Khoá luận “Xây dựng Ứng dụng Android quản lý dữ liệu hình ảnh mô tả giống lúa” do sinh viên Nguyễn Huy Hoàng (MSSV: DPM195207) thực hiện dưới sự hướng dẫn của TS. Nguyễn Văn Hòa.

Tác giả đã báo cáo kết quả nghiên cứu và được Hội đồng Khoa học và Đào tạo thông qua ngày …… tháng …… năm 2023.

|  |  |  |
| --- | --- | --- |
| **Phản biện 1** |  | **Phản biện 2** |
| *(Ký và ghi rõ chức danh, họ tên)* |  | *(Ký và ghi rõ chức danh, họ tên)* |
|  |  |  |
| **Giảng viên hướng dẫn** | | |
| *(Ký và ghi rõ chức danh, họ tên)* | | |
|  | | |

**LỜI CẢM ƠN**

Trải qua quá trình thực hiện đề tài khóa luận, em nhận thấy, việc cọ sát thực tế là vô cùng quan trọng – điều này giúp sinh viên xây dựng nền tảng lý thuyết được học ở trường vững chắc hơn. Điều này đã tạo cơ hội cho em học hỏi được nhiều kỹ năng cần thiết khi làm việc và củng cố lại kiến thức chuyên môn của bản thân. Thời gian thực hiện đề tài vừa qua, bản thân em cũng đã học hỏi và tiếp thu được nhiều kiến thức thực tế bổ ích, cũng như nâng cao những kiến thức chuyên môn sẵn có. Trên thực tế, quá trình học tập và rèn luyện bản thân từ chỗ còn bỡ ngỡ cho đến thiếu kinh nghiệm, em đã gặp phải rất nhiều khó khăn nhưng với sự giúp đỡ tận tình của quý thầy/cô khoa Công nghệ Thông tin đã nhiệt tình truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc nhất đến thầy Nguyễn Văn Hòa đã sắp xếp thời gian quý báu để hướng dẫn, nhận xét giúp em hoàn thành tốt bài báo cáo khóa luận tốt nghiệp này. Cảm ơn sự thông cảm và kiên nhẫn của thầy vì sự thiếu hụt, hạn chế về mặt kiến thức và kiên nhẫn trong quá trình hướng dẫn cho em và đó cũng là động lực để em cố gắng hoàn thiện bản thân cũng như đề tài này.

Em cũng muốn gửi lời cảm ơn đến Ban Giám hiệu và đội ngũ nhân viên tại các phòng Ban của Trường Đại học An Giang, đã tạo điều kiện thuận lợi và hỗ trợ cho em trong suốt quá trình học tập và rèn luyện tại trường.

Trong quá trình tìm hiểu, thực hiện đề tài, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp thầy, cô để em học thêm được nhiều kinh nghiệm trong các dự án sau này.

Sau cùng, em xin chân thành cảm ơn và xin gửi những lời chúc tốt đẹp nhất đến Thầy/Cô Trường đại học An Giang, Thầy/Cô Khoa Công nghệ Thông tin sức khỏe dồi dào, luôn đạt được những thành tựu to lớn trong công tác giảng dạy và tiếp tục dẫn dắt, truyền lửa cho thế hệ mai sau.

Trân trọng,

Nguyễn Huy Hoàng.

**TÓM TẮT ĐỀ TÀI**

Ngày nay, cùng với sự phát triển mạnh mẽ của công nghệ di động, và trên thực tế thông tin về các giống lúa rất đa dạng và ngày càng phức tạp nên việc quản lý dữ liệu giống lúa theo cách truyền thống dễ gây sai sót khi nhập liệu và lưu trữ. Quy trình thu thập số liệu được thực hiện bằng cách ghi chép trên giấy yêu cầu phải in tốn nhiều chi phí và gây không thuận tiện cho việc lưu trữ lâu dài.

Việc khảo sát và thu thập thông tin giống lúa trên giấy có rất nhiều mặt khó khăn. Do đó, nhu cầu quản lý dữ liệu, hình ảnh và mô tả giống lúa ngày càng tăng cao trong lĩnh vực nông nghiệp hiện nay. Với sự phát triển của công nghệ thông tin, việc sử dụng phần mềm quản lý dữ liệu để quản lý thông tin về giống lúa giúp cho việc nghiên cứu phát triển, bảo mật và trở nên hiệu quả, năng suất hơn.

Ứng dụng Android quản lý hình ảnh mô tả giống lúa sử dụng Flutter – được tạo ra nhằm thiết thực hóa hơn trong việc khảo sát và thu thập số liệu giống lúa, giúp các nhà nghiên cứu, chuyên gia trong lĩnh vực nông nghiệp có thể lưu trữ, quản lý liệu tốt nhất và có kế hoạch nhất. Ứng dụng lấy dữ liệu từ hệ thống API và cung cấp giao diện đơn giản phù hợp với mọi người dùng để có thể tương tác với hệ thống trong việc tra cứu và nhập liệu, điều chỉnh dữ liệu trong quá trình khảo sát và thu thập số liệu thực tế.

**MỤC LỤC**

[CHƯƠNG 1: ĐẶT VẤN ĐỀ 1](#_Toc135115133)

[1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI 1](#_Toc135115134)

[1.2. LÝ DO CHỌN ĐỀ TÀI 1](#_Toc135115135)

[1.3. MỤC TIÊU NGHIÊN CỨU CỦA ĐỀ TÀI 2](#_Toc135115136)

[1.4. ĐẶT VẤN ĐỀ 2](#_Toc135115137)

[1.5. PHẠM VI CỦA ĐỀ TÀI 2](#_Toc135115138)

[1.6. PHƯƠNG PHÁP NGHIÊN CỨU 3](#_Toc135115139)

[CHƯƠNG 2: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT 4](#_Toc135115140)

[2.1. TỔNG QUAN 4](#_Toc135115141)

[2.2. CƠ SỞ LÝ THUYẾT 4](#_Toc135115142)

[2.2.1. Hệ điều hành Android 4](#_Toc135115143)

[2.2.2. Tổng quan về Flutter 7](#_Toc135115144)

[2.2.3. Ngôn ngữ lập trình Dart 8](#_Toc135115145)

[2.2.4. Tổng quan về API 11](#_Toc135115146)

[2.2.5. JSON 14](#_Toc135115147)

[2.2.6. Mô hình MVC 15](#_Toc135115148)

[2.2.7. Một số công cụ được sử dụng 17](#_Toc135115149)

[2.2.8. Các khái niệm và số liệu mô tả về giống lúa 18](#_Toc135115150)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 20](#_Toc135115151)

[3.1. TỔNG QUAN VỀ HỆ THỐNG 20](#_Toc135115152)

[3.1.1. Mô tả bài toán 20](#_Toc135115153)

[3.1.2. Sơ đồ tổng quan hệ thống 21](#_Toc135115154)

[3.1.3. Yêu cầu chức năng 22](#_Toc135115155)

[3.1.4. Yêu cầu phi chức năng 24](#_Toc135115156)

[3.1.5. Sơ đồ quy trình nghiệp vụ 25](#_Toc135115157)

[3.1.6. Biểu mẫu nghiên cứu 27](#_Toc135115158)

[3.2. PHÂN TÍCH THIẾT KẾ CHI TIẾT 33](#_Toc135115159)

[3.2.1. Phân tích cấu trúc và yêu cầu thông tin API đối với phương thức POST, PUT/PATCH 33](#_Toc135115160)

[3.2.2. Mô tả các tác nhân 38](#_Toc135115161)

[3.2.3. Mô tả UseCase 38](#_Toc135115162)

[3.2.4. Sơ đồ UseCase 41](#_Toc135115163)

[3.2.5. Đặc tả một vài UseCase đại diện 50](#_Toc135115164)

[3.2.6. Sơ đồ tuần tự 54](#_Toc135115165)

[3.2.7. Sơ đồ lớp (dựa trên API) 57](#_Toc135115166)

[3.3. THIẾT KẾ GIAO DIỆN 58](#_Toc135115167)

[3.3.1. Giao diện đăng nhập 58](#_Toc135115168)

[3.3.2. Giao diện chính 59](#_Toc135115169)

[3.3.3. Giao diện tra cứu 59](#_Toc135115170)

[3.3.4. Giao diện thêm dữ liệu 60](#_Toc135115171)

[3.3.5. Giao diện sửa dữ liệu 60](#_Toc135115172)

[3.3.6. Giao diện xóa 61](#_Toc135115173)

[3.3.7. Giao diện xem chi tiết 61](#_Toc135115174)

[3.4. KẾT QUẢ 62](#_Toc135115175)

[3.5. Ý NGHĨA 62](#_Toc135115176)

[3.6. HƯỚNG PHÁT TRIỂN 62](#_Toc135115177)

[PHỤ LỤC 63](#_Toc135115178)

[1. DANH SÁCH CÁC LIÊN KẾT (URL) API, REQUEST – RESPONSE 63](#_Toc135115179)

[1.1. Người dùng (Users) 63](#_Toc135115180)

[1.2. Đăng nhập và lấy thông tin người dùng đăng nhập 64](#_Toc135115181)

[1.3. Đăng xuất 65](#_Toc135115182)

[1.4. Nhóm giống, giống, kiểu hình 66](#_Toc135115183)

[1.5. Giai đoạn trưởng thành 69](#_Toc135115184)

[1.6. Đối tượng tính trạng 70](#_Toc135115185)

[1.7. Đặc điểm tính trạng 71](#_Toc135115186)

[1.8. Giá trị tính trạng 72](#_Toc135115187)

[1.9. Loại giá trị đo 73](#_Toc135115188)

[1.10. Loại sâu bệnh 73](#_Toc135115189)

[2. CÀI ĐẶT HỆ THỐNG 74](#_Toc135115190)

[TÀI LIỆU THAM KHẢO 80](#_Toc135115191)

**DANH MỤC BẢNG BIỂU**

[Bảng 1. Yêu cầu chức năng hệ thống 22](#_Toc135115192)

[Bảng 2. Cấu trúc và yêu cầu thông tin API Giai đoạn trưởng thành 33](#_Toc135115193)

[Bảng 3. Cấu trúc và yêu cầu thông tin API Đối tượng tính trạng 33](#_Toc135115194)

[Bảng 4. Cấu trúc và yêu cầu thông tin API Đặc điểm tính trạng 34](#_Toc135115195)

[Bảng 5. Cấu trúc và yêu cầu thông tin API Giá trị tính trạng 34](#_Toc135115196)

[Bảng 6. Cấu trúc và yêu cầu thông tin API Chỉ tiêu ngoài đồng 35](#_Toc135115197)

[Bảng 7. Cấu trúc và yêu cầu thông tin API Chỉ tiêu trong nhà 36](#_Toc135115198)

[Bảng 8. Cấu trúc và yêu cầu thông tin API Chỉ tiêu sâu bệnh 36](#_Toc135115199)

[Bảng 9. Cấu trúc và yêu cầu thông tin API Giá trị đo ngoài đồng 37](#_Toc135115200)

[Bảng 10. Cấu trúc và yêu cầu thông tin API Giá trị đo trong nhà 37](#_Toc135115201)

[Bảng 11. Mô tả các tác nhân 38](#_Toc135115202)

[Bảng 12. Mô tả UseCase 38](#_Toc135115203)

[Bảng 13. Đặc tả UseCase Đăng nhập 50](#_Toc135115204)

[Bảng 14. Đặc tả Usecase Đăng xuất 50](#_Toc135115205)

[Bảng 15. Đặc tả UseCase Xem danh sách Giai đoạn trưởng thành 51](#_Toc135115206)

[Bảng 16. Đặc tả UseCase Thêm giai đoạn mới 51](#_Toc135115207)

[Bảng 17. Đặc tả UseCase Sửa thông tin giai đoạn trưởng thành 52](#_Toc135115208)

[Bảng 18. Đặc tả UseCase xóa giai đoạn trưởng thành 53](#_Toc135115209)

[Bảng 20. Request, Response lấy thông tin người dùng 63](#_Toc135115210)

[Bảng 21. Request, Response Đăng nhập 64](#_Toc135115211)

[Bảng 22. Request, Response Lấy thông tin người dùng đăng nhập 65](#_Toc135115212)

[Bảng 23. Request, Response Đăng xuất 65](#_Toc135115213)

[Bảng 24. Request, Response Lấy danh sách nhóm giống 66](#_Toc135115214)

[Bảng 25. Request, Response Lấy danh sách kiểu hình 67](#_Toc135115215)

[Bảng 26. Request, Response Lấy danh sách giống lúa 68](#_Toc135115216)

[Bảng 27. Request, Response Giai đoạn trưởng thành 69](#_Toc135115217)

[Bảng 28. Request, Response Đối tượng tính trạng 70](#_Toc135115218)

[Bảng 29. Request, Response Đặc điểm tính trạng 71](#_Toc135115219)

[Bảng 30. Request, Response Giá trị tính trạng 72](#_Toc135115220)

[Bảng 31. Request, Response Loại giá trị đo 73](#_Toc135115221)

[Bảng 32. Request, Response Loại sâu bệnh 73](#_Toc135115222)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Kiến trúc hệ điều hành Android 6](#_Toc135115223)

[Hình 2. Logo Flutter 7](#_Toc135115224)

[Hình 3. Một số tính năng của ngôn ngữ lập trình Dart 9](#_Toc135115225)

[Hình 4. Biểu đồ quy trình hoạt động của API 13](#_Toc135115226)

[Hình 5. Một đoạn dữ liệu JSON 14](#_Toc135115227)

[Hình 6. Mô hình luồng xử lý trong MVC 15](#_Toc135115228)

[Hình 7. Giao diện chính của Postman 17](#_Toc135115229)

[Hình 8. Logo Visual Code 17](https://d.docs.live.net/9aa00ebfe6dbb690/Desktop/TTCK-KLTN/BaoCao/KLTN/%5bBCKLTN%5d_HuyHoang_DPM195207%20(1).docx#_Toc135115230)

[Hình 9. Sơ đồ tổng quan hệ thống 21](#_Toc135115231)

[Hình 10. Sơ đồ chức năng hệ thống 24](#_Toc135115232)

[Hình 11. Sơ đồ quy trình nghiệp vụ 25](#_Toc135115233)

[Hình 12. Sơ đồ quy trình quản lý các tính trạng giống lúa 25](#_Toc135115234)

[Hình 13. Sơ đồ quy trình quản lý thông tin giống lúa 26](#_Toc135115235)

[Hình 14. Sơ đồ quy trình thu thập số liệu 26](#_Toc135115236)

[Hình 15. Biểu mẫu nhóm giống lúa 27](#_Toc135115237)

[Hình 16. Biểu mẫu kiểu hình giống lúa 28](#_Toc135115238)

[Hình 17. Biểu mẫu nhập liệu giá trị đo chỉ tiêu ngoài đồng 29](#_Toc135115239)

[Hình 18. Biểu mẫu nhập liệu giá trị đo chỉ tiêu trong nhà 30](#_Toc135115240)

[Hình 19. Biểu mẫu nhập liệu giá trị đo chỉ tiêu sâu bệnh 31](#_Toc135115241)

[Hình 20. Biểu mẫu ghi chú quy định tính trạng 32](#_Toc135115242)

[Hình 21. Sơ đồ UseCase Tổng quát 41](#_Toc135115243)

[Hình 22. Sơ đồ phân rã UseCase Quản lý nhóm giống 42](#_Toc135115244)

[Hình 23. Sơ đồ phân rã UseCase Quản lý kiểu hình 42](#_Toc135115245)

[Hình 24. Sơ đồ phân rã UseCase Quản lý Giống lúa 43](#_Toc135115246)

[Hình 25. Sơ đồ phân rã UseCase Quản lý mã ngoài đồng 43](#_Toc135115247)

[Hình 26. Sơ đồ phân rã UseCase Quản lý mã phòng thí nghiệm 44](#_Toc135115248)

[Hình 27. Sơ đồ phân rã UseCase Quản lý giai đoạn trưởng thành 44](#_Toc135115249)

[Hình 28. Sơ đồ phân rã UseCase Quản lý tính trạng 45](#_Toc135115250)

[Hình 29. Sơ đồ phân rã UseCase Loại giá trị đo 46](#_Toc135115251)

[Hình 30. Sơ đồ phân rã UseCase Loại sâu bệnh 46](#_Toc135115252)

[Hình 31. Sơ đồ phân rã UseCase Quản lý chỉ tiêu ngoài đồng 47](#_Toc135115253)

[Hình 32. Sơ đồ phân rã UseCase Quản lý chỉ tiêu trong nhà 48](#_Toc135115254)

[Hình 33. Sơ đồ phân rã UseCase Quản lý chỉ tiêu sâu bệnh 49](#_Toc135115255)

[Hình 34. Sơ đồ tuần tự UseCase Đăng nhập 54](#_Toc135115256)

[Hình 35. Sơ đồ tuần tự UseCase Đăng xuất 54](#_Toc135115257)

[Hình 36. Sơ đồ tuần tự cho UseCase Thêm giai đoạn trưởng thành 55](#_Toc135115258)

[Hình 37. Sơ đồ tuần tự UseCase Sửa dữ liệu 55](#_Toc135115259)

[Hình 38. Sơ đồ tuần tự UseCase Xóa dữ liệu 56](#_Toc135115260)

[Hình 39. Sơ đồ lớp quản lý giống lúa 57](#_Toc135115261)

[Hình 40. Giao diện đăng nhập 58](#_Toc135115262)

[Hình 42. Giao diện đăng nhập fail 58](#_Toc135115263)

[Hình 43. Giao diện đăng nhập success 58](#_Toc135115264)

[Hình 44. Giao diện chính 59](#_Toc135115265)

[Hình 45. Giao diện tra cứu 59](#_Toc135115266)

[Hình 46. Giao diện thêm dữ liệu 60](#_Toc135115267)

[Hình 47. Giao diện sửa dữ liệu 60](#_Toc135115268)

[Hình 48. Giao diện xóa dữ liệu 61](#_Toc135115269)

[Hình 49. Giao diện xem chi tiết 61](#_Toc135115270)

[Hình 50. Tải file SDK Flutter 74](#_Toc135115271)

**CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên viết tắt** | **Tên đầy đủ** |
| 1 | API | Application Programming Interface |
| 2 | CSDL | Cơ sở dữ liệu |
| 3 | DVM | Davik Vitual Machine |
| 4 | IDE | Integrated Development Environment |
| 5 | HTTP | HyperText Transfer Protocol |
| 6 | JSON | JavaScript Object Notation |
| 7 | MVC | Model – View – Controller |
| 8 | OOP | Object Oriented Programming |
| 9 | SDK | Sofware Development Kit |
| 10 | SQL | Structure Query Language |
| 11 | UI | User Interface |
| 12 | URL | Uniform Resource Locator |

# CHƯƠNG 1: ĐẶT VẤN ĐỀ

## 1.1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Hiện nay, công nghệ di động đang phát triển một các vượt bậc, mọi người đều sở hữu cho bản thân những thiết bị di động thông minh. Các ứng dụng được xây dựng cho thiết bị di động đã trở thành một phần thiết yếu phục vụ cho các nhu cầu trong cuộc sống hàng ngày như mua sắm, trao đổi thông tin, giải trí và giải quyết các nhu cầu nghiệp vụ của tất cả các ngành công nghiệp trên thế giới. Sự hiện diện của các thiết bị di động hầu hết các lĩnh vực, sự hữu ích và cần thiết này đã thúc đẩy hoạt động kinh doanh, nghiên cứu ngày càng phát triển mạnh mẽ.

Thực tế hơn, trong quá trình nghiên cứu, quản lý và cải thiện giống lúa, thông tin giống lúa trong nghiên cứu khoa học chỉ được nhập trên văn bản giấy. Điều  
này gây ra nhiều khó khăn và rủi ro trong quá trình nhập liệu và sử dụng  
chứng từ giấy. Nó làm giảm hiệu quả, hiệu suất lao động của các công trình  
nghiên cứu khoa học về giống lúa.

Thông tin về giống lúa ngày càng nhiều và phức tạp, khi thông tin trên văn bản giấy thường khó thay đổi thông tin theo ý muốn. Vì vậy, việc sử dụng phần mềm, ứng dụng di động hỗ trợ quản lý và thu thập thông tin giống lúa giúp các cá nhân, tổ chức nghiên cứu giống lúa hiệu quả hơn. Do đó, việc quản lý thông tin giống lúa bằng ứng dụng di động là rất cần thiết.

## 1.2. LÝ DO CHỌN ĐỀ TÀI

Điện thoại thông minh hiện đang là thiết bị di động được sử dụng phổ biến hiện nay bởi sự tiện lợi và ứng dụng hiệu quả trong mọi vấn đề của cuộc sống. Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất.

Cùng với sự phát triển ngày càng hiện đại, điện thoại thông minh được dùng vào nhiều mục đích như giải trí, công việc, học tập và hỗ trợ con người trong xử lý các tác vụ công việc một cách nhanh chóng. Với những khả năng tiện ích trên, việc sử dụng thiết bị di động trong nhập liệu phục vụ cho quá trình thu thập, quản lý và nghiên cứu rất cần thiết.

## 1.3. MỤC TIÊU NGHIÊN CỨU CỦA ĐỀ TÀI

Mục tiêu nghiên cứu của đề tài là Xây dựng hệ thống “Quản lý hình ảnh lúa giống” trên ứng dụng Android sử dụng framework Flutter, ứng dụng có khả năng giao tiếp với API để quản lý dữ liệu mô tả và nhập số liệu thực tế về đặc điểm của giống lúa được khảo sát trên đồng ruộng và trong phòng thí nghiệm nhằm hỗ trợ các nhà nghiên cứu, nông dân có thể thuận tiện hơn trong việc thu thập và quản lý mẫu giống lúa,phục vụ cho mục đích nghiên cứu khoa học trong lĩnh vực Nông nghiệp, giúp cải thiện, nâng cáo chất lượng lúa giống nước nhà.

## 1.4. ĐẶT VẤN ĐỀ

Trong công tác quản lý, các nhà nghiên cứu và quản lý lúa gạo phải quản  
lý nhiều thông tin mô tả các loại giống lúa khác nhau. Các thông tin này được  
sử dụng để nghiên cứu, phân tích đặc tính của giống lúa, từ đó giúp các nhà  
nghiên cứu, quản lý phát triển các giống lúa mới có chất lượng tốt hơn, đáp  
ứng nhu cầu sản xuất nông nghiệp.

Thực tế hơn, trong quá trình nghiên cứu, quản lý và cải thiện giống lúa, thông tin giống lúa trong nghiên cứu khoa học chỉ được nhập trên văn bản giấy. Điều  
này gây ra nhiều khó khăn và rủi ro trong quá trình nhập liệu và sử dụng  
chứng từ giấy. Nó làm giảm hiệu quả, hiệu suất lao động của các công trình  
nghiên cứu khoa học về giống lúa.

Thông tin về giống lúa ngày càng nhiều và phức tạp, khi thông tin trên văn bản giấy thường khó thay đổi thông tin theo ý muốn. Vì vậy, việc sử dụng phần mềm, ứng dụng di động hỗ trợ quản lý và thu thập thông tin giống lúa giúp các cá nhân, tổ chức nghiên cứu giống lúa hiệu quả hơn. Do đó, việc quản lý thông tin giống lúa bằng ứng dụng di động là rất cần thiết.

## 1.5. PHẠM VI CỦA ĐỀ TÀI

Ứng dụng được sử dụng cho đối tượng chính là các nhà nghiên cứu, sinh viên đang thực hiện nghiên cứu về giống lúa tại khu vực An Giang.

Các chức năng của ứng dụng được tập trung cụ thể gồm: tra cứu các thông tin về giống lúa, nhập dữ liệu , thông số về giống lúa khi thực hiện khảo sát, nghiên cứu.

## 1.6. PHƯƠNG PHÁP NGHIÊN CỨU

Phương pháp nghiên cứu thực tiễn (khảo sát thực tế, thu thập yêu cầu, dữ liệu, xác định các yêu cầu về nghiệp vụ, yêu cầu hệ thống ….): trực tiếp đi đến nơi trồng và nghiên cứu giống lúa để quan sát quá trình thu thập số liệu về thông tin giống lúa cần được hợp thức hóa vào hệ thống để quản lý.

Phương pháp nghiên cứu tài liệu (tham khảo các tài liệu từ website, các kênh youtube,…): tìm hiểu Cơ sở lý thuyết về hệ thống Rest API, các tài liệu về giống lúa.

Phương xử lý số liệu (thu thập thông tin từ việc phỏng vấn, thống kê, phân tích thông tin cơ sở dữ liệu): thu thập các số liệu về mô tả giống lúa, phân tích số liệu đó và đưa ra các giải pháp cho việc xây dựng phần mềm quản lý mô tả giống lúa.

# CHƯƠNG 2: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

## 2.1. TỔNG QUAN

Trong công tác quản lý, các nhà nghiên cứu và quản lý lúa gạo phải quản lý nhiều thông tin mô tả các loại giống lúa khác nhau. Các thông tin này được sử dụng để nghiên cứu, phân tích tính trạng hoặc kiểu hình của giống lúa, từ đó giúp các nhà nghiên cứu, quản lý phát triển các giống lúa mới có chất lượng tốt hơn, đáp ứng nhu cầu sản xuất nông nghiệp.

Một số nghiên cứu về đánh giá tiêu chuẩn giống lúa và lưu trữ thông tin:

- Nghiên cứu và đưa ra hệ thống tiêu chuẩn đánh giá nguồn gen lúa (SES – Standard Evaluation System for Rice 1996 và 2013 được nghiên cứu bởi IRRI - The International Rice Research Institute – Viện nghiên cứu lúa Quốc tế, Philippines), nghiên cứu xây dựng hệ thống tiêu chuẩn đánh giá nguồn gen giống lúa dựa trên đánh giá thực tiễn các đặc tính di truyền, đặc điểm tính trạng, hình thái, các yếu tố môi trường canh tác và các yếu tố sâu bệnh ảnh hưởng, đồng thời đánh giá chất lượng gạo của cây lúa. Từ đó xác định được nguồn gen triển vọng của cây lúa cùng với những đặc tính có lợi góp phần quan trọng trong quá trình cải tiến giống lúa. Hiện nay Ngân hàng gen lúa quốc tế tại IRRI (Philippines) hiện lưu trữ hơn 127.000 giống lúa từ khắp nơi trên thế giới. Những giống lúa này mang nguồn gen/tính trạng quý chưa được khai thác để tạo ra các giống lúa canh tác theo hướng bền vững trong một môi trường ngày càng bất lợi do các thay đổi khí hậu toàn cầu. Các tính trạng được tập trung khai thác cho việc cải tiến giống bao gồm chọn giống lúa giàu dinh dưỡng, chống chịu sâu bệnh hại và môi trường bất lợi như lũ lụt, khô hạn, giảm phát thải khí nhà kính.

- Năm 1977, Viện Lúa ÐBSCL đã xây dựng Ngân hàng Gen lúa đặt tại Viện và duy trì, bổ sung, trẻ hóa các giống lúa định kỳ nhằm đáp ứng yêu cầu bảo tồn giống. Ngân hàng Gen hiện lưu giữ, bảo tồn trên 4.000 mẫu giống lúa mùa địa phương, lúa hoang và lúa cải tiến để phục vụ công tác nghiên cứu tại Viện cũng như chia sẻ nguồn gen với các cơ quan nghiên cứu khoa học về cây lúa cả trong nước và quốc tế.

- Theo bài báo cáo nghiên cứu “Xây dựng Cơ sở dữ liệu và hệ thống tra cứu các giống lúa phổ biến ở Việt Nam” năm 2015 của nhóm sinh viên khoa Công nghệ Thông tin, Học viên Nông nghiệp Việt Nam. Nghiên cứu này xây dựng một mô hình thực nghiệm để thu thập, lưu trữ và hiển thị dữ liệu thông tin về giống lúa. Kết quả nghiên cứu, nhóm xây dựng được hệ thống tra cứu trực tuyến cơ sở dữ liệu về một số giống lúa phổ biến ở Việt Nam được xây dựng bằng ngôn ngữ lập trình PHP, cơ sở dữ liệu được quản trị thông qua hệ quản trị CSDL MySQL.

- Bài báo nghiên cứu “Xây dựng Cơ sở dữ liệu và Module tra cứu hình ảnh nảy mầm của một số giống lúa phổ biến ở Việt Nam” năm 2015 của nhóm sinh viên khoa Công nghệ Thông tin, Học viện Nông nghiệp Việt Nam. Nghiên cứu đã xây dựng được CSDL hình ảnh nảy mầm của giống lúa và được thực nghiệm lưu trữ va đánh giá được hiệu năng và tốc độ lưu trữ dữ liệu hình ảnh trên nhiều hệ quản trị CSDL như: Oracle, SQL Serve, Microsoft Access, MySQL,…. Để xây dựng module trên nền tảng web phục vụ cho quá trình tra cứu thông tin dữ liệu và hình ảnh nảy mầm của các giống lúa đem thử nghiệm nảy mầm.

## 2.2. CƠ SỞ LÝ THUYẾT

### **2.2.1. Hệ điều hành Android**

#### **2.2.1.1. Giới thiệu chung**

Android là hệ điều hành có mã nguồn mở dựa trên nền tảng Linux Kernel do chính Google phát hành được thiết kế dành cho các nhà phát triển thiết bị, các nhà mạng, lập trình viên có thể tiếp cận nhanh chóng, điều chỉnh tự do trong mã nguồn mở đó.

Hệ điều hành Android đang chiếm hơn 80% thị trường thiết bị di động hiện đại ngày nay. Nhu cầu sử dụng ứng dụng di động cũng ngày càng cao. Nhu cầu việc làm về lập trình trên hệ điều hành mobile phổ biến nhất thế giới này cũng vì thế mà tăng mạnh do tính mở và dễ tiếp cận của nó.

Lịch sử Android bắt đầu từ khoảng tháng 10 năm 2003, nghĩa là rất lâu trước cả thuật ngữ smartphone. Công ty Android Inc được thành lập tại Palo Alto, California bởi 4 người sáng lập là Rich Miner, Nick Sears, Chris White và Andy Rubin. Hiện Android là một thương hiệu của Google. Có khả năng tùy biến rất cao và có thể chạy trên nhiều thiết bị, nhiều kiến trúc vi xử lý (ARM / x86).

Android là hệ điều hành đã làm thay đổi khả năng của thiết bị di động, giúp tất cả mọi người đều được trải nghiệm smartphone và các thiết bị thông minh khác như máy tính bảng, đồng hồ, TV hay cả ô tô. Trải qua rất nhiều năm, Android đã đi từng bước vững chắc để trở thành hệ điều hành dành cho tất cả, liên tục cập nhật các tính năng và mang đến một hệ sinh thái thân thiện, an toàn, thông minh cho người dùng.

#### **2.2.1.2. Kiến trúc của Hệ điều hành Android**

Hệ điều hành Android là 1 ngăn xếp các thành phần phần mềm, được chia thành 5 phần và 4 lớp chính như sau:

**Linux Kernel**

Đây là lớp dưới cùng – Linux 3.6 với khoảng 115 bản vá. Lớp này cung cấp 1 cấp độ trừu tượng giữa phần cứng của thiết bị và các thành trình điều khiển phần cứng thiết yếu như máy ảnh, bàn phím, màn hình hiển thị... Đồng thời, hạt nhân (kernel) còn xử lý tất cả các thứ mà Linux có thể làm tốt như mạng kết nối và một chuỗi các trình điều khiển thiết bị, giúp cho giao tiếp với các thiết bị ngoại vi dễ dàng hơn.

**Các thư viện (Libraries)**

Ở trên lớp nhân Linux là tập các thư viện bao gồm WebKit - trình duyệt Web mã nguồn mở, được biết đến như thư viện libc, cơ sở dữ liệu SQLite - hữu dụng cho việc lưu trữ và chia sẻ dữ liệu ứng dụng, các thư viênj chơi và ghi âm audio, video, hay các thư viện SSL chiụ trách nhiệm bảo mật Internet...

Đây là các thư viện dựa trên Java phục vụ cho việc phát triển Android. Ví dụ của các thư viện này bao gồm các thư viện ứng dụng dùng để xây dựng giao diện người dùng, vẽ đồ họa hay truy cập cơ sở dữ liệu.

**Android Runtime**

Đây là phần thứ 3 của kiến trúc và nằm ở lớp thứ 2 từ dưới lên. Phần này cung cấp một bộ phận quan trọng là Dalvik Vỉtual Machine - là 1 loại Java Virtual Machine được thiết kế đặc biệt để tối ưu cho Android.

Dalvik VM sử dụng các đặc trưng của nhân Linux như quản lý bộ nhớ và đa luồng, những thứ mà đã có sẵn trong Java. Dalvik VM giúp mọ ứng dụng Android chạy trong tiến trình riêng của nó, với các thể hiện (instance) riêng của Dalvik virtual Machine.

Android Runtime cũng cung cấp 1 tập các thư viện chính giúp các nhà phát triển ứng dụng Android có thể viết ứng dụng Android bằng Java.

**Application Framework**

Lớp Android Framework cung cấp các dịch vụ cấp độ cao hơn cho các ứng dụng dưới dạng các lớp Java. Các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong ứng dụng của họ.

Android Framework bao gồm các dịch vụ chính sau:

*Activitty Manager* - Kiểm soát tất cả khía cạnh của vòng đời ứng dụng và ngăn xếp các Activity.

Content Providers - Cho phép các ứng dụng chia sẽ dữ liệu với các ứng dụng khác.

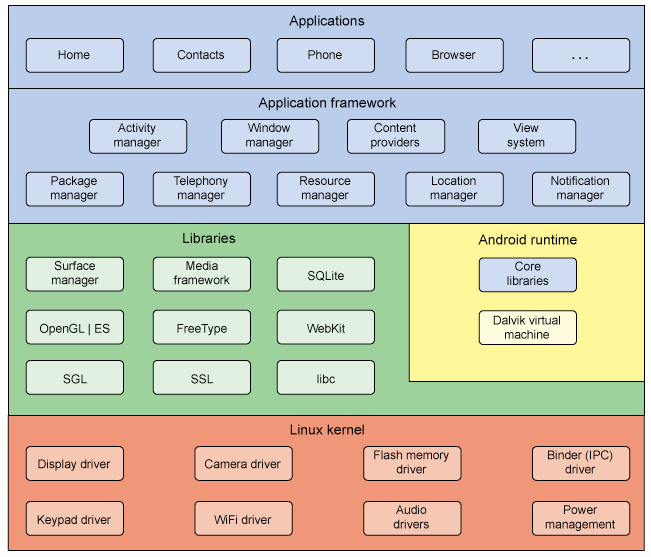
Resource Manager - Cung cấp quyền truy cập vào các tài nguyên như các chuỗi, màu sắc, các layout giao diện người dùng...

Notifications Manager - Cho phép các ứng dụng hiển thị cảnh báo và các thông báo cho người dùng.

View System - Tập các thành phần giao diện (view) được sử dụng để tạo giao diện người dùng.

**Application**

Lớp trên cùng của kiến trúc là Application. Các ứng dụng bạn tạo ra sẽ được cài đặt trên lớp này.



Hình 1. Kiến trúc hệ điều hành Android

### **2.2.2. Tổng quan về Flutter**

#### **2.2.2.1. Giới thiệu**

Flutter là một bộ công cụ phát triển phần mềm giao diện người dùng với mã nguồn mở do [Google](https://vietnix.vn/google-la-gi/) tạo ra. Flutter được sử dụng để phát triển các ứng dụng đa nền tảng cho nhiều [hệ điều hành](https://vietnix.vn/he-dieu-hanh-la-gi/) như: iOS, Android, [Linux](https://vietnix.vn/linux-la-gi/),… từ một cơ sở mã nguồn duy nhất.

Flutter được Google công bố vào đầu năm 2015 nhưng đến tháng 5 năm 2017 nền tảng này mới chính thức ra mắt thị trường. Với mục đích xây dựng và phát triển các ứng dụng chất lượng, mang đến cho người dùng những trải nghiệm tuyệt vời nhất trên hệ điều hành iOS và Android.

Shape

Description automatically generated with medium confidence

Hình 2. Logo Flutter

Flutter có hai thành phần chính đó là:

*Software Development Kit* **(**[SDK](https://vietnix.vn/sdk-la-gi/)**):** Ở đây tích hợp các công cụ có khả năng phát triển các ứng dụng của bạn, trong đó có các công cụ giúp bạn biên dịch mã của mình thành mã máy gốc (mã cho Android và iOS).

*UI Library based on widgets* **(**[Framework](https://vietnix.vn/framework-la-gi/)**):** Tổng hợp các thành phần giao diện người dùng (UI) có thể tái sử dụng như: text input, button, slider,… hỗ trợ bạn có thể cá nhân hóa theo nhu cầu sử dụng.

Flutter sử dụng Dart, ngôn ngữ này được tạo ra bởi Google vào tháng 10 năm 2011 và được cải tiến rất nhiều trong những năm qua. Dart tập trung vào phát triển giao diện người dùng và bạn có thể sử dụng nó để tạo các ứng dụng web và di động. Dart là một ngôn ngữ nhanh, hướng đối tượng với nhiều tính năng hữu ích như mixin, generic, isolate, và static type.

#### **2.2.2.2. Đặc điểm**

*Fast Development*: Tính năng Hot Reload hoạt động trong milliseconds để hiện thị giao diện tới bạn. Sử dụng tập hợp các widget có thể customizable để xây dựng giao diện trong vài phút. Ngoài ra Hot Reload còn giúp bạn thêm các tính năng, fix bug tiết kiệm thời gian hơn mà không cần phải thông qua máy ảo, máy android hoặc iOS. Mang lại hiệu xuất tối đa cho các lập trình viên.

Có rất nhiều các thành phần để xây dựng giao diện của Flutter vô cùng đẹp mắt theo phong cách Material Design và Cupertino, hỗ trợ nhiều các APIs chuyển động, smooth scrolling... Các widget của fluter kết hợp các sự khác biệt của các nền tảng ví dụ như scrolling, navigation, icons, font để cung cấp một hiệu năng tốt nhất tới iOS và Android.

Flutter có một cộng đồng mạnh mẽ và điều đó là một tín hiệu tốt cho những lập trình viên đang tìm hiểu về Flutter. Có rất nhiều người yêu thích việc chia sẻ kiến ​​thức và các nội dung hữu ích về lập trình trên trang web của họ hoặc các cộng đồng khác và Flutter có một cộng đồng rất lớn để bạn có thể trao đổi thông tin và chia sẻ cùng nhau.

### **2.2.3. Ngôn ngữ lập trình Dart**

#### **2.2.3.1. Giới thiệu chung**

Dart là ngôn ngữ lập trình hướng đối tượng (OOP) thuần túy, dựa trên lớp (class) với mọi giá trị là một đối tượng - một ngôn ngữ đa nền tảng mà chịu ảnh hưởng khá nhiều về mặt thiết kế, được dùng trong lập trình Flutter – bộ công cụ giao diện người dùng của Google để xây dựng các ứng dụng Mobile, Web và Desktop app đẹp, được biên dịch nguyên bản từ một cơ sở mã code duy nhất.

Dart được phát hành bởi google vào ngày 14 tháng 11 năm 2013 và được viết bởi Lars Bak và Kasper Lund. Ngôn ngữ này hướng đến việc giúp cho các developer xây dựng được các ứng dụng điện thoại và các web hiện đại. Nó có thể viết được cho phía client, server và hiện tại là mobile với Flutter.

Một số tính năng của Dart:

Diagram

Description automatically generated

Hình 3. Một số tính năng của ngôn ngữ lập trình Dart

**Mã nguồn mở** (Open source)

Dart là một ngôn ngữ lập trình mã nguồn mở, có nghĩa là nó có sẵn miễn phí. Nó được phát triển bởi Google, được phê duyệt bởi tiêu chuẩn ECMA và đi kèm với giấy phép BSD.

**Nền tảng độc lập** (Platform Independent)

Dart hỗ trợ tất cả các hệ điều hành chính như Windows, Linux, Macintosh, v.v. Dart có Máy ảo riêng được gọi là Dart VM, cho phép chúng tôi chạy mã Dart trong mọi hệ điều hành.

**Hướng đối tượng** (Object Oriented Programming – OOP)

Dart là một ngôn ngữ lập trình hướng đối tượng và hỗ trợ tất cả các khái niệm oops như lớp, kế thừa, giao diện và các tính năng gõ tùy chọn. Nó cũng hỗ trợ các khái niệm nâng cao như mixin, abstract, các lớp, hệ thống kiểu chung được sửa đổi và mạnh mẽ.

**Đồng nhất** (Concurrency)

Dart là một ngôn ngữ lập trình không đồng bộ, có nghĩa là nó hỗ trợ đa luồng sử dụng Isolates. Các vùng cách ly là các thực thể độc lập có liên quan đến các luồng nhưng không chia sẻ bộ nhớ và thiết lập giao tiếp giữa các quá trình bằng cách truyền thông điệp. Thông điệp nên được nối tiếp nhau để tạo hiệu quả truyền thông. Việc tuần tự hóa thông báo được thực hiện bằng cách sử dụng một ảnh chụp nhanh được tạo ra bởi đối tượng đã cho và sau đó truyền đến một vùng cách ly khác để giải mã.

**Thư viện mở rộng** (Extensive libraries)

Dart bao gồm nhiều thư viện tích hợp hữu ích bao gồm SDK (Bộ phát triển phần mềm), lõi, toán học, không đồng bộ, toán học, chuyển đổi, html, IO, v.v. Nó cũng cung cấp cơ sở để tổ chức mã Dart thành các thư viện với không gian tên riêng. Nó có thể sử dụng lại bằng câu lệnh nhập.

**Dễ học** (Easy to learn)

Như chúng ta đã thảo luận trong phần trước, học Dart không phải là nhiệm vụ của Hercules vì ​​chúng ta biết rằng cú pháp của Dart tương tự như Java, C #, JavaScript, kotlin, v.v. nếu bạn biết bất kỳ ngôn ngữ nào trong số này thì bạn có thể học Dart dễ dàng.

**Biên dịch linh hoạt** (Flexible Compilation)

Dart cung cấp sự linh hoạt để biên dịch mã và nhanh chóng. Nó hỗ trợ hai loại quy trình biên dịch, AOT (Ahead of Time) và JIT (Just-in-Time). Mã Dart được truyền bằng ngôn ngữ khác có thể chạy trong các nhà sản xuất web hiện đại.

**Kiểu An toàn** (Type safe)

Dart là ngôn ngữ an toàn kiểu, có nghĩa là nó sử dụng cả kiểm tra kiểu tĩnh và kiểm tra thời gian chạy để xác nhận rằng giá trị của một biến luôn khớp với kiểu tĩnh của biến, đôi khi nó được gọi là kiểu gõ âm thanh.

Mặc dù loại là bắt buộc, nhưng chú thích loại là tùy chọn vì loại nhiễu. Điều này làm cho mã dễ đọc hơn. Ưu điểm khác của ngôn ngữ an toàn kiểu chữ là khi chúng ta thay đổi phần mã, hệ thống sẽ cảnh báo chúng ta về sửa đổi mà chúng ta đã sửa trước đó.

**Các đối tượng** (Objects)

Dart coi mọi thứ như một đồ vật. Giá trị gán cho biến là một đối tượng. Các hàm, số và chuỗi cũng là một đối tượng trong Dart. Tất cả các đối tượng kế thừa từ lớp Đối tượng.

**Hỗ trợ trình duyệt** (Browser support)

Dart hỗ trợ tất cả các trình duyệt web hiện đại. Nó đi kèm với trình biên dịch dart2js để chuyển đổi mã Dart thành mã JavaScript được tối ưu hóa phù hợp với tất cả các loại trình duyệt web.

**Cộng đồng** (Community)

Dart có một cộng đồng lớn trên toàn thế giới. Vì vậy, nếu bạn gặp vấn đề trong khi viết mã thì rất dễ dàng tìm được trợ giúp. Nhóm các nhà phát triển chuyên dụng đang làm việc để nâng cao chức năng của nó.

#### **2.2.3.2. Đặc điểm**

Ngôn ngữ lập trình Dart có những đặc điểm ưu việt sau đây:

- Là một ngôn ngữ độc lập với nền tảng và hỗ trợ tất cả các hệ điều hành như Windows, Mac, Linux, v.v.

- Là một ngôn ngữ mã nguồn mở, có nghĩa là nó có sẵn miễn phí cho tất cả mọi người. Nó đi kèm với giấy phép BSD và được công nhận bởi tiêu chuẩn ECMA.

- Là một ngôn ngữ lập trình hướng đối tượng và hỗ trợ tất cả các tính năng của oops như kế thừa, giao diện và các tính năng kiểu tùy chọn.

- Dart đi kèm với trình biên dịch dar2js để truyền mã Dart thành mã JavaScript chạy trên tất cả các trình duyệt web hiện đại. Máy ảo Dart độc lập cho phép mã Dart chạy trong môi trường giao diện dòng lệnh, điều này rất hữu ích trong việc xây dựng các ứng dụng thời gian thực vì tính ổn định của nó.

Một tính năng đáng chú ý của nền tảng Dart là hỗ trợ "tải lại nóng" (hot reload) trong đó các sửa đổi trong tập tin nguồn có thể được chèn vào ứng dụng đang chạy. Flutter mở rộng sự hỗ trợ này cho tính năng "tải lại nóng giữ trạng thái (stateful hot reload), để các sửa đổi trong mã nguồn có thể được cập nhật ngay lập tức lên ứng dụng đang chạy mà không cần phải khởi động lại hoặc mất mát các trạng thái đang có.

### **2.2.4. Tổng quan về API**

#### **2.2.4.1. API**

API (Application Programming Interface - giao diện lập trình ứng dụng) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng.

Chẳng hạn, một chương trình máy tính có thể dùng các hàm API của hệ điều hành để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng thực hiện API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.

API được xây dựng trên chính 2 thành phần: Request và Response

**Về Request**

Một cái request đúng chuẩn cần có 4 thứ:

*URL* là địa chỉ duy nhất cho 1 request, thường là đường dẫn tới một hàm xử lí logic.

*Method*: HTTP request có tất cả 9 loại method, 4 loại được sử dụng phổ biến nhất là GET, POST, PUT, DELETE

*Headers* là nơi chứa các thông tin cần thiết của 1 request nhưng end-users không biết có sự tồn tại của nó. Ví dụ: độ dài của request body, thời gian gửi request, loại thiết bị đang sử dụng, loại định dạng cái response mà client có đọc được…

*Body*là nơi chứa thông tin mà client sẽ điền.

**Về Response**

Sau khi nhận được request từ phía client, server sẽ xử lý cái request đó và gửi ngược lại cho client 1 cái response. Cấu trúc của 1 response tương đối giống phần request nhưng Status code sẽ thay thế cho URL và Method. Tóm lại, nó có cầu trúc 3 phần: Status code, Headers, Body. Phần Header và body tương đối giống với request.

Status code (Mã hóa trạng thái thường được gọi là mã trạng thái) là một số nguyên 3 ký tự, trong đó ký tự đầu tiên của Status-Code định nghĩa loại Response và hai ký tự cuối không có bất cứ vai trò phân loại nào. Có 5 giá trị của ký tự đầu tiên:

1xx: Information (Thông tin): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận và quá trình xử lý request đang được tiếp tục.

2xx: Success (Thành công): Khi nhận được những mã như vậy tức là request đã được server tiếp nhận, hiểu và xử lý thành công

3xx: Redirection (Chuyển hướng): Mã trạng thái này cho biết client cần có thêm action để hoàn thành request

4xx: Client Error (Lỗi Client): Nó nghĩa là request chứa cú pháp không chính xác hoặc không được thực hiện.

5xx: Server Error (Lỗi Server): Nó nghĩa là Server thất bại với việc thực hiện một request nhìn như có vẻ khả thi.

Diagram

Description automatically generated

Hình 4. Biểu đồ quy trình hoạt động của API

#### **2.2.4.2. REST API**

REST (REpresentational State Tranfer - Chuyển trạng thái đại diện). REST xác định một tập hợp các hàm như GET, PUT, DELETE, v.v. mà máy khách có thể dùng để truy cập vào dữ liệu của máy chủ. Máy khách và máy chủ trao đổi dữ liệu qua giao thức HTTP.

REST API (còn được biết với tên gọi RESTful API) là một giao diện lập trình ứng dụng (API) mà tuân thủ các ràng buộc và quy ước kiến trúc REST được sử dụng trong việc giao tiếp giữa client và server.

API REST mang lại 4 lợi ích chính:

- **Tích hợp**: API được sử dụng để tích hợp ứng dụng mới với hệ thống phần mềm hiện tại. Điều này làm tăng tốc độ phát triển vì không cần phải viết lại từng chức năng từ đầu. Bạn có thể sử dụng API để tận dụng mã hiện có.

- **Đổi mới**: rất nhiều lĩnh vực có thể thay đổi khi một ứng dụng mới ra mắt. Doanh nghiệp cần khẩn trương phản ứng và hỗ trợ việc triển khai nhanh chóng các dịch vụ đổi mới. Họ có thể thực hiện việc này bằng cách thực hiện các thay đổi ở cấp độ API mà không cần phải viết lại toàn bộ mã.

- **Mở rộng**: API mang lại cơ hội độc đáo cho các doanh nghiệp để đáp ứng nhu cầu khách hàng của họ trên những nền tảng khác nhau. Ví dụ: API bản đồ cho phép tích hợp thông tin bản đồ qua các trang web, nền tảng Android, iOS, v.v. Mọi doanh nghiệp đều có thể cung cấp quyền truy cập tương tự vào cơ sở dữ liệu nội bộ của họ bằng API miễn phí hoặc trả phí.

- **Dễ duy trì**: API đóng vai trò là cổng giữa hai hệ thống. Mỗi hệ thống đều phải thực hiện các thay đổi nội bộ để API không bị tác động. Bằng cách này, mọi sự thay đổi về mã trong tương lai do một bên thực hiện sẽ không tác động đến bên còn lại.

### **2.2.5. JSON**

JavaScript Object Notation (thường được viết tắt là JSON) là một kiểu dữ liệu mở trong JavaScript. JSON là một kiểu dữ liệu trung gian bao gồm chủ yếu là text, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình. Cú pháp của JSON có 2 phần đó là key và value:

- Chuỗi JSON được bao lại bởi dấu ngoặc nhọn {}

- Các key, valuecủa JSON bắt buộc phải đặt trong dấu nháy kép {“}, nếu bạn đặt nó trong dấu nháy đơn thì đây không phải là một chuỗi JSON đúng chuẩn.

- Dấu phẩy “,” để ngăn cách giữa các cặp dữ liệu.

- Các key của JSON bạn nên đặt chữ cái không dấu hoặc số, dấu \_ và không có khoảng trắng., ký tự đầu tiên không nên đặt là số.

- File json có thể được lưu với bất kỳ phần mở rộng nào, tuy nhiên thông thường thì nó được lưu dưới phần mở rộng là .json hoặc .js.

Text

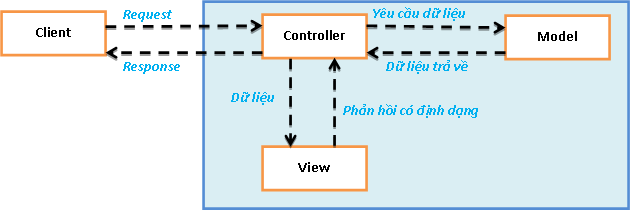
Description automatically generated with medium confidence

Hình 5. Một đoạn dữ liệu JSON

### **2.2.6. Mô hình MVC**

#### **2.2.6.1. Khái niệm**

Mô hình Model-View-Controller (MVC) là một mẫu kiến ​​trúc phân tách một ứng dụng thành ba thành phần logic chính Model, View và Controller. Mỗi thành phần kiến ​​trúc được xây dựng để xử lý khía cạnh phát triển cụ thể của một ứng dụng. MVC tách lớp logic nghiệp vụ và lớp hiển thị ra riêng biệt. Ngày nay, kiến ​​trúc MVC đã trở nên phổ biến để thiết kế các ứng dụng web cũng như ứng dụng di động.



Hình 6. Mô hình luồng xử lý trong MVC

Ba thành phần logic của mô hình MVC:

**Model**: thành phần model lưu trữ dữ liệu và logic liên quan của nó. Bao gồm các class function xử lý các tác vụ như truy vấn, thêm, sửa hoặc xóa dữ liệu.

**View** là một phần của ứng dụng đại diện cho việc trình bày dữ liệu được tạo bởi các dữ liệu mà chúng ta lấy từ dữ liệu trong model. Một view yêu cầu model cung cấp đầy đủ dữ liệu để nó hiển thị đầu ra cho người dùng và đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.

**Controller** là một phần của ứng dụng xử lý tương tác của người dùng. Bộ điều khiển diễn giải đầu vào chuột và bàn phím từ người dùng, thông báo cho model và view để thay đổi khi thích hợp. Là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View. Controller gửi các lệnh đến model để làm thay đổi trạng thái của nó.

#### **2.2.6.2. Ưu điểm, nhược điểm của mô hình MVC**

**Ưu điểm**:

- Bảo trì mã dễ dàng, giúp mở rộng và phát triển

- Thành phần MVC có thể được kiểm tra riêng với người dùng

- Hỗ trợ dễ dàng cho khách hàng mới

- Sự phát triển của các thành phần khác nhau có thể được thực hiện song song.

- Nó giúp bạn tránh sự phức tạp bằng cách chia một ứng dụng thành ba phần: Model, view và controller.

- Cung cấp hỗ trợ tốt nhất để phát triển dựa trên thử nghiệm

- Nó hoạt động tốt cho các ứng dụng Web được hỗ trợ bởi các nhóm các nhà thiết kế và phát triển web.

- Cung cấp sự phân tách rõ ràng các mối quan tâm (SoC).

- Công cụ tìm kiếm tối ưu hóa (SEO) thân thiện.

- Tất cả các layer (lớp) và các đối tượng là độc lập với nhau để bạn có thể kiểm tra chúng một cách riêng biệt.

**Nhược điểm**:

- Khó đọc, thay đổi, kiểm tra và sử dụng lại mô hình này.

- Việc điều hướng tổ chức có thể đôi khi phức tạp vì nó giới thiệu các layer mới đòi hỏi người dùng phải thích ứng với các tiêu chí của MVC.

- Không hỗ trợ xác nhận chính thức.

- Tăng độ phức tạp và kém hiệu quả của dữ liệu.

- Khó khăn khi sử dụng MVC với giao diện người dùng.

- Cần có nhiều lập trình viên để tiến hành lập trình song song.

- Phải có lượng kiến thức về công nghệ phải nhiều vì nó rất cần thiết.

- Duy trì rất nhiều mã trong controller (bộ điều khiển).

### **2.2.7. Một số công cụ được sử dụng**

#### **2.2.7.1. Postman**

Postman là một loại công cụ cho phép người dùng có thể thao tác với API, mà trong đó phổ biến nhất là REST. Với thử nghiệm API thì Postman là một trong những công cụ phổ biến vì được thực nghiệm nhiều nhất. Nhờ Postman lập trình viên có thể gọi Rest API mà không cần phải viết bất kỳ dòng code nào. Postman có khả năng hỗ trợ mọi phương thức HTTP bao gồm: POST, PUT, DELETE, PATCH, GET,... Ngoài ra, Postman còn cho phép lập trình viên lưu lại lịch sử của các lần request nên vô cùng tiện lợi cho nhu cầu sử dụng lại.

Graphical user interface, text, application, email

Description automatically generated

Hình 7. Giao diện chính của Postman

#### Logo, company name Description automatically generated**2.2.7.2. Visual Code**

Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code. Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux.

Hình 8. Logo Visual Code

Hơn thế nữa, VS Code còn cho khả năng tương thích với những thiết bị [**máy tính**](https://fptshop.com.vn/may-tinh-xach-tay) có cấu hình tầm trung vẫn có thể sử dụng dễ dàng.

Visual Studio Code hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting. Đặc biệt là tự hoàn thành mã thông minh, Snippets, và khả năng cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép các lập trình viên thay đổi Theme, phím tắt, và đa dạng các tùy chọn khác. Mặc dù trình soạn thảo Code này tương đối nhẹ, nhưng lại bao gồm các tính năng mạnh mẽ.

Một số ưu điểm của Visual Code:

- Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,…

- Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.

- Các tiện ích mở rộng rất đa dạng và phong phú.

- Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug),…

- Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.

### **2.2.8. Các khái niệm và số liệu mô tả về giống lúa**

#### **2.2.8.1. Các khái niệm**

**Phân loại nhóm giống lúa**

Viện Nghiên cứu lúa Quốc tế (IRRI 1996), cây lúa được phân loại dựa theo đặc điểm như sau:

- Phân loại theo quan điểm canh tác học: Quá trình thuần hóa và thích nghi với điều kiện sống và điều kiện canh tác khác nhau, cây lúa trồng được phân thành các nhóm: lúa nổi, lúa cạn,…

- Phân loại theo điều kiện tự nhiên, điều kiện sinh thái: tùy theo điều kiện sống tự nhiên tạo nên tính đặc trưng về hình thái cây lúa về đặc điểm lá, thân, bông, độ nở hạt, chống chịu, năng xuất.

- Phân loại theo địa lý: tùy theo vị trí địa lý cây lúa được trồng ở vĩ độ cao, thấp, ở vùng nhiệt độ cao, thấp,…

**Kiểu hình giống lúa**

Lúa là cây trồng đa dạng kiểu hình, mỗi giống có những đặc điểm riêng mà ta có thể dựa vào đó để nhận biết, đó là: thời gian sinh trưởng, cao cây, kích thước lá, màu sắc thân lá, dạng bông, dạng hạt, màu sắc hạt. Công tác chọn tạo và khảo nghiệm giống lúa, các nhà khoa học cần có những thông tin đầy đủ về các đặc điểm nguồn vật liệu khởi đầu của giống. Do đó, việc nghiên cứu đặc điểm hình thái, đặc điểm nông học, khả năng chống chịu của các giống lúa đã thu được nhiều kết quả có ý nghĩa.

**Giai đoạn sinh trưởng cây lúa**

Thời gian sinh trưởng của các giống lúa bắt đầu từ khi gieo đến khi thu hoạch được chia làm nhiều giai đoạn khác nhau, các giai đoạn sinh trưởng luôn biến động theo giống, mùa vụ, tác động của con người thông qua các biện pháp kỹ thuật trong quá trình sản xuất. Thời gian sinh trưởng của cây lúa có ý nghĩa quan trọng trong việc bố trí cơ cấu thời vụ, là điều kiện cần thiết để bố trí công thức luân canh tăng vụ, xây dựng chế độ luân canh cây trồng hợp lý nhằm nâng cao hiệu quả sử dụng đất

Thời gian sinh trưởng của một giống lúa cũng thay đổi (nhưng không lớn) nếu gieo trồng ở các thời vụ khác nhau, trong những điều kiện thời tiết, khí hậu, chân đất khác nhau.

Số ngày ở thời kỳ sinh trưởng sinh dưỡng thay đổi theo giống lúa (giống lúa ngắn ngày thì số ngày trong thời kỳ này rút ngắn và ngược lại). Số ngày ở các thời kỳ sinh trưởng sinh thực và thời kỳ chín được ổn định ít hoặc nhiều. Như vậy sự khác nhau trong suốt thời gian sinh trưởng được quyết định bởi số ngày ở kỳ sinh trưởng sinh dưỡng.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1. TỔNG QUAN VỀ HỆ THỐNG

### **3.1.1. Mô tả bài toán**

Ứng dụng Android quản lý dữ liệu hình ảnh mô tả giống lúa có chức năng chính là tra cứu các thông tin về giống lúa và thu thập số liệu về thuộc tính giống lúa được khảo sát từ thực tế. Các thông tin này được lưu trữ lên web API thông qua giao diện tra cứu, nhập liệu trên ứng dụng Android.

Hệ thống gồm một số chức năng cơ bản sau:

**Đăng nhập**: hệ thống yêu cầu người dùng sử dụng tài khoản được cung cấp từ nhà phát triển hoặc quản trị viên (Admin). Sau khi đăng nhập thành công người dùng có thể sử dụng ứng dụng để tra cứu và nhập liệu các thông tin giống lúa được thiết lập trên giao diện của ứng dụng.

**Tra cứu thông tin về giống lúa**: chức năng này cho người dùng khi đã đăng nhập vào ứng dụng để tra cứu các thông tin về giống lúa bao gồm: nhóm giống, kiểu hình, giống lúa, mã ngoài đồng, mã phòng thí nghiệm, giai đoạn trưởng thành, đối tượng tính trạng, đặc điểm tính trạng với giá trị tính trạng.

**Quản lý các danh mục chỉ tiêu** (chỉ tiêu ngoài đồng, chỉ tiêu trong nhà, chỉ tiêu sâu bệnh): sau khi đăng nhập vào hệ thống người dùng có thể nhập liệu và quản lý các thông tin về chỉ tiêu khi khảo sát, đo đạc giống lúa bao gồm:

- Chỉ tiêu ngoài đồng: loại giá trị đo, chỉ tiêu ngoài đồng, giá trị đo ngoài đồng.

- Chỉ tiêu trong nhà: loại giá trị đo, chỉ tiêu trong nhà, giá trị đo trong nhà.

- Chỉ tiêu sâu bệnh: loại sâu bệnh, chỉ tiêu sâu bệnh, giá trị đo sâu bệnh.

**Quản lý thông tin tài khoản cá nhân**: Người dùng có thể thay đổi thông tin cá nhân gồm: tên, địa chỉ, số điện thoại, mật khẩu,…

### **3.1.2. Sơ đồ tổng quan hệ thống**

Hệ thống quản lý dữ liệu hình ảnh mô tả giống lúa với thiết bị di động chạy hệ điều hành Android theo quy trình sau:

- Nhân viên nhập liệu (Staff data input) sẽ thu thập dữ liệu về giống lúa (Rice plant data collection) trên đồng ruộng, phòng thí nghiệm.

- Sau đó nhân viên nhập liệu sẽ nhập dữ liệu thông qua thiết bị di động (Android smartphone).

- Dữ liệu sau khi được nhập qua điện thoại sẽ được gửi yêu cầu lưu trữ qua hệ thống API.

- Thông qua phương thức HTTP, trong quá trình xử lý dữ liệu, dữ liệu sẽ được truyền tải từ hệ thống API và dữ liệu sẽ được lưu vào cơ sở dữ liệu (Database) của hệ thống.

- Quản lý hệ thống (Manager) có thể theo dõi và quản lý dữ liệu thông qua hệ web quản lý được xây dựng.

A picture containing weapon, text, screenshot

Description automatically generated

Hình 9. Sơ đồ tổng quan hệ thống

### **3.1.3. Yêu cầu chức năng**

- Quản lý các danh mục nhập số liệu giống lúa: Toàn bộ người dùng hệ thống đều có thể thực hiện tạo mới, sửa thông tin hoặc xóa thông tin các danh mục sau: chỉ tiêu ngoài đồng, chỉ tiêu trong nhà, chỉ tiêu sâu bệnh, giá trị đo trong nhà, giá trị đo ngoài đồng, giá trị đo sâu bệnh.

- Quản lý các danh mục thông tin giống lúa: Người quản tị hệ thống có thể tạo mới, thêm sửa thông tin hoặc xóa thông tin các danh mục: nhóm giống lúa, giống lúa, kiểu hình, các giai đoạn trưởng thành và tính trạng giống lúa, giá trị đo cơ bản và giá trị đo sâu bệnh.

- Chức năng tra cứu thông tin giống lúa: Toàn bộ người dùng hệ thống đều có thể thực hiện tra cứu thông tin các danh mục sau: nhóm giống lúa, giống lúa, kiểu hình, các giai đoạn trưởng thành và tính trạng giống lúa, giá trị đo cơ bản, giá trị đo sâu bệnh và các chỉ tiêu khảo sát đo giống lúa.

- Quản lý người dùng: Người quản trị hệ thống có quyền quản lý tất cả các tài khoản người dùng trong hệ thống. Người quản trị hệ thống có thể tạo mới, sửa thông tin, hoặc xóa thông tin của người dùng trong hệ thống. Đồng thời, người quản trị có thể phân cấp người dùng với các quyền hạn khác nhau làm tăng tính an toàn và bảo mật của hệ thống. Có 3 quyền bao gồm: Quản trị viên hệ thống, quản lý cơ sở, nhân viên nhập liệu.

Bảng 1. Yêu cầu chức năng hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Loại công việc** | **Tác nhân** |
| 1 | Quản lý danh mục nhập số liệu giống lúa:  - Chỉ tiêu ngoài đồng  - Chỉ tiêu trong nhà  - Chỉ tiêu sâu bệnh | - Tạo mới, sửa và xóa.  - Tra cứu | - Quản trị viên  - Nhân viên nhập liệu |
| 2 | Quản lý tài khoản | - Tạo tài khoản mới  - Phân quyền  - Chỉnh sửa hoặc xóa thông tin tài khoản  - Tra cứu | - Quản trị viên |
| - Xem, chỉnh sửa thông tin cá nhân  - Đổi mật khẩu | - Nhân viên nhập liệu |
| 3 | Tra cứu danh mục thông tin giống lúa:  - Nhóm giống  - Giống lúa  - Kiểu hình  - Giai đoạn trưởng thành  - Tính trạng giống lúa  - Loại sâu bệnh  - Giai đoạn trưởng thành  - Đối tượng tính trạng  - Đặc điểm tính trạng  - Giá trị tính trạng  - Loại giá trị đo  - Loại sâu bệnh | - Tạo mới, sửa và xóa.  - Tra cứu | - Quản trị viên |
| - Tra cứu | - Nhân viên nhập liệu |

A picture containing text, screenshot, font, number

Description automatically generated

Hình 10. Sơ đồ chức năng hệ thống

### **3.1.4. Yêu cầu phi chức năng**

Giao diện: đơn giản, hiện đại, giao diện thân thiện với người dùng, thuận tiện cho việc thu thập số liệu thực tế trên đồng ruộng.

Tính khả dụng: Mức độ tin cậy được ưu tiên cao, chạy tốt và khả dụng trên nhiều thiết bị di động phổ biến ngày nay.

Tính bảo mật: Đảm bảo các chức năng quản lý phải mang tính bảo mật  
vì chỉ sử dụng trong ít loại người dùng. Và các loại người dùng sẽ chỉ sử dụng những chức năng được phân quyền được định sẵn trong phạm vi phân quyền, để tránh sửa, xóa hoặc đánh cắp dữ liệu.

Tính tương tác: có khả năng tương tác dữ liệu từ website API đem lại sự tiện lợi cho người dùng.

Tính hiệu quả: Website được xây dựng dựa trên những yêu cầu, tài liệu  
thực tế. Phải đảm bảo khả năng vận hành tốt trong hệ thống nói chung và khả năng truy cập xử lý của người dùng nói riêng.

Tên chương trình: Xây dựng ứng dụng Android quản lý dữ liệu hình ảnh mô tả giống lúa.

### **3.1.5. Sơ đồ quy trình nghiệp vụ**

A picture containing text, screenshot, diagram, number

Description automatically generated

Hình 11. Sơ đồ quy trình nghiệp vụ

#### **3.1.5.1. Quy trình dành cho Quản lý**

Dựa theo biểu mẫu ghi chú quy định tính trạng, Quản lý sẽ thực hiện thêm, xóa, sửa nhằm quy định cho các danh mục quy định tính trạng.

A picture containing text, line, screenshot, receipt

Description automatically generated

Hình 12. Sơ đồ quy trình quản lý các tính trạng giống lúa

Các thông tin giống lúa nghiên cứu sẽ được Quản lý lưu vào hệ thống để  
quản lý thông tin giống lúa.

A picture containing text, screenshot, diagram, line

Description automatically generated

Hình 13. Sơ đồ quy trình quản lý thông tin giống lúa

#### **3.1.5.2. Quy trình dành cho Nhân viên**

Khi Nhân viên đi thu thập số liệu, trong quá trình thu thập số liệu thì Nhân viên sẽ dựa theo thông tin danh mục của Quản lý để thực hiện các quá trình thu thập, thêm, sửa, xóa các chỉ tiêu sao cho phù hợp.

A picture containing text, diagram, line, screenshot

Description automatically generated

Hình 14. Sơ đồ quy trình thu thập số liệu

### **3.1.6. Biểu mẫu nghiên cứu**

Biểu mẫu nghiên cứu được cung cấp bởi thầy Lê Thanh Phong thuộc Viện Biến đổi Khí hậu - Climate Change Institute. Biểu mẫu được dựa theo Lúa thuần - Quy trình kỹ thuật sản xuất hạt giống. (Bùi, 2006).

Một nhóm giống sẽ có nhiều giống lúa, nhằm phân biệt giống lúa thuộc nhóm giống nào.

A white card with black text

Description automatically generated with low confidence

Hình 15. Biểu mẫu nhóm giống lúa

Kiểu hình sẽ bao gồm các thông tin giống lúa, giá trị đo của các chỉ tiêu thuộc giống lúa đó gộp lại. Nhằm mục đích mô tả tổng quát toàn bộ thông tin cho người dùng.

A picture containing text, screenshot, document, font

Description automatically generated

Hình 16. Biểu mẫu kiểu hình giống lúa

Các giá trị đo của các cột thuộc nhóm chỉ tiêu ngoài đồng sẽ được nhập khi người dùng đi khảo sát giống lúa ngoài đồng. Và được đo ở giai đoạn lúa chín. Các giá trị đo sẽ được dựa theo quy chuẩn đặc điểm hình thái. (IRRI, 1996, tr.41)

A picture containing text, screenshot, line, number

Description automatically generated

Hình 17. Biểu mẫu nhập liệu giá trị đo chỉ tiêu ngoài đồng

Khi thu hoạch các giống lúa, các chỉ tiêu trong nhà sẽ được đo và nhập  
vào hệ thống. Các giá trị đo này sẽ được xác định dựa trên quy chuẩn chất  
lượng gạo (IRRI, 1996, tr.47) thu được từ các giống lúa.

A picture containing text, line, screenshot, number

Description automatically generated

Hình 18. Biểu mẫu nhập liệu giá trị đo chỉ tiêu trong nhà

Trong quá trình khảo sát các chỉ tiêu ngoài đồng, nếu phát hiện ra các nhóm sâu bệnh. Người dùng sẽ nhập các giá trị theo quy chuẩn thiệt hại mùa màng (bệnh hại). (IRRI, 1996, tr.22)

A picture containing text, screenshot, line, plot

Description automatically generated

Hình 19. Biểu mẫu nhập liệu giá trị đo chỉ tiêu sâu bệnh

A picture containing text, number, font, receipt

Description automatically generated

Hình 20. Biểu mẫu ghi chú quy định tính trạng

**Biểu mẫu ghi chú quy định tính trạng** được sắp xếp dựa theo 3 chỉ tiêu đo (chỉ tiêu ngoài đồng, chỉ tiêu trong nhà, chỉ tiêu sâu bệnh) được cung cấp bởi thầy Lê Thanh Phong thuộc Viện Biến đổi Khí hậu - Climate Change Institute.

Biểu mẫu dựa theo “**Hệ thống tiêu chuẩn đánh giá nguồn gen giống lúa**”. (IRRI, 1996). Các giá trị đo của các sẽ được người dùng dựa theo biểu mẫu ghi chú quy định tính trạng này để nhập giá trị đo theo các tính trạng trên biểu mẫu. Dựa theo mô tả thang điểm/mã số của IRRI (IRRI, 1996, tr.16), giai đoạn sinh trưởng của cây lúa gồm 9 giai đoạn:

1 - Nảy mầm

2 - Mạ

3 - Đẻ nhánh

4 - Vươn long

5 - Làm đòng

6 - Trỗ bông

7 - Chín sữa

8 - Vào chắc

9 - Chín

## 3.2. PHÂN TÍCH THIẾT KẾ CHI TIẾT

### **3.2.1. Phân tích cấu trúc và yêu cầu thông tin API đối với phương thức POST, PUT/PATCH**

**\*** Các trường id, created\_at và update\_at tự động cập nhật nên không được liệt kê vào các bảng Cấu trúc và yêu cầu thông tin API bên dưới.

#### **3.2.1.1. Giai đoạn trưởng thành (giaidoantruongthanhs)**

Bảng 2. Cấu trúc và yêu cầu thông tin API Giai đoạn trưởng thành

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| giaidoantt\_ten | Tên giai đoạn trưởng thành | String | có |
| giaidoantt\_mota | Mô tả giai đoạn trưởng thành | String | không |

#### **3.2.1.2. Đối tượng tính trạng (doituongtinhtrangs)**

Bảng 3. Cấu trúc và yêu cầu thông tin API Đối tượng tính trạng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| giaidoantruongthanh\_id | Mã giai đoạn trưởng thành  Khóa ngoại  (giaidoantruongthanhs) | int | có |
| doituongtt\_ten | Tên đối tượng tính trạng | String | có |
| doituongtt\_mota | Mô tả đối tượng tính trạng | String | không |

#### **3.2.1.3. Đặc điểm tính trạng (dacdiemtinhtrangs)**

Bảng 4. Cấu trúc và yêu cầu thông tin API Đặc điểm tính trạng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| doituongtinhtrang\_id | Mã đối tượng tính trạng  Khóa ngoại (doituongtinhtrangs) | int | có |
| dacdiemtt\_ten | Tên đối tượng tính trạng | String | có |

#### **3.2.1.4. Giá trị tính trạng (giatritinhtrang)**

Bảng 5. Cấu trúc và yêu cầu thông tin API Giá trị tính trạng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| dacdiemtinhtrang\_id | Mã đặc điểm tính trạng  Khóa ngoại (dacdiemtinhtrangs) | int | có |
| giatritt\_diem | Điểm đánh giá  giá trị tính trạng | String | có |

#### **3.2.1.5. Chỉ tiêu ngoài đồng (chitieungoaidongs)**

Bảng 6. Cấu trúc và yêu cầu thông tin API Chỉ tiêu ngoài đồng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| giong\_id | Mã giong  Khóa ngoại (giongs) | int | có |
| chitieungoaidong\_phienla | Đặc điểm phiến lá | String | không |
| chitieungoaidong\_taila | Đặc điểm tai lá | String | không |
| chitieungoaidong\_gocnhanh | Đặc điểm góc nhánh | String | không |
| chitieungoaidong\_bela | Đặc điểm góc bẹ lá | String | không |
| chitieungoaidong\_gocla | Đặc điểm góc lá | String | không |
| chitieungoaidong\_msgocla | Đặc điểm màu sắc góc lá | String | không |
| chitieungoaidong\_gocladong | Đặc điểm góc lá dòng | String | không |
| chitieungoaidong\_thoatcb | Đặc điểm thoát cổ bông | String | không |
| chitieungoaidong\_msgocthan | Đặc điểm màu sắc góc thân | String | không |
| chitieungoaidong\_dangbong | Đặc điểm dáng bông | String | không |
| chitieungoaidong\_congtrucbong | Đặc điểm congtrucbong | String | không |
| chitieungoaidong\_rau | Đặc điểm râu | String | không |

#### **3.2.1.6. Chỉ tiêu trong nhà (chitieutrongnhas)**

Bảng 7. Cấu trúc và yêu cầu thông tin API Chỉ tiêu trong nhà

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| giong\_id | Mã giống  Khóa ngoại (giongs) | int | có |
| chitieutrongnha\_giec2 | Đặc điểm gié C2 | String | không |
| chitieutrongnha\_dorunghat | Mức độ rụng hạt | String | không |
| chitieutrongnha\_msvotrau | Màu sắc vỏ trấu | String | không |
| chitieutrongnha\_dangthoc | Đặc điểm hình dáng thóc | String | không |
| chitieutrongnha\_mausacgao | Màu sắc gạo | String | không |
| chitieutrongnha\_tl1000hat | Trọng lượng 1000 hạt | String | không |
| chitieutrongnha\_doam | Độ ẩm | String | không |
| chitieutrongnha\_thom | Mùi thơm | String | không |
| chitieutrongnha\_danhgia | Đánh giá | String | không |

#### **3.2.1.7. Chỉ tiêu sâu bệnh (chitieusaubenhs)**

Bảng 8. Cấu trúc và yêu cầu thông tin API Chỉ tiêu sâu bệnh

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| giong\_id | Mã giống  Khóa ngoại (giongs) | int | có |
| chitieusaubenh\_chonloc | Chọn lọc giống lúa | String | không |
| chitieusaubenh\_danhgia | Đánh giá | String | không |

#### **3.2.1.8. Giá trị đo ngoài đồng (giatridongoaidongs)**

Bảng 9. Cấu trúc và yêu cầu thông tin API Giá trị đo ngoài đồng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| chitieungoaidong\_id | Mã chỉ tiêu ngoài đồng  Khóa ngoại (chitieungoaidongs) | int | có |
| loaigiatrido\_id | Mã giá trị đo  Khóa ngoại (loaigiatridos) | int | có |
| giatridongoaidong\_giatri | Số liệu giá trị đo | String | có |

#### **3.2.1.9. Giá trị đo trong nhà (giatridotrongnhas)**

Bảng 10. Cấu trúc và yêu cầu thông tin API Giá trị đo trong nhà

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Mô tả** | **Kiểu**  **dữ liệu** | **Bắt buộc** |
| chitieutrongnha\_id | Mã chỉ tiêu ngoài đồng  Khóa ngoại (chitieutrongnhas) | int | có |
| loaigiatrido\_id | Mã giá trị đo  Khóa ngoại (loaigiatridos) | int | có |
| giatridotrongnha\_giatri | Số liệu giá trị đo | String | có |

### **3.2.2. Mô tả các tác nhân**

Bảng 11. Mô tả các tác nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Mô tả** |
| 1 | Quản trị viên | Quản lý nhóm giống, kiểu hình, giống, mã ngoài đồng, mã phòng thí nghiệm, giai đoạn trưởng thành, đối tượng tính trạng, đặc điểm tính trạng, giá trị tính trạng, loại giá trị đo, chỉ tiêu ngoài đồng, giá trị đo ngoài đồng, chỉ tiêu trong nhà, giá trị đo trong nhà, loại sâu bệnh, chỉ tiêu sâu bệnh, giá trị đo sâu bệnh, tài khoản, phân quyền |
| 2 | Quản lý | Quản lý nhóm giống, kiểu hình, giống, mã ngoài đồng, mã phòng thí nghiệm, giai đoạn trưởng thành, đối tượng tính trạng, đặc điểm tính trạng, giá trị tính trạng, loại giá trị đo, chỉ tiêu ngoài đồng, giá trị đo ngoài đồng, chỉ tiêu trong nhà, giá trị đo trong nhà, loại sâu bệnh, chỉ tiêu sâu bệnh, giá trị đo sâu bệnh. |
| 3 | Nhân viên nhập liệu | Quản lý loại giá trị đo, chỉ tiêu ngoài đồng, giá trị đo ngoài đồng, chỉ tiêu trong nhà, giá trị đo trong nhà, loại sâu bệnh, chỉ tiêu sâu bệnh, giá trị đo sâu bệnh. |

### **3.2.3. Mô tả UseCase**

Bảng 12. Mô tả UseCase

|  |  |  |
| --- | --- | --- |
| **STT** | **Use case** | **Mô tả** |
| 1 | Đăng nhập | Người dùng đăng nhập vào hệ thống bằng email và password được tạo sẵn. |
| 2 | Đăng xuất | Người dùng đăng xuất khỏi hệ thống. |
| 3 | Đổi mật khẩu | Người dùng đăng nhập vào hệ thống có thể đổi mật khẩu hiện tại. |
| 4 | Quản lý nhóm giống | Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin nhóm giống (Xem, tìm kiếm). |
| 5 | Quản lý kiểu hình | Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin kiểu hình (xem, tìm kiếm). |
| 6 | Quản lý giống | Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin giống (xem, tìm kiếm). |
| 7 | Quản lý mã Ngoài đồng | Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin mã ngoài đồng (xem, tìm kiếm). |
| 8 | Quản lý mã Phòng thí nghiệm | Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin mã phòng thí nghiệm (xem, tìm kiếm). |
| 9 | Quản lý giai đoạn trưởng thành | Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin giai đoạn trưởng thành (thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 10 | Quản lý tính trạng | Bao gồm: đối tượng tính trạng, đặc điểm tính trạng và giá trị tính trạng  Quản trị viên hoặc quản lý đăng nhập vào hệ thống có thể quản lý thông tin tính trạng (thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 11 | Quản lý loại giá trị đo | Nhân viên quản lý (quản trị viên hoặc quản lý) đăng nhập vào hệ thống có thể quản lý thông tin loại giá trị đo (thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 12 | Quản lý chỉ tiêu ngoài đồng | Bao gồm: quản lý chỉ tiêu ngoài đồng và giá trị đo ngoài đồng  Nhân viên quản lý (quản trị viên hoặc quản lý) đăng nhập vào hệ thống có thể quản lý về chỉ tiêu đo ngoài đồng(thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 13 | Quản lý chỉ tiêu trong nhà | Bao gồm: quản lý chỉ tiêu trong nhà và giá trị đo trong nhà  Nhân viên quản lý (quản trị viên hoặc quản lý) đăng nhập vào hệ thống có thể quản lý về thông tin chỉ tiêu trong nhà (thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 14 | Quản lý loại sâu bệnh | Nhân viên quản lý (quản trị viên hoặc quản lý) đăng nhập vào hệ thống có thể quản lý thông tin loại sâu bệnh (thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 15 | Quản lý chỉ tiêu sâu bệnh | Bao gồm: quản lý chỉ tiêu sâu bệnh và giá trị đo sâu bệnh  Nhân viên quản lý (quản trị viên hoặc quản lý) đăng nhập vào hệ thống có thể quản lý thông tin chỉ tiêu sâu bệnh (thêm, sửa, xóa, xem chi tiết, tìm kiếm). |
| 16 | Quản lý tài khoản | Xem thông tin cá nhân, đổi mật khẩu |

### **3.2.4. Sơ đồ UseCase**

#### **3.2.4.1. Sơ đồ UseCase Tổng quát**

A picture containing text, diagram, drawing, line

Description automatically generated

Hình 21. Sơ đồ UseCase Tổng quát

#### **3.2.4.2. Sơ đồ phân rã UseCase Quản lý nhóm giống**

Diagram

Description automatically generated

Hình 22. Sơ đồ phân rã UseCase Quản lý nhóm giống

#### **3.2.4.3.** **Sơ đồ phân rã UseCase Quản lý kiểu hình**

Diagram

Description automatically generated

Hình 23. Sơ đồ phân rã UseCase Quản lý kiểu hình

#### **3.2.4.4. Sơ đồ phân rã UseCase Quản lý giống lúa**

Diagram

Description automatically generated

Hình 24. Sơ đồ phân rã UseCase Quản lý Giống lúa

**3.2.4.5. Sơ đồ phân rã UseCase Quản lý mã ngoài đồng**

Diagram

Description automatically generated

Hình 25. Sơ đồ phân rã UseCase Quản lý mã ngoài đồng

#### **3.2.4.6. Sơ đồ phân rã UseCase Quản lý mã phòng thí nghiệm**

Diagram

Description automatically generated

Hình 26. Sơ đồ phân rã UseCase Quản lý mã phòng thí nghiệm

#### **3.2.4.7. Sơ đồ phân rã UseCase Quản lý giai đoạn trưởng thành**

Diagram

Description automatically generated

Hình 27. Sơ đồ phân rã UseCase Quản lý giai đoạn trưởng thành

#### **3.2.4.8. Sơ đồ phân rã UseCase Quản lý tính trạng**

Diagram

Description automatically generated

Hình 28. Sơ đồ phân rã UseCase Quản lý tính trạng

#### **3.2.4.9. Sơ đồ phân rã Use Case Quản lý loại giá trị đo**

Diagram

Description automatically generated

Hình 29. Sơ đồ phân rã UseCase Loại giá trị đo

#### **3.2.4.10. Sơ đồ phân rã UseCase Quản lý loại sâu bệnh**

Diagram

Description automatically generated

Hình 30. Sơ đồ phân rã UseCase Loại sâu bệnh

#### **3.2.4.11. Sơ đồ phân rã UseCase Quản lý chỉ tiêu ngoài đồng**

Diagram

Description automatically generated

Hình 31. Sơ đồ phân rã UseCase Quản lý chỉ tiêu ngoài đồng

#### **3.2.4.12. Sơ đồ phân rã UseCase Quản lý chỉ tiêu trong nhà**

Diagram

Description automatically generated

Hình 32. Sơ đồ phân rã UseCase Quản lý chỉ tiêu trong nhà

#### **3.2.4.13. Sơ đồ phân rã UseCase Quản lý chỉ tiêu sâu bệnh**

Diagram

Description automatically generated

Hình 33. Sơ đồ phân rã UseCase Quản lý chỉ tiêu sâu bệnh

### **3.2.5. Đặc tả một vài UseCase đại diện**

#### **3.2.5.1. Đăng nhập**

Bảng 13. Đặc tả UseCase Đăng nhập

|  |  |
| --- | --- |
| **Tên:** Đăng nhập | **ID:** UC1 |
| **Tác nhân chính:** người dùng | |
| **Mô tả ngắn gọn:** Người dùng đăng nhập vào hệ thống | |
| **Điều kiện tiên quyết:** Hệ thống đang hoạt động | |
| **Điều kiện thực hiện:** Người dùng phải có tài khoản và chưa đăng nhập | |
| **Sự kiện kích hoạt:** Người dùng nhấn vào nút “Đăng nhập” trong menu hệ thống | |
| **Luồng sự kiện chính:**   1. Điền đầy đủ tên tài khoản và mật khẩu 2. Nhấn vào nút “Đăng nhập” 3. Hệ thống gửi yêu cầu đăng nhập, nếu không chính xác thì chuyển sang luồng thay thế 1 4. Thông tin hợp lệ thì đăng nhập người dùng và chuyển về trang chính | |
| **Các luồng thay thế:**   1. Thông báo thông tin đăng nhập không chính xác | |

#### **3.2.5.2. Đăng xuất**

Bảng 14. Đặc tả Usecase Đăng xuất

|  |  |
| --- | --- |
| **Tên:** Đăng xuất | **ID:** UC2 |
| **Tác nhân chính:** Người dùng | |
| **Mô tả ngắn gọn:** Đăng xuất người dùng khỏi hệ thống | |
| **Điều kiện tiên quyết:** Hệ thống đang hoạt động | |
| **Điều kiện thực hiện:** Người dùng đã đăng nhập | |
| **Sự kiện kích hoạt:** Người dùng nhấn vào nút “Đăng xuất”. | |
| **Luồng sự kiện chính:**   1. Người dùng nhấn vào nút “Đăng xuất” 2. Hệ thống gửi yêu cầu đăng xuất 3. Hệ thống loại bỏ đăng nhập và chuyển người dùng về trang chính | |

#### **3.2.5.3. Xem thông tin giai đoạn trưởng thành**

Bảng 15. Đặc tả UseCase Xem danh sách Giai đoạn trưởng thành

|  |  |
| --- | --- |
| **Tên:** Xem thông tin giai đoạn trưởng thành | **ID:** UC3 |
| **Tác nhân chính:** người dùng | |
| **Mô tả ngắn gọn:** Hiển thị thông tin chính của giai đoạn trưởng thành,thông tin chi tiết của giai đoạn trưởng thành | |
| **Điều kiện tiên quyết:** Hệ thống đang hoạt động | |
| **Điều kiện thực hiện:** Có ít nhất một giai đoạn trưởng thành | |
| **Sự kiện kích hoạt:** Người dùng nhấn Xem giai đoạn trưởng thành trên trang chính | |
| **Luồng sự kiện chính:**   1. Người dùng nhấn vào Xem giai đoạn trưởng thành trên trang chính 2. Người dùng nhấn vào từng giai đoạn có thể xem chi tiết về từng giai đoạn. 3. Hệ thống truy xuất thông tin giai đoạn và hiển thị lên màn hình | |

#### **3.2.5.4. Thêm giai đoạn trưởng thành mới**

Bảng 16. Đặc tả UseCase Thêm giai đoạn mới

|  |  |
| --- | --- |
| **Tên:** Thêm giai đoạn trưởng thành mới | **ID:** UC4 |
| **Tác nhân chính:** người dùng | |
| **Mô tả ngắn gọn:** Thêm giai đoạn trưởng thành mới của giống lúa vào dữ liệu | |
| **Điều kiện tiên quyết:** Hệ thống đang hoạt động | |
| **Điều kiện thực hiện:** Người dùng đã đăng nhập và thực hiện thêm giai đoạn | |
| **Sự kiện kích hoạt:** Người dùng nhấn nút Thêm trên màn hình giao diện hiển thị danh sách giai đoạn | |
| **Luồng sự kiện chính:**   1. Người dùng nhấn Xem giai đoạn trưởng thành trên trang chính 2. Người dùng nhấn nút “Thêm” 3. Hệ thống kiểm tra nếu thông tin nhập chưa đầy đủ thì thực hiện luồng thay thế 1 4. Thông tin hợp lệ thì hệ thống sẽ gởi yêu cầu thêm giai đoạn mới và trở về trang chính | |
| **Các luồng thay thế:**   1. Hệ thống yêu cầu điền đầy đủ thông tin để tiếp tục | |

#### **3.2.5.5. Sửa giai đoạn trưởng thành**

Bảng 17. Đặc tả UseCase Sửa thông tin giai đoạn trưởng thành

|  |  |
| --- | --- |
| **Tên:** Sửa thông tin giai đoạn trưởng thành | **ID:** UC5 |
| **Tác nhân chính:** Người dùng | |
| **Mô tả ngắn gọn:** Sửa thông tin giai đoạn trưởng thành | |
| **Điều kiện tiên quyết:** Hệ thống còn hoạt động | |
| **Điều kiện thực hiện:** Người dùng đã đăng nhập, người dùng có thể sửa giai đoạn trưởng thành | |
| **Sự kiện kích hoạt:** Người dùng nhấn nút sửa trong trang xem thông tin giai đoạn trưởng thành | |
| **Luồng sự kiện chính:**   1. Người dùng chọn giai đoạn trưởng thành cần chỉnh sửa thông tin 2. Hệ thống load dữ liệu của giai đoạn trưởng thành tương ứng vào các input để có thể thay đổi thông tin 3. Người dùng thay đổi các thông tin cần chỉnh sửa và nhấn vào nút “Lưu” 4. Hệ thống kiểm tra nếu thông tin nhập chưa đầy đủ thì thực hiện luồng thay thế 1 5. Thông tin hợp lệ thì hệ thống sẽ cập nhật thông tin mới cho giai đoạn trưởng thành và thông báo kết quả cho người dùng | |
| **Các luồng thay thế:**   1. Yêu cầu điền đầy đủ thông tin và tiếp tục | |

#### **3.2.5.6. Xóa giai đoạn trưởng thành**

Bảng 18. Đặc tả UseCase xóa giai đoạn trưởng thành

|  |  |
| --- | --- |
| **Tên:** Xóa giai đoạn trưởng thành | **ID:** UC6 |
| **Tác nhân chính:** người dùng | |
| **Mô tả ngắn gọn:** Xóa một giai đoạn trưởng thành | |
| **Điều kiện tiên quyết:** Hệ thống đang hoạt động | |
| **Điều kiện thực hiện:** Người dùng đã đăng nhập | |
| **Sự kiện kích hoạt:** Người dùng chọn giai đoạn trưởng thành cần xóa, nhấn nút “Xóa” và chọn đồng ý khi hộp thoại xác nhận xóa hiện lên | |
| **Luồng sự kiện chính:**   1. Người dùng chọn giai đoạn trưởng thành cần xóa và nhấn nút “Xóa” 2. Hệ thống sẽ hiển thị một hộp thoại để xác nhận có chắc muốn xóa hay không 3. Nếu chọn không hệ thống quay lại giao diện xem thông tin giai đoạn trưởng thành 4. Nếu chọn có hệ thống gửi yêu cầu xóa giai đoạn trưởng thành, đồng thời thông báo kết quả cho người dùng | |

### **3.2.6. Sơ đồ tuần tự**

#### **3.2.6.1. Sơ đồ tuần tự cho UseCase Đăng nhập**

Diagram

Description automatically generated with medium confidence

Hình 34. Sơ đồ tuần tự UseCase Đăng nhập

#### 

#### **3.2.6.2. Sơ đồ tuần tự cho đăng xuất**

Diagram

Description automatically generated

Hình 35. Sơ đồ tuần tự UseCase Đăng xuất

#### **3.2.6.3. Sơ đồ tuần tự cho các UseCase Thêm giai đoạn trưởng thành**

Diagram

Description automatically generated

Hình 36. Sơ đồ tuần tự cho UseCase Thêm giai đoạn trưởng thành

#### **3.2.6.4. Sơ đồ tuần tự UseCase Sửa thông tin giai đoạn trưởng thành**

Diagram

Description automatically generated

Hình 37. Sơ đồ tuần tự UseCase Sửa dữ liệu

#### **3.2.6.5. Sơ đồ tuần tự UseCase Xóa thông tin giai đoạn trưởng thành**

Diagram

Description automatically generated

Hình 38. Sơ đồ tuần tự UseCase Xóa dữ liệu

### **3.2.7. Sơ đồ lớp (dựa trên API)**



Hình 39. Sơ đồ lớp quản lý giống lúa

## 3.3. THIẾT KẾ GIAO DIỆN

### **3.3.1. Giao diện đăng nhập**

|  |  |  |
| --- | --- | --- |
| Graphical user interface, text, application  Description automatically generated  Hình 40. Giao diện đăng nhập | A screenshot of a phone  Description automatically generated with low confidence  Hình 42. Giao diện đăng nhập fail | A screenshot of a phone  Description automatically generated with low confidence  Hình 43. Giao diện đăng nhập success |

Ở trang giao diện đăng nhập, người dùng sẽ thực hiện nhập ‘email’ và  
‘password’ đã được cung cấp hoặc đăng ký.

### **3.3.2. Giao diện chính**

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 44. Giao diện chính

Giao diện chính sau khi đăng nhập thành công, tại đây có thể tra cứu: nhóm giống, giống lúa , kiểu hình , giai đoạn và tính trạng giống lúa,… ngoài ra còn quản lý bao gồm: giai đoạn trưởng thành, đối tượng tính trạng, đặc điểm tính trạng, giá  
trị tính trạng, loại giá trị đo, chỉ tiêu ngoài đồng, giá trị đo ngoài đồng, chỉ tiêu  
trong nhà, giá trị đo trong nhà, loại sâu bệnh, chỉ tiêu sâu bệnh, giá trị đo sâu  
bệnh.

### **3.3.3. Giao diện tra cứu**

Graphical user interface, application

Description automatically generated

Hình 45. Giao diện tra cứu

Tại giao diện tra cứu có thể nhập từ khóa vào ô tìm kiếm và dữ liệu sẽ hiển thị ra màn hình phù hợp với từ khóa tìm kiếm

### **3.3.4. Giao diện thêm dữ liệu**

Graphical user interface, application

Description automatically generated

Hình 46. Giao diện thêm dữ liệu

Khi nhấn vào nút Thêm hộp thoại nhập thông tin sẽ hiện ra, nhập thông tin và nhấn nút Lưu để hoàn thành việc thêm dữ liệu

### **3.3.5. Giao diện sửa dữ liệu**

Graphical user interface, application

Description automatically generated

Hình 47. Giao diện sửa dữ liệu

Khi chọn một dữ liệu cần sửa, nhấn vào nút Sửa, thông tin dữ liệu được đổ lên hộp thoại hiện lên, điều chỉnh thông tin và nhấn nút Lưu để hoàn thành việc sửa dữ liệu

### **3.3.6. Giao diện xóa**

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 48. Giao diện xóa dữ liệu

Chọn đối tượng cần xóa, hộp thoại xác nhận xóa hiện lên, chọn có hoặc không để hoàn thành việc xóa dữ liệu

### **3.3.7. Giao diện xem chi tiết**

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 49. Giao diện xem chi tiết

## 3.4. KẾT QUẢ

Trong suốt quá trình thực tập và thực hiện đề tài, em đã học được rất nhiều kiến thức mới giúp nâng cao khả năng chuyên môn của bản thân.

Ứng dụng đã thực hiện được những chức năng sau:

- Kết nối và đổ dữ liệu từ hệ thông API

- Đăng nhập hệ thống

- Tra cứu thông tin giống lúa

- Nhập số liệu khảo sát giống lúa.

## 3.5. Ý NGHĨA

Đề tài giúp các nhà nghiên cứu, các thu thập viên có thể thuận tiện hơn trong quá trình khảo sát và thu thập số liệu về thông tin giống lúa thực tế. Sản phẩm của đề tài là mô hình thực tế cho việc ứng dụng di động trong việc thu thập số liệu.

## 3.6. HƯỚNG PHÁT TRIỂN

Qua đề tài em nhận thấy ứng dụng cần được hoàn thiện thêm về:

- Tối ưu hóa về mặt dữ liệu, thời gian xử lý còn chậm, chưa được linh hoạt

- Ứng dụng phát triển thêm trên hệ điều hành IOS

- Nâng cấp thêm về giao diện, tang tinh tương tác với người dùng.

# PHỤ LỤC

## 1. DANH SÁCH CÁC LIÊN KẾT (URL) API, REQUEST – RESPONSE

### **1.1. Người dùng (Users)**

Bảng 20. Request, Response lấy thông tin người dùng

|  |
| --- |
| **Request** |
| **GET** <http://localhost>/api/users |
| **Response** |
| **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách tài khoản",      "data": [          {              "id": 1,              "name": "Admin",              "email": "Admin@gmail.com"          }      ]  } |

### **1.2. Đăng nhập và lấy thông tin người dùng đăng nhập**

**Đăng nhập**

Bảng 21. Request, Response Đăng nhập

|  |  |
| --- | --- |
| **Request** | |
| **POST** <http://localhost>/api/auth/logins | |
| **Query Parameter** | |
| **key** | **value** |
| email | email |
| password | matkhau |
| **Response** | |
| **Đăng nhập thành công**: (200 OK)  {      "status": "success",      "access\_token": "token code",      "token\_type": "Bearer",      "expires\_at": "2024-04-13 16:15:23"  }  **Đăng nhập thất bại**: (401 Unauthorized)  {      "status": "fails",      "message": "Unauthorized"  } | |

**Lấy thông tin người dùng đăng nhập**

Bảng 22. Request, Response Lấy thông tin người dùng đăng nhập

|  |  |
| --- | --- |
| **Request** | |
| **GET** <http://localhost>/api/auth/user | |
| **Authorization** | |
| Type: Bearer token | <token của người dùng> |
| **Response** | |
| **Lấy thông tin người dùng thành công**: (200 OK)  {      "id": 1,      "name": "Admin",      "email": "Admin@gmail.com",      "email\_verified\_at": null,      "address": "Admin",      "phone": "0123456789",      "created\_at": "2023-03-30T18:24:14.000000Z",      "updated\_at": "2023-03-30T18:24:14.000000Z"  } | |

### **1.3. Đăng xuất**

Bảng 23. Request, Response Đăng xuất

|  |  |
| --- | --- |
| **Request** | |
| **DELETE** <http://localhost>/api/auth/logout | |
| **Authorization** | |
| Type: Bearer token | <token của người dùng> |
| **Response** | |
| **Đăng xuất thành công**: (200 OK)  {"status": "success"} | |

### **1.4. Nhóm giống, giống, kiểu hình**

**Nhóm giống**

Bảng 24. Request, Response Lấy danh sách nhóm giống

|  |
| --- |
| **Request** |
| **GET** <http://localhost>/api/nhomgiongs |
| **Response** |
| **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách nhóm giống",      "data": [          {              "id": 1,              "nhomgiong\_code": "LMNN",              "nhomgiong\_ten": "Lúa mùa nhập nội",              "nhomgiong\_mota": "Đây là nhóm giống lúa thuộc lúa mùa nổi",              "nhomgiong\_ngayngam": "2002-04-04",              "nhomgiong\_ngaycay": null,              "created\_at": "31/03/2023",              "updated\_at": "05/04/2023"          },          {              "id": 2,              "nhomgiong\_code": "LMR",              "nhomgiong\_ten": "Lúa mùa rẫyy",              "nhomgiong\_mota": "Đây là nhóm giống lúa mùa rẫy",              "nhomgiong\_ngayngam": "2022-02-22",              "nhomgiong\_ngaycay": "2023-03-02",              "created\_at": "03/04/2023",              "updated\_at": "08/04/2023"          },  {…..}      ]  } |

**Kiểu hình**

Bảng 25. Request, Response Lấy danh sách kiểu hình

|  |
| --- |
| **Request** |
| **GET** <http://localhost>/api/kieuhinhs |
| **Response** |
| **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách kiểu hình",      "data": [          {              "id": 1,              "kieuhinh\_ten": "Kiểu hình thon thấp",              "kieuhinh\_mota": "Đây là kiểu hình thon thấp giống lúa",              "created\_at": "31/03/2023",              "updated\_at": "31/03/2023"          },  {…..}      ]  } |

**Giống**

Bảng 26. Request, Response Lấy danh sách giống lúa

|  |
| --- |
| **Request** |
| **GET** <http://localhost>/api/giongs |
| **Response** |
| **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách giống",      "data": [          {              "id": 1,              "giong\_ten": "Lúa f2",              "nhomgiong\_ten": "Lúa mùa nổi",              "kieuhinh\_ten": "Kiểu hình 02",              "giong\_nguongoc": "Thoại Sơn",              "giong\_mota": "Đây là lúa mùa nổi bộ gen f2",              "giong\_hinhanh": "images/lua-f2.jpeg",              "created\_at": "11/04/2023",              "updated\_at": "11/04/2023"          },  {…..}      ]  } |

### **1.5. Giai đoạn trưởng thành**

Bảng 27. Request, Response Giai đoạn trưởng thành

|  |  |
| --- | --- |
| **Request** | **Response** |
| **GET**  http://localhost/api/giaidoantruongthanhs | **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách giai đoạn trưởng thành",      "data": [          {              "id": 2,              "giaidoantt\_ten": "Giai đoạn hạt nảy mầm",              "giaidoantt\_mota": "Đây là giai đoạn nảy mầm",              "created\_at": "06/04/2023",              "updated\_at": "13/04/2023"          },          {…}      ]  } |
| **POST**  <http://localhost/api/giaidoantruongthanhs>  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | giaidoantt\_ten | Tên giai đoạn trưởng thành | | giaidoantt\_mota | mô tả giai đoạn trưởng thành | | **Thêm thành công** (201 Created)  {      "status": true,      "message": "Giai đoạn trưởng thành đã lưu thành công",      "data": {          "id": 3,          "giaidoantt\_ten": "Tên giai đoạn trưởng thành",          "giaidoantt\_mota": "mô tả giai đoạn trưởng thành",          "created\_at": "13/04/2023",          "updated\_at": "13/04/2023"      }  } |
| **PUT / PATCH**  <http://localhost/api/giaidoantruongthanhs/id>  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | giaidoantt\_ten | Tên giai đoạn | | giaidoantt\_mota | Mô tả giai đoạn | | **Cập nhật thành công** (200 OK)  {      "status": true,      "message": "Giai đoạn trưởng thành đã cập nhật thành công",      "data": {          "id": 165,          "giaidoantt\_ten": "Tên giai đoạn",          "giaidoantt\_mota": "Mô tả giai đoạn",          "created\_at": "13/04/2023",          "updated\_at": "13/04/2023"      }  } |
| **DELETE**  http://localhost/api/giaidoantruongthanhs/id | **Xóa thành công** (200 OK)  {      "status": true,      "message": "Giai đoạn trưởng thành đã được xoá} |

### **1.6. Đối tượng tính trạng**

Bảng 28. Request, Response Đối tượng tính trạng

|  |  |
| --- | --- |
| **Request** | **Response** |
| **GET** http://localhost/api/doituongtinhtrangs | |
| **POST**  [http://localhost/api/doituongtinhtrangs](http://localhost/api/giaidoantruongthanhs)  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | giaidoantruongthanh\_id | 1 | | doituongtt\_ten  doituongtt\_mota | Thân  Đây là thân | | **Thêm thành công** (201 Created)  {      "status": true,      "message": "Đối tượng tính trạng đã lưu thành công",      "data": {              "id": 2,              "giaidoantt\_ten": "Giai đoạn hạt nảy mầm",              "doituongtt\_ten": "Thân",              "doituongtt\_mota": "Đây là thân",              "created\_at": "12/04/2023",              "updated\_at": "12/04/2023"      }  } |
| **PUT / PATCH**  [http://localhost/api/doituongtinhtrangs/id](http://localhost/api/giaidoantruongthanhs/id)  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | Giaidoantruongthanh\_id | 1 | | doituongtt\_ten  doituongtt\_mota | Thân  Đây là thân cây lúa | | **Cập nhật thành công** (200 OK)  {      "status": true,      "message": "Đối tượng tính trạng đã cập nhật thành công",      "data": {           "id": 2,              "giaidoantt\_ten": "Giai đoạn hạt nảy mầm",              "doituongtt\_ten": "Thân",              "doituongtt\_mota": "Đây là thân cây lúa",              "created\_at": "12/04/2023",              "updated\_at": "12/04/2023"      }  } |
| **DELETE** http://localhost/api/doituongtinhtrangs/id | |

### **1.7. Đặc điểm tính trạng**

Bảng 29. Request, Response Đặc điểm tính trạng

|  |  |
| --- | --- |
| **Request** | **Response** |
| **GET** http://localhost/api/dacdiemtinhtrangs | |
| **POST**  [http://localhost/api/dacdiemtinhtrangs](http://localhost/api/dacdiemtinhtrangs%20)  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | doituongtinhtrang\_id | 1 | | dacdiemtt\_ten | Màu xanh | | **Thêm thành công** (201 Created)  {      "status": true,      "message": "Đặc điểm tính trạng đã lưu thành công",      "data": {              "id": 1,              "doituongtt\_ten": "lá",              "dacdiemtt\_ten": "Màu xanh",              "created\_at": "06/04/2023",              "updated\_at": "06/04/2023"      }  } |
| **PUT / PATCH**  <http://localhost/api/dacdiemtinhtrangs/id>  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | doituongtinhtrang\_id | 1 | | dacdiemtt\_ten | Màu vàng | | **Cập nhật thành công** (200 OK)  {      "status": true,      "message": "Đặc điểm tính trạng đã cập nhật thành công",      "data": {           "id": 1,              "doituongtt\_ten": "lá",              "dacdiemtt\_ten": "Màu vàng",              "created\_at": "06/04/2023",              "updated\_at": "06/04/2023"      }  } |
| **DELETE** http://localhost/api/dacdiemtinhtrangs/id | |

### **1.8. Giá trị tính trạng**

Bảng 30. Request, Response Giá trị tính trạng

|  |  |
| --- | --- |
| **Request** | **Response** |
| **GET** http://localhost/api/giatritinhtrangs | |
| **POST**  [http://localhost/api/giatritinhtrangs](http://localhost/api/giatritinhtrangs%20)  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | giatritt\_diem | 5 | | dacdiemtinhtrang\_id | 1 | | **Thêm thành công** (201 Created)  {      "status": true,      "message": "Giá trị tính trạng đã lưu thành công",      "data": {  "id": 1,              "dacdiemtt\_ten": "Màu xanh",              "giatritt\_diem": "5",              "created\_at": "06/04/2023",              "updated\_at": "06/04/2023"      }  } |
| **PUT / PATCH**  <http://localhost/api/giatritinhtrangs/id>  **Query Parameter**   |  |  | | --- | --- | | **key** | **value** | | giatritt\_diem | 7 | | dacdiemtinhtrang\_id | 1 | | **Cập nhật thành công** (200 OK)  {      "status": true,      "message": "Giá trị tính trạng đã cập nhật thành công",      "data": {           "id": 1,              "dacdiemtt\_ten": "Màu xanh",              "giatritt\_diem": "7",              "created\_at": "06/04/2023",              "updated\_at": "06/04/2023"      }  } |
| **DELETE** http://localhost/api/giatritinhtrangs/id | |

### **1.9. Loại giá trị đo**

Bảng 31. Request, Response Loại giá trị đo

|  |
| --- |
| **Request** |
| **GET** <http://localhost>/api/loaigiatridos |
| **Response** |
| **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách loại giá trị đo",      "data": [          {              "id": 1,              "loaigiatrido\_ten": "chiều dài lá",              "loaigiatrido\_donvi": "cm",              "created\_at": "09/04/2023",              "updated\_at": "09/04/2023"          }      ]  } |

### **1.10. Loại sâu bệnh**

Bảng 32. Request, Response Loại sâu bệnh

|  |
| --- |
| **Request** |
| **GET** <http://localhost>/api/loaisaubenhs |
| **Response** |
| **Lấy danh sách thành công**: (200 OK)  {      "status": true,      "message": "Danh sách loại sâu bệnh",      "data": [          {              "id": 1,              "loaisaubenh\_ten": "Rầy nâu",              "loaisaubenh\_donvi": "con",              "loaisaubenh\_mota": "Đây là bệnh do rầy nâu gây ra",              "loaisaubenh\_hinhanh": "images/ray-nau.jpeg",              "created\_at": "14/04/2023",              "updated\_at": "14/04/2023"          }]} |

## 2. CÀI ĐẶT HỆ THỐNG

**Bước 1**: Truy cập vào trang <https://docs.flutter.dev/get_started/install/windows>, chọn vào tải Flutter SDK về máy

Graphical user interface, text, application

Description automatically generated

Hình 50. Tải file SDK Flutter

**Bước 2**: Tạo Thư mục (src) trong ổ C:, sau đó tiếp tục tạo Thư mục (flutter) trong Thư mục (src), giải nén file SDK vừa tải về vào Thư mục (flutter)

**Bước 3**: Tại Start Search Bar trên thanh Taskbar, gõ từ khóa “env” rồi chọn vào **Edit environment variables for your account**.

Hộp thoại System Properties hiện ra, tại tab Advanced chọn vào Environment Variables…

Graphical user interface, text, application, email

Description automatically generated

Tiếp theo hộp thoại hiện ra, chọn vào Path trong System variable như hình và nhấn Edit..

Graphical user interface, text, application, email

Description automatically generated

Tại Hộp thoại Edit environment variables thêm đường dẫn C:\src\flutter\bin rồi nhấn OK

Graphical user interface, text, application

Description automatically generated

**Bước 4**: Cài đặt Android Studio

Truy cập trang <https://developer.android.com/studio> để tiến hành tải và cài đặt phiên bản mới nhất của Android Studio

**Bước 5**: Cài đặt máy Mobile ảo bằng Android Studio

Sau khi tải và cài đặt Android Studio, mở Android Studio lên, chọn **Device Manager** icon, chọn **Create Device**

Chọn kích thước thiết bị Mobile, sau đó nhấn Next

Graphical user interface

Description automatically generated

Chọn System Image cho phiên bản di động, nhấn Next

Graphical user interface, application

Description automatically generated

Đặt tên cho thiết bị và nhấn Finish

A screenshot of a computer

Description automatically generated with medium confidence

**Bước 6**: Cài đặt Visual Code (VS Code)

Cài đặt Visual Code tại được dẫn: <https://code.visualstudio.com/>

**Bước 7**: Cài đặt các Extensions để lập trình flutter được trên VS Code

A screenshot of a computer

Description automatically generated

**Bước 8**: Kiểm tra các công cụ đã được cài đặt để lập trình flutter bằng **flutter doctoc**

Tại Start Search Bar trên thanh Taskbar, gõ từ khóa “env” rồi chọn vào **Command Prompt** hoặc nhấn tổ hợp phím Window + R, gõ cmd

Tại màn hình Terminal, gõ lệnh **flutter doctor**, kết quả hiển thị như hình bên dưới là đã có thể lập trình được flutter

Text

Description automatically generated

**Bước 9:** chạy Chương trình Flutter

Mở Chương trình cần thực hiện chạy trên VSCode

Chọn vào góc dưới bên phải của VSCode



Trên thành tìm kiếm của VSCode sẽ hiển thị các thiết bị máy ảo để chọn và chạy máy ảo.

A screenshot of a computer

Description automatically generated with medium confidence

Chọn Run -> Start Debugging hoặc nhấn phím F5 để chạy chương trình trên máy ảo

# TÀI LIỆU THAM KHẢO

[1] Andy Rubin, Rich Miner, Nick Sears, and Chris White, “Android version history,” Wikipedia. 2003. https://en.wikipedia.org/w/index.php?title=Android\_version\_history&oldid=1150178037

[2] S. Anders Thorhauge, “Flutter History,” Wikipedia. 2017. https://en.wikipedia.org/w/index.php?title=Flutter\_(software)&oldid=1150454609

[3] flutter dev, “State management Flutter,” 2007. https://docs.flutter.dev/development/data-and-backend/state-mgmt/intro

[4] S. Anders Thorhauge, “Flutter tutorial,” 2017. https://docs.flutter.dev/

[5] Lars Bak, “Dart tutorial,” 2011. https://dart.dev/guides/index

[6] amazon, “API tutorial,” Amazon Web Services, Inc., 2015. https://aws.amazon.com/vi/what-is/api/

[7] Hiếu, “Web API là gì? Tính năng và cách hoạt động của Web API,” 2021. https://bizflycloud.vn/tin-tuc/web-api-la-gi-tinh-nang-va-cach-hoat-dong-cua-web-api-20210705180908777.htm

[8] B. SMITH, Beginning JSON. Apress, 2015.

[9] J. Deacon, “Model-View-Controller (MVC) Architecture,” 2009.

[10] Nguyễn Thị Diệu Hiền , Nguyễn Văn Vượng, Nguyễn Thị Ngọc , Hoàng Thị Mai , Dương Văn Quân, “ĐÁNH GIÁ CÁC TÍNH TRẠNG HÌNH THÁI NÔNG HỌC CỦA MỘT SỐ GIỐNG LÚA THUẦN CHẤT LƯỢNG”, 2007

[11] Trung tâm Tài nguyên di truyền Viện nghiên cứu lúa Quốc tế, “1996. IRRI. Hệ thống tiêu chuẩn đánh giá cây lúa”, 1996

[12] 1099 Manila, “2013 IRRI – INTERNATIONAL RICE RESEARCH INSTITUTE”, 2013

[13] Nhóm sinh viên khoa CNTT, Học viện Nông nghiệp Việt Nam, “XÂY DỰNG CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG TRA CỨU CÁC GIỐNG LÚA PHỔ BIẾN Ở VIỆT NAM”, 2015.

[14] Nhóm sinh viên khoa CNTT, Học viện Nông nghiệp Việt Nam, “XÂY DỰNG CƠ SỞ DỮ LIỆU VÀ MODULE TRA CỨU HÌNH ẢNH NẢY MẦM CỦA MỘT SỐ GIỐNG LÚA PHỔ BIẾN Ở VIỆT NAM”, 2018.