**BÁO CÁO QUÁ TRÌNH**

**Xây dựng ứng dụng quản lý công việc với Serverless Service**

# Bảng danh sách công việc

|  |  |
| --- | --- |
| Thành viên | Công việc |
| Nguyễn Phúc Bình | * Tìm hiểu các service của Firebase nhằm nắm cách hoạt động cũng như ứng dụng chúng vô ứng dụng demo. * Thiết kế cơ sở dữ liệu cho hệ thống.Thiết kế Hub trong hệ thống ứng dụng. * Xử lí truy vấn dữ liệu và quản lý realtime với Firestore và quản lý người dùng với Firebase Auth. * Trang Board view trong Client web |
| Trần Thanh Hiền |

# Tìm hiểu các service của Firebase nhằm nắm cách hoạt động cũng như ứng dụng chúng vô ứng dụng demo.

Các service được cung cấp từ phía firebase gồm:

**Firebase Authentication:** Với việc hỗ trợ từ phía Firebase authentication thì có thể giúp đăng ký đăng nhập tài khoản thông qua service đã cung cấp một cách dễ dàng và có thể nhanh chóng xây dựng việc bảo mật đăng nhập cho ứng dụng demo thông qua việc đăng nhập, kiểm tra user hiện tại bên trong app và kiểm tra authentication của user một cách dễ dàng. Đồng thời với service này hỗ trợ ứng dụng có thể đăng nhập với một số hình thức khác như bằng Google.

**Firesbase Firestore:**

* Firestore là dịch vụ cung cấp cơ sở dữ liệu NoSQL cho người dùng có thể truy cập và lưu trữ theo phương pháp các collection tròng giúp dễ dàng xây dựng hệ thống có mức độ phân cấp đối tượng như trong ứng dụng mà nhóm đang xây dựng là ứng dụng phân công công việc khi các đối tượng được phân cấp thấp dần từ Board ( Bảng công việc), List ( Danh sách cộng việc), Task ( công việc) và các comment ( Bình luân). Việc xây dựng phân cấp Dữ liệu như vậy giúp ứng dụng có thể dễ dàng quản lý các đối tượng một cách dễ dàng hơn.
* Đồng thời Firestore còn cung cấp cho người dùng việc hỗ trợ cập nhật và thay đổi dữ liệu theo thời gian thực nhằm đáp ứng việc quản lý công việc sử dụng cho nhiều người sử dụng và nhận biết sự thay đổi một cách nhanh chóng.

**Firebase Storage:** Firebase Storage là nơi hỗ trợ cung cấp nơi lưu trữ các dữ liệu lớn như hình ảnh video nhằm lưu trữ các thông tin cũng như description tương ứng bên trong các task và các thông tin liên quan đến profile người dùng.

**Firebase Function:**

* Với Firebase Function giúp người dùng có thể xây dựng những hàm tương tác với các hoạt động bên ngoài hệ thống firebase hoặc có thể tùy chỉnh các cách xử lí bên trong các service khác theo một cách mà mình mong muốn.
* Nhóm sử dụng Firebase Function trong việc kiểm tra và gửi thông các thông tin mã OTP khi người dùng đăng nhập giúp có thể tùy chỉnh được quy trình đăng ký đăng nhập theo ý muốn của nhóm.

**Firebase Hosting:** Cung cấp cho lập trình viên dễ dàng host ứng dụng web lên trên firebase một cách dễ dàng sau khi hoàn thành việc xây dựng web đó đồng thời có thể kết nối đến các service khác của firebase một cách dễ dàng.

**Firebase Emulator:**

* Firebase Emulator là phần quan trọng nhất trong việc xây dựng một ứng dụng chạy bằng firebase vì đối với một serverless service như Firebase thì việc triển khai trên hệ thống sẽ tốn khá nhiều chi phí và phù hợp khi sản phẩm đã lên production và sẵn sàng chi trả những hao phí đó của dịch vụ Firebase nên khi đó Firebase Emulator sẽ cung cấp máy ảo chạy trên hệ thống local giúp cho việc triển khai cũng như thử nghiệm trong lúc xây dựng sản phẩm sẽ không bị ảnh hưởng đến chi phí phát sinh khi xây dựng sản phẩm.
* Với Emulator giúp nhóm có thể xây dựng ứng dụng một cách tốt nhất và nhanh chóng cũng như tối ưu chi phí phát triển cho ứng dụng hiện tại.

# Thiết kế cơ sở dữ liệu cho hệ thống.

Xây dựng hệ thống dữ liệu trên firebase firestore NoSQL.

Với việc hệ thống cơ sở dữ liệu Firestore được xây dựng theo cơ chế phân lớp Collection, Doc và sub Collection nên việc xây dựng cơ sở dữ liệu ở đây sẽ khác đôi chút so với lại 1 số cơ sở dữ liệu NoSQL khác như mongoDb.

1. **Users Collection**

{

"uid": "string",

"displayName": "string",

"email": "string",

"avatarURL": "string",

"boards": ["boardId1", "boardId2", ...]

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Board Collection**

{

"boardId": "string", // Unique ID

"name": "string",

"description": "string",

"ownerId": "uid", // ID of the owner

"members": ["uid1", "uid2", ...], // List of user IDs part of the board

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Lists SubCollection** (Sub Collection của Board)

{

"listId": "string",

"name": "string",

"position": "number", // Vị trí của list trong board

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Card SubCollection** (Sub Collection của List)

{

"cardId": "string",

"title": "string",

"description": "string",

"assignedTo": ["uid1", "uid2", ...], // List of user IDs assigned to this card

"labels": ["label1", "label2", ...], // Optional labels for the card

"dueDate": "timestamp",

"position": "number", // Vị trí của card trong list

"attachments": ["attachmentId1", "attachmentId2", ...], // List of attachment IDs

"comments": ["commentId1", "commentId2", ...], // List of comment IDs

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Comment SubCollection** (Sub Collection của Card)

{

"commentId": "string",

"authorId": "uid", // ID of the user who made the comment

"content": "string",

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Attachment SubCollection** (Sub Collection của Card)

{

"attachmentId": "string",

"fileName": "string",

"fileURL": "string", // URL dẫn đến file trên Firebase Storage hoặc một dịch vụ lưu trữ khác

"uploadedBy": "uid", // ID của người upload

"uploadedAt": "timestamp"

}

Mối quan hệ dữ các collection cũng như các đối tượng:

* **Users** và **Boards**: Quan hệ n-n
* **Boards** và **Lists**: Quan hệ 1-n
* **Lists** và **Cards**: Quan hệ 1-n
* **Cards** và **Comments**: Quan hệ 1-n
* **Cards** và **Attachments**: Quan hệ 1-n

# Xử lí truy vấn dữ liệu và quản lý realtime với Firestore và quản lý người dùng với Firebase Auth

* Setup cấu hình server Firebase Emulators để có thể xây dựng và xử lí các chức năng liên quan đến Firestore và Firebase Authentication trên Emulator đó.
* Code các service để truy vấn dữ liệu bên phía cơ sở dữ liệu firestore, tạo kết nối realtime giữa server và client khi có sự thay đổi của 1 đối tượng nào đó bên trong 1 collection được đăng kí realtime để kiểm tra và thay đổi dựa trên các thông tin mới của dữ liệu. Đồng thời các sự thay đổi thêm xóa sửa các đối tượng bên trong hệ thống.
* Xử lí việc kiểm tra các thông tin người dùng, đăng ký đăng nhập tài khoản và các yếu tố kiểu tra bảo mật để đảm bảo bảo mật người dùng khi truy vấn dữ liệu.

# Trang Board view trong Client web

A screenshot of a computer

Description automatically generated

Hình 1. Create First Team page 1

A screenshot of a computer

Description automatically generated

Hình 2: Create First Team page 2

A screenshot of a computer

Description automatically generated

Hình 3: Board Home page