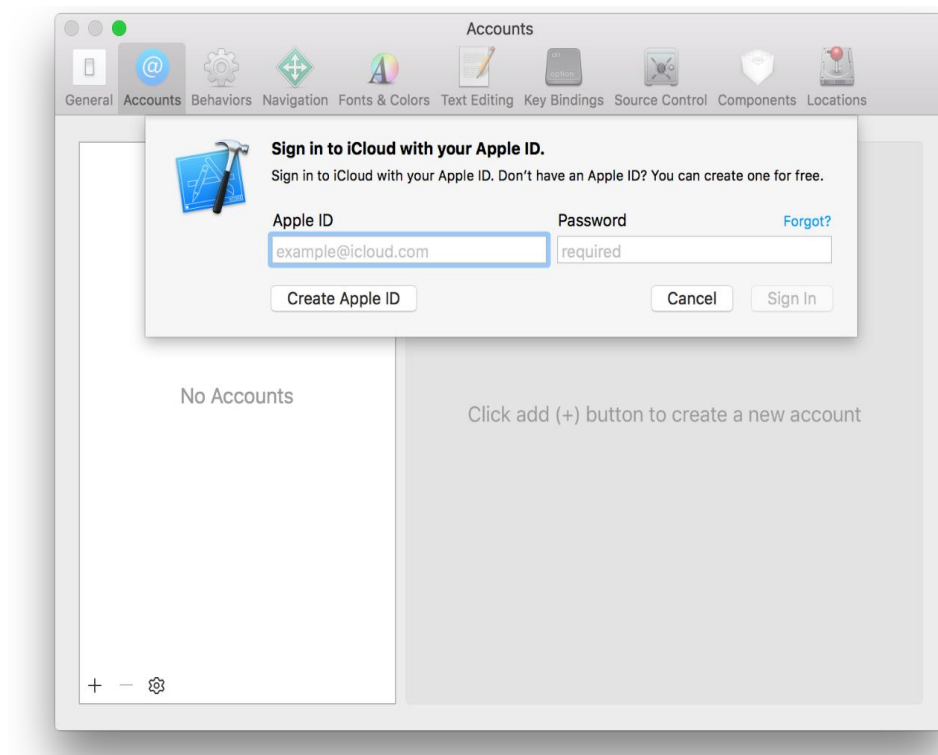
ภาพที่ ผ.1 เมนู Preferences ใน Xcode

2.2 แสดงหน้าจอไดอะล็อกใหม่ขึ้นมา ให้ทำการคลิกไปที่แท็บ Account แล้วคลิกปุ่มเครื่องหมายบวก เลือก “Add Apple ID...”



ภาพที่ ผ.2 เพิ่มบัญชี Apple ID ลงใน Xcode

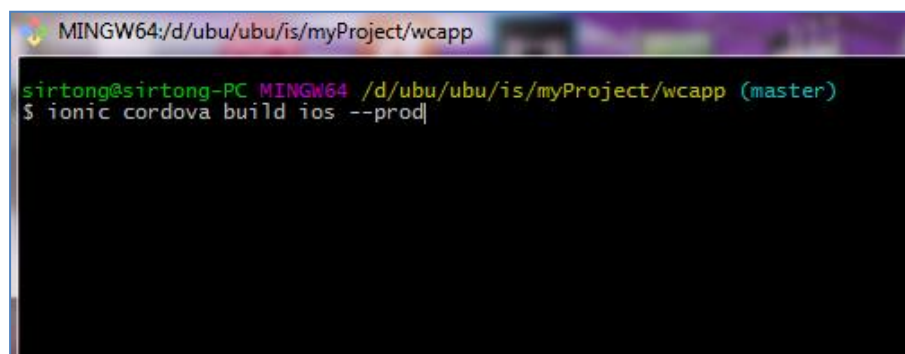
2.3 แสดงหน้าจอเด้งออกสำหรับการล็อกอินบัญชี Apple ID ให้ทำการกรอกอีเมลและรหัสผ่านให้ถูกต้อง เสร็จแล้วคลิกปุ่ม “Sign in”



ภาพที่ ผ.3 ล็อกอินบัญชี Apple ID ใน Xcode

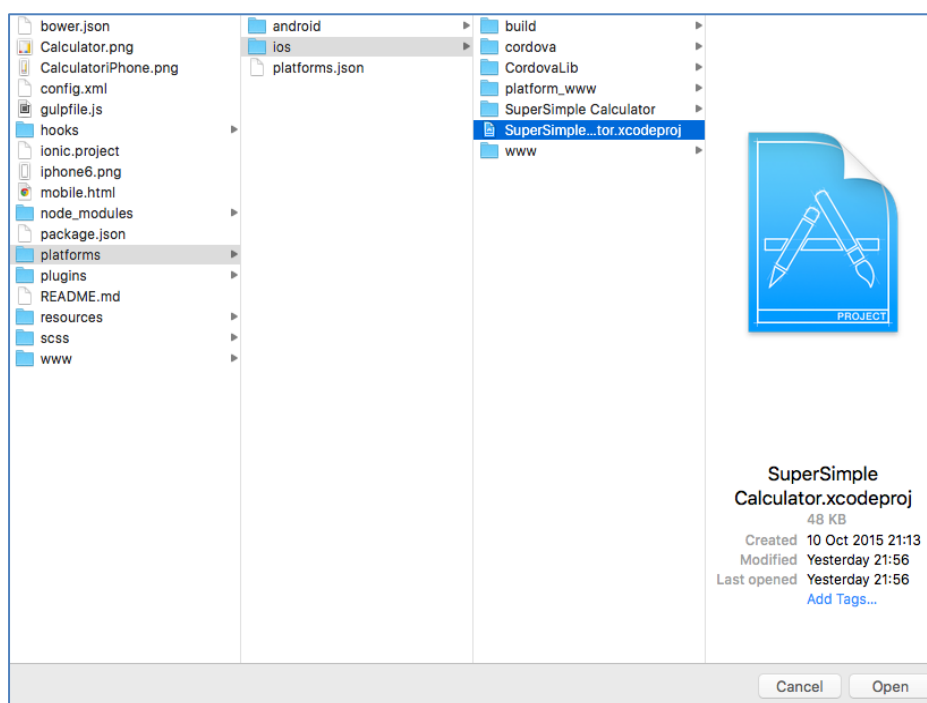
3. การรันแอปพลิเคชัน

3.1 ทำการ build โปรเจกต์โดยใช้คำสั่ง “ionic cordova build ios --prod”



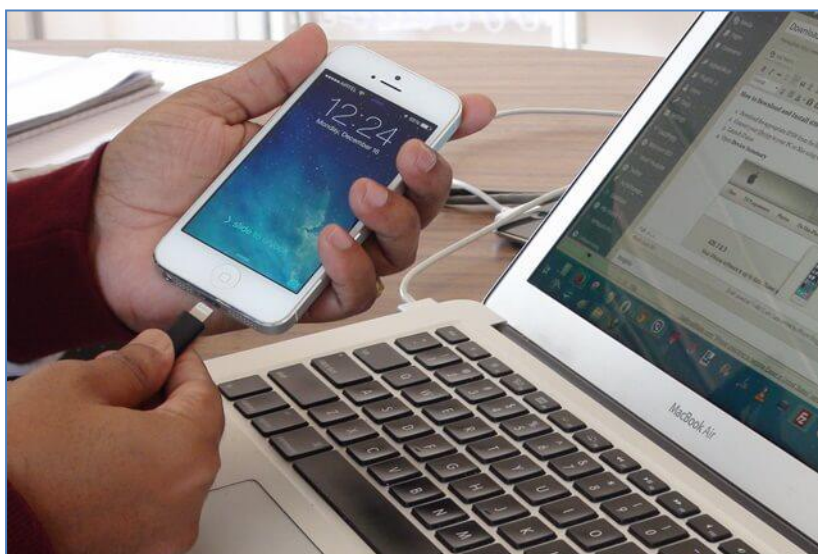
ภาพที่ ผ.4 ทำการ Build Project ในแพลตฟอร์มของ iOS

3.2 ใน Xcode ทำการเปิดไฟล์นามสกุล “.xcodeproj” ของโปรเจกที่ Build โดยจะอยู่ในโฟลเดอร์ “platforms/ios/”



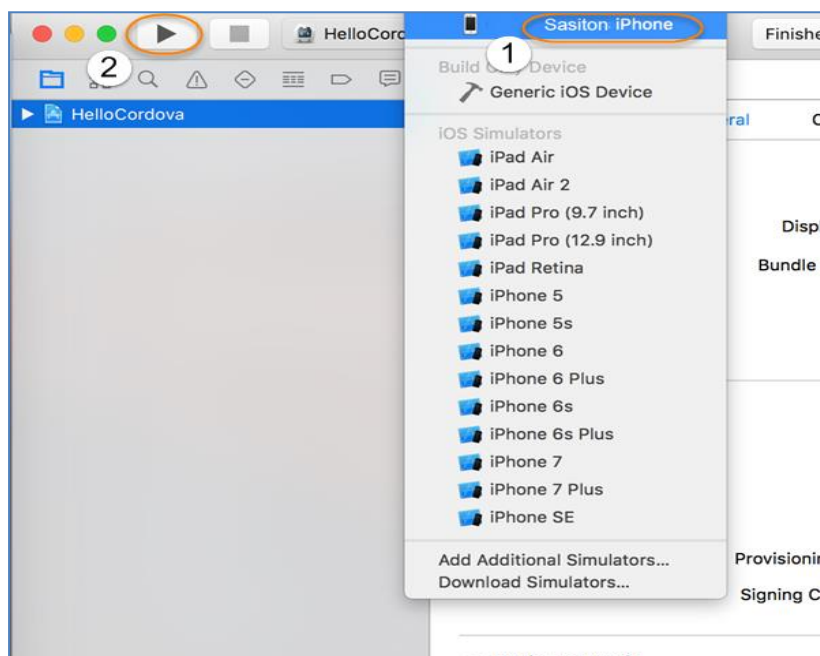
ภาพที่ ผ.5 เปิดไฟล์ .xcodeproj ใน Xcode

3.3 ทำการเชื่อมต่อโทรศัพท์มือถือไอโฟน กับ เครื่อง Mac เพื่อ Deploy แอปพลิเคชัน



ภาพที่ ผ.6 เชื่อมต่อ iPhone เข้ากับ Mac OS

3.4 ใน Xcode ทำการเลือก Device ที่จะรัน โดยคลิกเลือกชื่อ iPhone ที่เชื่อมต่อเอาไว้ จากนั้นคลิกปุ่มรัน เพื่อติดตั้งและใช้งานแอปพลิเคชันบนโทรศัพท์มือถือ



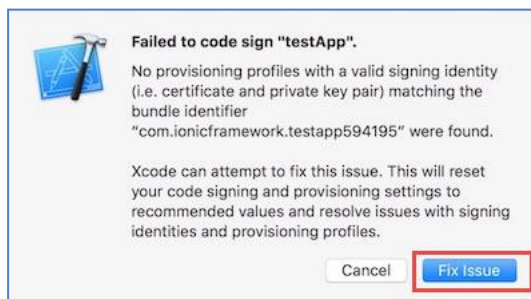
ภาพที่ ผ.7 รันแอปพลิเคชันเพื่อแสดงผลบน iPhone

4. การลงนามแอปพลิเคชัน

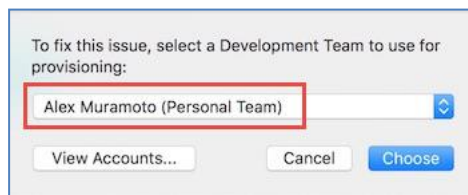
หากแอปพลิเคชันยังไม่ได้ลงนามลายเซ็นไว้ จะพบ Error หลังจากรันแอปพลิเคชัน โดยจะมีวิธีแก้ปัญหาดังนี้

4.1 สำหรับ Xcode 7

4.1.1 ระบบจะแจ้งเตือนการลงนามแอปพลิเคชันขึ้นมา ให้ทำการคลิกปุ่ม “Fix Issue” แล้วทำการเลือกโปรไฟล์เป็น Personal Team



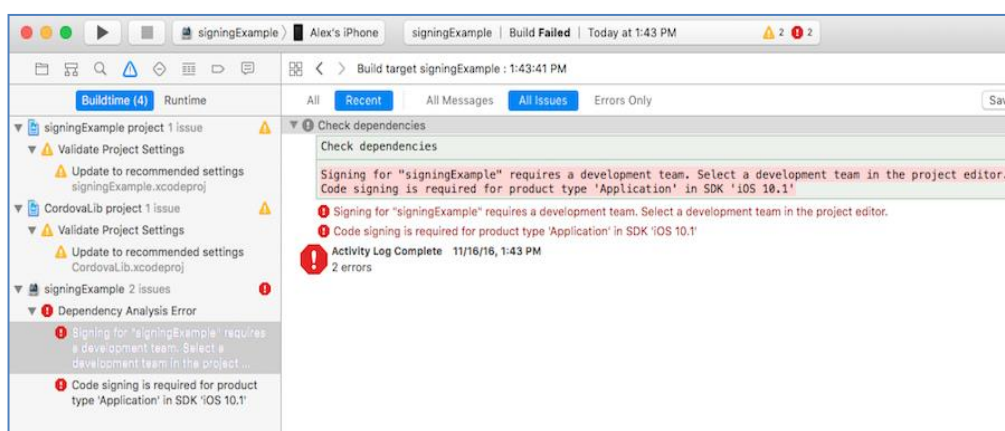
ภาพที่ ผ.8 หน้าจอแจ้งเตือนการลงนามแอปพลิเคชันบน Xcode 7



ภาพที่ ผ.9 หน้าจอเลือกโปรไฟล์เป็น Personal Team

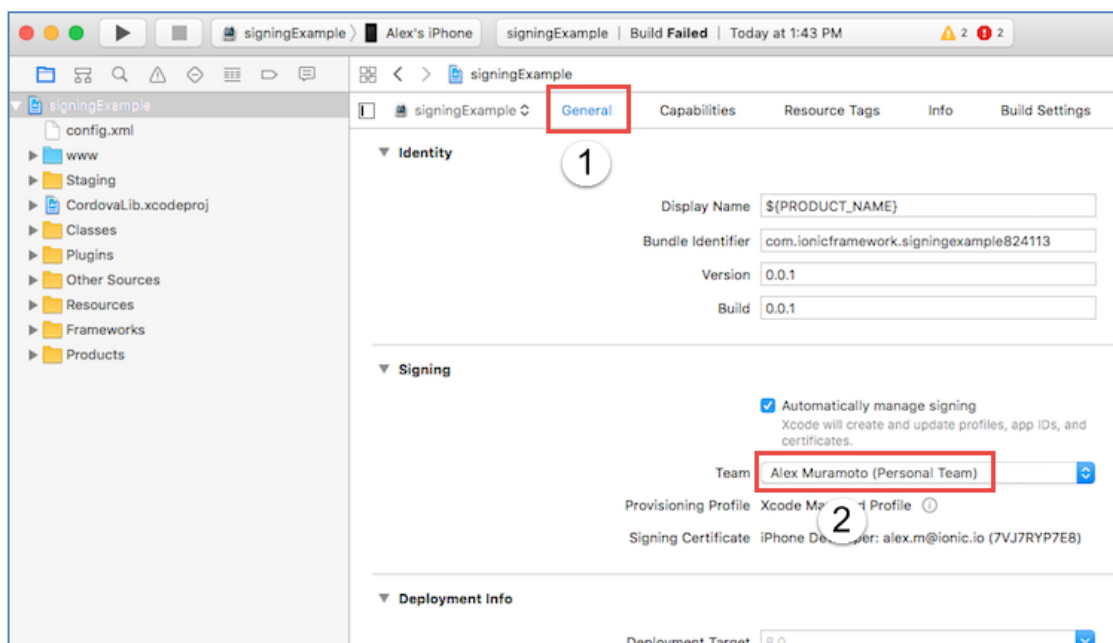
4.2 สำหรับ Xcode 8

4.2.1 ระบบจะแจ้งเตือน Error ออกทางหน้าจอดังนี้



ภาพที่ ผ.10 หน้าจอ Error การลงนามแอปพลิเคชันบน Xcode 8

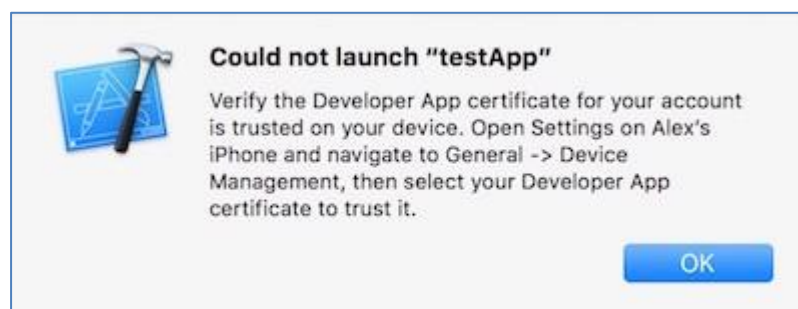
4.2.2 ทำการเลือก Certificate เพื่อลงนามแอปพลิเคชัน โดยเข้าไปที่โปรเจกต์ แล้วเลือก “General” ไปที่ Signing ในช่อง Team เลือก “ชื่อ (Personal Team)”



ภาพที่ ผ.11 หน้าจอการลงนามแอปพลิเคชันบน Xcode 8

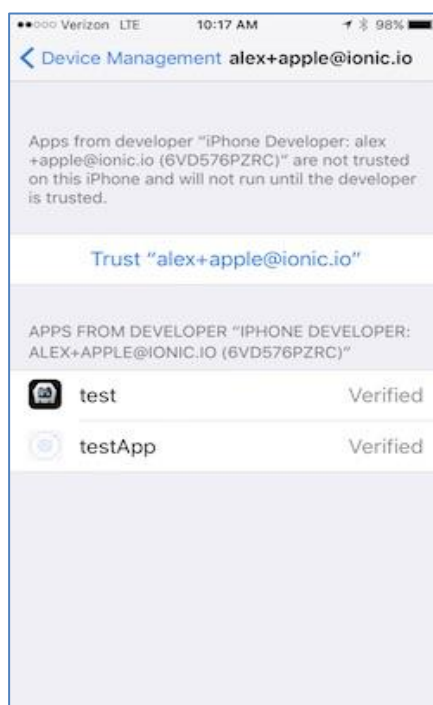
5. การรับรองความน่าเชื่อถือของ Certificate

5.1 เมื่อรันแอปพลิเคชันระบบแจ้งเตือนเกี่ยวกับความน่าเชื่อถือของ Certificate



ภาพที่ ผ.12 หน้าจอแจ้งเตือนความน่าเชื่อถือของ Certificate

5.2 เปิดโทรศัพท์มือถือไอโฟนเข้าไปที่การตั้งค่า “General > Device Management” จะพบอีเมลของ Apple ID ที่ใช้ลงนามแอปพลิเคชัน คลิก “trust” เพื่อยืนยันการติดตั้งแอปพลิเคชันลงในโทรศัพท์มือถือไอโฟน

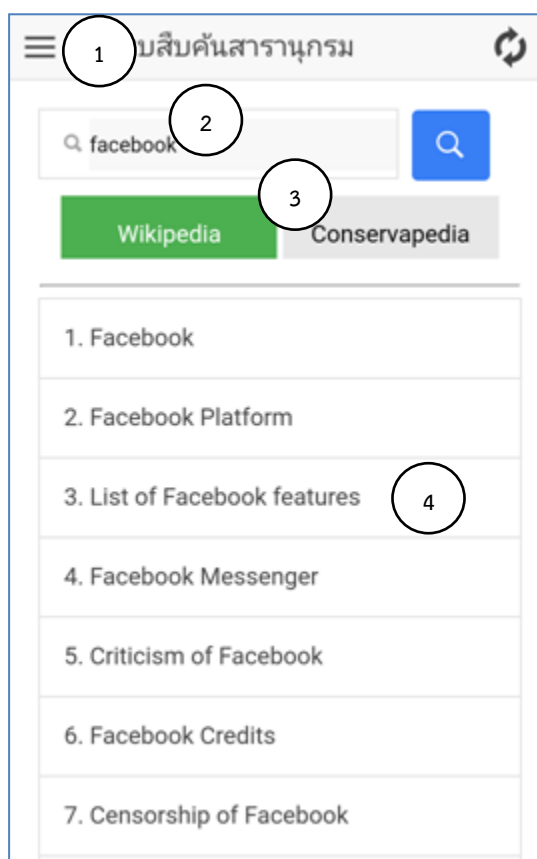


ภาพที่ ผ.13 หน้าจอการรับรองความน่าเชื่อถือของแอปพลิเคชัน

5.3 ทำการรันแอปพลิเคชัน ด้วยการคลิกปุ่ม “RUN” บน Xcode หรือใช้คำสั่ง “ionic cordova run ios –device”

6. การใช้งานแอปพลิเคชัน

6.1 การสืบค้นข้อมูลวิกิพีเดีย และ คอนเซอร์เวทีเดีย



ภาพที่ ผ.14 หน้าจอการสืบค้นข้อมูลสารานุกรมในวิกิพีเดียและคอนเซอร์เวทีเดีย

จากภาพที่ ผ.14

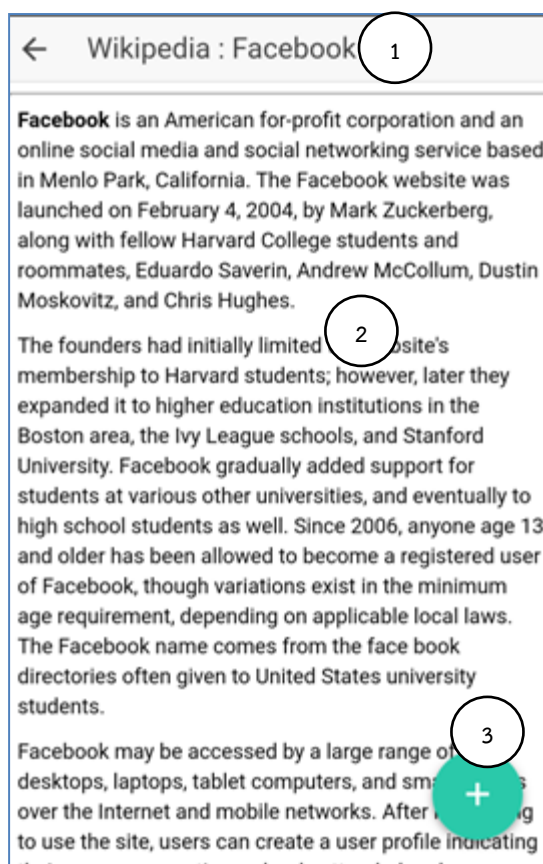
หมายเลข 1 เมนูหลัก

หมายเลข 2 ช่องกรอกคำค้น

หมายเลข 3 แถบแสดงผลการสืบค้นของวิกิพีเดียและคอนเซอร์เวทีเดีย

หมายเลข 4 ผลการสืบค้นข้อมูล

6.2 การแสดงผลหน้ารายละเอียดข้อมูล



ภาพที่ ผ.15 หน้าจอแสดงรายละเอียดข้อมูล

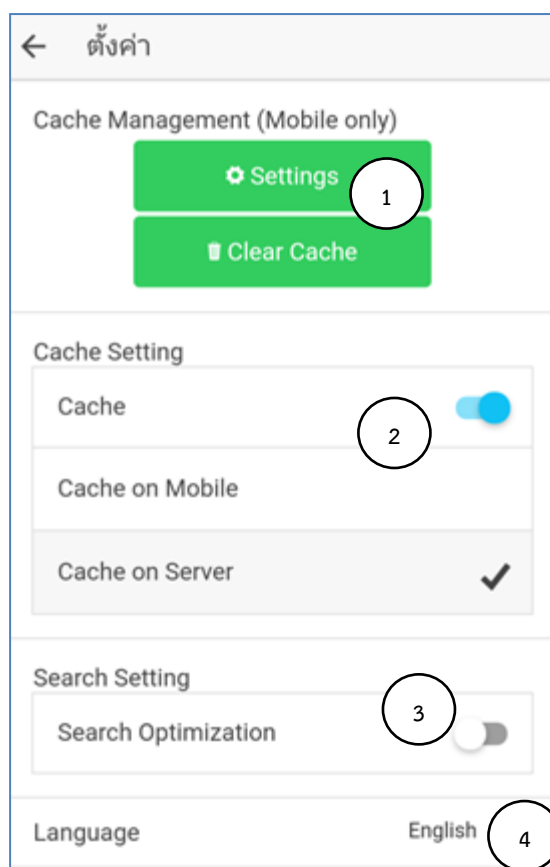
จากภาพที่ ผ.15

หมายเลข 1 ชื่อหัวเรื่อง

หมายเลข 2 ข้อมูลรายละเอียดของหัวเรื่อง

หมายเลข 3 ปุ่มที่ใช้สำหรับแชร์ข้อมูลและแคปเจอหน้าจอ

6.3 การตั้งค่าระบบสำหรับสืบค้นข้อมูล

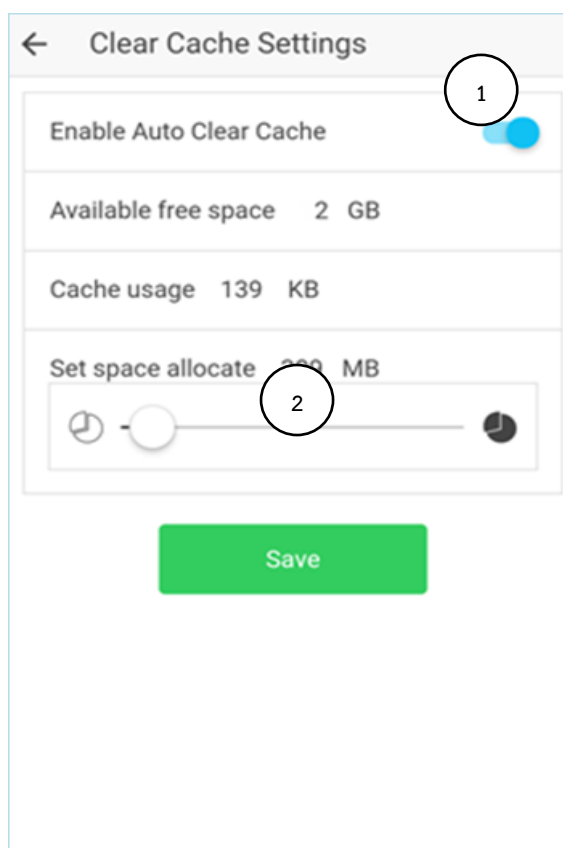


ภาพที่ ผ.16 หน้าจอการตั้งค่าระบบสำหรับสืบค้นข้อมูล

จากภาพที่ ผ.16

- หมายเลข 1 ปุ่มฟังก์ชันการตั้งค่าแคชและเคลียร์แคช
- หมายเลข 2 ตั้งค่าแคชสำหรับการสืบค้นข้อมูล
- หมายเลข 3 ตั้งค่าสืบค้นแบบเพิ่มประสิทธิภาพ
- หมายเลข 4 ตั้งค่าภาษาสำหรับการสืบค้นข้อมูล

6.4 การตั้งค่าแคชบนมือถือ



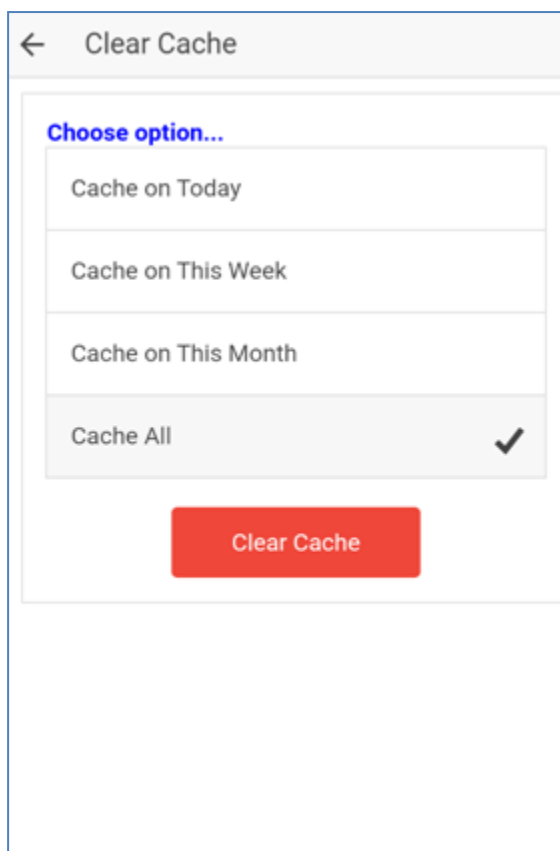
ภาพที่ ผ.17 หน้าจอการตั้งค่าแคชบนมือถือ

จากภาพที่ ผ.17

หมายเลข 1 ตั้งค่าเปิด/ปิดการเคลียร์แคชอัตโนมัติ

หมายเลข 2 กำหนดพื้นที่จัดเก็บแคชบนมือถือ

6.5 การเคลียร์แคชแบบแมนนวล



ภาพที่ ผ.18 หน้าจอการเคลียร์แคชแบบแมนนวล

จากภาพที่ ผ.18 ให้ทำการเลือกช่วงเวลาของแคชในระบบที่ต้องการเคลียร์ออกไป เสร็จแล้วคลิกปุ่ม Clear Cache

6.6 ข้อมูลประวัติการสืบค้น

← ประวัติการสืบค้น	
Found : 2 Items	Today ▾
วันที่ 2017/11/01	
wikipedia	10:53:45
liverpool	10:53:19

ภาพที่ ผ.19 หน้าจอข้อมูลประวัติการสืบค้น

จากภาพที่ ผ.19

หมายเลข 1 ไอคอนเคลียร์ข้อมูลประวัติคำค้นทั้งหมด

หมายเลข 2 รายการประวัติคำค้นในแต่ละวัน

ภาคผนวก ข
แบบสอบถามความพึงพอใจ

แบบสอบถามความพึงพอใจผู้ใช้งานทั่วไป
ในการใช้ระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทิฟ
ด้วยไอออนิกเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส

ข้อชี้แจง กรุณาทำเครื่องหมาย ✓ ในข้อที่ตรงกับความเป็นจริงและในช่องที่ตรงกับความคิดเห็นของท่านมากที่สุด

ตอนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

1. เพศ ☐ 1) ชาย ☐ 2) หญิง
2. อายุ ☐ 1) ต่ำกว่า 20 ปี ☐ 2) 21 - 30 ปี ☐ 4) 31 -40 ปี ☐ 5) 41 ปีขึ้นไป
3. ระดับการศึกษาสูงสุด

☐ 1) มัธยมศึกษาตอนต้น
☐ 3) ปริญญาตรี

☐ 2) มัธยมศึกษาตอนปลาย/ปวช./ปวส.
☐ 4) สูงกว่าปริญญาตรี
4. อาชีพ

☐ 1) เจ้าหน้าที่/พนักงานบริษัทเอกชน
☐ 3) อื่น ๆ

☐ 2) นักเรียน-นักศึกษา

ประเด็นวัดความพอใจ	ระดับความพึงพอใจ				
	5	4	3	2	1
	ดีมาก	ดี	พอใช้	น้อย	น้อยที่สุด
1. ด้านความสวยงาม					
2. ด้านความถูกต้องของข้อมูล					
3. ด้านความง่ายต่อการใช้งาน					

ประเด็นวัดความพอใจ	ระดับความพึงพอใจ				
	5	4	3	2	1
	ดีมาก	ดี	พอใช้	น้อย	น้อยที่สุด
4. ด้านความเร็วในการประมวลผล					
5. ด้านฟังก์ชันการใช้งานตรงกับความต้องการ					

ตอนที่ 3 ข้อเสนอแนะ

.....

.....

.....

.....

-ขอขอบคุณท่านเป็นอย่างสูงในการตอบแบบสอบถามนี้-

แบบสอบถามความพึงพอใจผู้ใช้เชี่ยวชาญ
ในการใช้ระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดีย
ด้วยไอโฟนิกเฟรมเวิร์ก เจสันและแคช บนระบบปฏิบัติการไอโอเอส

ข้อชี้แจง กรุณาทำเครื่องหมาย ✓ ในข้อที่ตรงกับความเป็นจริงและในช่องที่ตรงกับความคิดเห็นของท่านมากที่สุด

ตอนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

1. เพศ ☐ 1) ชาย ☐ 2) หญิง
2. อายุ ☐ 1) ต่ำกว่า 30 ปี ☐ 2) 31 - 40 ปี ☐ 4) 41 – 50 ปี ☐ 5) 51 ปีขึ้นไป
3. ระดับการศึกษาสูงสุด
☐ 1) ปริญญาตรี ☐ 2) สูงกว่าปริญญาตรี
4. อาชีพ
☐ 1) เจ้าหน้าที่/พนักงานบริษัทเอกชนไอที ☐ 2) นักศึกษา
☐ 3) นักวิชาการคอมพิวเตอร์ ☐ 4) อื่น ๆ

ประเด็นวัดความพอใจ	ระดับความพึงพอใจ				
	5	4	3	2	1
	ดีมาก	ดี	พอใช้	น้อย	น้อยที่สุด
1. ด้านความถูกต้องของข้อมูล					
2. ด้านความเร็วในการประมวลผล					

ตอนที่ 3 ข้อเสนอแนะ

.....

.....

.....

.....

-ขอขอบคุณท่านเป็นอย่างสูงในการตอบแบบสอบถามนี้-

ภาคผนวก ค
บทความวิจัย



การพัฒนาระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวเดียบนระบบปฏิบัติการไอโอเอส โดยใช้เทคนิคเจสันและแคช

ศศิธร ประสงค์^{1*} และ สมปอง เวฬุวนาธร²

^{1,2}สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี

*E-mail: sasiton.ubu@outlook.com, sompong.v@ubu.ac.th

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาและประเมินประสิทธิภาพระบบสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวเดียบนระบบปฏิบัติการไอโอเอส (iOS) ซึ่งพัฒนาโดยใช้ไอออนิกเฟรมเวิร์ก (Ionic Framework) และฐานข้อมูลเอสคิวไลต์ (SQLite) ใช้เทคนิคการทำแคชบนมือถือ (Cache Mobile) และแคชบนเซิร์ฟเวอร์ (Cache Server) เข้ามาช่วยเพิ่มประสิทธิภาพในการสืบค้น ซึ่งการทำแคชจะเก็บเป็นไฟล์รูปแบบเจสัน (JSON) เพื่อช่วยเพิ่มประสิทธิภาพความเร็วการแลกเปลี่ยนข้อมูลระหว่างไคลเอนท์ (Client) และเซิร์ฟเวอร์ (Server) โดยในการวัดประสิทธิภาพความเร็วในการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวเดียพบว่าการสืบค้นวิกิพีเดียและคอนเซอร์เวเดียผ่านแคชบนมือถือใช้เวลาตอบสนองเร็วที่สุดเท่ากับ 9 มิลลิวินาที และการสืบค้นผ่านเว็บไซต์ของคอนเซอร์เวเดียโดยตรงใช้เวลาตอบสนองช้าที่สุดเท่ากับ 825 มิลลิวินาที ในส่วนของการวัดประสิทธิภาพการแสดงผลหน้ารายละเอียดพบว่าการแสดงผลข้อมูลวิกิพีเดียผ่านแคชบนมือถือใช้เวลาตอบสนองเร็วที่สุดเท่ากับ 30 มิลลิวินาที และการแสดงผลหน้ารายละเอียดผ่านเว็บไซต์ของคอนเซอร์เวเดียโดยตรงใช้เวลาตอบสนองช้าที่สุดเท่ากับ 2207 มิลลิวินาที การประเมินความพึงพอใจของผู้ใช้ระบบจำนวน 10 คนพบว่าอยู่ในระดับดี โดยได้ค่าเฉลี่ยเท่ากับ 3.94 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.61 ดังนั้นจึงสามารถสรุปได้ว่าระบบที่พัฒนาขึ้นสามารถนำไปใช้ในการสืบค้นข้อมูลบนวิกิพีเดียและคอนเซอร์เวเดียได้จริง การนำเทคนิคเจสันและแคชช่วยให้การสืบค้นมีประสิทธิภาพมากขึ้น

คำสำคัญ: สืบค้นข้อมูล วิกิพีเดีย คอนเซอร์เวเดีย เจสัน แคช

The Development of Information Search System in Wikipedia and Conservapedia with iOS Operating System using JSON and Cache Techniques

Sasiton Prasong^{1*} and Sompong Valuvanathorn²

^{1,2}Major in Information Technology, Department of Mathematics, Statistics, and Computer, Faculty of Science, Ubonrachatani University.

*E-mail: sasiton.ubu@outlook.com, sompong.v@ubu.ac.th

Abstract

This research aims to develop and evaluate the efficiency of the information search system in Wikipedia and Conservapedia on iOS operating system. This system is developed using the Ionic Framework and SQLite database. There are caching techniques both cache on mobile and cache on server to improve search efficiency. Caching is stored as a JSON format to enhance the speed of data exchange between the client and server. The measuring a response time of search in Wikipedia and Conservapedia show that the response time of search in Wikipedia and Conservapedia via cache on mobile used least is 9



millisecond and the response time of search in Conservapedia without cache used most is 825 millisecond. In the part of measuring a response time of details page in Wikipedia and Conservapedia, the result indicated that the response time of details page in Wikipedia via cache on mobile used least is 30 millisecond and the response time of details page in Conservapedia without cache used most is 2207 millisecond. Furthermore, the five-point Likert scales is used to evaluate satisfaction from 10 users. The results reveal that is good level (the average is equal to 3.94 and S.D. is equal to 0.61). Overall, this shows that the proposed system can be used effectively.

Keywords: Information Search, Wikipedia, Conservapedia, JSON, Cache

บทนำ

ในปัจจุบันมีข้อมูลข่าวสารเกิดขึ้นทุก ๆ วัน ข้อมูลในเรื่องเดียวกันถูกสร้างและจัดเก็บไว้ในหลาย ๆ ที่ แต่ละที่จึงมีเนื้อหาที่แตกต่างกัน ทำให้การรับข้อมูลเหล่านั้นยากต่อการตรวจสอบความถูกต้องและการสืบค้นข้อมูลด้วยเครื่องเงินเป็นเพียงเครื่องมือที่ใช้สำหรับการสืบค้นข้อมูลหรือบทความบนเว็บไซต์ที่มีความสอดคล้องกับคำค้นเท่านั้น โดยไม่มีการตรวจสอบว่าบทความที่ค้นเจอนั้นมีเนื้อหาถูกต้องและมีความน่าเชื่อถือหรือไม่ จากปัญหาดังกล่าวจึงมีแนวคิดในการใช้ข้อมูลบนเว็บไซต์สารานุกรมออนไลน์มาพัฒนาบนระบบปฏิบัติการโอเอสไอโอเอส (iOS) เพื่อให้สามารถสืบค้นข้อมูลจากแหล่งที่มีความน่าเชื่อถือได้โดยตรง

วิกิพีเดีย (Wikipedia) [1][2] เป็นเว็บไซต์สารานุกรมเสรีที่ประกอบด้วยฐานข้อมูลเชิงสารานุกรมขนาดใหญ่และมีการอ้างอิงแหล่งที่มาที่มีความน่าเชื่อถือ รวมทั้งมีการปรับปรุงแก้ไขข้อมูลใหม่ด้วยความทันสมัยอยู่ตลอดเวลา วิกิพีเดียจึงเป็นเว็บไซต์ที่ได้รับการยอมรับในด้านความถูกต้องของข้อมูลและมีจำนวนข้อมูลมหาศาลที่ครอบคลุมหลายภาษาทั่วโลก ผู้วิจัยจึงได้นำวิกิพีเดียมาเป็นฐานข้อมูลที่ใช้สำหรับการสืบค้น ทั้งนี้ผู้วิจัยเห็นว่าการสืบค้นข้อมูลสารานุกรมเพียงแหล่งเดียวอาจไม่ครอบคลุมเนื้อหาและข้อเท็จจริงทั้งหมด จึงมีแนวคิดในการใช้ข้อมูลจากเว็บไซต์สารานุกรมอื่นเพิ่มเติมเพื่อช่วยเพิ่มความชัดเจนของข้อมูลที่ได้ออกมาจากการสืบค้นมากขึ้น

คอนเซอร์เวเดีย (Conservapedia) [3] เป็นเว็บไซต์สารานุกรมเสรีที่มีการจัดเก็บฐานข้อมูลเชิงสารานุกรม และสามารถแก้ไขปรับปรุงข้อมูลได้เช่นเดียวกับวิกิพีเดีย ผู้วิจัยจึงได้นำคอนเซอร์เวเดียมาใช้เป็นฐานข้อมูลในการสืบค้นอีกหนึ่งแหล่ง

อย่างไรก็ตามในการสืบค้นข้อมูลโดยตรงจากเว็บไซต์วิกิพีเดียและคอนเซอร์เวเดียนี้มีการเชื่อมต่อนิเตอร์เน็ตตลอดเวลาเนื่องจากมีการรับ-ส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ทุกครั้งที่มีการสืบค้นทำให้การแสดงผลช้าและเสียแบนด์วิดท์เป็นจำนวนมาก

เพื่อให้การสืบค้นข้อมูลมีประสิทธิภาพและเกิดความรวดเร็วในการแสดงผล ผู้วิจัยจึงได้นำเทคนิคการทำแคช (Caching) มาใช้สำหรับการจัดสรรทรัพยากรให้มีประสิทธิภาพมากขึ้น โดยลดการใช้แบนด์วิดท์ให้น้อยที่สุด นอกจากนี้ผู้วิจัยได้ใช้รูปแบบข้อมูลเจสัน (JSON) มาใช้ในการโอนถ่ายข้อมูลระหว่างไคลเอนต์กับเซิร์ฟเวอร์ ซึ่งจากการศึกษาการเปรียบเทียบประสิทธิภาพการแลกเปลี่ยนข้อมูลระหว่างเอกซ์เอ็มแอล (XML) และเจสัน (JSON) พบว่าเจสันสามารถส่งข้อมูลได้รวดเร็วกว่าและรองรับการทำงานร่วมกับภาษาโปรแกรมหลายภาษา



ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1. การสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวตฟิเคีย

เว็บไซต์วิกิพีเดียและคอนเซอร์เวตฟิเคียเป็นเว็บไซต์สารานุกรมเสรีที่พัฒนาด้วยซอฟต์แวร์โอเพนซอส (Opensource) เดียวกันมีโครงสร้างเว็บไซต์เหมือนกัน เป็นเว็บแอปพลิเคชันที่พัฒนาขึ้นด้วยภาษาพีเอชพี และ มายเอสคิวแอล โดยมีคุณสมบัติที่สนับสนุนให้นักพัฒนาสามารถเขียนโปรแกรมเพื่อเข้าถึงฐานข้อมูลสารานุกรมของเว็บไซต์ได้ โดยผ่าน Web Service API ซึ่งคอยให้บริการการเข้าถึงเมตาเดตาและฐานข้อมูลของเว็บไซต์ผ่าน HTTP Protocol โดยเมื่อไคลเอนต์ทำการร้องขอข้อมูลไปยังเซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการประมวลผลและตอบกลับข้อมูลมายังไคลเอนต์ ซึ่งรูปแบบการแลกเปลี่ยนข้อมูลมีหลายรูปแบบ เช่น เอกซ์เอ็มแอล หรือ เจสัน เป็นต้น ในการพัฒนาระบบนี้ผู้วิจัยเลือกใช้เป็นรูปแบบเจสันเพราะมีขนาดโครงสร้างข้อมูลที่เล็กกว่า

2. แคช

แคช [4] คือ ส่วนของข้อมูลที่ถูกเก็บไว้ในคอมพิวเตอร์เพื่อใช้ในการใช้งานครั้งต่อไปโดยไม่ต้องเรียกข้อมูลจากต้นแหล่งอีกครั้ง นิยมใช้ในกรณีที่เรียกข้อมูลจากต้นแหล่งได้ยาก เมื่อแคชถูกสร้างขึ้นการเรียกใช้ข้อมูลในครั้งต่อไปจะถูกอ่านข้อมูลจากแคชแทนการอ่านข้อมูลจากต้นฉบับหรือต้นแหล่งเพื่อประหยัดเวลาและค่าใช้จ่าย รวมถึงการเพิ่มความเร็วในการเรียกใช้งาน แคชนิยมใช้เมื่อรูปแบบการใช้ข้อมูลมีลักษณะที่ใกล้เคียงกันและมีการใช้ซ้ำบ่อย

3. เจสัน

เจสัน [5] คือ รูปแบบข้อมูลสายอักขระ (String) ชนิดหนึ่งที่ถูกจัดเก็บในรูปแบบที่สามารถอ่านและเข้าใจได้ง่าย เปรียบเสมือนรูปแบบของอาร์เรย์ (Array) ชนิดหนึ่งที่ใช้รับส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ โดยทั้ง 2 ฝ่ายสามารถทำการเข้ารหัสและถอดรหัสโดยใช้เจสันเอนโค้ด (JSON Encode) และ เจสันดีโค้ด (JSON Decode) เพื่ออ่านค่าตัวแปรเหล่านั้น

4. ระบบปฏิบัติการไอโอเอส

ไอโอเอส [6] คือ ระบบปฏิบัติการบนอุปกรณ์พกพา เช่น iPhone iPad และ iPod Touch พัฒนาและจัดจำหน่ายโดยบริษัทแอปเปิล ไอโอเอสแตกต่างจากระบบปฏิบัติการวินโดวส์ของไมโครซอฟท์และแอนดรอยด์ของกูเกิลตรงที่แอปเปิลไม่อนุญาตให้นำไอโอเอสไปติดตั้งบนอุปกรณ์ที่ไม่ใช่อุปกรณ์ของแอปเปิล ไอโอเอสมีส่วนติดต่อประสานกับผู้ใช้ (User Interface) ที่มีพื้นฐานแนวคิดมาจากการควบคุมโดยตรง (Direct Manipulation) ด้วยการใช้มัลติทัช องค์ประกอบของการควบคุม คือ การใช้นิ้วเลื่อน สวิทช์ และปุ่ม เพื่อเป็นการควบคุมอุปกรณ์

5. งานวิจัยที่เกี่ยวข้อง

Nurzhan Nurseitov [7] ได้ทำการศึกษาการเปรียบเทียบประสิทธิภาพระหว่างเอกซ์เอ็มแอลกับเจสัน ซึ่งเป็นรูปแบบข้อมูลที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างกัน พบว่าการใช้รูปแบบเจสันมีขนาดโครงสร้างข้อมูลที่เล็กจึงทำให้สามารถส่งข้อมูลได้รวดเร็วกว่ารูปแบบเอกซ์เอ็มแอล อัยพรพรณ เอโกบล [8] ได้ทำการศึกษาการทำเจสันแคชด้วยโนเอสคิวแอล พบว่าเทคนิคการแคชข้อมูลเจสันในรูปแบบแฟกไฟล์ (Text File) สามารถทำให้ระบบมีประสิทธิภาพทางเวลาดีขึ้นและการจัดเก็บข้อมูลด้วยโนเอสคิวแอลมีประสิทธิภาพดีกว่าฐานข้อมูลที่ใช้ภาษาเอสคิวแอล (SQL) เช่น มายเอสคิวแอล ดังนั้นการนำเจสันแคช และ โนเอสคิวแอลมาประยุกต์ใช้ร่วมกันทำให้ระบบทั้งหมดมีประสิทธิภาพดียิ่งขึ้น Tomas Cerny [9] ได้ทำการศึกษาการทำงานร่วมกันของเว็บแคชในเครือข่าย ซึ่งเป็นแนวคิดที่ใช้เว็บแคชหลาย ๆ เว็บแคชมาทำงานร่วมกันให้เกิดประสิทธิภาพ โดยเฉพาะการนำไปใช้กับเว็บเซิร์ฟเวอร์ที่มีการร้องขอข้อมูลซ้ำ ๆ กันของไคลเอนต์ โดยงานวิจัยนี้พบว่าการออกแบบและนำไปใช้กับเครือข่ายออนไลน์แบบ P2P ได้ผลลัพธ์ที่ดี P. Ravi Kiran [10] ได้ทำการศึกษาการเปรียบเทียบข้อแตกต่างการเข้าถึงข้อมูลเซิร์ฟเวอร์โดยใช้อุปกรณ์มือถือและการทำแคชข้อมูลบนมือถือ ซึ่งเป็นการเปรียบเทียบระหว่างการเข้าถึงแบบระบบตัวกลางคอยบริการข้อมูลหรือเว็บเซิร์ฟเวอร์กับแบบที่ไม่มีระบบตัวกลางคอยให้บริการข้อมูลหรือเป็นการใช้ J2ME Driver ในการเชื่อมต่อกับฐานข้อมูลของเซิร์ฟเวอร์โดยตรง ผลที่ได้จากการทดลองการเข้าถึงแบบไม่มีตัวกลางมีความยุ่งยากและข้อจำกัดมากกว่า

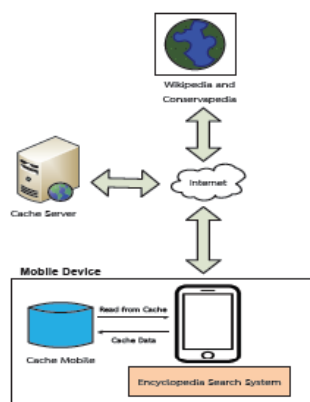


วิธีการวิจัย

งานวิจัยนี้ ผู้วิจัยได้นำหลักการวิเคราะห์และออกแบบระบบตามหลักวิศวกรรมซอฟต์แวร์ โดยใช้การเขียนแผนผังการไหลของกระแสข้อมูล (Data Flow Diagram) และการออกแบบฐานข้อมูลโดยใช้โมเดลความสัมพันธ์ระหว่างข้อมูล (E-R Diagram) การพัฒนาระบบใช้รูปแบบโมบายแอปพลิเคชัน (Mobile Application) ด้วยไอออนิกเฟรมเวิร์ค (Ionic Framework) [11] กับ ฐานข้อมูลเอสคิวไลต์ (SQLite) [12] และ ในรูปแบบเว็บแอปพลิเคชัน (Web-Based Application) พัฒนาด้วยภาษาโปรแกรมพีเอชพี (PHP) [13] กับ ฐานข้อมูลมายเอสคิวแอล (MySQL) [14] โดยในส่วนของการแลกเปลี่ยนข้อมูลระหว่างไคลเอนต์ และ เซิร์ฟเวอร์ จะใช้รูปแบบเจสันในการส่ง-รับข้อมูล

1. ภาพรวมของระบบ

สถาปัตยกรรมซอฟต์แวร์ของงานวิจัยนี้มีองค์ประกอบสำคัญ 3 ส่วน คือ 1) แอปพลิเคชันบนมือถือ เป็นส่วนของหน้าจอหลักที่ใช้ในการสืบค้นข้อมูล แสดงผลลัพธ์ และจัดการแคชบนมือถือ 2) แคลเซิร์ฟเวอร์ เป็นส่วนที่ใช้สำหรับจัดเก็บแคชไว้บนเซิร์ฟเวอร์ในกรณีที่มีการตั้งค่าการสืบค้นโดยผ่านแคชเซิร์ฟเวอร์ 3) เว็บไซต์สารานุกรม ประกอบด้วยเว็บไซต์วิกิพีเดียและเว็บไซต์คอนเซอร์เวทีฟเดีย ซึ่งเป็นเว็บไซต์สารานุกรมที่ผู้วิจัยจะนำมาใช้เป็นแหล่งข้อมูลในการสืบค้น

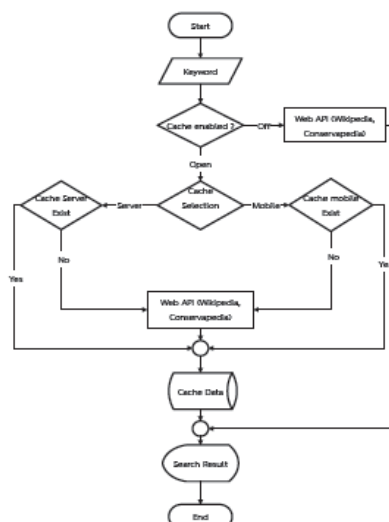


รูปที่ 1 ภาพรวมของระบบ

จากภาพที่ 1 ในการสืบค้นข้อมูลจะใช้วิธีการตั้งค่าการสืบค้น 2 แบบ คือ 1) แบบไม่ผ่านแคช จะสืบค้นจากเว็บไซต์วิกิพีเดียและคอนเซอร์เวทีฟเดียโดยตรง 2) แบบผ่านแคช จะให้เลือกว่าต้องการสืบค้นผ่านแคชมือถือหรือแคชเซิร์ฟเวอร์ โดยให้เลือกรายอย่างใดอย่างหนึ่ง

2. การออกแบบและพัฒนาโปรแกรม

ในงานวิจัยนี้ ใช้การออกแบบและพัฒนาระบบตามกระบวนการทำงาน (Flow Chart) แสดงได้ดังภาพที่.2

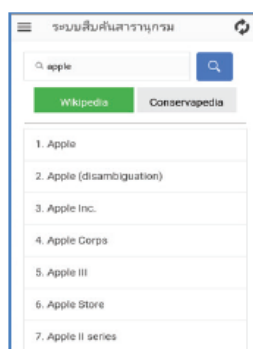


รูปที่ 2 แผนภาพแสดงกระบวนการทำงาน

จากรูปที่ 2 เมื่อผู้ใช้ทำการสืบค้นข้อมูลโดยการกรอกคำค้นส่งเข้าไปในระบบ ระบบจะทำการตรวจสอบว่ามีเปิดใช้งานแคชหรือไม่ ถ้าไม่เปิดแคชจะทำการสืบค้นจากเว็บไซต์วิกิพีเดียและคอนเซอร์เวเดียโดยตรง แต่ถ้าเปิดแคชระบบจะทำการตรวจสอบว่ามีคำค้นเป็นแคชบนมือถือ หรือ แคชบนเซิร์ฟเวอร์แล้วทำการเช็คคำค้นนี้มีอยู่ในแคชหรือไม่ ถ้ามีจะนำผลการสืบค้นจากแคชมาแสดง แต่ถ้าไม่มีในแคชระบบจะทำการสืบค้นจากเว็บไซต์วิกิพีเดียและคอนเซอร์เวเดียแล้วนำผลลัพธ์ที่ได้มาแสดง3พร้อมกับทำการแคชเก็บไว้ในระบบ

ผลการวิจัย

จากผลการวิจัยและพัฒนาแอปพลิเคชันทำให้สามารถนำมาใช้เป็นต้นแบบในการทดสอบและวัดประสิทธิภาพ ดังแสดงในภาพที่3 ถึงภาพที่7



ภาพที่ 3 ภาพหน้าจอการสืบค้นข้อมูล

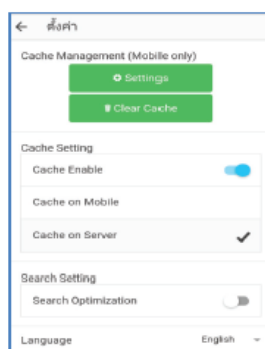


จากรูปที่ 3 แสดงการสืบค้นข้อมูลโดยใช้คำค้น “apple” และแสดงผลลัพธ์คำที่ค้นเจอ



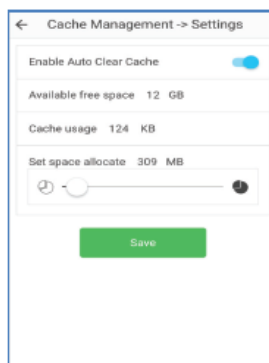
ภาพที่ 4 ภาพหน้าจอรายละเอียดข้อมูล

จากรูปที่ 4 แสดงหน้าจอรายละเอียดของคำค้นหรือหัวเรื่อง “Apple Inc.” ที่เลือกจากภาพที่ 3



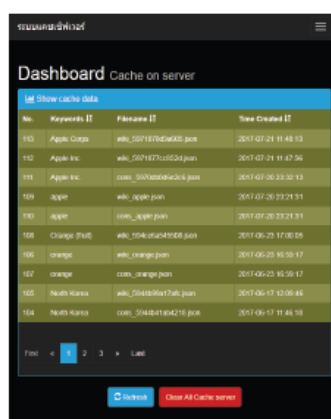
รูปที่ 5 ภาพหน้าจอการตั้งค่าระบบสืบค้น

จากรูปที่ 5 เป็นหน้าจอของการตั้งค่าการสืบค้น ประกอบด้วย 4 ส่วน คือ 1) การจัดการแคชบนมือถือ ซึ่งจะมีปุ่มตั้งค่าและเคลียร์แคช 2) การตั้งค่าแคช เป็นการตั้งค่าการเปิด/ปิดแคช ถ้าเปิดใช้งานแคชจะมีให้เลือกว่าจะใช้แคชบนมือถือหรือแคชบนเซิร์ฟเวอร์ 3) การตั้งค่าเพิ่มประสิทธิภาพการสืบค้น เป็นวิธีการนำข้อมูลไปโฟกัสของผู้ใช้มาเข้ากระบวนการปรับปรุงคิวรีเพื่อช่วยให้การสืบค้นมีประสิทธิภาพมากขึ้น 4) ภาษา เป็นการเลือกภาษาที่ใช้ในการสืบค้น



ภาพที่ 6 รูปหน้าจอการจัดการแคชบนมือถือ

รูปที่ 6 เป็นหน้าจอของการตั้งค่าแคชบนมือถือ โดยจะบอกพื้นที่ว่าง พื้นที่ข้อมูลแคช และ การตั้งค่าขนาดของพื้นที่สำหรับเก็บแคช โดยการเคลียร์แคชบนมือถือจะทำการเคลียร์อัตโนมัติในลักษณะเข้าก่อนออกก่อน (First-In-First-Out) เมื่อข้อมูลเต็มพื้นที่ขนาดที่ใช้เก็บแคช



รูปที่ 7 ภาพหน้าจอแสดงแคชที่จัดเก็บบนเซิร์ฟเวอร์

จากรูปที่ 7 เป็นการแสดงรายละเอียดข้อมูลแคชที่อยู่บนเซิร์ฟเวอร์ กรณีที่ตั้งค่าสับค้นผ่านแคชเซิร์ฟเวอร์ระบบจะทำการค้นจากแคชเซิร์ฟเวอร์

1. การเปรียบเทียบประสิทธิภาพเวลาการตอบสนอง

งานวิจัยนี้ได้ทดสอบเปรียบเทียบประสิทธิภาพความเร็วของการสืบค้นข้อมูลทั้ง 3 แบบ คือ แบบไม่มีแคช แบบมีแคชบนมือถือ และแบบมีแคชบนเซิร์ฟเวอร์ ซึ่งผู้วิจัยได้ทำการทดลองดังนี้ 1) ทาค่าเฉลี่ยเวลาการตอบสนองของการ



สืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดีย 2) หาค่าเฉลี่ยเวลาการตอบสนองของการแสดงผลหน้ารายละเอียดของวิกิพีเดียและคอนเซอร์เวทีฟเดีย

1.1d หาค่าเฉลี่ยเวลาการตอบสนองของการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดีย

ในการทดลองนี้ทำการทดสอบโดยใช้คำค้นที่เป็นภาษาอังกฤษ 10 คำค้น แต่ละคำค้นทำการคลิกค้น 3 ครั้ง แต่ละครั้งทำการเก็บค่าเวลาตอบสนอง (Response Time) เพื่อนำมาคำนวณหาค่าเฉลี่ย (Average) ของความเร็วในการตอบสนองแต่ละแบบ คำที่ได้มีหน่วยเป็นมิลลิวินาที (Millisecond) โดยการทดสอบนี้ใช้เครื่อง iPhone4s และอินเทอร์เน็ตความเร็ว 512 Kbps แสดงได้ดังตารางที่ 1

ตารางที่ 1 ตารางแสดงค่าเฉลี่ยเวลาการตอบสนองในการสืบค้นข้อมูลวิกิพีเดียและคอนเซอร์เวทีฟเดีย

	วิกิพีเดีย (มิลลิวินาที)	คอนเซอร์เวทีฟเดีย (มิลลิวินาที)
แคชมือถือ	9	9
แคชเซิร์ฟเวอร์	104	108
ไม่ใช่แคช	812	825

จากตารางที่ 1 ผลการสืบค้นข้อมูลผ่านแคชบนมือถือใช้เวลาการตอบสนองน้อยที่สุดเท่ากับ 9 มิลลิวินาที และการสืบค้นข้อมูลแบบไม่ใช่แคชในคอนเซอร์เวทีฟเดียใช้เวลานานที่สุดเท่ากับ 825 มิลลิวินาที

1.2 หาค่าเฉลี่ยเวลาการตอบสนองของการแสดงผลหน้ารายละเอียดของวิกิพีเดียและคอนเซอร์เวทีฟเดีย

ในการทดลองนี้ทำการทดสอบโดยการเลือกหัวเรื่องมา 10 หัวเรื่อง คลิกเข้าไปหน้ารายละเอียดในแต่ละหัวเรื่อง 3 ครั้ง แต่ละครั้งทำการเก็บค่าเวลาการตอบสนอง แล้วนำมาคำนวณหาค่าเฉลี่ยของความเร็วในการตอบสนองแต่ละแบบ คำที่ได้มีหน่วยเป็นมิลลิวินาที โดยการทดสอบนี้ใช้เครื่อง iPhone4s และอินเทอร์เน็ตความเร็ว 512 Kbps แสดงได้ดังตารางที่ 2

ตารางที่ 2 ตารางแสดงค่าเฉลี่ยเวลาการตอบสนองในการแสดงผลข้อมูลวิกิพีเดียและคอนเซอร์เวทีฟเดีย

	วิกิพีเดีย (มิลลิวินาที)	คอนเซอร์เวทีฟเดีย (มิลลิวินาที)
แคชมือถือ	30	34
แคชเซิร์ฟเวอร์	188	204
ไม่ใช่แคช	1019	2207

จากตารางที่ 2 ผลการแสดงผลหน้ารายละเอียดข้อมูลของวิกิพีเดียผ่านแคชมือถือใช้เวลาการตอบสนองน้อยที่สุดเท่ากับ 30 มิลลิวินาที และผลการแสดงผลหน้ารายละเอียดข้อมูลของคอนเซอร์เวทีฟเดียโดยไม่ใช่แคช ใช้เวลาการตอบสนองมากที่สุดเท่ากับ 2207 มิลลิวินาที

2. การประเมินความพึงพอใจต่อการใช้งานระบบ

ผู้วิจัยได้ออกแบบการประเมินผลวัดความพึงพอใจต่อการใช้งานแอปพลิเคชัน โดยแบ่งออกเป็น 5 ด้าน คือ ด้านความสวยงาม ด้านความถูกต้องของข้อมูล ด้านความง่ายต่อการใช้งาน ด้านความเร็วในการประมวลผล และ ด้านฟังก์ชันการใช้งานตรงกับความต้องการ



ในส่วนของการระดับคะแนนการประเมินนั้น แบ่งเป็น 5 ระดับคือ ดีมาก 5 คะแนน ดี 4 คะแนน ปานกลาง 3 คะแนน พอใช้ 2 คะแนน และควรปรับปรุง 1 คะแนน โดยในการประเมินความพึงพอใจนี้จะใช้กลุ่มตัวอย่างจากผู้ใช้งานแอปพลิเคชันจำนวน 10 คน ในการตอบแบบสอบถามหลังจากที่ใช้งานแอปพลิเคชัน ซึ่งผลได้ที่แสดงดังตารางที่ 3

ตารางที่ 3 ระดับความพึงพอใจต่อการใช้งานแอปพลิเคชัน

รายการประเมิน	ระดับความพึงพอใจ		
	$\bar{x}$	S.D.	คุณภาพ
1.ด้านความสวยงาม	3.70	0.67	ดี
2.ด้านความถูกต้องของข้อมูล	3.80	0.63	ดี
3.ด้านความง่ายต่อการใช้งาน	4.20	0.63	ดี
4.ด้านความเร็วในการประมวลผล	4.10	0.56	ดี
5.ด้านฟังก์ชันการใช้งานตรงกับความต้องการ	3.90	0.56	ดี
สรุปผล	3.94	0.61	ดี

สรุป

1. สรุปผลการวิจัย

การทำวิจัยในครั้งนี้ผู้วิจัยได้คำนึงถึงประโยชน์ที่ได้จากสืบค้นข้อมูลจากเว็บไซต์สารานุกรมวิกิพีเดียและคอนเซอร์เวทีฟเดีย ซึ่งถือเป็นแหล่งสารสนเทศที่สำคัญ จึงพัฒนาแอปพลิเคชันนี้ขึ้นมาเพื่อเป็นเครื่องมือในการสืบค้นและอำนวยความสะดวกในการสืบค้นข้อมูลสารานุกรมให้มีประสิทธิภาพมากขึ้น

จากการทดสอบวัดประสิทธิภาพความเร็วเวลาในการตอบสนองการสืบค้นข้อมูลในวิกิพีเดียและคอนเซอร์เวทีฟเดีย พบว่าการสืบค้นผ่านแคชบนมือถือใช้เวลาการตอบสนองน้อยที่สุดเท่ากับ 9 มิลลิวินาที ในขณะที่การสืบค้นข้อมูลในคอนเซอร์เวทีฟเดียแบบไม่ใช้แคชนั้นใช้เวลาการตอบสนองมากที่สุดเท่ากับ 825 มิลลิวินาที ในส่วนของการแสดงผลหน้ารายการละเอียดพบว่าการแสดงผลหน้ารายการละเอียดในวิกิพีเดียผ่านแคชบนมือถือใช้เวลาตอบสนองน้อยที่สุดเท่ากับ 30 มิลลิวินาที ในขณะที่การแสดงผลหน้ารายการละเอียดในคอนเซอร์เวทีฟเดียแบบไม่ใช้แคชนั้นใช้เวลาการตอบสนองมากที่สุดเท่ากับ 2207 มิลลิวินาที ดังนั้นจึงสรุปได้ว่าการสืบค้นผ่านแคชช่วยให้การสืบค้นข้อมูลมีความรวดเร็วมากขึ้นและช่วยลดแบนด์วิดท์ของการใช้งานอินเทอร์เน็ตอีกด้วย

การประเมินความพึงพอใจต่อการใช้งานระบบในแต่ละด้านรวมกันมีค่าเฉลี่ยเท่ากับ 3.94 ค่าความเบี่ยงเบนมาตรฐานเท่ากับ 0.61 ซึ่งถือว่าอยู่ในระดับดี จึงสรุปได้ว่าระบบที่พัฒนาขึ้นสามารถนำไปใช้ในการสืบค้นข้อมูลบนวิกิพีเดียและคอนเซอร์เวทีฟเดียได้จริง

2. ข้อเสนอแนะ

งานวิจัยนี้ ผู้วิจัยมีข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาและการประยุกต์ใช้ดังต่อไปนี้

2.1 การสืบค้นข้อมูลสารานุกรมใช้ฐานข้อมูลจาก 2 แหล่งข้อมูล คือ วิกิพีเดียและคอนเซอร์เวทีฟเดีย ซึ่งถือว่าแหล่งข้อมูลที่มีอยู่ยังมีน้อย ระบบควรพัฒนาให้สามารถสืบค้นข้อมูลจากเว็บไซต์สารานุกรมอื่น ๆ ได้ด้วย

2.2 การสืบค้นแบบเพิ่มประสิทธิภาพใช้ข้อมูลโปรไฟล์มาใช้ในการปรับปรุงคิวรีในการสืบค้น วิธีที่ใช้อยู่ยังแสดงผลลัพท์ออกมาได้ไม่เต็มที่เท่าที่ควร ซึ่งจุดนี้ควรออกแบบอัลกอริธึมใหม่ โดยการกำหนดค่าน้ำหนักของข้อมูลโปรไฟล์ให้เหมาะสมและปรับปรุงคิวรีให้มีประสิทธิภาพเพื่อให้การสืบค้นได้ผลลัพธ์ที่ถูกต้องและแม่นยำมากขึ้น



เอกสารอ้างอิง

- [1] วิกิพีเดีย[ออนไลน์], <http://th.wikipedia.org/wiki/วิกิพีเดีย/>. (วันที่สืบค้น:10กรกฎาคม2557).
- [2] Adam R. Brown “Wikipedia as a Data Source for PoliticalScientists: Accuracy and Completeness of Coverage” *Brigham Young University*, 2011.
- [3] คอนเซอร์เวเดีย [ออนไลน์], http://www.conservapedia.com/Main_Page (วันที่สืบค้น:7พฤศจิกายน2559).
- [4] แคช[ออนไลน์],<https://th.wikipedia.org/wiki/แคช/>. (วันที่สืบค้น:10กรกฎาคม2557).
- [5] เจสัน[ออนไลน์],<https://th.wikipedia.org/wiki/เจสัน/>. (วันที่สืบค้น:10กรกฎาคม2557).
- [6] ไอโอเอส[ออนไลน์], <https://th.wikipedia.org/wiki/ไอโอเอส/>. (วันที่สืบค้น:10กรกฎาคม2557).
- [7] Nurzhan Nurseitov “Comparison of JSON and XML Data Interchange Formats: A Case Study” *Department of Computer Science. Montana StateUniversity*, 2010.
- [8] อัยพรณ เอโกบล “การทำเจสันแคชด้วยโนเอสคิวแอล” ภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์ ,2558
- [9] Tomas Cerny “Cooperative Web Cache” *Faculty of Electrical Engineering, Czech Technical University*, 2011.
- [10] P. Ravi Kiran “A Comparative Study on Java Mobile Clients with Data Caching in Accessing Information Systems” *International Journal of P2P Network Trends and Technology (IJPTT)*, Volume 8, May 2014.
- [11] ไอโอนิคเฟรมเวิร์ค [ออนไลน์], <http://ionicframework.com/>. (วันที่สืบค้น: 7 พฤศจิกายน, 2559).
- [12] เอสคิวไลท์ [ออนไลน์], <https://www.sqlite.org/>. (วันที่สืบค้น: 7 พฤศจิกายน, 2559).
- [13] พีเอชพี [ออนไลน์], <http://php.net/>. (วันที่สืบค้น: 10 กรกฎาคม, 2557).
- [14] มายเอสคิวแอล [ออนไลน์], <https://www.mysql.com/>. (วันที่สืบค้น: 10 กรกฎาคม,2557).

ประวัติผู้วิจัย

ชื่อ นามสกุล	นายศศิธร ประสงค์
ประวัติการศึกษา	เทคโนโลยีบัณฑิต สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยบูรพา พ.ศ. 2546
ประวัติการทำงาน	พ.ศ. 2546–2550 (Web Programmer) บริษัท Inpros Markting system Co., Ltd. พ.ศ. 2550–2551 (Application Developer) บริษัท Optimussoft Co., Ltd. พ.ศ. 2551–2553 (Software Engineer) บริษัท FDS Networks (Thailand) Co., Ltd.
ปัจจุบัน	Freelance Programmer and Internet Marketing