

# M2: DESIGN PRESENTATION & MODELING

## DIAGRAMS

---

กลุ่ม SevenYod

รายชื่อสมาชิก

1. นายกฤตนนท์	วิเศษชาติ	รหัสนักศึกษา 56010020
2. นายปารเมศ	แดงพรวน	รหัสนักศึกษา 56010751
3. นายศรีพงศ์	อริยะเดช	รหัสนักศึกษา 56011191
4. นายศุภวิชญ์	คงสวัสดิ์	รหัสนักศึกษา 56011248
5. นายศุภวิชญ์	บุญหะกิจ	รหัสนักศึกษา 56011250
6. นายอชิยุต	รัชตเจริญชัยกุล	รหัสนักศึกษา 56011397
7. นายอัศธรณพัชรฐ์	ตั้งกิจศิริ	รหัสนักศึกษา 56011439

## ANALYSIS

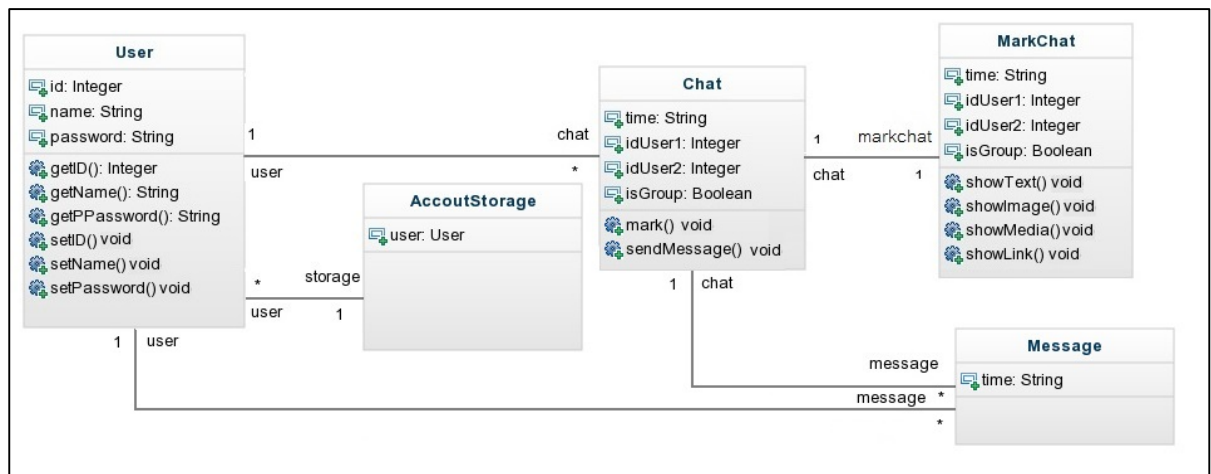
---

### ○ Use cases

- Login: ผู้ใช้เข้าหน้าจอล็อกอิน ผู้ใช้กรอกUsernameและPassword จากนั้นกดปุ่มLogin ระบบตรวจสอบความถูกต้องจากข้อมูลที่ใช้กรอกกับฐานข้อมูล ระบบให้ผู้ใช้เข้าสู่หน้าจอหลักของโปรแกรม
- Register: ผู้ใช้เข้าหน้าลงทะเบียน ผู้ใช้กดปุ่มยืนยันข้อตกลงของระบบ ระบบให้ผู้ใช้เข้าสู่หน้ากรอกข้อมูลสำหรับลงทะเบียน ผู้ใช้กรอกข้อมูลทั้งหมดในหน้าลงทะเบียน ผู้ใช้กดปุ่มRegister ระบบตรวจสอบความถูกต้องของข้อมูลที่ใช้กรอกจากนั้นเก็บลงฐานข้อมูล ระบบแจ้งสถานะความสำเร็จในการลงทะเบียนไปยังอีเมลล์ที่ผู้ใช้ได้กรอกในขั้นตอนกรอกข้อมูล ระบบให้ผู้ใช้เข้าหน้าจอหลักของโปรแกรม
- Forgot password: ผู้ใช้กดปุ่มลืมรหัสผ่านที่หน้าจอล็อกอิน ระบบให้ผู้ใช้เข้าสู่หน้าจอลืมรหัสผ่าน ผู้ใช้กรอกอีเมลล์ที่ใช้ในการสมัครสมาชิก ระบบเช็คข้อมูลจากอีเมลล์ที่ผู้ใช้กรอก ระบบแจ้งข้อมูล User nameและPasswordไปทางอีเมลล์ของผู้ใช้ ผู้ใช้เข้าอีเมลล์ตรวจสอบรายละเอียดที่ใช้ล็อกอิน
- Send & Receive message: ผู้ใช้เลือกผู้ใช้เป้าหมายที่ต้องการจะสนทนา ผู้ใช้กรอกข้อความหรือเลือกไฟล์ที่ต้องการจะส่งจากนั้นกด ระบบแสดงข้อความหรือไฟล์ที่ผู้ใช้กรอกไปยังหน้าจอสนทนา
- Marking system: ผู้ใช้กดค้างบนข้อความ ไฮเปอร์เท็กซ์ รูป หรือ ไฟล์ จากนั้นระบบจะทำการมาร์คหรือยกเลิกการมาร์คให้ผู้ใช้ระบบทำการมาร์คอัตโนมัติสำหรับไฮเปอร์เท็กซ์ รูป และ ไฟล์
- Notification system: ระบบทำการแจ้งเตือนข้อความเข้าโดยการแสดงสัญลักษณ์ที่ปุ่ม
- เมื่อด้านซ้ายบน , เมนู People และ รายชื่อเพื่อน มีเสียงแจ้งเตือน และแจ้งเตือนผ่านNotification bar
- Search: ผู้ใช้กดปุ่มเพื่อทำการค้นหาโดยผู้ใช้จะกรอกชื่อผู้ใช้คนอื่น ข้อความ วันที่ ชื่อลิงค์ หรือ ชื่อไฟล์ต่างๆที่ต้องการค้นหาจากนั้นระบบจะทำการค้นหาและแสดงผลลัพธ์ ผู้ใช้จะเห็นผลลัพธ์ของการค้นหาปรากฏขึ้นที่หน้าจอ
- Profile Management: ผู้ใช้กดที่ปุ่มบริเวณมุมซ้ายบนของหน้าแอปพลิเคชัน ผู้ใช้เลือกแท็บโปรไฟล์ ผู้ใช้กดปุ่มแก้ไขจากนั้นผู้ใช้แก้ไขข้อมูลต่างๆแล้วกดปุ่มเซฟ ระบบจะทำการบันทึกข้อมูลใหม่ที่ผู้ใช้กรอก ผู้ใช้จะเห็นข้อมูลโดยรวมของตัวเอง
- System setting: ผู้ใช้กดปุ่มบริเวณมุมซ้ายบนของหน้าแอปพลิเคชัน ผู้ใช้เลือกแท็บตั้งค่าจากนั้นผู้ใช้เลือกการตั้งค่าต่างๆของแอปพลิเคชัน

- Add friends: ผู้ใช้กดปุ่มเพิ่มเพื่อน ผู้ใช้กรอกUsernameที่ต้องการจะเพิ่มเพื่อนจากนั้นกดปุ่มค้นหา ระบบตรวจสอบเช็คหาUsernameที่ผู้ใช้กรอกจากนั้นแสดงรายชื่อในหน้าจอเพิ่มเพื่อน ผู้ใช้เลือกเพื่อนจากนั้นกดปุ่มเพิ่ม ระบบนำรายชื่อที่ผู้ใช้เพิ่มใส่เพิ่มลงในรายชื่อเพื่อนของผู้ใช้
- User group management: ผู้ใช้กดปุ่มด้านขวาบนเพื่อจัดการสมาชิกในกลุ่ม เช่น เชิญ และ ลบสมาชิกในกลุ่มสนทนา
- Filter marked system: ผู้ใช้กดปุ่มเมนูบริเวณด้านบนของหน้าจอ เพื่อเลือก Content ที่ต้องการจะแสดง

## ○ Domain Model



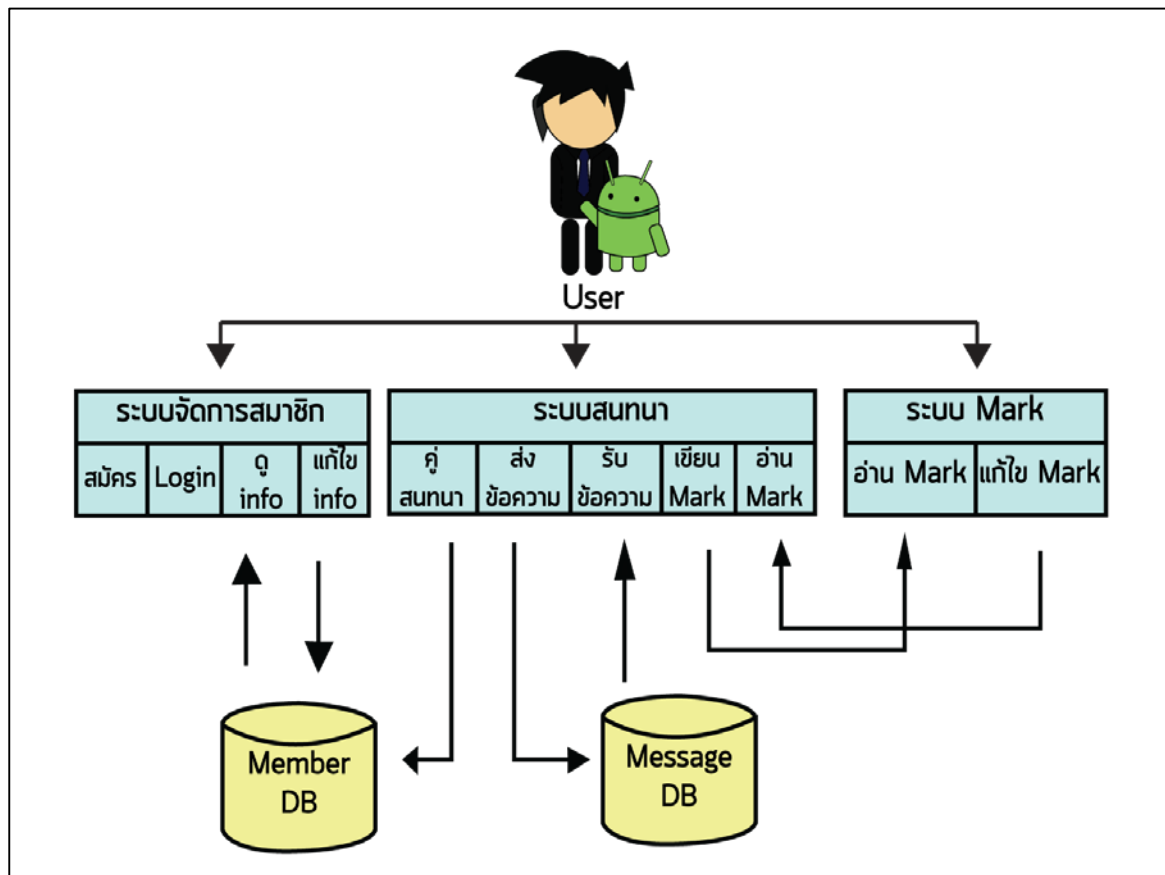
- User มีความสัมพันธ์ Many to 1 กับ Account storage เพราะ ข้อมูลของ ทุกๆUser ถูกบันทึกที่ AccountStorage
- User มีความสัมพันธ์ 1 to Many กับ Chat เพราะ User สามารถ Chat ได้มากกว่า 1 dialog
- User มีความสัมพันธ์ 1 to Many กับ Message เพราะ User สามารถส่ง message ได้มากกว่า 1 message
- Chat มีความสัมพันธ์ 1 to 1 กับ MarkChat เพราะ MarkChat เก็บ Content ที่ถูก Mark ไว้ของ แต่ละ Chat
- Chat มีความสัมพันธ์ 1 to Many กับ Message เพราะ Chat เก็บ Message ทั้งหมดไว้

## ○ Abstraction

- User(ผู้ใช้) ซึ่งประกอบไปด้วย ชื่อผู้ใช้ รหัสผ่าน และ หมายเลขไอดีประจำตัว
- AccountStorage(แฟ้มเก็บข้อมูลผู้ใช้)ซึ่งประกอบไปด้วย ข้อมูลทุกอย่างของทุก User
- Chat (กระดานสนทนา) คือบันทึกการสนทนาและเวลา ของ Message
- Message (บทสนทนา) ประกอบไปด้วย ข้อความธรรมดา รูปภาพ สื่อมัลติมีเดีย Hyperlink
- MarkChat (บทสนทนาที่ถูกปักหมุด)ประกอบไปด้วย บันทึกการสนทนาและเวลา ใน Chat ซึ่งผู้ใช้เลือกเองว่าควรที่จะเก็บมาอยู่ใน MarkChat หรือไม่

## DESIGN

### – Application Architecture



- ระบบจัดการสมาชิก ประกอบไปด้วยระบบการสมัคร การlogin ระบบดูและแก้ไขข้อมูลส่วนตัวโดยระบบนี้จะเชื่อมต่อกับฐานข้อมูลสมาชิก
- ระบบสนทนา ประกอบไปด้วยระบบอ่านประวัติสนทนา ระบบรับ-ส่งข้อความและระบบMarkบทสนทนาที่สำคัญโดยระบบนี้จะเชื่อมต่อกับฐานข้อมูลสมาชิกและฐานข้อมูลประวัติการสนทนา
- ระบบMark ประกอบไปด้วยระบบอ่านบทสนทนาที่ถูกMark และระบบแก้ไขบทสนทนาที่ถูก Mark โดยระบบนี้ติดต่อกับระบบสนทนา

## – Subsystem / Components

### – ระบบจัดการสมาชิก

- ระบบการสมัครสมาชิก: มีหน้าที่เพื่อให้ผู้ใช้กรอกข้อมูลรายละเอียดต่างๆที่ใช้ในการสมัคร แล้วระบบจะนำข้อมูลจัดเก็บลงฐานข้อมูล
- ระบบLogin: มีหน้าที่นำข้อมูลของ username และ password ที่ผู้ใช้กรอกไปเพื่อตรวจสอบว่ามีข้อมูลในฐานข้อมูลหรือไม่
- ระบบดูข้อมูลส่วนตัว: มีหน้าที่เรียกข้อมูลต่างๆของผู้ใช้จากฐานข้อมูล
- ระบบแก้ไขข้อมูลส่วนตัว: มีหน้าที่เรียกข้อมูลของผู้ใช้ออกมาจากฐานข้อมูลแล้วนำข้อมูลที่ผู้ใช้กรอกบันทึกลงฐานข้อมูล

### – ระบบสนทนา

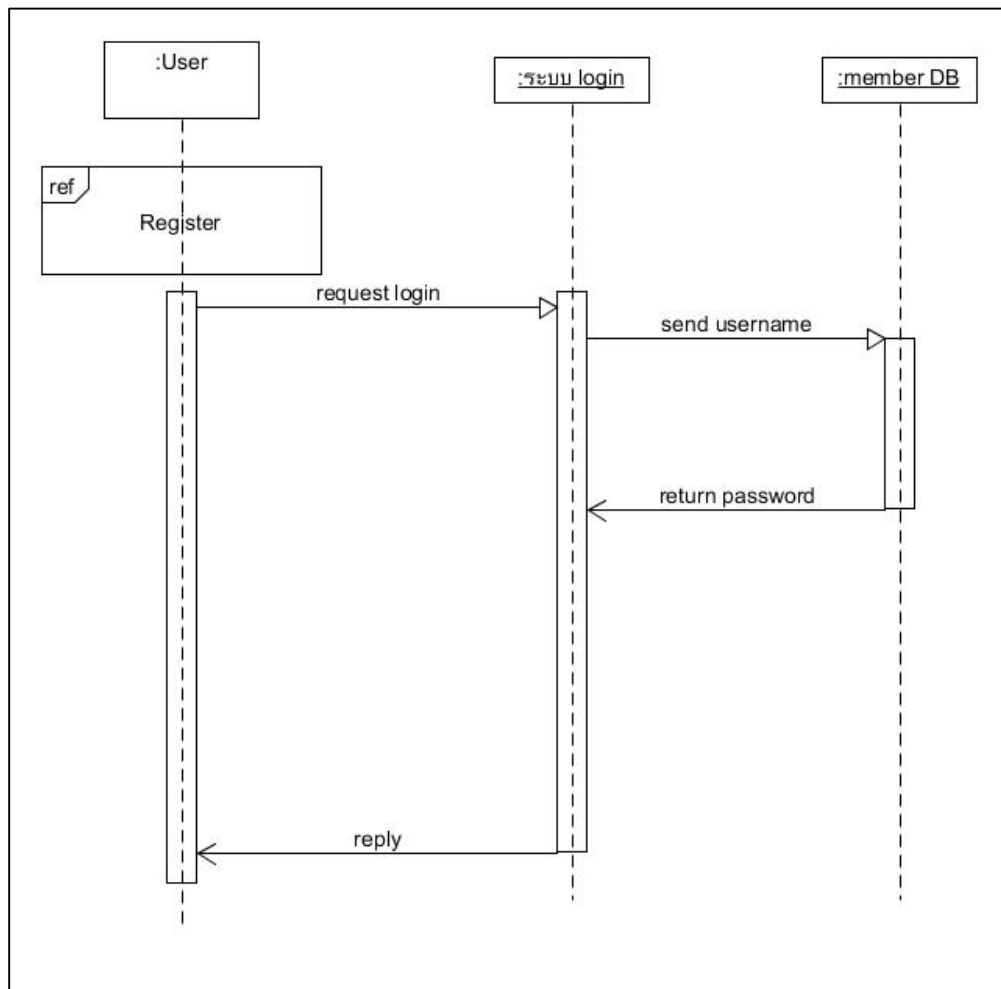
- ระบบอ่านประวัติสนทนา: มีหน้าที่เรียกข้อมูลของข้อความที่ผู้ใช้รับ - ส่งจากฐานข้อมูล
- ระบบส่งข้อความ: มีหน้าที่ส่งข้อความที่ผู้ใช้ต้องการไปยังผู้ใช้ที่ต้องการส่งข้อความนั้น
- ระบบรับข้อความ: มีหน้าที่รับข้อความที่ผู้ใช้ได้รับมาจากผู้อื่น
- ระบบเขียนMark: มีหน้าที่Markข้อความ รูป และ ไฟล์ทั้งที่ผู้ใช้Markเอง และ ระบบMarkอัตโนมัติ
- ระบบอ่านMark: มีหน้าที่แสดงผลข้อความ รูป และ ไฟล์ที่ถูกMarkไว้

### – ระบบMark

- ระบบอ่านMark: มีหน้าที่แสดงผลข้อความ รูป และ ไฟล์ที่ถูกMarkไว้
- แก้ไขmark: มีหน้าที่ลบข้อความ รูป และ ไฟล์

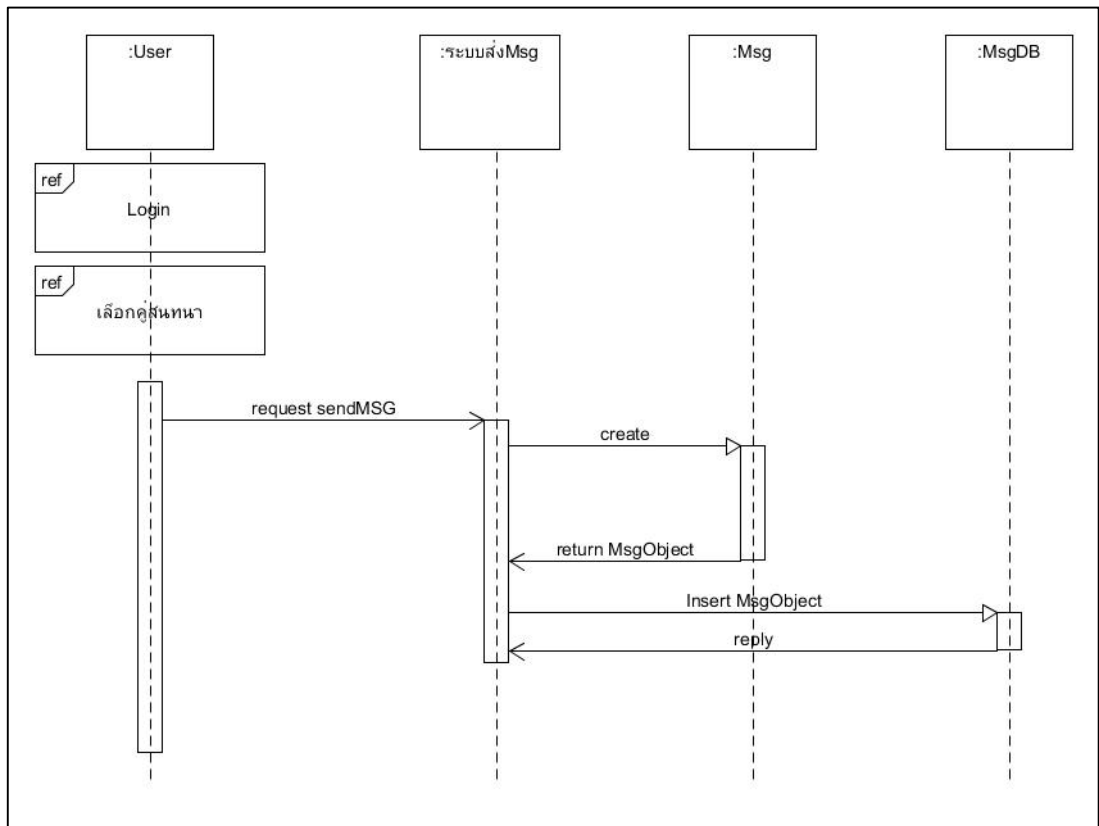
## Sequence diagram

### ระบบLogin



จากรูปเป็น Sequence diagram ของ Subsystem ส่วนของระบบLogin โดยผู้ใช้งานจะต้องสมัครสมาชิกก่อนเข้าใช้งาน จากนั้นกรอก username password และกดปุ่มLogin เพื่อส่งRequestไปที่ระบบ Login จากนั้นระบบLoginจะส่งRequestขอ password โดยการส่ง username ที่ผู้ใช้งานกรอกเข้ามา ระบบ member DB จะReturn password ของ username ที่ถูก Request กลับไป ระบบLoginจะทำการตรวจสอบความถูกต้องระหว่าง password ที่ผู้ใช้งานกรอกเข้ามากับส่วนที่ระบบmember DB ตอบกลับมว่า ตรงกันหรือไม่ จากนั้นระบบLoginก็จะตอบผลการ Login ให้กับผู้ใช้งานว่าถูกต้องหรือไม่

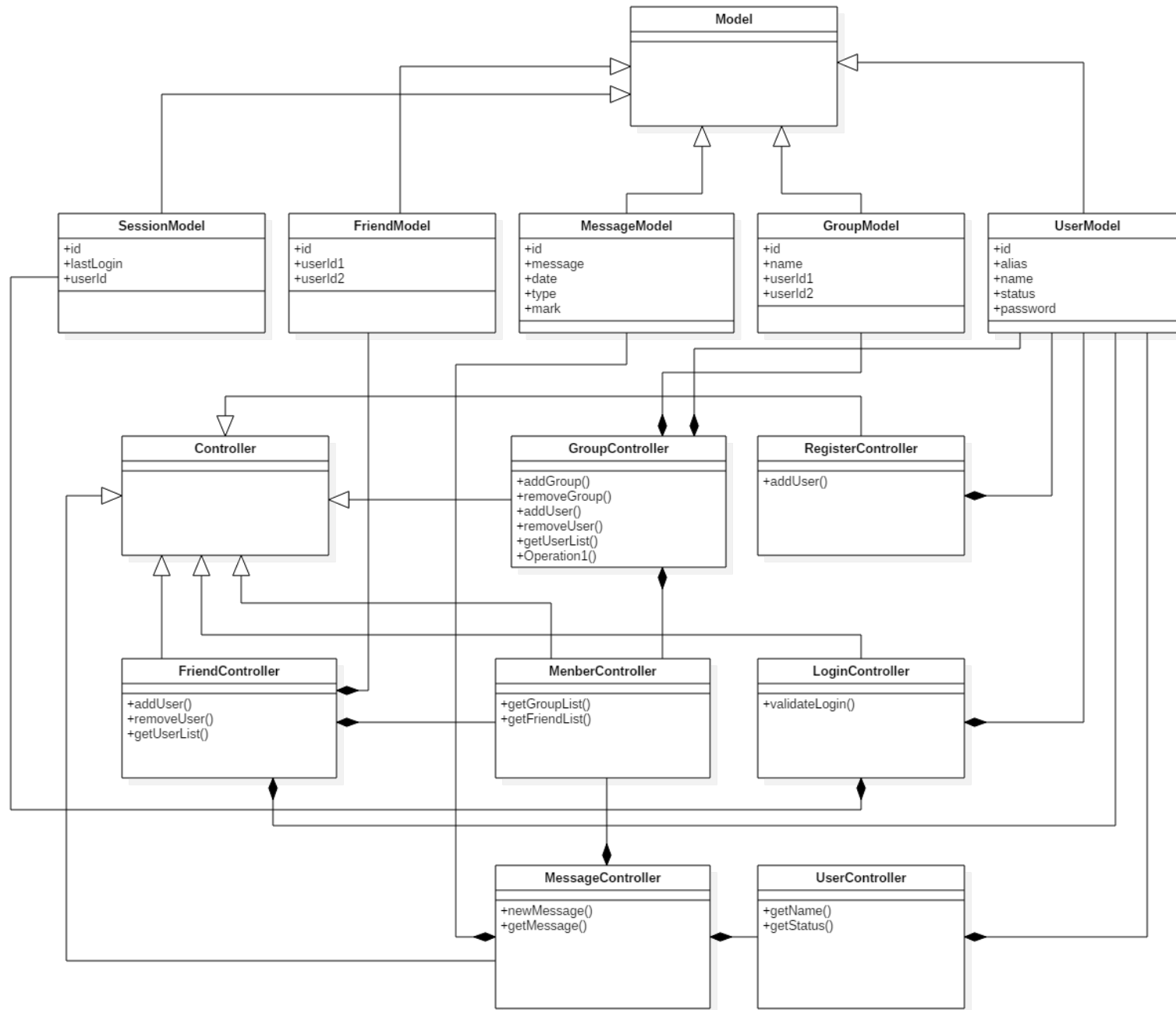
## ระบบส่งข้อความ



จากรูปเป็น Sequence diagram ของ Subsystem ส่วนของระบบส่งข้อความ โดยผู้ใช้งานจะต้องผ่านการ Login และการเลือกคู่สนทนา จากนั้นผู้ใช้งานจะ Request พร้อมกับข้อความที่ต้องส่งไปยังระบบส่งข้อความ จากนั้นระบบส่งข้อความจะสร้าง object ของ message จากนั้น message จะ return object ของ message กลับมา ระบบสนทนาจะทำการบันทึก object ของ message ลงไป message DB จากนั้น message DB จะตอบผลการบันทึกข้อมูลกลับมา



- Design diagram

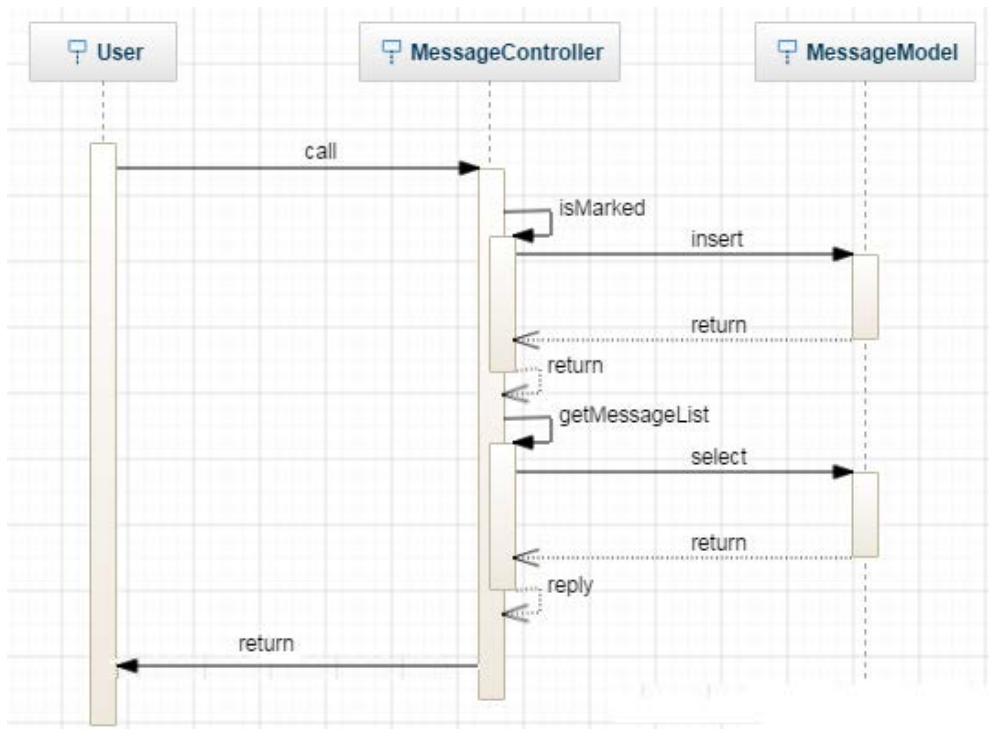


จาก Design diagram ด้านบนจะอธิบายได้ว่า

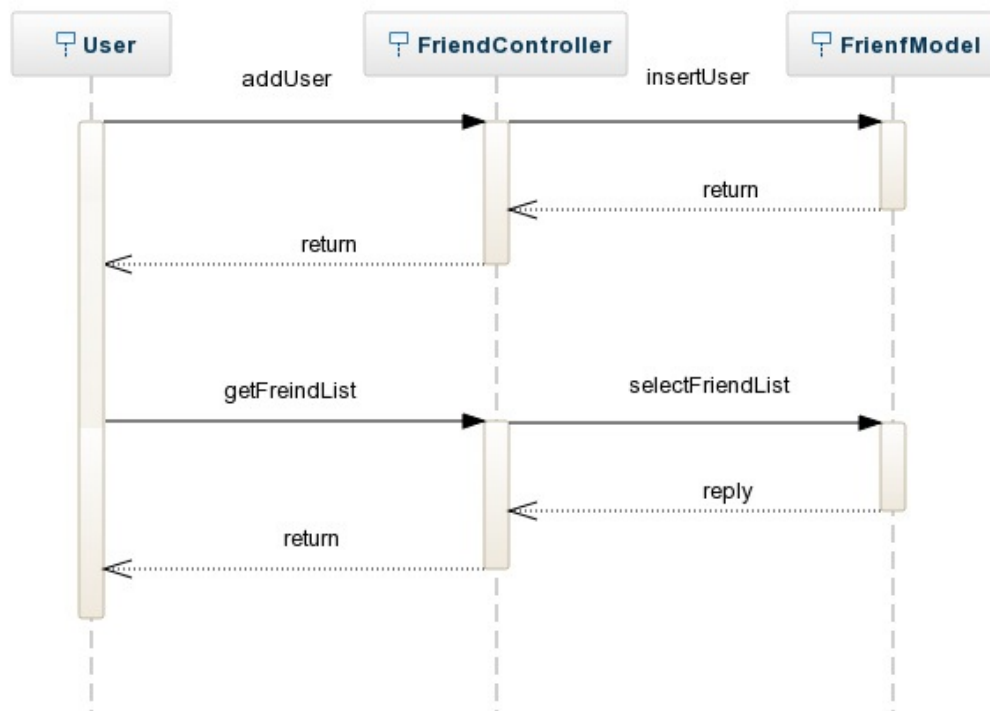
- Model: ใช้ในการติดต่อ CRUD ใน Database
- SessionModel: ใช้ในการติดต่อ Database เกี่ยวกับการเก็บ log ของการ login
- FriendModel: ใช้ในการติดต่อ Database เกี่ยวกับการสร้างเพื่อนของ user แต่ละ user ที่เป็นเพื่อนกัน
- GroupModel: ใช้ในการติดต่อ Database เกี่ยวกับการสร้างกลุ่มของ user แต่ละ user ที่อยู่ในกลุ่ม
- MessageModel: ใช้ในการติดต่อ Database เกี่ยวกับการเก็บข้อมูลข้อความของ user แต่ละ user
- UserModel: ใช้ในการติดต่อ Database เกี่ยวกับข้อมูลของแต่ละ user
- Controller: ใช้ในการจัดการ ข้อมูลและการติดต่อระหว่าง model และ view ในการ
- GroupController: ใช้ในการจัดการกลุ่ม เช่น การสร้างกลุ่ม การลบกลุ่ม การเพิ่มลบ user ลงในกลุ่ม
- FriendController: ใช้ในการจัดการเพื่อน เช่น การเพิ่มลบเพื่อน
- RegisterController: ใช้ในการจัดการ Register
- LoginController: ใช้ในการจัดการ login ของ user และเก็บ log ของการ login โดยการติดต่อกับ SessionController
- MemberController: ใช้ในการจัดการรายชื่อเพื่อน และ รายชื่อกลุ่ม
- MessageController: ใช้ในการสร้างข้อความที่ต้องการส่งว่าส่งไปที่ เพื่อน หรือ กลุ่ม โดยการติดต่อกับ Member
- UserController: ใช้ในการ user เช่น status การดึงชื่อ user

Sequence diagram ของ use case

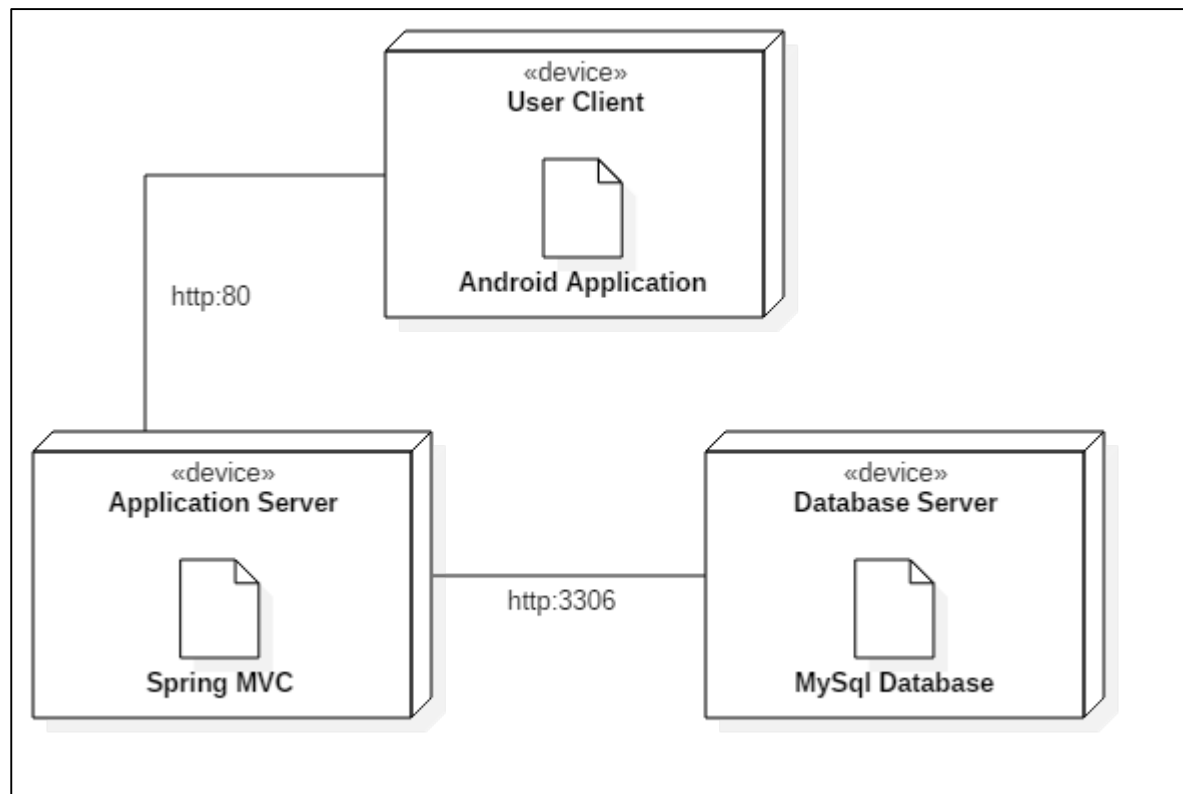
ระบบMark



ระบบเพิ่มเพื่อน



– Deployment Diagrams



– Application Server

เป็น linux centos v.7 ที่มีการติดตั้ง Spring MVC

ทำหน้าที่ในการรับ http request แล้วทำการ เรียกใช้ Spring MVC ตามที่ได้รับ request

– Database Server

เป็น linux centos v.7 ที่มีการติดตั้ง Mysql Service ไว้

ทำหน้าที่ในการรับ http request ของ database แล้วทำการ Query Database จาก Mysql แล้วตอบกลับไปที่ Application Server

– User Client

เป็น Mobile Client เช่น โทรศัพท์ หรือ tablet

ทำหน้าที่ในการรับส่งข้อมูล http request ของ Android Application

## IMPLEMENTATION PLAN

---

### o Milestones

ลำดับ	การทำงาน	วันที่
1	ติดตั้งและศึกษา My SQL	3-4 ตุลาคม 2558
2	ติดตั้งและศึกษา Spring MVC	3-4 ตุลาคม 2558
3	ออกแบบระบบฝั่ง Frontend	2-7 ตุลาคม 2558
4	ออกแบบระบบฝั่ง Backend	3-8 ตุลาคม 2558
5	ทำระบบ Register	9-11 ตุลาคม 2558
6	ทำระบบ Login	10-12 ตุลาคม 2558
7	ทำระบบกู้คืน Password	11-13 ตุลาคม 2558
8	ทำระบบจัดการข้อมูลส่วนตัว	14-17 ตุลาคม 2558
9	ทำระบบเพื่อน	15-18 ตุลาคม 2558
10	ทำระบบสนทนา	18-22 ตุลาคม 2558
11	ทำระบบ Mark	22-25 ตุลาคม 2558
12	ทำระบบแจ้งเตือน	22-25 ตุลาคม 2558
13	ทำระบบค้นหา	23-26 ตุลาคม 2558
14	ทำระบบสนทนาแบบกลุ่ม	27 ตุลาคม - 1 พฤศจิกายน 2558
15	ทำระบบ Filter mark	29 ตุลาคม - 1 พฤศจิกายน 2558
16	ทดสอบระบบ	2-3 พฤศจิกายน 2558
17	แก้ไขระบบ	4-8 พฤศจิกายน 2558