|  |  |  |
| --- | --- | --- |
| **Tuần** | **Nội dung** | **Ghi chú** |
| 1 | Nhận đề tài |  |
| 2 | Tìm hiểu công nghệ điện toán đám mây |  |
| 3 | Tìm hiểu gói thư viện Google App Engine |  |
| 4-6 | Cài đặt gói thư viện |  |
| 6-10 | Viết ứng dụng demo |  |
| 10-12 | Triển khai hiện thực chương trình |  |
| 13 | Báo cáo kết thúc. Kiểm tra và đánh giá |  |

**ĐỒ ÁN 3**

**CHỦ ĐỀ: NGHIÊN CỨU CÔNG NGHỆ ĐIỆN TOÁN ĐÁM MÂY. VIẾT MỘT ỨNG DỤNG DEMO DÙNG GÓI THƯ VIỆN GOOGLE APP ENGINE CỦA GOOGLE**

**Họ tên: LÊ MINH NHẬT MSSV: 17004126**

**Họ tên: NGUYỄN MINH TIẾN MSSV: 17004189**

**Giáo viên hướng dẫn: NGUYỄN THỊ HỒNG YẾN– Khoa CNTT**

**LỜI CẢM ƠN**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép chúng em được bày tỏ lòng biết ơn đến cô Nguyễn Thị Hồng Yến (giáo viên hướng dẫn) đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay,chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến y Cô Nguyễn Thị Hồng Yến ở Khoa CNTT đã truyền đạt vốn kiến thức quý báu cho chúng chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của chúng em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, em xin chân thành cảm ơn cô – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn chúng em hoàn thành tốt bài báo cáo này trong thời gian qua.

Bài báo cáo thực tập thực hiện trong khoảng thời gian gần 6 tuần. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót , cjusng em rất mong nhận được những ý kiến đóng góp quý báu của Cô để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

Chúng em xin chân thành cảm ơn!

**CHƯƠNG I: NGHIÊN CỨU CÔNG NGHỆ ĐIỆN TOÁN ĐÁM MÂY**

**1.Khái niệm**

**Điện toán đám mây** ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh" \o "Tiếng Anh): *cloud computing*), còn gọi là **điện toán máy chủ ảo**, là mô hình [điện toán](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_to%C3%A1n" \o "Điện toán) sử dụng các công nghệ [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh" \o "Máy tính) và phát triển dựa vào mạng [Internet](https://vi.wikipedia.org/wiki/Internet). Thuật ngữ "đám mây" ở đây là lối nói ẩn dụ chỉ mạng Internet (dựa vào cách được bố trí của nó trong sơ đồ mạng máy tính) và như sự liên tưởng về độ phức tạp của các cơ sở hạ tầng chứa trong nó. Ở mô hình điện toán này, mọi khả năng liên quan đến [công nghệ thông tin](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_th%C3%B4ng_tin" \o "Công nghệ thông tin) đều được cung cấp dưới dạng các "dịch vụ", cho phép người sử dụng truy cập các dịch vụ công nghệ từ một nhà cung cấp nào đó "trong đám mây" mà không cần phải có các kiến thức, kinh nghiệm về công nghệ đó, cũng như không cần quan tâm đến các cơ sở hạ tầng phục vụ công nghệ đó. Theo tổ chức [IEEE](https://vi.wikipedia.org/wiki/IEEE) "*Nó là hình mẫu trong đó thông tin được lưu trữ thường trực tại các máy chủ trên Internet và chỉ được được lưu trữ tạm thời ở các máy khách, bao gồm [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n" \o "Máy tính cá nhân), trung tâm giải trí, máy tính trong doanh nghiệp, các phương tiện máy tính cầm tay,...*". Điện toán đám mây là khái niệm tổng thể bao gồm cả các khái niệm như [phần mềm dịch vụ](https://vi.wikipedia.org/wiki/SaaS" \o "SaaS), [Web 2.0](https://vi.wikipedia.org/wiki/Web_2.0) và các vấn đề khác xuất hiện gần đây, các xu hướng công nghệ nổi bật, trong đó đề tài chủ yếu của nó là vấn đề dựa vào Internet để đáp ứng những nhu cầu điện toán của người dùng. Ví dụ, dịch vụ [Google AppEngine](https://vi.wikipedia.org/wiki/Google_AppEngine) cung cấp những ứng dụng kinh doanh trực tuyến thông thường, có thể truy nhập từ một [trình duyệt web](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web" \o "Trình duyệt web), còn các [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m" \o "Phần mềm) và [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u" \o "Dữ liệu) đều được lưu trữ trên các máy chủ.

**2.Lợi ích của cloud computing**

**- Tính linh động:** Người dung có thể thoải mái lựa chọn các dịch vụ phù hợp với nhu cầu của mình, cũng như có thể bỏ bớt những thành phần mà mình không muốn.(Thay vì phải ra hằng trăm USD cho 1 bộ MS office, ta có thể mua riêng lẻ từng phần hoặc chi trả 1 khoản phí rất nhỏ mỗi khi sử dụng 1 phần nào đó của nó)

**- Giảm bớt phí:** Người dung không chỉ giảm bớt chi phí bản quyền mà còn giảm phần lớn chi phí cho việc mua và bảo dưỡng máy chủ. Việc tập hợp ứng dụng của nhiều tổ chức lại 1 chỗ sẽ giúp giảm chi phí đầu tư ban đầu, cũng như tăng hiệu năng sử dụng các thiết bị này một cách tối đa.

**-** **Tạo nên sự độc lập:** Người dùng sẽ không còn bị bó hẹp với 1 thiết bị hay 1 vị trí cụ thể nào nữa. Với điện toán đám mây, phần mềm, dữ liệu có thể được truy cập và sử dụng từ bất kì đâu, trên bất kì thiết bị nào mà không cần phải quan tâm đến giới hạn phần cứng cũng như địa lý. (Bạn có thể chơi Call of Duty 6 trên iPad hoặc iPhone mà không cần quan tâm đến cấu hình của nó)

**-** **Tăng cường độ tin cậy:** Dữ liệu trong mô hình điện toán đám mây được lưu trữ một cách phân tán tại nhiều cụm máy chủ tại nhiều vị trí khác nhau. Điều này giúp tăng độ tin cậy, độ an toàn của dữ liệu mỗi khi có sự cố hoặc thảm họa xảy ra. (Hãy tưởng tượng 1 ngày nào đó, server yêu quý của công ty tự nhiên bốc cháy với toàn bộ dữ liệu quý giá bên trong, bạn sẽ làm gì?).

**- Bảo mật:** Việc tập trung dữ liệu từ nhiều nguồn khác nhau sẽ giúp các chuyên gia bảo mật tăng cường khả năng bảo vệ dữ liệu của người dùng, cũng như giảm thiểu rủi ro bị ăn cắp toàn bộ dữ liệu. (Dữ liệu được đặt tại 6 máy chủ khác nhau 🡪 trong trường hợp hacker tấn công, bạn cũng sẽ chỉ bị lộ 1/6. Đây là 1 cách chia sẻ rủi ro giữa các tổ chức với nhau)

**-** **Bảo trì dễ dàng:** Mọi phần mềm đều nằm trên server, lúc này, người dùng sẽ không cần lo lắng cập nhật hay sửa lỗi phần mềm nữa. Và các lập trình viên cũng dễ dàng hơn trong việc cài đặt, nâng cấp ứng dụng của mình

**3.Ứng dụng điện toán đám mây là gì?**

**3.1 Cơ sở dữ liệu lớn và khả năng mở rộng vô tận**

Công nghệ điện toán đám mây phù hợp đối với những doanh nghiệp cần vận hành cơ sở dữ liệu lớn nhưng ngân sách hạn chế. Công nghệ này cũng rất hữu ích đối với những doanh nghiệp không đủ chuyên môn để vận hành lượng cơ sở dữ liệu lớn đó. Điện toán máy chủ ảo giúp đội ngũ IT có cơ sở sở liệu mạnh mẽ mà không cần có cơ sở hạ tầng (server). Nhà cung cấp dịch vụ công nghệ điện toán đám mây hỗ trợ và chịu trách nhiệm về các hoạt động bảo trì, vận hành hệ thống dữ liệu. Người dùng có trách nhiệm duy nhất là xử lý dữ liệu của mình. Công nghệ điện toán đám mây giúp doanh nghiệp mở rộng vô tận cơ sở dữ liệu. Nó cũng có khả năng thu nhỏ cơ sở dữ liệu trong khoảng vài giây. Đây là điểm cộng so với cách cài đặt truyền thống, mất vài tuần hoặc vài tháng để cài đặt server, dữ liệu và các thiết bị khác.

**3.2 Kiểm tra và phát triển**

Để đảm bảo ứng dụng của người dùng chạy trơn tru thì việc kiểm tra và phát triển là những bước đặc biệt quan trọng. Để kiểm tra và thử nghiệm thành công rất cần một môi trường mô phỏng. Môi trường này có khả năng tái tạo các hoạt động kinh doanh thực tế và xác nhận những kết quả thu được sau khi mô phỏng. Cloud computing với những nguồn lực sẵn có sẽ giúp người dùng tận dụng được thời gian, công sức khi tự tay xây dựng môi trường mô phỏng. Người dùng sẽ được cung cấp các môi trường khác nhau và phù hợp với từng nhu cầu cụ thể cho mục đích mô phỏng.

**3.3 Lưu trữ website**

Nếu hệ thống hiện tại không đáp ứng được so với sự phát triển không ngừng của doanh nghiệp thì việc lưu trữ website trên đám mây là việc làm cần thiết. Lưu trữ website trên nền tảng đám mây sẽ tăng khả năng mở rộng cho doanh nghiệp. Nếu chẳng may có vấn đề xảy ra, người dùng chỉ cần chuyển website sang server gần nhất. Thanh toán theo nhu cầu thực tế, khả năng bảo mật cao mà công nghệ điện toán đám mây mang lại giúp nó nhận được sự ưu ái của đông đảo người dùng trên toàn cầu.

**3.4 Phân tích Big Data**

Công nghệ điện toán máy chủ ảo có khả năng phân tích dữ liệu lớn (Big Data). Dữ liệu không được thu gọn khi được lưu trữ trên đám mây nhưng việc quản lý nó đơn giản, dễ tiếp cận và hiệu quả hơn.

**3.5 Lưu trữ, chia sẻ dữ liệu**

Dữ liệu được lưu trữ dựa trên công nghệ điện toán đám mây giúp cho việc chia sẻ, truy xuất và lưu trữ thuận tiện hơn rất nhiều. Dropbox, Google Drive hay Shutterstock là những ví dụ cụ thể của dịch vụ này. Công việc sẽ được thúc đẩy nhanh chóng và hiệu quả hơn với các “văn phòng ảo” – nơi mà đồng nghiệp và bạn có thể cập nhật tình hình dự án dễ dàng.

**3.6 Sao lưu, khôi phục dữ liệu**

Công nghệ điện toán đám mây có khả năng sao lưu và khôi phục dữ liệu theo đúng quy trình, tiết kiệm thời gian và chi phí. Sử dụng công nghệ đám mây để phục hồi dữ liệu của doanh nghiệp rất hiệu quả mà tiết kiệm chi phí. Nếu thảm họa tự nhiên xuất hiện, dữ liệu của doanh nghiệp được tách biệt trên đám mây nên có thể tự động khôi phục lại.

**3.7 Quản lý doanh nghiệp**

Cloud CRM bao gồm rất nhiều ứng dụng như Email, thông tin của các hoạt động trong quá khứ và hiện tại,…giúp doanh nghiệp xây dựng, duy trì mối quan hệ tích cực với khách hàng. Hay với Cloud ERP, người dùng có thể theo dõi nguồn lực, thu thập thông tin, cập nhật kế hoạch ngân sách,… Cloud ERP phù hợp với cả doanh nghiệp nhỏ, vừa hay tập đoàn đa quốc gia.

Nói chung, công nghệ điện toán đám mây là công nghệ phân phối tài nguyên công nghệ thông tin theo nhu cầu trong môi trường internet. Với điện toán đám mây, người dùng sẽ không cần mua, bảo trì server hay trung tâm dữ liệu mà có thể tiếp cận dịch vụ công nghệ như năng lượng điện toán, lưu trữ cơ sở dữ liệu,

**CHƯƠNG II: TÌM HIỂU GOOGLE APP ENGINE**

**1.Khái niệm:**

**Google App Engine** (gọi tắt là **GAE** hay **App Engine**) là một môi trường phát triển ứng dụng dựa trên công nghệ [điện toán đám mây](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_to%C3%A1n_%C4%91%C3%A1m_m%C3%A2y" \o "Điện toán đám mây). Ở đó [Google](https://vi.wikipedia.org/wiki/Google) cung cấp một hệ thống gồm: [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình), hệ [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u" \o "Cơ sở dữ liệu), các [thư viện lập trình](https://vi.wikipedia.org/w/index.php?title=Th%C6%B0_vi%E1%BB%87n_l%E1%BA%ADp_tr%C3%ACnh&action=edit&redlink=1" \o "Thư viện lập trình (trang chưa được viết)), người lập trình sẽ viết ứng dụng và ứng dụng này sẽ chạy trên các máy chủ của Google.

Google App Engine được cung cấp miến phí cho mỗi ứng dụng trong một giới hạn, khi các ứng dụng vượt qua mức quá hạn mức thì phải trả thêm khoản phí cho mức vượt quá này.

**2.Ưu điểm:**

- GAE cho phép được host miễn phí với dung lượng 500 MB lưu trữ và cho phép 10 GB băng thông lưu chuyển mỗi ngày hay tương đương 5 triệu pageview hàng tháng, vượt qua mức này bạn sẽ phải trả phí.

- Dùng GAE chúng ta khỏi phải thiết kế database, viết SQL để truy vấn data, map data vo object. Chúng ta chỉ cần design các class và GAE tự động lo phần làm việc với database.

- So với các dịch vụ điện toán đám mây khác như [Amazon EC2](https://vi.wikipedia.org/w/index.php?title=Amazon_EC2&action=edit&redlink=1), hoặc với các hình thức viết trên máy chủ dịch vụ truyền thống App Engine cung cấp một môi trường thuận lợi hơn để viết đặc biệt cho các ứng dụng có nhu cầu mở rộng cho nhiều người dùng đồng thời.

- Khi các mô hình khác yêu cầu người dùng cài đặt và thiết lập cấu hình thì App Engine chỉ cần một màn hình quản trị để quản trị cũng như mở rộng hay thu nhỏ mức cung ứng dịch vụ.

**3.Khuyết điểm:**

* Các [tệp tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin" \o "Tập tin) khi lưu trữ lên App Engine chỉ có thể đọc-xóa (không sửa chữa).[[3]](https://vi.wikipedia.org/wiki/Google_App_Engine#cite_note-3)
* App Engine chỉ có thể chạy qua các yêu cầu HTTP (các dịch vụ khác có thể chạy lệnh qua [Console terminal](https://vi.wikipedia.org/w/index.php?title=Console_terminal&action=edit&redlink=1))(Các task queue cũng chạy thông qua yêu cầu HTTP).
* Bạn sẽ phụ thuộc hoàn toàn vào các công nghệ của Google và rất khó có thể tách ra thành một ứng dụng độc lập
* Điểm giới hạn là phải viết trên các dịch vụ mà App Engine cung cấp. App Engine giải quyết các khó khăn trong việc phát triển và quản trị hệ thống khi xây dựng một ứng dụng lớn (hàng trăm yêu cầu trên giây)
* Các ứng dụng dùng python có thể sử dụng các thư viện python ngoài các thư viện được cung cấp bởi App Engine nhưng phải là các thư viện viết bằng python [C](https://vi.wikipedia.org/wiki/C_(programming_language)) và [Pyrex](https://vi.wikipedia.org/w/index.php?title=Pyrex_(programming_language)&action=edit&redlink=1) không được hỗ trợ.
* Các ứng dụng java chỉ sử dụng được tập các (The JRE Class White List) các lớp viết từ JRE tiêu chuẩn.[[4]](https://vi.wikipedia.org/wiki/Google_App_Engine#cite_note-google1-4)
* Không hỗ trợ tên miền 'naked' (không có tiền tố www) ví dụ [http://example.com](http://example.com/) sẽ không chạy trên Appengine mà phải là [http://www.example.com](http://www.example.com/).[[5]](https://vi.wikipedia.org/wiki/Google_App_Engine#cite_note-5)
* Datastore không lọc (filter) được hai thuộc tính cùng một lúc ví dụ *age > 4 and age < 10*.
* Chỉ hỗ trợ 2 ngôn ngữ là: Python và Java. Một số ngôn ngữ khác như PHP cũng có thể chạy được nếu cài cùng với bộ chuyển từ PHP sang Java

**4.Tính năng:**

* App Identity (Python, Java, Go)
* Blobstore (Python, Java, Go)
* Capabilities (Python, Java, Go)
* Channel (Python, Java, Go)
* Google Cloud Endpoints (Python, Java, Go)
* Images (Python, Java, Go): xử lý hình ảnh
* Logs (Python, Java, Go): ghi nhật ký
* Mail (Python, Java, Go): dịch vụ gửi thư điện tử
* Memcache (Python, Java, Go): cache
* Multitenancy (Python, Java, Go)
* OAuth (Python, Java, Go): đăng nhập sử dụng [OAth](https://vi.wikipedia.org/w/index.php?title=OAth&action=edit&redlink=1" \o "OAth (trang chưa được viết))
* Prospective Search (Python, Java, Go)
* Search (Python, Java, Go): tìm kiếm
* Task Queues (Python, Java, Go): lập lịch công việc cho máy chủ
* Sockets (Python, Java, Go)
* URL Fetch (Python, Java, Go)
* Users (Python, Java, Go): thông tin người dùng đang đăng nhập
* XMPP (Python, Java, Go)

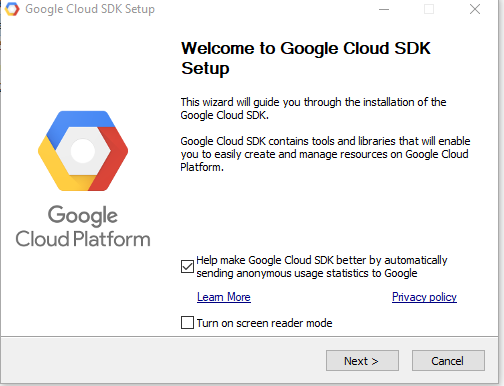
**CHƯƠNG 3: HƯỚNG DẪN CÀI ĐẶT**

**1.Cài đặt Google Cloud SDK**

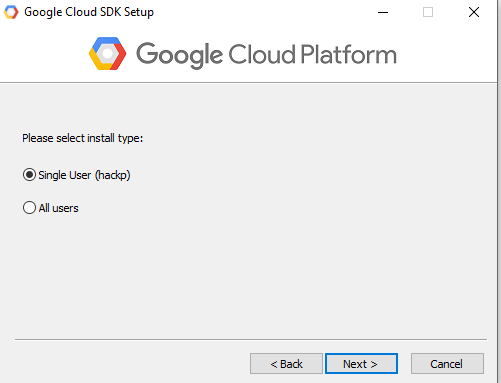
Vào link sau để tải: [**https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe**](https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe)

**Sau khi tải xong lần lượt thực hiện các bước sau:**

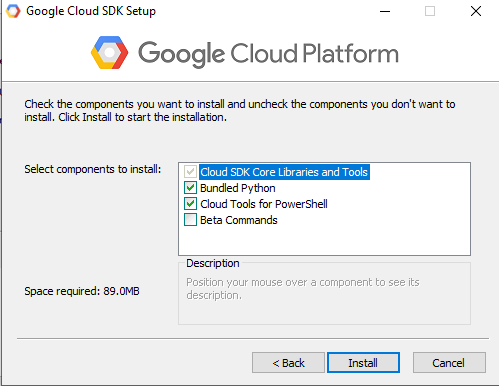
**B1:Check “Help make Google Cloud….” -> Next**



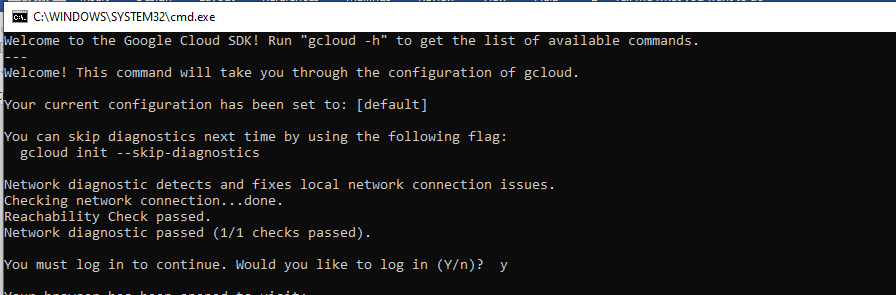
**B2:Next**



**B3:Install**

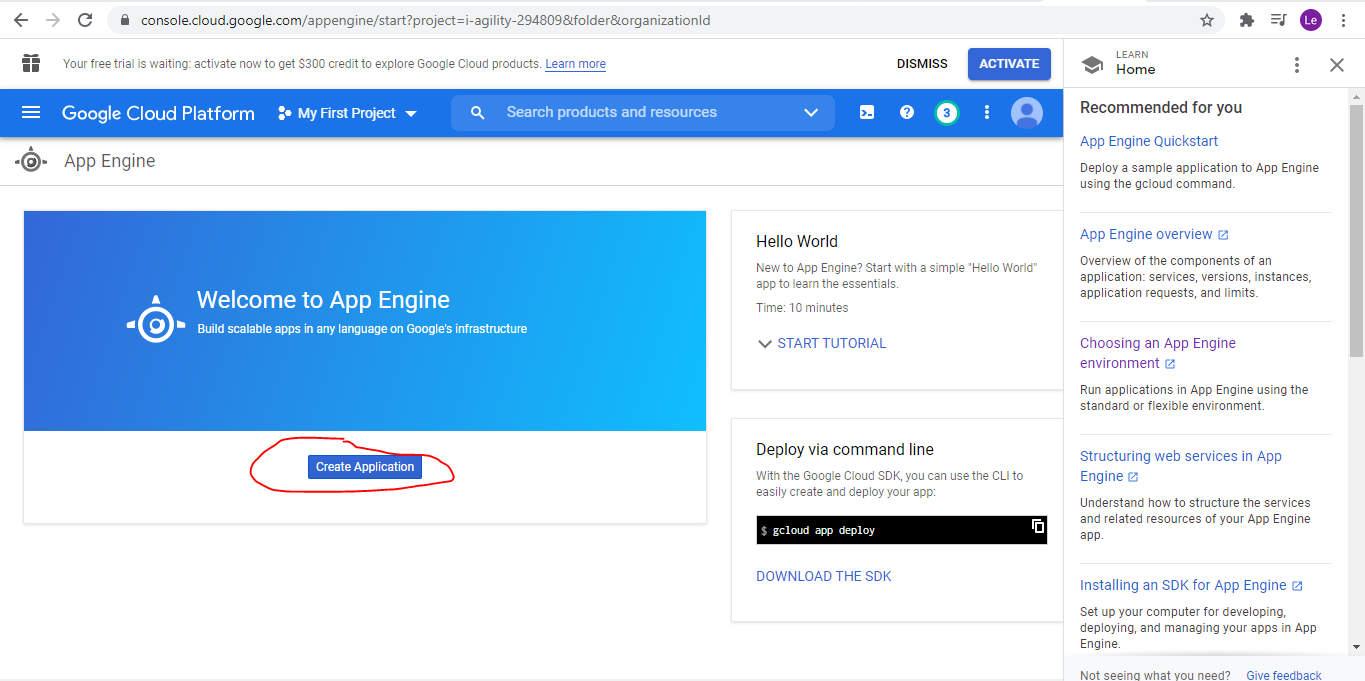


**B4: Sau khi cài đặt xong sẽ hiện ra một cmd để đăng nhập vào tài khoản google 🡪 Gõ “Y” để vào bước đăng nhập**



**B5:Sau khi đăng nhập thành công. Ta mở trình và vào địa chỉ sau:**

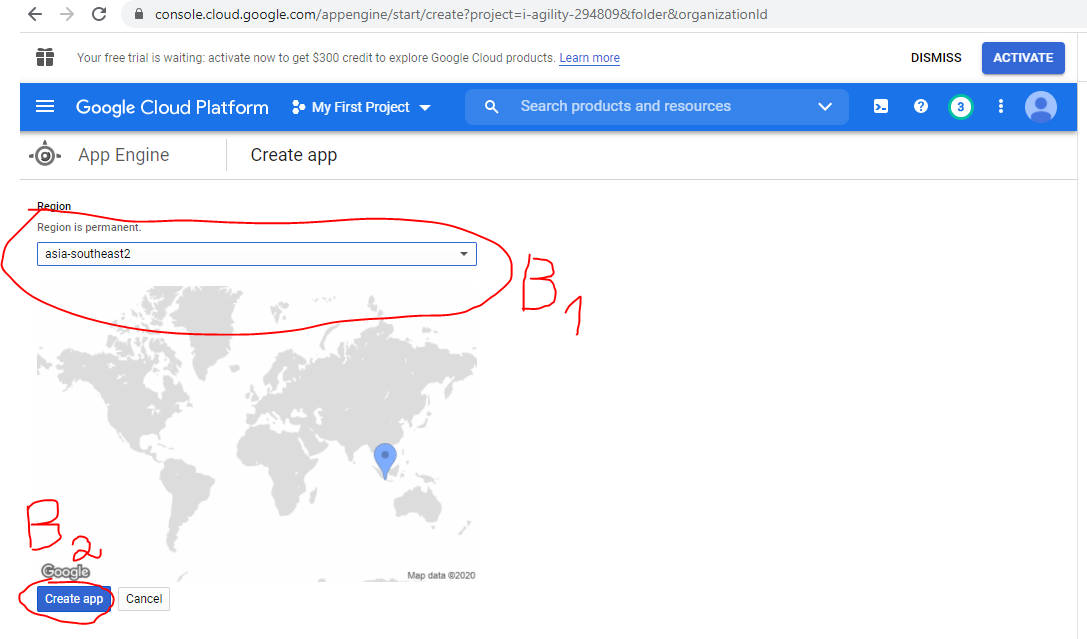
[**https://console.cloud.google.com/appengine/start?project=i-agility-294809&folder&organizationId**](https://console.cloud.google.com/appengine/start?project=i-agility-294809&folder&organizationId)**. Click vào “Create Application”**



**B6:**

**-Chọn “asia-southeast2”**

**-Create app**

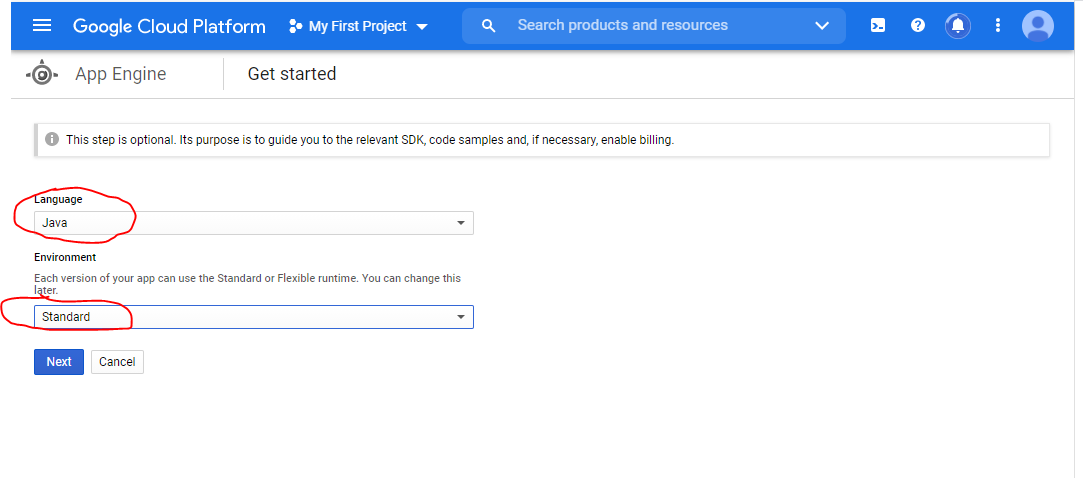


**B7:**

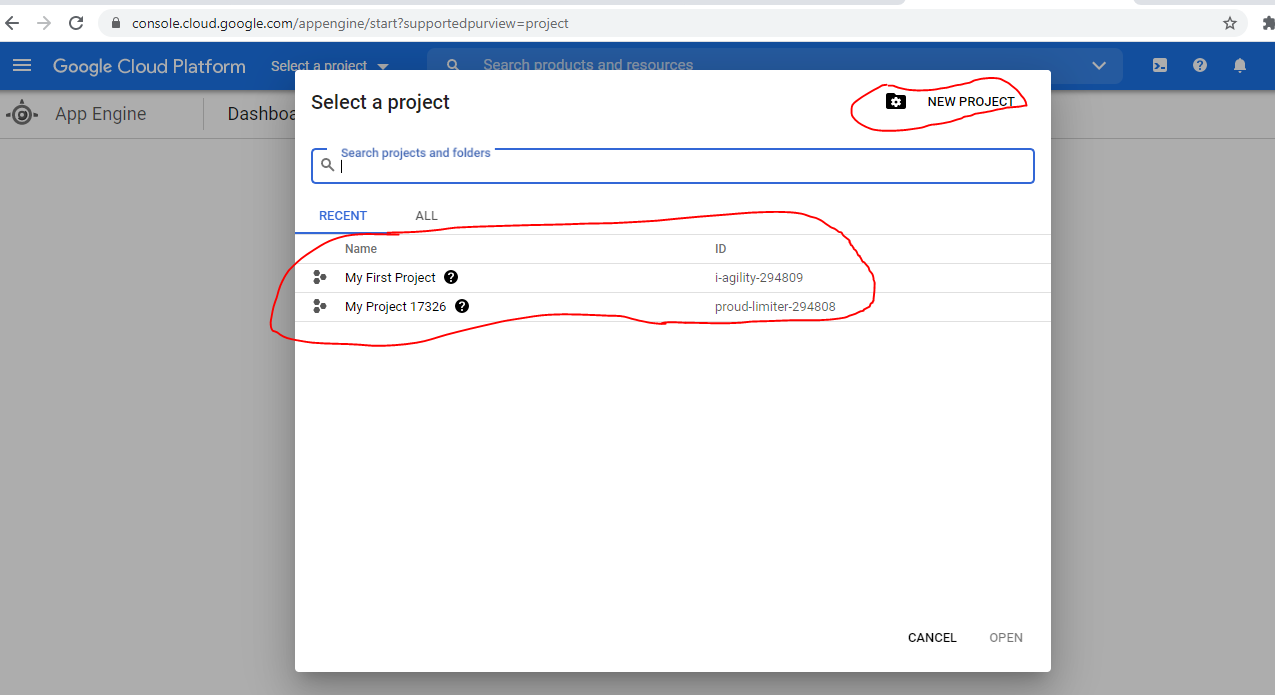
**-Chọn “Java”**

**-Chọn “Standard”**

**-Next**



**B8: Sau khi tạo app sẽ ra giao diện để tạo project**

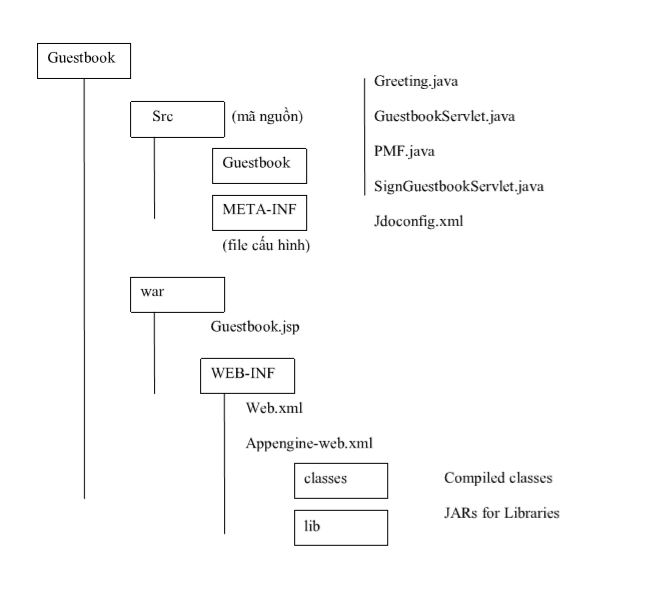


**Tạo một project:**

Các ứng dụng App Engine Java sử dụng các chuẩn java servlet để tương tác với môi trường máy chủ wed. Các file của một ứng dụng bao gồm: các file class đã được biên dịch từ file java, các file JAR của bộ thư viện, các file tĩnh(css,…) và các file xml cấu hình. Tất cả được sắp xếp theo một cấu trúc thư mục và nằm trong thư mục WAR.

**a.Cấu trúc thư mục project:**

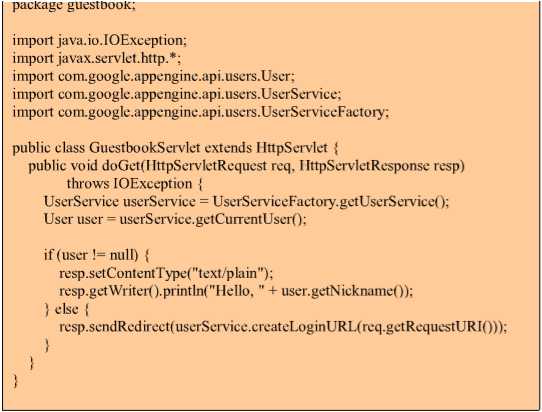
Một thư mục với tên Guestbook được tạo để chứa dự án. Bên trong là 2 thư mục, một thư mục mang tên/src để chứa mã nguồn java và một thư mục/war để chứa các file class được biên dịch từ file nguồn java. Thư mục war được xem là một ứng dụng hoàn chỉnh dùng để up lên Google App.

Tạo cây thư mục như sau:

**b.The servlet class:**

Các ứng dụng App Engine Java sử dụng java servlet API để tương tác với máy chủ web. Một HTTP Servlet là một lớp ứng dụng có khả năng xử lý và phản hồi các yêu cầu web. Lớp này thừa kế lớp javax.servlet.GenericServlet hoặc lớp javx.servlet.http.HttpServlet.

Trong thư mục src/guestbook tạo một file có tên GuestbookServlet.java có nội dung:

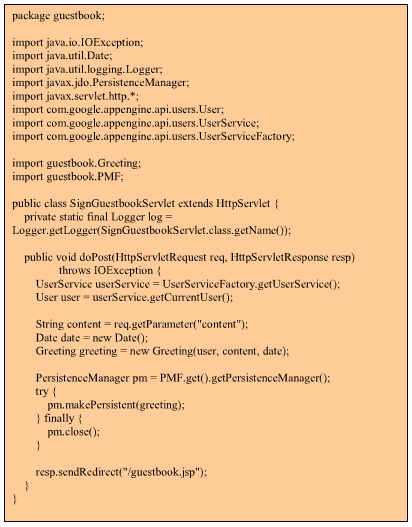


Google App Engine cung cấp một số dịch vụ hữu ích dựa trên cơ sở hạ tầng Google, có thể truy cập bởi các ứng dụng bằng cách sử dụng thư viện có trong SDK. Một trong những dịch vụ người dùng, cho phép tích hợp các ứng của bạn với tài khoản người dùng Google. Với dịch vụ người dùng, người dùng của bạn có thể sử dụng Google các tài khoản mà họ đã có thể đăng nhập vào ứng dụng của bạn.

Trong đoạn code trên, class GuestbookServlet sử dụng User API để kiểm tra xem người dùng đã đăng nhập với tài khoản Google. Nếu chưa, thì người dùng sẽ được chuyển đến màn hình đăng nhập tài khoản Google.userService.createLoginURL(…) sẽ trả về URL của màn hình đăng nhập.

**c.Signgustbookservlet.java file:**

Trong src/guestbook/ tạo 1 file SignGuestbookServlet.java có nội dung sau:

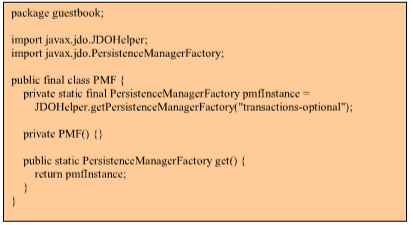


Đoạn code này tạo đối tượng Greeting mới bằng việc gọi hàm. Để lưu đối tượng vào kho dữ liệu, nó tạo ra PersistenceManager nhờ vào việc sử dụng PersistenceManagerFactory. Khi makePersistent được trả về, các đối tượng mới được lưu trong kho dữ liệu.

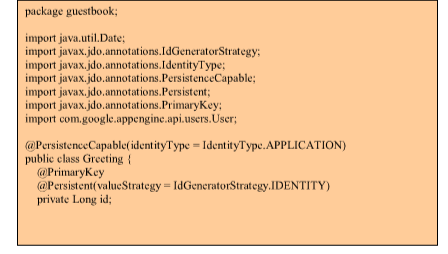
**d.PMF file:**

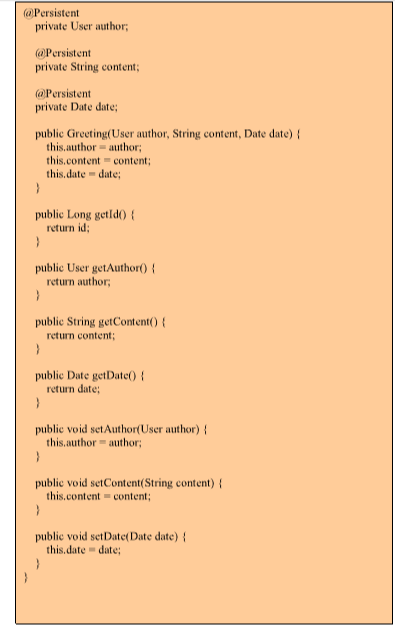
Mỗi khi yêu cầu sử dụng, kho dữ liệu sẽ tạo ra một đối tượng mới cảu lớp PersistenceManager thông qua thư class PersistenceManagerFactory.

Trong thư mục src/guestbook tạo một file PMF.java có nội dung:



**e.Getting.java**

 Trong thư mục src/guestbook ta tạo một file Greeting.java có nội dung:



Lớp này sẽ định nghĩa 3 thuộc tính: author, content và date. Ba cái trường này sẽ được chú thích @Persistence để thông báo cho DataNuclcus lưu chúng như 3 đối tượng thuộc tính trong Google App.

f.jdoconfig.xmlfile:

Trong thư mục war/WEB-INF/classes/META-INF/, ta tạo file jdoconfig.xml để cấu hình Google APP thành nền tảng truy xuất.



**g.guestbook.jsp file:**

Trong war/ ta tạo 1 file guestbook.jsp để làm giao diện cho người sử dụng post các bình luận.



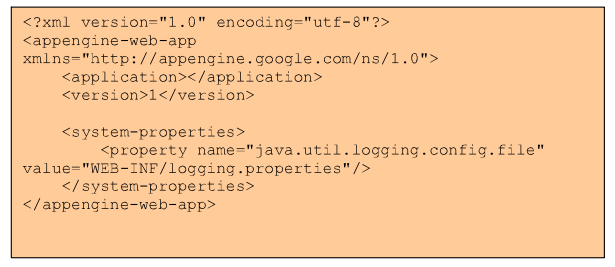
**h.web.xml file:**

Khi máy chủ web nhận được yêu cầu, nó xác định servlet class để gọi bằng cách sử dụng một file cấu hình được gọi là” mô hình triển khai ứng dụng web”. File này được đặt tên là web.xml, va fnawfm trong war/WEB-INF/ thư mục trong WAR.WEB-INF/ và web.xml là một phần cảu đặt tả servlet.

**i.theappengine-web.xml:**

App Engine cần một tập tin cấu hình bổ sung để tìm ra cách để triển khai và chạy ứng dụng. File này được đặt tên appengine-web.xml, và nằm trong WEB-INF/ cùng với web.xml. Nó bao gồm các ID đăng ký cảu các ứng dụng của bạn (Eclipse tạo ra điều này với một ID trống để bạn điền vào sau), số phiên bản của ứng dụng của bạn , và danh sách các file đó phải được đối xử như các file tĩnh ( như hình ảnh và CSS) và các file tài nguyên (như JSP và dữ liệu ứng dụng khác).

Trong thư mục war/WEB-INF/, một tập tin có tên appengine-web.xml có nội dung sau đây:



Triển khai ứng dụng:

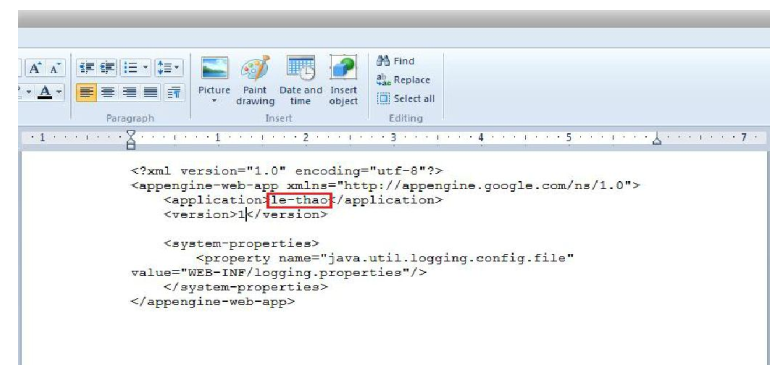
Để upload code và các file chương trình của bạn lên Google App Engine ta dung một tập lệnh có trong SKD có tên: appcfg.cmd

Cần chuẩn bị:

Đưa ID vào ứng dụng :

Cần 1 ID application đã được đăng ký trên google để upload ứng dụng, theo như cách đăng ký phần trên, ta có được ID application là : …

Ta vào thư mục war/WEB-INF/appengine-web.xml mở file appengine-web.xml lên, đặt ID application vào giữa thẻ <application></application> rồi save lại như hình vẽ:



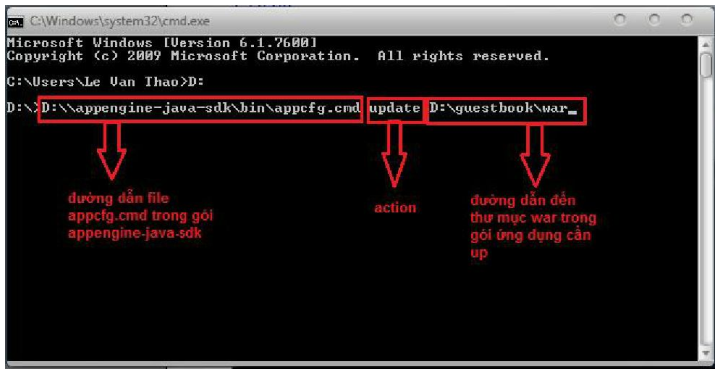
Bắt đầu upload:

Sau đó giải nén ra ổ D ta được D:\appengine-java-sdk, mở cmd ta gõ:

D: //Chuyển về ổ D



Gõ: D:\\appengine-java-sdk\bin\appcfg.cmd update D:\guestbook\war để upload như hình vẽ



Điền mail vào pass nếu chương trình hỏi và upload thành công như hình dưới:

