

Lời cam đoan

Chúng tôi xin cam đoan rằng đề tài luận văn tốt nghiệp: "Xây dựng hệ thống đánh giá luận văn tốt nghiệp" là do chúng tôi thực hiện dưới sự hướng dẫn của Th.S Nguyễn Thị Ái Thảo. Tất cả các tham khảo từ các công trình khác đều được ghi rõ trong mục "Tài liệu tham khảo". Nội dung của luận văn này chưa từng được công bố trước đây dưới bất kì hình thức nào. Nếu có bất kì sai phạm nào, chúng tôi xin chịu hoàn toàn trách nhiệm trước Ban Chủ nhiệm Khoa và Ban Giám hiệu Nhà trường.

Nhóm sinh viên thực hiện

Lời cảm ơn

Trước hết, chúng tôi xin trân trọng gửi lời cảm ơn đến Th.S Trần Thị Ái Thảo - giảng viên khoa Khoa học và Kỹ thuật Máy tính (trường Đại học Bách Khoa - ĐHQG TP.HCM). Trong thời gian vừa qua, thầy đã tận tình chỉ dạy, hướng dẫn cho chúng tôi những kiến thức quý báu. Bên cạnh đó thầy cũng tạo mọi điều kiện thuận lợi để chúng tôi hoàn thành luận văn trong thời gian sớm nhất.

Bên cạnh đó, chúng tôi xin gửi lời cảm ơn đến Trường Đại học Bách Khoa - ĐHQG TPHCM, nơi đã tạo cho nhóm chúng tôi môi trường học tập tốt. Quý thầy cô khoa Khoa học và Kỹ thuật Máy tính, những người đã truyền tri thức cùng tâm huyết của mình cho nhóm chúng tôi vốn kiến thức trong suốt thời gian qua.

Lời cuối cùng, chúng tôi xin gửi lời cảm ơn sâu sắc đến gia đình, bạn bè - những người đã sát cánh bên chúng tôi, là động lực to lớn để chúng tôi hoàn thành luận văn một cách trọn vẹn nhất

Nhóm sinh viên thực hiện

Lưu trữ tài liệu

Nhóm coi đây được coi là tài liệu kỹ thuật nên quyết định dùng file Markdown (README.md là một ví dụ của markdown) để soạn thảo thay cho các công cụ soạn thảo khác.

Phiên bản mới nhất của tài liệu này được cập nhật tại github.

<https://github.com/datai999/thesis-document/tree/main/report>

Tóm tắt nội dung

Đề tài luận văn mà nhóm chúng tôi thực hiện là "Xây dựng hệ thống đánh giá luận văn tốt nghiệp".

Trong quá trình thực hiện đề tài này, nhóm chúng tôi đã thực hiện qua ba giai đoạn, cụ thể như sau :

- **Giai đoạn 1:** Tìm hiểu phân tích đề tài, đưa ra các phân tích và thiết kế sơ bộ hệ thống. Tìm hiểu các và thử nghiệm và lựa chọn công nghệ phù hợp.
- **Giai đoạn 2:** Xây dựng hệ thống với các tính năng cơ bản. Triển khai hệ thống đáp ứng trên nền tảng web.
- **Giai đoạn 3:** Mở rộng các tính năng của hệ thống. Mở rộng sang nền tảng ios và android. Mở rộng đối tượng sử dụng.

Báo cáo đề cương luận văn tốt nghiệp của nhóm sẽ trình bày thông qua các nội dung chính sau đây:

- **Chương 1:** Tổng quan giới thiệu đề tài
- **Chương 2:** Phân tích yêu cầu đề tài
- **Chương 3:** Lựa chọn công nghệ
- **Chương 4:** Lập trình UI web, android, ios với Expo
- **Chương 5:** Thiết kế kiến trúc hệ thống
- **Chương 6:** Hiện thực hệ thống
- **Chương 7:** Tổng kết và hướng phát triển

Mục lục

- 1 [Tổng quan giới thiệu đề tài](#) [9]
 - 1.1 Giới thiệu [10]
 - 1.2 Các vấn đề hiện tại [10]
 - 1.3 Mục tiêu của đề tài [11]
 - 1.4 Phạm vi của đề tài [11]
 - 1.5 Khó khăn và thử thách [11]
- 2 [Phân tích yêu cầu đề tài](#) [12]
 - 2.1 Phân chia người dùng [13]
 - 2.2 Tính năng của hệ thống [13]
- 3 [Lựa chọn công nghệ](#) [14]
 - 3.1 Các loại ứng dụng [15]
 - 3.1.1 Web App và Web-base App [15]
 - 3.1.2 Native App, Hybrid App và Cross Platform App [17]
 - 3.1.3 Kết luận và lựa chọn loại ứng dụng [21]
 - 3.2 Ứng dụng đa trang và Ứng dụng đơn trang [22]
 - 3.2.1 Ứng dụng đa trang [22]
 - 3.2.2 Ứng dụng đơn trang [22]
 - 3.2.3 Kết luận lựa chọn kiểu ứng dụng [22]
 - 3.3 Ngôn ngữ lập trình front-end [23]
 - 3.3.1 JavaScript [23]
 - 3.3.2 TypeScript [24]
 - 3.3.3 Dart [25]
 - 3.3.4 Kết luận lựa chọn ngôn ngữ front-end [25]
 - 3.4 Frame-work front-end [26]
 - 3.4.1 React [26]
 - 3.4.2 Angular [27]
 - 3.4.3 Vue [28]
 - 3.4.4 Lựa chọn frame-work front-end [29]
 - 3.5 Ngôn ngữ lập trình và frame-work back-end [30]
 - 3.5.1 Java và Spring-boot [30]
 - 3.5.2 C# và ASP .NET [31]
 - 3.5.3 Python và Django [32]
 - 3.5.4 JavaScript và ExpressJS [33]
 - 3.5.5 Lựa chọn ngôn ngữ lập trình và frame-work [34]
 - 3.6 Cơ sở dữ liệu [35]
 - 3.6.1 Cơ sở dữ liệu quan hệ [36]
 - 3.6.2 Cơ sở dữ liệu hướng tài liệu [37]
 - 3.6.3 Cơ sở dữ liệu dạng khóa - giá trị [38]
 - 3.6.4 Cơ sở dữ liệu mô hình wide – colum [39]
 - 3.6.5 Cơ sở dữ liệu dạng bộ máy tìm kiếm [40]
 - 3.6.6 Lựa chọn cơ sở dữ liệu [41]

- 4 [Lập trình UI web, android, ios với Expo](#) [42]
 - 4.1 Hoàn cảnh và ý tưởng [43]
 - 4.1.1 Từ web đến di động [43]
 - 4.1.2 Từ di động đến web [43]
 - 4.2 Expo [45]
 - 4.2.1 Giới thiệu Expo [45]
 - 4.2.2 Chạy thử ứng dụng từ Expo [46]
- 5 [Thiết kế kiến trúc hệ thống](#) [47]
 - 5.1 Thiết kế cơ sở dữ liệu [48]
 - 5.1.1 Nền tảng thiết kế cơ sở dữ liệu [48]
 - 5.1.2 Các bảng nhóm khác (br) [51]
 - 5.1.3 Các bảng nhóm nhân sự (ps) [53]
 - 5.1.4 Các bảng nhóm đề tài (tp) [56]
 - 5.1.5 Các bảng nhóm điểm (sc) [61]
 - 5.2 Thiết kế API [65]
 - 5.2.1 Các API cơ sở [65]
 - 5.2.2 Các API khác [66]
 - 5.3 Thiết kế luồng giao diện [67]
 - 5.4 Thiết kế một số chức năng [68]
 - 5.4.1 Chức năng xác thực người dùng [68]
 - 5.4.2 Chức năng xác thực các yêu cầu từ người dùng [69]
 - 5.4.3 Chức năng thêm dữ liệu [70]
- 6 [Hiện thực hệ thống](#) [71]
 - 6.1 Lưu trữ [72]
 - 6.2 Các môi trường [74]
 - 6.3 Chia nhiệm vụ [75]
 - 6.4 Filter và Interception request [76]
 - 6.5 Code back-end [77]
 - 6.5.1 Phân chia source code [77]
 - 6.5.2 Modal class diagram [77]
 - 6.5.3 Service class diagram [78]
 - 6.5.4 Controller class diagram [78]
 - 6.6 CI-CD [79]
- 7 [Tổng kết và hướng phát triển](#) [80]
 - 7.1 Kết quả đạt được [81]
 - 7.2 Thuận lợi - Khó khăn [82]
 - 7.3 Hướng phát triển [82]

Danh sách bảng

- Chương 3: Lựa chọn công nghệ
 - Bảng 3.1.3: Lựa chọn loại ứng dụng [21]
- Chương 4: Lập trình UI web, android, ios với Expo
 - Bảng 4.1.2: So sánh React Native + React Native Web với Expo [44]
- Chương 5: Thiết kế kiến trúc hệ thống
 - 5.1: Thiết kế cơ sở dữ liệu
 - 5.1.1: Nền tảng thiết kế cơ sở dữ liệu
 - Bảng 5.1.1.c: Đặc tả chi tiết bảng BaseTable [49]
 - Bảng 5.1.1.d: Ví dụ về đa ngôn ngữ bằng json [49]
 - Bảng 5.1.1.e.1: Giảm dư thừa dữ liệu bằng array [50]
 - Bảng 5.1.1.e.2: Kết quả giảm dư thừa dữ liệu bằng array [50]
 - 5.1.2: Các bảng nhóm khác (br)
 - Bảng 5.1.2.a.1: Đặc tả chi tiết bảng br_const_data [51]
 - Bảng 5.1.2.a.2: Ví dụ về bảng br_const_data [51]
 - Bảng 5.1.2.b.1: Đặc tả chi tiết bảng br_setting [52]
 - Bảng 5.1.2.b.2: Chuẩn bị bảng br_const_data [52]
 - Bảng 5.1.2.b.3: Ví dụ về bảng br_setting [52]
 - 5.1.3: Các bảng nhóm nhân sự (ps)
 - Bảng 5.1.3.a: Đặc tả chi tiết bảng PersonBaseTable [53]
 - Bảng 5.1.3.b: Đặc tả chi tiết bảng ps_teacher [54]
 - Bảng 5.1.3.c: Đặc tả chi tiết bảng ps_student [54]
 - Bảng 5.1.3.d.1: Chuẩn bị bảng br_const_data [54]
 - Bảng 5.1.3.d.2: Ví dụ về bảng PersonBaseTable [54]
 - Bảng 5.1.3.d.3: Chuẩn bị bảng br_const_data [55]
 - Bảng 5.1.3.d.4: Ví dụ về bảng ps_academy_staff [55]
 - Bảng 5.1.3.d.5: Ví dụ về bảng ps_teacher [55]
 - Bảng 5.1.3.d.6: Ví dụ về bảng ps_student [55]
 - 5.1.4 Các bảng nhóm đề tài (tp)
 - Bảng 5.1.4.a: Đặc tả chi tiết bảng tp_topic [57]
 - Bảng 5.1.4.b: Đặc tả chi tiết bảng tp_council [58]
 - Bảng 5.1.4.c: Đặc tả chi tiết bảng tp_topic_assign [58]
 - Bảng 5.1.4.d.1: Chuẩn bị bảng br_const_data [59]
 - Bảng 5.1.4.d.2: Ví dụ về bảng tp_topic [60]
 - Bảng 5.1.4.d.3: Ví dụ về bảng tp_council [60]
 - Bảng 5.1.4.d.4: Ví dụ về bảng tp_topic_assign [60]
 - 5.1.5 Các bảng nhóm điểm (sc)
 - Bảng 5.1.5.a: Đặc tả chi tiết bảng sc_criterion [62]
 - Bảng 5.1.5.b: Đặc tả chi tiết bảng sc_criterion_template [62]
 - Bảng 5.1.5.c: Đặc tả chi tiết bảng sc_score [62]
 - Bảng 5.1.5.d.1: Chuẩn bị bảng br_const_data [63]
 - Bảng 5.1.5.d.2: Chuẩn bị bảng br_setting [63]

- Bảng 5.1.5.d.3: Ví dụ về bảng sc_criterion [63]
 - Bảng 5.1.5.d.4: Ví dụ về bảng sc_criterion_template [63]
 - Bảng 5.1.5.d.5: Chuẩn bị bảng br_const_data [64]
 - Bảng 5.1.5.d.6: Chuẩn bị bảng tp_council [64]
 - Bảng 5.1.5.d.7: Chuẩn bị bảng tp_topic_assign [64]
 - Bảng 5.1.5.d.8: Bảng tính toán điểm số [64]
 - Bảng 5.1.5.d.9: Ví dụ về bảng sc_score [64]
- 5.2: Thiết kế API
 - Bảng 5.2.1: Các API cơ bản [65]
 - Bảng 5.2.2: Các API khác [66]
- Chương 6: Hiện thực hệ thống
 - Bảng 6.3: Phân chia nhiệm vụ [75]

Danh sách sơ đồ

- Chương 5: Thiết kế kiến trúc hệ thống
 - Sơ đồ 5.1.1.b: Mối quan hệ Foreign key giữa các nhóm table [48]
 - Sơ đồ 5.1.2: ERD diagram của nhóm br [51]
 - Sơ đồ 5.1.3: ERD diagram của nhóm ps [53]
 - Sơ đồ 5.1.4: ERD diagram của nhóm tp [56]
 - Sơ đồ 5.1.5: ERD diagram của nhóm sc [61]
 - Sơ đồ 5.3: UI flow chart [67]
 - Sơ đồ 5.4.1: Login sequence diagram [68]
 - Sơ đồ 5.4.2: Valid request sequence diagram [69]
 - Sơ đồ 5.4.3: Import data sequence diagram [70]
- Chương 6: Hiện thực hệ thống
 - Sơ đồ 6.2: Environment diagram [74]
 - Sơ đồ 6.4: Valid request implement sequence diagram [76]
 - 6.5 Code back-end
 - Sơ đồ 6.5.1: Phân chia source code [77]
 - Sơ đồ 6.5.2: Modal class diagram [77]
 - Sơ đồ 6.5.3: Service class diagram [78]
 - Sơ đồ 6.5.4: Controller class diagram [78]
 - Sơ đồ 6.6: Triển khai CI-CD [79]

Danh sách hình ảnh

- Hình 1: Tổng quan giới thiệu đề tài [9]
- Hình 2: Phân tích yêu cầu đề tài [12]
- Hình 3: Lựa chọn công nghệ [14]
 - Hình 3.1.1.a: Web Application [15]
 - Hình 3.1.2.b: Web-base app: m.facebook.com [16]
 - Hình 3.1.2: Native App, Hybrid App và Cross Platform App [17]
 - Hình 3.1.2.a: Native App [18]
 - Hình 3.1.2.b: Hybrid App [19]
 - Hình 3.1.2.c: Cross Platform App [20]
 - Hình 3.3.1: Logo JavaScript [23]
 - Hình 3.3.2: Logo TypeScript [24]
 - Hình 3.3.3: Logo Dart [25]
 - Hình 3.4.1: Logo Frame-work React [26]
 - Hình 3.4.2: Logo Frame-work Angular [27]
 - Hình 3.4.3: Logo Frame-work Vue [28]
 - Hình 3.4.4.a: Frame-work Angular vs Vue vs React [29]
 - Hình 3.4.4.b: Dùng React-native code web [29]
 - Hình 3.5.1.a: Logo Java [30]
 - Hình 3.5.1.b: Logo Spring-boot [30]
 - Hình 3.5.2.a: Logo C# [31]
 - Hình 3.5.2.b: Logo ASP .NET [31]
 - Hình 3.5.3.a: Logo Python [32]
 - Hình 3.5.3.b: Logo Django [32]
 - Hình 3.5.4.a: Logo JavaScript [33]
 - Hình 3.5.4.b: Logo Express [33]
 - Hình 3.6: Phân loại cơ sở dữ liệu [35]
 - Hình 3.6.2.a: Logo Mongo [37]
 - Hình 3.6.2.b: Logo Couchbase [37]
 - Hình 3.6.3: Logo Redis vs Logo Memcached [38]
 - Hình 3.6.4: Logo Cassandra vs Logo Hbase [39]
 - Hình 3.6.5: Logo Elasticsearch [40]
 - Hình 3.6.6: Logo PostgreSQL [41]
- Hình 4: Lập trình UI web, android, ios với Expo [42]
 - Hình 4.1.2: React Native Web [43]
 - Hình 4.2.1: Logo hệ sinh thái Expo [45]
 - Hình 4.2.2: Màn hình web(trái) và Màn hình terminal (phải) khi start Expo [46]
- Hình 5: Thiết kế kiến trúc hệ thống [47]
- Hình 6: Hiện thực hệ thống [71]
 - Hình 6.1.a Logo Markdown [72]
 - Hình 6.1.b Logo Heroku [73]
 - Hình 6.1.c Logo Firebase [73]
 - Hình 6.1.d Expo web Builds [73]
 - Hình 6.2 Trang web Expo app [74]
- Hình 7: Tổng kết và hướng phát triển [80]

1. Tổng quan giới thiệu đề tài

Đầu tiên, nhóm sẽ trình bày các vấn đề hiện tại từ đó giới thiệu về lý do hình thành đề tài. Theo đó nhóm sẽ mô tả mục tiêu, phạm vi của đề tài cũng như những khó khăn và thử thách nhóm thấy được trong những bước ban đầu nhận được đề tài này.



Hình 1: Tổng quan giới thiệu đề tài

1.1 Giới thiệu

Năm 2020-2021 là những năm đầy biến động, khó khăn và những thách thức lớn, đại dịch COVID-19 đã làm thay đổi xã hội, thay đổi nền kinh tế, giáo dục của nhiều nước trên thế giới, mọi người ở nhà, tiếp xúc các thiết bị Internet nhiều hơn, làm việc ở nhà, đi chợ online, học online,...

Việc số hóa tài liệu trở nên cần thiết hơn. Cần công nghệ hóa các phương pháp làm việc cũng như các dịch vụ hoạt động ổn định với lượng người dùng chuyển đột ngột từ môi trường thực tại qua internet.

Qua thực tế nhóm em thấy rằng hệ thống đánh giá luận văn của trường Đại học Bách khoa - Đại học Quốc gia TP.HCM cần được phát triển để có thể đáp ứng với nhiều khó khăn trong tương lai hơn. Đó là lý do chính nhóm em đề xuất đề tài "Xây dựng hệ thống đánh giá luận văn tốt nghiệp".

1.2 Các vấn đề hiện tại

Hệ thống đánh giá luận văn của trường Đại học Bách khoa - Đại học Quốc gia TP.HCM hiện tại đang lưu trữ và sử dụng dữ liệu dưới dạng các file Microsoft Excel, Microsoft Word và một số tài liệu giấy. Cách thức hoạt động này được triển khai từ nhiều năm trước với những ưu điểm như sau:

- Vì sử dụng các phần mềm phổ biến nên nhân viên cũ và mới đều có thể tương tác dễ dàng
- Dễ mở rộng nếu có thêm các nghiệp vụ mới.
- Dễ tạo báo cáo, thống kê bằng Microsoft Word, Microsoft Excel.
- Nhân viên mới dễ dàng làm quen và sử dụng hệ thống.

Tuy nhiên, cách hoạt động này chỉ tỏ ra hiệu quả khi hệ thống còn ít người dùng, dữ liệu còn nhỏ, nghiệp vụ ít, đơn giản. Ở thời điểm hiện tại, khi lượng dữ liệu và các nghiệp vụ ngày càng nhiều, giãn cách xã hội không còn dễ dàng trao đổi trực tiếp với nhau thì cách thức hoạt động này thể hiện những hạn chế như sau:

- Việc trao đổi thông tin không phù hợp với mùa dịch.
- Không có cơ chế phân quyền có thể dẫn đến các thao tác vượt quá quyền hạn, khó dò lỗi.
- Khó thao tác với các thiết bị di động.
- Mọi thông báo đều phụ thuộc ở mail
- Dữ liệu ngày càng nhiều, khó lưu trữ, quản lý cũng như truy suất

1.3 Mục tiêu của đề tài

Mục tiêu của đề tài này là xây dựng một hệ thống (website, mobile app) hỗ trợ đánh giá luận văn tốt nghiệp của Đại học Bách Khoa - Đại học Quốc gia TP.HCM với các tính năng cơ bản như sau:

- Tạo và quản lý thông tin của đề tài
- Gán sinh viên/ nhóm sinh viên thực hiện đề tài
- Tạo giảng viên hướng dẫn, giảng viên phản biện cho đề tài
- Tạo Hội đồng chấm luận văn
- Tạo các tiêu chí đánh giá cho sinh viên.
- Chấm điểm cho từng sinh viên trong nhóm luận văn. (Phân biệt điểm GVHD, GVPB, và các thành viên hội đồng)

1.4 Phạm vi đề tài

Phạm vi đối tượng mà đề tài hướng đến là:

- Cán bộ giáo vụ trường Đại học Bách Khoa - Đại học Quốc gia TP.HCM
- Giảng viên trường Đại học Bách Khoa - Đại học Quốc gia TP.HCM

1.5 Khó khăn thử thách

Các khó khăn, thử thách mà đề tài gặp phải

- Khó khăn khách quan:
 - Số lượng nghiệp vụ lớn, thường xuyên thay đổi
 - Đầy đủ các chức năng nhưng vẫn phải đảm bảo dễ dàng sử dụng
 - Hiện thực trên nhiều nền tảng (web, ios, android)
- Khó khăn chủ quan:
 - Đội ngũ phát triển một thành viên
 - Đội ngũ phát triển không có kinh nghiệm front-end

2. Phân tích yêu cầu đề tài

Software requirement là tài liệu miêu tả những yêu cầu của khách hàng về sản phẩm phần mềm, những hành vi của đối tượng trong sản phẩm đó, đó là những yêu cầu về chức năng hoặc phi chức năng mà sản phẩm phần mềm cần đáp ứng được.

Việc hiểu đúng Software requirement rất quan trọng trong các bước để phát triển phần mềm. Phân tích requirement sơ sài dẫn tới việc sản phẩm cuối cùng không đáp ứng được yêu cầu, thậm chí là sai lệch hoàn toàn mong muốn.

Hiểu rõ những điểm quan trọng trong việc phân tích requirement, ngay sau khi nhận đề tài, việc đầu tiên nhóm làm là phân tích requirement từ đó đề ra một số tính năng cơ bản mà hệ thống cần đáp ứng được.



Hình 2: Phân tích yêu cầu đề tài

2.1 Phân chia người dùng

Dựa vào những yêu cầu thực tế của hệ thống và những khảo sát thực tế đối với các cá nhân có nhu cầu sử dụng. Nhóm đưa ra các vai trò, người dùng chính trong hệ thống như sau:

- Quản trị hệ thống
- Giáo vụ
- Giảng viên

2.2 Tính năng của hệ thống

- Hoạt động trên 3 nền tảng web, ios, android
- Quản lý thông tin nhân sự (Giáo vụ, Giảng viên, Sinh viên)
 - Thêm, sửa, xóa, lưu trữ, tìm kiếm
 - Nhập liệu từ file Excel
 - Xuất dữ liệu
- Quản lý thông tin đề tài
 - Thêm, sửa, xóa, lưu trữ, tìm kiếm
 - Nhập liệu từ file Excel
 - Xuất dữ liệu
 - Gán sinh viên thực hiện
 - Gán giảng viên hướng dẫn, giảng viên phản biện, hội đồng
- Quản lý thông tin hội đồng
 - Thêm, sửa, xóa, lưu trữ, tìm kiếm
 - Xuất dữ liệu
 - Thêm thành viên hội đồng
 - Gán hội đồng cho đề tài
- Quản lý điểm
 - Tạo tiêu chí, mẫu tiêu chí để chấm điểm
 - Tạo phương thức chấm điểm (ABCD, thang điểm 10, thang điểm 100,...)
 - Chấm điểm cho từng sinh viên
- Xuất các báo cáo, thống kê
 - Xuất trên màn hình
 - Xuất file Excel, Word

3. Lựa chọn công nghệ

Lựa chọn công nghệ cần dựa trên 3 yếu tố

- Dự án phù hợp với công nghệ nào
- Công nghệ có phù hợp với hiện tại và tương lai
- Tài nguyên hiện có: đội ngũ phát triển, vốn, ...

Để lựa chọn được công nghệ đáp ứng được 3 yếu tố trên, nhóm đã cố gắng đọc tài liệu cũng như trải nghiệm sơ qua để có được sự lựa chọn ưng ý nhất. Sau khi tổng hợp lại, bỏ qua những framework nhỏ, nhóm sẽ giới thiệu qua những công nghệ chính mà nhóm sẽ đưa lên bàn cân trong chương này.



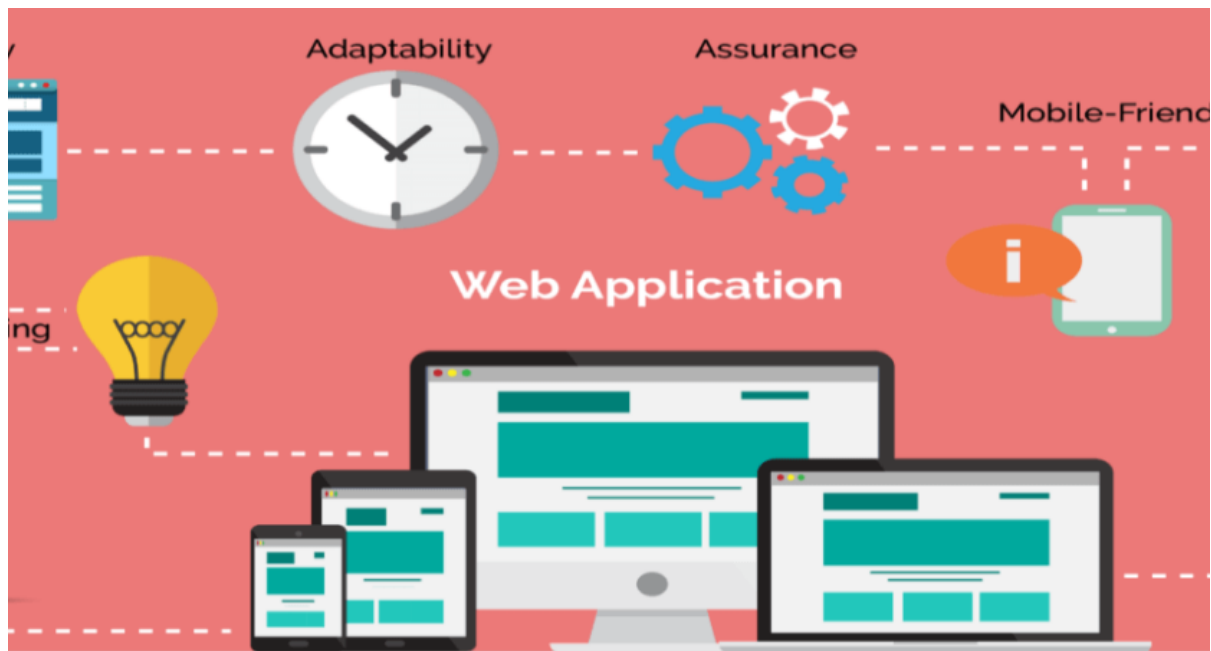
Hình 3: Lựa chọn công nghệ

3.1 Các loại ứng dụng

3.1.1 Web App và Web-base App

3.1.1.a Web App

Ứng dụng web chính là một phần mềm ứng dụng nền tảng web để chạy các phần mềm theo mong muốn và nhu cầu của người sử dụng. Thông qua ứng dụng web người dùng có thể thực hiện được nhiều công việc khác nhau. Web app có tính tương tác cao hơn website rất nhiều.



Hình 3.1.1.a: Web Application

Ưu điểm:

- Tương thích với nhiều nền tảng.
- Không cần cài đặt trên máy.
- Rất phổ biến, được hỗ trợ mạnh mẽ.
- Chỉ cần deploy lên server và sử dụng

Nhược điểm:

- Phụ thuộc vào trình duyệt.
- Không thể hỗ trợ offline.
- Muốn truy cập người dùng phải nhớ link.

Nên dùng khi:

- Dự án có các nghiệp vụ phức tạp.
- Dự án cần mở rộng về sau.

Với các nghiệp vụ đơn giản như quảng cáo, ra một sản phẩm mới, phát hành miễn phí sản phẩm trong thời gian ngắn hạn thì nên dùng website thuần túy hoặc landing page.

3.1.1.b Web-base App

Web-based app – Ứng dụng web: Là ứng dụng chạy trên nền web, được viết bằng các ngôn ngữ web như HTML5, CSS, javascript hoặc jQuery Mobile. Về cơ bản là website với giao diện giống như app và nội dung được load từ web. Ứng dụng web dễ hình dung nhất là m.facebook.com



Hình 3.1.2.b: Web-base app: m.facebook.com

Ưu điểm:

- Tương thích với nhiều nền tảng.
- Không cần cài đặt trên máy.
- Tiết kiệm tài nguyên và nhân lực.
- Chỉ cần deploy lên server và sử dụng

Nhược điểm:

- Phụ thuộc vào trình duyệt.
- Tốc độ xử lý chậm.
- Không mang lại trải nghiệm trọng vẹn.
- Không thể hỗ trợ offline.
- Muốn truy cập người dùng phải nhớ link.

Nên dùng khi:

- Dự án có nguồn lực và kinh phí thấp.
- Ứng dụng không đăng tải lên các store.
- Nhà phát triển có kinh nghiệm thuần web.

3.1.2 Native App, Hybrid App và Cross Platform App

Nhắc đến lập trình ứng dụng di động hiện nay thì ba thuật ngữ quan trọng không thể bỏ qua đó là Native App, Hybrid App và Cross Platform App. Mỗi công nghệ có những ưu điểm và nhược điểm khác nhau cần cân nhắc để lựa chọn phù hợp với dự án.



Hình 3.1.2: Native App, Hybrid App và Cross Platform App

3.1.2.a Native App

Native app thường được gọi là ứng dụng gốc. Là những ứng dụng được viết riêng cho một loại nền tảng như iOS, Android, Windows Phone bằng các ngôn ngữ tương ứng.



Hình 3.1.2.a: Native App

Ưu điểm:

- Về mặt performance thì native app chạy nhanh đặc biệt trong game.
- Truy cập phần cứng mạnh mẽ.
- Có nhiều lợi thế ở chế độ offline.

Nhược điểm:

- Không hỗ trợ đa nền tảng.
- Tốn nhân lực và tài nguyên khi dự cần cần đa nền tảng.

Nên dùng khi:

- Ứng dụng cần sự ăn khớp với nền tảng cao.
- Ứng dụng cần truy cập sâu vào quản lí phần cứng.
- Ứng dụng yêu cầu UI cao hoặc graphic animation như game.

3.1.2.b Hybrid App

Hybrid App là tên gọi cho các ứng dụng lai. Ứng dụng lai được tạo nên từ ba công nghệ front end bao gồm HTML, CSS và JavaScript. Lập trình viên sẽ tạo một khung rỗng chứa web và được chạy ở thiết bị di động.



Hình 3.1.2.b: Hybrid App

Ưu điểm:

- Có được sự kế thừa từ Native App và Cross Platform App.
- Tiết kiệm tài nguyên và nhân lực.
- Dễ bảo trì và phát triển.
- Tương thích với nhiều nền tảng.

Nhược điểm:

- Giao diện phụ thuộc quá nhiều vào web bên trong khung chứa.
- Tốc độ xử lý không nhanh bằng Native App và Cross Platform App.

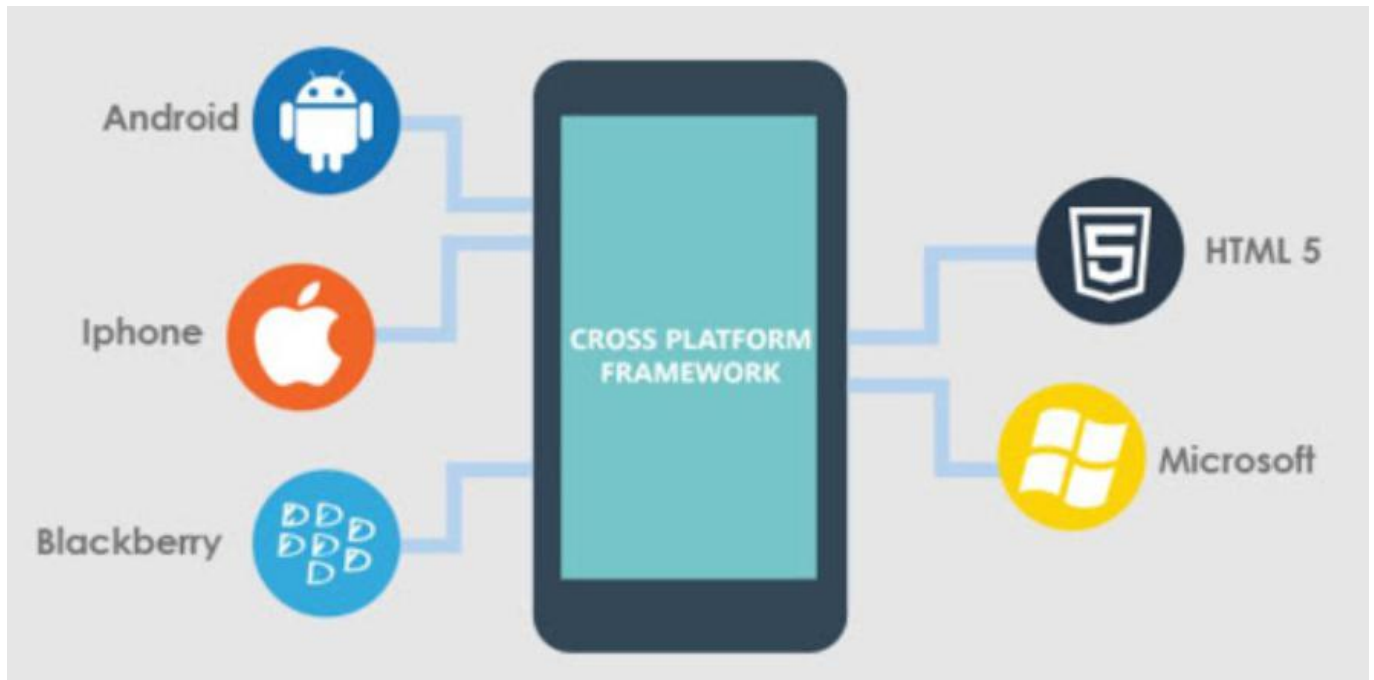
Nên dùng khi:

- Ứng dụng cần xuất bản trên nhiều nền tảng.
- Nhà phát triển có kinh nghiệm thuần web.

3.1.2.c Cross Platform App

Cross Platform còn một tên gọi khác là Multi Platform Thuật ngữ này dùng để chỉ các ứng dụng đa nền tảng, phù hợp để cài đặt cho nhiều loại thiết bị khác nhau. Đây được xem là giải pháp hiệu quả nhất để giải quyết các vấn đề khi sử dụng Native app.

Với Cross Platform, nhà phát triển chỉ cần sử dụng một ngôn ngữ và lập trình ứng dụng một lần. Chúng sẽ được biên dịch sang các phiên bản khác để phù hợp cho những nền tảng khác nhau



Hình 3.1.2.c: Cross Platform App

Ưu điểm:

- Tối ưu thời gian và chi phí phát triển.
- Dễ bảo trì và phát triển.
- Tương thích với nhiều nền tảng.

Nhược điểm:

- Nhiều vấn đề phát sinh từ frame-work khi phải biên dịch chuyển đổi giữa các ngôn ngữ.

Nên dùng khi:

- Ứng dụng cần xuất bản trên nhiều nền tảng.
- Ít thời gian và nhân lực cũng như chi phí phát triển.

3.1.3 Kết luận và lựa chọn loại ứng dụng

Với nhiều sự lựa chọn, nhóm sẽ tạo bảng đánh giá thang điểm 5 dựa trên các tiêu chí như sau

Bảng 3.1.3: Lựa chọn loại ứng dụng

Tiêu chí	Web App	Web-base App	Native App	Hybrid App	Cross Platform App
Hiệu năng	5	3	5	3	4
Trải nghiệm người dùng	5	3	5	3	5
Được hỗ trợ tốt từ cộng đồng	5	2	5	3	4
Chạy được trên web	5	2	0	0	3
Chạy được trên ios, android	2	4	2	4	5
Ít tốn nhân lực	3	5	1	4	3
Đội ngũ phát triển có kinh nghiệm	5	3	2	1	4
Tổng điểm	29	22	20	18	28

Qua bảng đánh nhóm chọn Web App và Cross Platform App.

3.2 Ứng dụng đa trang và Ứng dụng đơn trang

Khi phát triển một ứng dụng, có hai kiểu thiết kế phổ biến hiện nay là ứng dụng đa trang (Multi-page Application-MPA) và ứng dụng đơn trang (Single-page Application-SPA). Mỗi kiểu thiết kế có những ưu và nhược điểm riêng phù hợp với các ứng dụng khác nhau. Vì vậy để có thể phát triển ứng dụng một cách hiệu quả, nhà phát triển phải cân nhắc lựa chọn cách thiết kế phù hợp nhất với nhu cầu ứng dụng của mình.

3.2.1 Ứng dụng đa trang

Ứng dụng đa trang (MPA) là một web-app hay một website chứa nhiều trang liên kết và các trang con, được điều hướng bằng menu. Ứng dụng đa trang hoạt động theo kiểu "truyền thống", các thay đổi như hiển thị dữ liệu sẽ được thực hiện bằng cách hiển thị một trang mới trong trình duyệt.

Hiện nay, ứng dụng đa trang được sử dụng rộng rãi trong nhiều lĩnh vực như thương mại điện tử

Ưu điểm:

- Cho phép khả năng mở rộng ứng dụng không giới hạn thông qua menu.
- Luồng điều hướng dễ dàng theo dõi.
- Hỗ trợ tốt cho SEO, các ứng dụng đa trang dễ dàng phân cấp.

Nhược điểm:

- Các ứng dụng đa trang có nội dung lớn tải chậm, ảnh hưởng đến trải nghiệm của người dùng.
- Khó thích nghi tốt với thiết bị di động.

3.2.2 Ứng dụng đơn trang

Ứng dụng đơn trang (SPA) là một web-app hay một website tương tác với người dùng bằng cách tải lại một phần của trang hiện tại thay vì tải toàn bộ trang mới từ máy chủ. Cách hoạt động này hạn chế sự gián đoạn trải nghiệm của người dùng khi chuyển giữa các trang, giúp cho việc trải nghiệm ứng dụng gần giống với các desktop application.

Ngày nay, ứng dụng đơn trang được sử dụng rộng rãi trên toàn thế giới, đặc biệt phù hợp các ứng dụng với lượng truy cập lớn, yêu cầu tốc độ cao như mạng xã hội, email, map. Một số ứng dụng sử dụng SPA nổi tiếng như Facebook, Instagram, Google Map, Gmail,...

Ưu điểm:

- Tốc độ nhanh.
- Có khả năng làm việc với cache tốt, nên hiệu quả khi sử dụng ở chế độ offline.
- Thích nghi tốt với thiết bị di động.

Nhược điểm:

- Không tối ưu hoá cho SEO tốt (các công cụ tìm kiếm). Việc tìm kiếm các nội dung trong ứng dụng đơn trang bằng các công cụ tìm kiếm (google.com, bing.com) thường ít hiệu quả.
- Khó khăn trong việc mở rộng.

3.2.3 Kết luận lựa chọn kiểu ứng dụng

Nhóm sẽ chọn ứng dụng đơn trang để phù hợp chuyển đổi giữa 2 nền tảng web và di động.

3.3 Ngôn ngữ lập trình front-end

Ngôn ngữ lập trình front-end phổ biến nhất hiện nay là JavaScript, bản nâng cao của nó là TypeScript.

Ngoài ra với sự mạnh mẽ đến từ framework Flutter kéo theo Dart cũng trở thành một sự lựa chọn cần cân nhắc đến.

3.3.1 JavaScript

JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới được tạo ra vào năm 1995.



Hình 3.3.1: Logo JavaScript

Ưu điểm:

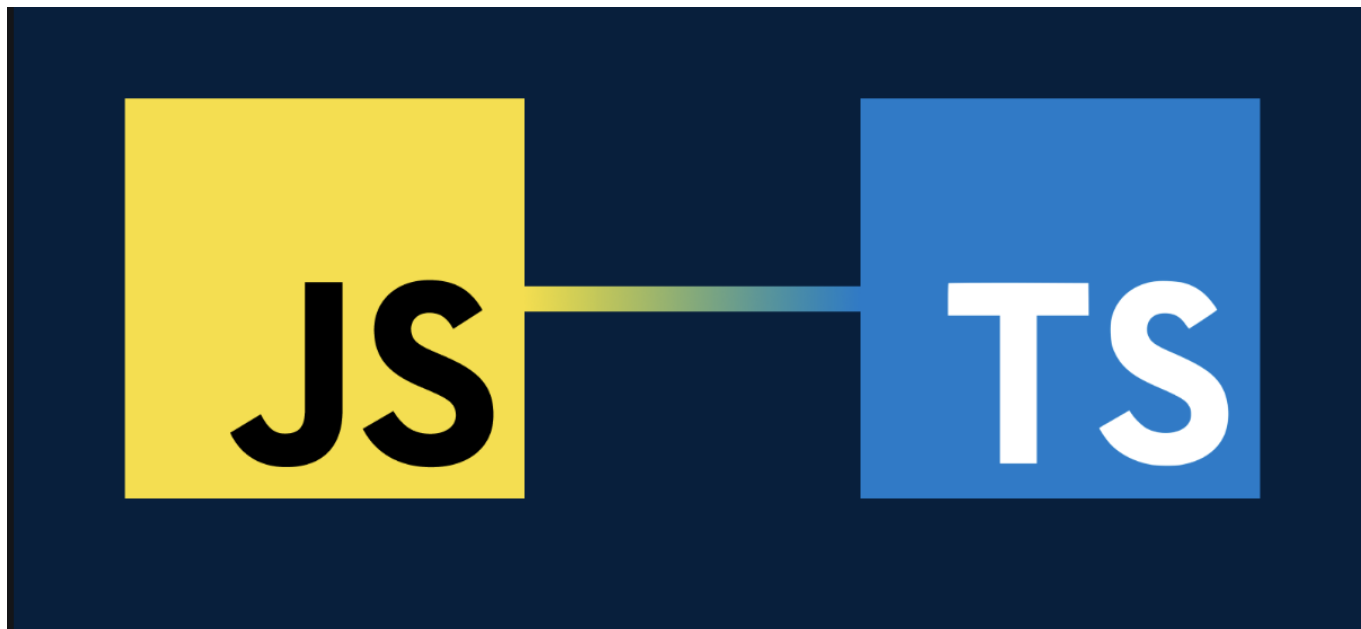
- Không cần một compiler vì web browser có thể biên dịch nó bằng HTML.
- Nó dễ học hơn các ngôn ngữ lập trình khác.
- Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn.
- Hoạt động trên nhiều trình duyệt, nhiều nền tảng.
- Giúp website tương tác tốt hơn với khách truy cập.
- Nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

Nhược điểm:

- Dễ bị khai thác
- Có thể được dùng để thực thi mã độc trên máy tính của người dùng;
- JS code snippets lớn
- Có thể triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.
- Khó có sự kiểm soát về mặt design code.
- Không hỗ trợ lập trình hướng đối tượng.

3.3.2 TypeScript

TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft, nó có thể được coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript



Hình 3.3.2: Logo TypeScript

Ưu điểm:

- Dễ phát triển ứng dụng lớn.
- Hỗ trợ các tính năng của bản JavaScript mới nhất.
- Dễ dàng design và cấu trúc hệ thống.
- Thừa hưởng hầu như tất cả ưu điểm của JavaScript.

Nhược điểm:

- Phải biên dịch.
- TypeScript không phải là ngôn ngữ độc lập, nó phụ thuộc vào JavaScript.
- Khó tiếp cận hơn nhiều so với JavaScript.

3.3.3 Dart

Dart là một ngôn ngữ lập trình đa mục đích của Google có thể dùng để phát triển web ui, server và cả ứng dụng di động.



Hình 3.3.3: Logo Dart

Ưu điểm:

- Biên dịch nhanh.
- Dễ học với những lập trình viên biết C++, C# và Java.
- Code UI nhanh và đẹp.
- Rất phù hợp với lập trình Reactive.
- Code được sử dụng ở nhiều nền tảng Web UI, Server, Cross Platform App.
- Tài liệu tập trung duy nhất tại một trang.

Nhược điểm:

- Ít sự lựa chọn frame-work, library hơn JavaScript.
- Dù đã release 1.0 chính thức, tuy nhiên còn khá mới. Một số plugin rất quan trọng như Google Map vẫn còn đang phát triển.
- Google đã có thời gian "bỏ con giữa chợ" khi Flutter chưa có sự ảnh hưởng nhất định.

3.3.4 Kết luận và lựa chọn ngôn ngữ front-end

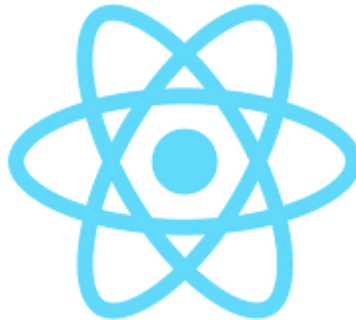
Nhóm quyết định lựa chọn JavaScript với những lý do sau:

- Đội ngũ phát triển không có kinh nghiệm với TypeScript.
- JavaScript có frame-work Expo (React Native) có thể làm Web UI, Cross Platform App giống Flutter.
- Cộng đồng hỗ trợ lâu đời và mạnh mẽ.
- Thị trường nhiều lập trình viên JavaScript hơn, dễ kế thừa và phát triển về sau.

3.4 Frame-work front-end

3.4.1 React

React là một thư viện JavaScript, được Facebook ra mắt từ năm 2013. Đây là một thư viện tuyệt vời để xây dựng các ứng dụng web lớn, nơi dữ liệu có thể thay đổi thường xuyên



Hình 3.4.1: Logo Frame-work React

Ưu điểm:

- Dễ học, nhờ thiết kế đơn giản, sử dụng JSX (một cú pháp giống như HTML) để tạo khuôn mẫu. Tài liệu hướng dẫn của Facebook rất chi tiết.
- Tốc độ ứng dụng vô cùng ấn tượng. Tất cả là nhờ kỹ thuật DOM ảo của React và tối ưu hóa rendering.
- Hỗ trợ render phía máy chủ rất tốt. Điều này làm cho React trở thành một frame-work mạnh mẽ cho các ứng dụng tập trung vào nội dung.
- Hỗ trợ tạo ứng dụng Progressive Web App (PWA) nhanh chóng. Đơn giản chỉ cần câu lệnh: "creat-react-app".
- Data-binding một chiều, có nghĩa là ít side effects không mong muốn.
- Mô hình Redux: một mô hình quản lý trạng thái ứng dụng cực tốt. Và một điều rất hay là nó rất dễ học.
- React theo trường phái lập trình hàm (Functional Programming), tạo mã dễ kiểm tra và có khả năng tái sử dụng cao.
- Ngoài ra, còn một ưu điểm cực lớn nữa, đó là nếu bạn biết về React thì bạn sẽ dễ dàng học thêm React Native.

Nhược điểm:

- Thiếu tài liệu hướng dẫn chính chủ: Chính vì sự phát triển siêu nhanh của ReactJS mà tài liệu hướng dẫn chính chủ không theo kịp. Hầu hết các bài viết hướng dẫn trên mạng bị lỗi thời.
- React đang đi theo hướng lập trình hàm, điều này sẽ gây đôi chút khó chịu và ác cảm với các dev đã quen với lập trình hướng đối tượng (OOP).
- Việc trộn giữa templating với logic ứng dụng (JSX) có thể gây nhầm lẫn cho một số nhà phát triển lúc đầu.

3.4.2 Angular

Angular là một JavaScript MVVM frame-work, được ra mắt từ 2009. Đây là một JS frame-work nổi đình đám trong giới front-end, giúp xây dựng các ứng dụng web có tính tương tác cao.



Hình 3.4.2: Logo Frame-work Angular

Ưu điểm:

- Hỗ trợ TypeScript
- Data-binding: Nó tự động đồng bộ hóa dữ liệu giữa thành phần model và view.
- Tài liệu hướng dẫn chi tiết và đầy đủ. Với tài liệu hướng dẫn chi tiết từ chính nhà cung cấp sẽ giúp người mới tìm hiểu Angular nhanh chóng tiếp cận và làm chủ frame-work này. Nhờ đó mà giảm thời gian training, thảo luận với đồng nghiệp... khi mà trên trang chủ của họ có tất cả.
- MVVM (Model-View-ViewModel): cho phép bạn chia dự án thành nhiều phần độc lập với nhau. Từ đó bạn sẽ dễ dàng chỉnh sửa, thêm tính năng cũng như bảo trì dự án sau này.
- Dependency Injection: DI là một design pattern mà được nhắc đến rất nhiều trong triết lý clean code SOLID. Với DI, nó cho phép bạn làm giảm sự phụ thuộc giữa các Object với nhau. Các object càng độc lập với nhau càng tốt, mục đích để sau này nâng cấp, sửa đổi... sẽ hạn chế ảnh hưởng tới các object khác.
- Cấu trúc và kiến trúc của Angular được tạo ra đặc biệt cho khả năng mở rộng dự án lớn

Nhược điểm:

- Angular khá là đa dạng các thành phần/khái niệm như Injectables, Components, Pipes, Modules... Điều này khiến cho việc tìm hiểu trở nên khó khăn hơn một chút so với React vốn chỉ có một thành phần duy nhất là Component.
- Chậm hơn React và Vuejs.

3.4.3 Vue

Vue.js là một frame-work JavaScript, được ra mắt vào năm 2013, hoàn toàn phù hợp để tạo giao diện người dùng có khả năng thích ứng cao và các ứng dụng trang đơn phức tạp.



Hình 3.4.3: Logo Frame-work Vue

Ưu điểm:

- Empowered HTML: có thể giúp tối ưu hóa việc xử lý các khối HTML khi sử dụng các thành phần khác nhau.
- Tài liệu hướng dẫn chi tiết.
- Khả năng tương thích cao vì khá tương đồng với Angular và React về mặt thiết kế và kiến trúc.
- Khả năng tích hợp tốt, nó có thể được sử dụng cho việc xây dựng cả ứng dụng (single-page applications) phức tạp hoặc chỉ một phần của ứng dụng. Điều này cho phép cập nhật, nâng cấp ứng dụng mà không ảnh hưởng quá nhiều tới hệ thống hiện tại.
- Large scaling: js có thể phát triển templates có khả năng tái sử dụng cao.
- Kích thước nhỏ.

Nhược điểm:

- Thiếu tài nguyên: Vuejs vẫn có thị phần khá nhỏ so với React hoặc Angular.
- Việc chia sẻ kiến thức trong cộng đồng vẫn còn khá ít.

3.4.4 Lựa chọn frame-work front-end

Các front-end frame-work cho phép lập trình viên bắt đầu phát triển một trang web mới dễ dàng và tốt hơn. Tuy nhiên việc chọn ra frame-work tốt nhất thực sự là một điều không hề dễ dàng.



Hình 3.4.4.a: Frame-work Angular vs Vue vs React

Ở các phần trước nhóm đã chọn Cross Platform App cho di động và ngôn ngữ front-end là JavaScript nên app di động sẽ được viết bằng React Native. Với sự tương thích giữa các frame-work có thể thấy React có thể là lựa chọn tốt để đồng hành cùng React Native.



Hình 3.4.4.b: Dùng React-native code web

Sau quá trình tìm hiểu nhóm tìm ra Expo, là frame-work React Native có thể render tốt trên web nên nhóm quyết định sẽ dùng React Native để dựng web.

3.5 Ngôn ngữ lập trình và frame-work back-end

3.5.1 Java và Spring-boot

Java là ngôn ngữ lập trình back-end phổ biến nhất.

Đây là ngôn ngữ lập trình đa năng, đa nền tảng, hướng đối tượng (object-oriented) và có tính đồng thời (concurrent) được phát triển bởi Oracle. Ngôn ngữ này có thể biên dịch (compile) mã lệnh dưới dạng mã byte Java (Java bytecode) để chạy trên Java Virtual Machine (JVM). Nhờ đó, bất kỳ thiết bị nào có thể cài đặt JVM thì có thể chạy được các ứng dụng viết bằng Java.



Hình 3.5.1.a: Logo Java

Spring là một frame-work phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code...

Spring Boot là một web frame-work dựa trên Java. Nó được tạo ra bởi Spring với mục đích để sử dụng và cung cấp một ứng dụng dựa trên Spring ở cấp độ sản xuất mà bạn có thể cần chạy. Có thể nói, Spring boot là bản rút gọn của Spring



Hình 3.5.1.b: Logo Spring-boot

Ưu điểm:

- Tạo các ứng dụng Spring độc lập
- Khả năng mở rộng cao
- Tài liệu tuyệt vời
- Được xây dựng cho các ứng dụng quy mô lớn sử dụng cách tiếp cận đám mây
- Hệ sinh thái mở rộng

3.5.2 C# và ASP .NET

C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.



Hình 3.5.2.a: Logo C#

ASP.Net là một nền tảng phát triển web được cung cấp bởi Microsoft và được sử dụng để tạo các ứng dụng dựa trên web. ASP.Net được phát hành lần đầu tiên vào năm 2002. Các ứng dụng ASP.Net cũng có thể được viết bằng nhiều ngôn ngữ .Net. Chúng bao gồm C#, VB.Net và J#.



Hình 3.5.2.b: Logo ASP .NET

Ưu điểm:

- Chạy mượt và ổn định cao hơn các ngôn ngữ khác.
- Tốc độ tải trang được tối ưu.
- Dễ dàng tùy chỉnh để tương thích với công nghệ mới.
- Độ bảo mật cao do kế thừa từ Java.

3.5.3 Python và Django

Python là một trong những ngôn ngữ backend phát triển nhanh và phổ biến hàng đầu thế giới. Nhờ có cú pháp đơn giản, dễ hiểu cùng nhiều tài liệu hướng dẫn trực tuyến nên Python rất phổ biến cho các lập trình viên mới.



Hình 3.5.3.a: Logo Python

Django là một frame-work Python cấp cao được xây dựng với ý tưởng kèm theo các "batteries included". Có nghĩa là hầu hết mọi thứ mà một lập trình viên muốn có đều có sẵn mặc định. Do đó, ít phải cài các plugin của bên thứ ba và yên tâm khi mọi thứ trong Django hoạt động nhịp nhàng cùng nhau. Vì vậy Django rất phù hợp với các ứng dụng lớn và không nên sử dụng với ứng dụng nhỏ.



Hình 3.5.3.b: Logo Django

Ưu điểm:

- Khả năng tùy biến cao
- Không cần làm từ đầu, tốc độ phát triển nhanh
- Có thể mở rộng
- Cộng đồng lớn và tài liệu nhiều

3.5.4 JavaScript và ExpressJS

Với NodeJS hỗ trợ để có thể lập trình ngôn ngữ Javascript để chạy trên server. Với sự phổ biến của JavaScript, dễ học và sự hỗ trợ mạnh mẽ từ các frame-work thì NodeJS đang là xu thế dẫn đầu hiện nay.



Hình 3.5.4.a: Logo JavaScript

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

Express là một frame-work nhanh, tối giản cho NodeJS. Nó cung cấp một layer mỏng của các tính năng ứng dụng web cơ bản, mà không làm mờ đi các tính năng của NodeJS. Hơn nữa, nó dễ dàng xây dựng một API mạnh mẽ với sự trợ giúp của các phương thức HTTP khác nhau và các phần mềm trung gian có sẵn.



Hình 3.5.4.b: Logo Express

Ưu điểm:

- API định tuyến tốt
- Tối giản
- Thiết lập rất dễ dàng và nó đơn giản để học
- Số lượng lớn các plugin có sẵn để sử dụng

3.5.5 Lựa chọn ngôn ngữ lập trình và frame-work back-end

Chúng ta có thể thấy Python là một ngôn ngữ rất dễ học, dễ sử dụng, có cú pháp đơn giản, cộng đồng hỗ trợ lớn, nhiều công cụ và công nghệ hỗ trợ. Đây cũng là ngôn ngữ mà được rất nhiều người yêu thích, tuy nhiên Python lại có những hạn chế như tốc độ khá chậm, chạy đơn luồng, vì vậy sẽ không thể đáp ứng được khi ứng dụng càng ngày càng có nhiều kết nối.

NodeJS có đặc điểm là tốc độ rất nhanh, xử lý nhiều kết nối tốt, dễ dàng mở rộng để phát triển. Tuy nhiên NodeJS gây hao tốn tài nguyên và thời gian. NodeJS còn non trẻ, giữa các version có sự thay đổi rất lớn.

Không có Visual Studio thì khó có thể viết code. ASP.NET có phí sử dụng khá cao, không phù hợp sử dụng cho các doanh nghiệp nhỏ hay các cá nhân.

Java là một ngôn ngữ mạnh, linh hoạt, bảo mật cao, viết một lần thực thi khắp nơi, lập trình đa luồng tốt có thể đáp ứng những hệ thống cần tốc độ cao. Hệ thống viết bằng Java có kiến trúc thiết kế tốt thì dễ mở rộng, có tính hướng đối tượng đầy đủ phù hợp với các nghiệp vụ phức tạp. Lịch sử dày dặn giúp Java có cộng đồng hỗ trợ mạnh mẽ về các tài nguyên có sẵn và tài liệu đầy đủ. Lập trình viên Java dễ tìm, dễ dàng kế thừa hệ thống.

Vậy nhóm quyết chọn Java cùng với Spring-boot frame-work để sử dụng trong dự án này.

3.6 Cơ sở dữ liệu

Khái niệm

Cơ sở dữ liệu (Database) là một tập hợp các dữ liệu có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức

Phân loại

- Cơ sở dữ liệu quan hệ (SQL)
- Cơ sở dữ liệu phi quan hệ (NoSQL):
 - Cơ sở dữ liệu hướng tài liệu
 - Cơ sở dữ liệu dạng khoá – giá trị
 - Cơ sở dữ liệu mô hình wide – column
 - Cơ sở dữ liệu dạng bộ máy tìm kiếm



Hình 3.6: Phân loại cơ sở dữ liệu

3.6.1 Cơ sở dữ liệu quan hệ

Hệ thống quản lý cơ sở dữ liệu quan hệ được phát triển vào năm 1970 với nền tảng lý thuyết vững chắc và có ảnh hưởng lớn các hệ thống khác hôm nay. Tên tiếng Anh: Relational database management system – RDBMS.

Các RDBMS lưu trữ dữ liệu với dạng “quan hệ”: các bảng với dòng và cột nơi mọi thông tin dữ liệu được lưu trữ như một giá trị của một ô cụ thể. Dữ liệu trong một RDBMS được quản lý thông qua giao thức nổi tiếng SQL – ngôn ngữ truy vấn cấu trúc. SQL là giao thức được chuẩn hoá, đem đến sự đa nhiệm và tính dễ đoán biết.

Ưu điểm:

- Các hệ thống RDBMS hữu dụng trong việc xử lý các dữ liệu được cấu trúc kỹ càng và hỗ trợ ACID – 4 thuộc tính quan trọng của bất kỳ hệ thống cơ sở dữ liệu nào:
 - Tính nguyên tử (Atomicity). Một giao dịch có nhiều thao tác khác biệt thì hoặc là toàn bộ các thao tác hoặc là không một thao tác nào được hoàn thành.
 - Tính nhất quán (Consistency). Một giao dịch hoặc là sẽ tạo ra một trạng thái mới và hợp lệ cho dữ liệu, hoặc trong trường hợp có lỗi sẽ chuyển toàn bộ dữ liệu về trạng thái trước khi thực thi giao dịch.
 - Tính độc lập (Isolation). Một giao dịch đang thực thi và chưa được xác nhận phải bảo đảm tách biệt khỏi các giao dịch khác.
 - Tính bền vững (Durability). Dữ liệu được xác nhận sẽ được hệ thống lưu lại sao cho ngay cả trong trường hợp hỏng hóc hoặc có lỗi hệ thống, dữ liệu vẫn đảm bảo trong trạng thái chuẩn xác.
- Dữ liệu được lưu trữ và truy xuất dễ dàng bằng các lệnh truy vấn SQL. Cấu trúc dữ liệu cũng có thể được mở rộng nhanh chóng, việc bổ sung thêm các dữ liệu mới cũng không ảnh hưởng tới các data có sẵn.
- Các RDBMS còn có khả năng cấp quyền truy xuất và chỉnh sửa thông tin cho các loại người dùng khác nhau (admin, user, khách vãng lai, etc.).

Nhược điểm:

- Không xử lý tốt các dữ liệu phi cấu trúc.
- Các dữ liệu khi bị chia cắt cần được viết lại dưới dạng khác để đọc hơn là ở dạng bảng tính (table), và tốc độ xử lý dữ liệu cũng khá chậm.
- Việc thay đổi cơ sở dữ liệu dạng RDBMS cũng khá khó vì tính quy củ chặt chẽ của nó.
- RDBMS tốn nhiều chi phí hơn các hệ thống cơ sở dữ liệu khác trong việc xây dựng và phát triển.
- Việc nâng cấp và mở rộng máy chủ theo chiều dọc hoặc ngang (*horizontal/ vertical scalability) phức tạp.
- Việc phân mảnh (*sharding) trong khi vẫn giữ vững các tiêu chí về ACID là một thử thách lớn.

Nên dùng khi:

- Các trường hợp khi giữ vững tính toàn vẹn dữ liệu – dữ liệu không thể bị chỉnh sửa dễ dàng là tối cần thiết.
- Thông tin nội bộ

3.6.2 Cơ sở dữ liệu hướng tài liệu

Cơ sở dữ liệu hướng tài liệu hay còn gọi là Document store, một thiết kế riêng biệt cho việc lưu trữ tài liệu dạng văn kiện JSON, BSON hoặc XML. Vì là cấu trúc dữ liệu không ràng buộc khác với SQL, các cơ sở dữ liệu này không đòi hỏi người dùng tự tạo bảng nhập liệu trước khi nhập dữ liệu vào. Các tài liệu có thể chứa bất kỳ dữ liệu nào. Cơ sở dữ liệu dạng này có các cặp khoá – giá trị nhưng cũng có đính kèm các trị số siêu dữ liệu (*metadata) giúp việc truy vấn (*query) dễ dàng hơn.

Các cơ sở dữ liệu tiêu biểu:



Hình 3.6.2.a: Logo Mongo



Hình 3.6.2.b: Logo Couchbase

Ưu điểm:

- Cơ sở dữ liệu hướng tài liệu rất linh hoạt, có thể xử lý dữ liệu nửa cấu trúc và không cấu trúc rất tốt.
- Không cần quan tâm tới dạng dữ liệu khi setup, điều này tốt trong trường hợp không lường trước được dạng dữ liệu nào bạn sẽ cần lưu trữ.
- Người dùng có thể thiết kế một cấu trúc cho một tài liệu cụ thể mà không ảnh hưởng tới các tài liệu khác.
- Schema cho cơ sở dữ liệu cũng có thể được tùy chỉnh mà không gây ra thời gian downtime, giúp đem đến high availability (tính sẵn sàng cao).
- Thời gian ghi dữ liệu cũng rất nhanh.
- Dễ dàng mở rộng theo chiều ngang của chúng. Quá trình chia sẻ cũng dễ hiểu và dễ thao tác hơn so với cơ sở dữ liệu quan hệ, nên document store có thể mở rộng nhanh và dễ dàng.

Nhược điểm:

- Cơ sở dữ liệu dạng lưu trữ tài liệu hy sinh các yếu tố ACID để đổi lấy sự linh hoạt.
- Việc truy vấn chỉ có thể được thực hiện trong từng tài liệu, không thể truy vấn dữ liệu trên nhiều tài liệu khác nhau.

Nên dùng khi:

- Dữ liệu phi cấu trúc hoặc không có cấu trúc
- Quản lý nội dung
- Phân tích dữ liệu chuyên sâu
- Tạo mẫu nhanh

3.6.3 Cơ sở dữ liệu dạng khoá – giá trị

Key-value stores là kiểu lưu trữ đơn giản nhất trong các loại cơ sở dữ liệu NoSQL đồng thời nó cũng là kiểu lưu trữ cho tất cả các Hệ quản trị cơ sở dữ liệu NoSQL. Thông thường, các hệ quản trị cơ sở dữ liệu Key-value lưu trữ dữ liệu dưới dạng key (là một chuỗi duy nhất) liên kết với value có thể ở dạng chuỗi văn bản đơn giản hoặc các tập, danh sách dữ liệu phức tạp hơn. Các giá trị này có thể được gán bất cứ loại giá trị nào: Số, Chuỗi giá trị, Bộ đếm, JSON, XML, HTML, PHP, Nhị phân, Hình ảnh, Video ngắn, Danh sách,... Quá trình tìm kiếm dữ liệu thường sẽ được thực hiện thông qua key, điều này dẫn đến sự hạn chế về độ chính xác.

Các cơ sở dữ liệu tiêu biểu:



Hình 3.6.3: Logo Redis vs Logo Memcached

Ưu điểm:

- Rất linh hoạt và có thể xử lý nhiều loại dữ liệu một cách nhanh chóng.
- Các chìa khoá được dùng để truy xuất thẳng tới các giá trị tìm kiếm, mà không cần thông qua quá trình index, giúp quá trình tìm kiếm diễn ra nhanh chóng.
- Linh động: lưu trữ key – value có thể được chuyển từ hệ thống này sang hệ thống khác mà không cần code lại.
- Có thể mở rộng theo chiều ngang dễ dàng và chi phí vận hành thấp.

Nhược điểm:

- Tính linh hoạt của CSDL dạng key – value bị đánh đổi bởi tính chính xác.
- Rất khó để truy xuất giá trị chính xác vì dữ liệu được lưu trữ theo blob, nên kết quả trả về hầu như đều theo blob.
- Khó khăn khi báo cáo số liệu hoặc cần chỉnh sửa một phần của các giá trị.
- Không phải objects nào cũng có thể được cấu hình thành cặp chìa khoá – giá trị được.

Nên dùng khi:

- Khuyến nghị các sản phẩm / thông tin tương tự
- Thông tin và thiết lập người dùng
- Dữ liệu phi cấu trúc như review sản phẩm, bình luận của blog
- Quản lý session trên diện rộng
- Dữ liệu được truy xuất thường xuyên nhưng không thường xuyên được cập nhật

3.6.4 Cơ sở dữ liệu mô hình wide – colum

Mô hình wide – column là một dạng lưu cơ sở dữ liệu phi quan hệ lưu trữ theo dạng cột. Mô hình này có vài điểm tương đồng với mô hình key – value nhưng cũng có vài tính chất của dạng cơ sở dữ liệu quan hệ.

Mô hình wide – column dựa trên khái niệm keyspace thay vì schema. Một keyspace bao gồm nhiều cụm column (tương tự như table nhưng linh hoạt hơn về cấu trúc), mỗi cụm bao gồm nhiều hàng và nhiều cột riêng biệt. Mỗi hàng không cần phải có số lượng hoặc loại cột. Một timestamp quyết định phiên bản gần nhất của data.

Các cơ sở dữ liệu tiêu biểu:



Hình 3.6.4: Logo Cassandra vs Logo Hbase

Ưu điểm:

- Có thể xử lý dữ liệu cấu trúc và phi cấu trúc, đồng thời cũng dễ dàng nâng cấp.
- Khả năng mở rộng theo chiều ngang cũng dễ dàng và nhanh chóng hơn cơ sở dữ liệu quan hệ.
- Có khả năng nén tốt hơn cơ sở dữ liệu dạng dòng.
- Data set lớn có thể dễ dàng duyệt hơn.
- Xử lý tốt các yêu cầu truy xuất tập trung.

Nhược điểm:

- Upload và update số liệu cá nhân rất khó.
- Chậm hơn so với cơ sở dữ liệu quan hệ khi xử lý các giao dịch.

Nên dùng khi:

- Xử lý các dữ liệu lớn (big data) khi tốc độ cần thiết.
- Lưu trữ dữ liệu big data.
- Các project có quy mô lớn.

3.6.5 Cơ sở dữ liệu dạng bộ máy tìm kiếm

Cơ sở dữ liệu dạng bộ máy tìm kiếm (Elasticsearch) về cốt lõi là một bộ máy tìm kiếm và không hoàn toàn là cơ sở dữ liệu chuyên biệt như các loại trên, nhưng ngày càng được các nhà phát triển tận dụng để giảm thiểu độ lag khi tìm kiếm thông tin. Elasticsearch được xem như một cơ sở dữ liệu dạng phi quan hệ, dựa trên nền tảng lưu trữ dữ liệu dạng văn kiện, thiết kế chuyên biệt để tối ưu hoá lưu trữ và trao đổi dữ liệu nhanh chóng.

Các cơ sở dữ liệu tiêu biểu:



Hình 3.6.5: Logo Elasticsearch

Ưu điểm:

- Có khả năng mở rộng cao, với schema linh hoạt và tốc độ trả về thông số lưu trữ nhanh.
- Hỗ trợ khả năng tìm kiếm nâng cao: tìm kiếm full text, khuyến nghị các kết quả tìm kiếm, và hỗ trợ các thông tin tìm kiếm phức tạp.
- Có tính năng "Stemming" là từ để chỉ kỹ thuật dùng để biến đổi 1 từ về dạng gốc bằng cách đơn giản là loại bỏ 1 số ký tự nằm ở cuối được xem như biến thể của từ.

Nhược điểm:

- Elasticsearch được sử dụng với hình thức thay thế hoặc bổ trợ cho CSDL có sẵn hơn là độc lập.
- Độ ổn định và bảo mật kém, không có giao thức xác định và xác nhận danh tính của một khách truy cập hoặc điều khiển đăng nhập.
- Không hỗ trợ thanh toán.

Nên dùng khi:

- Cung cấp các thông tin phụ hỗ trợ việc ra quyết định khi tìm kiếm dữ liệu
- Lưu file tạm

3.6.6 Lựa chọn cơ sở dữ liệu

3.6.6.a Lựa chọn loại cơ sở dữ liệu

Dựa vào các đối tượng mà hệ thống hướng tới, hệ thống cần:

- Đảm bảo tính toàn vẹn, tính nhất quán, giảm dư thừa dữ liệu
- Cần đảm bảo mối liên hệ giữa điểm số, người chấm điểm và sinh viên được chặt chẽ
- Đảm bảo tính minh bạch của các điểm số
- Xuất báo cáo
- Các tính năng và kiểu dữ liệu có thể dự đoán trước
- Ít có sự thay đổi lớn về nghiệp vụ khi hệ thống dần đi vào ổn định
- Không cần dữ liệu phân tán

Với những phân tích đánh giá như trên, nhóm quyết định lựa chọn sử dụng hệ cơ sở dữ liệu quan hệ.

Chấp nhận nhược điểm tồn tại về cơ sở dữ liệu quan hệ:

- Khó khăn trong việc thiết kế
- Khó khăn khi gặp sự thay đổi về nghiệp vụ nhất định

3.6.6.b Lựa chọn loại cơ sở dữ liệu

Nhóm sẽ chọn PostgreSQL để sử dụng trong dự án.



Hình 3.6.6: Logo PostgreSQL

Ưu điểm của PostgreSQL so với các cơ sở dữ liệu quan hệ khác:

- PostgreSQL được dự đoán sẽ chiếm vị trí SQL Database phổ biến nhất của MySQL trong vòng 3 năm
- Hỗ trợ Analytic functions
- Ngoài các Data types chuẩn sql thì còn có array, hstore, object (người dùng định nghĩa)
- Full text search: không giới hạn
- Thuật toán JOIN sử lý tối ưu hơn
- DELETE nhanh hơn
- Hỗ trợ JSON

4 Lập trình UI web, android, ios với Expo

Ở chương 3, nhóm đã gặp khó khăn trong việc lựa chọn JavaScript UI framework để giảm bớt công việc front-end giữa các nền tảng. Giải quyết điều đó, nhóm đã có ý tưởng dùng React-native để dựng web và nhóm đã tìm ra Expo.

Ở chương này, thông qua các mục nhóm sẽ trả lời các câu hỏi: Từ đâu mà nhóm tìm đến Expo? Expo là gì? Expo làm được gì? Tại sao nên chọn Expo?



Hình 4: Lập trình UI web, android, ios với Expo

4.1 Hoàn cảnh và ý tưởng

Ở những giai đoạn đầu của dự án, nhóm đối diện phải khó khăn lớn về front-end vì những lý do sau:

- Đội ngũ phát triển 1 thành viên và không hề có kinh nghiệm front-end.
- Quá nhiều frame-work và library UI để lựa chọn.
- Hiện thực trên nhiều nền tảng khiến khối lượng công việc front-end trở nên rất nhiều so với back-end.

4.1.1 Từ web đến di động

Ý tưởng ban đầu giải quyết vấn đề trên là tạo dựng một sản phẩm web trong giai đoạn đề cương. Sau đó mở rộng bằng cách nhúng web đó xuống di động bằng webview. Tuy nhiên, giải pháp này vướng phải nhiều vấn đề như sau:

- Trải nghiệm tệ ở ứng dụng di động.
- Hiệu năng của ứng dụng cũng giảm rất nhiều.
- Ngoài ra nó còn phụ thuộc rất lớn vào trang web hỗ trợ responsive.
- Các components UI phức tạp rất khó tái sử dụng.
- Các components UI nhỏ khi đưa xuống di động thì bị "phình ra" khó kiểm soát.

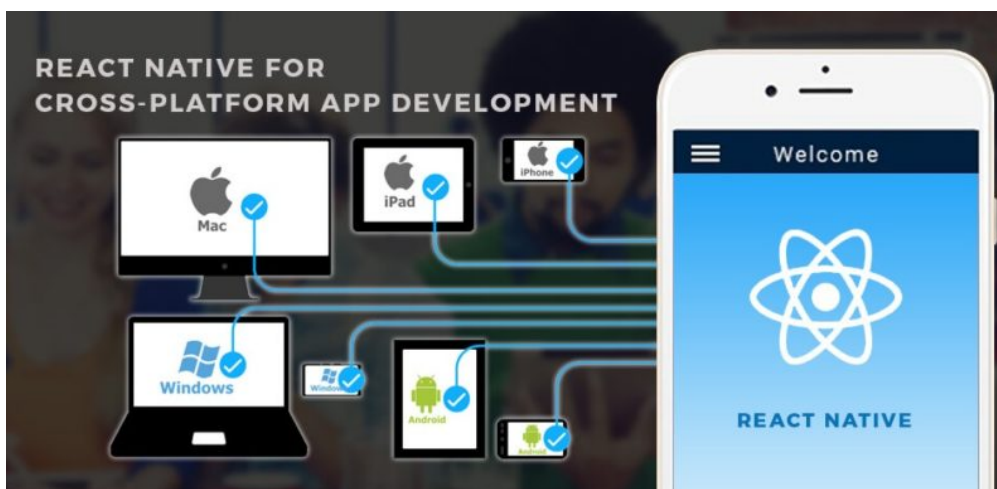
Đây chưa thực sự là một giải pháp tốt.

4.1.2 Từ di động đến web

Phương án này là dùng Flutter hoặc React Native code di động, sau đó chuyển code lên web. Flutter thì có Flutter Web, tương tự React Native có React Native Web dùng để code web.

Những điểm lợi thế so với dùng webview

- Đáp ứng trải nghiệm ứng dụng di động tốt.
- Hiệu năng của ứng dụng được đảm bảo.
- Giảm phụ thuộc vào trang web xuống rất nhiều.
- Có thể tái sử dụng các components UI nhỏ và lớn khi chuyển từ di động đến web.



Hình 4.1.2: React Native Web

Qua trải nghiệm React Native Web chưa thật sự thỏa mãn sự mong đợi từ nhóm.

Nhóm đã tìm được Expo, một frame-work JavaScript khác có nhiều điểm nổi bật hơn bộ đôi React Native + React Native Web.

Sau đây là một số so sánh về các frame-work

Bảng 4.1.2: So sánh React Native + React Native Web với Expo

	React Native + React Native Web	Expo
Android Studio and XCode để chạy code tốt	Cần	Chỉ cần terminal
Máy Mac	Cần	Không cần
Kết nối thiết bị test	Cáp USB	Cáp USB hoặc chia sẻ qua mạng LAN
Chia sẻ app	Build file .apk .ipa	Thêm phương thức share QR code hoặc link
Cung cấp JS APIs (Push-Notifications, Asset Manager)	Cài thủ công với npm	Dùng gói của Expo cung cấp
Thời gian cấu hình	Phức tạp	Chỉ cần install và chạy bằng command
Build app để thử nghiệm	Cần	Không cần
Phân phối app đến store	Thủ công	Dùng command của Expo cung cấp
Viết code Native như Java/Objective-C/Swift	Có thể	Không thể
Kích thước ứng dụng ban đầu	Nhỏ và gán từng phần vào nhờ npm	Lớn nhưng cung cấp đầy đủ
Khả năng tương tác với các thư viện khác	Tốt	Hạn chế vì ràng buộc với Expo
Debug	Bình thường	Khó khăn hơn vì gom nhiều nền tảng chung
Build file local	Có thể	Phải đẩy lên cloud để Expo build

Chấp nhận kích thước ứng dụng ban đầu lớn và khó debug giữa các nền tảng.

Với ứng dụng chấm điểm không cần quá thiệp quá sâu vào từng nền tảng cũng như UI phức tạp để vay mượn từ các library khác. Nhóm tin rằng hệ sinh thái của Expo là lựa chọn hợp lý cho dự án này.

Với những phân tích trên nhóm sẽ dùng Expo để dựng toàn bộ UI cho dự án.

4.2 Expo

4.2.1 Giới thiệu Expo

Expo là một framework được xây dựng xung quanh React Native để giúp bạn xây dựng các dự án iOS và Android bằng JavaScript và React và Web. Nó giống như Laravel hay Symfony cho các nhà phát triển PHP, hoặc Ruby on Rails cho các nhà phát triển Ruby. Expo cung cấp một lớp nằm trên cùng của React Native API để giúp chúng dễ sử dụng và quản lý.



Hình 4.2.1: Logo hệ sinh thái Expo

Expo cung cấp các công cụ giúp lập trình viên dễ dàng khởi tạo và kiểm thử các ứng dụng React Native. Sau cùng, nó cung cấp các thành phần UI và các dịch vụ thường chỉ có sẵn khi cài đặt một thành phần React Native của bên thứ ba. Tất cả đều được cung cấp thông qua Expo SDK.

Expo không phụ thuộc vào IDE, lập trình viên có thể code với bất cứ phần mềm chỉnh sửa văn bản nào.

Expo không cần config, các bước chạy rất đơn giản

4.2.2 Chạy thử ứng dụng từ Expo

Cài đặt Expo

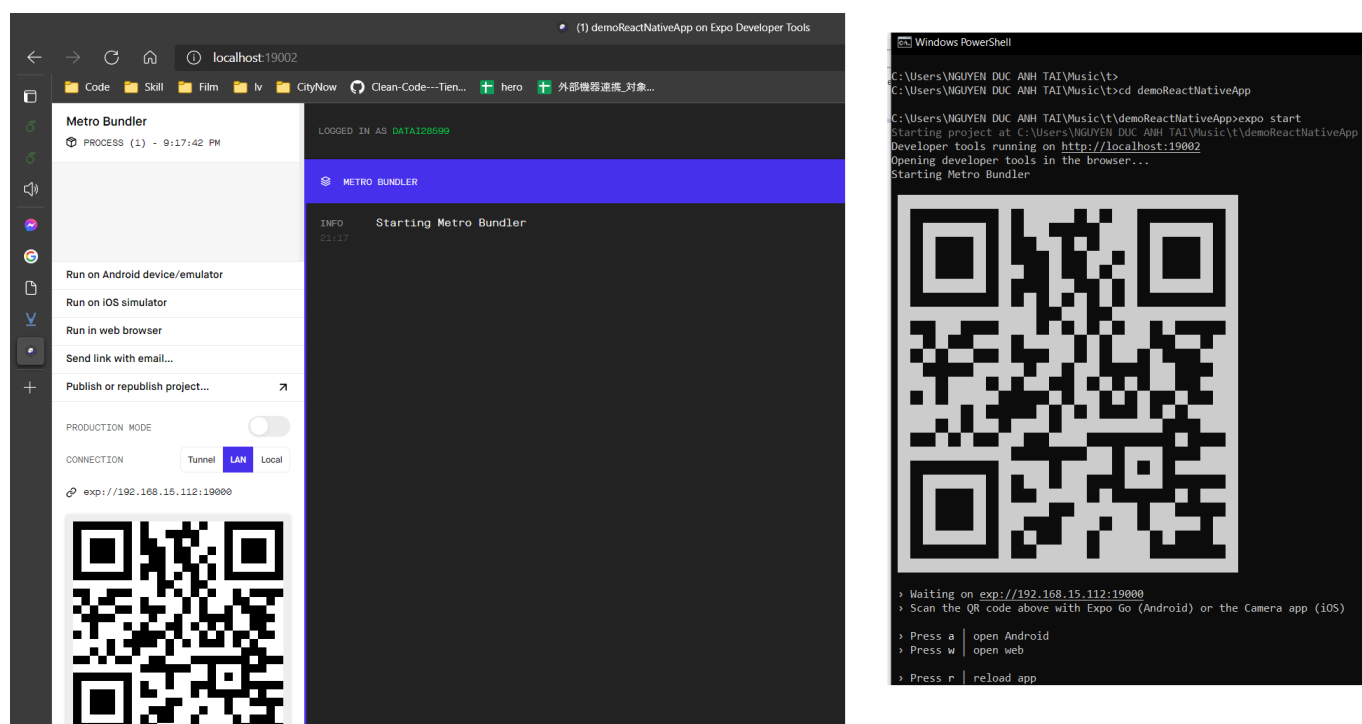
```
npm install --global expo-cli
```

Khởi tạo project expo

```
expo init demoReactNativeApp  
cd demoReactNativeApp
```

Chạy project

```
expo start
```



Hình 4.2.2: Màn hình web(trái) và Màn hình terminal (phải) khi start Expo

Lập trình viên hoặc người dùng thử cần cài đặt ứng dụng Expo Go trên CH Play hoặc App Store để quét các QR Code đến từ Expo. Sau khi quét, Expo sẽ tải mã JavaScript xuống di động vào start nó thành app di động. Lập trình viên có thể debug trên điện thoại thông qua mạng LAN mà không cần dây USB tương ứng. Sau khi thoát, Expo Go không hề để lại bản dựng nào trên điện thoại.

Chúng ta thấy sẽ có các option cho chúng ta chạy build ở các nền tảng khác nhau và expo còn support chia sẻ ứng dụng qua email, SMS hay puslish lên trang web <https://expo.io/> để build .apk hay đẩy ra các store.

Lập trình viên cũng có thể xem log của từng thiết bị tại màn hình này.

5 Thiết kế kiến trúc hệ thống

Thiết kế kiến trúc hệ thống có vai trò rất quan trọng trong quy trình xây dựng phần mềm vì có được cái nhìn đầy đủ, đúng đắn, chính xác về hệ thống sẽ xây dựng trong tương lai nhằm thuận lợi cho việc sửa chữa, bổ sung và phát triển hệ thống khi có sự thay đổi. Hơn thế nữa, thiết kế còn tránh được những sai lầm trong cài đặt hay lạc lối trong quá trình thi công phần mềm.



Hình 5: Thiết kế kiến trúc hệ thống

5.1 Thiết kế cơ sở dữ liệu

5.1.1 Nền tảng thiết kế cơ sở dữ liệu

5.1.1.a Chia nhóm các bảng

Dựa vào yêu cầu của đề tài thì nhóm sẽ phân loại các table của database ra làm 4 nhóm

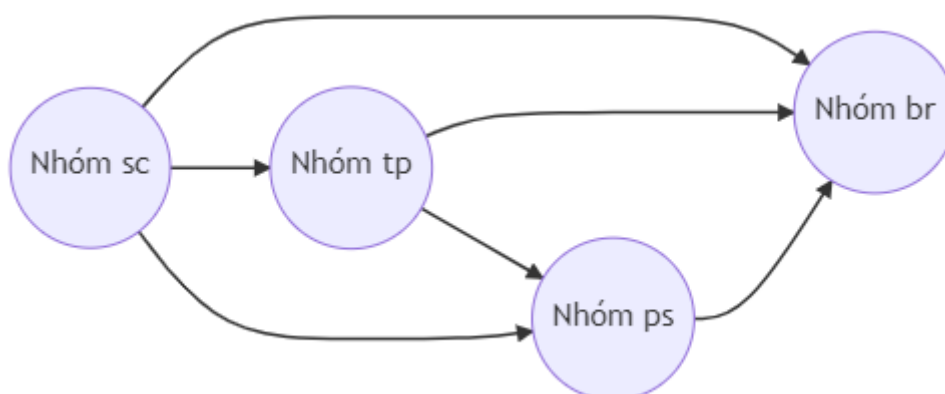
- Nhóm các table liên quan đến nhân sự (person), kí hiệu là **ps**
- Nhóm các table liên quan đến chủ đề (topic), kí hiệu là **tp**
- Nhóm các table liên quan đến điểm (score), kí hiệu là **sr**
- Nhóm các table còn lại (break), kí hiệu là **br**

Tên các table sẽ được bắt đầu bằng kí hiệu thể hiện cho nhóm của mình.

Ví dụ: Table ps_teacher thuộc nhóm liên quan đến nhân sự và chứa dữ liệu của giảng viên.

5.1.1.b Mối quan hệ giữa các nhóm bảng

Mối quan hệ giữa các khóa ngoại (Foreign key) của các table trong nhóm được thể hiện bằng lược đồ sau.



Sơ đồ 5.1.1.b: Mối quan hệ Foreign key giữa các nhóm table

Với lược đồ này, nhóm sẽ quy định rằng các bảng thuộc nhóm có mũi tên đi ra có thể có Foreign key reference đến các bảng thuộc nhóm có mũi tên đi vào tương ứng.

Vậy sẽ có 3 luật thiết kế Foreign key giữa các nhóm sẽ được tuân thủ:

- Tất cả các bảng ở mọi nhóm có thể có Foreign key đến các bảng thuộc nhóm **br**
- Các bảng thuộc nhóm **tp** có thể có Foreign key đến các bảng thuộc nhóm **ps**
- Các bảng thuộc nhóm **sc** có thể có Foreign key đến các bảng thuộc nhóm **tp** và **ps**

5.1.1.c Bảng cơ sở

Mọi table đều extends **BaseTable** chứa các thông tin sau

Dữ liệu: Các dữ liệu cơ bản mọi record đều có

Bảng 5.1.1.c: Đặc tả chi tiết bảng BaseTable

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
id	bigint	không	auto increment	Định danh mỗi dòng trong bảng
deleted	boolean	không	false	Xóa logic dòng
created_at	timestamp without time zone	không	thời gian tạo record	Thời gian tạo record
updated_at	timestamp without time zone	không	thời gian cập nhật record	Thời gian cập nhật record
created_by	bigint	không	không	Mã số người tạo record
updated_by	bigint	không	không	Mã số người cập nhật record

Với cách thiết kế này, nhóm sẽ sử dụng duy nhất id làm Primary key cho toàn bộ bảng và không sử dụng khóa chính tổng hợp (composite-key). Nếu cần thêm field để xác định danh cho dòng, nhóm sẽ dùng ràng buộc (constraint) unit.

5.1.1.d Đa ngôn ngữ bằng json

PostgreSQL có hỗ trợ kiểu dữ liệu JSON, nhóm sẽ tận dụng để lưu trữ đa ngôn ngữ.

Ví dụ

Bảng 5.1.1.d: Ví dụ về đa ngôn ngữ bằng json

id	multi_lang_field
1	{"en":"Computer Science","vi":"Khoa học máy tính"}

5.1.1.e Giảm dư thừa dữ liệu bằng array

Các thiết kế từng bảng sẽ tuân thủ thêm nguyên tắc giảm số dòng (record) khi có thể bằng các tận dụng array của PostgreSQL

Ví dụ một bảng sau

Bảng 5.1.1.e.1: Giảm dư thừa dữ liệu bằng array

id	Loại sản phẩm	Tên sản phẩm
1	A	Sting
2	A	Pepsi
3	A	Coca

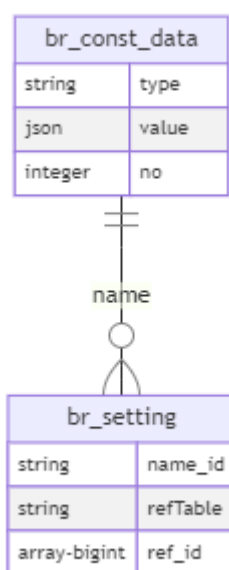
Sẽ được chuyển thành

Bảng 5.1.1.e.2: Kết quả giảm dư thừa dữ liệu bằng array

id	Loại sản phẩm	Tên sản phẩm
1	A	[String, Pepsi, Coca]

Cách thiết kế này giúp giảm số lượng dữ liệu dư thừa, tránh group by trong câu query.

5.1.2 Các bảng nhóm khác (br)



Sơ đồ 5.1.2: ERD diagram của nhóm br

5.1.2.a Bảng br_const_data

Dữ liệu: Các dữ liệu ít hoặc gần như không thay đổi

Ràng buộc: type & value là unit

Bảng 5.1.2.a.1: Đặc tả chi tiết bảng br_const_data

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
type	string	không	không	Tên của loại dữ liệu ít thay đổi
value	json	không	không	Giá trị đa ngôn ngữ của dữ liệu ít thay đổi
no	integer	có	không	Thứ tự hiển thị trên UI

Bảng 5.1.2.a.2: Ví dụ về bảng br_const_data

id	type	value	no
1	degree	{"en": "Bachelor", "vi": "Cử nhân"}	1
2	degree	{"en": "Master", "vi": "Thạc sĩ"}	2
3	degree	{"en": "Doctor", "vi": "Tiến sĩ"}	3
4	degree	{"en": "Professor", "vi": "Giáo sư"}	4

5.1.2.b Bảng br_setting

Dữ liệu: Các dữ liệu cài đặt chung của hệ thống

Ràng buộc: name_id là unit

Bảng 5.1.2.b.1: Đặc tả chi tiết bảng br_setting

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
name_id	string	không	không	Tên của setting
ref_table	string	không	không	Table chứa giá trị setting
ref_id	array-bigint	không	không	Các record được setting

Ví dụ: Chỉ một số cấp bậc học vấn được hiển thị trên màn hình

Bảng 5.1.2.b.2: Chuẩn bị bảng br_const_data

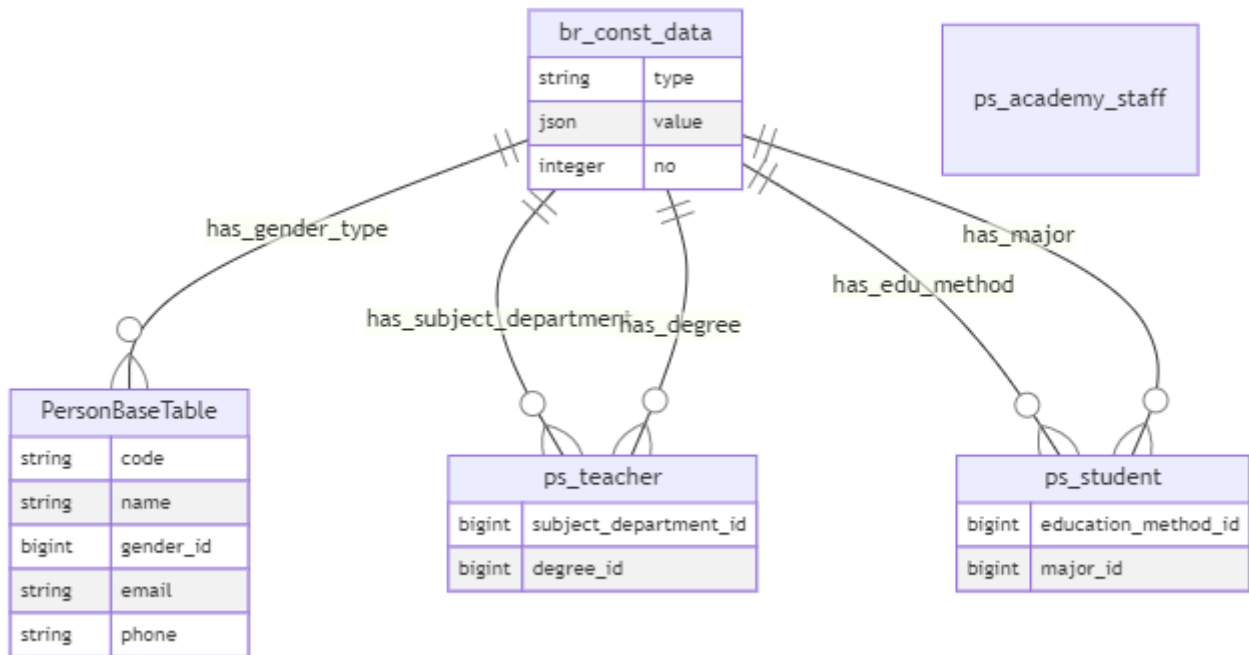
id	type	value	no
1	degree	{"en": "Bachelor", "vi": "Cử nhân"}	1
2	degree	{"en": "Master", "vi": "Thạc sĩ"}	2
3	degree	{"en": "Doctor", "vi": "Tiến sĩ"}	3
4	degree	{"en": "Professor", "vi": "Giáo sư"}	4
5	setting	{"en": "Degree visible", "vi": "Trình độ học vấn hiển thị"}	null

Bảng 5.1.2.b.3: Ví dụ về bảng br_setting

id	name_id	ref_table	ref_id
1	5	br_const_data	[2,3]

5.1.3 Các bảng nhóm nhân sự (ps)

Mọi table thuộc nhóm **ps** đều extends **PersonBaseTable**, **PersonBaseTable** extend **BaseTable**



Sơ đồ 5.1.3: ERD diagram của nhóm ps

5.1.3.a Bảng PersonBaseTable

Là bảng cơ sở của nhóm **ps**. Nó là bảng ảo không có thực trong database. Các bảng thuộc nhóm **ps** đều có tất cả các field thuộc bảng này.

Dữ liệu: Thông tin cơ bản của nhân sự

Ràng buộc: code là unit

Bảng 5.1.3.a: Đặc tả chi tiết bảng PersonBaseTable

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
code	string	không	không	Mã số nhân viên, mã số sinh viên
name	string	có	không	Tên đầy đủ
gender_id	bigint	có	không	Giới tính
email	string	không	không	HCMUT mail
phone	string	có	không	Số điện thoại

5.1.3.b Bảng ps_academy_staff

Dữ liệu: Thông tin giáo vụ

Các field của bảng này bằng các field thuộc bảng BaseTable + field thuộc bảng PersonBaseTable

5.1.3.b Bảng ps_teacher

Dữ liệu: Thông tin giảng viên

Bảng 5.1.3.c: Đặc tả chi tiết bảng ps_teacher

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
subject_department_id	bigint	có	không	Phòng ban giảng viên làm việc
degree_id	bigint	có	không	Học vị của giảng viên

5.1.3.c Bảng ps_student

Dữ liệu: Thông tin sinh viên

Bảng 5.1.3.c: Đặc tả chi tiết bảng ps_student

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
education_method_id	bigint	có	không	Phương thức đào tạo sinh viên
major	bigint	có	không	Chuyên ngành của sinh viên

5.1.3.d Ví dụ về bảng PersonBaseTable, ps_academy_staff, ps_teacher, ps_student

Ví dụ về bảng PersonBaseTable:

Bảng 5.1.3.d.1: Chuẩn bị bảng br_const_data

id	type	value	no
1	gender	{"en": "Male", "vi": "Name"}	null
2	gender	{"en": "Female", "vi": "Nữ"}	null

Bảng 5.1.3.d.2: Ví dụ về bảng PersonBaseTable

id	code	name	gender_id	email	phone
1	1713015	Nguyễn Đức Anh Tài	1	tai@hcmut.edu.vn	0905345670

Ví dụ về bảng `ps_academy_staff`, `ps_teacher`, `ps_student`:

Bảng 5.1.3.d.3: Chuẩn bị bảng `br_const_data`

id	type	value	no
1	gender	{"en": "Male", "vi": "Name"}	null
2	gender	{"en": "Male", "vi": "Name"}	null
3	degree	{"en": "Bachelor", "vi": "Cử nhân"}	1
4	degree	{"en": "Master", "vi": "Thạc sĩ"}	2
5	degree	{"en": "Doctor", "vi": "Tiến sĩ"}	3
7	subjectDepartment	{"en": "Information System", "vi": "Hệ thống thông tin"}	null
8	subjectDepartment	{"en": "Software Technology", "vi": "Công nghệ phần mềm"}	null
9	subjectDepartment	{"en": "Systems and Networks", "vi": "Hệ thống và mạng"}	null
10	subjectDepartment	{"en": "Computer Science", "vi": "Khoa học máy tính"}	null
11	subjectDepartment	{"en": "Computer Engineering", "vi": "Kỹ thuật máy tính"}	null
12	educationMethod	{"en": "Formal", "vi": "Chính quy"}	null
13	major	{"en": "Computer Science", "vi": "Khoa học máy tính"}	null

Bảng 5.1.3.d.4: Ví dụ về bảng `ps_academy_staff`

id	code	name	gender_id	email	phone
1	1713015	Nguyễn Đức Anh Tài	1	tai@hcmut.edu.vn	0905345670

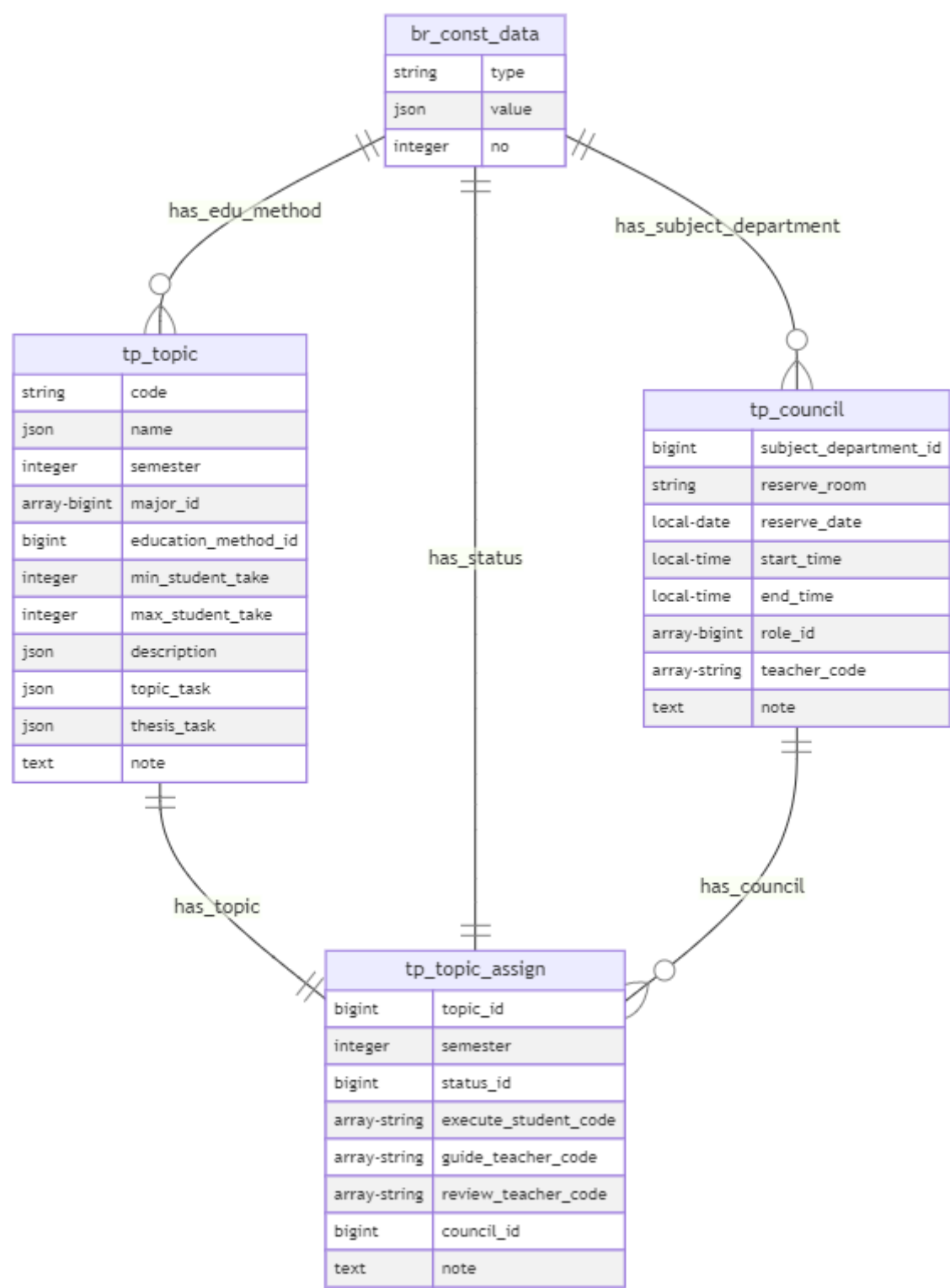
Bảng 5.1.3.d.5: Ví dụ về bảng `ps_teacher`

id	code	name	gender_id	email	phone
1	1713015	Nguyễn Đức Anh Tài	1	tai@hcmut.edu.vn	0905345670
id	subject_department_id	degree_id			
1	7	4			

Bảng 5.1.3.d.6: Ví dụ về bảng `ps_student`

id	code	name	gender_id	email	phone
1	1713015	Nguyễn Đức Anh Tài	1	tai@hcmut.edu.vn	0905345670
id	education_method_id	major			
1	12	13			

5.1.4 Các bảng nhóm đề tài (tp)



Sơ đồ 5.1.4: ERD diagram của nhóm tp

5.1.4.a Bảng tp_topic

Dữ liệu: Thông tin về đề tài

Ràng buộc: min_student_take, max_student_take >= 1

Bảng 5.1.4.a: Đặc tả chi tiết bảng tp_topic

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
code	string	không	không	Mã đề tài
name	json	có	không	Tên đề tài
semester	integer	có	không	Học kỳ bắt đầu
major_id	bigint	có	không	Chuyên ngành của đề tài
education_method_id	bigint	có	không	Phương thức đào tạo của đề tài
min_student_take	integer	có	không	Số sinh viên tối thiểu
max_student_take	integer	có	không	Số sinh viên tối đa
description	json	có	không	Mô tả đề tài
topic_task	json	có	không	Nhiệm vụ giai đoạn đề cương
thesis_task	json	có	không	Nhiệm vụ giai đoạn luận văn
note	text	có	không	Ghi chú thêm về đề tài

5.1.4.b Bảng tp_council

Dữ liệu: Thông tin về hội đồng

Bảng 5.1.4.b: Đặc tả chi tiết bảng tp_council

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
subject_department_id	bigint	có	không	Phòng ban của hội đồng
reserveRoom	string	có	không	Phòng đặt cho ngày bảo vệ
reserveDate	date	có	không	Ngày bảo vệ luận văn
startTime	time without timezone	có	không	Thời gian bắt đầu bảo vệ
endTime	time without timezone	có	không	Thời gian dự kiến kết thúc
role_id	bigint	có	không	Các vai trò trong hội đồng
teacher_code	array-string	có	không	Mã giảng viên tương ứng với vai trò
note	text	có	không	Ghi chú thêm về hội đồng

5.1.4.c Bảng tp_topic_assign

Dữ liệu: Thông tin về việc gán các nghiệp vụ cho đề tài

Bảng 5.1.4.c: Đặc tả chi tiết bảng tp_topic_assign

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
topic_id	bigint	không	không	Đề tài cần được gán nghiệp vụ
semester	integer	không	không	Học kì được gán của đề tài
status_id	bigint	không	không	Trạng thái hiện tại của đề tài
execute_student_code	array-string	có	không	Mã những sinh viên thực hiện đề tài
guide_teacher_code	array-string	có	không	Mã những giảng viên hướng dẫn
review_teacher_code	array-string	có	không	Mã những giảng viên phản biện
council_id	bigint	có	không	Id của hội đồng
note	text	có	không	Ghi chú thêm

5.1.4.d Ví dụ về bảng tp_topic, tp_council, tp_topic_assign

Một đề tài có tên tiếng việt là "ABC" tên tiếng anh là "A", mã đề tài là 123, được đăng ký ở học kỳ 201, dành cho tối đa 3 sinh viên khoa học máy tính chính quy. Nhiệm vụ giai đoạn đề cương là: thiết kế db. Nhiệm vụ giai đoạn luận văn là: code.

Ở học kỳ 201, đề tài này được hướng dẫn bởi 1 giảng viên có mã số là 0001 và được thực hiện bởi 2 sinh viên có mã số là 1713015 và 1713016. Cuối học kì được phản biện bởi giảng viên có mã số là 0002

Ở học kì 202, vì dịch covid nên nhóm sinh viên xin bảo lưu đề tài.

Ở học kì 203, vì lý do cá nhân, sinh viên có mã số 1713016 rút khỏi đề tài. Sinh viên mã số 1713015 tiếp tục đề tài, trải qua sự phản biện của 2 giảng viên có mã số 0002 và 0003. Sau đó sinh viên đã bảo vệ đề tài dưới hội đồng của phòng ban hệ thống thông tin gồm chủ tịch mã số 1111, thư ký mã số 1113 tại phòng 404-H6.

Bảng 5.1.4.d.1: Chuẩn bị bảng br_const_data

id	type	value	no
1	major	{"en":"Computer Science","vi":"Khoa học máy tính"}	1
2	major	{"en":"Computer Engineering","vi":"Kỹ thuật máy tính"}	2
3	major	{"en":"Interdisciplinary","vi":"Liên ngành"}	3
4	educationMethod	{"en":"Formal","vi":"Chính quy"}	null
5	educationMethod	{"en":"advanced","vi":"Tiên tiến chất lượng cao"}	null
6	topicStatus	{"en":"Outline","vi":"Đề cương"}	1
7	topicStatus	{"en":"Thesis","vi":"Luận văn"}	2
8	topicStatus	{"en":"Reserve","vi":"Bảo lưu"}	3
9	subjectDepartment	{"en":"Information System","vi":"Hệ thống thông tin"}	null
10	subjectDepartment	{"en":"Software Technology","vi":"Công nghệ phần mềm"}	null
11	councilRole	{"en":"Secretary 1","vi":"Thư ký 1"}	3
12	councilRole	{"en":"Secretary 2","vi":"Thư ký 2"}	4
13	councilRole	{"en":"Chairman","vi":"Chủ tịch"}	1
14	councilRole	{"en":"Vice Chairman","vi":"Phó chủ tịch"}	2

Bảng 5.1.4.d.2: Ví dụ về bảng tp_topic

id	code	name	semester	major_id	education_method_id	description
1	123	{"en": "A", "vi": "ABC"}	201	1	4	null

id	min_student_take	max_student_take	topic_task	thesis_task	note
1	1	3	{"en": null, "vi": "thiết kế db"}	{"en": null, "vi": "code"}	"Thiếu mô tả đề tài"

Bảng 5.1.4.d.3: Ví dụ về bảng tp_council

id	subject_department_id	reserveRoom	reserveDate	startTime	endTime
1	9	404-H6	null	null	null

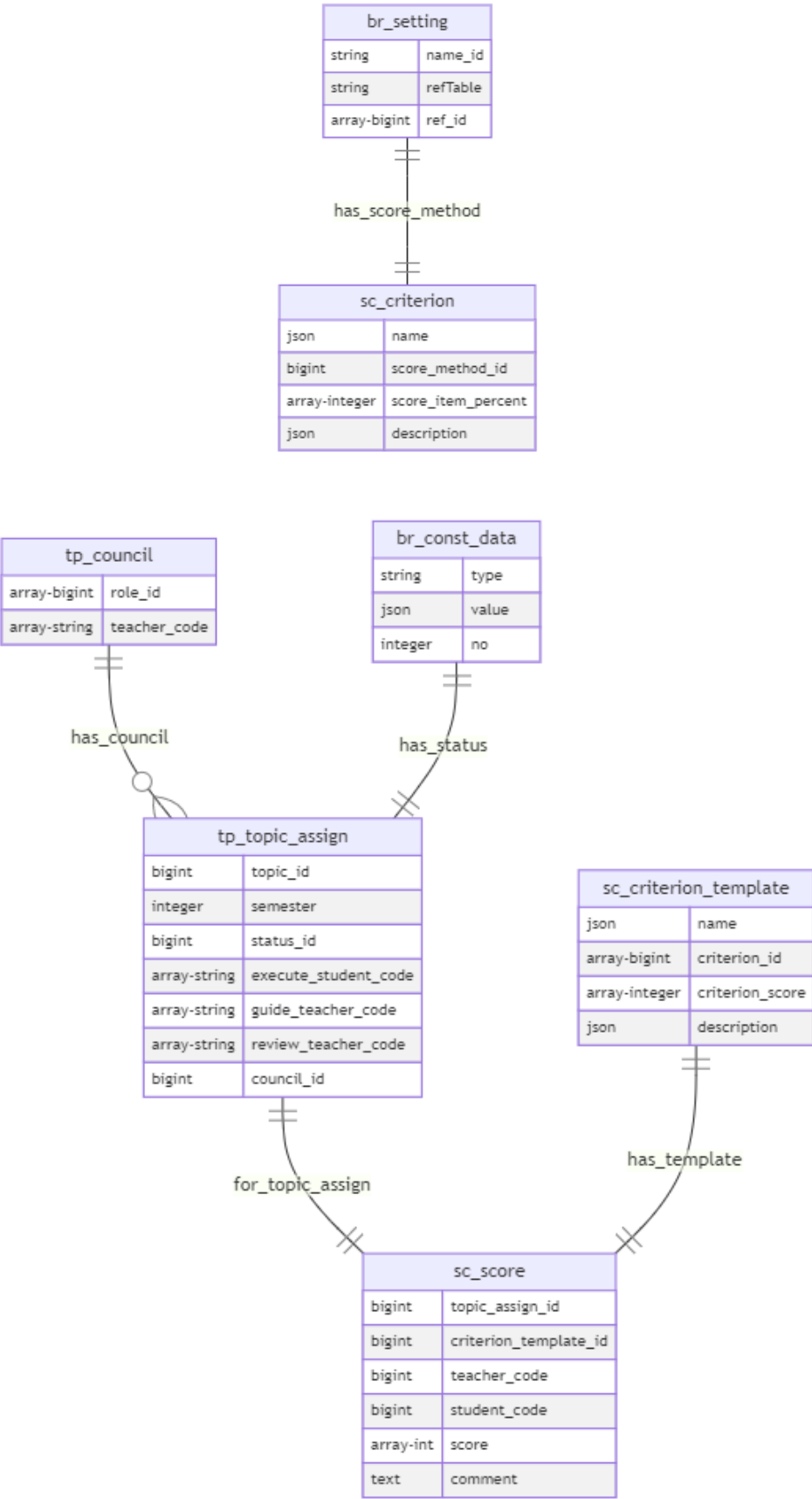
id	role_id	teacher_code	note
1	[13,14,11,12]	[1113,null,1111,null]	"Thiếu ngày và thời gian, thiếu thành phần hội đồng"

Bảng 5.1.4.d.4: Ví dụ về bảng tp_topic_assign

id	topic_id	semester	status_id	execute_student_code	council_id
1	1	201	6	[1713015, 1713016]	null
2	1	202	8	[1713015, 1713016]	null
3	1	203	7	[1713015]	1

id	guide_teacher_code	review_teacher_code	note
1	[0001]	[0002]	null
2	[0001]	null	"vì dịch covid nên nhóm sinh viên xin bảo lưu đề tài"
3	[0001]	[0002, 0003]	"vì lý do cá nhân, sinh viên có mã số 1713016 rút khỏi đề tài."

5.1.5 Các bảng nhóm điểm (sc)



Sơ đồ 5.1.5: ERD diagram của nhóm sc

5.1.5.a Bảng sc_criterion

Dữ liệu: Thông tin tiêu chí

Bảng 5.1.5.a: Đặc tả chi tiết bảng sc_criterion

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
name	json	không	không	Tên tiêu chí
score_method_id	bigint	không	không	Phương thức chấm điểm
score_item_percent	array-integer	không	không	Phần trăm số điểm mỗi bậc của tiêu chí
description	json	có	không	Mô tả tiêu chí

5.1.5.b Bảng sc_criterion_template

Dữ liệu: Thông tin về mẫu tiêu chí

Bảng 5.1.5.b: Đặc tả chi tiết bảng sc_criterion_template

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
name	json	không	không	Tên tiêu chí
criterion_id	array-bigint	không	không	Danh sách tiêu chí
criterion_score	array-integer	không	không	Điểm tối đa cho mỗi tiêu chí
description	json	có	không	Mô tả về mẫu tiêu chí

5.1.5.c Bảng sc_score

Dữ liệu: Thông tin về điểm số

Bảng 5.1.5.c: Đặc tả chi tiết bảng sc_score

Trường	Kiểu dữ liệu	Chứa null	Mặc định	Mô tả
topic_assign_id	bigint	không	không	Chấm điểm cho giai đoạn của đề tài
criterion_template_id	bigint	không	không	Mẫu được dùng để chấm điểm
teacher_code	string	không	không	Mã giảng viên chấm điểm
student_code	string	không	không	Mã sinh viên được chấm điểm
score	array-integer	có	không	Danh sách điểm tương ứng với danh sách tiêu chí trong mẫu
comment	text	có	không	Bình luận của giảng viên khi chấm điểm

5.1.5.d Ví dụ về bảng `sc_criterion`, `sc_criterion_template`, `sc_score`

Một mẫu tiêu chí chấm điểm số 45 được dùng chấm đề cương và cả luận văn, gồm 2 tiêu chí là "Đạt 80% khối lượng công việc" và "Báo cáo chi tiết". 2 tiêu chí này được chấm theo phương thức ABC. Tiêu chí đầu tiên có số điểm tối đa là 20 điểm, tiêu chí thứ hai tối đa 10 điểm. Tiêu chí 1 thì A được toàn bộ số điểm, B được một nửa điểm và C được 30% điểm. Tiêu chí 2 thì A được toàn bộ số điểm, B được 80% số điểm, C được 60% số điểm.

Bảng 5.1.5.d.1: Chuẩn bị bảng `br_const_data`

id	type	value	no
1	scoreMethodItem	{"en":"A","vi":"A"}	1
2	scoreMethodItem	{"en":"C","vi":"C"}	3
3	scoreMethodItem	{"en":"B","vi":"B"}	2
4	scoreMethodName	{"en":null,"vi":"Phương thức chấm điểm ABC"}	null
5	topicTemplate	{"en":null,"vi":"Mẫu tiêu chí chấm đề cương hiện tại"}	null
6	thesisTemplate	{"en":null,"vi":"Mẫu tiêu chí chấm luận văn hiện tại"}	null

Bảng 5.1.5.d.2: Chuẩn bị bảng `br_setting`

id	name_id	ref_table	ref_id
21	4	br_const_data	[1,3,2]
22	5	sc_criterion_template	[41]
23	6	sc_criterion_template	[41]

Bảng 5.1.5.d.3: Ví dụ về bảng `sc_criterion`

id	name	score_method_id	score_item_percent	description
31	{"en":null,"vi":"Đạt 80% khối lượng công việc"}	21	[100,50,30]	{"en":null,"vi":"A: 100% điểm, B: 50% điểm, C: 30% điểm"}
32	{"en":null,"vi":"Báo cáo chi tiết"}	21	[100,80,60]	{"en":null,"vi":"A: 100% điểm, B: 80% điểm, C: 30% điểm"}

Bảng 5.1.5.d.4: Ví dụ về bảng `sc_criterion_template`

id	name	criterion_id	criterion_score	description
41	{"en":null,"vi":"Mẫu số 45"}	[31,32]	[20,10]	null

Sử dụng mẫu chấm điểm trong ví dụ trên.

Sinh viên có mã số 1713015 làm luận văn đề tài có id 123 ở học kỳ 201. Sinh viên này được giảng viên hướng dẫn mã số 0001 cho điểm A và B, giảng viên phản biện mã số 0002 cho điểm B và C. Một hội đồng gồm chủ tịch mã số 1111, thư ký mã số 1113. Thư ký đã chấm điểm cho sinh viên lần lượt điểm C và A.

Bảng 5.1.5.d.5: Chuẩn bị bảng br_const_data

id	type	value	no
7	topicStatus	{"en":"Thesis","vi":"Luận văn"}	2
8	councilRole	{"en":"Chairman", "vi":"Chủ tịch"}	1
9	councilRole	{"en":"Secretary", "vi":"Thư ký"}	2

Bảng 5.1.5.d.6: Chuẩn bị bảng tp_council

id	role_id	teacher_code
51	[8,9]	[1111,1113]

Bảng 5.1.5.d.7: Chuẩn bị bảng tp_topic_assign

id	topic_id	semester	status_id	execute_student_code
61	123	201	7	[1713015]
id	guide_teacher_code	review_teacher_code	council_id	
61	[0001]	[0002]	51	

Bảng 5.1.5.d.8: Bảng tính toán điểm số

Tiêu chí	A	B	C
Tiêu chí 1	$20 \times 100 / 100 = 20$	$20 \times 50 / 100 = 10$	$20 \times 30 / 100 = 6$
Tiêu chí 2	$10 \times 100 / 100 = 10$	$10 \times 80 / 100 = 8$	$10 \times 60 / 100 = 6$

Bảng 5.1.5.d.9: Ví dụ về bảng sc_score

id	topic_assign_id	criterion_template_id	teacher_code	student_code	score	comment
71	61	41	0001	1713015	[20,8]	null
72	61	41	0002	1713015	[10,6]	null
73	61	41	1111	1713015	null	null
74	61	41	1113	1713015	[6,10]	null

5.2 Thiết kế API

5.2.1 Các API cơ sở

Định nghĩa một số API cơ bản phục vụ thêm sửa xóa

{x} = /api + [/const, /setting, /teacher, /student, /topic, /council, /topic/assign, /criterion, /criterion/template, /score]

{entity} là table tương ứng {x}

Bảng 5.2.1: Các API cơ bản

Mã API	Endpoint	Phương thức	Tham số	Kết quả
BA01	{x}/	GET		Các record được build
BA02	{x}/paging	GET	int:number, int:size, string: sort, bool:descend	Các record tại trang <i>number</i> , được sắp xếp theo <i>sort</i>
BA03	{x}/	POST	{entity}	Tạo mới hoặc cập nhật dữ liệu
BA04	{x}/all	POST	array-{entity}	Tạo mới dữ liệu số lượng nhhieuf
BA05	{x}/example	POST	{entity}	Dữ liệu khớp với <i>example</i>
BA06	{x}/	DELETE		Xóa tất cả dữ liệu trong bảng
BA...	{x}/...

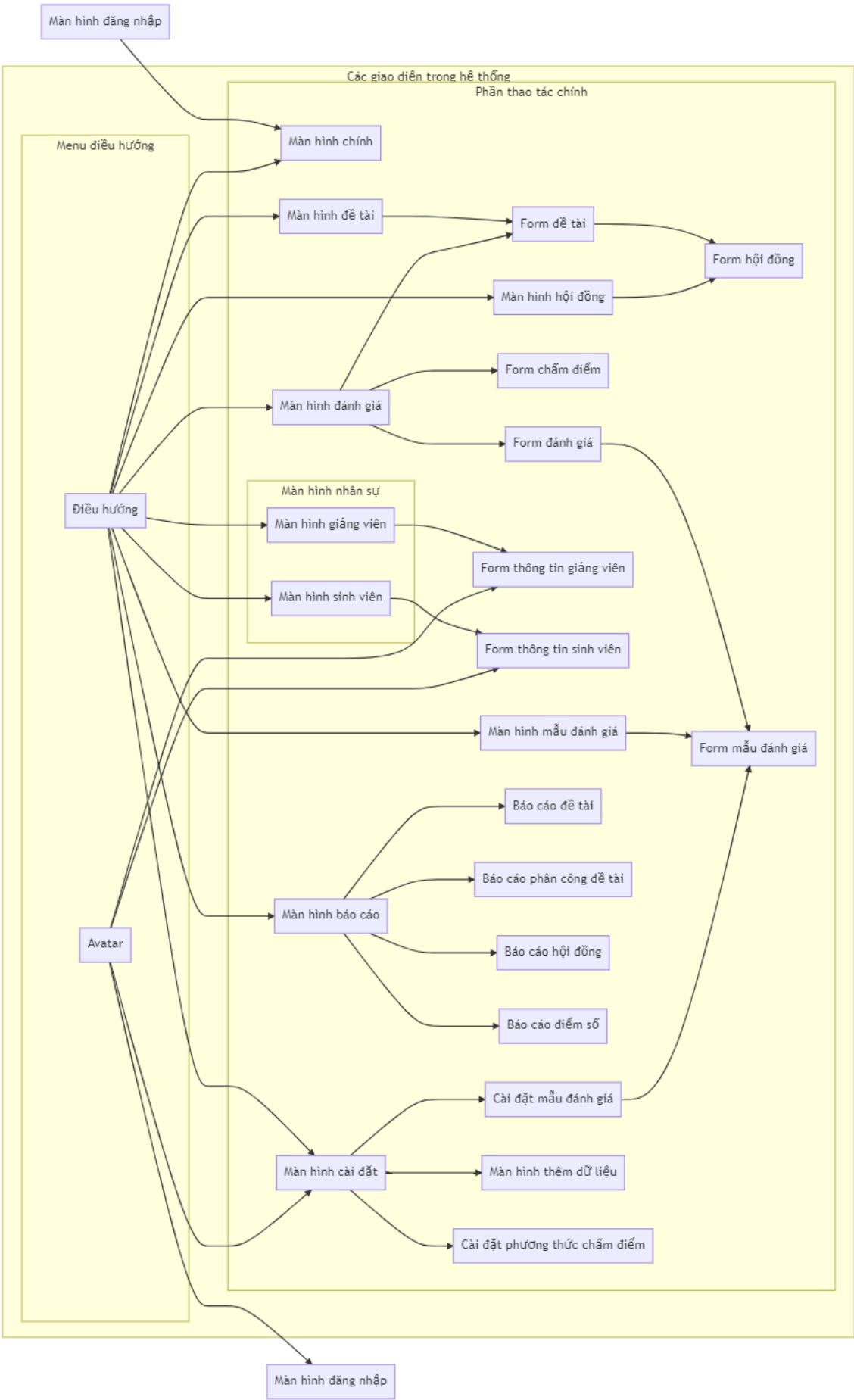
5.2.2 Các API khác

Các API phục vụ riêng một số nghiệp vụ nhất định

Bảng 5.2.2: Các API khác

Mã API	Endpoint	Phương thức	Tham số	Kết quả
CO01	/const/types	GET		Thông tin các dữ liệu trong bảng br_const_data được group by type order by no
SE01	/setting/template	GET	bool:thesis	Thông tin mẫu đánh giá đề cương hoặc luận văn hiện tại setting
SE02	/setting/template	POST	bool:thesis, {entity}	Setting thông tin mẫu đánh giá dùng cho đề cương hoặc luận văn
TC01	/teacher/search	GET	string:value	Tìm kiếm giảng viên
SD02	/student/search	GET	string:value	Tìm kiếm sinh viên
TP01	/topic/assign/search/teacher	GET	string:code, string:sort, bool:descend	Tìm kiếm các topic assign theo mã số giảng viên
TP02	/topic/assign/search/topic	GET	string:topicName, string:sort, bool:descend	Tìm kiếm các topic assign theo tên đề tài
RP01	/report/topic	GET	localTime:startTime, localTime:endTime, bool:file, set:topicId	Xuất báo cáo đề tài
RP01	/report/topicAssign	GET	localTime:startTime, localTime:endTime, bool:file, set:topicAssignId	Xuất báo cáo phân công đề tài
RP01	/report/council	GET	localTime:startTime, localTime:endTime, bool:file, set:councilId	Xuất báo cáo hội đồng
RP01	/report/score	GET	localTime:startTime, localTime:endTime, bool:file, set:scoreId	Xuất báo cáo điểm
IP01	/import/template	GET	string:type	Xuất mẫu file thêm dữ liệu
...

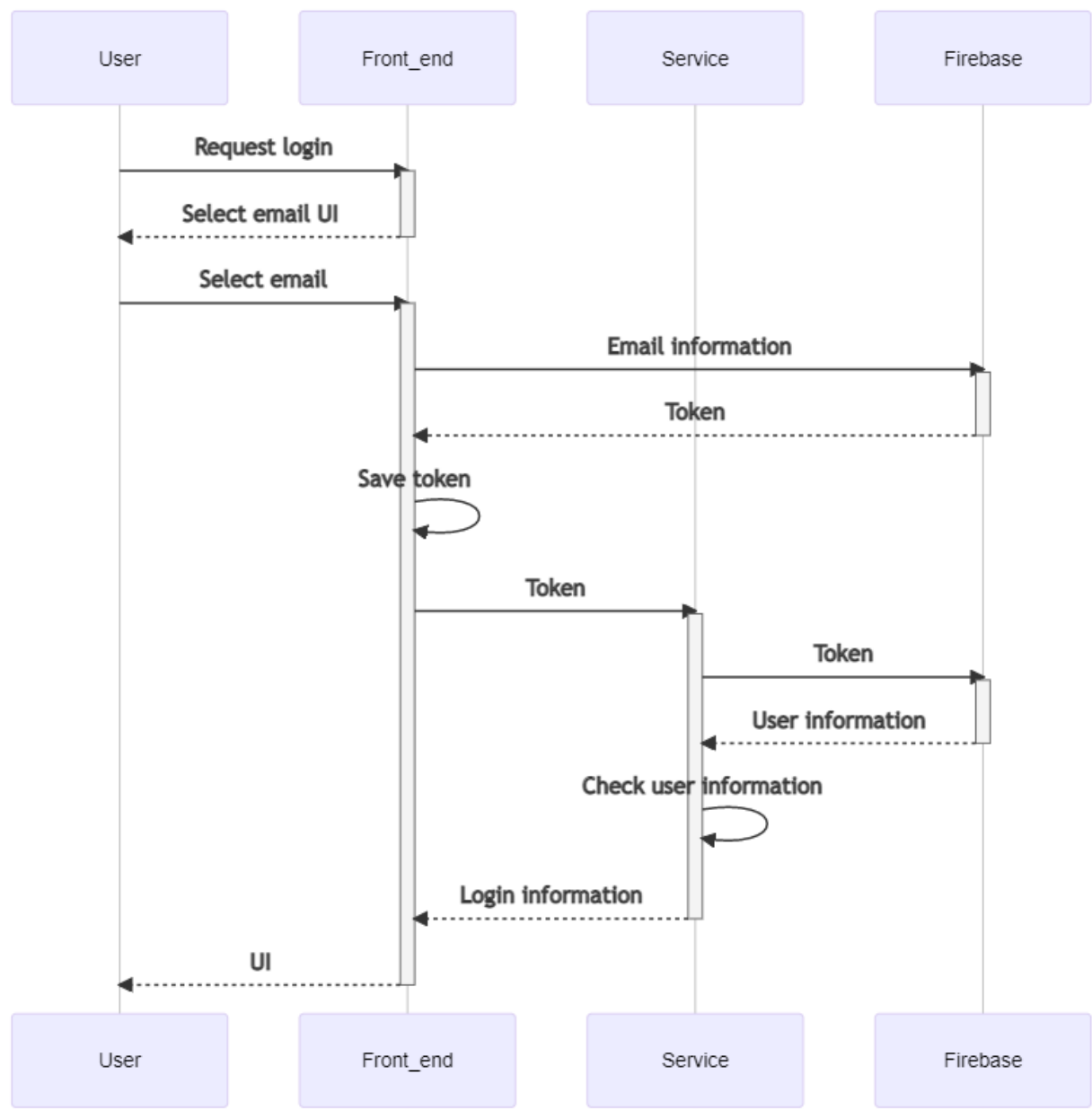
5.3 Thiết kế luồng giao diện



Sơ đồ 5.3: UI flow chart

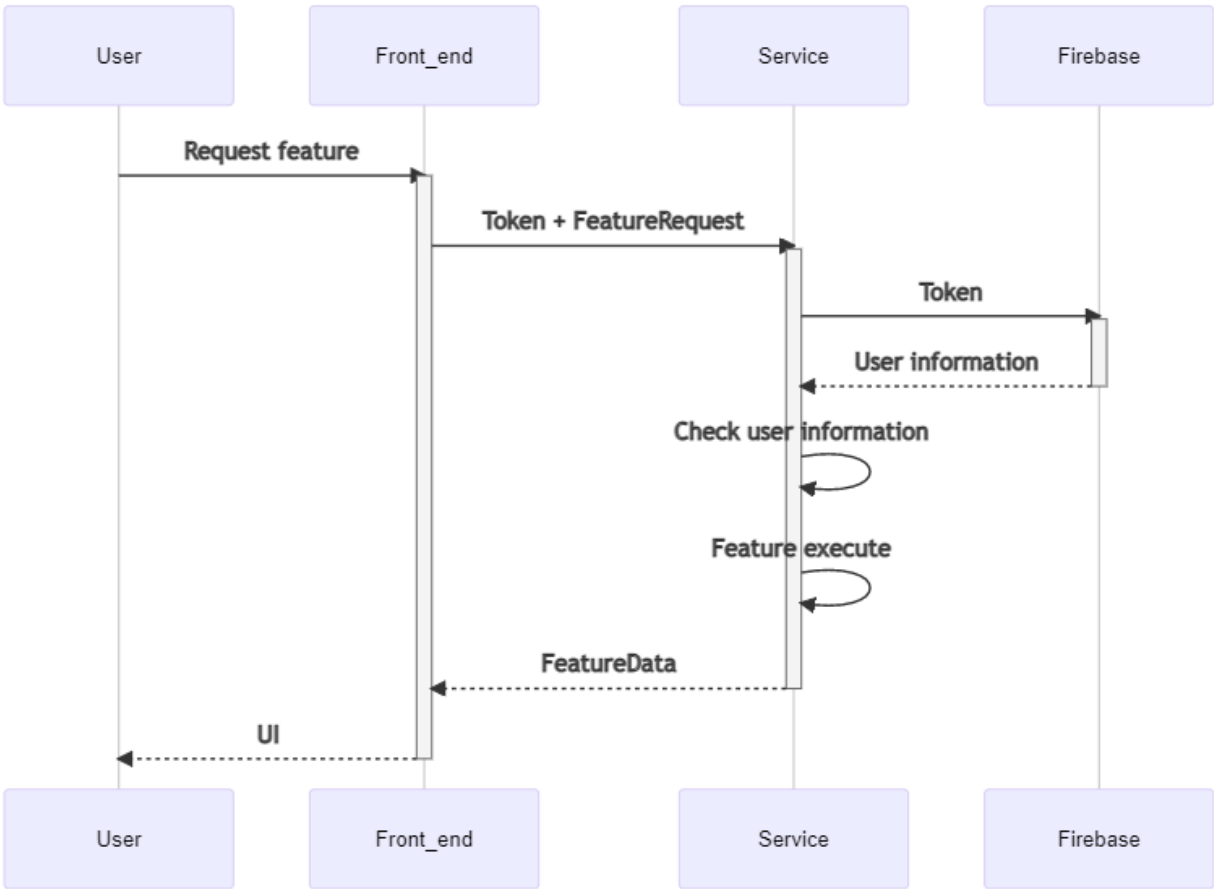
5.4 Thiết kế một số chức năng

5.4.1 Chức năng xác thực người dùng



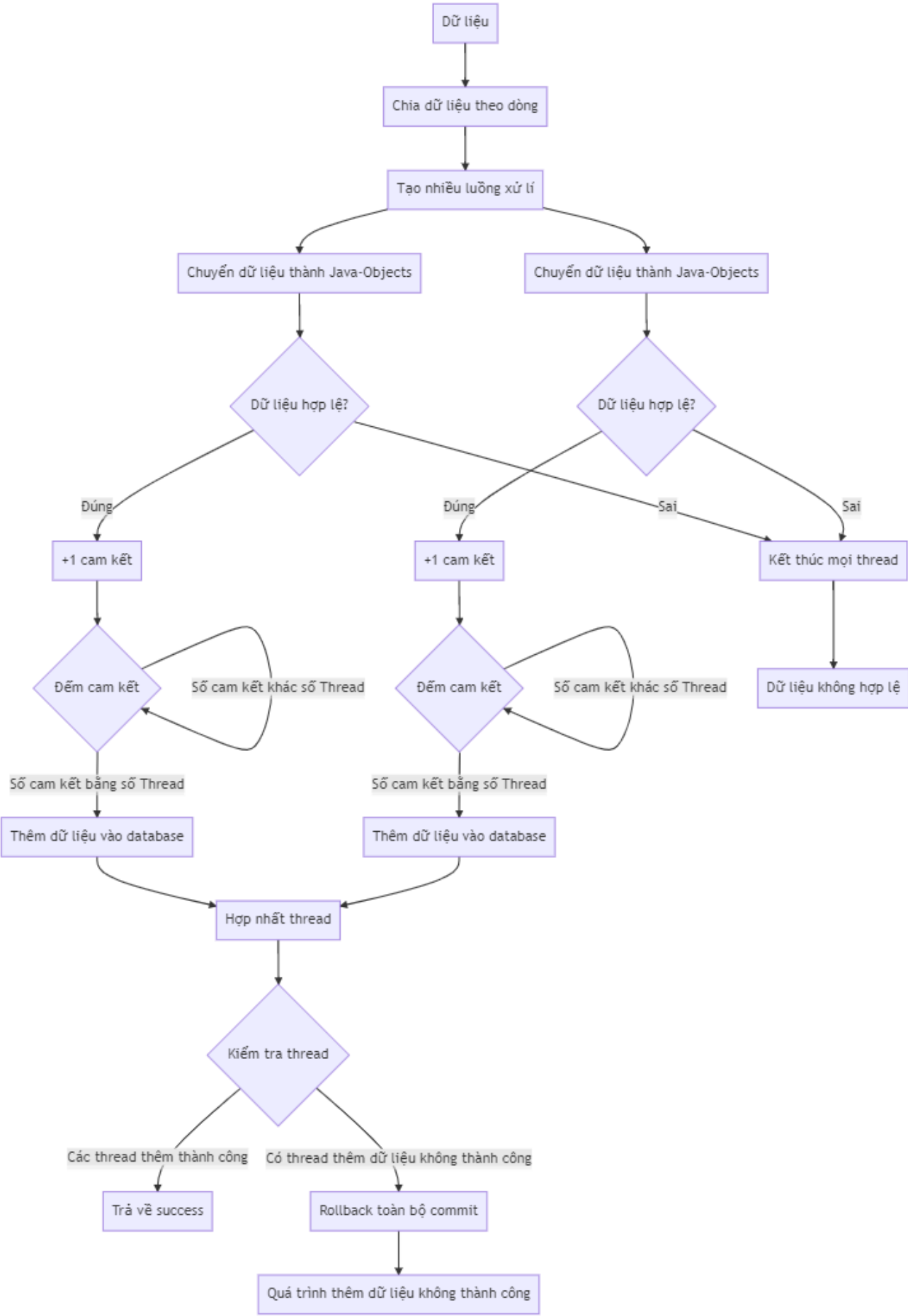
Sơ đồ 5.4.1: Login sequence diagram

5.4.2 Chức năng xác thực các yêu cầu từ người dùng



Sơ đồ 5.4.2: Valid request sequence diagram

5.4.3 Chức năng thêm dữ liệu



Sơ đồ 5.4.3: Import data sequence diagram

6 Hiện thực hệ thống

Sau khi thiết kế, nhóm đã triển khai thi công phần mềm bám sát những gì đã vẽ ra.

Trong chương này, nhóm sẽ đưa ra bảng phân chia nhiệm vụ, từ đó hiện thực một phần của hệ thống trong giai đoạn đề cương.



Hình 6: Hiện thực hệ thống

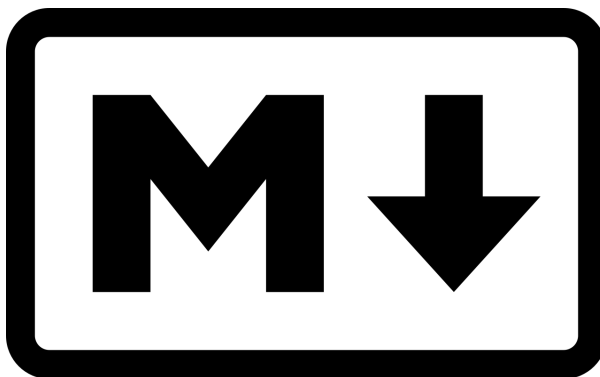
6.1 Lưu trữ

Tài liệu: Các tài liệu tìm kiếm được để hỗ trợ về mặt nghiệp vụ

<https://github.com/datai999/thesis-document>

Báo cáo: Báo cáo đề cương, báo cáo luận văn được viết bằng code markdown.

Markdown là một ngôn ngữ đánh dấu đơn giản với cú pháp được định dạng bằng cách dùng văn bản thô



Hình 6.1.a Logo Markdown

Ưu điểm:

- Đơn giản
- Có thể chèn mã HTML
- Quen thuộc với các lập trình viên (ví dụ: README.md)
- Có thể quản lý version và lịch sử thông qua git

<https://github.com/datai999/thesis-document/tree/main/report>

Code front-end: Code giao diện

<https://github.com/datai999/thesis>

Code back-end: Code server

<https://github.com/datai999/thesis-service>

Database staging: Cơ sở dữ liệu mỗi trường staging
<https://www.heroku.com/>


Hình 6.1.b Logo Heroku

Gmail: Thông tin dùng cho đăng nhập
<https://firebase.google.com/>


Hình 6.1.c Logo Firebase

App android staging: Các file .apk được build trên trang web Expo sau đó đc tải xuống và upload lại lên Google Drive.

The screenshot shows the Expo web interface. At the top, there's a header with 'Expo' logo, user 'datai28599', and project 'thesis'. A search bar and navigation icons are on the right. On the left, a sidebar lists 'Dashboard', 'General', 'Activity', 'Builds' (selected), 'Credentials', and 'Secrets'. The main area is titled 'Builds' with the subtitle 'View all of the builds associated with your project.' Below this, there are two build entries for the 'thesis' project. Each entry has a table with columns: Status, Platform, Distribution, Channel, SDK Version, and Build Type. The first build is 'Finished' on 'Android', distributed on 'Play Store', with 'default' channel, '41.0.0' SDK version, and 'Classic' build type, completed '8 days ago'. The second build is identical but completed '9 days ago'.

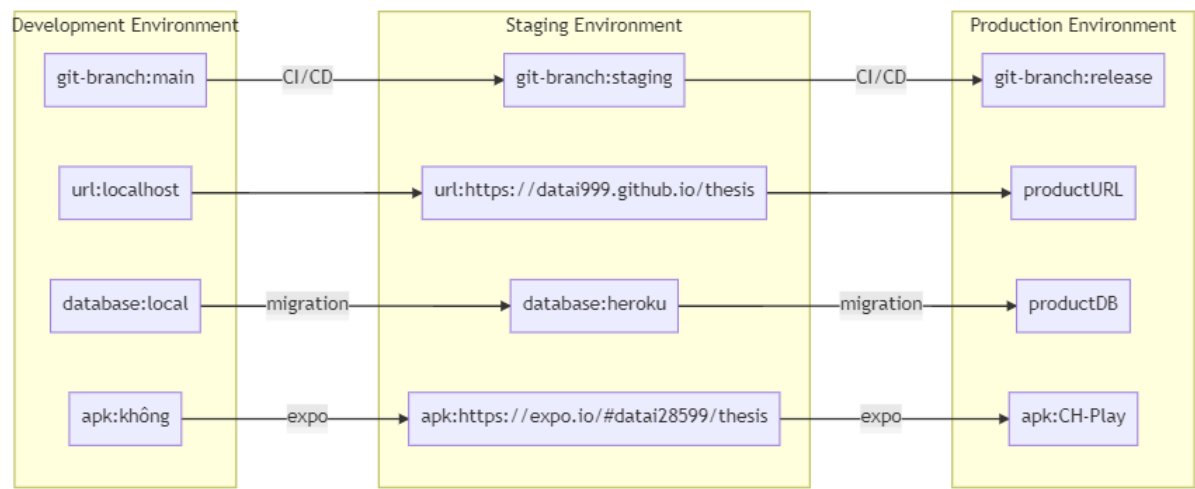
Status	Platform	Distribution	Channel	SDK Version	Build Type
Finished	Android	Play Store	default	41.0.0	Classic
Finished	Android	Play Store	default	41.0.0	Classic

Hình 6.1.d Expo web Builds

Thư mục Google Drive giữa sinh viên và giảng viên

6.2 Các môi trường

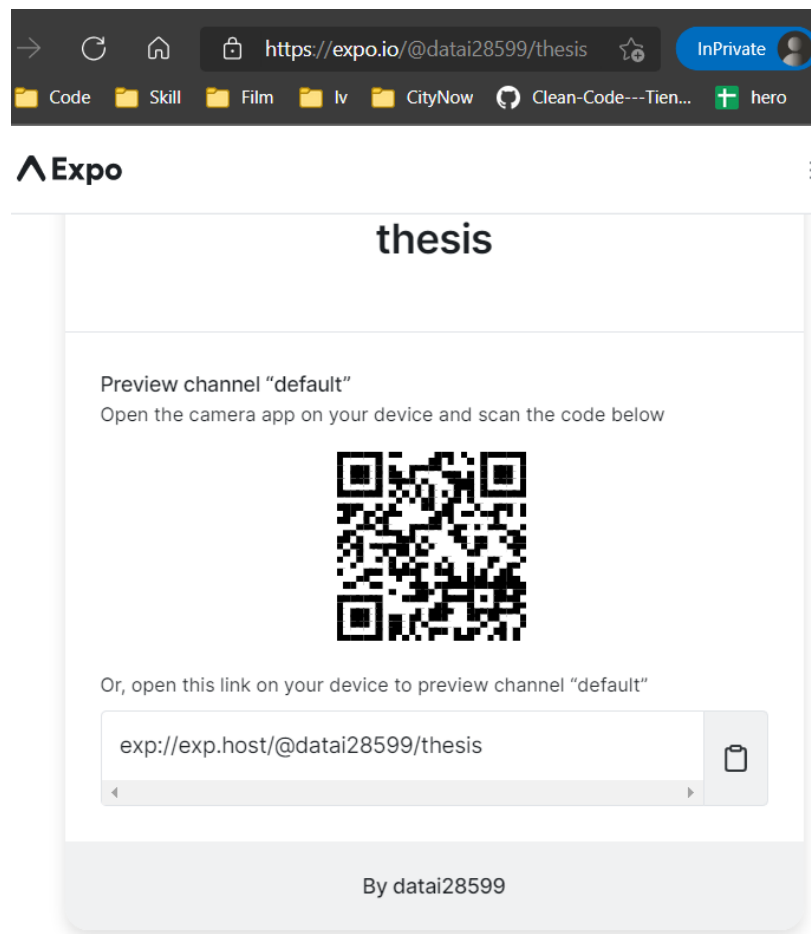
Với dự án này nhóm sẽ sử dụng 3 môi trường để phát triển ứng dụng.



Sơ đồ 6.2: Environment diagram

Vì công cụ vẽ không cho phép kí tự @ nên nhóm đã để dấu # thay thế.

Staging Environment file apk: <https://expo.io/#datai28599/thesis>



Hình 6.2 Trang web Expo app

6.3 Chia nhiệm vụ

Với những tính năng của hệ thống và cách thiết kế cũng như công nghệ đã chọn. Nhóm chia thành 20 nhiệm vụ chính được phân bổ theo bảng như sau

Bảng 6.3: Phân chia nhiệm vụ

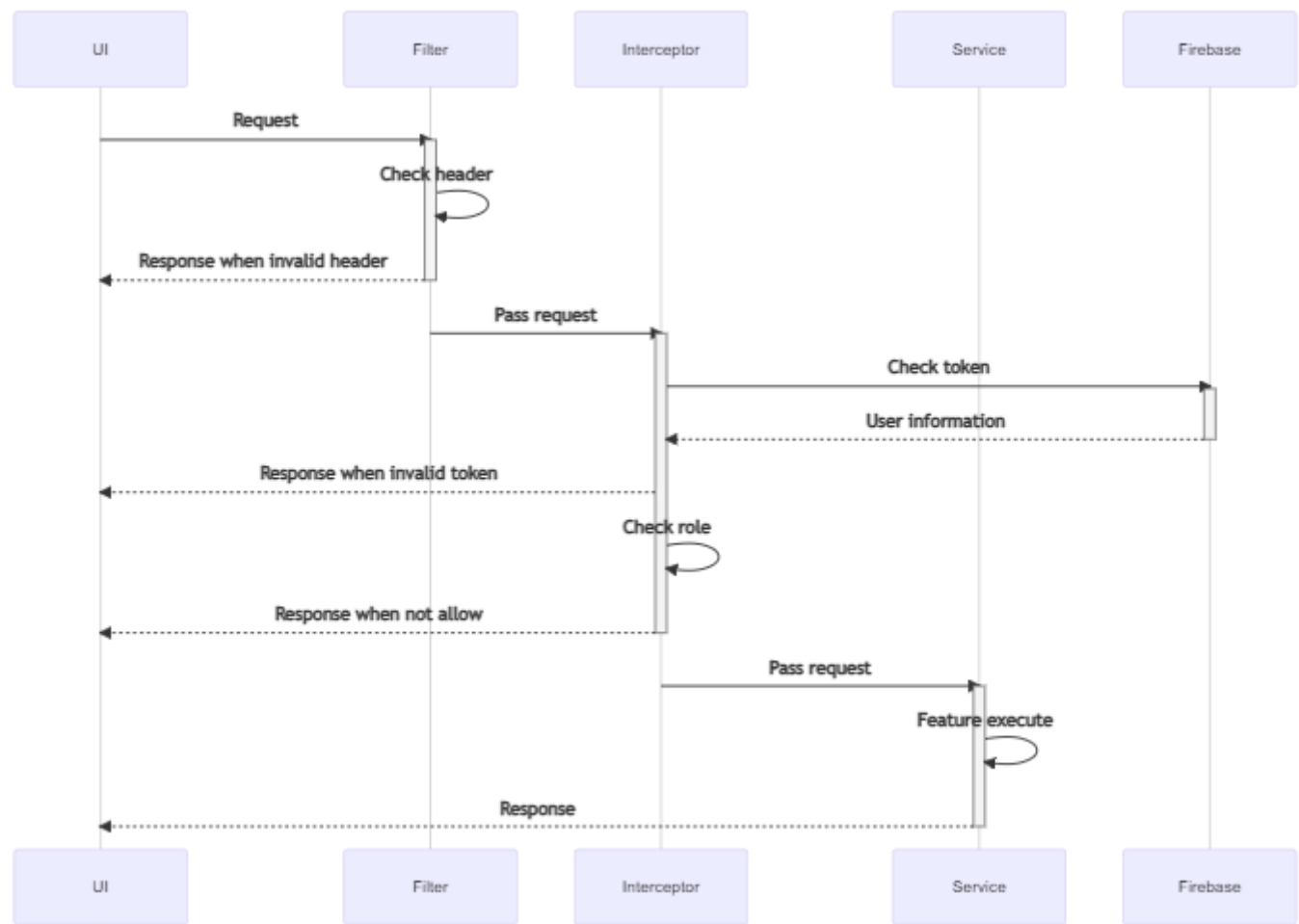
STT	Thành phần	Web	Android	IOS	Staging	Production
1	Menu điều hướng	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
2	Chức năng về đề tài	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
3	Chức năng về giảng viên	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
4	Chức năng về sinh viên	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
5	Chức năng về nhân viên	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
6	Chức năng về hội đồng	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
7	Phân công đề tài	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
8	Chức năng về tiêu chí	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
9	Chức năng về mẫu đánh giá	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
10	Cài đặt mẫu đánh giá hiện tại	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
11	Đăng nhập	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
12	Đăng xuất	Đề cương	Đề cương	Luận văn	Đề cương	Luận văn
13	Chấm điểm	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
14	Xuất báo cáo	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
15	Thêm dữ liệu từ file Excel	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
16	Phân chia quyền người dùng	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
17	Các cài đặt hệ thống	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
18	Thông báo qua mail	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
19	Kiểm tra trên IOS	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn
20	Đưa lên môi trường Production	Luận văn	Luận văn	Luận văn	Luận văn	Luận văn

6.4 Filter và Interception request

Hiện thực chứng năng kiểm tra token ở mục **6.4.2**.

Với Spring Boot, lớp filter không thể can thiệp sâu vào hệ thống. Vậy nên ở lớp này sẽ kiểm tra xem header có đầy đủ và đúng format field hay không.

Lớp Interceptor có thể can thiệp các tài nguyên trong hệ thống. Lớp này sẽ kiểm tra token và xác thực quyền hạn của người dùng có được phép sử dụng dịch vụ tương ứng hay không.



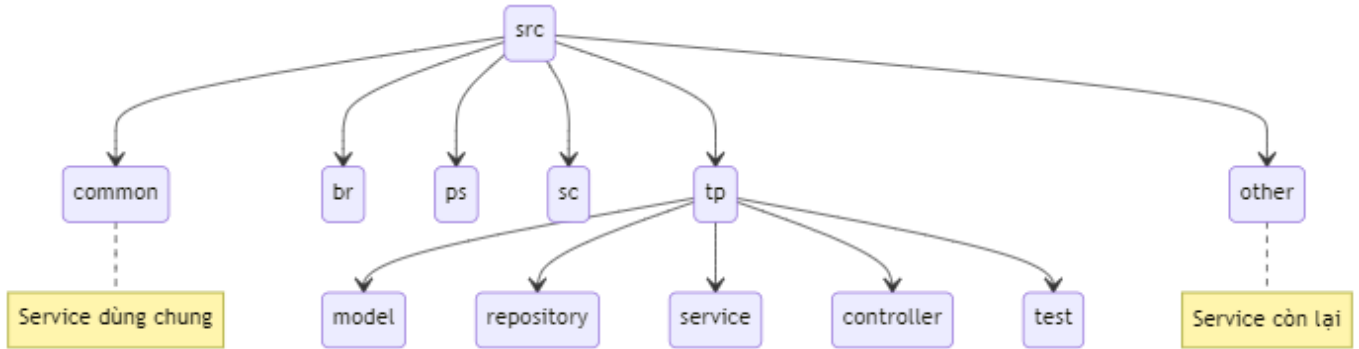
Sơ đồ 6.4: Valid request implement sequence diagram

6.5 Code back-end

6.5.1 Phân chia source code

Tổ chức phân chia source code được triển khai theo module, các module được chia theo design group database đã được giới thiệu ở mục **5.1.1.a**

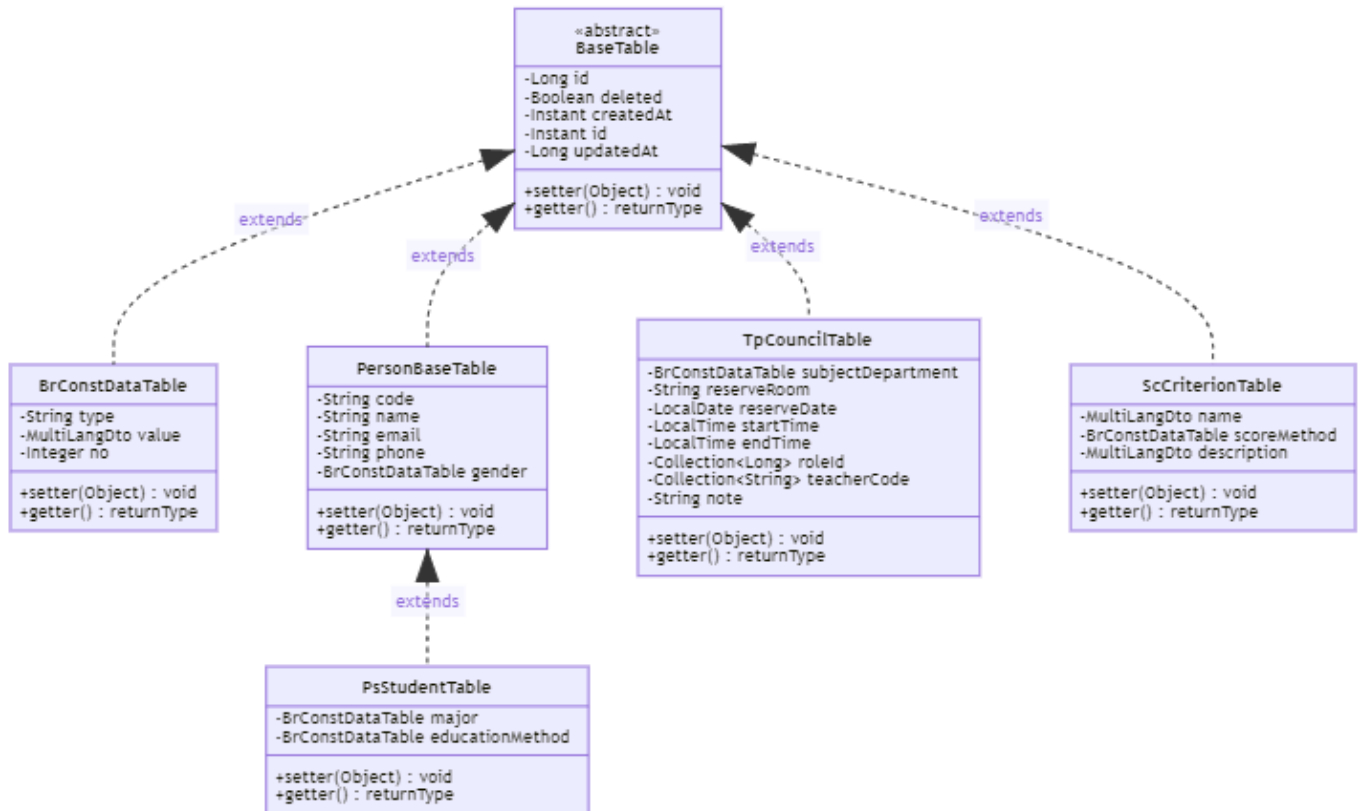
Mỗi module thì được phân chia dựa trên mô hình MVC (Model, View, Controller).



Sơ đồ 6.5.1: Phân chia source code

6.5.2 Modal Class diagram

Triển khai các thiết kế database ở mục **5.1** với đại diện các nhóm **br**, **ps**, **tp**, **sc** lần lượt là các table **br_const_data**, **ps_student**, **tp_council**, **sc_criterion**.

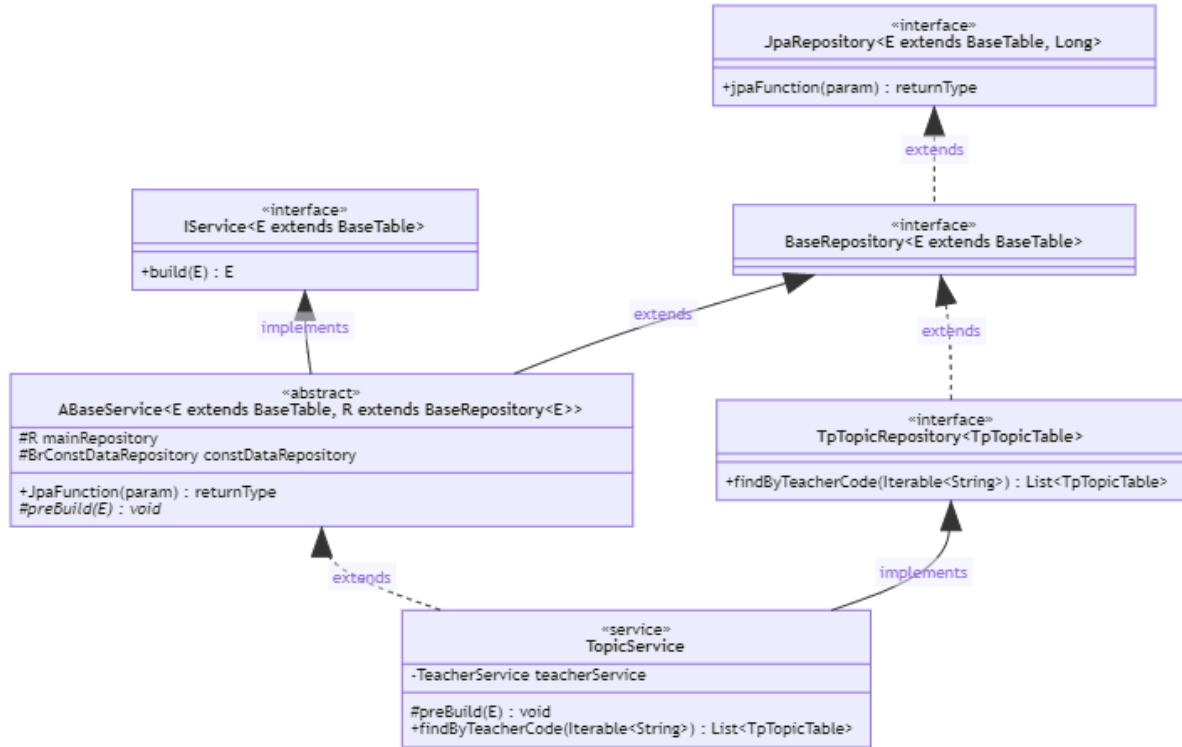


Sơ đồ 6.5.2: Modal class diagram

6.5.3 Service Class diagram

Thiết kế các service được thừa hưởng các dịch vụ có sẵn trong JpaRepository và có thể linh hoạt tùy biến các API riêng cho từng service.

Ví dụ với dịch vụ đề tài (TopicService).

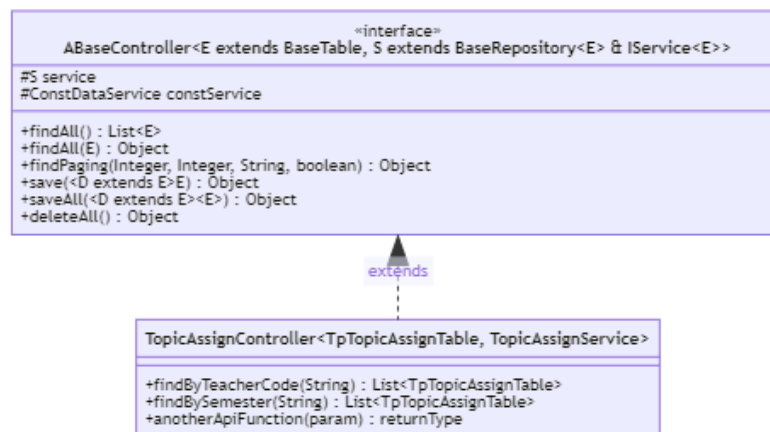


Sơ đồ 6.5.3: Service class diagram

6.5.4 Controller class diagram

Các controller (trừ các controller về import/report data) đều được extends từ một abstract class nên các controller này đều dùng chung một số API có sẵn trong **ABaseController**.

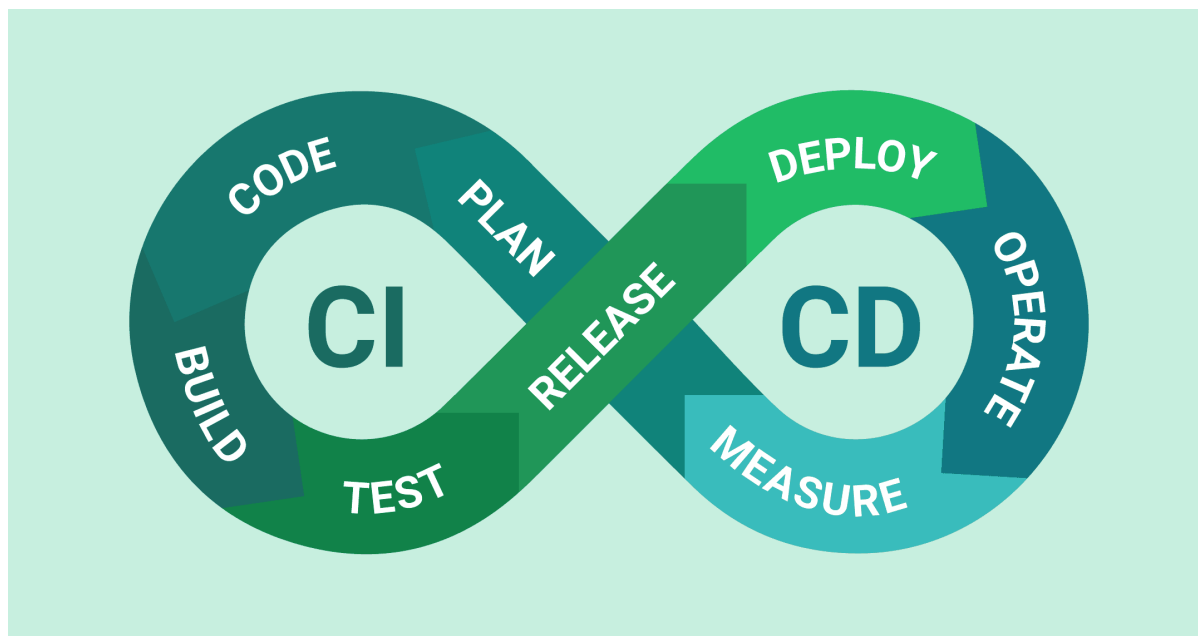
Ví dụ với các API gán các thành phần vào đề tài (**TopicAssignController**).



Sơ đồ 6.5.4: Controller class diagram

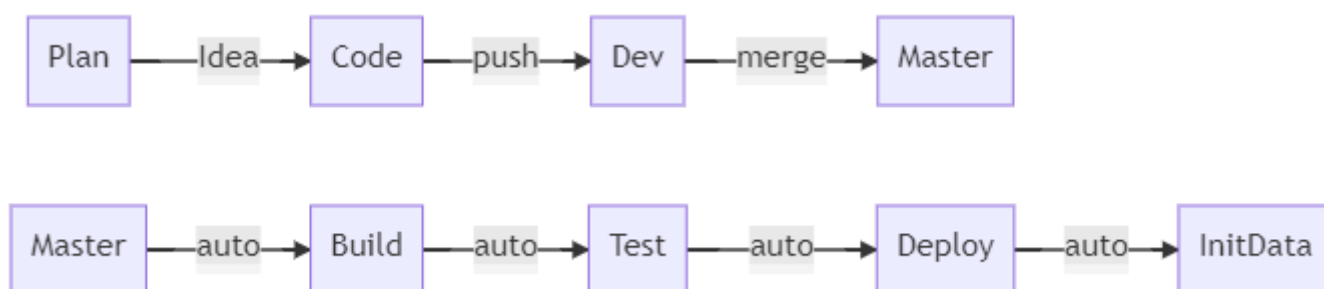
6.6 CI-CD

CI/CD là một bộ đôi công việc, bao gồm CI (Continuous Integration) và CD (Continuous Delivery), là quá trình tích hợp (integration) thường xuyên, nhanh chóng hơn khi code cũng như thường xuyên cập nhật phiên bản mới (delivery).



Hình 6.6: CI-CD

Triển khai một quy trình CI-CD đơn giản.



Sơ đồ 6.6: Triển khai CI-CD

- Plan: Yêu cầu về tính năng, kế hoạch sửa chữa,...
- Code: Hiện thực plan
- Dev: Nhánh develop thuộc môi trường develop
- Master: Nhánh thuộc môi trường staging(hoặc production)
- Build: build app, kiểm tra error và đếm số warning
- Test: Thực hiện một lệnh call API ở back-end
- Deploy: Deploy lên môi trường staging(hoặc production)
- InitData: Chỉ có bên code back-end, thêm một số dữ liệu cần thiết vào database

7 Tổng kết và hướng phát triển

Cuối cùng, nhóm sẽ nhìn nhận, đánh giá chung về những việc đã làm trong giai đoạn đề cương, điểm ra những thuận lợi và khó khăn trong giai đoạn vừa qua từ đó đề ra phương hướng phát triển cho đề tài trong giai đoạn tiếp theo.



Hình 7: Tổng kết và hướng phát triển

7.1 Kết quả đạt được

Trong thời gian thực hiện đề cương nhóm đã đạt được một số kết quả sau:

- Phân tích yêu cầu dự án: nắm bắt mục tiêu, phạm vi đề tài, các chức năng cần có của hệ thống. Từ đó đưa ra quyết định phù hợp cho hệ thống trong tương lai.
- Tìm hiểu cơ sở lý thuyết về các công nghệ liên quan:
 - Tìm hiểu và so sánh các nền tảng: Web App, Web-base App, Native App, Hybrid App và Cross Platform App
 - Ứng dụng đơn trang và đa trang
 - Các ngôn ngữ front-end: JavaScript, TypeScript và Dart
 - Các JavaScript framework front-end: React, Angular và Vue
 - Công nghệ back-end: Spring-boot(Java), ASP .NET(C#), Django(Python), ExpressJS(Javascript)
 - Một số loại database: Cơ sở dữ liệu quan hệ, Cơ sở dữ liệu hướng tài liệu, Cơ sở dữ liệu dạng khoá – giá trị, Cơ sở dữ liệu mô hình wide – colum, Cơ sở dữ liệu dạng bộ máy tìm kiếm
- Lựa chọn công nghệ phù hợp cho dự án
 - Web App và Cross Platform App
 - Ứng dụng đơn trang
 - Dùng framework Expo(Javascript) cho front-end
 - Dùng framework Spring-boot(Java) cho back-end
 - Cơ sở dữ liệu là PostgreSQL
- Thử nghiệm framework Expo
- Thiết kế một số thành phần của hệ thống
 - Thiết cơ sở dữ liệu
 - Một số api đơn giản
 - Luồng thao tác chính của UI
 - Một số tính năng như: xác thực người dùng, xác thực request, multi thread khi thêm dữ liệu
- Hiện thực các nhiệm vụ
 - Setup môi trường develop và staging
 - Xác thực người dùng
 - Xác thực request
 - Triển khai code back-end OOP
 - CI-CD đơn giản
 - Các chức năng cơ bản của hệ thống

7.2 Thuận lợi - Khó khăn

7.2.1 Thuận lợi

Sau thời gian nghiên cứu, tìm hiểu đề tài nhóm nhận thấy đề tài có những thuận lợi sau:

- Chưa gặp các yêu cầu về nghiệp vụ phức tạp
- Kinh nghiệm từ đề tài có thể áp dụng vào công việc trong tương lai
- Nhận được sự chia sẻ, hướng dẫn tận tình từ Th.S Nguyễn Thị Ái Thảo

7.2.2 Khó khăn

Bên cạnh những thuận lợi nêu trên thì đề tài cũng gặp rất nhiều khó khăn:

- Số lượng nghiệp vụ lớn, đa dạng
- Hiện thực trên nhiều nền tảng
- Dùng chung framework Expo cho nhiều nền tảng nên thường xuyên bị cản chức năng
- Đội ngũ và phát triển một thành viên và không có kinh nghiệm front-end

7.3 Hướng phát triển

Thực hiện các nhiệm vụ trong mục **6.3** điển hình như:

- Chức năng chấm điểm
- Chức năng xuất báo cáo
- Chức năng thêm dữ liệu
- Triển khai ứng dụng trên môi trường IOS
- Triển khai lên môi trường Production

Tài liệu tham khảo

STT	Ngày truy cập	Link
1	18/06/2021	http://itacenter.vn/cach-phan-biet-giua-native-app-cross-platform-va-hybrid-hieu-qua
2	18/06/2021	https://viblo.asia/p/tim-hieu-ve-native-app-va-hybrid-app-3P0IPnPmKox
3	18/06/2021	https://niithanoi.edu.vn/front-end-frame-work-pho-bien-nhat.html
4	18/06/2021	https://vntalking.com/react-vs-angular-vs-vue-js-lua-chon-nao-tot-nhat.html
5	18/06/2021	https://www.codeinwp.com/blog/angular-vs-vue-vs-react/
6	19/06/2021	https://vn.got-it.ai/blog/top-7-ngon-ngu-lap-trinh-backend-pho-bien-nhat
7	19/06/2021	https://niithanoi.edu.vn/web-framework-tot-nhat.html
8	19/06/2021	https://tel4vn.edu.vn/cac-loai-co-so-du-lieu/
9	25/06/2021	https://topdev.vn/blog/trien-khai-ci-cd-voi-gitlab/