



Object-oriented Analysis and Design, fall 2015

### Cloud Monitoring & Planner Application

55011082	Wannapong	Charoenruangsap
56010784	Pongpayak	Boontaetae
56011009	Ratchanon	Thongkookiatikoon
56011044	Wachirapan	Surasorn
56011055	Worranittha	Kraisittipong
56011127	Winut	Jiraruekmongkol
56011190	Srikote	Naewchampa
56011219	Phanakawin	Thongtae

Department of Computer Engineering

Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang

## Cloud Monitoring & Planner Application

### Analysis

#### ○ Use Cases

- ยืนยันตัวตนผู้ใช้  
ผู้ใช้เข้าใช้งาน Application กรอกชื่อผู้ใช้ และรหัสผ่าน แล้วกดเข้าสู่ระบบ
- เข้าดู Dashboard  
ผู้ใช้เข้าใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Dashboard เพื่อดูสถานะและกราฟของทรัพยากรระบบ
- ได้รับข้อมูลแจ้งเตือนการใช้งาน  
ผู้ใช้เข้าใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Message เพื่อดูข้อความแจ้งเตือนการใช้งานต่างๆ เช่น แจ้งเตือนเมื่อระบบมีการใช้งานทรัพยากรในระดับที่สูง
- รับรายงานการใช้งานทรัพยากร  
ผู้ใช้เข้าใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Report เพื่อดูบันทึกรายงานการใช้งานของระบบในแต่ละเดือน
- รับใบเสร็จค่าใช้จ่ายบริการ  
ผู้ใช้เข้าใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Bill เพื่อดูใบเสร็จค่าใช้จ่ายบริการทั้งหมดของแต่ละเดือน
- รับข้อมูลการใช้งานบริการในอนาคต  
ผู้ใช้เข้าใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Plan เลือก Recommended Plan เพื่อดู Plan ที่ระบบแนะนำเพื่อให้เหมาะสมกับการใช้งานของผู้ใช้ในเดือนถัดไป

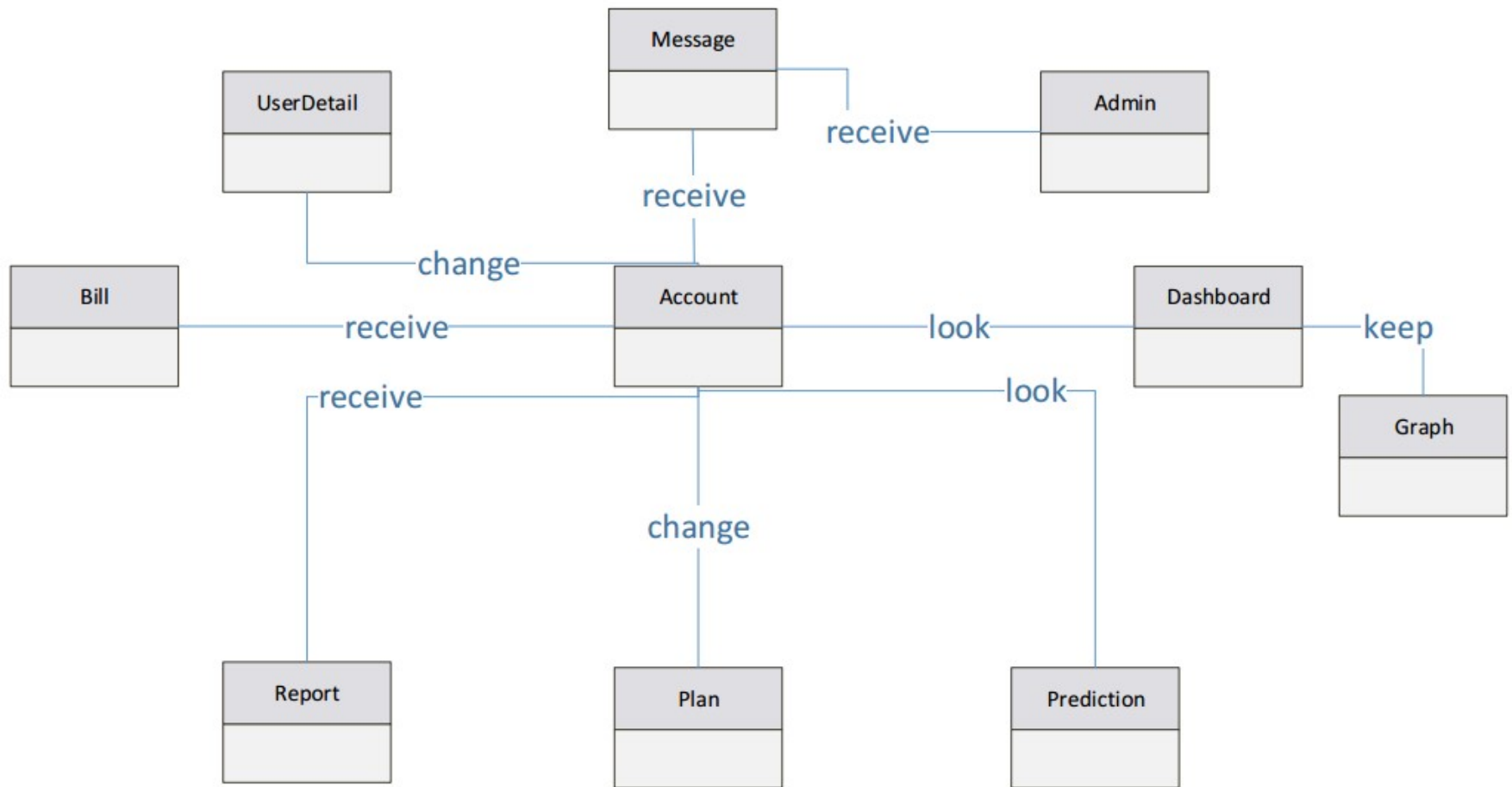
- เปลี่ยนแปลงค่าของทรัพยากร

ผู้ใช้ใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Plan เลือก Change Plan เพื่อปรับเปลี่ยน Plan ของระบบที่ใช้งาน

- เปลี่ยนโปรไฟล์ผู้ใช้บริการ

ผู้ใช้ใช้งาน Application หลังจากเข้าสู่ระบบแล้ว เลือกใช้เมนู Profile เลือก Edit แล้วแก้ไขข้อมูลตามที่ผู้ใช้ต้องการแล้วบันทึก

○ Domain Model



### จาก Domain Model ด้านบน อธิบายได้ว่า

- Account สามารถเปลี่ยนแปลงข้อมูลส่วน UserDetail และ Plan ได้
- Account สามารถดูสถานะการใช้งาน Cloud จาก Dashboard และดู Prediction ที่เหมาะสมสำหรับผู้ใช้งานจากระบบได้
- Account จะได้รับ Message จากระบบ และได้รับ Bill และ Report ของแต่ละเดือนได้
- Admin ได้รับ Message จากระบบกรณีที่ทรัพยากรไม่เพียงพอต่อการใช้งานในปัจจุบัน
- Dashboard เก็บ Graph แสดงการใช้ทรัพยากรของผู้ใช้

○ Abstraction

Account

ผู้ใช้งานหลักของระบบ

Bill

เป็นรายจ่ายของแต่ละเดือน ขึ้นอยู่กับ Plan ที่เลือกใช้ และการใช้งานของผู้ใช้

Report

รวมยอดการใช้งานของแต่ละเดือน แจ้งเป็น Report

Plan

แผนการใช้งานของ Cloud ซึ่งผู้ใช้สามารถปรับเปลี่ยนได้

Prediction

การแนะนำ Plan ในอนาคตจากระบบ

Dashboard

บอกสถานะการใช้งาน Cloud ของผู้ใช้ โดยแสดงเป็นกราฟ

Graph

กราฟบอกสถานะการใช้งาน Cloud ของผู้ใช้

Admin

ผู้ดูแลระบบ Cloud

Message

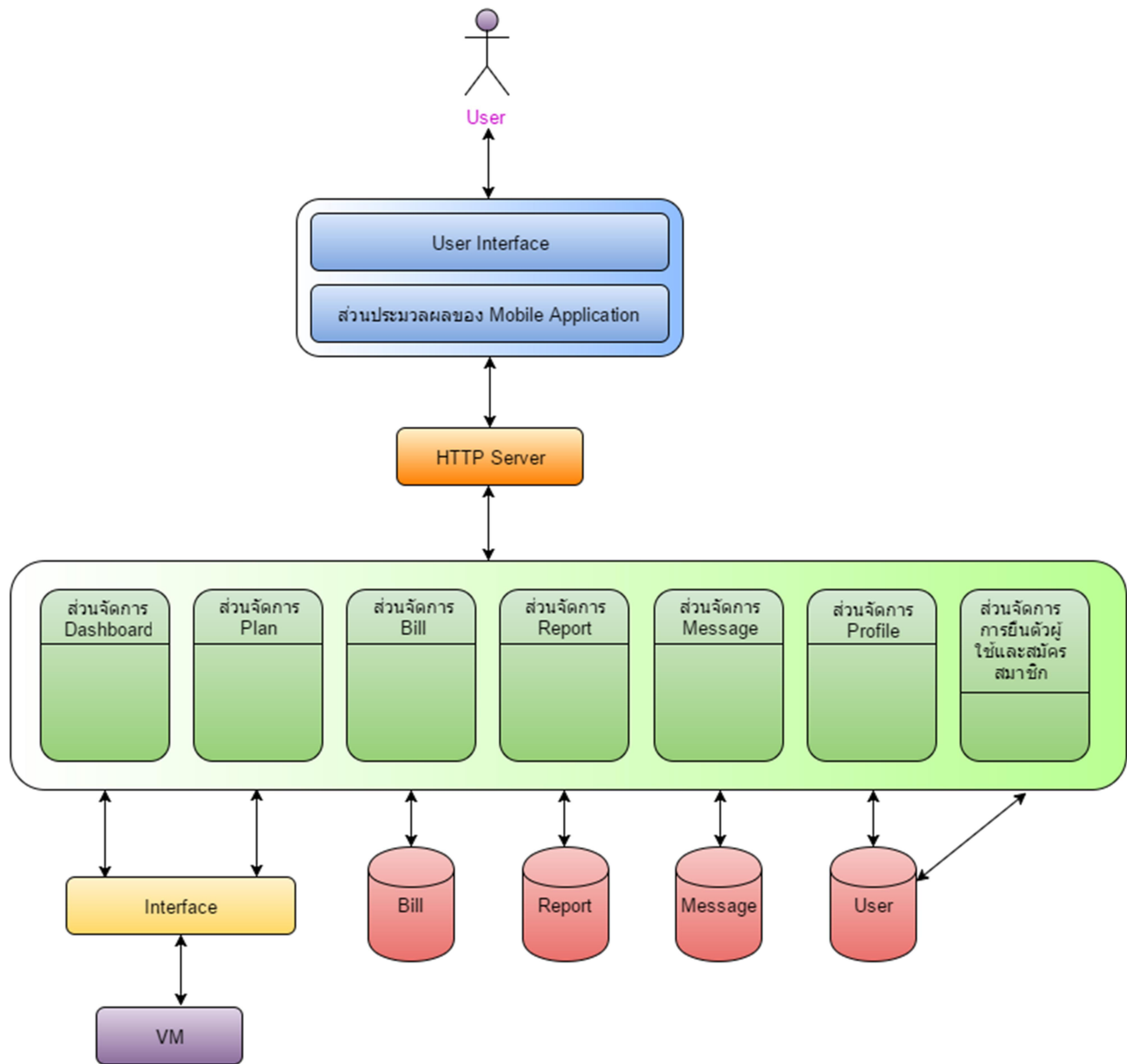
ข้อความจากระบบ ซึ่งจะมีทั้งส่งไปบอกผู้ใช้ และส่งไปหา Admin

UserDetail

รายละเอียดของผู้ใช้ อาจจะประกอบด้วย ชื่อ, email, Id เพื่อระบุตัวผู้ใช้

## Design

### ○ Application Architecture





## ○ Subsystems/Components

- ส่วนติดต่อ Dashboard

- ตัวจัดการข้อมูลทรัพยากร : มีหน้าที่รับคำร้องขอ ดู Dashboard จากผู้ใช้และส่งคำร้องขอข้อมูลทรัพยากรระบบของผู้ใช้ทั้งหมดจากระบบของผู้ให้บริการ และนำข้อมูลนั้นนำส่งไปยังหน่วยแสดงผล

- ส่วนติดต่อ Plan

- ตัวจัดการแสดงผลแผนการงานปัจจุบัน : มีหน้าที่นำข้อมูลแผนการใช้งานปัจจุบันของผู้ใช้ส่งไปยังหน่วยแสดงผล
  - ตัวจัดการการเปลี่ยนแปลงแผนการใช้งาน : มีหน้าที่ในการรับข้อมูลการปรับเปลี่ยนทรัพยากรจากผู้ใช้ แล้วติดต่อไปยัง Interface ที่ติดต่อกับ VM เพื่อปรับสเกลทรัพยากรต่าง ๆ ตามแผนที่ผู้ใช้เลือก

- ส่วนติดต่อ Bill

- ตัวคำนวณค่าใช้จ่ายของผู้ใช้งานรายเดือน : มีหน้าที่คำนวณค่าใช้จ่ายของระบบ Cloud ทั้งหมดที่ผู้ใช้ใช้บริการอยู่ โดยจะคิดเป็นรายเดือนหากเดือนไหนไม่มีการชำระจะมีการทยอยค่าใช้จ่ายไปเดือนถัดไป หากเกินสองเดือนจะมีการแจ้งข้อความเตือนไปยังผู้ใช้

- ส่วนติดต่อ Report

- ตัวประมวลผลสถิติการใช้งาน : มีหน้าที่คำนวณสถิติการใช้งานในแต่ละเดือน โดยนำข้อมูลการใช้งานที่เก็บไว้ใน Database นำมาวิเคราะห์การใช้งานในเดือนที่ผ่านมาหนึ่งเดือน และนำผลที่ได้จากการคำนวณไปทำการแนะนำ Plan ที่เหมาะสมกับเจ้าของระบบ(Prediction)
  - ตัวบันทึกการใช้งาน : จะทำการตรวจจับค่าของการใช้งานของแต่ละทรัพยากรในแต่ละวัน โดยจะบันทึกค่าการใช้งานสูงสุด ค่าต่ำสุด และเวลาลงไปยัง Database

- ตัวคำนวณการใช้งาน : จะทำการดึงข้อมูลสถิติในแต่ละเดือน มาทำแบบรายงานผล  
การใช้งานในแต่ละวัน และ คำนวณสรุปสถิติการใช้งานรายเดือนว่าช่วงวันและเวลาไหน  
ของเดือนมีการใช้งานมากที่สุดและน้อยสุด แล้วส่งไปให้ ส่วนแสดงผล

- ส่วนติดต่อ Message

- ตัวจัดการควบคุมข้อความ : มีหน้าที่รับคำร้องขอ ดู Message จากผู้ใช้และส่งคำร้อง  
ขอข้อความทั้งหมดของผู้ใช้ทั้งหมดจาก Database และนำข้อมูลนั้น นำส่งไปยังหน่วย  
แสดงผล โดยมีการสร้าง Object Message ซึ่งจะประกอบไปด้วยข้อความและส่วน  
สถานะการอ่านข้อความ

- ตัวจัดการอัปเดตข้อความ : อัปเดตข้อความที่เข้ามาใหม่ และจัดการสถานะการอ่าน  
ข้อความ

- ส่วนติดต่อ Profile

- ตัวจัดการแสดงผลข้อมูลผู้ใช้ : มีหน้าที่นำข้อมูลผู้ใช้งานส่งไปยังหน่วยแสดงผล

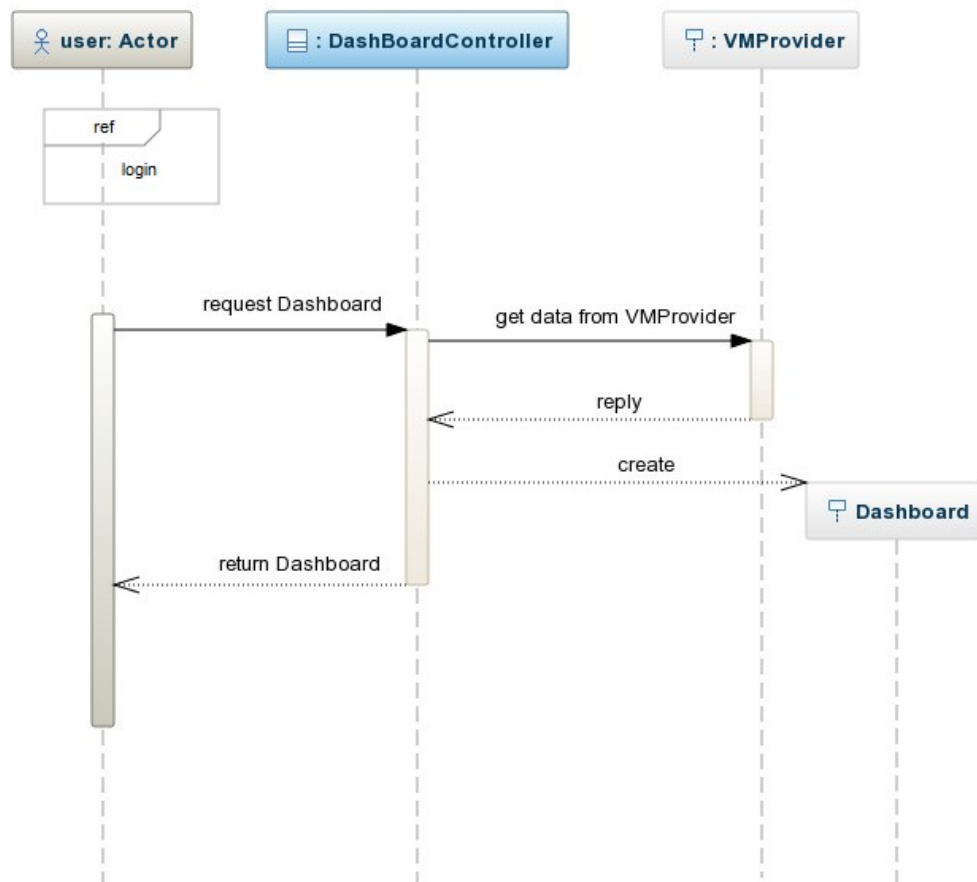
- ตัวจัดการแก้ไขข้อมูลผู้ใช้ : มีหน้าที่ในการแก้ไขข้อมูลต่าง ๆ ของผู้ใช้งาน เช่น ชื่อ  
อีเมล พาสเวิร์ด เป็นต้น โดยจะรับข้อมูลจากผู้ใช้งานแล้วทำการ Update ข้อมูลใน  
Database

- ส่วนจัดการยืนยันตัวและสมัครสมาชิกผู้ใช้งาน

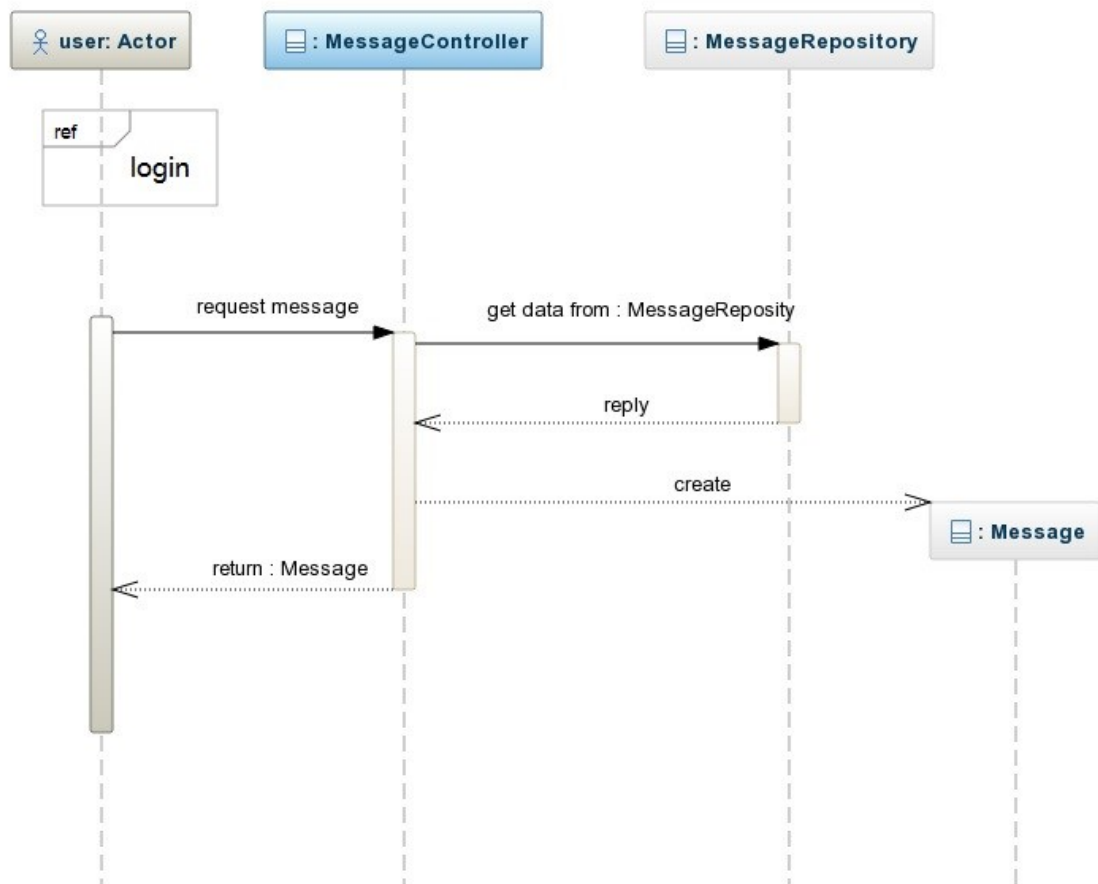
- ส่วนจัดการยืนยันตัวผู้ใช้ : มีหน้าที่ตรวจสอบข้อมูลยืนยันตัวตนผู้ใช้ มีอยู่ใน  
Database หรือไม่ หากไม่มีข้อมูลที่ตรงกับค่าที่ได้รับมาให้รีเทิร์นค่าผิดพลาดไป

- ส่วนจัดการสมัครสมาชิก : มีหน้าที่นำข้อมูลที่ใช้กรอกสมัครสมาชิก ไปทำการ Insert  
ลงใน Database ผู้ใช้งาน

## Sequence diagram

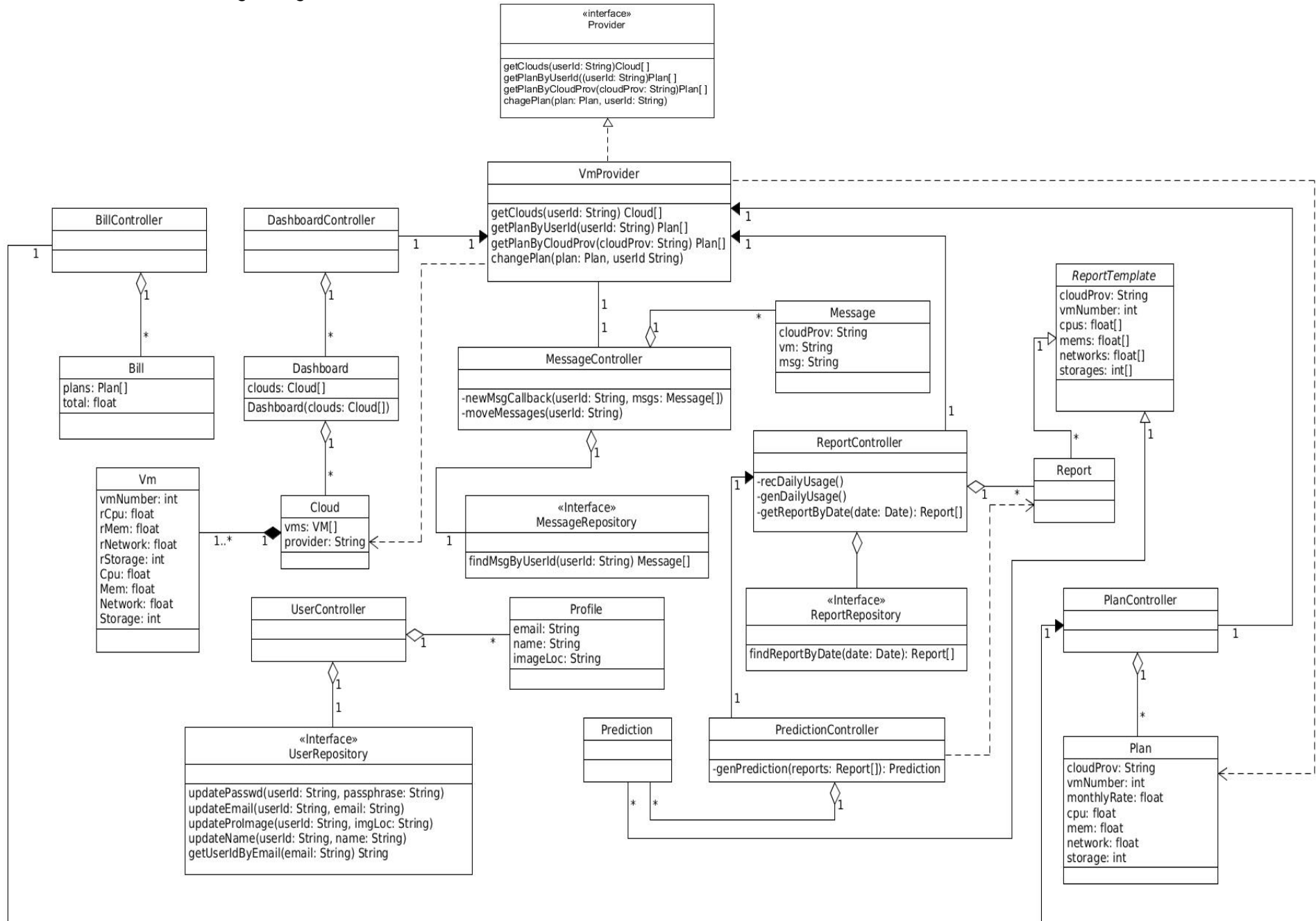


จากรูป คือภาพ Sequence diagram ของ Subsystem ส่วนติดต่อ Dashboard โดยเมื่อผู้ใช้ Login เข้าใช้งานแล้วสำเร็จแล้ว จากนั้นเลือกเมนู Dashboard โดย Request ข้อความไปยังคลาส DashboardController จากนั้น DashboardController จะการนำ ID ส่งไปยังส่วนของ VMProvider เพื่อให้รีเทิร์นข้อมูลระบบทั้งหมดของผู้ใช้กลับมายัง DashboardController จากนั้น DashboardController จะทำการสร้าง Object Dashboard โดยเซตข้อมูลที่ได้รับจาก VMProvider แล้วทำการรีเทิร์นค่าของ Object Dashboard ไปยังผู้ใช้



จากรูป คือภาพ Sequence diagram ของ Subsystem ส่วนติดต่อ Message โดยเมื่อผู้ใช้ Login เข้าใช้งานแล้วสำเร็จแล้ว จากนั้นเลือกเมนู Message โดย Request ข้อความไปยังคลาส MessageController จากนั้น MessageController จะการนำ ID ส่งไปยังส่วนของ MessageRepository เพื่อให้รีเทิร์นข้อความทั้งหมดของผู้ใช้กลับมายัง MessageController จากนั้น MessageController จะทำการสร้าง object Message โดยเซตข้อมูลที่ได้รับจาก MessageRepository แล้วทำการรีเทิร์นค่าของ Object Message ไปยังผู้ใช้

## ○ Design Diagram

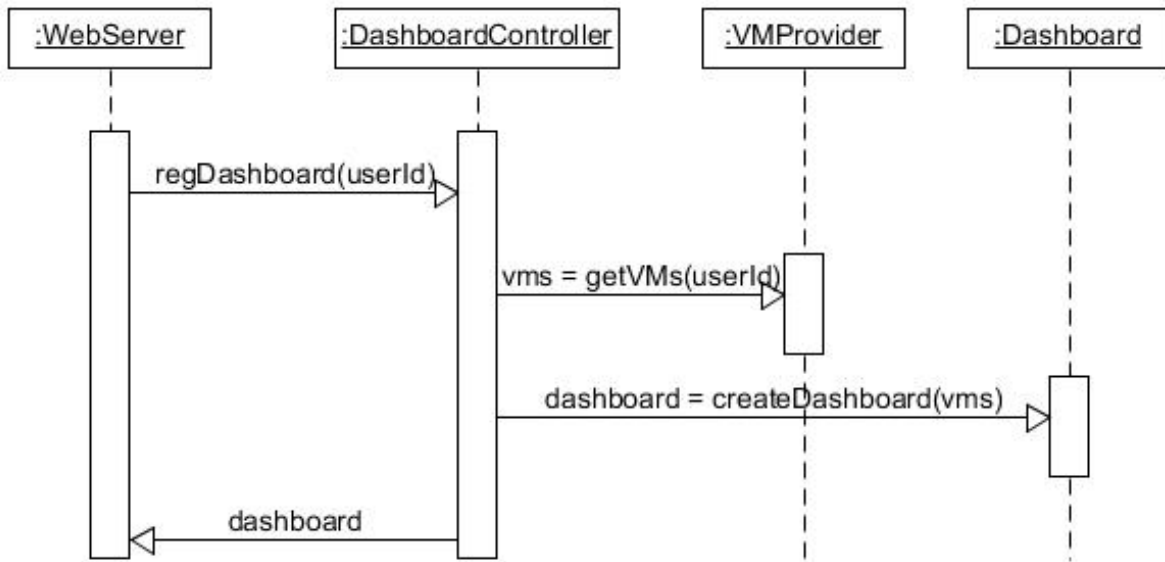


### จาก Design Diagram ด้านบน จะอธิบายได้ว่า

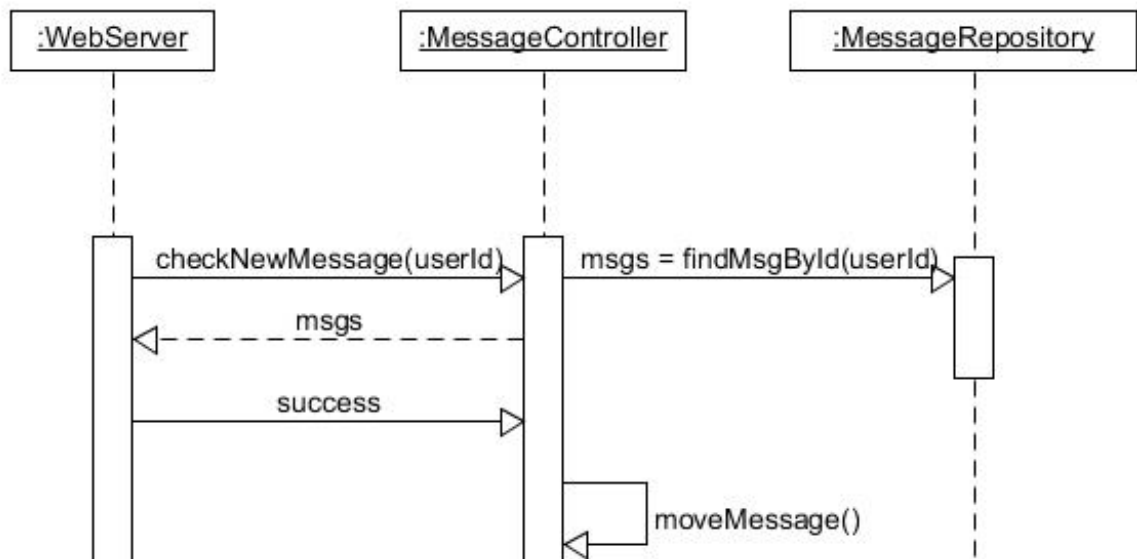
- Class Controller ทุก Class ทำหน้าที่รับ Request ที่มาจากฝั่ง Android โดยแต่ละ Class จะแยกกันจัดการ (Handle) Request ตามหน้าที่ของตัวเอง เช่น DashboardController จะจัดการกับ Request ที่เกี่ยวข้องกับหน้า Dashboard ใน Android Application อาทิ การร้องขอหน้า Dashboard
- Interface Repository ใช้ในการติดต่อกับ Database โดยในส่วนของการ Implementation ตัว Spring Framework จะเป็นตัวจัดการให้
- ReportTemplate เป็น Abstract Class ที่ถูก Extend โดย Report และ Prediction เนื่องจากทั้งสอง Class นั้นมีลักษณะที่เหมือนกันอยู่ตรงที่ ทั้งสอง Class แสดงข้อมูลเป็นรายงานตามช่วงระยะเวลาหนึ่ง เช่น Report แสดงข้อมูลการใช้ทรัพยากรโดยเฉลี่ยของ Cloud ในแต่ละวันในระยะเวลาหนึ่งเดือนที่ผ่านมา เป็นต้น
- VmProvider เป็นส่วนที่คอยรองรับการติดต่อระหว่างตัว Application และระบบ Cloud ภายนอก
- VmProvider จะส่งข้อความไปให้กับ MessageController เมื่อระบบ Cloud ได้สร้าง Alert Message ขึ้นมา โดยเมื่อ MessageController ได้รับข้อความก็จะไปเรียกใช้ newMessageCallback() Method
- ReportController จะทำการเก็บข้อมูลการใช้ทรัพยากรโดยเฉลี่ยของ Cloud ในแต่ละวันจนครบหนึ่งเดือนแล้วจึงทำการเก็บลงไปใน Database
- Class Cloud และ Class Vm ใช้สำหรับเก็บข้อมูลทรัพยากรที่ได้มาจาก VmProvider เพื่อนำไปใช้แสดงผลใน Dashboard ต่อไป

Sequence diagram ของ use case

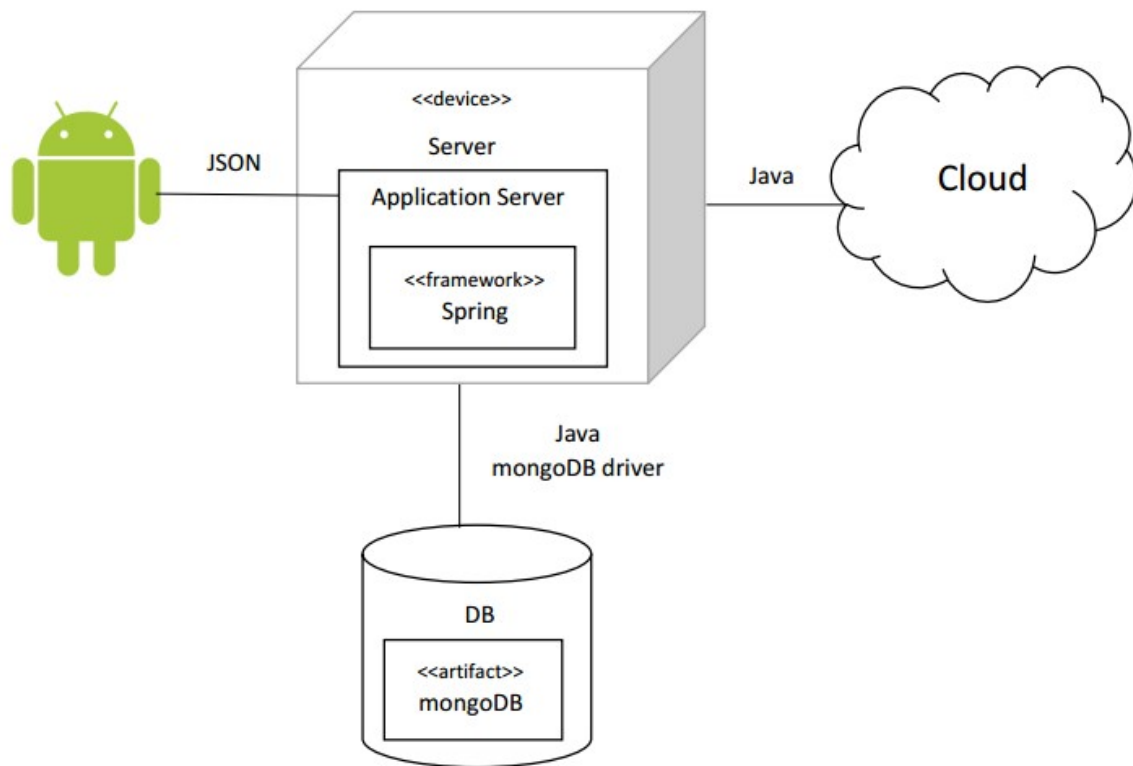
- เข้าดู Dashboard



- ได้รับข้อมูลแจ้งเตือนการใช้งาน



## ○ Deployment Diagram



## Software / Technology

- Application บนระบบปฏิบัติการ Android “CMC”
- Server รัน CentOS 7 สำหรับรัน Spring และ MongoDB
- Spring framework เพื่อรันเป็น RESTful web service ใช้เพื่อติดต่อ ควบคุม และจัดการข้อมูลให้ Users และทำการติดต่อเพื่อเก็บข้อมูลกับ ส่วนจัดการของ Cloud
- MongoDB เพื่อเก็บข้อมูลลงบน Database ทั้งข้อมูลผู้ใช้ ข้อมูลต่างๆของ Cloud (VMs)
- การสื่อสารกันระหว่าง Android application และ Spring จะอยู่ในรูปแบบของ JSON
- การสื่อสารระหว่าง Spring และ Cloud จะมี Interface พัฒนาด้วยภาษา Java



## Implementation Plan

