**Đề xuất mô hình hồ sơ sức khỏe điện tử trên nền tảng đám mây cho thiết bị di động**

**3.1 – Đề xuất mô hình ứng dụng hồ sơ y tế điện tử**

Mạng lưới cơ sở y tế tại Việt Nam được phân cấp có quy mô khác biệt giữa các tuyến: bệnh viện tuyến trung ương, bệnh viện đa khoa tuyến tỉnh, tuyến huyện, trung tâm y tế xã, y tế thôn bản. Bộ y tế đang gấp rút triển khai ứng dụng CNTT trong quản lý bệnh viện với từng mức độ quản lý khác nhau, chỉ dừng lại ở mức độ quản lý trên máy tính, lưu trữ đầy đủ thông tin cần thiết về bệnh nhân đối với một số bệnh viện, hoặc cao hơn nữa là trao đổi hồ sơ bệnh án, hội chẩn từ xa giữa các bệnh viện. Dù quy mô ứng dụng lớn hay nhỏ, để có thể trao đổi thông tin y tế giữa các cơ sở y tế với nhau thì phải giải quyết được ba bài toán cơ bản:

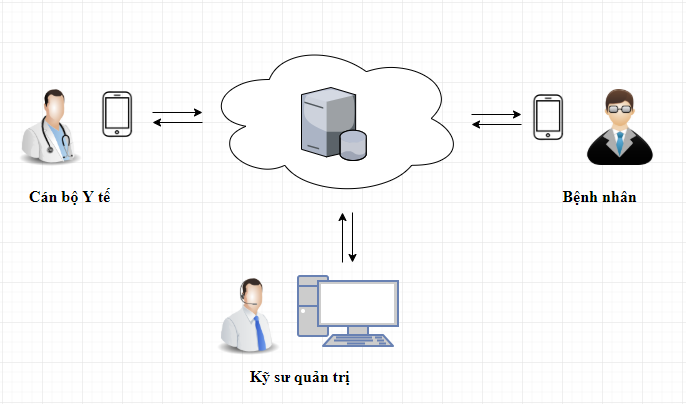
* Thứ nhất là bài toán về ngữ nghĩa, nghĩa là bên gửi và bên nhận phải diễn giải giống nhau về các thông tin được trao đổi.
* Thứ hai là bài toán về cú pháp, nghĩa là bên gửi và bên nhận phải thống nhất với nhau về quy trình trao đổi thông tin.
* Thứ ba là bài toán về mô hình truyền – nhận.

Với thực trạng là các đơn vị triển khai hồ sơ y tế điện tử riêng lẻ còn chưa theo chuẩn, muốn xây dựng một hệ thống bệnh án thống nhất thì đầu tiên thông tin cần phải được chuẩn hóa thỏa mãn đòi hỏi của hai bài toán về ngữ nghĩa và cú pháp. Do đó, tiêu chí chung là xây dựng hệ thống quản lý theo chuẩn quốc tế, mà chủ yếu là chuẩn HL7 trong hệ thống quản lý bệnh viện đã được trình bày cụ thể ở chương trước và chuẩn DICOM trong quản lý hình ảnh.

Trong phạm vi đề tài này, tác giả đưa ra đề xuất về mô hình truyền-nhận bằng các thiết bị di động thông qua các dịch vụ cơ sở dữ liệu lưu trữ trên đám mây. Mô hình không những đáp ứng được các yêu cầu khắt khe của trao đổi dữ liệu y tế, mà còn góp phần thu hẹp sự chênh lệch rất lớn về cơ sở vật chất giữa cơ sở y tế ở các tuyến khác nhau. Phần đầu tiên của chương này sẽ xây dựng mô hình cũng như chi tiết cách thức chuẩn hóa dữ liệu, trao đổi dữ liệu trong hệ thống hồ sơ y tế điện tử.

**3.1.1 Mô hình ứng dụng**

Tác giả xin đưa ra đề xuất mô hình tổng thể về trao đổi thông tin hồ sơ y tế:



Hình 3-1 Mô hình quản lý-chia sẻ hồ sơ y tế điện tử

Hệ thống 3-1 bao gồm mặt sau Back-end và mặt trước Front-end. Hai thành phần này được kết nối qua mạng Internet.

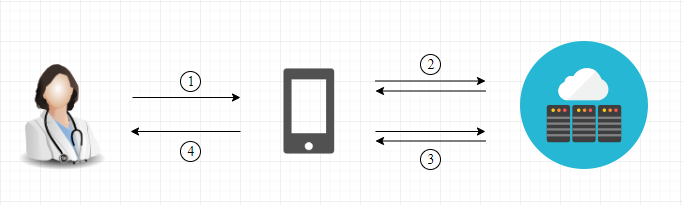
* Phần back-end là cơ sở dữ liệu được đặt trên đám mây. Hồ sơ y tế được lưu trữ trên đám mây theo các chuẩn quốc tế và được giám sát toàn quyền bởi các kỹ sư quản trị hệ thống. Nhà cung cấp dịch vụ chịu trách nhiệm quản lý và bảo mật dữ liệu, đảm bảo cloud server luôn hoạt động xử lý trả về thông tin khi có yêu cầu truy xuất.
* Phần front-end là ứng dụng trên thiết bị di động hỗ trợ người dùng (bao gồm cả bác sĩ, y tá và bệnh nhân) tương tác với hệ thống. Ứng dụng có nhiệm vụ gửi các yêu cầu xử lý dữ liệu từ người dùng (truy xuất, khởi tạo, cập nhật, xóa dữ liệu...) tới cloud server, nhận phản hồi từ cloud server, giải mã dữ liệu và hiển thị kết quả lên thiết bị của người dùng. Một thành phần front-end khác là giao diện Web của Cloud Firestore mà người quản trị hệ thống sử dụng để quản lý toàn dự án.

Hệ thống này có khả năng trao đổi thông tin cả 3 chiều trong trường hợp cần trao đổi hồ sơ bệnh án, trong đó không chỉ các bệnh viện tham gia hệ thống có thể gửi và nhận dữ liệu hồ sơ (gồm bệnh viện trực tiếp khám cho bệnh nhân và bệnh viện muốn tham khảo hồ sơ bệnh án để hội chẩn từ xa), bên cạnh đó bệnh nhân cũng được trao quyền tự tra cứu thông tin liên quan đến sức khỏe của bản thân.

**3.1.2 Chi tiết quy trình trao đổi thông tin hồ sơ y tế**

a/ Quy trình truy xuất hồ sơ y tế

Ứng dụng hồ sơ y tế điện tử phải được xây dựng một giao diện rõ ràng, thân thiện và dễ sử dụng đối với mọi người để thuận tiện cho quá trình trao đổi hồ sơ y tế. Quy trình đăng nhập và trao đổi dữ liệu với máy chủ như sau:

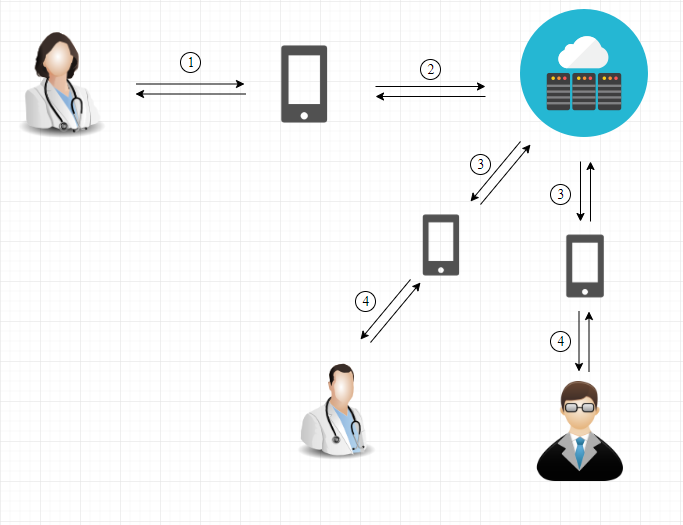


* Bước 1: Đầu tiên, khi người dùng muốn giao tiếp với cơ sở dữ liệu đám mây phải đăng nhập vào hệ thống bằng cách nhập thông tin tài khoản đã được cấp vào ứng dụng trên thiết bị di động.
* Bước 2: Xác thực quyền truy cập được tiến hành như sau:
  + Qua mạng Internet, thông tin về tài khoản và mật khẩu được ứng dụng di động gửi lên xác minh ở dịch vụ quản lý điện toán đám mây.
  + Nếu tài khoản không hợp lệ, xác thực lỗi xảy ra, ứng dụng yêu cầu người dùng nhập lại thông tin chính xác.
  + Nếu tài khoản được xác định là hợp lệ, một xác thực được phản hồi về ứng dụng và chuyển qua giao diện hiển thị dữ liệu.
* Bước 3: Ứng dụng gửi yêu cầu lấy thông tin hồ sơ y tế điện tử lên dịch vụ điện toán đám mây.
  + Nếu tài khoản có quyền đọc, dữ liệu tương ứng sẽ được phản hồi về.
* Bước 4: Ứng dụng hồ sơ y tế điện tử sẽ thực hiện các công việc sau:
  + Tiếp nhận bản tin phản hồi từ đám mây.
  + Giải mã bản tin dựa theo chuẩn (HL7, DICOM) và trình bày thông tin lên giao diện trực quan đối với người dùng.

b/ Quy trình trao đổi hồ sơ y tế

Khi một bệnh nhân nhập viện/xuất viện/khám chữa bệnh, chính các bác sĩ và y tá bằng tài khoản cá nhân trực tiếp nhập thông tin của bệnh nhân vào các danh mục trên ứng dụng di động, sau đó gửi lên lưu trữ ở cơ sở dữ liệu đám mây. Do đó, những cán bộ y tế ở các cơ sở y tế khác có thể nhanh chóng trích xuất được dữ liệu miễn là được cấp quyền. Việc giới hạn quyền truy cập vào một nhánh dữ liệu trong phạm vi các cá nhân có trách nhiệm liên quan được dịch vụ điện toán đám mây hỗ trợ đầy đủ.

Quá trình trao đổi được mô tả như sau:



* Bước 1: Sau khi đăng nhập, cán bộ y tế thực hiện nhập dữ liệu về lần thăm khám của bệnh nhân vào giao diện của ứng dụng di động.
* Bước 2: Ứng dụng thực hiện các công việc sau:
  + Thực hiện tái cấu trúc (mã hóa) dữ liệu theo chuẩn HL7 và DICOM.
  + Thông qua các lệnh API gửi dữ liệu đã xử lý lên cơ sở dữ liệu đám mây theo thời gian thực.
  + Phản hồi cho người gửi thông báo thông tin đã được lưu trữ thành công. Hoặc lưu trữ thông tin khi thiết bị di động ngoại tuyến do gặp trục trặc về đường mạng kết nối Internet. Ứng dụng tự động thực hiện lại việc gửi dữ liệu khi có kết nối trở lại.
* Bước 3: Khi một bên thứ ba có thẩm quyền muốn truy cứu lịch sử khám chữa bệnh của bệnh nhân, họ phải đăng nhập tài khoản được cấp quyền trên ứng dụng, nhập vào thông tin định danh bệnh nhân và gửi yêu cầu truy xuất thông tin lên CSDL.
  + Khi yêu cầu được xác minh là hợp lệ gói dữ liệu kết quả phản hồi về ứng dụng xử lý.
  + Trường hợp bên thứ ba đã đăng nhập sẵn, máy chủ server sẽ tự động gửi gói dữ liệu liên quan đến sự thay đổi mới tới các ứng dụng trên thiết bị cập nhật lại thông tin.
* Bước 4: Ứng dụng nhận dữ liệu thực hiện những công việc sau:
  + Giải mã gói dữ liệu liên quan thông tin khám chữa bệnh của bệnh nhân dựa theo chuẩn HL7 và DICOM.
  + Hiển thị hồ sơ sức khỏe của bệnh nhân lên giao diện người dùng.

Ở phần tiếp theo, tác giả tiến hành lựa chọn một dịch vụ cung cấp đám mây cơ sở dữ liệu phục vụ cho quá trình thử nghiệm hệ thống.

**3.1.3 Thiết kế mô hình đám mây PaaS**

Điện toán đám mây là giải pháp điện toán dựa trên Internet, ở đó các máy chủ trong đám mây được cấu hình để làm việc cùng nhau cung cấp tài nguyên và các ứng dụng khác nhau sử dụng sức mạnh điện toán tập hợp, cứ như thể là chúng đang chạy trên một hệ thống duy nhất.

Mô hình điện toán đám mây của hồ sơ y tế điện tử được thiết kế thuộc kiểu PaaS (Platform as a Service) – nền tảng như một dịch vụ. PaaS là một dạng dịch vụ pay-per-use – trả tiền theo định mức hay chỉ trả tiền cho những gì sử dụng. Nhà cung cấp sẽ cho khách hàng thuê cơ sở hạ tầng bao gồm các máy chủ server, ổ cứng lưu trữ, mạng và các bảo vệ an ninh nâng cao. Bên cạnh đó họ cũng trợ giúp cả hệ điều hành (ví dụ Windows, Linux) cho tới Runtime (ví dụ Docker, NodeJS, C#). Điều này giúp nhà phát triển tập trung hoàn toàn vào viết ứng dụng mà chỉ cần cấu hình cơ sở hạ tầng trước khi triển khai.

Cụ thể hơn, mô hình điện toán đang rất phổ biến là BaaS (Backend as a Service) thuộc trong nhóm Paas – là mô hình mà ở phần backend lập trình viên không có phải lo cài đặt server (serverless). Những kiểu backend này về lý thuyết sẽ có thể mở rộng không giới hạn mà vẫn có độ tin cậy và khả năng chịu lỗi cao. Một số cơ sở dữ liệu nổi tiếng là:

* NoSQL: Google Firebase, Google Cloud Firestore, AWS DynamoDB
* SQL: AWS Aurora Serverless
* Files: AWS S3, Google Cloud Storage
* Data Warehouse: Google BigQuery

Tại sao điện toán đám mây lại trở thành xu thế:

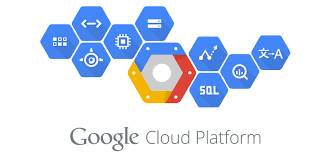
* Giảm chi phí: điện toán đám mây có thể làm giảm cả chi phí vốn (CAPEX) lẫn chi phí vận hành (OPEX), vì các tài nguyên chỉ được mua khi cần và chỉ phải trả tiền khi sử dụng.
* Giảm thời gian phát triển và bảo trì: sử dụng điện toán đám mây cho phép nhà phát triển tập trung vào phát triển phần mềm mà không cần quan tâm đến chi phí bảo trì phần cứng.
* Khả năng mở rộng linh hoạt: Điện toán đám mây cho phép khả năng điều chỉnh quy mô ngay lập tức mà gần như không phải dừng hoạt động của máy chủ.

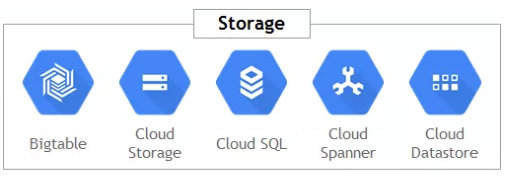
Ngoài ra, mô hình dữ liệu trên nền tảng đám mây còn có khả năng duy trì dữ liệu ngay cả khi thiết bị ngoại tuyến, truy cập độ trễ thấp ở mọi nơi trên thế giới và đồng bộ hóa dữ liệu theo thời gian thực trên tất cả các nền tảng di động.

**3.2 – Lựa chọn dịch vụ điện toán đám mây**

**3.2.1 Giới thiệu nền tảng Google Cloud Platform**

Một sự lựa chọn tốt nhất hiện nay khi xây dựng, thử nghiệm và triển khai các ứng dụng trên cơ sở hạ tầng đang tin cậy và có khả năng mở rộng cao là nền tảng đám mây của Google - Google Cloud Platform (GCP).





Hình 3-2 Nền tảng đám mây Google và các dịch vụ lưu trữ

Google Cloud Platform là nền tảng điện toán đám mây cho phép các doanh nghiệp, tổ chức, hay cá nhân xây dựng và chạy các ứng dụng của mình trên hệ thống mà Google cung cấp, đồng thời sử dụng các sản phẩm có sẵn như Google Search, Google Maps, Google Chrome,… GCP hỗ trợ tất cả các dịch vụ thiết yếu bao gồm: Storage, Big Data, Compute Engine, Networking, Management, Developer Tools,…

Nhờ sự thống trị toàn cầu của các thiết bị di động chạy hệ điều hành Android, Google đã phát triển hệ thống đám mây thông qua điện thoại và máy tính bảng của họ. Chức năng cơ bản nhất mà chúng ta rất hay sử dụng là cho phép việc đồng bộ hóa điện thoại của mình với các thiết bị khác bao gồm máy tính bảng và PC khi sử dụng trình duyệt web Chrome. Nền tảng Google cung cấp một số công cụ được phát triển tốt và đáng tin cậy, bao gồm các công cụ tính toán, công cụ vùng chứa (storage), lưu trữ đám mây (firestore, datastore) và BigQuery. Với tốc độ nhanh chóng của Internet toàn cầu, những hạn chế tồn tại về không gian lưu trữ trên điện thoại thông minh sẽ biến mất hoàn toàn nhờ tiềm năng lưu trữ đám mây ví dụ như Google Drive.

Về dịch vụ lưu trữ (Storage) của GCP là Google Cloud Storage (GCS). GCS là một kho lưu trữ và truy cập dữ liệu trực tuyến trên cơ sở hạ tầng của GCP. Dịch vụ này kết hợp hiệu suất và khả năng mở rộng gần như vô hạn của Google Cloud với khả năng chia sẻ vàbảo mật cao. Đây là một cơ sở hạ tầng như một dịch vụ (IaaS), có thể so sánh với dịch vụ lưu trữ trực tuyến của Amazon S3.

Người dùng có thể sử dụng Cloud Storage cho các tình huống bao gồm:

* Lưu trữ dữ liệu và khôi phục;
* Phân phối các đối tượng dữ liệu lớn cho người dùng thông qua tải xuống trực tiếp;
* Kho lưu trữ tích hợp cho phân tích và máy học.

**3.2.2 Lựa chọn dịch vụ Cloud Firestore**

Cloud Firestore là cơ sở dữ liệu dạng NoSQL được nâng cấp từ Firebase realtime database – một dịch vụ điện toán đám mây cho các ứng dụng Web và di động có hiệu năng cao và khả năng mở rộng tự động. Các ứng dụng với cloud firestore cho phép các nhà phát triển tự động đồng bộ dữ liệu giữa các thiết bị theo thời gian thực, ngoài ra cũng có thể xây dựng các ứng dụng có khả năng hoạt động ngoại tuyến.

Hiện tại Cloud Firestore đang là bản thử nghiệm Beta và có thể truy cập trực tiếp tại địa chỉ: <https://console.firebase.google.com>



Hình Dịch vụ Cloud Firestore của Google

Ưu điểm của Cloud Firestore:

- Triển khai hoạt động nhanh với CSDL NoSQL: Cloud Firestore cung cấp khá nhiều API, hỗ trợ đa nền tảng (mobile, Web, IoT) đơn giản hóa các quá trình quản lý, lưu trữ, đồng bộ cũng như truy xuất dữ liệu, hosting, xác thực người dùng.

- Đồng bộ dữ liệu trên các thiết bị online và offline: Cloud Firestore tích hợp sẵn tính năng hỗ trợ offline một local cache phục vụ và chứa dữ liệu.

- Bảo mật: Tích hợp bộ điều khiển truy cập và cho phép quá trình xác minh thông qua các cú pháp quy định sẵn.

- Sự ổn định: Cloud Firestore hoạt động dựa trên nền tảng cloud cung cấp bởi Google, không lo bị DDOS (tấn công bằng từ chối dịch vụ phân tán – Distributed Denial of Service) hoặc nghẽn mạng, việc bảo trì cũng không phải dừng hoạt động của Server.

Nhược điểm của Cloud Firestore

Cơ sở dữ liệu của Firestore được tổ chức theo kiểu JSON collection và documents (sẽ được trình bày ở mục tiếp theo), không phải kiểu TABLE như MYSQL nên trong một số trường hợp cần thiết phải chuyển đổi từ SQL Database sang Firestore Database.

Kiểu dữ liệu

Cloud Store hỗ trợ rất nhiều kiểu dữ liệu

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Mô tả** |
| Array | [1, 2, 3] |
| Boolean | True, false |
| Bytes | Tối đa 1,048,487 bytes (1 MiB - 89 bytes). Chỉ 1,500 bytes đầu tiên được xem xét bởi các truy xuất. |
| Data and time |  |
| Floating-point number |  |
| Geographical point | Tọa độ |
| Integer | 64-bit, có dấu |
| Map | Key-value  {a: “aaa”, b: “baz”} |
| Null |  |
| Reference | Tham chiếu tới document nằm trong project khác |
| Text string | UTF-8 encoded. Tối đa 1,048,487 bytes (1 MiB - 89 bytes). Chỉ 1,500 bytes đầu tiên được xem xét bởi các truy xuất. |

Bảng Mô tả các kiểu dữ liệu của Cloud Firestore

Chú ý: Cần đọc kỹ hướng dẫn trước khi sử dụng kiểu dữ liệu Array, firestore không cho phép chèn, sửa đổi, hay xóa bỏ phần tử ở một vị trí cụ thể như các ngôn ngữ lập trình.

Truy vấn (Querying)

Cloud Firestore truy vấn với việc đánh chỉ mục và sử dụng cả lọc và sắp xếp với nhau. Nó mạnh mẽ hơn với phương thức where() với 3 tham số như một trường để lọc, một phép toán và một giá trị để so sánh. Ví dụ một số hàm như whereEqualTo(), whereLessThan(), whereGreaterThanOrEqualTo(), whereLessThanOrEqualTo(), ... làm cho việc truy vấn trở nên hiệu quả hơn và dễ dàng hơn. Truy vấn mặc định được đánh chỉ số theo kết quả trả về. Tuy nhiên Cloud Firesotre lại không hỗ trợ các truy vấn sau:

* Mỗi truy vấn chỉ được thực hiện trên một bộ sưu tập nhất định.
* Truy vấn các thành viên của mảng riêng lẻ. Nhà phát triển có thể sử dụng các kỹ thuật sử dụng cho List, ArrayList, Set để truy vấn.
* Truy vấn logic OR bạn nên tạo truy vấn riêng cho từng truy vấn và hợp nhất với nhau.
* Truy vấn có mệnh đề "!=". Trường hợp này lập trình viên nên tạo truy vấn cho từng trường hợp lớn hơn hoặc nhỏ hơn của truy vấn đó.

Sự ổn định và hiệu năng (Reliability and performance)

Cloud Firestore hiện tại vừa kết thúc giai đoạn thử nghiệm với phiên bản beta: Sự ổn định của bản thử nghiệm không phải lúc nào cũng được như bản đã hoàn thành, dữ liệu được đặt ở nhiều nơi đảm bảo tính mở rộng và độ tin cậy cao, khi nó vượt qua giai đoạn thử nghiệm sẽ hứa hẹn sự ổn định và hiệu năng tốt hơn so với phiên bản tiền nhiệm FireBase Database.

Khả năng mở rộng (Scalability)

Trước đây, FireBase Database nếu muốn mở rộng phải yêu cầu sự phân tán, có thể chia sẻ cho khoảng 100.000 kết nối và 1000 lần ghi / giây trong một database. Mở rộng thì phải yêu cầu phân tán dữ liệu của bạn trên nhiều cơ sở dữ liệu. Với Cloud Firestore, việc mở rộng này là hoàn toàn tự động, vậy nên lập trình viên sẽ không cần phải quan tâm đến việc dữ liệu của mình lưu trữ theo nhiều phiên bản.

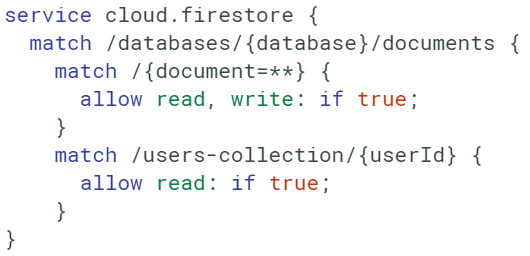
Bảo mật (Security)

Cloud Firestore thì bảo mật đơn giản và mạnh mẽ cho SDK di động, web và server: SDK trên thiết bị di động và web sử dụng các quy tắc bảo mật của Cloud Firestore. SDK máy chủ sử dụng quản lý danh tính và truy cập ( Identity and Access Management - IAM), các quy tắc không xếp tầng trừ khi bạn sử dụng ký tự đại diện, xác thực dữ liệu diễn ra một cách tự động, quy tắc có thể hạn chế việc truy vấn nếu như người dùng truy vấn vào dữ liệu không có quyền truy cập thì toàn bộ truy vấn sẽ không thành công.

Cơ chế xác thực của Firestore (Firestore Authentication) hỗ trợ người dùng xác thực vào Firestore sử dụng FirestoreUI để đăng nhập hoặc dùng Firebase Authentication SDK tích hợp trực tiếp một vài phương thức đăng nhập vào ứng dụng di động. Các phương thức đăng nhập đang được hỗ trợ là:

* Tài khoản thư điện tử
* Tài khoản Google
* Tài khoản mạng xã hội như Facebook, Twitter, Github
* Số điện thoại
* Và còn có cả đăng nhập ở chế độ khách Anonymous

Các quy tắc bảo mật được Firestore xây dựng đơn giản nhưng hiệu quả. Lập trình viên có thể cấp quyền đọc/ghi/xóa cho từng tài khoản trên từng nhánh dữ liệu bằng các cú pháp đơn giản như sau.



Hình Thiết lập quyền truy cập dữ liệu

Để đáp ứng tính bảo mật, mỗi tài khoản chỉ được cấp quyền truy cập (đọc, ghi, sửa đổi) nhất định dựa theo chức năng và nhiệm vụ. Firestore hỗ trợ thiết lập quyền theo từng nhánh như sau:

* Quyền read (đọc): gồm các quyền con là get (lấy dữ liệu), list (liệt kê dữ liệu)
* Quyền write (ghi): gồm các quyền con là create (tạo mới), update (cập nhật), delete (xóa)

Cách tính phí của Google cloud firestore

Một điều quan trọng cần phải cân nhắc là Google tiến hành thu phí sử dụng dịch vụ lưu trữ như thế nào. Khi nhà phát triển sử dụng Cloud Firestore, Google sẽ tính cước theo cách sau:

* Số lượng đọc, ghi, và xóa dữ liệu được thực hiện
* Tổng dung lượng cơ sở dữ liệu sử dụng, bao gồm khối lượng dụng cho metadata và đánh số indexes.
* Tổng băng thông sử dụng

Dung lượng và băng thông được tính theo gigabyte (GiB). Từ ngày 03/03/2019, tỉ lệ tính phí được áp dụng như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Miễn phí hàng ngày | Cước đơn vị  ($) | Cước đơn vị  (VND) | Đơn vị tính cước |
| Đọc document | 50,000 | $0.06 | 1,380 đồng | 100,000 reads |
| Ghi document | 20,000 | $0.18 | 4,140 đồng | 100,000 writes |
| Xóa document | 20,000 | $0.02 | 460 đồng | 100,000 deletes |
| Dung lượng dữ liệu | 1 GB | $0.18 | 4,140 đồng | GB/tháng |

Bảng Giá cước sử dụng dịch vụ Cloud Firestore

Trong quá trình thực hiện đồ án, tác giả chỉ cần đăng ký bằng tài khoản thư điện tử Gmail là có thể sử dụng chế độ miễn phí (Spark plan).

3.3 Tổng quát về cơ sở dữ liệu

3.3.1 Cơ sở dữ liệu NoSQL

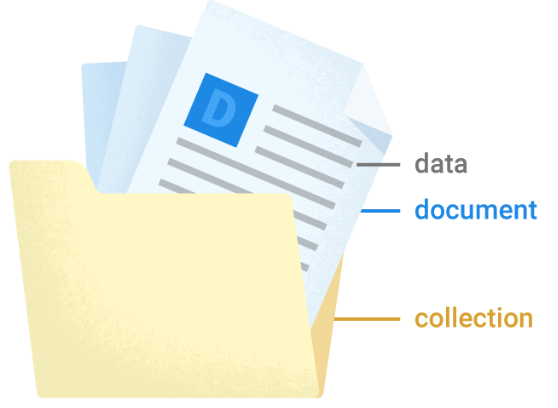
3.3.2 Cơ sở dữ liệu Firestore

Cloud Firestore tương tự Firebase realtime database đều lưu trữ dữ liệu người dùng trên nền tảng đám mây theo dạng NoSQL, dữ liệu được đồng bộ tới tất cả các máy khách kết nối với nó theo thời gian thực và vẫn khả dụng khi ứng dụng ngoại tuyến. Chúng phù hợp để làm những ứng dụng cần thời gian cập nhật nhanh chóng như là một ứng dụng chat, chia sẻ dữ liệu tức thời khi nhiều người làm cùng một việc.

Cloud Firestore lưu trữ dữ liệu được sắp xếp trong các bộ sưu tập (collection), điều này cũng giúp việc lưu trữ dữ liệu dễ dàng vì nó tương tự như JSON. Dữ liệu phân cấp, phức tạp cũng dễ dàng tổ chức hơn để sắp xếp các tài liệu bên trong đó. Nó cũng yêu cầu ít chuẩn hóa và làm gọn dữ liệu.

Mô hình dữ liệu của firestore được phân cấp theo cấu trúc cây gồm 3 thành phần cơ bản:

* Collection: bộ sưu tập
* Document: tập dữ liệu
* Field: trường



Hình . Các thành phần dữ liệu của firestore

Dữ liệu được chia thành các Collection khác nhau. Trong mỗi collection có chứa một hoặc nhiều Document. Tên của document trong một collection là duy nhất.

Một document là một tập hợp của nhiều trường, là một cặp các từ khóa-giá trị (key-value), trong đó từ khóa là duy nhất trong phạm vi một document.

Ngoài ra một document có thể chứa nhiều collection con, gọi là sub-collection. Điều này có nghĩa là cơ sở dữ liệu chứa trên firestore có thể mở rộng theo cả chiều rộng lẫn chiều sâu một cách linh hoạt.

Ví dụ một cây dữ liệu 3 lớp như sau:

**Collection**

doc

doc

**Collection**

doc

doc

**Collection**

doc

doc

**Collection**

doc

doc

**Collection**

doc

doc

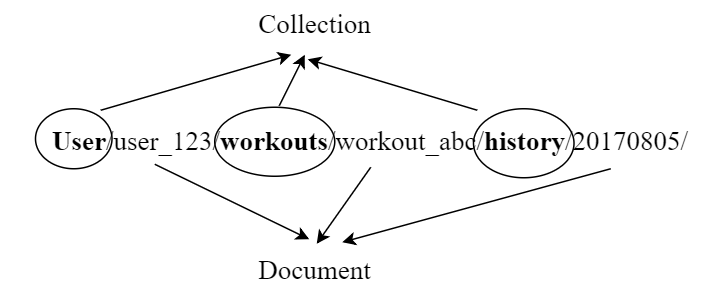
**Collection**

doc

doc

Hình Minh họa cây dữ liệu 3 lớp

Con trỏ tham chiếu có thứ tự luân phiên xen kẽ bắt buộc giữa collection và document. Giả sử cây dữ liệu:



Hình Ví dụ collection – document

Tham chiếu document của minh họa trên là:

Document\_ref.collection(“User”).document(“user\_123”).collection(“workouts”).document(“workout\_abc”).collection(“history”).document(“20170805”)

Tổ chức dữ liệu trên Firestore

Có 3 kiểu tổ chức dữ liệu chính

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Nested data in codument | Sub-collection | Root-level collections |

Bảng các kiểu tổ chức dữ liệu Cloud Firestore cơ bản

* Nested data in document: document chỉ là một tập hợp các trường dữ liệu.
* Sub-collection: bên cạnh các trường dữ liệu, document còn chứa thêm cả các sub-collection mà bên trong sub-collection có thêm nhiều document.
* Root-level collections: có nhiều collection ngang hàng nhau ở gốc.

Một số chú ý về giới hạn của cơ sở dữ liệu trên Firestore

* Giới hạn dung lượng của một document là 1 megabyte, tương đương tối đa 20,000 trường. Firestore tự động đánh số thứ tự index cho tất cả document hỗ trợ query, ngoài ra nhà phát triển có thể tùy chính kiểu đánh số tăng dần hoặc giảm dần.
* Giới hạn về tốc độ đọc ghi là 1 phép ghi/giây cho một document. Nếu nhiều phép ghi được gọi tại một thời điểm thì chỉ một phép ghi được thực hiện, các phép còn lại sẽ được gọi lại tự động.
* Không thể chỉ đọc một phần của document, ứng dụng phải đọc toàn bộ nội dung của document theo đối tượng (object). Và cũng không thể thiết lập quy tắc bảo mật riêng cho từng thành phần của document, quy tắc bảo mật được áp dụng chung cho toàn bộ document.
* Các phép truy xuất dữ liệu là các phép query “cạn”, nghĩa là chỉ tìm kiếm một document trong một collection duy nhất.
* Giới hạn chiều sâu của cây tối đa là 500 lớp.

**3.4 – Triển khai mô hình ứng dụng**

\*\*\* Đối với ứng dụng di động

Ứng dụng thử nghiệm eHealth được viết bằng phần mềm Android Studio phiên bản 3.1. Yêu cầu thiết bị di động chạy hệ điều hành Android phiên bản 4.4 – API level 19 trở lên (tốt nhất khi dùng thiết bị chạy phiên bản 7.0 – API level 24 trở lên).

Cài đặt file eHealthApp\_1.0.apk dung lượng 6,4 Mb.

\*\*\* Đối với cơ sở dữ liệu

Truy cập địa chỉ: <https://console.firebase.google.com>

Sử dụng trình duyệt Web: Firefox, Chrome, Opera, Internet Explorer…

Đăng nhập bằng tài khoản thư điện tử Gmail là có thể sử dụng chế độ thử nghiệm miễn phí Spark plan của Cloud Firestore. Project eHealthRecord-Db được khởi tạo mới và tùy chỉnh làm cơ sở dữ liệu cho quá trình thử nghiệm.

**3.4.1 Cấp quyền đăng nhập**

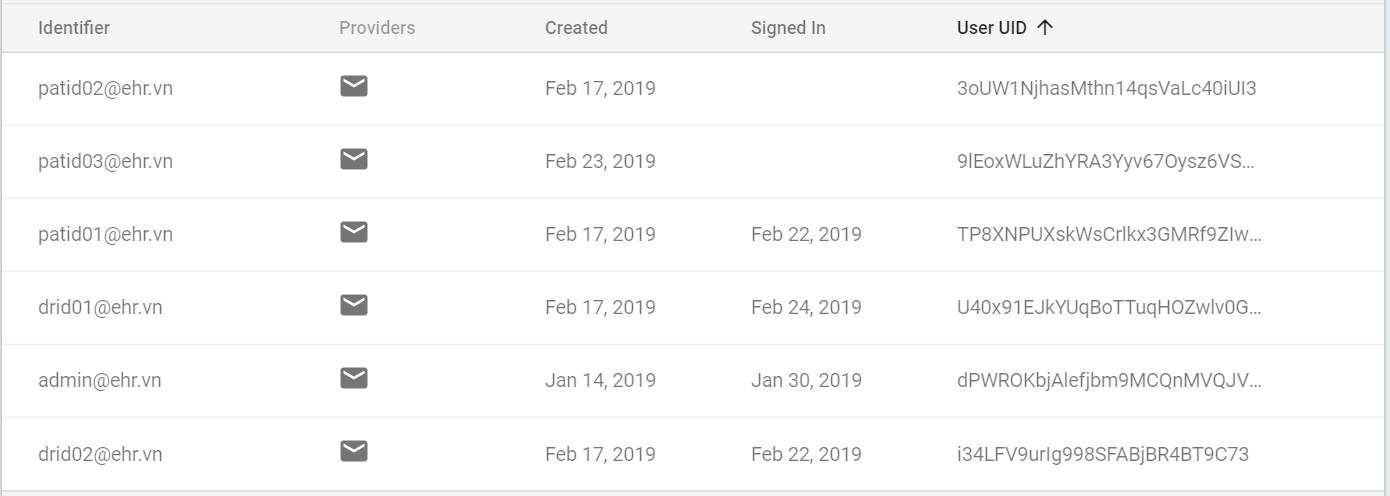
Thiết lập quyền truy cập bằng tài khoản thư điện tử.

Firestore hỗ trợ phương thức đăng nhập tài khoản sử dụng thư điện tử và mật khẩu.

Mỗi bệnh nhân hoặc bác sĩ được cấp một mã số ID tương ứng với tên email đăng nhập. Địa chỉ “@ehr.vn” là mặc định dùng chung cho toàn hệ thống người dùng.

Ví dụ:

* Người quản trị (admin) có tài khoản: admin - thì tương ứng với địa chỉ mail [admin@ehr.vn](mailto:admin@ehr.vn)
* Bác sĩ được cung cấp tài khoản: drid01; drid02…
* Bệnh nhân được cấp tài khoản: patid01; patid02…



Hình. Quản lý tài khoản đăng nhập

Người dùng chỉ cần nhập mã số id (phần nằm trước dấu @), dưới dạng chữ in hoa hoặc chữ viết thường). Ứng dụng sẽ tự động chuyển về định dạng chữ viết thường và thêm vào địa chỉ mail “@ehr.vn”.

Ví dụ:

|  |  |
| --- | --- |
| Người dùng nhập | Phần mềm gửi lên hệ thống đăng nhập |
| DR01 | dr01@ehr.vn |
| Patid01 | patid01@ehr.vn |

Bảng. Định dạng tên đăng nhập người dùng của ứng dụng

Mật khẩu phải thỏa mãn có tối thiểu 6 kí tự. Người quản trị hệ thống có toàn quyền khôi phục mật khẩu, vô hiệu hóa tài khoản và xóa hoàn toàn tài khoản đăng nhập ra khỏi hệ thống.

Quyền truy cập xử lý dữ liệu được quy định như sau:

|  |  |
| --- | --- |
| Loại tài khoản | Quyền |
| Quản trị hệ thống | Đọc/ghi/sửa đổi thông tin của bác sĩ và bệnh nhân |
| Bác sĩ (drid…) | Đọc/ghi/sửa đổi thông tin liên quan tới bệnh nhân |
| Bệnh nhân (patid….) | Chỉ đọc/ghi thông tin cá nhân của chính bản thân |

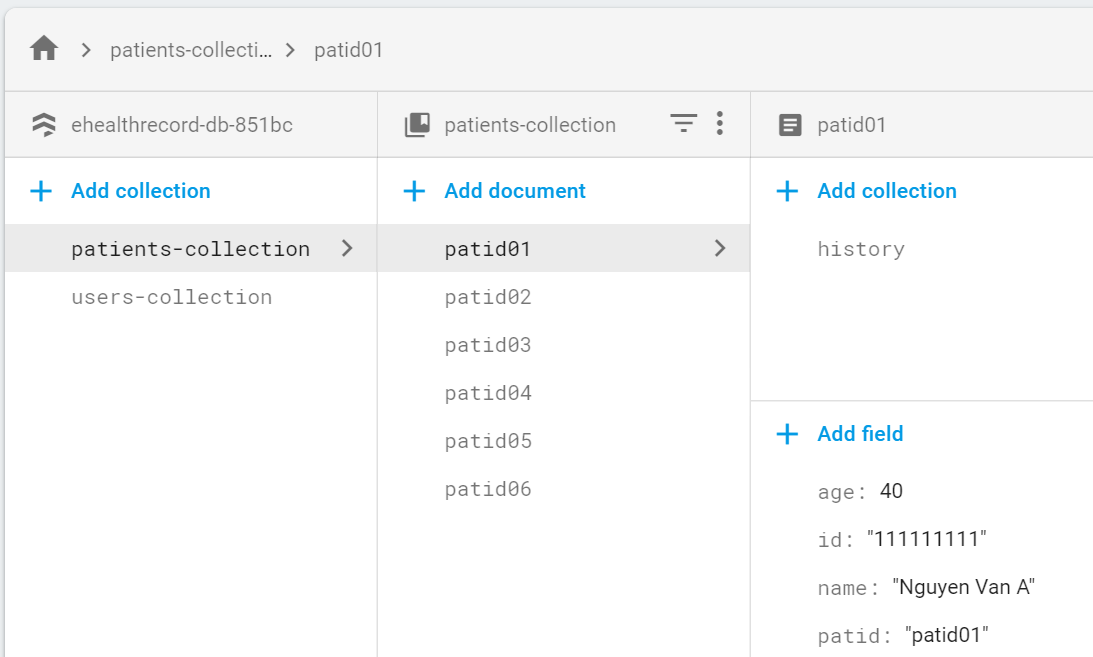
**3.4.2 Tổ chức dữ liệu**

Áp dụng đối với hồ sơ y tế:

Xây dựng một Collection gốc có tên là “patients-collection” quản lý toàn bộ thông tin của bệnh nhân.

Trong collection gốc, mỗi document có tên tương ứng với mã số id của bệnh nhân, lưu trữ thông tin cá nhân của bệnh nhân đó.

Dữ liệu bên trong document, tương ứng với các thông tin cá nhân là các trường (họ tên, số chứng minh thư nhân dân, tuổi) và một sub-collection “history” lưu các thông tin về lịch sử khám chữa bệnh của người đó. Trường “patid” giúp ứng dụng ánh xạ giữa đối tượng bệnh nhân và tên document tương ứng.



Hình. Cây quản lý bệnh nhân

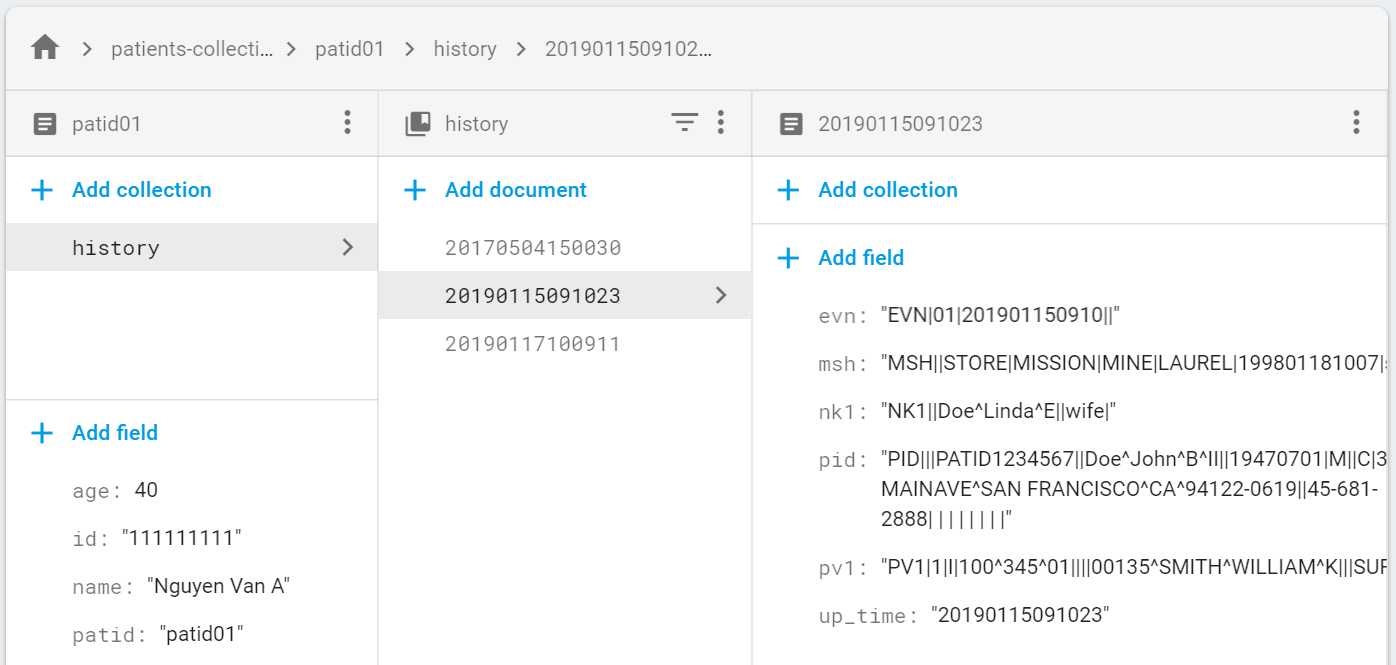
Ví dụ:

* Document ‘patid01’ tương ứng với mã số bệnh nhân ‘patid01’ và tài khoản có địa chỉ email ‘patid01@ehr.vn’. Bệnh nhân tên là Nguyễn Văn A, 40 tuổi, có số chứng minh thư là 111111111.

Bên trong Collection “history”, là thông tin về quản trị bệnh nhân (nhập viện, chuyển viện, xuất viện…) được ghi nhận bằng các bản tin HL7.

Tên của một Document là thời gian bản tin được ghi nhận trên hệ thống theo định dạng ‘yyyymmddhhmmss’ hoặc theo định dạng UnixTime.

Mỗi document lưu các bản tin HL7 gồm nhiều trường, trong đó từ khóa của trường là mã sự kiện (msh, evn, pid…) và giá trị tương ứng của trường là thông điệp HL7.



Hình. Collection lưu trữ thông tin thăm khám của bệnh nhân

**3.4.3 Cài đặt và chạy chương trình**

Sau khi đã xây dựng được bộ dữ liệu mẫu trên Cloud Firestore, tác giả tiến hành thử nghiệm các thao tác tương tác dữ liệu cơ bản từ thiết bị di động.

* Thử nghiệm 1: Đăng nhập tài khoản xác thực và đọc dữ liệu từ đám mây
* Thử nghiệm 2: Ghi dữ liệu lên đám mây
* Thử nghiệm 3: Xóa dữ liệu trên đám mây

Thử nghiệm 1: Đăng nhập tài khoản xác thực và đọc dữ liệu từ đám mây

Ứng dụng eHealth được cài đặt ban đầu sẽ có giao diện khởi động yêu cầu đăng nhập tài khoản. Người dùng có thể đăng nhập bằng tài khoản quản trị viên/bác sĩ hoặc tài khoản của bệnh nhân theo quy ước ở mục 3.4.1

|  |  |
| --- | --- |
|  |  |
| a – tài khoản hợp lệ | b – tài khoản không hợp lệ |

Hình Thử nghiệm đăng nhập tài khoản

Sau khi đăng nhập thành công, đối với tài khoản của nhân viên cơ sở y tế có thẩm quyền khám và chữa bệnh như bác sĩ hoặc y tá hỗ trợ, ứng dụng sẽ lên hiển thị giao diện danh sách bệnh nhân và một số thông tin cá nhân (tham khảo hình – a). Đây chính là các đối tượng document nằm trong collection gốc “patients-collection” mà ứng dụng thực hiện querry và tập trung vào một mảng.

Đối với tài khoản của người bệnh sau khi đăng nhập, hoặc khi bác sĩ lựa chọn một bệnh nhân bất kỳ trong danh sách. Dữ liệu về số lần thăm khám của họ được ứng dụng đọc từ sub-collection “history” và hiển thị lên giao diện (tham khảo hình – b).

|  |  |
| --- | --- |
|  |  |
| A – đọc document ở root collection | B – đọc document ở sub-collection |

Hình. Thử nghiệm đọc dữ liệu từ đám mây

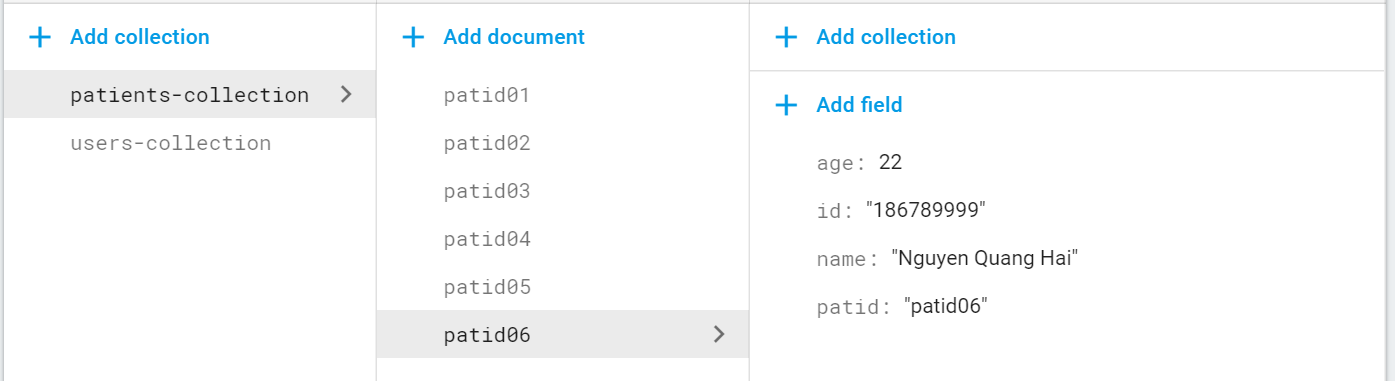
Thử nghiệm 2: Ghi dữ liệu lên đám mây

Đối với các nhân viên cơ sở y tế có thẩm quyền, tài khoản của họ có chức năng thêm mới bệnh nhân vào danh sách bằng cách nhập các thông tin cá nhân của bệnh nhân và thực hiện ghi dữ liệu theo định dạng document object lên collection gốc “patients-collection”. Thông tin của bệnh nhân mới sẽ nhanh chóng được cập nhật trên giao diện Web Cloud Firestore. Tham khảo hình – a.

Cả bác sĩ và bệnh nhân có thể thêm các thông tin về lần thăm khám của bệnh nhân là các document object nằm ở nhánh sub-collection “history”. Đối tượng document này bao gồm các trường của bản tin HL7 EVN, MSH, PID, NK1, PV1.

|  |  |
| --- | --- |
|  |  |
| a – ghi dữ liệu lên root collection | b – ghi dữ liệu lên sub-collection |

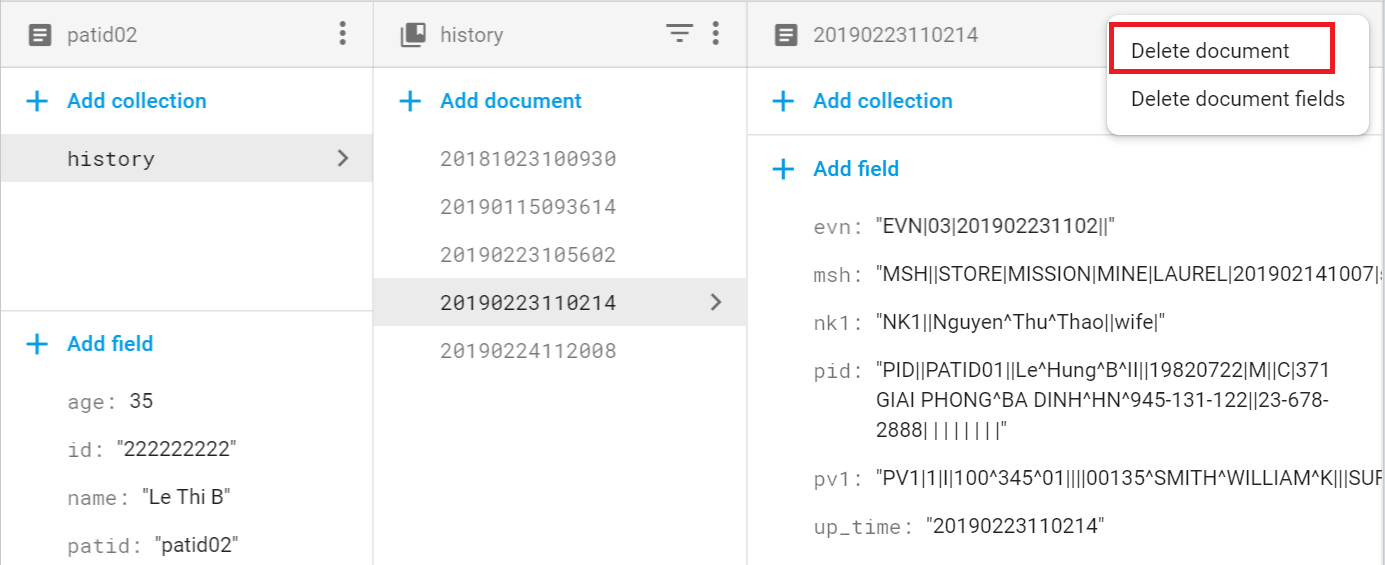
Hình Thử nghiệm ghi dữ liệu lên Firestore



Hình. Document mới hiển thị ở giao diện Web

Thử nghiệm 3: Xóa dữ liệu trên đám mây

Trên giao diện Web của Cloud Firestore, tác giả thực hiện xóa một document bất kỳ. Các thiết bị di động đồng thời cũng tự động cập nhật danh sách mới



Hình . Thử nghiệm xóa dữ liệu trên Firestore

3.4.4 Đánh giá

Kết quả thực nghiệm cho thấy, hệ thống thử nghiệm trao đổi hồ sơ y tế điện tử trên nền điện toán đám mây đáp ứng được yêu cầu về chia sẻ thông tin theo thời gian thực khi thực hiện các thao tác quản trị cơ bản như đọc / ghi / xóa dữ liệu từ thiết bị di động và trên giao diện Web của dịch vụ Cloud Firestore. Khi tài khoản có thẩm quyền sửa đổi thông tin trên một thiết bị di động và gửi lên server, thì gần như ngay lập tức, sự thay đổi được thông báo tới các thiết bị đang lắng nghe dữ liệu đó và tự động cập nhật.

Về tính bảo mật, người dùng phải có tài khoản xác thực mới có thể truy cập dữ liệu từ cơ sở dữ liệu nằm trên Firestore. Quyền xử lý đọc/ghi/xóa/cập nhật dữ liệu được phân cấp cho từng nhánh document, mỗi người dùng chỉ có thể xem và lưu thêm các sự kiện liên quan tới cá nhân họ đảm bảo sự riêng tư.

Tham khảo:

https://vi.wikipedia.org/wiki/Điện\_toán\_đám\_mây

<https://viettelidc.com.vn/tin-tuc/co-ban-ve-dien-toan-dam-may-cac-khoi-tang-va-cac-mo-hinh-dien-toan-dam-may>

<https://toidicodedao.com/2018/10/23/so-sanh-iaas-paas-saas-la-gi/>

<https://hackernoon.com/the-serverless-series-what-is-serverless-d651fbacf3f4>