

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ**



**BÁO CÁO THỰC TẬP THỰC TẾ  
CÔNG NGHỆ THÔNG TIN  
CT450**

**Giảng viên hướng dẫn**  
TS. Ngô Bá Hùng

**Sinh viên thực tập**  
Nguyễn Văn Vĩ  
MSSV: B1507343

**Cán bộ hướng dẫn thực tập**  
KS. Nguyễn Đăng Nhựt Tân

**Hồ Chí Minh, ngày 13 tháng 7 năm 2019**

## **LỜI CẢM ƠN**

Thực tập thực tế là một học phần rất quan trọng và cần thiết đối với các bạn sinh viên. Bản thân em thật sự rất may mắn khi có cơ hội trở thành thực tập sinh cũng như là nhân viên thử việc tại TMA Solutions. Mặc dù thời gian thực tập tại công ty là rất ngắn nhưng nhờ có sự chỉ bảo tận tình của các anh chị mà em đã học hỏi, tiếp cận được với những công nghệ mới cùng kỹ thuật hiện đại. Em xin chân thành cảm ơn các Thầy Cô của Khoa CNTT & TT đã truyền đạt những kiến thức bổ ích cũng như đã đến thăm, lắng nghe những tâm tư và hướng dẫn em trong thời gian thực tập tại công ty. Trong quá trình thực tập tại công ty, em khó tránh khỏi các sai sót, rất mong Thầy, Cô và các anh chị bỏ qua. Bên cạnh đó, do kiến thức còn hạn hẹp nên bài báo cáo cũng khó tránh khỏi những thiếu sót. Rất mong nhận được ý kiến đóng góp từ anh Tân và các anh chị trong công ty. Cuối lời, em kính chúc anh Tân nhiều sức khỏe và công tác tốt. Kính chúc các anh chị trong công ty TMA Solutions luôn dồi dào sức khỏe, đạt được nhiều thành công trong công việc và cuộc sống. Em xin chân thành cảm ơn!

Sinh viên thực tập

**Nguyễn Văn Vĩ**

**PHIẾU ĐÁNH GIÁ BÁO CÁO KẾT QUẢ THỰC TẬP**  
**HỌC KỲ 3 – 2018-2019**  
**(Dùng cho giáo viên chấm báo cáo thực tập)**

M-TT-04

Họ và tên cán bộ chấm báo cáo: TS. Ngô Bá Hùng  
Họ tên sinh viên thực tập: Nguyễn Văn Vĩ  
Mã số SV: B1507343

Nội dung đánh giá	Điểm tối đa	Điểm chấm
<b>I. Hình thức trình bày</b>	<b>1.0</b>	
I.1 Đúng format của khoa (Trang bìa, trang lời cảm ơn, trang đánh giá thực tập của khoa, trang mục lục và các nội dung báo cáo). Sử dụng đúng mã và font tiếng Việt (Unicode Times New Roman, Size 13)	0.5	
I.2 Trình bày mạch lạc, súc tích, không có lỗi chính tả	0.5	
<b>II. Phiếu theo dõi</b>	<b>4.75</b>	
II.1 Có lịch làm việc đầy đủ cho 8 tuần	0.25	
II.2 Số buổi thực tập tại cơ quan trong 1 tuần $\geq 6$ ; ít hơn 6 buổi 0.0 điểm	1.0	
II.3 Hoàn thành tốt kế hoạch công tác ghi trong lịch làm việc. Cách tính điểm = (điểm cộng của cán bộ hướng dẫn/100) x 3.5	3.5	
<b>III. Nội dung thực tập (quyển báo cáo)</b>	<b>4.25</b>	
III.1 Có được sự hiểu biết tốt về cơ quan nơi thực tập	0.5	
III.2 Phương pháp thực hiện phù hợp với nội dung công việc được giao	1.0	
III.3 Kết quả củng cố lý thuyết	0.5	
III.4 Kết quả rèn luyện kỹ năng thực hành	0.5	
III.5 Kinh nghiệm thực tiễn thu nhận được	0.5	
III.6 Kết quả công việc có đóng góp cho cơ quan nơi thực tập	1.25	
<b>TỔNG CỘNG</b>	<b>10.0</b>	
Điểm trừ		
<b>Điểm còn lại</b>		

Lưu ý:

Không dự họp để nghe phổ biến TTTT: trừ 1 điểm

Không gửi phiếu giao việc về khoa đúng hạn (đến 17.07.2019 theo dấu bưu điện): trừ 1 điểm

....., ngày....tháng....năm.....

GIÁO VIÊN CHẤM BÁO CÁO

(*ký tên*)

## MỤC LỤC

<b>CHƯƠNG I: KẾT QUẢ TÌM HIỂU VỀ TỔ CHỨC PHÒNG QUẢN LÝ KHOA HỌC, TRƯỜNG ĐẠI HỌC CẦN THƠ</b>	1
Báo cáo kết quả tìm hiểu tổ chức hành chính, nhân sự cơ quan nơi thực tập.	1
Báo cáo kết quả tìm hiểu về các hoạt động chuyên ngành và môi trường làm việc tại cơ quan thực tập.	1
<b>CHƯƠNG II: NỘI DUNG CÔNG VIỆC ĐƯỢC GIAO</b>	3
Nội dung công việc được phân công	3
Những công việc đã làm được	3
Phương pháp thực hiện	3
<b>CHƯƠNG III: KẾT QUẢ QUA ĐỢT THỰC TẬP</b>	10
Java Core	10
Design Pattern	10
Spring Core/WebMVC/AOP + Spring boot	10
Hibernate	10
Postgresql	10
RabbitMQ	11
Docker engine / Container	11
Kinh nghiệm thực tiễn đã tích lũy	11

## **CHƯƠNG I: KẾT QUẢ TÌM HIỂU VỀ TỔ CHỨC PHÒNG QUẢN LÝ KHOA HỌC, TRƯỜNG ĐẠI HỌC CẦN THƠ**

### **1. Báo cáo kết quả tìm hiểu tổ chức hành chính, nhân sự cơ quan nơi thực tập.**

- TMA – Hình thành và phát triển:
  - TMA Solutions( gọi tắt là TMA) : Công ty TNHH Giải pháp phần mềm Tường Minh là một công ty tư nhân, 100% vốn Việt Nam, kinh doanh các dịch vụ liên quan đến phát triển phần mềm.
  - Được thành lập vào 03/1997 bởi bà Bùi Ngọc Anh, lúc bấy giờ công ty chỉ mới có 6 kỹ sư. Sau hơn 20 năm thành lập và phát triển vững mạnh, số lượng kỹ sư CNTT đang làm việc tại công ty hiện nay đã hơn 2000 nhân viên.
  - Hiện nay TMA đang có 6 lab tại TP.HCM và 1 lab tại Bình Định đã đi vào hoạt động từ ngày 11/6/2018. Ngoài ra TMA còn có 5 chi nhánh ở nước ngoài: Nhật Bản, Canada, Mỹ, Châu Âu và Úc
  - TMA hiện đang là công ty phần mềm hàng đầu Đông Nam Á và hiện là đối tác tin cậy có nhiều năm kinh nghiệm thực hiện các dự án lớn và phức tạp với hơn 100 khách hàng đến từ hơn 25 quốc gia trên toàn thế giới.
  - Lĩnh vực hoạt động của TMA hiện nay là: Tích hợp hệ thống. Xuất khẩu phần mềm. Giải pháp phần mềm. Cung cấp các giải pháp dịch vụ viễn thông. Đào tạo sinh viên và nhân viên về các kiến thức phần mềm và kỹ năng mềm.
- Sơ đồ tổ chức TMA Solutions:
  - 4 DG được chia thành 17 DC.
  - Hội đồng quản trị (BOD).
  - Business & Shared Services.
  - CDF.
  - Delivery Shared Services.
  - Tài chính và Pháp lý.
  - HR/ Admin Support.
  - TMA Overseas.

### **2. Báo cáo kết quả tìm hiểu về các hoạt động chuyên ngành và môi trường làm việc tại cơ quan thực tập.**

- Các hoạt động chuyên ngành:
  - Phát triển phần mềm.
  - Kiểm thử phần mềm.
  - Porting & Chuyển đổi.
  - Hỗ trợ sản xuất.
  - Dịch vụ quản lý CNTT.
  - Thiết kế trực quan.

Cung cấp giải pháp cho nhiều lĩnh vực:

- Viễn thông.
- Tài chính và bảo hiểm.
- Thương mại điện tử, bán lẻ và phân phối.

- Nhà cung cấp phần mềm độc lập.
- Phương tiện truyền thông và xuất bản.
- Chăm sóc sức khỏe.
- Giáo dục.
- Khách sạn và du lịch.
- Hậu cần và giao thông
- Môi trường làm việc tại TMA Solutions
  - Thời gian làm việc: Từ thứ 2 đến thứ 6, buổi sáng từ 8h30 đến 12h, chiều từ 13h30 đến 18h.
  - Môi trường làm việc tại TMA thân thiện và chuyên nghiệp không chỉ ở cơ sở vật chất mà còn thể hiện qua chính sách hỗ trợ tài chính.
  - Ngay khi gia nhập vào TMA, các bạn sinh viên đều được giới thiệu cụ thể về công ty, các quy định cũng như vị trí làm việc. Trong quá trình thực tập kết hợp thử việc, em được training về kỹ năng làm việc, tham gia CLB Toastmaster để bổ sung kỹ năng tiếng Anh cần thiết khi làm việc tại TMA.

## CHƯƠNG II: NỘI DUNG CÔNG VIỆC ĐƯỢC GIAO

### 1. Nội dung công việc được phân công

Tuần	Thời gian	Công việc
1	20/05/2019 - 24/05/2019	Java Basic
2	27/05/2019 - 31/05/2019	Some design patterns: Singleton, Abstract factory, Factory method, Builder, Decorator, Observer, Visitor
3	03/06/2019 - 07/06/2019	Spring Core/WebMVC/AOP + Spring boot
4	10/06/2019 - 14/06/2019	Hibernate
5	17/06/2019 - 21/06/2019	Postgresql
6	24/06/2019 - 28/06/2019	RabbitMQ
7	01/07/2019 - 05/07/2019	Docker engine / Container
8	08/07/2019 - 12/07/2019	Learning logic of project

### 2. Những công việc đã làm được

- Ôn tập một số nội dung cơ bản về Java Basic.
- Tìm hiểu và áp dụng thử một số "design pattern".
- Tìm hiểu về các khái niệm trong Spring Core/WebMVC/AOP + Spring boot.
- Thao tác và xử lý dữ liệu thông qua Hibernate.
- Kết nối cơ sở dữ liệu của Postgresql.
- Tìm hiểu và sử dụng RabbitMQ để gửi và nhận messages.
- Tìm hiểu về docker engine và docker-container.
- Thử tìm hiểu về dự án của người hướng dẫn.

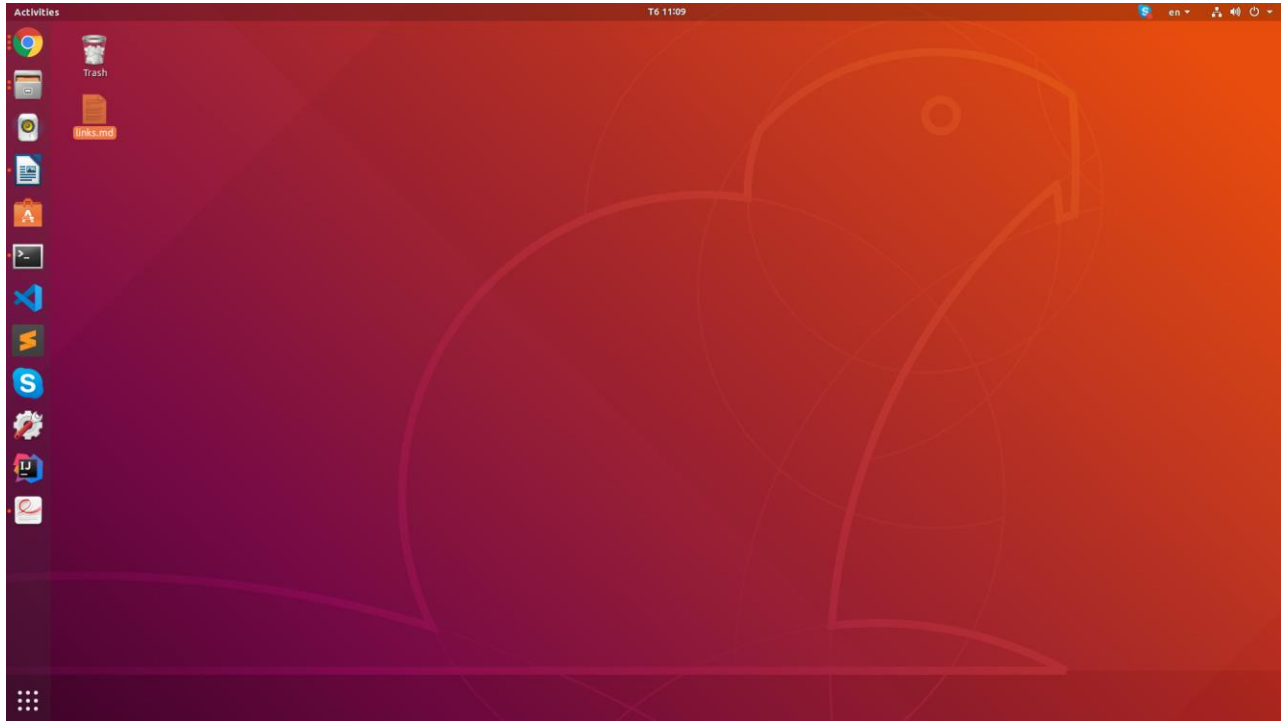
### 3. Phương pháp thực hiện

- Cài đặt môi trường:

- Cài đặt ubuntu, config proxy, authentication, update, upgrade
  - Ubuntu là một hệ điều hành máy tính dựa trên Debian GNU/Linux, một bản phân phối Linux thông dụng. Tên của nó bắt nguồn từ "ubuntu" trong tiếng Zulu, có nghĩa là "tình người", mô tả triết lý ubuntu: "Tôi được là chính mình nhờ có những người xung quanh," một khía cạnh tích cực của cộng đồng. Mục đích của Ubuntu bao gồm việc cung cấp một hệ điều hành ổn định, cập nhật cho người dùng thường, và tập

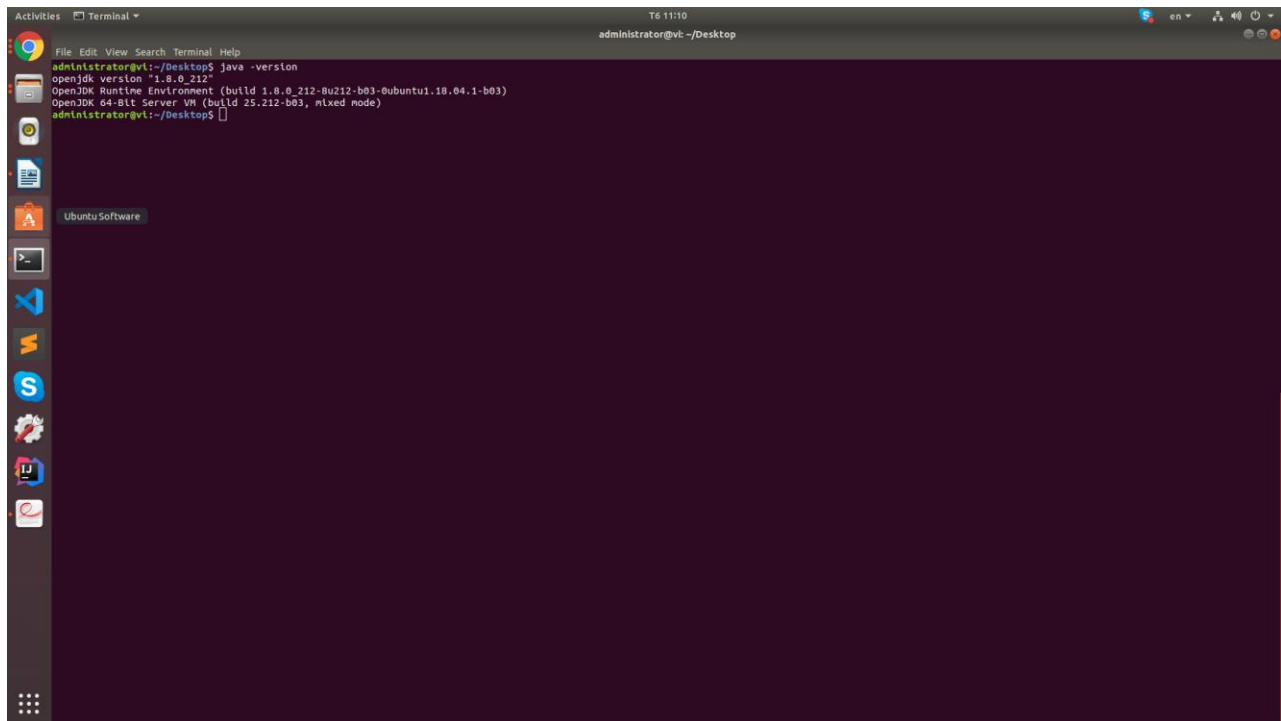


trung vào sự tiện dụng và dễ dàng cài đặt. Ubuntu đã được đánh xếp hạng là bản phân phối Linux thông dụng nhất cho máy tính để bàn, chiếm khoảng 30% số bản Linux được cài đặt trên máy tính để bàn năm 2007.



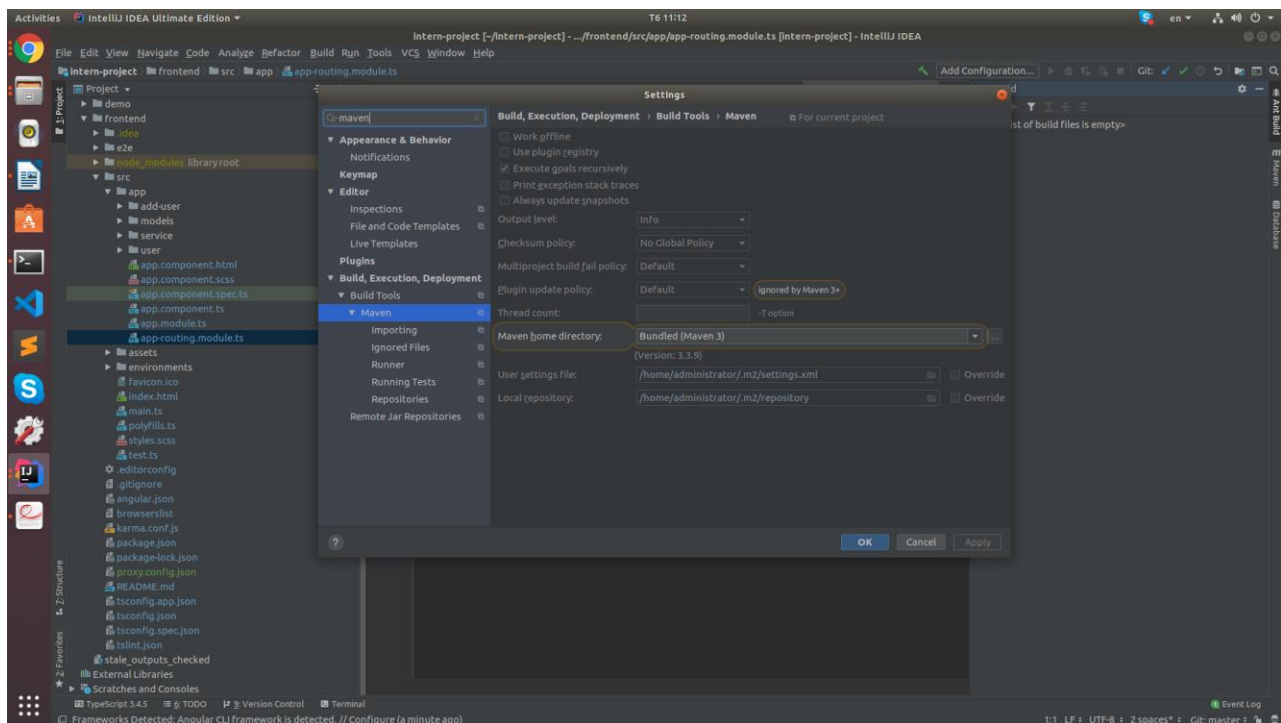
Hình 1. Hệ điều hành Ubuntu được sử dụng trong quá trình thực tập

- Cài đặt jdk
  - Java Development Kit (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Máy ảo Java-Java Virtual Machine) và JRE (Java Runtime Environment - Môi trường Java Runtime). Việc phân biệt giữa ba công nghệ này, cũng như hiểu được cách chúng kết nối với nhau là rất quan trọng.
  - JVM là thành phần Java nền tảng để chạy các chương trình.
  - JRE là phần on-disk của Java, phân tạo ra JVM.
  - JDK cho phép các developer tạo các chương trình Java, trong đó các chương trình có thể được JVM và JRE xử lý và chạy.



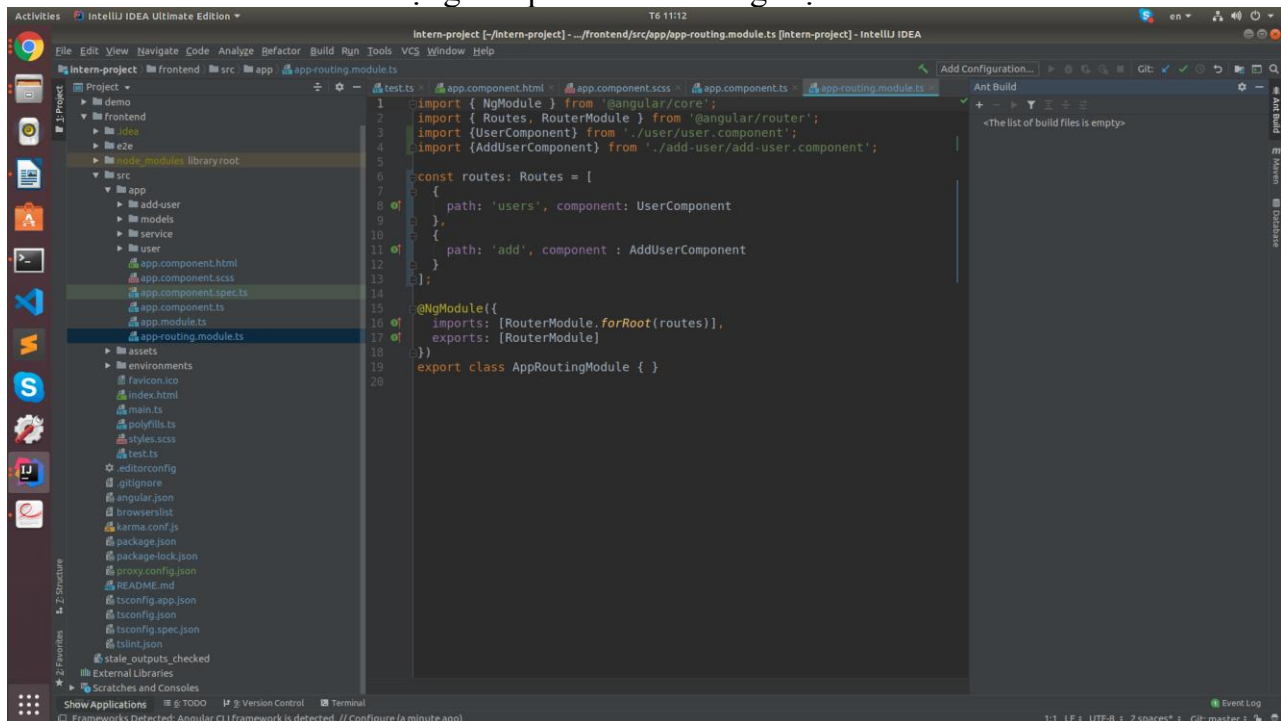
Hình 2. Phiên bản JDK của hệ thống.

- Cài đặt maven
  - Maven là công cụ quản lý và thiết lập tự động 1 dự án phần mềm. Chủ yếu dùng cho các lập trình viên java, nhưng nó cũng có thể được dùng để xây dựng và quản lý các dự án dùng C#, Ruby, Scala hay ngôn ngữ khác.
  - Maven phục vụ mục đích tương tự như Apache Ant, nhưng nó dựa trên khái niệm khác và cách hoạt động khác.
  - Maven hỗ trợ việc tự động hóa các quá trình tạo dự án ban đầu, thực hiện biên dịch, kiểm thử, đóng gói và triển khai sản phẩm.
  - Được phát triển bằng ngôn ngữ Java cho phép Maven chạy trên nhiều nền tảng khác nhau: Windows, Linux và Mac OS...



Hình 3. Maven được tích hợp sẵn trong trình IDE IntelliJ.

- Cài đặt IDE IntelliJ IDE
  - IntelliJ IDEA là một IDE Java để phát triển các phần mềm máy tính. Nó được phát triển bởi JetBrains (trước đây gọi là IntelliJ), nó được cấp phép Apache 2 cho phiên bản cộng đồng, và một phiên bản thương mại độc quyền. Cả hai có thể được sử dụng cho phát triển thương mại.



Hình 4. IntelliJ IDE

- Cài đặt PostgreSQL

- PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.
- PostgreSQL được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.
- PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

```

administrator@vt:~$ cd -Desktop
bash: cd: -Desktop: No such file or directory
administrator@vt:~$ cd Desktop
administrator@vt:~/Desktop$ sudo psql -h localhost -U snt -d snt_dev
[sudo] password for administrator:
Password for user snt:
psql (11.4 (Ubuntu 11.4-1.pgdg18.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

snt_dev=> \d
      Large objects
ID | Owner | Description
-----
(0 rows)

snt_dev=> \dt
snt_dev=> \dt+
snt_dev=> \dr
Invalid command \dr. Try \? for help.
snt_dev=> \r
Query buffer reset (cleared).
snt_dev=> \l

      List of databases
Name | Owner | Encoding | Collate | Ctype | Access privileges
-----
demo | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
snt_dev | snt | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =Tc/snt,+
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | snt=Ctc/snt,+
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres,+
         |         |         |         |         | postgres=Ctc/postgres,+
(5 rows)

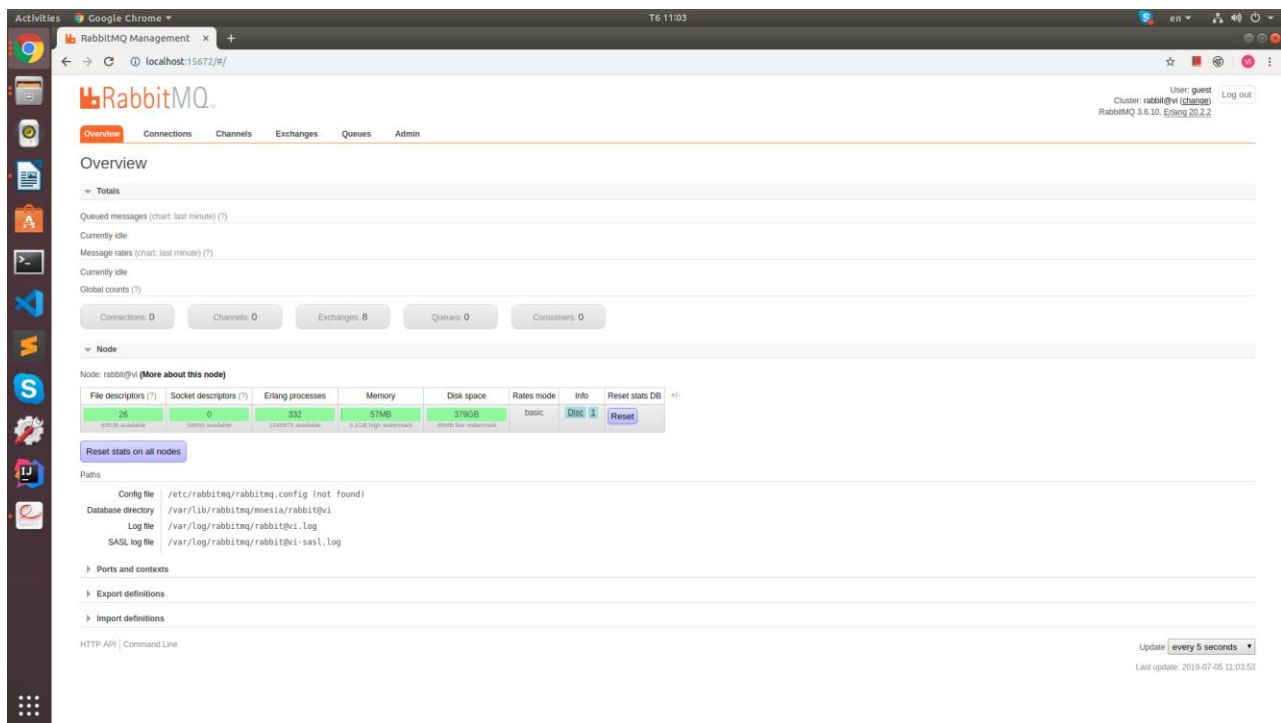
snt_dev=>

```

Hình 5. Các database có trong hệ quản trị cơ sở dữ liệu PostGreSQL

- Cài đặt RabbitMQ

- RabbitMQ là một message broker ( message-oriented middleware) sử dụng giao thức AMQP - Advanced Message Queue Protocol (Đây là giao thức phổ biến, thực tế rabbitmq hỗ trợ nhiều giao thức). RabbitMQ được lập trình bằng ngôn ngữ Erlang. RabbitMQ cung cấp cho lập trình viên một phương tiện trung gian để giao tiếp giữa nhiều thành phần trong một hệ thống lớn ( Ví dụ openstack - Một công nghệ rất thú vị hi vọng một ngày nào đó tôi đủ sức để viết vài bài về chủ đề này ). RabbitMQ sẽ nhận message đến từ các thành phần khác nhau trong hệ thống, lưu trữ chúng an toàn trước khi đẩy đến đích.



Hình 6. Giao diện tương tác với hệ thống quản lý message của RabbitMQ

- Cài đặt Bower

- Bower là một công cụ quản lý resource cho lập trình front-end được open source bởi Twitter. Đã qua cái thời bạn phải download từng thư viện CSS hay JS về bỏ vào project của bạn, rồi nâng cấp version bằng tay khi down đi down lại version mới. Bower giúp tìm kiếm, cài đặt, nâng cấp và cố định dependency cho một front-end project.

- Cài đặt Nodejs

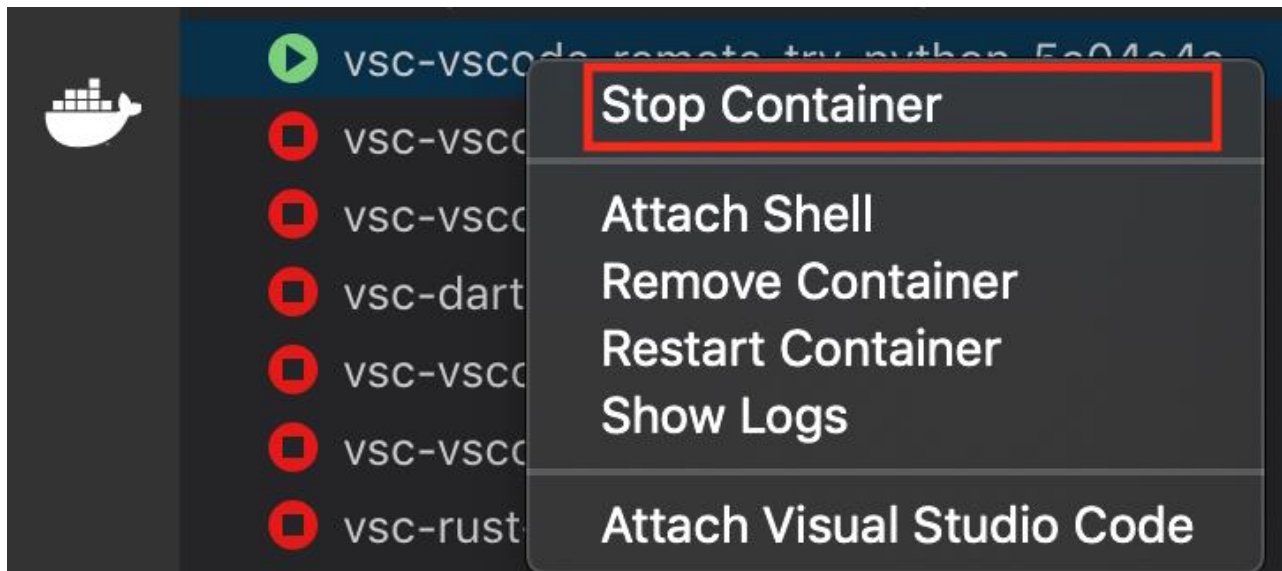
- Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường thì bạn cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì; Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ - vậy thì không có lý do gì để nói nó không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được.

- Cài đặt Docker, Docker compose

- Docker là một nền tảng để cung cấp cách dễ building, deploying và running ứng dụng một cách dễ dàng trên nền tảng ảo hóa. Có nhiều công nghệ Container khác như Solaris Zones, BSD jails, và LXC. Nhưng tại sao Docker lại phát triển, phổ biến nhanh chóng?
- Ease of use: Docker rất dễ cho mọi người sử dụng từ developers, systems admins, architects v.v. nó tận dụng lợi thế của container để build, test nhanh chóng. Có thể đóng gói ứng

dùng trên laptop của họ và chạy trên public cloud, private cloud..v.v... Câu thần chú là “Build once, run anywhere”.

- Speed: Docker container rất nhẹ và nhanh, bạn có thể tạo và chạy docker container trong vài giây so sánh với VMs thì mỗi lần chạy VMs cần rất nhiều thời gian khởi động.
- DockerHub: là một “app store for docker images”. trên DockerHub có hàng ngàn public images được tạo bởi cộng đồng. Dễ dàng tìm thấy những image mà bạn cần và chỉ cần pull về và sử dụng với một số sửa đổi nhỏ.
- Modularity and Scalability: Bạn có thể chia nhỏ những chức năng của ứng dụng thành các container riêng lẻ. Ví dụ ứng dụng Database chạy trên một container và Redis cache có thể chạy trên một container khác trong khi ứng dụng Node.js lại chạy trên một cái khác nữa. Với Docker, rất dễ để liên kết các container với nhau để tạo thành một ứng dụng, làm cho nó dễ dàng scale, update các thành phần độc lập với nhau.



Hình 7. Plugin docker liệt kê danh sách các container trên visual studio code.

-Phương pháp tìm hiểu và học tập:

- Tìm hiểu các tài liệu học tập thông qua internet.
- Thực tập một số ví dụ mẫu có sẵn qua internet.
- Trao đổi và chia sẻ kỹ năng với người hướng dẫn.
- Giúp đỡ bạn cùng thực tập.

## CHƯƠNG III: KẾT QUẢ QUA ĐỢT THỰC TẬP

### 1. Java Core

Với câu khẩu hiệu là, "viết một lần, chạy mọi nơi", cũng không có gì khó hiểu khi Java đã trở thành ngôn ngữ lập trình phổ biến nhất được sử dụng bởi các lập trình viên giỏi, những người muốn dành nhiều thời gian của mình để thư giãn trong các quán cà phê hơn là phải ngồi hí hục viết những dòng code lập đi lập lại một cách tẻ nhạt.

### 2. Design Pattern

Đối với các dự án phần mềm các lập trình viên còn thiếu thức về chất lượng dòng lệnh của mình nên khi thực hiện coding trên các dự án lớn họ thường gặp các vấn đề mở rộng dự án cũng như bảo trì phần mềm. Điều này làm ảnh hưởng rất nghiêm trọng về tương lai lâu dài của họ sau này vì thế mà thông qua design pattern mà em hiểu ra được tầm quan trọng và giá trị của nó trong các dự án có quy mô lớn.

### 3. Spring Core/WebMVC/AOP + Spring boot

Các ứng dụng web phần lớn hiện nay có sử dụng java làm nền tảng để phát triển ứng dụng web cũng như xây dựng API, việc làm này đòi hỏi rất nhiều các thư viện và cấu hình phức tạp cũng như quá trình tổ chức các thư mục vì thế mà Spring Boot là một framework tốt và ổn định thực hiện các thao tác trên một cách tự động giúp lập trình viên chú tâm hơn vào quá trình phát triển phần mềm mà không cần quá bận tâm vào quá trình xây dựng cấu hình cũng như tổ chức thư mục cho hệ thống.

### 4. Hibernate

Với JDBC ta phải map các trường trong bảng với các thuộc tính của Java object một cách “thủ công”. Với Hibernate sẽ hỗ trợ ta map một cách “tự động” thông qua các file cấu hình map XML hay sử dụng các annotation.

Code sử dụng Hibernate là độc lập với hệ quản trị cơ sở dữ liệu, nghĩa là ta không cần thay đổi câu lệnh HQL khi ta chuyển từ hệ quản trị CSDL MySQL sang Oracle, hay các hệ quản trị CSDL khác... Do đó rất dễ để ta thay đổi CSDL quan hệ, đơn giản bằng cách thay đổi thông tin cấu hình hệ quản trị CSDL trong file cấu hình.

Với những ứng dụng Java làm việc với cơ sở dữ liệu lớn hàng trăm triệu bản ghi, việc có sử dụng Lazy loading trong truy xuất dữ liệu từ database mang lại lợi ích rất lớn. Nó giống như việc ta có thể bỏ từng chiếc đĩa của bộ đĩa to thay vì bỏ cả bộ đĩa.

### 5. Postgresql

Về mô hình dữ liệu:

- PostgreSQL không chỉ là cơ sở dữ liệu quan hệ, nó là quan hệ hướng đối tượng. Điều này cung cấp cho nó một vài lợi thế so với các cơ sở dữ liệu SQL mã nguồn mở khác như MySQL, MariaDB và Firebird.

Các kiểu dữ liệu và cấu trúc (đặc biệt hỗ trợ một số kiểu dữ liệu):

- Địa chỉ mạng
- Mạng nhiều chiều
- Dữ liệu hình học
- Hỗ trợ JSON
- Tạo một kiểu định nghĩa riêng

Tính toàn vẹn dữ liệu:

- PostgreSQL phân đầu để phù hợp với chuẩn ANSI-SQL:2008, tuân thủ đầy đủ ACID (Atomicity, Consistency, Isolation và Durability), và cũng được biết đến các tham chiếu vững chắc của nó (rock-solid referential) và toàn vẹn giao dịch (transactional integrity). Các khóa chính, hạn chế và các ràng buộc ngoài, các ràng buộc duy nhất, các ràng buộc not null, kiểm tra các ràng buộc và các tính năng toàn vẹn dữ liệu khác để chắc chắn chỉ dữ liệu hợp lệ mới được lưu trữ.

## 6. RabbitMQ

- Thực sự, với lập trình viên thì rabbitmq rất đáng giá. Nếu không có các hệ thống message broker như rabbitmq thì bất cứ lúc nào cần đẩy data giữa các thành phần trong hệ thống, lập trình viên cần một kết nối trực tiếp. Một hệ thống càng lớn. Số thành phần càng nhiều, mức độ trao đổi message giữa các thành phần cũng vì thế tăng lên khiến việc lập trình trở nên phức tạp. Tôi từng đọc vài bài báo về lập trình thì thấy họ khuyến cáo các lập trình viên chỉ nên tập trung vào business logic của ứng dụng còn các công tác hậu trường thì nên được tái sử dụng các giải pháp đã có. Rabbitmq cũng là một giải pháp rất tốt trong các kiến trúc hệ thống lớn.

## 7. Docker engine / Container

- Ảo hóa các hệ thống mà không cần phải thay đổi cấu hình cài đặt hoặc sử dụng một hệ điều hành ảo của máy tính ảo giảm thiểu được chi phí trong quá trình phát triển phần mềm.
- Container hay Docker Container là một. Hoạt động giống như một thư mục (directory), chứa tất cả những thứ cần thiết để một ứng dụng có thể chạy được. Mỗi một Docker Container được tạo ra từ một Docker Image. Các thao tác với một container: chạy, bật, dừng, di chuyển, và xóa. Không bị các yếu tố liên quan đến môi trường hệ thống làm ảnh hưởng tới, cũng như không làm ảnh hưởng tới các phần còn lại của hệ thống.
- Cách thức hoạt động: Ví dụ 1 ứng dụng web được developer A nào đó đóng gói bằng Docker Engine, cái đóng gói được gọi là Docker Image. Trong Docker Image chứa Ubuntu đã cài đặt sẵn apache và ứng dụng web. Docker Image được đưa lên Docker Hub. Sau đó Docker Image được chia sẻ cho các developer B,C,D,E. Từ Docker Image, developer B,C,D,E có thể tạo ra Docker Container, run Docker Container này là có thể thao tác với ứng dụng web mà không cần phải cài đặt môi trường hay các phần mềm liên quan.

## 8. Kinh nghiệm thực tiễn đã tích lũy

Thông qua đợt thực tập lần này, em đã rút ra rất nhiều kinh nghiệm thực tiễn cho bản thân và em nhận thấy rằng việc tự học thông qua tài liệu, tự sắp xếp các nguồn thông tin tìm được, tích lũy thêm kinh nghiệm về giao tiếp nơi công sở.