

Final Project Document & Deliverables

กลุ่ม

MTTX

ชื่อโครงการ

Online music controller

Github repository

CE-KMITL-OOAD-2014/Online-Music-Controller

รายชื่อสมาชิก

นายวรภาพ

บุญประไพ รหัส 55011080

นายสุกฤษฎี

มัทนะไพศาล

รหัส 55011320

วิชา Object Oriented Analysis and Design, ภาคการศึกษาที่ 1/2557

สารบัญ

บทคัดย่อ	1
บทนำและรายละเอียดการวิเคราะห์หัวข้อ	1
งานที่เกี่ยวข้อง	1
ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ	2
แนวทางการใช้งาน	3
สถาปัตยกรรมของระบบ	8
แผนภาพของคลาสหลัก	11
รายละเอียดการพัฒนาซอฟต์แวร์	12
ผลการทดสอบซอฟต์แวร์	13
บทสรุป	18
บรรณานุกรม	19

บทคัดย่อ :

Online music controller เป็นเครื่องเล่นเพลงที่สามารถควบคุมการเล่นเพลงได้ผ่านหน้าเว็บโดยผู้ใช้งานต้องทำการลงทะเบียนและ Log in เพื่อทำการเข้าระบบโดยแต่ละคนจะมีการระบุเครื่องเล่นของตนเอง และสามารถเพิ่มเพลงทางไกลผ่านทางหน้าเว็บ

บทนำและรายละเอียดการวิเคราะห์หัวข้อ :

เสียงเพลง เป็นสิ่งที่อยู่คู่กับสังคมมนุษย์มาอย่างช้านาน โดยที่ทุกวันนี้คนส่วนใหญ่จะมี Music Player เป็นของตัวเองไม่ว่าจะเป็นโทรศัพท์มือถือ หรือ iPod Walkman portable Digital Audio Player แต่สิ่งเหล่านี้เมื่อคนใช้ต้องการจะควบคุมต้องทำการควบคุมที่เครื่องเล่นเท่านั้น ไม่สามารถควบคุมผ่านสื่ออื่นได้ ทำให้เกิดความยุ่งยาก ทางผู้พัฒนาได้เห็นถึงปัญหานี้จึงคิดที่จะพัฒนาเครื่องเล่นที่สามารถควบคุมได้ผ่านสื่ออื่น โดยเริ่มต้นที่การควบคุมผ่านหน้าเว็บ โดยใช้สื่ออินเทอร์เน็ตในการเชื่อมต่อกับเครื่องเล่นเพลง เพื่อมานำแก้ปัญหาดังกล่าว

งานที่เกี่ยวข้อง :

โดยตัว online music controller นั้น ได้รับแรงบันดาลใจมาจากซอฟต์แวร์ RuneAudio (<http://www.runeaudio.com/>) ซึ่งมีความสามารถในการแปลง embedded hardware ให้กลายเป็นเครื่องเล่นเพลงที่สามารถควบคุมที่ไหนก็ได้ผ่าน application ของ RuneAudio เอง โดย source ของเพลงนั้นจะมาจากภายนอก เช่น flash drive ,web radio เป็นต้น ซึ่งจะต่างจาก online music controller ที่จะใช้ storage ของตัวเครื่องเล่นเองในการเก็บเพลง และเลือกใช้ช่องทางในการติดต่อเป็น ช่องทางเว็บ เพื่อให้สามารถควบคุมได้ผ่านอุปกรณ์หลากหลายชนิด พร้อมใช้ระบบ login เพื่อเข้าถึงเครื่องเล่นเพลงที่ต้องการ

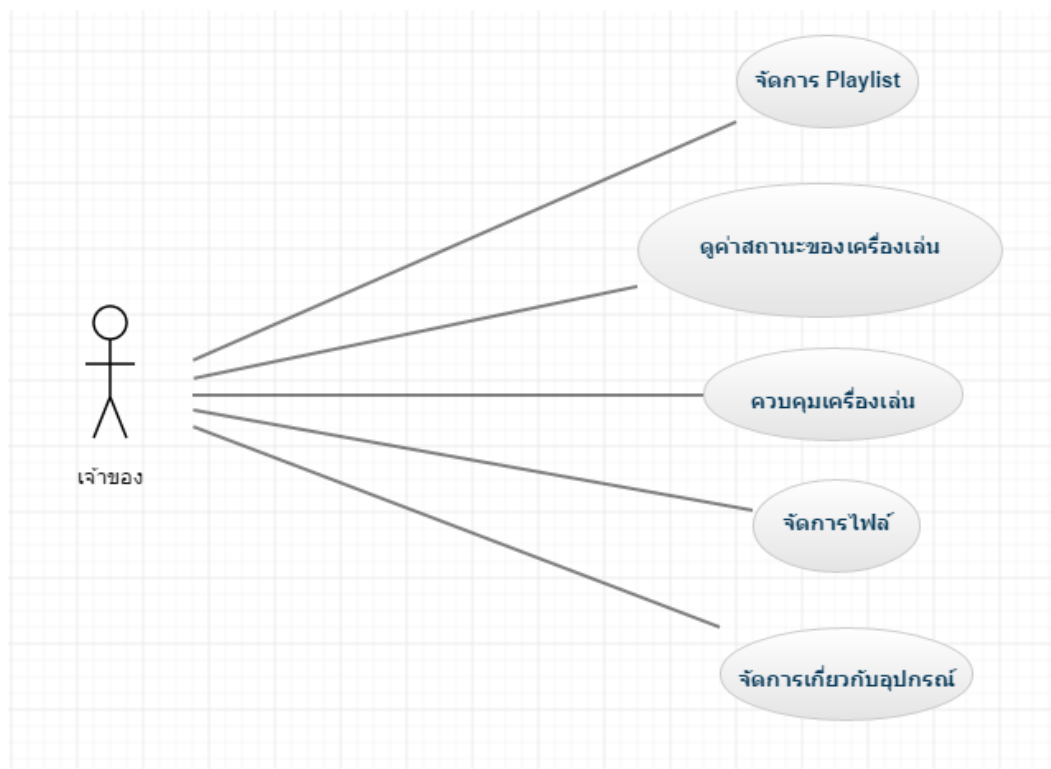
ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

Functional requirement
การระบุตัวตนผู้ใช้
สามารถเล่นเพลงนามสกุล mp3, wav ได้
ผู้ใช้แต่ละคนจะมี playlist ของตนเองได้
เจ้าของเครื่องจะสามารถเพิ่ม/ลบ ไฟล์เพลงในเครื่องเล่นได้
ควบคุมและแสดงผลการทำงานผ่านหน้าเว็บเพจ

Non-Functional requirement
แสดงการเปลี่ยนแปลงหน้าเว็บแบบทันทีทันใด(real-time)
ปลอดภัยจากช่องโหว่จากคุกกี้ (cookie vulnerabilities) และการส่งคำร้อง (request vulnerabilities)
มีความสะดวกสบายจากการควบคุมผ่านอุปกรณ์หลายรูปแบบ
มีความเสถียรในการรองรับผู้ใช้หลายคน
รองรับ AUX 3.5 มม.
เครื่องเล่นใช้พลังงานต่ำ

แนวทางการใช้งาน

Use-case diagram



Use case specifications

Use case name : ควบคุมเครื่องเล่น

Use case purpose : ใช้ในการควบคุมเครื่องเล่น (Play/Pause Backward/Forward Stop) โดย User เป็นคนป้อนคำสั่งโดยการกดปุ่มผ่านหน้าเว็บ

Preconditions : ต้องอยู่ในสถานะเข้าสู่ระบบและมีเครื่องเล่นที่ Online อยู่

Postconditions : เครื่องเล่นจะทำตามคำสั่งของผู้ใช้

Limitations : เครื่องเล่นต้องอยู่ในสถานะ Online

Assumption : เครื่องเล่นจะอยู่ในสถานะออนไลน์ตลอดเวลา

Primary scenario :

- 1) เข้าสู่ระบบ
- 2) กดมาที่หน้าควบคุมเครื่องเล่น
- 3) เลือกเครื่องเล่น
- 4) เลือกการควบคุมผ่านปุ่มบนหน้าเว็บ

Alternate scenarios :

Condition 1: บัญชีหรือรหัสผ่านผู้ใช้ไม่ถูกต้อง

1a) ระบบจะทำการย้อนกลับไปหน้าจอ เข้าสู่ระบบ พร้อมหน้าแจ้งเตือน

1b) กลับไปยัง Primary scenario

- Use case name** : จัดการไฟล์
- Use case purpose** : ทำให้ผู้ใช้ที่มีสถานะเป็นเจ้าของเครื่องเล่นสามารถจัดการไฟล์ (เพิ่ม / ลบ)
- Preconditions** : ต้องอยู่ในสถานะเข้าสู่ระบบ และมีเครื่องเล่นที่ลงทะเบียนแล้ว
- Postconditions** : เครื่องเล่นผู้ใช้ระบบจะมีเพลงที่ผู้ใช้อัปโหลด
- Limitations** : ชื่อไฟล์เพลงต้องไม่มีอักขระพิเศษหรือช่องว่าง และเครื่องเล่นเพลงต้องออนไลน์อยู่
- Assumption** : ผู้ใช้จะไม่อัปโหลดไฟล์ที่ไม่ใช่ไฟล์เพลงลงไป

Primary scenario :

- A) User เข้าสู่ระบบ
- B) กดมาที่หน้าจัดการบัญชีผู้ใช้
- C) เลือกเพลงจากช่อง Choose file
- D) กดที่ปุ่ม "upload"
- E) ระบบจะทำการอัปโหลดไฟล์และอัปเดตฐานข้อมูลเพลง

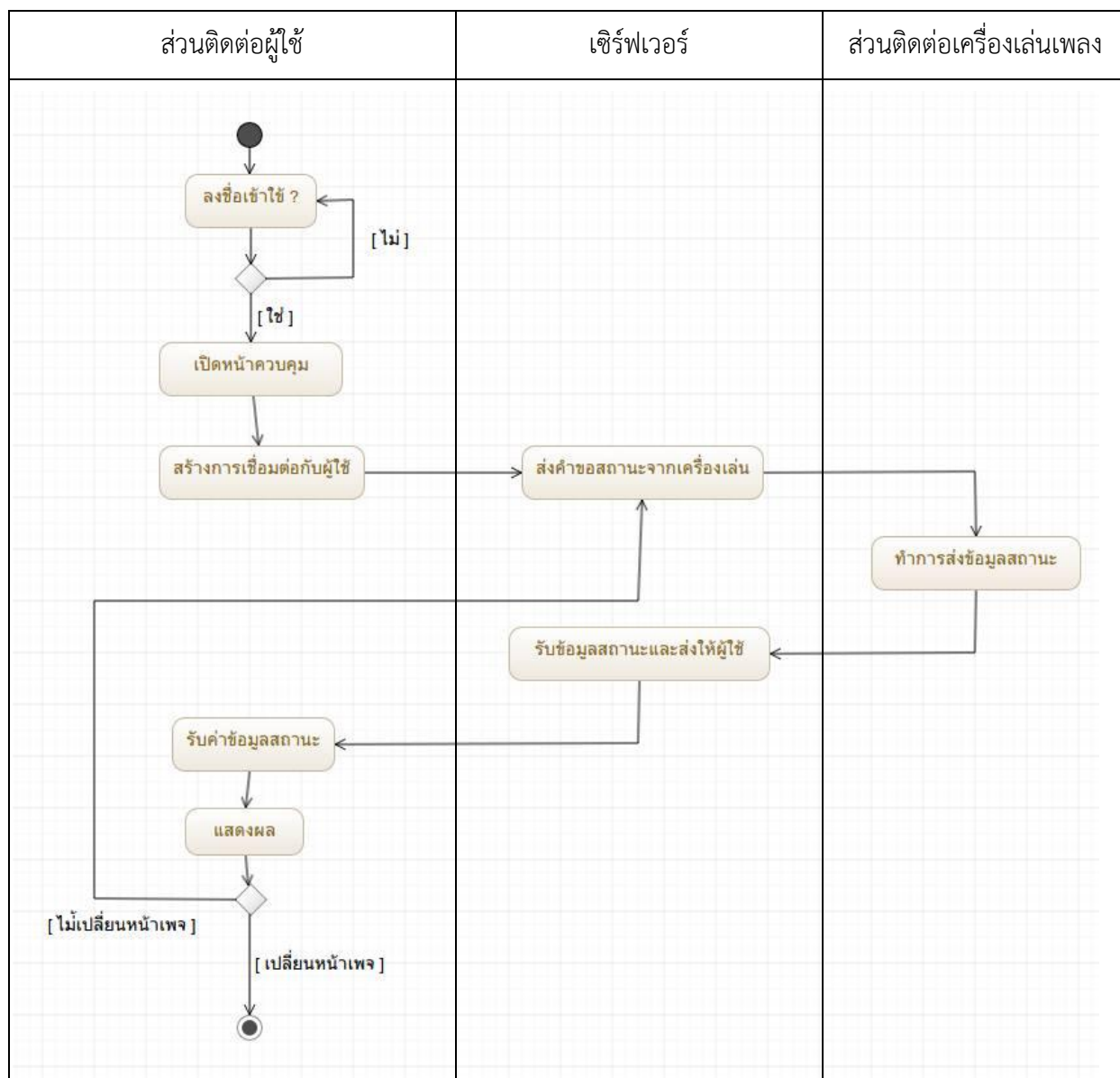
Alternate scenarios :

Condition 1: บัญชีหรือรหัสผ่านผู้ใช้ไม่ถูกต้อง

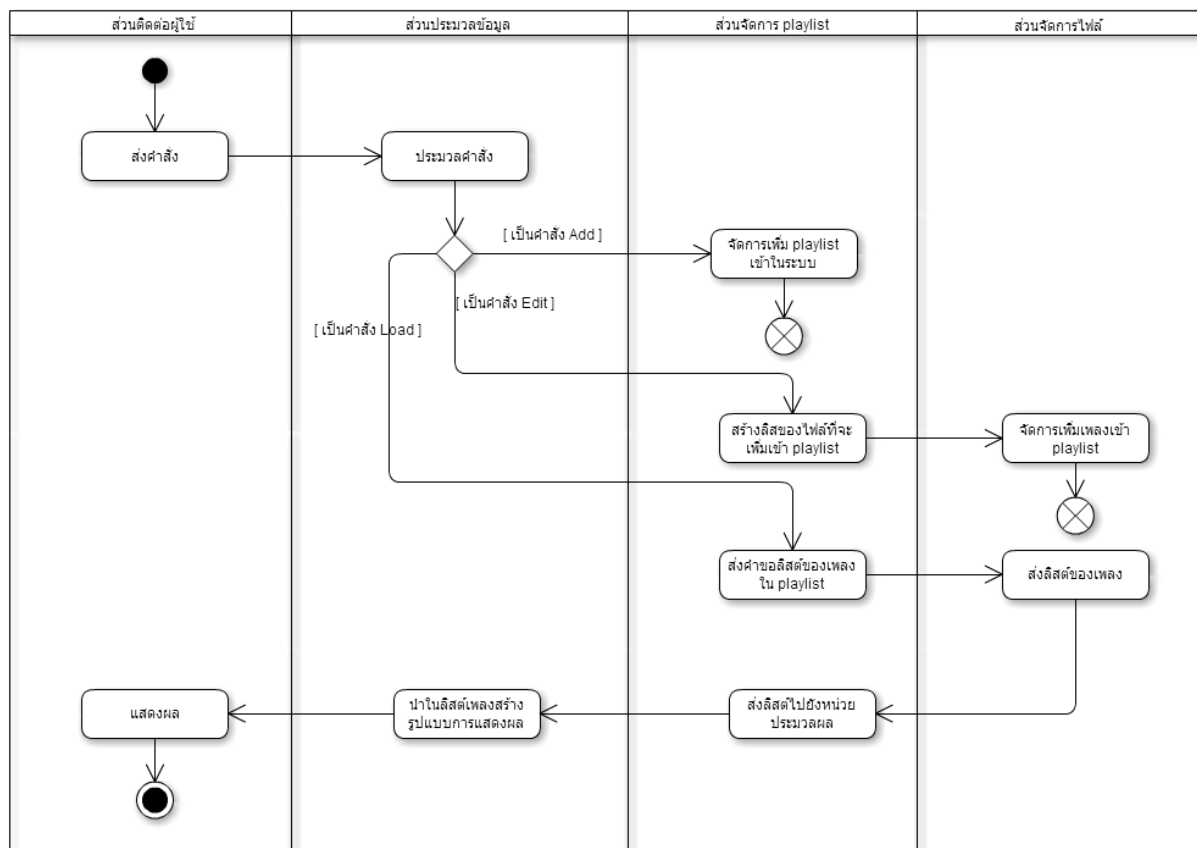
- A1) ผู้ใช้กรอก ชื่อผู้ใช้หรือรหัสผ่าน ไม่ถูกต้อง
- A2) ระบบจะทำการย้อนกลับไปที่หน้า เข้าสู่ระบบ อีกครั้ง
- A3) กลับไปยัง Primary scenario

Activity diagram

Use case name : ดูค่าสถานะเครื่องเล่น



Use case name : จัดการ playlists



สถาปัตยกรรมของระบบ

Problem Analysis

ระบบของ Online music player จะเกี่ยวข้องกับการควบคุมเครื่องเล่นเพลงโดยมีการเข้าถึงจากผู้ใช้งานหลายคน ทั้งยังมีการจัดหมวดหมู่เพลง และเพิ่มลบไฟล์เพลงได้ ดังนั้นจึงสามารถวิเคราะห์ส่วนของปัญหาได้ดังนี้

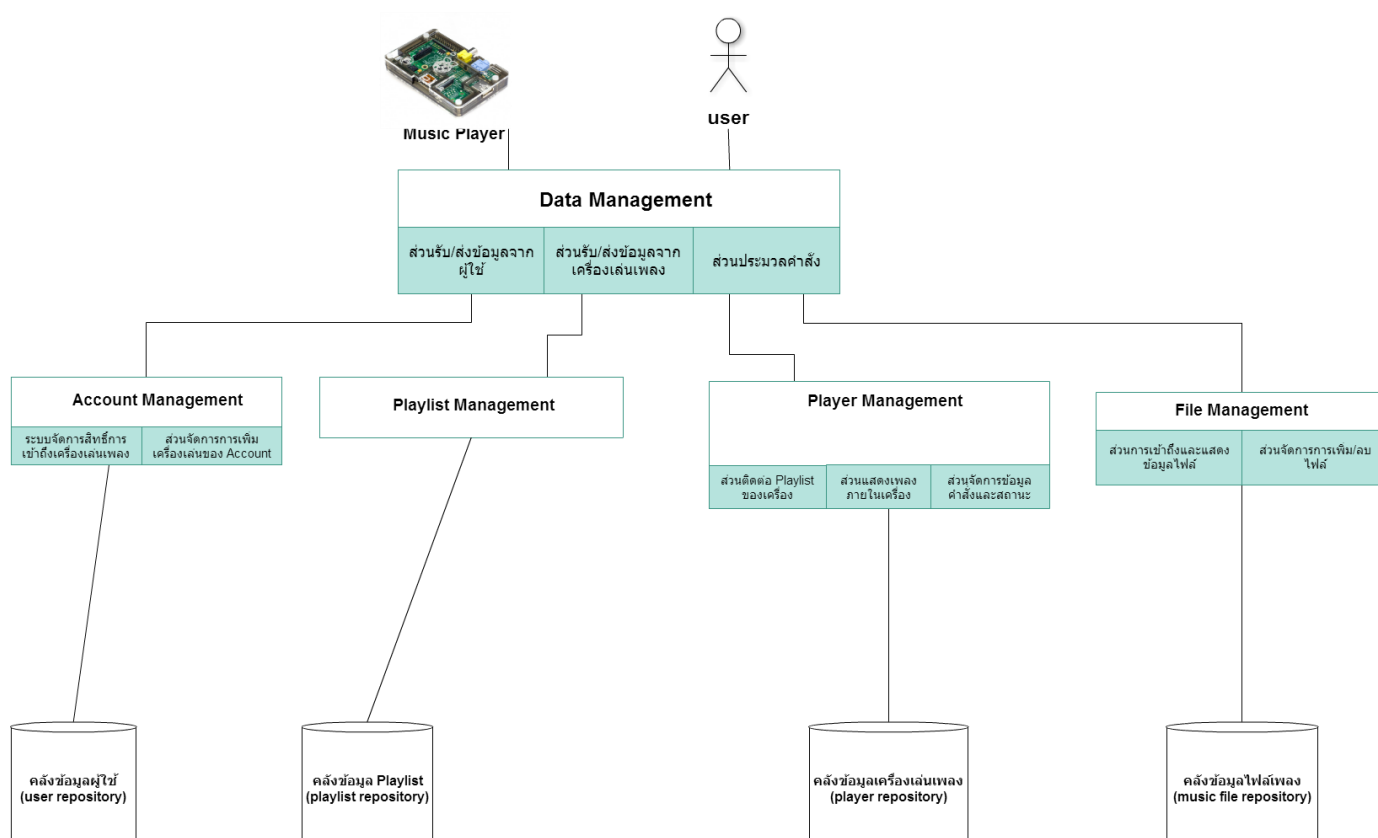
Abstraction

- ผู้ใช้ ประกอบไปด้วย ข้อมูลผู้ใช้ อุปกรณ์ของผู้ใช้ อุปกรณ์ที่ผู้ใช้มีสิทธิ์เข้าถึง
- ไฟล์เพลง ประกอบไปด้วย ID3(ชื่อเพลง, ชนิดเพลง, ความยาวเพลง, ศิลปิน, อัลบั้ม, Bit Rate)
- เพลลิสต์ ประกอบด้วย ชื่อเพลลิสต์ รายชื่อเพลงในลิสต์
- เครื่องเล่นเพลง ประกอบไปด้วย IDเครื่องเล่น รายชื่อเพลงที่มี สถานะเครื่องเล่นเพลง

Component

- ส่วนรับ/ส่งข้อมูลจากผู้ใช้
- ส่วนรับ/ส่งข้อมูลจากเครื่องเล่นเพลง
- ส่วนประมวลผลคำสั่ง
- ส่วนติดต่อเพลลิสต์ของเครื่อง
- ส่วนแสดงเพลงภายในเครื่อง
- ส่วนจัดการข้อมูลคำสั่งและสถานะของเครื่อง
- ระบบจัดการสิทธิ์การเข้าถึงเครื่องเล่นเพลง
- ส่วนจัดการการเพิ่มเครื่องเล่นของบัญชีผู้ใช้
- ส่วนเข้าถึงและแสดงข้อมูลไฟล์
- ส่วนจัดการการเพิ่ม/ลบ ไฟล์

Application Architecture



Data Management ใช้ในการรับส่งข้อมูลติดต่อกับ User และ Player และมีการประมวลผลเพื่อจะข้อมูลคำสั่งไปยังส่วนการทำงานที่ถูกต้อง (Route)

- ส่วนรับ/ส่งข้อมูลจาก User : เป็นส่วนที่ใช้สำหรับการติดต่อระหว่างระบบกับ User
- ส่วนรับ/ส่งข้อมูลจาก Player : เป็นส่วนที่ใช้สำหรับการติดต่อระหว่างระบบกับ Player
- ส่วนประมวลผลคำสั่ง : นำข้อมูลที่ได้มาจากสองส่วนด้านบนมาประมวลผลและส่งกลับไป User หรือ Player

Player Management ใช้แสดงถึงเครื่องเล่นเพลง มีการจัดการกับเพลง Playlist ข้อมูลคำสั่งและสถานะ

- ส่วนแสดงเพลงภายในเครื่อง : ใช้สำหรับการแสดงเพลงภายในเครื่องเล่นที่ระบุ
- ส่วนติดต่อ Playlist ของเครื่อง : ใช้สำหรับการประมวลผลคำสั่งเกี่ยวกับ Playlist ของเครื่องเล่น
- ส่วนประมวลผลข้อมูลคำสั่งและสถานะ : ใช้สำหรับการจัดการข้อมูลคำสั่งและสถานะของเครื่องเล่นเพลง

Playlist Management ใช้แสดงถึง Playlist การจัดการภายใน และฐานข้อมูลของ Playlist

Account Management ใช้จัดการการเข้าถึงเครื่องเล่นเครื่องเล่นของผู้ใช้ภายในระบบ

- ระบบจัดการสิทธิ์การเข้าถึงเครื่องเล่นเพลง : ใช้ในการจัดการสิทธิ์ในการควบคุมเครื่องเล่นของผู้ใช้
- ส่วนจัดการการเพิ่มเครื่องเล่นของ account : ใช้สำหรับเพิ่มเครื่องเล่นของ User

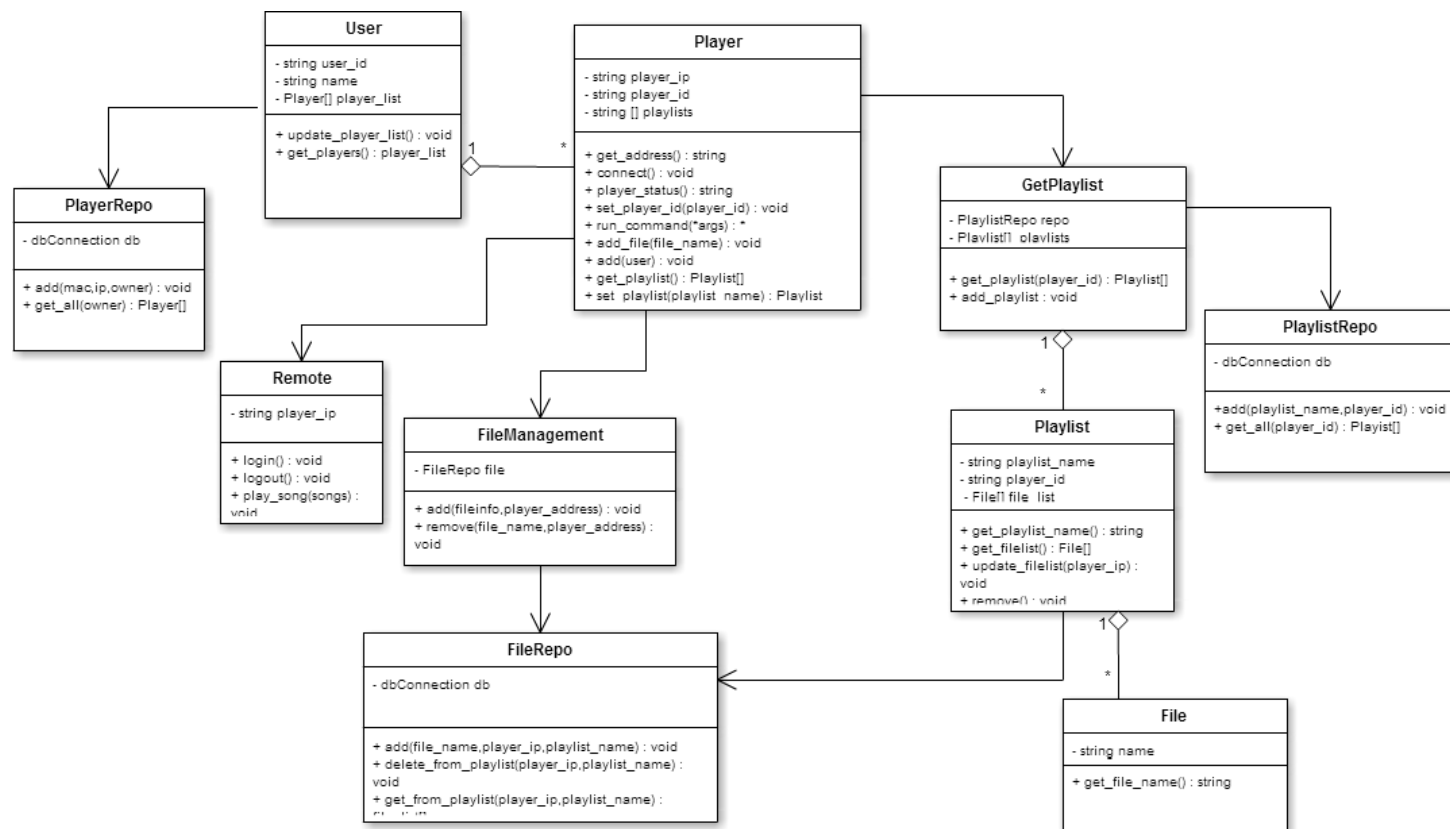
File Management ใช้จัดการไฟล์ข้อมูล และ ฐานข้อมูลของไฟล์ในระบบ

- ส่วนเข้าถึงและแสดงข้อมูลไฟล์ : ใช้สำหรับการดึงข้อมูลไฟล์เพลงจากเครื่องเล่น
- ส่วนจัดการการเพิ่ม/ลบ ไฟล์ : ใช้สำหรับการเพิ่มหรือลบไฟล์

แบ่งการทำงานต่าง ๆ ตามหน้าที่ เพื่อเวลาที่จะเปลี่ยนแปลงหรือปรับปรุงจะได้กระทำการเป็นส่วน ๆ

แผนภาพของคลาสหลัก

Class diagram



User : เป็นคลาสของผู้ใช้งานที่มีการสืบทอดคุณสมบัติในการแสดงคุณสมบัติเจ้าของ หรือ เพื่อใช้ในการเพิ่มอุปกรณ์

Player : เป็นคลาสในการใช้ติดต่อควบคุมอุปกรณ์เครื่องเล่นเพลง รับสถานะการทำงานของเครื่องเล่นเพลง เพิ่มไฟล์ และจัดการเพลลิสต์ในเครื่องเล่นเพลง

Remote : เป็นคลาสที่ใช้ในการจัดการรูปแบบคำสั่งที่ใช้ควบคุมอุปกรณ์เครื่องเล่นเพลง

Playlist : เป็นคลาสที่ใช้ในการจัดการเพลลิสต์เพลง

File : ใช้ในการอ่านค่าหรือจัดการไฟล์ของระบบ

- ติดตั้ง Python Tornado เป็น web server และ web framework, ใช้ระบบปฏิบัติการ Ubuntu 14.04 LTS, ใช้ระบบดาต้าเบส MySQL
- Library ที่ใช้ในการพัฒนาระบบ torndb, paramiko, python-mysql
- ใช้ภาษา Python ในการพัฒนา
- ใช้ Bootstrap เป็น frontend framework
- ระบบทำงานอยู่ใน Server IP address : 161.246.5.31
- มี URL คือ <https://161.246.5.31>

Online music controller			2014															
Deliverables	ผู้รับผิดชอบ	Week(s)	Sep	1	2	3	4	Oct	1	2	3	4	Nov	1	2	3	4	
Planning phase																		
ศึกษาภาษา python	Worapob,Sukrit	3		1	2	3	4			5								
ศึกษา Tornado framework	Worapob,Sukrit	4		n	n	n												
จัดเตรียม enviroments	Sukrit	1			n													
Design	Worapob,Sukrit	3			n	n	n											
จัดทำ Slide Presentation รอบแรก	Worapob,Sukrit	1								n								
Design Presentation	Worapob,Sukrit	1								n								
Implementation phase						1 2			3 4 5									
ออกแบบหน้าของเว็บไซต์	Sukrit	1			n													
ออกแบบและสร้างดาต้าเบส	Worapob	1			n													
ทำระบบตามแบบที่ออกแบบไว้	Worapob,Sukrit	4				n				n	n	n						
Testing phase									1									
ทดสอบและปรับปรุงระบบ	Worapob,Sukrit	1							n									
Closure phase															1 2 3			
ส่ง Software	Worapob,Sukrit	1												n				
จัดทำ Presentation	Worapob,Sukrit	1													n			
ทำการ Presentation	Worapob,Sukrit	2													n	n		
Annotations																		
W	Duration (Weeks)																	
n	normal																	
d	dependent on others																	
c	critical																	

ผลการทดสอบซอฟต์แวร์

Unit test

Test code : Online-Music-Controller / test.py

```
import unittest
from mock import Mock
from module import Playlist, Player, File

class Foo(object):
    def next(self):
        return "next"
    def play_song(self, argValue):
        return "play "+argValue
    def play_pause(self):
        return "pp"
    def previous(self):
        return "prev"
    def logout(self):
        pass

class TestCase(unittest.TestCase):
    def test_get_file_list(self):
        self.playlist = Playlist.Playlist("test","1234")

        mockPL1 = Mock(get_file_name = "a")
        mockPL2 = Mock(get_file_name = "b")
        mockPL3 = Mock(get_file_name = "c")
        mockPL4 = Mock(get_file_name = "d")

        self.playlist.file_list.append(mockPL1)
        self.playlist.file_list.append(mockPL2)
        self.playlist.file_list.append(mockPL3)
        self.playlist.file_list.append(mockPL4)

        test =[mockPL1 ,mockPL2 ,mockPL3 ,mockPL4]

        self.assertEqual(test, self.playlist.get_filelist())

    def test_run_command(self):
        self.player = Player.Player("1234")
        mock_remote = Foo()
        self.player.remote = mock_remote
        self.assertEqual("next", self.player.run_command("next"))
        self.assertEqual("pp", self.player.run_command("play_pause"))
        self.assertEqual("prev", self.player.run_command("previous"))
        self.assertRaises(IndexError, Lambda: self.player.run_command())

if __name__ == '__main__':
    unittest.main()
```

Test result :

```
..
-----
Ran 2 tests in 0.000s

OK
[Finished in 0.1s]
```

Evaluation

✓ จาก M3: Implementation Progress & Test Plan

I. การควบคุมเครื่องเล่น

- จุดประสงค์การทดลอง
 - เพื่อทดสอบการติดต่อกับเครื่องเล่นจริง
 - ทดสอบว่าเครื่องเล่นทำงานตามที่สั่งไปได้อย่างถูกต้องหรือไม่
 - ทดสอบการทำงานของ Server
- สิ่งที่วัด
 - เครื่องเล่นทำงานตามที่สั่งหรือไม่
- วิธีการทดลอง
 - เข้าสู่ระบบหรือสมัครสมาชิกแล้วเข้าสู่ระบบ
 - ทำการทดลองโดยการคลิกที่ปุ่มควบคุมต่าง ๆ (Play/Pause, Next, Previous) หรือการเล่นตามเพลงที่เลือกโดยการใส่ชื่อเพลงในช่องชื่อเพลงแล้วคลิกปุ่ม (Play)
 - ตรวจสอบความถูกต้องของคำสั่งโดยดูจากเครื่องเล่นจริง
- สิ่งที่ต้องใช้ในการทดลอง
 - เครื่องเล่นเพลง (Raspberry PI)
 - Software ที่ใช้ในการเล่นเพลงของเครื่องเล่น (Media Player Daemon(mpd))
 - บัญชีผู้ใช้ที่ใช้งานได้จริง
 - เพลงในเครื่องเล่นเพลง
- ผลที่ได้จากการทดลอง
 - ระบบทำงานได้อย่างถูกต้องแต่การใช้งานยังไม่สะดวกนัก
- สรุปและสิ่งที่ได้จากการทดลองนี้
 - ต้องปรับปรุงส่วนติดต่อผู้ใช้ (User Interface) ให้ใช้งานง่ายขึ้นและเพิ่มความสวยงาม

II. ทดสอบระบบอัปโหลดไฟล์

- จุดประสงค์การทดลอง
 - ทดสอบการทำงานส่วน Upload ไฟล์ว่าไฟล์นั้นไปอยู่ตาม Directory ที่ระบุไว้หรือไม่ และไฟล์ที่ส่งไปสามารถใช้งานได้ถูกต้องจริงหรือไม่
- สิ่งที่วัด
 - ใน Directory ปลายทางมีไฟล์อยู่จริง และสามารถใช้งานได้ปกติ
- วิธีการทดลอง
 - เข้าสู่ระบบหรือสมัครสมาชิกแล้วเข้าสู่ระบบ
 - เลือกไปยังหน้าการจัดการข้อมูลผู้ใช้ผ่าน link “account”
 - กดปุ่ม “Choose File”
 - เลือกไฟล์ที่จะทำการ Upload
 - กดปุ่ม “upload”
- สิ่งที่ต้องใช้ในการทดลอง
 - เครื่องเล่นเพลง (Raspberry PI)
 - บัญชีผู้ใช้ที่ใช้งานได้จริง
 - ไฟล์ในเครื่องผู้ใช้
 - Directory ในเครื่องปลายทาง
- ผลที่ได้จากการทดลอง
 - ใน Directory ปลายทางมีไฟล์อยู่จริง และสามารถใช้งานได้ปกติ แต่ไฟล์ text (py,txt) ไม่สามารถใช้งานได้
- สรุปและสิ่งที่ได้จากการทดลองนี้
 - ต้องเพิ่มส่วนบังคับการ upload ไฟล์ ให้สามารถ upload ได้เฉพาะไฟล์เพลง

✓ ผลการทดลองที่สมบูรณ์

ในส่วนการทดลองครั้งนี้จะมีการเพิ่มการทำงานของ Playlist เข้ามาเกี่ยวข้องด้วย

III. การควบคุมเครื่องเล่น

- จุดประสงค์การทดลอง
 - เพื่อทดสอบการติดต่อกับเครื่องเล่นจริง
 - ทดสอบว่าเครื่องเล่นทำงานตามที่สั่งไปได้อย่างถูกต้องหรือไม่
 - ทดสอบการทำงานของ Server
- สิ่งที่วัด
 - เครื่องเล่นทำงานตามที่สั่งหรือไม่
- วิธีการทดลอง
 - เข้าสู่ระบบหรือสมัครสมาชิกแล้วเข้าสู่ระบบ
 - ทำการทดลองโดยการคลิกที่ปุ่มควบคุมต่าง ๆ (Play/Pause, Next, Previous) หรือการเล่นตามเพลงที่เลือกโดยการเลือกเพลงใน Playlist ที่ UI แล้วคลิกปุ่ม (Play)
 - ตรวจสอบความถูกต้องของคำสั่งโดยดูจากเครื่องเล่นจริง
- สิ่งที่ต้องใช้ในการทดลอง
 - เครื่องเล่นเพลง (Raspberry PI)
 - Software ที่ใช้ในการเล่นเพลงของเครื่องเล่น (Media Player Daemon(mpd))
 - บัญชีผู้ใช้ที่ใช้งานได้จริง
 - Playlist ภายในระบบจริง
 - เพลงในเครื่องเล่นเพลง
- ผลที่ได้จากการทดลอง
 - ระบบทำงานได้อย่างถูกต้อง และมี UI ที่เข้าใจง่ายยิ่งขึ้น
- สรุปและสิ่งที่ได้จากการทดลองนี้
 - พัฒนารูปแบบ UI ที่เข้าใจง่ายยิ่งขึ้น

IV. ทดสอบระบบอัปโหลดไฟล์

- จุดประสงค์การทดลอง
 - ทดสอบการทำงานส่วน Upload ไฟล์ว่าไฟล์นั้นไปอยู่ตาม Directory ที่ระบุไว้หรือไม่ และไฟล์ที่ส่งไปสามารถใช้งานได้ถูกต้องจริงหรือไม่
- สิ่งที่วัด
 - ใน Directory ปลายทางมีไฟล์อยู่จริง และสามารถใช้งานได้ปกติ
- วิธีการทดลอง
 - เข้าสู่ระบบหรือสมัครสมาชิกแล้วเข้าสู่ระบบ
 - เลือกไปยังหน้าการจัดการข้อมูลผู้ใช้งาน link “account”
 - กดปุ่ม “Choose File”
 - เลือกไฟล์ที่จะทำการ upload
 - กดปุ่ม “upload”
- สิ่งที่ต้องใช้ในการทดลอง
 - เครื่องเล่นเพลง (Raspberry PI)
 - บัญชีผู้ใช้ที่ใช้งานได้จริง
 - ไฟล์ในเครื่องผู้ใช้
 - Directory ในเครื่องปลายทาง
- ผลที่ได้จากการทดลอง
 - ใน Directory ปลายทางมีไฟล์อยู่จริง และสามารถใช้งานได้ปกติ และมีการอัปเดตฐานข้อมูลที่ถูกต้อง
- สรุปและสิ่งที่ได้จากการทดลองนี้
 - ต้องเพิ่มส่วนบังคับการ upload ไฟล์ ให้สามารถ upload ได้เฉพาะไฟล์เพลง

บทสรุป

สิ่งที่คิดว่าจะได้รับจากการทำโครงการนี้ กลุ่ม MTTX นั้นเป็นกลุ่มที่สนใจในด้านเกี่ยวกับด้าน Embedded system ตอนที่เริ่มคิดหัวข้อกลุ่ม MTTX ได้มีแนวคิดที่อยากจะทำอุปกรณ์ Gadget ขึ้นมา เพื่อที่จะให้เกิดความสะดวกในชีวิตประจำวัน และจะได้ใช้ความรู้ที่ได้เรียนมาประยุกต์ใช้กับด้าน Embedded system ให้เกิดประโยชน์สูงสุด หนึ่งในสมาชิกของกลุ่มนั้นมีความสนใจในเรื่องของเสียงเพลง คิดว่าถ้ามี เครื่องเล่นเพลงที่ควบคุมทางไกลผ่านอุปกรณ์อื่นๆได้ จะเป็นอะไรที่สะดวกสบายมากขึ้น จึงได้มีการเริ่มที่จะ ทำขึ้น สำหรับโครงการนี้สิ่งที่ทีม MTTX หวังว่าผู้ใช้จะได้รับประสบการณ์ฟังเพลงแบบใหม่ ที่สามารถควบคุม ที่ไหนก็ได้ผ่านระบบที่ไม่ถูกจำกัดโดยระยะทาง และอุปกรณ์ที่จะควบคุม ซึ่งจะทำให้รู้สึกว่าการฟังเพลงนั้นอยู่ รอบตัวเราจริงๆ

ตลอดระยะเวลาที่ทำงานนั้นก็ได้พบปัญหาตลอด อย่างแรกคือด้านการพัฒนาระบบที่ไม่เข้าใจ Domain Logic ต้องมีการออกแบบใหม่บ่อยครั้ง ทำให้เกิดความเข้าใจว่าการจะพัฒนาซอฟต์แวร์ใดๆการ เข้าใจระบบอย่างถ่องแท้จะช่วยลดปัญหาที่จะเกิดขึ้นได้มากๆ ทั้งนี้ความเข้าใจในแนวคิด Object Oriented ก็ เป็นสิ่งสำคัญอย่างมากในการพัฒนาระบบ ทุกครั้งที่ออกแบบระบบไว้ไม่ดี ไม่ชัดเจน มักจะทำให้การทำงาน ถัดไปเป็นไปอย่างยากลำบากและจะต้องกลับมาคิดใหม่เสมอๆว่าทำไมไปมีข้อผิดพลาดอะไร และกลับมา วางแผนใหม่ แต่เมื่อมีการออกแบบที่ดีและชัดเจนแล้วจะทำให้ส่วนนั้นพัฒนาต่อไปได้ง่ายขึ้นรู้ว่าจะต้องทำ อะไรต่อขาดเหลืออะไร ปัญหาต่อมาคือการใช้ framework และภาษาในการพัฒนาไม่เหมือนกลุ่มอื่นๆ ทำให้ ต้องใช้เวลาไปกับการค้นคว้าด้วยตัวเอง ทั้งในหนังสือและอินเทอร์เน็ตจำนวนมาก แต่ก็ช่วยให้รู้สึกว่าการพัฒนา มีความสามารถค้นคว้าด้วยตัวเองที่มากขึ้น ไม่ลำบากเหมือนช่วงแรกๆ จึงได้รู้ว่าตนเองนี้ได้พัฒนาขึ้นบ้างแล้ว และปัญหาสุดท้ายที่ได้เจอคือปัญหาด้านการจัดการตัวเองซึ่งมักจะเป็นปัญหาหลักในทุกๆงาน ที่จะมักใช้เวลา ให้เสียเปล่าไปโดยที่ไม่ได้ทำอะไร จนกระทั่งได้มาเริ่มทำโครงการนี้ ครั้งแรกที่ได้รับโครงการนี้แล้ว ทั้งยังมี งานของวิชาอื่นๆมากมาย ก็เกิดความรู้สึกไม่สบายใจอยู่ตลอดว่า มีงาน และก็มีแบบนั้นเรื่อยๆ เพราะยังไม่ได้ เริ่มทำเลย จนเมื่อได้เริ่มทำแล้ว ต้องคอยบริหารจัดการเวลาอยู่ตลอด ต้องคิดอยู่ตลอดเวลา คิดว่าวันนี้จะทำ อะไร พรุ่งนี้จะทำอะไร ต่อไปจะทำอะไร ที่ทำมาดีแล้วหรือไม่ จนรู้สึกว่าการทำมาก่อนได้มีแนวคิดแบบนี้อะไรๆ ก็น่าจะดีขึ้น หลายๆอย่างเกิดขึ้นจากการเริ่มต้น แต่ทั้งหมดแล้วก็ขึ้นอยู่กับความพยายามและการจัดการตัวเอง ให้เป็นด้วยด้วย

บรรณานุกรม

Introduction to Tornado, 1ST edition O'reilly

Learning python powerful object-oriented programming, 5TH edition O'reilly

<http://getbootstrap.com/>

<http://www.w3schools.com/>

<http://api.jquery.com/>

<http://www.tornadoweb.org/>

<http://www.paramiko.org/>

<https://github.com/bdarnell/torndb>

http://www.tutorialspoint.com/python/python_database_access.htm