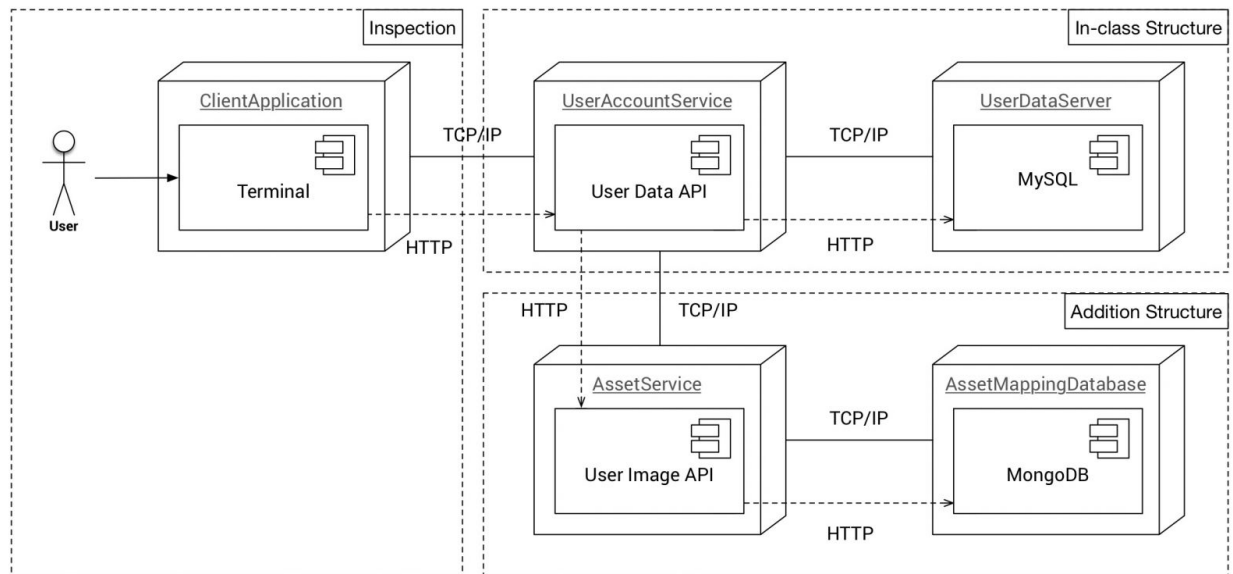


ให้ใช้โค้ดเริ่มต้นโดยโคลนจาก

<https://github.com/2110366-2566-2/DockerAssignment.git>



Containers ด้านบน เป็นโครงสร้างสำหรับ โปรแกรมค้นหาข้อมูลผู้ใช้งาน โดยดัดแปลงและเพิ่มเติมจากที่ได้บรรยายไปในชั้นเรียน ซึ่งการทำงานของโปรแกรมนี้อคือ

1. จะเริ่มต้นใช้งานจาก *ClientApplication* ภายในส่วน *Inspection* ด้วยการส่งชื่อ *username* ที่ต้องการค้นหาให้กับ *UserAccountService* ผ่านการร้องขอข้อมูลแบบ GET ของโปรโตคอล HTTP ซึ่งอยู่ใน *In-class Structure*
2. *In-class Structure* จะเป็นโครงสร้างเช่นเดียวกันกับที่บรรยายในชั้นเรียน โดยที่ *UserAccountService* จะรับข้อมูล *username* เพื่อค้นหาหมายเลขโทรศัพท์ของ *username* ดังที่ระบุ จากฐานข้อมูล MySQL ภายใน *UserDataServer*
3. *UserDataServer* นั้นได้เตรียมข้อมูลตั้งต้นไว้ภายในโฟลเดอร์ *user\_data* แล้วเรียบร้อย
4. นอกจากพฤติกรรมปกติตามที่บรรยายในชั้นเรียนแล้ว ได้ปรับเปลี่ยนให้ *UserAccountService* นั้นส่ง *username* ไปยัง *AssetService* เพื่อนำไปค้นหาข้อมูลรูปภาพที่ตรงกันกับ *username* ที่ต้องการ โดยข้อมูลที่อยู่ใน *profile\_asset.json* ให้ใช้ **mongo** แทน MySQL ข้างต้น

## สิ่งที่นิสิตต้องทำ

### 1. เติมคำสั่งต่างๆ ใน Dockerfiles ของ

- 1) user\_account\_service ใช้ base image ของ node เป็น 17-alpine
- 2) user\_data โดย based image ของ mysql เป็น 8.0
- 3) asset\_service ใช้ base image ของ node เป็น 17-alpine
- 4) asset\_mapping\_seed ใช้ base image mongo เป็น 5.0  
โดย Dockerfile ของ asset\_mapping ได้ทำไว้ให้แล้ว

### 2. Start ระบบทั้งหมดโดยใช้คำสั่ง docker-compose

### 3. เข้าไปที่ docker container ของ user\_data และทำการ แก้ไข setting ของ mysql โดยใช้คำสั่งต่อไปนี้

*(Why this step is required?)*

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '123';
```

```
ALTER USER 'users_service' IDENTIFIED WITH mysql_native_password BY '123';
```

```
flush privileges;
```

### 4. ทดสอบว่า containers ของระบบทำงานได้ถูกต้อง โดยใช้คำสั่งต่อไปนี้

```
$ curl -L http://localhost/?username=alice | json_pp
```

## ข้อมูลส่งออก

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	92	100	92	0	0	7605	0
--:--:-- --:--:-- --:--:-- 9200							
{							
"phoneNumber" : "1234567890",							
"profile_image" : "https://picsum.photos/1000",							
"username" : "alice"							
}							

ถ้าข้อมูลส่งออกได้ผลตามข้างต้น

ให้นำ zip folder และส่งกลับมาใน MyCourseville โดยมี Dockefiles ของ services ต่างๆ และ docker-compose.yml ที่แก้ไขให้สามารถนำมารัน docker-compose แล้วได้ผลตามข้างต้น

---

## Appendix

Asset Mapping Database (ในส่วนของ mongo)

เพื่ออำนวยความสะดวก ได้จัดเตรียม *AssetMappingDatabase* พร้อมทั้งเพิ่มข้อมูลด้านในไว้ให้แล้วเรียบร้อย โดยหลังจากเรียกคำสั่ง docker-compose build หรือ docker-compose up -d แล้ว ถ้าเรียกคำสั่ง

```
$ docker exec -it [container name of asset_mapping] mongo
```

และคำสั่งของ mongo ตามตัวอย่างต่อไปนี้ จะเห็นรายการ userProfile ทั้งหมดในระบบ

```
> use user
> db.userProfile.find()
{ "_id" : ObjectId("59f03cfa6574d1b68c74caf6"), "user_id" :
"5C66F17A-534E-48C2-8352-2631EBC61094", "uname" : "trudy",
"profile_image" : "https://picsum.photos/900?random", "last_update" :
1508821332 }
{ "_id" : ObjectId("59f03cfa6574d1b68c74caf7"), "user_id" :
"610982C8-226A-483B-B597-22B0768C82AF", "uname" : "carol",
"profile_image" : "https://picsum.photos/1000", "last_update" :
1508821332 }
{ "_id" : ObjectId("59f03cfa6574d1b68c74caf8"), "user_id" :
"D4660FDE-180D-4535-8D97-B94BA226B72C", "uname" : "eve",
"profile_image" : "https://picsum.photos/1000", "last_update" :
1508821332 }
{ "_id" : ObjectId("59f03cfa6574d1b68c74caf9"), "user_id" :
"C916EA7C-C9C7-4564-AA38-46F0B1CFB915", "uname" : "bob",
```

```
"profile_image" : "https://picsum.photos/1000", "last_update" :  
1508821332 }  
  
{ "_id" : ObjectId("59f03cfa6574d1b68c74cafa"), "user_id" :  
"0888D506-108D-44BF-B75B-F8406CD9C4D4", "uname" : "alice",  
"profile_image" : "https://picsum.photos/1000", "last_update" :  
1508821332 }  
  
{ "_id" : ObjectId("59f03cfa6574d1b68c74cafb"), "user_id" :  
"BCE16C91-69BD-4FB4-9066-CFE01CEBC8B2", "uname" : "frank",  
"profile_image" : "https://picsum.photos/g/1000?random",  
"last_update" : 1508821332 }  
  
{ "_id" : ObjectId("59f03cfa6574d1b68c74cafe"), "user_id" :  
"A47CC1B8-0B92-48D0-B681-5BAA346008C2", "uname" : "victor",  
"profile_image" : "https://picsum.photos/g/1200?randome",  
"last_update" : 1508821332 }  
  
...
```