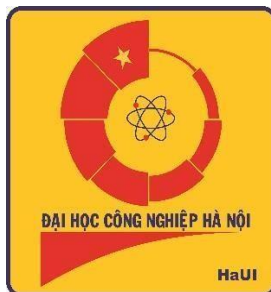


BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH

XÂY DỰNG ỨNG DỤNG DI ĐỘNG QUẢN LÝ BUG
CHO CÔNG TY KPI TRÊN NỀN TẢNG ANDROID
SỬ DỤNG NGÔN NGỮ LẬP TRÌNH JAVA

CBHD: TS. Nguyễn Thị Hoa Huệ

Sinh viên: Đoàn Thị Cẩm Uyên

Mã số sinh viên: 2020605390

Hà Nội – Năm 2024

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU GIAO ĐỀ TÀI ĐỒ ÁN/ KHÓA LUẬN TỐT NGHIỆP

Họ tên sinh viên: Đoàn Thị Cẩm Uyên Mã SV: 2020605390

Lớp: 2020DHKHT02 Ngành: Khoa học máy tính Khóa: K15 (2020-2024)

Tên đề tài: *Xây dựng ứng dụng di động quản lý bug cho công ty KPI trên nền tảng Android sử dụng ngôn ngữ lập trình Java*

Mục tiêu đề tài:

- Sử dụng ngôn ngữ Java để xây dựng ứng dụng theo mô hình hướng đối tượng, cơ sở dữ liệu được quản lý tập trung bởi Firebase.
- Đặt ra bài toán. Phân tích thiết kế và thiết kế hệ thống theo hướng đối tượng.
- Xây dựng giao diện người dùng và phát triển một số chức năng như: Quản lý các thành viên trong đội nhóm, quản lý bug theo dự án, theo dõi và phân công thực hiện xử lý bug. Thống kê kết quả thực hiện xử lý bug theo dự án, theo người xử lý.
- Biết cách lập kế hoạch, thực hiện và báo cáo kiểm thử sản phẩm.

Kết quả dự kiến:

- Xác định được yêu cầu bài toán và phạm vi của ứng dụng
- Phân tích thiết kế đầy đủ những tính năng cơ bản của hệ thống
- Hoàn thành giao diện và cài đặt được các chức năng và yêu cầu đã đặt ra.
- Báo cáo tổng hợp của đề tài.
- Thuyết minh và đĩa CD đính kèm.
- Mã nguồn được quản lý với Git và lưu trữ trên GitHub.
- Kiểm thử sản phẩm và báo cáo kết quả kiểm thử.

Thời gian thực hiện: từ 18/03/2024 đến 18/05/2024

NGƯỜI HƯỚNG DẪN

(Ký và ghi rõ họ tên)

Nguyễn Thị Hoa Huệ

TRƯỞNG ĐƠN VỊ

(Ký và ghi rõ họ tên)

Đặng Trọng Hợp

LỜI CẢM ƠN

Đồ án tốt nghiệp chuyên ngành Khoa học máy tính với đề tài “*Xây dựng ứng dụng di động quản lý bug cho công ty KPI trên nền tảng Android sử dụng ngôn ngữ lập trình Java*” là kết quả sau hai tháng nghiên cứu và tìm hiểu của cá nhân em dưới sự hướng dẫn tận tình của cô giáo TS. Nguyễn Thị Hoa Huệ.

Em xin gửi lời cảm ơn chân thành đến toàn thể các thầy cô giáo hiện đã và đang công tác tại Trường Đại học Công Nghiệp Hà Nội đã đồng hành cùng em trong 4 năm học, truyền đạt cho em những kiến thức bổ ích để em có nền tảng lý thuyết vững chắc và dạy em những bài học cuộc sống thực tế ý nghĩa.

Đồng thời, em cũng xin gửi lời tri ân sâu sắc nhất đến cô TS. Nguyễn Thị Hoa Huệ – người đã đồng hành cùng em trong quá trình thực hiện đồ án này. Em xin chân thành cảm ơn những lời góp ý, động viên bổ ích của cô giúp em hoàn thành đề tài của mình một cách tốt nhất.

Vì thời gian còn hạn chế em chưa có cơ hội cụ thể hóa hết các ý tưởng của mình với đề tài này, em rất mong sẽ được quý thầy cô và các bạn đón nhận cũng như đưa ra những lời góp ý để đề tài của em hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

MỤC LỤC.....	iii
DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT ...	vi
DANH MỤC HÌNH ẢNH	vii
DANH MỤC BẢNG BIỂU	viii
MỞ ĐẦU.....	1
1. Lý do chọn đề tài.....	1
2. Đối tượng, mục tiêu và phạm vi nghiên cứu.....	1
3. Kết quả dự kiến	2
4. Bố cục đề tài.....	2
CHƯƠNG 1. TỔNG QUAN HỆ THỐNG QUẢN LÝ BUG.....	3
1.1 Giới thiệu chung.....	3
1.2 Tầm quan trọng của quản lý bug.....	4
1.3 Các công nghệ sử dụng	6
1.3.1 Java.....	6
1.3.2 Android Studio	8
1.3.3 Firebase	9
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13
2.1 Khảo sát bài toán.....	13
2.1.1 Khảo sát thực trạng	13
2.1.2 Yêu cầu người dùng	14
2.1.3 Mục tiêu khảo sát	14
2.2 Giới thiệu về ứng dụng.....	15
2.2.1 Yêu cầu phần cứng:.....	15
2.2.2 Hoạt động của hệ thống:	15

2.2.3	Mục tiêu hoạt động:.....	16
2.3	Yêu cầu phi chức năng.....	16
2.4	Yêu cầu chức năng.....	16
2.5	Biểu đồ use case.....	17
2.5.1	Các tác nhân của hệ thống.....	17
2.5.2	Biểu đồ use case tổng quát.....	17
2.5.3	Biểu đồ usecase chi tiết.....	19
2.6	Đặc tả chi tiết use case	20
2.6.1	Use case Quên mật khẩu	20
2.6.2	Use case Đăng nhập	21
2.6.3	Use case Xem project.....	22
2.6.4	Use case Xem bug.....	23
2.6.5	Use case Quản lý project.....	24
2.6.6	Use case Quản lý thông tin cá nhân	26
2.6.7	Use case Xem công việc	27
2.6.8	Use case Quản lý bug.....	28
2.6.9	Use case Bình luận	30
2.7	Thiết kế chi tiết.....	31
2.7.1	Biểu đồ lớp	31
2.7.2	Biểu đồ tuần tự của UC Quản lý project.....	32
2.7.3	Biểu đồ tuần tự của UC Quản lý bug	33
2.8	Thiết kế cơ sở dữ liệu.....	34
2.8.1	Thiết kế bảng.....	34
2.8.2	Cấu trúc dữ liệu trên Firebase	36
CHƯƠNG 3. KẾT QUẢ VÀ KIỂM THỬ.....		39
3.1	Một số kết quả đạt được.....	39

3.1.1	Giao diện Client	39
3.1.2	Giao diện Quản lý	45
3.2	Kiểm thử phần mềm.....	47
3.2.1	Kế hoạch kiểm thử	47
3.2.2	Kiểm thử chức năng phía người dùng.....	48
3.2.3	Kiểm thử chức năng phía Admin	50
3.2.4	Kết quả kiểm thử	51
KẾT LUẬN		52
TÀI LIỆU THAM KHẢO.....		54

DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

Ký hiệu	Giải thích
UC	Use Case
UI	Giao diện người dùng (User Interface)
CPU	Bộ xử lý trung tâm (Central Processing Unit)
Dev	Lập trình viên (Developer)
CD	Đĩa Compact (Compact Disc)
Git	Hệ thống quản lý phiên bản phân tán
Github	Nền tảng lưu trữ mã nguồn và hợp tác phát triển phần mềm
Firebase	Nền tảng phát triển ứng dụng di động và web của Google
Java	Ngôn ngữ lập trình hướng đối tượng
Android Studio	Môi trường phát triển tích hợp (IDE) cho Android
UML	Ngôn ngữ mô hình hoá thống nhất (Unified Modeling Language)
JSON	Định dạng dữ liệu JavaScript Object Notation
HTTP	Giao thức truyền tải siêu văn bản (Hypertext Transfer Protocol)
VOPC	Biểu đồ lớp phân tích (View Of Participating Classes)
Basic Flow	Biểu đồ trình tự
CSDL	Cơ sở dữ liệu

DANH MỤC HÌNH ẢNH

Hình 2. 1 Mô hình của ứng dụng ..	Lỗi! Thẻ đánh dấu không được xác định.
Hình 2. 2 Use case tổng quát.....	17
Hình 2. 3 Use case phía admin.....	19
Hình 2. 4 Use case phía tester	19
Hình 2. 5 Biểu đồ lớp phân tích	31
Hình 2. 6 Biểu đồ tuần tự của UC Quản lý project.....	32
Hình 2. 7 Biểu đồ tuần tự của UC Quản lý bug	33
Hình 2. 8 Biểu đồ thực thể liên kết	34
Hình 2. 9 Dữ liệu users.....	36
Hình 2. 10 Dữ liệu projects	36
Hình 2. 11 Dữ liệu bugs	37
Hình 2. 12 Dữ liệu devices.....	37
Hình 2. 13 Dữ liệu comments	38
Hình 3. 1 Màn hình đăng nhập	39
Hình 3. 2 Màn hình quên mật khẩu.....	40
Hình 3. 3 Màn hình trang chủ	40
Hình 3. 4 Màn hình công việc	41
Hình 3. 5 Màn hình mọi người.....	41
Hình 3. 6 Màn hình menu Drawer	42
Hình 3. 7 Màn hình thêm mới bug.....	42
Hình 3. 8 Màn hình profile.....	43
Hình 3. 9 Màn hình chi tiết project	43
Hình 3. 10 Menu bộ lọc bug.....	44
Hình 3. 11 Màn hình sửa bug.....	44
Hình 3. 12 Màn hình chi tiết bug	45
Hình 3. 13 Màn hình thêm mới project.....	45
Hình 3. 14 Màn hình sửa project.....	46
Hình 3. 15 Màn hình xác nhận xóa project.....	46
Hình 3. 16 Màn hình tạo tài khoản người dùng	47

DANH MỤC BẢNG BIỂU

Bảng 2. 1 Đặc tả use case Quên mật khẩu	20
Bảng 2. 2 Đặc tả use case Đăng nhập	21
Bảng 2. 3 Đặc tả use case Xem project.....	22
Bảng 2. 4 Đặc tả use case Xem bug	23
Bảng 2. 5 Đặc tả usecase Quản lý project.....	24
Bảng 2. 6 Đặc tả usecase Quản lý thông tin cá nhân	26
Bảng 2. 7 Đặc tả usecase Xem công việc	27
Bảng 2. 8 Đặc tả usecase Quản lý bug.....	28
Bảng 2. 9 Đặc tả usecase Bình luận	30
Bảng 2. 10 Bảng Users.....	34
Bảng 2. 11 Bảng Project.....	35
Bảng 2. 12 Bảng Bugs.....	35
Bảng 2. 13 Bảng Comments.....	36

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại thông tin số ngày nay, phần mềm đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày và hoạt động kinh doanh. Tuy nhiên, quá trình phát triển phần mềm không tránh khỏi những sai sót dẫn đến các lỗi phát sinh, ảnh hưởng trực tiếp đến hiệu quả và chất lượng của sản phẩm cuối cùng. Nhận thức được tầm quan trọng của việc quản lý và khắc phục lỗi trong phát triển phần mềm, đề tài "Phát triển hệ thống quản lý bug dựa trên Java cho Công ty KPI" đã được chọn để nghiên cứu và triển khai. Đề tài này không chỉ giúp nâng cao chất lượng sản phẩm mà còn góp phần cải thiện quy trình làm việc và tăng cường sự hài lòng của Người dùng.

2. Đối tượng, mục tiêu và phạm vi nghiên cứu

Đối tượng:

- Các lỗi phần mềm xảy ra trong quá trình phát triển các sản phẩm công nghệ tại công ty KPI.

- Nhóm phát triển phần mềm: Lập trình viên, kiểm thử viên và những người liên quan trực tiếp đến quá trình phát triển ứng dụng

- Quản lý dự án: Người có trách nhiệm quản lý quy trình phát triển phần mềm, sử dụng ứng dụng để theo dõi tiến độ giải quyết bug.

- Người dùng cuối: Sử dụng ứng dụng di động, báo cáo lỗi và nhận phản hồi về trạng thái giải quyết bug

Phạm vi đề tài:

Phạm vi của đề tài tập trung vào việc phát triển một ứng dụng di động chuyên dụng để quản lý bug trong quy trình phát triển phần mềm của công ty KPI trên nền tảng Android, sử dụng ngôn ngữ lập trình Java. Ứng dụng sẽ giúp theo dõi, báo cáo và giải quyết các bug được phát hiện trong quá trình phát triển ứng dụng di động của công ty. Đặc biệt, nó sẽ hỗ trợ người dùng cuối báo cáo vấn đề hoặc lỗi họ gặp phải khi sử dụng ứng dụng, cung cấp giao diện đơn giản và dễ sử dụng. Ứng dụng này sẽ được tích hợp với các công cụ quản lý dự án

Agile như Scrum, Kanban để tự động hóa quá trình báo cáo và giải quyết bug, đồng thời hỗ trợ quản lý quyền truy cập dựa trên vai trò của người dùng. Đảm bảo tính bảo mật của dữ liệu bug và đưa ra các biện pháp bảo dưỡng để đảm bảo ứng dụng luôn hoạt động ổn định và hiệu quả sau khi triển khai.

Mục tiêu:

- Mục tiêu chính của đề tài là thiết kế và phát triển một hệ thống quản lý bug hiệu quả, dễ sử dụng và có thể tích hợp một cách mượt mà với các công cụ và quy trình hiện có tại công ty KPI. Hệ thống này sẽ giúp theo dõi, phân loại, và quản lý các lỗi phần mềm một cách hiệu quả, từ đó giảm thiểu ảnh hưởng tiêu cực của các lỗi đến sản phẩm cuối cùng và thời gian đưa sản phẩm ra thị trường.

3. Kết quả dự kiến

- Xác định được yêu cầu bài toán và phạm vi của ứng dụng
- Phân tích thiết kế đầy đủ những tính năng cơ bản của hệ thống
- Hoàn thành giao diện và cài đặt được các chức năng và yêu cầu đã đặt ra.
- Báo cáo tổng hợp của đề tài.
- Thuyết minh và đĩa CD đính kèm.
- Mã nguồn được quản lý với Git và lưu trữ trên GitHub.
- Kiểm thử sản phẩm và báo cáo kết quả kiểm thử.

4. Bố cục đề tài

Trong báo cáo đồ án này, cấu trúc của đề tài được chia thành 3 chương:

- Chương 1: Tổng quan về nội dung nghiên cứu
- Chương 2: Phân tích và thiết kế hệ thống
- Chương 3: Kết quả và kiểm thử

CHƯƠNG 1. TỔNG QUAN HỆ THỐNG QUẢN LÝ BUG

Chương 1 sẽ trình bày tổng quan về dự án xây dựng ứng dụng di động quản lý bug cho công ty KPI. Các nội dung được đưa ra để giới thiệu cụ thể, chi tiết về ứng dụng web, người sử dụng, tính năng, lĩnh vực hoạt động của ứng dụng dự định xây dựng.

1.1 Giới thiệu chung

Trong khuôn khổ đề án tốt nghiệp này, tôi đã phát triển một hệ thống quản lý bug dựa trên ngôn ngữ lập trình Java cho công ty KPI. Hệ thống này được thiết kế nhằm mục đích cải thiện quá trình theo dõi, phân loại và quản lý các lỗi phần mềm, đồng thời tăng cường chất lượng sản phẩm và hiệu quả của quy trình phát triển phần mềm tại công ty.

Hệ thống này nhằm phục vụ đội ngũ nhà phát triển và kiểm thử tại công ty KPI, cho phép họ dễ dàng ghi nhận, phân tích và khắc phục các sự cố kỹ thuật trong quá trình phát triển sản phẩm. Sự hỗ trợ từ hệ thống không chỉ tăng cường hiệu quả công việc mà còn thúc đẩy việc duy trì và nâng cao chất lượng sản phẩm.

Các tính năng nổi bật của hệ thống quản lý bug bao gồm:

- + Theo dõi bug: Cung cấp khả năng theo dõi toàn diện từ lúc bug được phát hiện đến khi được giải quyết hoàn toàn.
- + Phân loại bug: Cho phép phân loại bug theo các tiêu chí như mức độ ưu tiên, loại lỗi, và trạng thái hiện tại.
- + Báo cáo và phân tích: Tích hợp chức năng tạo báo cáo chi tiết về tình hình lỗi của các ứng dụng đang phát triển và tiến độ công việc của các thành viên trong nhóm.
- + Giao diện người dùng thân thiện: Thiết kế giao diện đơn giản, dễ sử dụng, giúp người dùng tương tác với hệ thống một cách hiệu quả.

1.2 Tầm quan trọng của quản lý bug

- Định Nghĩa

Quản lý bug (bug tracking) là quá trình theo dõi, ghi nhận, quản lý và giải quyết các lỗi (bug) xuất hiện trong phần mềm trong suốt quá trình phát triển và bảo trì. Bug là các lỗi, khiếm khuyết hoặc các vấn đề trong phần mềm khiến nó hoạt động không như mong đợi hoặc gây ra những trải nghiệm không tốt cho người dùng. Một hệ thống quản lý bug thường bao gồm các công cụ và quy trình để phát hiện, ghi lại, phân loại, theo dõi, giải quyết và báo cáo các lỗi này.

- Quy trình quản lý bug : Quy trình quản lý bug thường bao gồm các bước sau:

+ Ghi nhận bug : Khi một bug được phát hiện, thông tin chi tiết về bug cần được ghi lại trong một hệ thống quản lý bug. Các thông tin này bao gồm:

- Mô tả bug: Mô tả ngắn gọn nhưng chi tiết về vấn đề gặp phải. Cần mô tả rõ ràng để người khác có thể hiểu và tái hiện lại vấn đề.
- Bước tái hiện: Các bước cụ thể cần thực hiện để tái hiện lại bug. Điều này giúp lập trình viên xác định nguyên nhân gốc rễ của vấn đề.
- Môi trường: Thông tin về môi trường mà bug xảy ra, bao gồm hệ điều hành, trình duyệt, thiết bị, phiên bản phần mềm, và bất kỳ cấu hình đặc biệt nào.
- Mức độ ưu tiên: Đánh giá mức độ quan trọng của bug, từ thấp đến cao, giúp xác định thứ tự ưu tiên khi xử lý các bug.
- Người phát hiện bug: Thông tin về người phát hiện và ghi nhận bug, để có thể liên hệ nếu cần thêm thông tin chi tiết.

+ Phân loại và ưu tiên : Bug sau khi được ghi nhận sẽ được phân loại và ưu tiên dựa trên mức độ ảnh hưởng đến phần mềm và người dùng. Các mức độ ưu tiên thường bao gồm:

- Low: Bug không ảnh hưởng nhiều đến chức năng chính của phần mềm, có thể gây khó chịu nhưng không ngăn cản người dùng sử dụng phần mềm.
- Medium: Bug ảnh hưởng đến một số chức năng nhưng không gây gián đoạn nghiêm trọng. Người dùng vẫn có thể sử dụng phần mềm, nhưng với một số hạn chế.

- High: Bug gây gián đoạn nghiêm trọng, ảnh hưởng lớn đến người dùng. Một số chức năng quan trọng không hoạt động đúng cách.
- Critical: Bug gây sập hệ thống hoặc làm cho phần mềm không thể sử dụng được. Cần khắc phục ngay lập tức để tránh gây thiệt hại lớn.

+ Gán và theo dõi

Bug sẽ được gán cho các thành viên trong đội phát triển để khắc phục. Quá trình theo dõi bao gồm việc cập nhật trạng thái của bug qua các giai đoạn như:

- Open: Bug mới được ghi nhận và chưa được xử lý.
- In Progress: Bug đang được xử lý bởi một lập trình viên.
- Resolved: Bug đã được khắc phục nhưng chưa được kiểm tra lại bởi đội kiểm thử.
- Verified: Bug đã được khắc phục và kiểm tra lại, xác nhận là không còn tồn tại.
- Closed: Bug đã được khắc phục hoàn toàn và đóng lại.

+ Kiểm tra và đóng bug

Sau khi một bug được khắc phục, cần phải kiểm tra lại để đảm bảo rằng bug đã được giải quyết hoàn toàn. Đội kiểm thử sẽ tái hiện lại các bước để đảm bảo rằng bug không còn tồn tại và phần mềm hoạt động đúng như mong đợi. Nếu bug không còn tồn tại, nó sẽ được đánh dấu là "Closed". Nếu vẫn còn vấn đề, bug sẽ được mở lại và tiếp tục xử lý.

• Lợi ích của quản lý bug

- + Cải thiện chất lượng phần mềm : Quản lý bug hiệu quả giúp xác định và khắc phục các lỗi, cải thiện chất lượng phần mềm. Việc theo dõi và xử lý bug kịp thời giúp giảm thiểu số lượng lỗi tồn đọng và nâng cao sự ổn định của phần mềm.
- + Tăng cường hiệu quả làm việc : Quản lý bug giúp đội ngũ phát triển theo dõi và xử lý các vấn đề một cách có tổ chức và hiệu quả. Công cụ quản lý bug cung cấp các báo cáo và thông tin chi tiết về tiến độ xử lý bug, giúp lập trình viên tập trung vào các vấn đề quan trọng.
- + Nâng cao trải nghiệm người dùng : Việc khắc phục các bug kịp thời giúp đảm bảo rằng người dùng có trải nghiệm tốt nhất khi sử dụng phần

mềm. Người dùng sẽ ít gặp phải các vấn đề kỹ thuật và có thể sử dụng phần mềm một cách trơn tru và hiệu quả hơn.

+ Lập kế hoạch và dự đoán : Dữ liệu từ việc quản lý bug giúp lập kế hoạch tốt hơn cho các giai đoạn phát triển và bảo trì phần mềm. Các báo cáo về số lượng và loại bug giúp dự đoán các vấn đề có thể xảy ra và chuẩn bị các biện pháp khắc phục kịp thời.

- Thách thức trong quản lý bug

+ Phát hiện bug: Việc phát hiện và ghi nhận tất cả các bug có thể rất khó khăn, đặc biệt là trong các hệ thống phức tạp. Một số bug có thể không xuất hiện thường xuyên hoặc chỉ xuất hiện trong các điều kiện cụ thể, làm cho việc phát hiện trở nên khó khăn.

+ Ưu tiên bug: Xác định mức độ ưu tiên chính xác cho mỗi bug có thể gây tranh cãi và đòi hỏi sự đánh giá kỹ lưỡng. Đôi khi, các bug có mức độ ưu tiên thấp có thể gây ra hậu quả nghiêm trọng nếu không được xử lý kịp thời.

+ Quản lý khối lượng lớn bug: Trong các dự án lớn, số lượng bug cần theo dõi và xử lý có thể rất lớn, yêu cầu công cụ và quy trình quản lý hiệu quả. Việc theo dõi và xử lý một lượng lớn bug có thể gây áp lực lớn cho đội ngũ phát triển.

+ Giao tiếp và phối hợp: Quản lý bug hiệu quả đòi hỏi sự giao tiếp và phối hợp tốt giữa các thành viên trong đội ngũ phát triển, kiểm thử và quản lý dự án. Thiếu sự giao tiếp và phối hợp có thể dẫn đến việc bug không được xử lý kịp thời hoặc bị bỏ sót.

1.3 Các công nghệ sử dụng

1.3.1 Java

- Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

- Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun Microsystems năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

- Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun Microsystems.

- Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” .
 Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

- Đặc điểm của ngôn ngữ lập trình Java:

- Hướng đối tượng
- Độc lập phần cứng và hệ điều hành
- Ngôn ngữ thông dịch
- Cơ chế thu gom rác tự động
- Đa luồng
- Tính an toàn và bảo mật

- Ưu điểm của Java:

- Độ tin cậy cao
- Tính đa nền tảng
- Quản lý bộ nhớ tự động
- Công cụ phát triển phong phú
- Hỗ trợ đa luồng

- Nhược điểm của Java:

- Tốc độ chậm hơn so với các ngôn ngữ lập trình như C hoặc C++.
- Java có thể chạy trên nhiều nền tảng khác nhau, những ứng dụng này có thể cần đến một trình biên dịch hoặc máy ảo Java riêng biệt để có thể chạy trên các thiết bị di động.
- Sử dụng bộ nhớ lớn hơn so với một số ngôn ngữ lập trình khác như C hoặc C++.
- Cú pháp phức tạp hơn so với một số ngôn ngữ lập trình: C++, C#, Python, Ruby.

- Java được sử dụng để:

- Phát triển ứng dụng máy tính
- Phát triển ứng dụng di động

- Phát triển game
- Phát triển web
- Phát triển các hệ thống quản lý cơ sở dữ liệu
- Phát triển các ứng dụng IoT
- Phát triển các ứng dụng trí tuệ nhân tạo

1.3.2 Android Studio

Android Studio là môi trường phát triển tích hợp chính thức (IDE) dành cho nền tảng Android, được phát triển và hỗ trợ bởi Google. IDE này được thiết kế đặc biệt để cung cấp một công cụ mạnh mẽ và toàn diện cho các nhà phát triển Android, giúp họ thiết kế, phát triển, và kiểm thử ứng dụng một cách hiệu quả. Android Studio được giới thiệu lần đầu tại hội nghị Google I/O vào năm 2013 và từ đó trở thành công cụ được ưa chuộng nhất trong cộng đồng phát triển Android.

- Giao diện thiết kế trực quan: Android Studio cung cấp một giao diện người dùng đồ họa cho phép kéo và thả các thành phần giao diện người dùng (UI), làm cho quá trình thiết kế UI trở nên trực quan và dễ dàng hơn.

- Hỗ trợ máy ảo Android (Android Emulator): Android Studio tích hợp sẵn Android Emulator giúp các nhà phát triển có thể kiểm thử ứng dụng của họ trên nhiều cấu hình thiết bị và phiên bản Android khác nhau mà không cần phải có thiết bị vật lý.

- Hỗ trợ Kotlin: Từ khi Kotlin trở thành ngôn ngữ lập trình chính thức cho Android vào năm 2017, Android Studio đã cung cấp hỗ trợ toàn diện cho Kotlin, giúp phát triển ứng dụng trở nên nhanh chóng và hiệu quả hơn.

- Tích hợp Gradle: Android Studio sử dụng Gradle làm hệ thống tự động hóa xây dựng, cho phép quản lý thư viện, phụ thuộc và các tác vụ xây dựng một cách dễ dàng.

- Profiler: Android Studio bao gồm các công cụ profiler mạnh mẽ giúp theo dõi hiệu suất ứng dụng theo thời gian thực và xác định các vấn đề về hiệu suất như rò rỉ bộ nhớ và sử dụng CPU.

- Cải tiến code: IDE này cung cấp tính năng phân tích và cải tiến code tự động, giúp nhận diện và sửa chữa các lỗi và vấn đề về mã nguồn.

Android Studio mang lại nhiều ưu điểm cho các nhà phát triển:

- Tối ưu hóa quá trình phát triển: Nhờ vào các công cụ và tính năng tiên tiến, nhà phát triển có thể giảm thiểu thời gian và công sức cần thiết cho việc phát triển ứng dụng.

- Dễ dàng học tập và sử dụng: Giao diện người dùng trực quan và hỗ trợ đầy đủ từ Google làm cho Android Studio trở thành lựa chọn lý tưởng cho cả những người mới bắt đầu và những nhà phát triển chuyên nghiệp.

- Cộng đồng lớn mạnh: Sự ủng hộ mạnh mẽ từ Google và cộng đồng phát triển rộng lớn giúp đảm bảo rằng các nhà phát triển có thể tìm thấy sự hỗ trợ và tài nguyên khi cần thiết.

1.3.3 Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web được Google mua lại vào năm 2014. Nó cung cấp một loạt các dịch vụ và công cụ giúp các nhà phát triển xây dựng, cải thiện và mở rộng ứng dụng một cách nhanh chóng và hiệu quả. Firebase được thiết kế để giúp giảm thiểu các rào cản phát triển thường gặp và tăng tốc độ đưa sản phẩm ra thị trường thông qua việc cung cấp các giải pháp backend, analytics, và hỗ trợ quảng cáo và marketing tự động.

Các dịch vụ chính của Firebase

- Authentication (Xác thực)
 - Firebase Authentication giúp các nhà phát triển quản lý và xác thực người dùng một cách dễ dàng và an toàn. Nó hỗ trợ nhiều phương thức đăng nhập:
 - Email/Password: Cho phép người dùng đăng nhập bằng email và mật khẩu.
 - Số điện thoại: Sử dụng OTP (One-Time Password) gửi qua tin nhắn SMS để xác thực.

- Đăng nhập qua các mạng xã hội: Google, Facebook, Twitter, GitHub.
- Đăng nhập ẩn danh: Cho phép người dùng tương tác với ứng dụng mà không cần phải đăng nhập chính thức.
- Lợi ích:
 - + Tích hợp dễ dàng: Firebase cung cấp SDK và API đơn giản để tích hợp vào ứng dụng.
 - + Bảo mật cao: Firebase Authentication tuân thủ các tiêu chuẩn bảo mật cao nhất.
 - + Quản lý người dùng hiệu quả: Bảng điều khiển Firebase Console giúp theo dõi và quản lý người dùng một cách hiệu quả.
- Realtime Database (Cơ sở dữ liệu thời gian thực)
 - Firebase Realtime Database là một cơ sở dữ liệu NoSQL, lưu trữ và đồng bộ hóa dữ liệu giữa các client theo thời gian thực.
 - Lợi ích:
 - + Đồng bộ hóa thời gian thực: Mọi thay đổi dữ liệu sẽ được cập nhật tức thì trên tất cả các thiết bị.
 - + Offline Capabilities: Hỗ trợ chế độ ngoại tuyến, cho phép ứng dụng hoạt động ngay cả khi không có kết nối internet. Dữ liệu sẽ được đồng bộ hóa lại khi kết nối được khôi phục.
 - + Scalability: Khả năng mở rộng tốt, phù hợp với cả ứng dụng nhỏ lẫn ứng dụng lớn.
- Cloud Firestore
 - Cloud Firestore là một cơ sở dữ liệu NoSQL khác của Firebase, cung cấp khả năng lưu trữ dữ liệu có cấu trúc và không có cấu trúc, hỗ trợ các truy vấn phức tạp và khả năng mở rộng tốt hơn so với Realtime Database.
 - Lợi ích:
 - + Querying: Hỗ trợ truy vấn linh hoạt và mạnh mẽ.

- + Scalability: Được thiết kế để mở rộng dễ dàng, đáp ứng nhu cầu của các ứng dụng lớn.
- + Offline Capabilities: Tương tự như Realtime Database, hỗ trợ chế độ ngoại tuyến.
- **Firestore**
 - Firestore cung cấp giải pháp lưu trữ file trên đám mây, cho phép lưu trữ và quản lý các file như hình ảnh, video, âm thanh, và các file khác.
 - Lợi ích:
 - + Integration: Dễ dàng tích hợp với Firebase Authentication và Firebase Realtime Database.
 - + Security: Sử dụng cơ sở hạ tầng bảo mật của Google Cloud Storage.
 - + Large File Support: Hỗ trợ lưu trữ và truyền tải các file lớn một cách hiệu quả.
- **Firebase Hosting**
 - Firebase Hosting là một dịch vụ hosting tĩnh và động với hiệu suất cao, bảo mật, và có thể mở rộng.
 - Lợi ích:
 - + Fast Deployments: Cho phép triển khai các ứng dụng web một cách nhanh chóng và dễ dàng.
 - + SSL: Tự động cấu hình SSL, đảm bảo tất cả các kết nối đều được mã hóa.
 - + CDN: Tích hợp với CDN của Google, giúp tăng tốc độ tải trang web.
- **Firebase Cloud Messaging (FCM)**
 - Firebase Cloud Messaging cho phép gửi thông báo đẩy (push notifications) và tin nhắn từ server đến các thiết bị người dùng.
 - Lợi ích:
 - + Free: Miễn phí sử dụng.
 - + Cross-Platform: Hỗ trợ gửi tin nhắn đến iOS, Android, và web.

- + Customizable: Tùy chỉnh thông báo và nội dung tin nhắn dễ dàng.
- Firebase Analytics
 - Firebase Analytics cung cấp thông tin chi tiết về cách người dùng tương tác với ứng dụng của bạn.
 - Lợi ích:
 - + Detailed Insights: Theo dõi các sự kiện và hành vi người dùng.
 - + Custom Audiences: Tạo nhóm người dùng tùy chỉnh để phân tích chi tiết hơn.
 - + Integration: Dễ dàng tích hợp với các công cụ khác của Firebase.
- Firebase Crashlytics
 - Firebase Crashlytics giúp theo dõi, phát hiện, và phân tích các lỗi xảy ra trong ứng dụng.
 - Lợi ích:
 - + Real-Time Alerts: Cảnh báo tức thì khi có sự cố.
 - + Detailed Reports: Báo cáo chi tiết về các sự cố, giúp xác định và khắc phục vấn đề nhanh chóng.
 - + Custom Logging: Cho phép thêm các thông tin log tùy chỉnh vào báo cáo sự cố.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Chương 2 sẽ trình bày chi tiết về ứng dụng Quản lý bug: các yêu cầu hệ thống, biểu đồ use case, đặc tả và phân tích các use case, thiết kế cơ sở dữ liệu.

2.1 Khảo sát bài toán

2.1.1 Khảo sát thực trạng

Công ty KPI hiện đang đối mặt với một số thách thức trong quản lý và giải quyết các lỗi phần mềm trong quá trình phát triển sản phẩm. Cụ thể, công ty gặp phải các vấn đề như việc phát hiện và theo dõi lỗi không thống nhất, thiếu các công cụ để phân tích xu hướng lỗi, và sự chậm trễ trong việc giải quyết các lỗi đã được báo cáo. Điều này không chỉ ảnh hưởng đến tiến độ phát triển sản phẩm mà còn gây ra sự không hài lòng của Người dùng do chất lượng phần mềm không ổn định.

Trong ngành công nghệ thông tin, việc quản lý lỗi phần mềm là một trong những yếu tố quan trọng nhất đảm bảo chất lượng sản phẩm. Một hệ thống quản lý bug hiệu quả có thể giúp các công ty:

- Nâng cao chất lượng sản phẩm: Giảm thiểu số lượng và mức độ nghiêm trọng của các lỗi trong phần mềm.
- Tối ưu hóa quy trình làm việc: Tạo điều kiện thuận lợi cho việc phát hiện, báo cáo, và xử lý lỗi một cách nhanh chóng và hiệu quả.
- Cải thiện sự hài lòng của người dùng: Cung cấp sản phẩm và dịch vụ không chỉ đáp ứng mà còn vượt trên mong đợi của người dùng.
- Giảm chi phí phát triển: Phát hiện sớm và xử lý kịp thời các lỗi giúp giảm thiểu chi phí sửa chữa và bảo trì phần mềm.

2.1.2 Yêu cầu người dùng

Mã	Là...,	tôi muốn...,	để....
1	Quản lý	Quản lý các project, bug 1 cách hiệu quả dễ dàng. Tạo mới các bug và project.	Theo dõi tiến độ và ưu tiên các vấn đề
2	Lập trình viên	Xem xét các bug và sửa chữa bug. Cập nhật tiến độ sửa chữa các bug	Giải quyết các bug 1 cách linh hoạt, cập nhật tình hình nhanh chóng về các bug
3	Kiểm thử viên	Tạo mới bug và gán cho lập trình viên.	Quản lý các bug của dự án, kiểm tra tính đúng đắn khi lập trình viên sửa xong

2.1.3 Mục tiêu khảo sát

- Xác định mức độ hiểu biết và sử dụng công cụ quản lý bug hiện tại của nhóm phát triển phần mềm.
- Đo lường các tính năng mà họ thấy thiếu sót trong công cụ hiện tại.
- Tìm hiểu về các yêu cầu cụ thể mà nhóm phát triển phần mềm và quản lý dự án đặt ra cho một ứng dụng quản lý bug mới
- Xác định những tính năng cần thiết và mong đợi từ ứng dụng mới để cải thiện quy trình phát triển phần mềm

- Thu thập phản hồi từ nhóm phát triển và kiểm thử về các vấn đề mà họ gặp phải với công cụ quản lý bug hiện tại
- Xác định các điểm yếu cần cải thiện và các tính năng mà họ mong đợi từ một công cụ mới
- Đo lường mức độ hài lòng với công cụ hiện tại và sự sẵn lòng chuyển đổi sang một ứng dụng mới
- Xác định các yếu tố chính ảnh hưởng đến quyết định chuyển đổi sang ứng dụng mới
- Xác định các yêu cầu kỹ thuật cần thiết cho ứng dụng mới, bao gồm hỗ trợ nền tảng Android, tích hợp công cụ phát triển phần mềm hiện có, và bảo mật dữ liệu
- Ưu tiên tính năng quan trọng nhất cần có trong phiên bản đầu tiên của ứng dụng

2.2 Giới thiệu về ứng dụng

2.2.1 Yêu cầu phần cứng:

- Smartphone chạy hệ điều hành Android 5.0 hoặc cao hơn
- Điện thoại được kết nối internet
- Cài đặt phiên bản mới nhất của ứng dụng quản lý bug

2.2.2 Hoạt động của hệ thống:

Quản lý bug: Người dùng có thể tạo mới các bug khi phát hiện lỗi trong ứng dụng hoặc phát triển mới. Các bug sẽ được gán mức ưu tiên và được theo dõi theo trạng thái của chúng (như đã xác nhận, đang xử lý, đã giải quyết).

Giao việc và theo dõi tiến độ: Quản lý dự án có thể giao việc giải quyết bug cho các thành viên nhóm phát triển. Họ có thể theo dõi tiến độ của các bug được giao và nhận thông báo khi có sự thay đổi trạng thái.

Phản hồi người dùng: Người dùng cuối có thể báo cáo các bug hoặc vấn đề mà họ gặp phải khi sử dụng ứng dụng. Hệ thống sẽ tự động ghi lại phản hồi này và tạo bug tương ứng để nhóm phát triển xử lý.

2.2.3 Mục tiêu hoạt động:

Mục tiêu của hệ thống là cung cấp một nền tảng linh hoạt và hiệu quả để quản lý bug trong quy trình phát triển phần mềm. Điều này giúp giảm thiểu thời gian và công sức cần thiết để xử lý bug, từ đó tăng cường chất lượng và hiệu suất của dự án.

2.3 Yêu cầu phi chức năng

- Ngôn ngữ trên hệ thống sử dụng ngôn ngữ tiếng Việt
- Độ tin cậy cao
- Thời gian phản hồi nhanh
- Tính khả dụng cao
- Đảm bảo an toàn cho dữ liệu bug và thông tin nhạy cảm
- Giao diện đơn giản, dễ sử dụng
- Khả năng kết nối và làm việc cùng các công cụ khác trong quy trình phát triển
- Có thể tùy chỉnh và mở rộng để phù hợp với các yêu cầu cụ thể
- Cung cấp tài liệu hướng dẫn và hỗ trợ trực tuyến để người dùng có thể sử dụng và bảo dưỡng hệ thống.

2.4 Yêu cầu chức năng

Quản lý có các chức năng:

- Đăng nhập
- Quản lý bình luận
- Quản lý tài khoản người dùng
- Quản lý bug
- Quản lý project
- Quản lý công việc

Kiểm thử viên (Tester) và lập trình viên (Dev) có các chức năng:

- Đăng nhập
- Xem project

- Xem bug
- Quản lý bug
- Xem bình luận
- Thêm bình luận
- Xem công việc
- Quản lý thông tin cá nhân

2.5 Biểu đồ use case

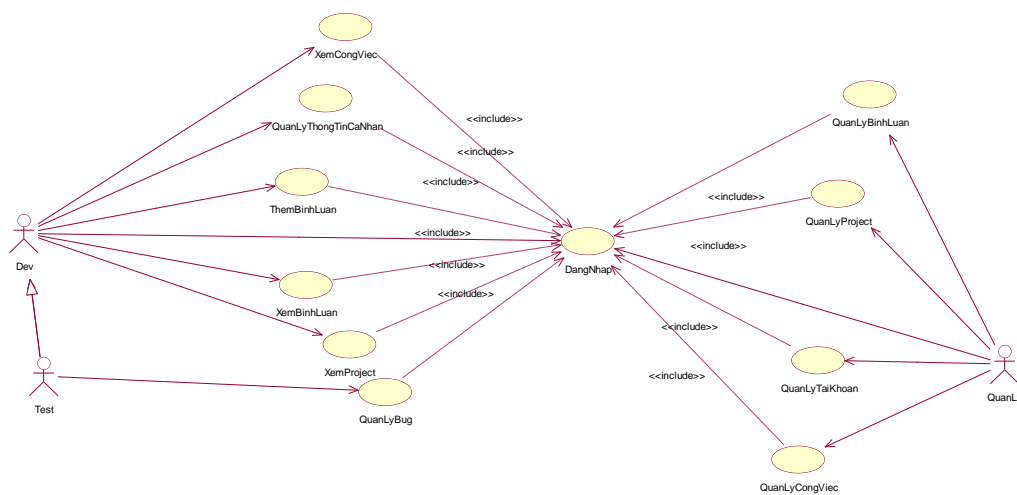
2.5.1 Các tác nhân của hệ thống

Test: là những người truy cập vào ứng dụng Quản lý bug để thực hiện các chức năng của ứng dụng.

Dev: là những người truy cập vào ứng dụng Quản lý bug để thực hiện các chức năng của ứng dụng.

Admin (Người quản trị): thực hiện các nhiệm vụ quản trị, duy trì sự hoạt động của các chức năng hệ thống.

2.5.2 Biểu đồ use case tổng quát



Hình 2. 1 Use case tổng quát

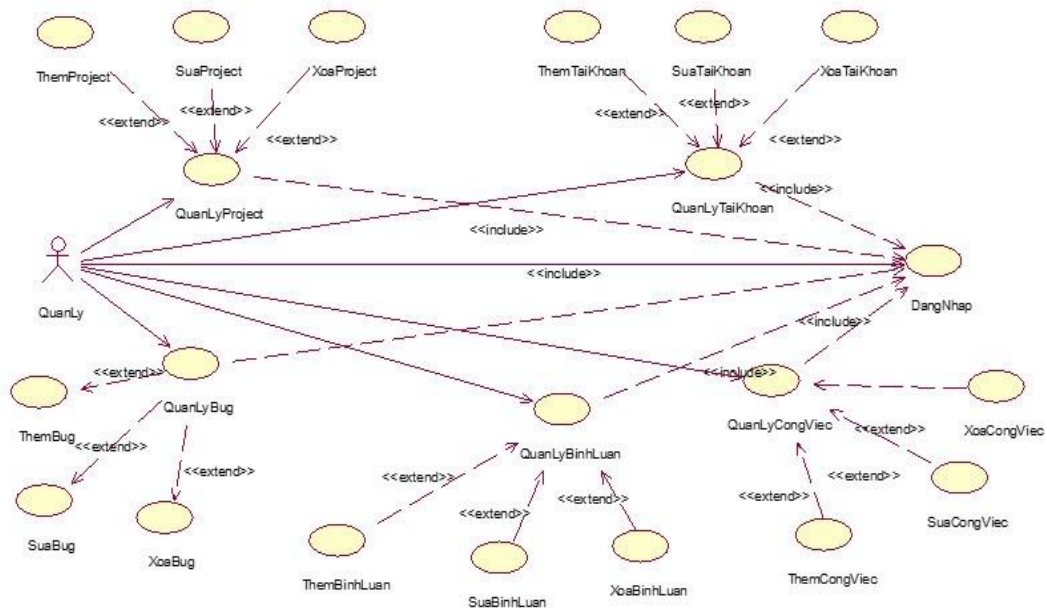
Trong đó:

1. Thêm bình luận: Cho phép quản lý bình luận trong chi tiết bug
2. Xem bình luận: Cho phép người dùng xem bình luận trong chi tiết bug

3. Quản lý bình luận: Cho phép quản lý xem, thêm, sửa, xóa các bình luận
4. Xem project: Cho phép người dùng xem danh sách các project gồm: tên project, thời gian bắt đầu, trạng thái.
5. Quản lý project: cho phép quản lý xem thêm, sửa, xóa project
6. Xem công việc: Cho phép quản lý xem danh sách công việc (project, bug) mình đã tạo hoặc đang theo dõi
7. Quản lý công việc: Cho phép quản lý xem, thêm, sửa, xóa các công việc
8. Xem bug: Cho phép người dùng xem danh sách các bug trong project
9. Quản lý bug: Cho phép quản lý và người dùng xem thêm, sửa, xóa bug
10. Quản lý thông tin cá nhân: cho phép quản lý và người dùng xem, chỉnh sửa thông tin cá nhân.
11. Quản lý tài khoản: Cho phép quản lý xem, thêm, sửa xóa tài khoản người dùng
12. Đăng nhập: cho phép quản lý đăng nhập tài khoản vào hệ thống..

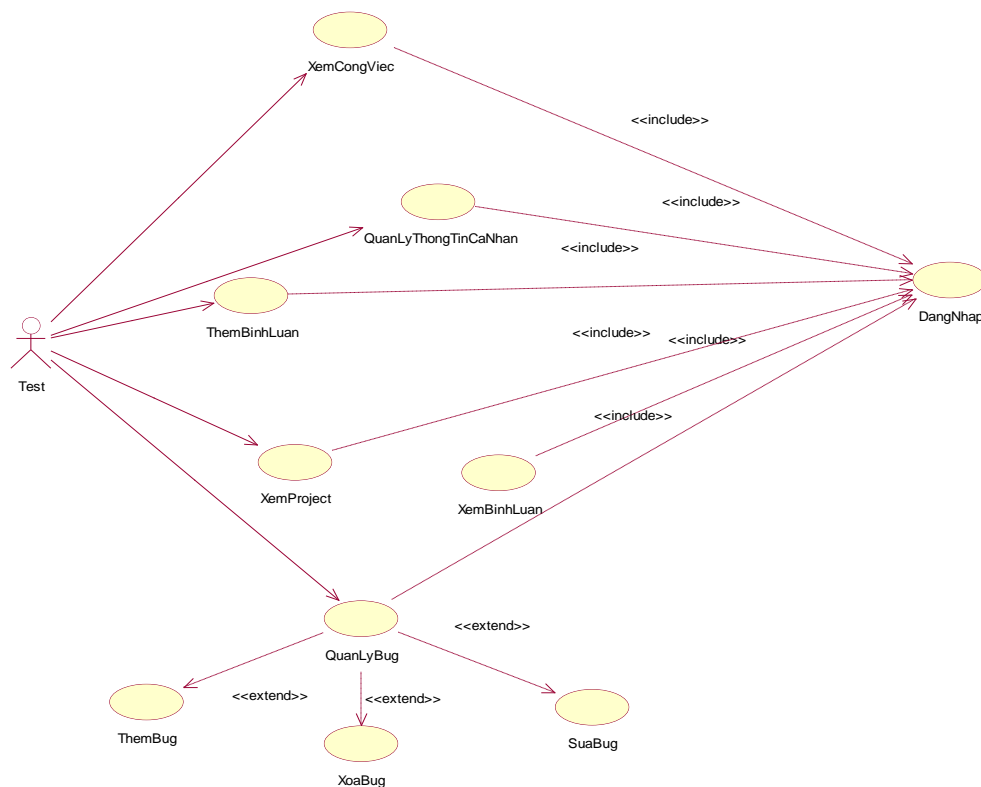
2.5.3 Biểu đồ usecase chi tiết

Biểu đồ Use case phía quản lý:



Hình 2. 2 Use case phía admin

Biểu đồ Use case phía lập trình viên (Dev) và kiểm thử viên (Tester):



Hình 2. 3 Use case phía tester

2.6 Đặc tả chi tiết use case

2.6.1 Use case Quên mật khẩu

Bảng 2. 1 Đặc tả use case Quên mật khẩu

Mã UC: UCFE01	Tên UC: Quên mật khẩu		
Tác nhân	Quản lý, dev, tester		
Mô tả	UC này cho phép quản lý, dev, tester tạo lại mật khẩu khi bị quên		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi quản lý, dev, tester ấn “Forget password” tại màn hình Log in		
Tiền điều kiện	Không		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Người dùng	Click “Forgot password”
	2	Hệ thống	Hiển thị form nhập gmail của tài khoản
	3	Người dùng	Nhập Gmail
	4	Hệ thống	Kiểm tra, tiến hành gửi đường link thay đổi mật khẩu tới gmail vừa điền
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động
	1..4	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì

		hệ thống sẽ thông báo lỗi và UC kết thúc.	
Hậu điều kiện	Không		

2.6.2 Use case Đăng nhập

Bảng 2. 2 Đặc tả use case Đăng nhập

Mã UC: UCFE02	Tên UC: Đăng nhập		
Tác nhân	Quản lý, dev, tester		
Mô tả	UC này cho phép người dùng đăng nhập tài khoản vào hệ thống.		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người dùng chọn nút “Log in” trên màn hình Log in		
Tiền điều kiện	Quản lý đã tạo tài khoản từ trước		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Tác nhân	Click “Đăng nhập” trên menu
	2	Hệ thống	Hiện thị form đăng nhập
	3	Tác nhân	Nhập Gmail, mật khẩu và click vào nút “Log in”.
	4	Hệ thống	Hiện thị màn hình Trang chủ. UC kết thúc
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động
	4a	Hệ thống	Nếu người dùng nhập thông tin bị thiếu hoặc sai định dạng thì hệ

			thống sẽ hiển thị thông báo lỗi. Use case kết thúc.
	1..4	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.3 Use case Xem project

Bảng 2. 3 Đặc tả use case Xem project

Mã UC: UCFE03	Tên UC: Xem project		
Tác nhân	Quản lý, dev, tester		
Mô tả	UC này cho phép quản lý, dev, tester xem chi tiết project.		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi quản lý, dev, tester chọn 1 project bất kỳ.		
Tiền điều kiện	Không		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Quản lý, dev, tester	Khi thực hiện ấn vào 1 project bất kỳ trên trang chủ
	2	Hệ thống	Hiển thị thông tin chi tiết của project bao gồm: Tên, mô tả, danh sách các bug. UC kết thúc
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động

	1..2	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.4 Use case Xem bug

Bảng 2. 4 Đặc tả use case Xem bug

Mã UC: UCFE04	Tên UC: Xem bug		
Tác nhân	Quản lý, dev, tester		
Mô tả	UC này cho phép quản lý, dev, tester xem chi tiết bug		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi quản lý, dev, tester chọn 1 bug		
Tiền điều kiện	Trong màn hình chi tiết project hoặc công việc (nếu có)		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Quản lý, dev, tester	Click vào bug bất kỳ
	2	Hệ thống	Hiển thị thông tin chi tiết của bug gồm: ảnh, mô tả, assignment (phân công), trạng thái, comment (bình luận). UC kết thúc
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động

	2a	Hệ thống	Nếu chưa có comment nào thì hệ thống hiển thị “Chưa có comment nào”
	1..2	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.5 Use case Quản lý project

Bảng 2. 5 Đặc tả usecase Quản lý project

Mã UC: UCFE05	Tên UC: Quản lý project		
Tác nhân	Quản lý		
Mô tả	UC này cho phép quản lý thêm, sửa, xóa project		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi quản lý chọn icon 3 chấm dọc trên mỗi project		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Quản lý	Click icon 3 chấm dọc trên project cần thực hiện
	2	Hệ thống	Hiển thị menu lựa chọn: Sửa hoặc xóa project
	3	Quản lý	Nếu quản lý chọn Sửa thì luồng sự kiện tiếp theo là các bước 4,5,6. Chọn sửa project
	4	Hệ thống	Hiển thị chi tiết project gồm tất cả các thông tin của project

	5	Quản lý	Chỉnh sửa các thông tin phù hợp và ấn “Cập nhật”
	6	Hệ thống	Cập nhật lại thông tin của project
	7	Quản lý	Nếu quản lý chọn Xóa thì luồng sự kiện tiếp theo là các bước 8,9,10. Chọn nút “Xóa project” trong menu
	8	Hệ thống	Hiển thị thông báo xác nhận xóa project
	9	Quản lý	Chọn “Xóa”
	10	Hệ thống	Xóa project đó trong danh sách project.
	11	Quản lý, tester	Nếu quản lý chọn Thêm thì luồng sự kiện tiếp theo là các bước 12, 13, 14. Click “Thêm project” tại menu Drawers
	12	Hệ thống	Hiển thị form để người dùng điền các thông tin của project
	13	Quản lý, tester	Điền đầy đủ thông tin, sau đó ấn “Thêm”
	14	Hệ thống	Thêm và cập nhật lại danh sách project. Use case kết thúc.
Luồng sự kiện rẽ nhánh			
	#	Thực hiện bởi	Hành động
	11	Quản lý, tester	Khi ấn “Thêm” mà chưa điền đầy đủ thông tin, hệ thống thông báo các trường dữ liệu còn thiếu cho người dùng.

	1..9	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.6 Use case Quản lý thông tin cá nhân

Bảng 2. 6 Đặc tả usecase Quản lý thông tin cá nhân

Mã UC: UCFE06	Tên UC: Quản lý thông tin cá nhân		
Tác nhân	Quản lý, dev, tester		
Mô tả	UC này cho phép phép quản lý, dev, test xem, chỉnh sửa thông tin cá nhân.		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người dùng chọn icon ảnh ở góc trên bên phải		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Người dùng	Chọn icon ảnh ở góc trên bên phải, sau đó chọn “Edit profile information”
	2	Hệ thống	Hiển thị thông tin gồm: Tên, chức vụ, số điện thoại, email
	3	Người dùng	Chỉnh sửa thông tin các trường và click nút “Cập nhật”
	4	Hệ thống	Kiểm tra thông tin và cập nhật lại thông tin tài khoản. UC kết thúc
Luồng sự kiện rẽ nhánh			

	#	Thực hiện bởi	Hành động
	1..4	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.7 Use case Xem công việc

Bảng 2. 7 Đặc tả usecase Xem công việc

Mã UC: UCFE07	Tên UC: Xem công việc		
Tác nhân	Quản lý, dev, tester		
Mô tả	UC này cho phép xem danh sách các project (Quản lý) các bug mà mình đảm nhiệm		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi quản lý, dev, tester ấn vào nút thứ 2 trên thanh điều hướng		
Tiền điều kiện	Không có		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Người dùng	Khi quản lý, dev, tester ấn vào nút thứ 2 trên thanh điều hướng
	2	Hệ thống	Hiển thị thông tin về các project, bug mà mình đảm nhiệm hoặc theo dõi (nếu có). UC kết thúc.
Luồng sự kiện rẽ nhánh	#	Thực hiện bởi	Hành động

	1..2	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.8 Use case Quản lý bug

Bảng 2. 8 Đặc tả usecase Quản lý bug

Mã UC: UCFE08	Tên UC: Quản lý bug																	
Tác nhân	Quản lý, dev, tester																	
Mô tả	UC này cho phép quản lý, dev, tester thêm, sửa, xoá bug																	
Sự kiện kích hoạt chức năng	UC này bắt đầu khi Quản lý, dev, tester ấn vào icon 3 chấm dọc trên bug. Khi ấn “Thêm bug” trên menu Drawer																	
Tiền điều kiện	Người dùng đã mua hàng thành công.																	
Luồng sự kiện cơ bản	<table><tr><th>#</th><th>Thực hiện bởi</th><th>Hành động</th></tr><tr><td>1</td><td>Quản lý, dev, tester</td><td>Click icon 3 chấm dọc trên bug</td></tr><tr><td>2</td><td>Hệ thống</td><td>Hiển thị menu lựa chọn Sửa hoặc Xoá bug</td></tr><tr><td>3</td><td>Người dùng</td><td>Nếu người dùng chọn Sửa bug thì luồng sự kiện tiếp theo là các bước 4,5,6. Click “Sửa bug”</td></tr><tr><td>4</td><td>Hệ thống</td><td>Hiển thị thông tin của bug vừa click</td></tr></table>			#	Thực hiện bởi	Hành động	1	Quản lý, dev, tester	Click icon 3 chấm dọc trên bug	2	Hệ thống	Hiển thị menu lựa chọn Sửa hoặc Xoá bug	3	Người dùng	Nếu người dùng chọn Sửa bug thì luồng sự kiện tiếp theo là các bước 4,5,6. Click “Sửa bug”	4	Hệ thống	Hiển thị thông tin của bug vừa click
#	Thực hiện bởi	Hành động																
1	Quản lý, dev, tester	Click icon 3 chấm dọc trên bug																
2	Hệ thống	Hiển thị menu lựa chọn Sửa hoặc Xoá bug																
3	Người dùng	Nếu người dùng chọn Sửa bug thì luồng sự kiện tiếp theo là các bước 4,5,6. Click “Sửa bug”																
4	Hệ thống	Hiển thị thông tin của bug vừa click																

	5	Quản lý, dev, tester	Người dùng điền thông tin phù hợp, sau đó nhấn “Sửa”
	6	Hệ thống	Hệ thống sẽ reset lại dữ liệu và trở về màn hình trang chủ.
	7	Quản lý, tester	Nếu người dùng chọn Xóa bug thì luồng sự kiện tiếp theo là các bước 8, 9, 10. Click “Xóa bug”
	8	Hệ thống	Hiển thị menu xác nhận có xóa bug không
	9	Quản lý, tester	Ấn đồng ý để thực hiện xóa “Huỷ” để huỷ
	10	Hệ thống	Hệ thống sẽ cập nhật lại dữ liệu và trở về trang chủ.
	11	Quản lý, tester	Nếu người dùng chọn Thêm bug thì luồng sự kiện tiếp theo là các bước 12, 13, 14. Khi người dùng click “Thêm bug” tại menu Drawer
	12	Hệ thống	Hệ thống sẽ hiển thị form cho người dùng điền các thông tin
	13	Quản lý, tester	Điền các thông tin phù hợp và ấn “Thêm”
	14	Hệ thống	Hệ thống sẽ thêm mới bug và trở về trang chủ. UC kết thúc
Luồng sự kiện rẽ nhánh			
	#	Thực hiện bởi	Hành động

	13	Hệ thống	Hệ thống sẽ thông báo khi người dùng điền thiếu
	1..10	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

2.6.9 Use case Bình luận

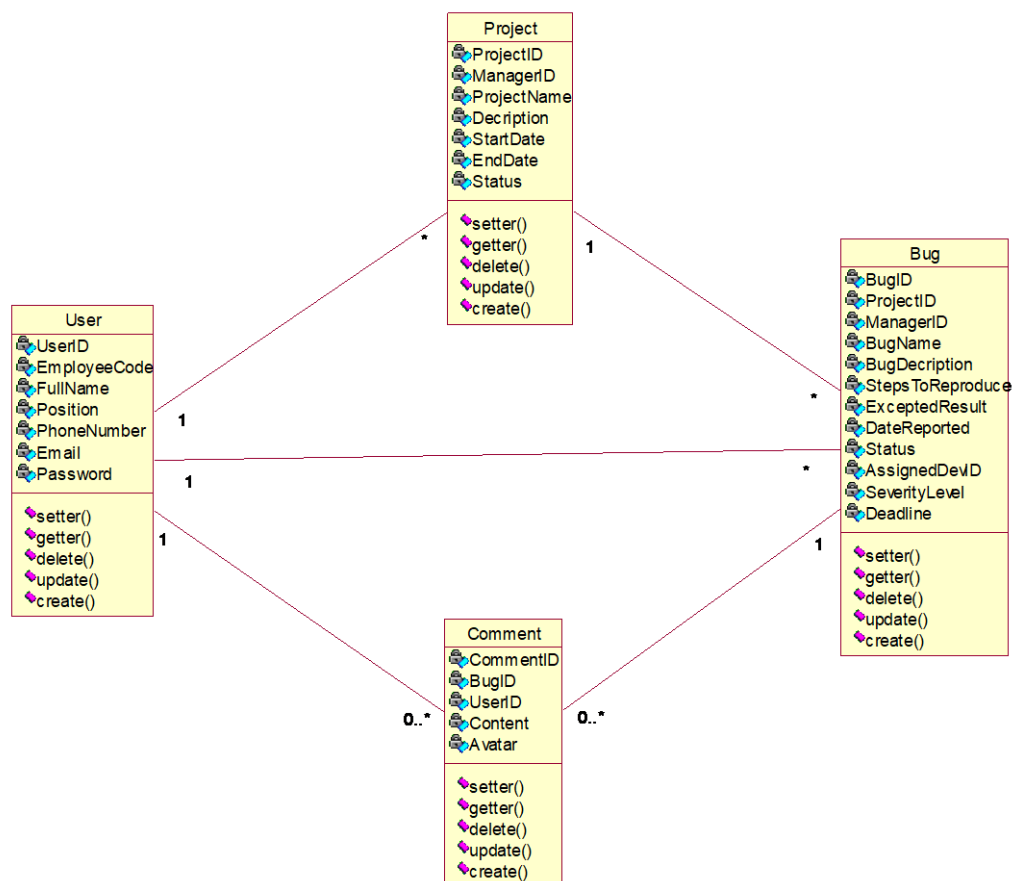
Bảng 2. 9 Đặc tả usecase Bình luận

Mã UC: UCFE09	Tên UC: Bình luận		
Tác nhân	Quản lý, tester, dev		
Mô tả	UC này cho phép người dùng có thể thêm bình luận vào bug và thông báo tới những người liên quan		
Sự kiện kích hoạt chức năng	UC này bắt đầu khi người dùng nhập bình luận và ấn icon “Gửi”		
Tiền điều kiện	Phải ấn vào bug muốn bình luận		
Luồng sự kiện cơ bản	#	Thực hiện bởi	Hành động
	1	Quản lý, dev, tester	Nhập bình luận, ấn gửi
	2	Hệ thống	Cập nhật lên màn hình bình luận vừa rồi ở ngay dưới khu “Comment”. Sẽ thông báo cho dev đang fix. UC kết thúc
Luồng sự kiện rẽ nhánh			

	#	Thực hiện bởi	Hành động
	1..2	Hệ thống	Tại bất kỳ thời điểm nào trong quá trình thực hiện UC nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ thông báo lỗi và UC kết thúc.
Hậu điều kiện	Không		

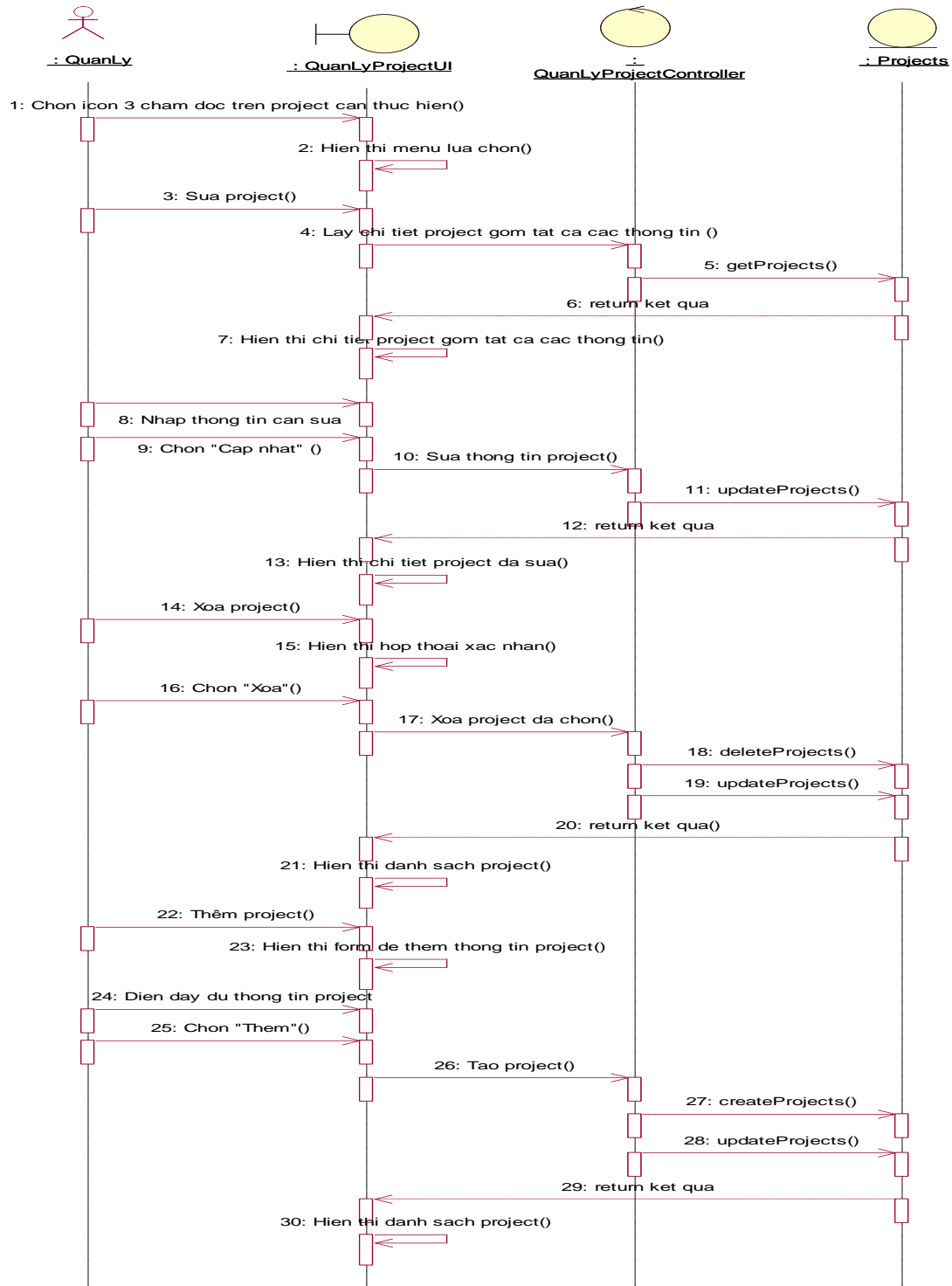
2.7 Thiết kế chi tiết

2.7.1 Biểu đồ lớp



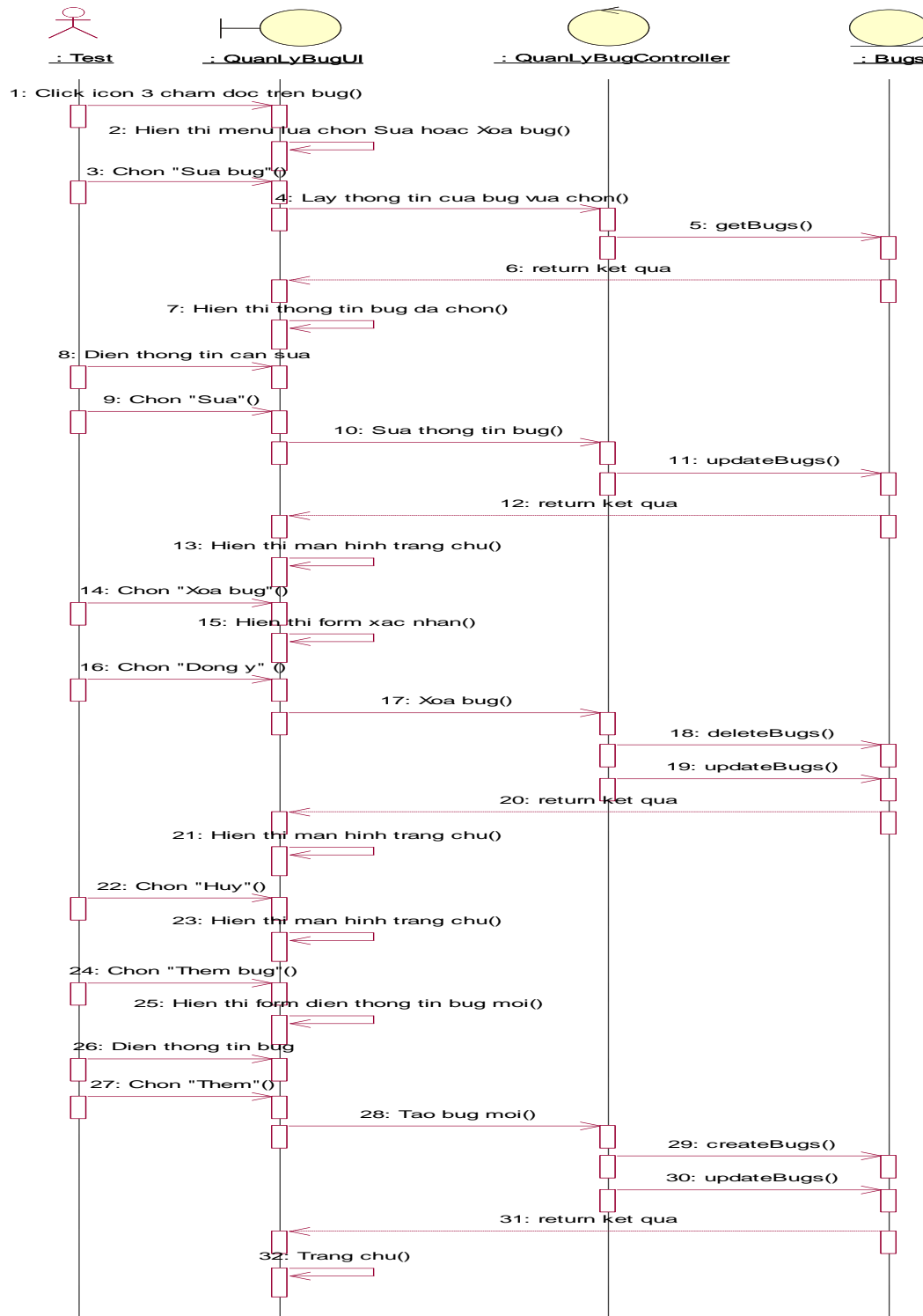
Hình 2. 4 Biểu đồ lớp phân tích

2.7.2 Biểu đồ tuần tự của UC Quản lý project



Hình 2. 5 Biểu đồ tuần tự của UC Quản lý project

2.7.3 Biểu đồ tuần tự của UC Quản lý bug



Hình 2. 6 Biểu đồ tuần tự của UC Quản lý bug

2.8 Thiết kế cơ sở dữ liệu

Mô hình công nghệ ứng dụng:

Server	Firebase
Client	Android
Database	Firebase

Biểu đồ Entity Relationship Diagram



Hình 2. 7 Biểu đồ thực thể liên kết

2.8.1 Thiết kế bảng

Bảng user dùng để lưu trữ thông tin của người dùng

Bảng 2. 10 Bảng Users

Tên trường	Kiểu	Độ dài	Khóa	Giải thích
UserID	varchar	255	PK	ID của người dùng
EmployeeCode	varchar	255		Mã nhân viên
FullName	varchar	255		Họ tên
Position	varchar	255		Chức vụ
PhoneNumber	varchar	255		Số điện thoại
Email	varchar	255		Email

Password	varchar	255		Mật khẩu
----------	---------	-----	--	----------

Bảng project để lưu thông tin các project

Bảng 2. 11 Bảng Project

Tên trường	Kiểu	Độ dài	Khóa	Giải thích
ProjectID	varchar	255	PK	Mã dự án
ManagerID	varchar	255		Mã quản lý
ProjectName	varchar	255		Tên dự án
Description	varchar	255		Mô tả
StartDate	Datetime			Ngày bắt đầu
EndDate	Datetime			Ngày kết thúc
Status	Varchar	255	FK	Trạng thái

Bảng bugs để lưu thông tin các bug trong các dự án

Bảng 2. 12 Bảng Bugs

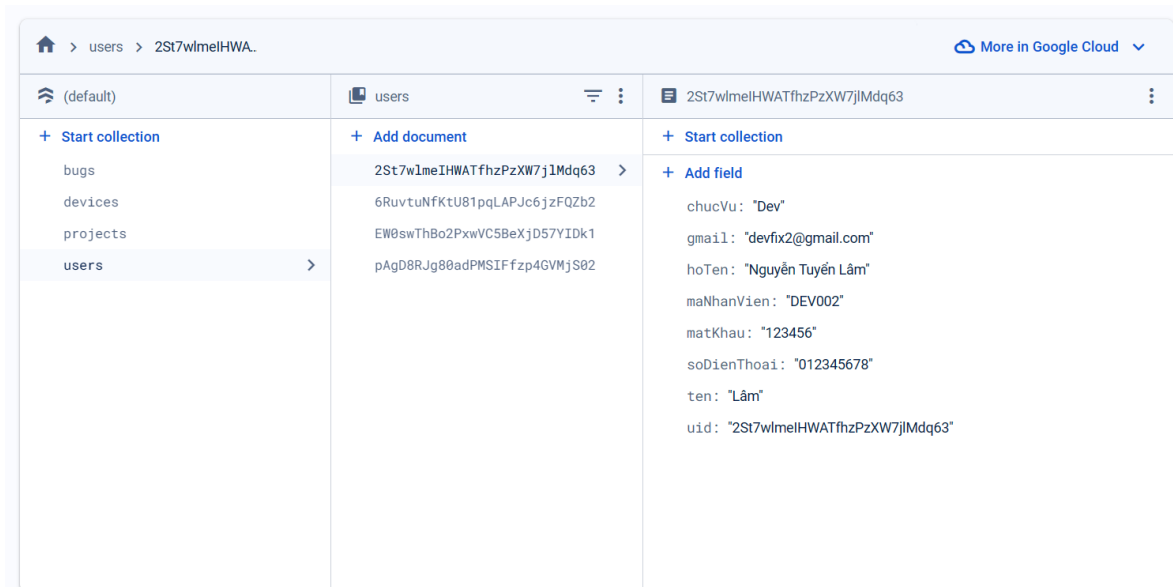
Tên trường	Kiểu	Độ dài	Khóa	Giải thích
BugID	varchar	255	PK	Mã bug
ProjectID	varchar	255	FK	Mã project
ManagerID	varchar	255	FK	Mã quản lý bug
BugName	varchar	255		Tên bug
BugDescription	varchar	255		Mô tả bug
StepsToReproduce	varchar	255		Các bước thực hiện khi lỗi
ExpectedResult	varchar	255		Kết quả mong muốn khi bug được fix
DateReported	datetime			Ngày xuất hiện bug
Status	varchar	255		Trạng thái của bug
AssignedDevID	varchar	255	FK	Mã dev đảm nhận fix
SeverityLevel	varchar	255		Mức độ nghiêm trọng của bug
Deadline	datetime			Thời hạn để dev fix bug

Bảng comments để lưu thông tin comment.

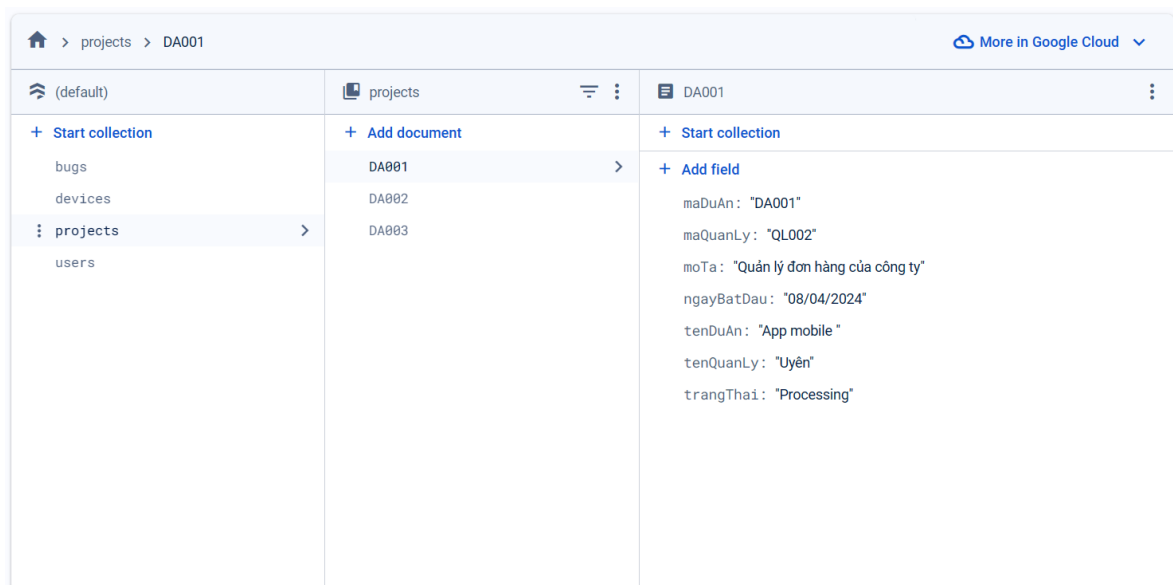
Bảng 2. 13 Bảng Comments

Tên trường	Kiểu	Độ dài	Khóa	Giải thích
CommentID	varchar	255	PK	Mã comment
BugID	varchar	255	FK	Mã bug
UserID	varchar	255	FK	Mã user comment
Content	varchar	255		Nội dung comment
Avartar	varchar	255		Ảnh đại diện của nhân viên tạo comment

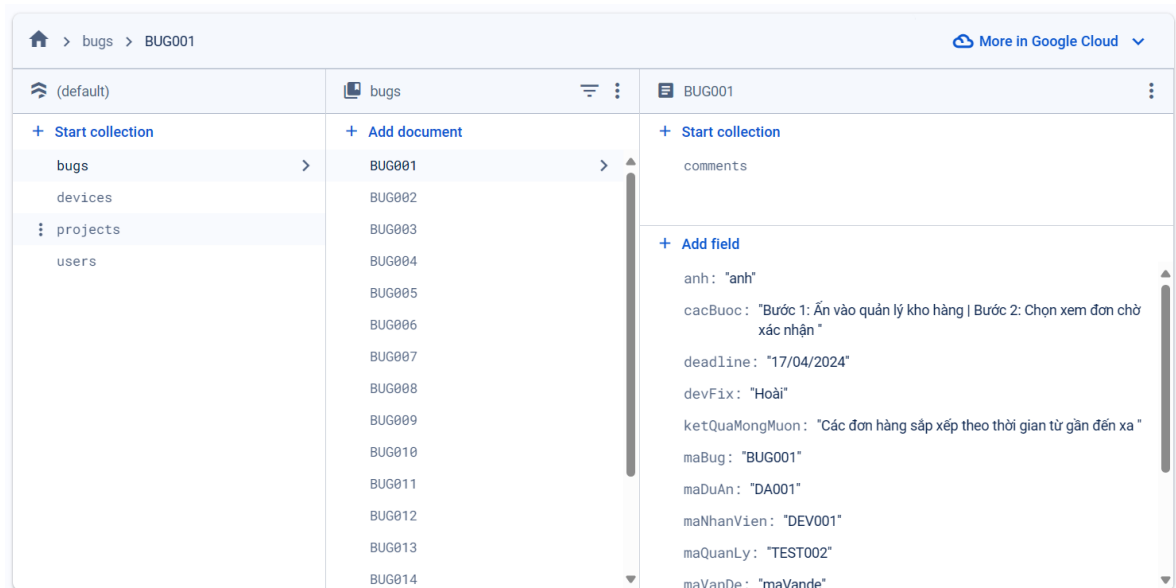
2.8.2 Cấu trúc dữ liệu trên Firebase



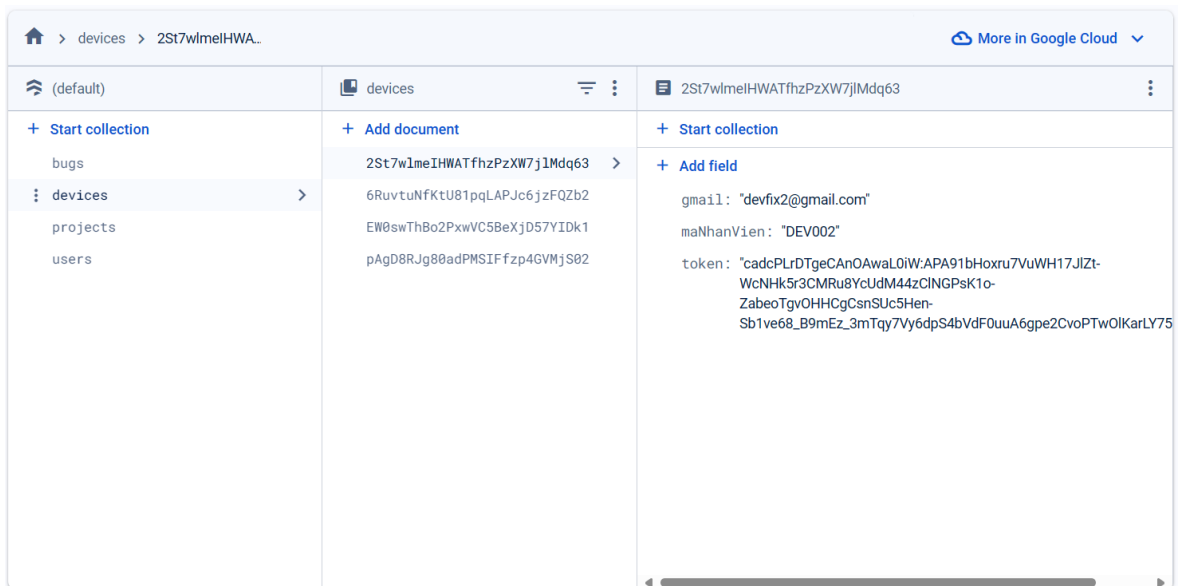
Hình 2. 8 Dữ liệu users



Hình 2. 9 Dữ liệu projects



Hình 2. 10 Dữ liệu bugs



Hình 2. 11 Dữ liệu devices

🏠 > bugs > BUG001 > comments > CMT001 More in Google Cloud		
<div>BUG001</div> <div> + Start collection </div> <div> + Add field </div> <div> anh: "anh" cacBuoc: "Bước 1: Ấn vào quản lý kho hàng Bước 2: Chọn xem đơn chờ xác nhận" deadline: "17/04/2024" devFix: "Hoài" ketQuaMongMuon: "Các đơn hàng sắp xếp theo thời gian từ gần đến xa" maBug: "BUG001" maDuAn: "DA001" </div>	<div>comments</div> <div> + Add document </div> <div> CMT001 CMT002 CMT003 CMT004 CMT005 </div>	<div>CMT001</div> <div> + Start collection </div> <div> + Add field </div> <div> anhDaiDien: "6RuvtuNfKtU81pqLAPJc6jzFQZb2" maNhanVien: "QL001" noiDung: "Reprehenderit ipsum est cupidatat incididunt cillum voluptate in nostrud culpa et exercitation adipisicing. Exercitation dolor magna aute consectetur aliqua laboris excepteur consequat veniam qui in ex labore elit. Cillum veniam deserunt nisi ullamco veniam incididunt minim enim incididunt qui deserunt ea enim. Nostrud cupidatat dolor anim aliquip culpa. Esse nisi quis nisi Lorem eiusmod ullamco dolore aute enim reprehenderit anim et laboris incididunt. Quis labore sit laborum ipsum commodo labore pariatur adipisicing ut do deserunt eiusmod magna exercitation. Magna irure qui Lorem fugiat pariatur reprehenderit excepteur adipisicing. Proident cillum quis commodo laboris eu veniam aute cupidatat enim. Ullamco reprehenderit fugiat fugiat ex Lorem irure dolore quis reprehenderit. Adipisicing pariatur consequat ullamco sit elit amet ipsum sunt adipisicing anim aliqua ut. Duis anim pariatur nulla ullamco ea proident eiusmod quis. Et incididunt ex ex ullamco labore voluptate excepteur sunt anim voluptate" </div>

Hình 2. 12 Dữ liệu comments

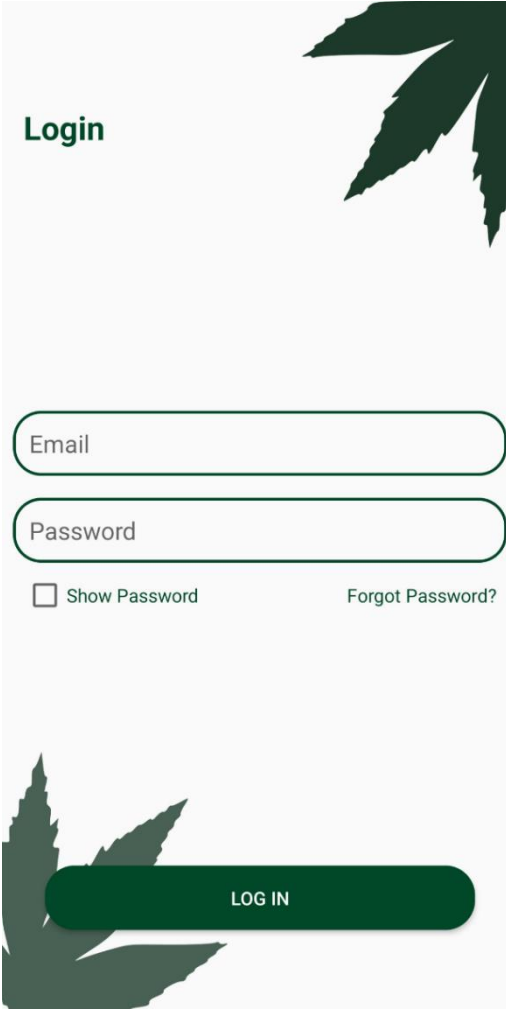
CHƯƠNG 3. KẾT QUẢ VÀ KIỂM THỬ

Từ việc phân tích và thiết kế hệ thống trong chương 2, chương 3 sẽ đề cập đến kết quả thực hiện của ứng dụng Quản lý bug. Chương 3 sẽ trình bày các chức năng cần có của ứng dụng, được hiển thị trực tiếp trên màn hình. Đồng thời, cũng sẽ thực hiện kiểm thử phần mềm để đánh giá xem hệ thống hoạt động đúng theo yêu cầu đã được đặt ra hay không.

3.1 Một số kết quả đạt được

3.1.1 Giao diện Client

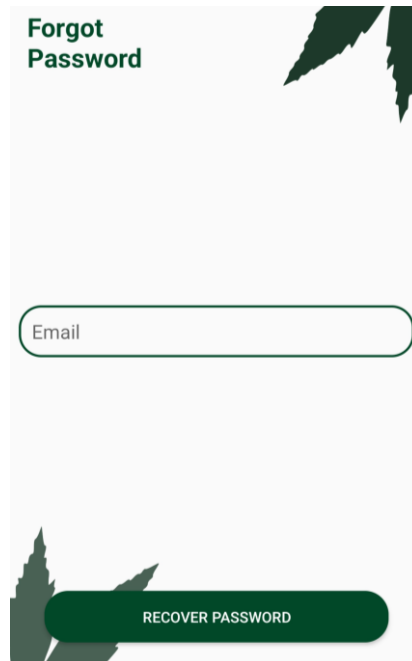
Giao diện đăng nhập được minh hoạ bằng hình dưới đây, người dùng nhập tài khoản và mật khẩu đã được cấp:



The login form is titled "Login" in bold black text. It features two input fields: "Email" and "Password", both with rounded rectangular borders. Below the "Password" field, there is a checkbox labeled "Show Password" and a link labeled "Forgot Password?". At the bottom, there is a large, rounded rectangular button labeled "LOG IN" in white text on a dark green background. A large, stylized green leaf graphic is positioned in the top right corner of the form area.

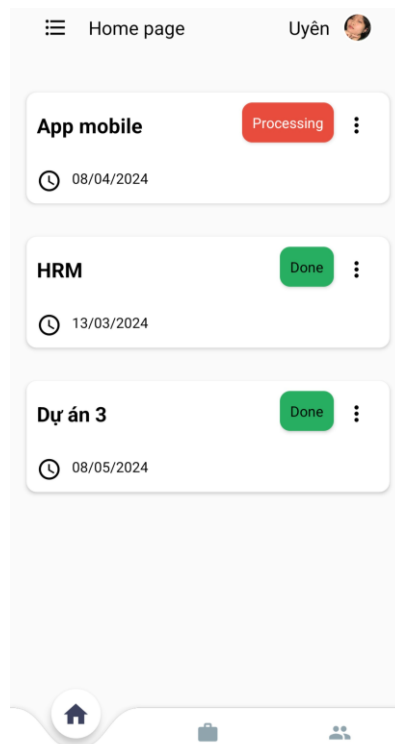
Hình 3. 1 Màn hình đăng nhập

Khi quên mật khẩu người dùng có thể sử dụng chức năng quên mật khẩu, người dùng cần xác thực qua email và nhập lại mật khẩu mới:



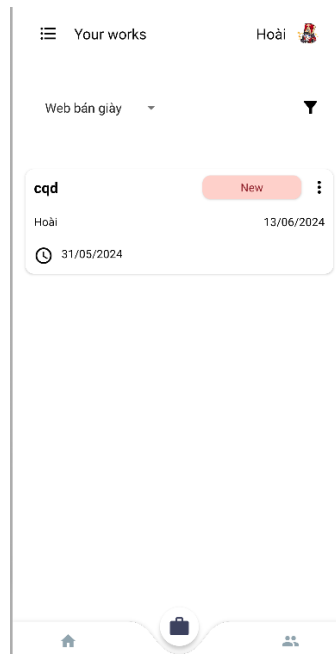
Hình 3. 2 Màn hình quên mật khẩu

Sau khi đăng nhập thành công, người dùng được chuyển đến màn hình trang chủ, tại đây cũng là nơi hiển thị danh sách các project của công ty. Trong đó có các thông tin cơ bản của project và trạng thái của nó:



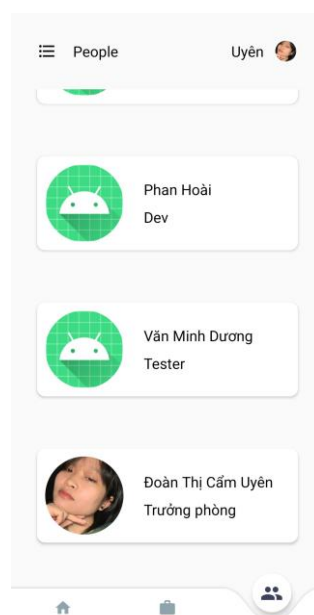
Hình 3. 3 Màn hình trang chủ

Một việc rất cần thiết khi quản lý công việc hằng ngày là cần xem nhanh các công việc của mình, màn hình công việc hiển thị các dự án và bug đang theo dõi hoặc các bug mà mình được gán fix, tại đây cũng có một bộ lọc theo trạng thái và theo project để có thể xem nhanh các công việc:



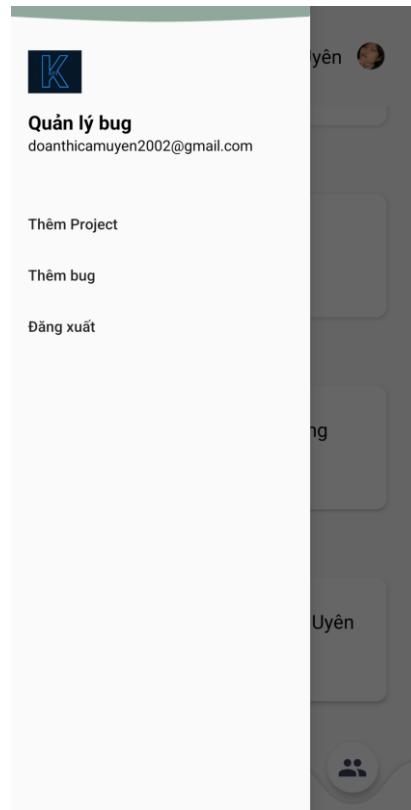
Hình 3. 4 Màn hình công việc

Ngoài theo dõi công việc của bản thân thì người dùng có thể xem công việc của người trong nhóm, khi ấn vào tên từng người sẽ chuyển đến giao diện bảng công việc của người đó:



Hình 3. 5 Màn hình mọi người

Khi ấn nút góc trên trái, hiển thị menu Drawer, tại đây sẽ có các tác vụ nhanh: Thêm project, Thêm bug, Đăng xuất:




Hình 3. 6 Màn hình menu Drawer

← **Thêm mới bug**


Tên lỗi *

Mức độ nghiêm trọng *



Người theo dõi: Tài - Quản lý ▼

Dev fix: Lâm - Dev ▼

Deadline * 

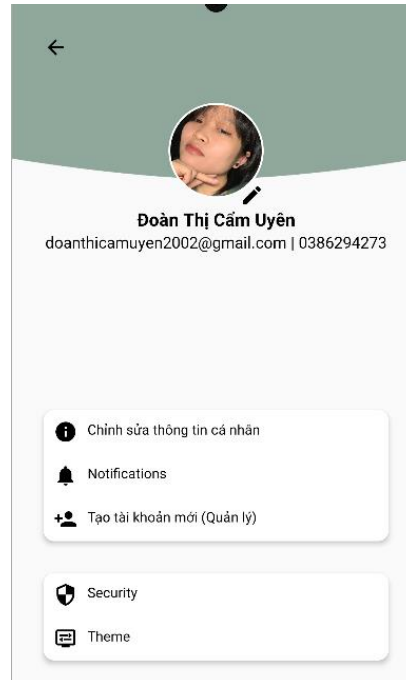
Kết quả mong muốn*

Mô tả *

Số bước * OK

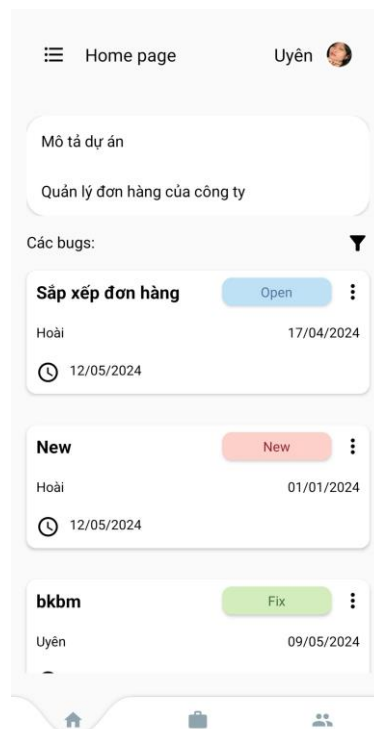
Hình 3. 7 Màn hình thêm mới bug

Khi ấn vào ảnh đại diện ở màn hình trang chủ thì mọi người sẽ xem được thông tin cá nhân của mình, có thể lựa chọn sửa thông tin cá nhân:



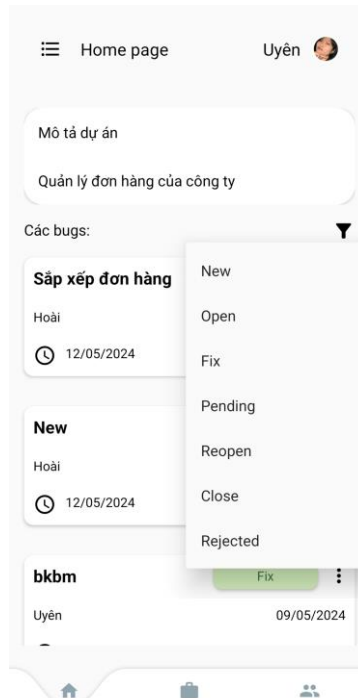
Hình 3. 8 Màn hình profile

Tại màn hình trang chủ, khi ấn vào một project, bạn sẽ được chuyển đến màn hình thông tin chi tiết của project. Tại đây cũng hiển thị danh sách các bug của nó:



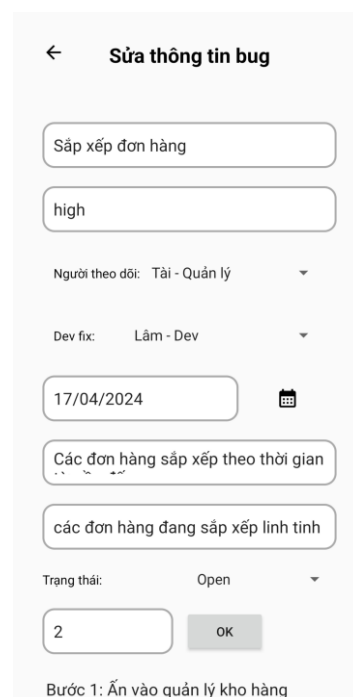
Hình 3. 9 Màn hình chi tiết project

Để người dùng có thao tác nhanh hơn thì tại đây cũng có thiết kế một bộ lọc các bug theo trạng thái, khi lựa chọn trạng thái muốn xem, hệ thống sẽ hiển thị danh sách các bug phù hợp với bộ lọc:



Hình 3. 10 Menu bộ lọc bug

Với các bug, tester có thể chỉnh sửa các thông tin của nó khi cần. Còn đối với dev thì có thể vào sửa trạng thái của bug để cập nhật đến bên test tiến độ fix:



Hình 3. 11 Màn hình sửa bug

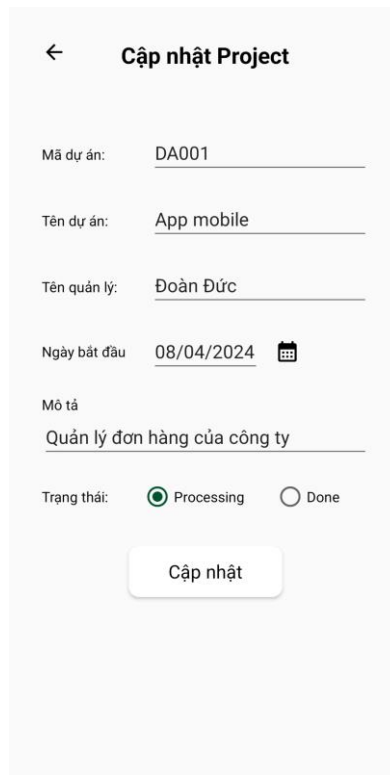
Tại màn hình chi tiết project, khi ấn vào bug hệ thống chuyển đến màn hình chi tiết bug. Tại đây hiển thị đầy đủ các thông tin của bug, người được phân công và trạng thái của nó. Người dùng có thể thêm các bình luận để trao đổi nội dung sửa của các bug tại đây:

Hình 3. 12 Màn hình chi tiết bug

3.1.2 Giao diện Quản lý

Quản lý có thể thêm mới, sửa và xoá project:

Hình 3. 13 Màn hình thêm mới project




← **Cập nhật Project**

Mã dự án:

Tên dự án:

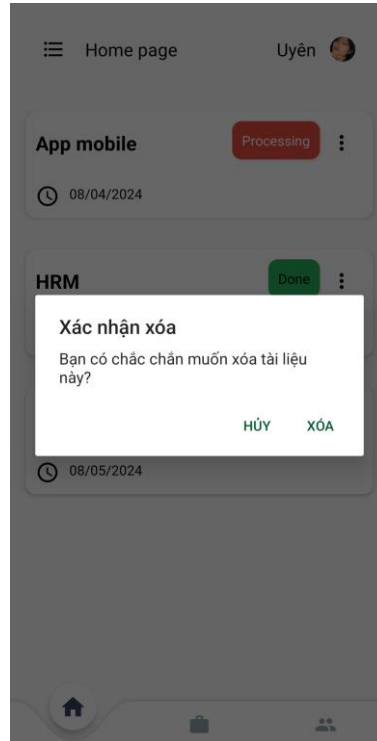
Tên quản lý:

Ngày bắt đầu: 

Mô tả

Trạng thái: ☒ Processing ☐ Done

Hình 3. 14 Màn hình sửa project



Hình 3. 15 Màn hình xác nhận xoá project

Admin có thể thêm tài khoản người dùng, vì đây là ứng dụng sử dụng cho nội bộ công ty nên các tài khoản cần được công ty tạo và cấp cho nhân viên sử dụng:



The image shows a vertical form with six input fields and a button at the bottom. The fields are labeled in Vietnamese: 'Tên đầy đủ của bạn' (Your full name), 'Chức vụ' (Position), 'SĐT' (Phone), 'Gmail', 'Mật khẩu' (Password), and 'Mã nhân viên' (Employee ID). The button is labeled 'Tạo tài khoản' (Create account).

Hình 3. 16 Màn hình tạo tài khoản người dùng

Còn những giao diện khác giống với giao diện của người dùng.

3.2 Kiểm thử phần mềm

3.2.1 Kế hoạch kiểm thử

- Những chức năng được kiểm thử

- Đăng nhập: Người dùng đăng nhập vào hệ thống
- Đăng xuất: Người dùng đăng xuất tài khoản ra khỏi hệ thống
- Chức năng của nhân viên
 - Xem project: Người dùng được xem danh sách các project, các thông tin cơ bản của nó, ấn vào project để xem chi tiết.
 - Bug: Người dùng được quản lý bug mà mình tạo ra, được gán cho. Tester sẽ tạo và gán các bug cho dev. Dev nhận được các bug gán cho mình, thực hiện fix và cập nhật trạng thái của nó. Ấn vào bug để xem chi tiết.
 - Bình luận: Người dùng được xem và thêm các bình luận vào trong chi tiết bug.

- Công việc: Mọi người có thể theo dõi công việc của mình và của mọi người.
- Quản lý thông tin cá nhân: Người dùng xem và chỉnh sửa thông tin cá nhân.
- Chức năng của Admin
 - Quản lý project: Admin có thể xem, sửa và xóa project.
 - Quản lý bug: Admin có thể xem, sửa và xóa bug.
 - Quản lý tài khoản người dùng: Admin có thể xem, thêm và xóa tài khoản người dùng.
 - Quản lý bình luận: Admin có thể xem, thêm, sửa và xóa bình luận vào chi tiết bug.
 - Quản lý công việc: Admin có thể xem, thêm, sửa và xóa công việc của mọi người.
 - Quản lý thông tin cá nhân: Admin xem và chỉnh sửa thông tin cá nhân.

3.2.2 Kiểm thử chức năng phía người dùng

Bảng 3. 1 Kiểm thử chức năng phía người dùng

STT	Case	Đầu vào	Đầu ra mong muốn	Kết quả
1	Kiểm tra chức năng đăng nhập	Nhập tài khoản & mật khẩu hợp lệ	Đăng nhập thành công và đưa người dùng sang trang chủ.	Pass
		Nhập tài khoản & mật khẩu không hợp lệ	Đăng nhập không thành công và thông báo ra màn hình.	Pass
2	Kiểm tra chức năng đăng xuất	Đăng xuất tài khoản	Đăng xuất khỏi tài khoản, chuyển sang trang chủ của Người dùng.	Pass

3	Kiểm tra chức năng lấy lại mật khẩu	Ấn Forgot Password	Được gửi mã xác thực về gmail và đổi mật khẩu mới	Pass
4	Kiểm tra chức năng lưu thông tin đăng nhập	Đăng nhập lại từ lần thứ 2	Không cần phải nhập thông tin đăng nhập, chuyển vào thẳng trang chủ	Pass
5	Kiểm tra chức năng xem project	Xem danh sách project	Hiển thị danh sách tất cả các project.	Pass
		Xem chi tiết project	Hiển thị đúng đủ thông tin chi tiết, danh sách các bug của project.	Pass
6	Kiểm tra chức năng quản lý bug	Tester thêm sửa xóa bug	Được thêm, sửa, xóa các bug, gán các bug cho dev, theo dõi trạng thái của bug	Pass
		Dev xem, sửa bug	Dev xem được tất cả các bug, sửa các bug được gán cho mình	Pass
7	Kiểm tra chức năng bình luận	Xem và thêm bình luận	Xem các bình luận có trong chi tiết bug, thêm bình luận khi cần	Pass
8	Kiểm tra chức năng xem công việc	Xem các công việc	Xem các công việc của mình đã tạo hoặc được giao, xem được các công việc của mọi người	Pass
9	Kiểm tra chức năng lọc công việc	Bộ lọc tìm kiếm công việc theo trạng thái	Lọc được các công việc theo trạng thái, xem nhanh được các công việc đang thực hiện	Pass

10	Kiểm tra chức năng quản lý thông tin cá nhân	Xem và sửa thông tin cá nhân	Xem được các thông tin cơ bản của tài khoản. chỉnh sửa các thông tin cá nhân	Pass
----	--	------------------------------	--	------

3.2.3 Kiểm thử chức năng phía Admin

Bảng 3. 2 Kiểm thử chức năng phía Admin

STT	Case	Đầu vào	Đầu ra mong muốn	Kết quả
1	Kiểm tra chức năng quản lý project	Nhập thông tin hợp lệ	Thêm project thành công, sửa được project	Pass
		Nhập thông tin không hợp lệ	Báo lỗi ở các trường nhập không hợp lệ	Pass
2	Kiểm tra chức năng quản lý bug	Nhập thông tin bug hợp lệ	Thêm, sửa bug thành công và hiển thị danh sách các bug lên màn hình	Pass
		Nhập thông tin bug không hợp lệ	Báo lỗi tại input có dữ liệu không hợp lệ	Pass
3	Kiểm tra chức năng quản lý tài khoản người dùng	Nhập thông tin hợp lệ	Thêm, sửa tài khoản người dùng thành công và đăng nhập lại	Pass
		Nhập thông tin không hợp lệ	Báo lỗi tại input có dữ liệu không hợp lệ	Pass
4	Kiểm tra chức năng quản lý công việc	Nhập thông tin công việc hợp lệ	Thêm, sửa công việc thành công và hiển thị danh sách các bug lên màn hình	Pass

		Nhập thông tin bug không hợp lệ	Báo lỗi tại input có dữ liệu không hợp lệ	Pass
5	Kiểm tra chức năng quản lý bình luận	Nhập thông tin bình luận hợp lệ	Thêm, sửa bình luận thành công và hiển thị danh sách các bug lên màn hình	Pass
		Nhập thông tin bug không hợp lệ	Báo lỗi tại input có dữ liệu không hợp lệ	Pass
6	Kiểm tra chức năng quản lý tài khoản cá nhân	Xem và chỉnh sửa thông tin cá nhân	Xem được các thông tin cơ bản của tài khoản. chỉnh sửa các thông tin cá nhân	Pass

3.2.4 Kết quả kiểm thử

- Tỷ lệ test case đạt (Passed): 100%
- Hệ thống chạy ổn định trên nền tảng Android.

KẾT LUẬN

Đề tài “**Xây dựng ứng dụng di động Quản lý bug cho công ty KPI trên nền tảng Android sử dụng ngôn ngữ lập trình Java**” xuất phát từ những nhu cầu thực tế mà ngày nay trong ngành công nghệ thông tin cần có. Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và vận dụng được những kiến thức công nghệ được học từ trường, từ nơi thực tập và đã hoàn thành được những phần sau đây.

- Khảo sát và phân tích yêu cầu.
- Phân tích thiết kế và đặc tả hệ thống.
- Thiết kế cơ sở dữ liệu.
- Xây dựng được những chức năng đã đặt ra ban đầu.

***Hướng phát triển**

- Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
- Từng bước xây dựng ứng dụng đa nền tảng
- Hoàn thiện chương trình, sửa chữa các lỗi, cải thiện tính bảo mật.
- Xây dựng thêm các tính năng thống kê và các chức năng cho quản lý.
- Tích hợp đăng nhập qua Facebook, Google.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.
- [2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phụng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.
- [3] Nguyễn Bá Nghiễn, *Lập trình java*, NXB thông tin và truyền thông.
- [4] Trần Đình Quân, *Phát triển ứng dụng Android với Android Studio*, Nhà xuất bản Đại học Quốc

Tiếng Anh

- [1] Ian F. Darwin, *Android Cookbook: Problems and Solutions for Android Developers*, 2nd Edition, O'Reilly Media, 2017.
- [2] Phil Dutson, *Android Development Patterns: Best Practices for Professional Developers*, Addison-Wesley Professional, 2016.

Website

- [1] Trang tài liệu Android studio (truy cập lần cuối 12/05/2023) <https://developer.android.com/?hl=vi>
- [2] Trang tài liệu Java (truy cập lần cuối 11/05/2023) <https://www.w3schools.com/java/default.asp>
- [3] Trang tin tức về lập trình (truy cập lần cuối 12/05/2023) <https://viblo.asia>