BỘ THÔNG TIN VÀ TRUYỀN THÔNG HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

Đề tài: Xây dựng ứng dụng tạo lịch họp trên điện thoại di động

Giáo viên hướng dẫn : ThS. NGUYỄN ANH HÀO

Sinh viên thực hiện : NGÔ THỊ TRANG NHUNG

Mã số sinh viên : N20DCCN051

Lóp : D20CQCNPM01 - N

Khóa : 2020 - 2025

Hệ : ĐẠI HỌC CHÍNH QUY

TPHCM, 5 tháng 8 năm 2024

BỘ THÔNG TIN VÀ TRUYỀN THÔNG HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

Đề tài: Xây dựng ứng dụng tạo lịch họp trên điện thoại di động

Giáo viên hướng dẫn : ThS. NGUYỄN ANH HÀO

Sinh viên thực hiện : NGÔ THỊ TRANG NHUNG

Mã số sinh viên : N20DCCN051

Lóp : D20CQCNPM01 - N

Khóa : 2020 - 2025

Hệ : ĐẠI HỌC CHÍNH QUY

TPHCM, 5 tháng 8 năm 2024

LÒI CẨM ƠN

Lời đầu tiên, em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kỳ vừa qua. Từ những kiến thức, kinh nghiệm và lời dạy bảo của các thầy cô mà em đã ngày càng trưởng thành hơn, tự tin đối đầu với các thử thách trong học tập, công việc và cuộc sống để đạt được những thành tựu mà bản thân mong muốn.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến ThS. Nguyễn Anh Hào. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho em trong suốt thời gian học tập và thực hiện đề tài. Kính chúc thầy và gia đình nhiều sức khoẻ và thành công trong cuộc sống.

Tuy nhiên, vì thời gian và kiến thức chuyên môn còn hạn chế cùng nhiều yếu tố khách quan khác nên trong quá trình thực hiện đề tài không tránh khỏi những sai sót. Em rất mong nhận được sự thông cảm và góp ý từ cô để có thể học hỏi thêm những kiến thức mới và kinh nghiệm để đề tài thực tập tốt nghiệp này có kết quả tốt nhất. Một lần nữa, em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 5 tháng 08 năm 2024 Sinh viên thực hiện

Ngô Thị Trang Nhung

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

MŲC LŲC

LỚI CẨM ƠN	••••
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	
DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH	••••
KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT	••••
LỜI NÓI ĐẦU	
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	1
1.1 Tổng quan về đề tài	1
1.1.1 Lí do chọn đề tài	1
1.1.2 Mục đích	1
1.1.2 Mục tiêu	2
1.1.3 Phương pháp tiến hành	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT ĐỀ TÀI	4
2.1 Tổng quan về công nghệ thông tin và quản lý công việc	4
2.1.1 Khái niệm công nghệ thông tin và quản lý công việc	4
2.1.2 Sự kết hợp giữa công nghệ thông tin và quản lý công việc	4
2.1.3 Các hình thức ứng dụng công nghệ thông tin trong quản lý công việc	4
2.1.4 Lợi ích của việc ứng dụng công nghệ thông tin trong quản lý công việc	5
2.1.5 Thách thức và giải pháp	5
2.2 Giới thiệu về lập trình Android	6
2.2.1 Tổng quan về android	6
2.2.1 Kiến trúc cơ bản của hệ điều hành Android	7
2.3 Giới thiệu về Java	8
2.3.1 Tổng quan về Java	8
2.3.2 Ưu nhược điểm của Java	9
2.3.3 Java trong lập trình ứng dụng	. 12
2.3.4 Lợi ích của việc sử dụng Java trong lập trình ứng dụng	. 12
2.4 Giới thiêu về Node.js	. 13

2.4.1 Tổng quan về Node.js	13
2.4.2 Các tính năng nổi bật của Node.js	13
2.4.3 Node js làm việc như thế nào?	14
2.4.4 NPM: hệ thống quản lý gói cho Node.js	15
2.4.5 Node.js trong phát triển ứng dụng	16
2.4.6 Lợi ích của việc sử dụng Node.js	17
2.4.7 Cách cài đặt Node.js	17
2.5 Giới thiệu về MySQL	18
2.5.1 Tổng quan về MySQL	18
2.5.2 Các tính năng nổi bật của MySQL	19
2.4.3 MySQL trong phát triển ứng dụng	19
2.5.4 Lợi ích của việc sử dụng MySQL	20
2.5.5 Cách cài đặt MySQL	20
2.6 Giới thiệu về SDK của Điện thoại Di động	20
2.6.1 Tổng quan về SDK	20
2.6.2 Các Thành Phần Chính của SDK	21
2.6.3 Các SDK Chính Cho Di Động	22
CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG	24
3.1 Khảo sát hệ thống ứng dụng tạo lịch họp trên điện thoại di động	24
3.2 Các chức năng của hệ thống	25
3.2.1 Yêu cầu chức năng của hệ thống:	25
3.3 Bối cảnh của hệ thống	26
3.3.1 Định nghĩa vấn đề mà đề tài sẽ giải quyết	26
3.3.2 Hiện trạng trước khi sử dụng phần mềm:	26
3.3.3 Đề xuất giải pháp của đề tài	27
3.4 Định nghĩa các tương tác cần thiết trên PM	36
3.4.1. Lược đồ Use Case	36
3.5 Định nghĩa yêu cầu và ràng buộc đối với phần mềm	42
3.5.1 Yêu cầu từ môi trường nghiệp vụ (business)	42
3.5.2 Yêu cầu từ môi trường vận hành (operation)	43

3.5.3 Yêu cầu từ môi trường phát triển (development)	44
CHƯƠNG 4: THIẾT KẾ PHẦN MỀM	46
4.1 Lược đồ usecase	46
4.1.1 Use case tổng quát	46
4.1.2 Use case đăng nhập, đăng ký	46
4.1.3 Use case hop	47
4.1.4 Use case kết thúc cuộc họp	47
4.2 Sơ đồ tuần tự	48
4.2.1 Sơ đồ tuần tự đăng nhập	48
4.2.2 Sơ đồ tuần tự tạo cuộc họp	49
4.2.3 Sơ đồ tuần tự xem danh sách cuộc họp	50
4.2.4 Sơ đồ tuần tự nội dung họp	51
4.3 Thiết kế các thành phần của phần mềm	52
4.3.1 Đăng ký	52
4.3.2 Đăng nhập	55
4.3.3 Trang chủ admin	57
4.3.4 Tạo cuộc họp	58
4.3.5 Xem danh sách nhân viên	62
4.3.6 Xem danh sách cuộc họp và chi tiết cuộc họp	64
4.3.7 Danh sách người tham gia trong một cuộc họp	66
4.3.8 Xem kết quả cuộc họp	68
4.3.9 Chia sẽ thông tin cuộc họp	71
4.4 Thiết kế cơ sở dữ liệu và ràng buộc	72
4.4.1 Thiết lập mô hình thực thể	72
4.4.2 Thiết kế cơ sở dữ liệu	74
CHƯƠNG 5. KẾT LUẬN - HƯỚNG PHÁT TRIỂN	83
DANH MỤC TÀI LIỆU THAM KHẢO	85

DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Bảng 3.1 Yêu cầu từ môi trường nghiệp vụ	43
Bảng 3.2 Các yêu cầu chức năng	
Bảng 3.3 Các yêu cầu chất lượng	44
Bảng 3.4 Ràng buộc về phương pháp/công nghệ	
Bảng 3.5 Ràng buộc về phương pháp/công nghệ	
Bång 4.1 Bång Users	77
Bång 4.2 Bång Meetings	78
Bång 4.3 Bång MeetingParticipants	79
Bång 4.4 Bång MeetingResults	80
Bång 4.5 Bång Summaries	80
Bång 4.6 Bång SummaryApproval	81
Bång 4.7 Bång MeetingReminders	82
Sơ đồ 3.1 Lược đồ công tác lập lịch họp	
Sơ đồ 3.2 Lược đồ cộng tác lập lịch họp khi có phần mềm	31
Sơ đồ 3. 3 Sơ đồ tuần tự lập lịch họp	31
Sơ đồ 3.4 Sơ đồ tuần tự quản lý danh sách tham dự	
Sơ đồ 3.5 Sơ đồ tuần tự gửi kết quả cuộc họp	33
Sơ đồ 3.6 Sơ đồ tuần tự gửi kết đánh giá cuộc họp	34
Sơ đồ 3.7 Sơ đồ tuần tự gửi nhắc nhở	
Sơ đồ 3.8. Sơ đồ Use Case 1: tạo cuộc họp	36
Sơ đồ 3.9 Sơ đồ tuần tự tạo cuộc họp	
Sơ đồ 3.10 Sơ đồ Use Case 2: xem danh sách cuộc họp	
Sơ đồ 3.11 Sơ đồ tuần tự xem thông tin cuộc họp	
Sơ đồ 3.12 Sơ đồ Use Case 3: Quản lý người dung	
Sơ đồ 3.13 Sơ đồ tuần tự quản lý người dung	39
Sơ đồ 3.14 Sơ đồ Use Case 5: Xem Thông Tin Cuộc Họp	41
Sơ đồ 3.15 Sơ đồ tuần tự xem thông tin cuộc họp	42
Sơ đồ 4.1 Use case tổng quát	46
Sơ đồ 4.2 Use case đăng nhập, đăng ký	
Sơ đồ 4.3 Use case họp	47
Sơ đồ 4.4 Use case kết thúc cuộc họp	47
Sơ đồ 4.5 Sơ đồ tuần tự đăng nhập	48
Sơ đồ 4. 6 Sơ đồ tuần tư tao cuộc họp	

Sơ đồ 4.7 Sơ đồ tuần tự xem danh sách cuộc họp	50
Sơ đồ 4.8 Sơ đồ tuần tự nội dung họp	51
Sơ đồ 4.9 ERD	
Sơ đồ 4.10 Diagram	75
Sơ đồ 4.11 Sơ đồ lớp	
Hình 2.1 Java	8
Hình 4.1Đăng Ký	
Hình 4.2 SP_User_Register	
Hình 4. 3 Đăng nhập người dùng	55
Hình 4.4 SP_User_Login	57
Hình 4.5 Trang chủ admin	57
Hình 4. 6 Tạo cuộc họp	
Hình 4.7 Tạo cuộc họp	59
Hình 4.8 SP_Create_Meeting	61
Hình 4.9 danh sách nhân viên	62
Hình 4.10 SP_Get_Employee_List	63
Hình 4. 11 danh sách cuộc họp	
Hình 4.12 Chi tiết cuộc họp	64
Hình 4.13 SP_Get_Meeting_List	66
Hình 4.14 Danh sách người tham gia một cuộc họp	66
Hình 4.15 SP_Get_Participants_By_Meeting	68
Hình 4. 16 Kết quả cuộc họp với phê duyệt	68
Hình 4. 17 SP_Approve_Meeting_Result	
Hình 4.18 Chia sẻ thông tin cuộc họp	71
Hình 4.19 Chia sẻ thông tin cuộc họp	71

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

STT	Ký hiệu	Nguyên nghĩa
1	OOAD	Phân tích và thiết kế hướng đối tượng (Object-Oriented
		Analysis and Design)
2	CNTT	Công nghệ thông tin
3	AI	Trí tuệ nhân tạo (Artificial Intelligence)
4	HD	Độ nét cao (High Definition)
5	Heap	Vùng bộ nhớ động dùng để lưu trữ các đối tượng được tạo
	_	động
6	Stack	Vùng bộ nhớ dùng để lưu trữ các biến cục bộ, tham số hàm và
		địa chỉ trả về
7	Oracle	Hệ quản trị cơ sở dữ liệu quan hệ lớn nhất thế giới
8	StackOverflow	Trang web hỏi đáp dành cho lập trình viên
9	ERD	Sơ đồ thực thể-quan hệ (Entity-Relationship Diagram)
10	JVM	Máy ảo Java (Java Virtual Machine)
11	GUI	Giao diện người dùng đồ họa (Graphical User Interface)
12	MVC	Mô hình kiến trúc phần mềm chia thành ba phần chính: Model,
		View, và Controller
13	Java Swing	Thư viện đồ họa của Java để xây dựng các ứng dụng GUI
14	JavaFX	Thư viện đồ họa của Java để xây dựng các ứng dụng GUI
15	NPM	Trình quản lý gói cho JavaScript (Node Package Manager)
16	HTTP	Giao thức truyền siêu văn bản (Hypertext Transfer Protocol)
17	API	Giao diện lập trình ứng dụng (Application Programming
		Interface)
18	SDK	Bộ công cụ phát triển phần mềm (Software Development Kit)
19	PM	Phần mềm (Phần mềm quản lý)
20	UML	Ngôn ngữ mô hình hóa thống nhất (Unified Modeling
		Language)
21	Use Case	Tình huống sử dụng (Use Case) - Mô tả các tương tác giữa
		người dùng và hệ thống
22	Sequence	Sơ đồ tuần tự (Sequence Diagram) - Mô tả tuần tự các bước và
	Diagram	thông tin trao đổi giữa các đối tượng trong hệ thống

LỜI NÓI ĐẦU

Trong nhịp sống hiện đại, việc quản lý thời gian hiệu quả là vô cùng quan trọng. Đặc biệt trong môi trường làm việc năng động, các cuộc họp trở thành một phần không thể thiếu. Tuy nhiên, việc lên lịch và quản lý các cuộc họp truyền thống thường gặp nhiều khó khăn. Nhận thấy nhu cầu này, em đã xây dựng một ứng dụng tạo lịch họp trên điện thoại di động nhằm giúp người dùng dễ dàng tạo, quản lý và tham gia các cuộc họp, từ đó tiết kiệm thời gian và nâng cao hiệu quả làm việc.

Nhận thấy những hạn chế trên, em đã lựa chọn đề tài "Xây dựng ứng dụng tạo lịch họp trên điện thoại di động" để nghiên cứu và phát triển. Ứng dụng này nhằm mục tiêu cung cấp một giải pháp toàn diện, giúp người dùng dễ dàng tạo, quản lý và tham gia các cuộc họp một cách hiệu quả.

Với ứng dụng này, người dùng có thể tạo lịch họp nhanh chóng và dễ dàng, chỉ với vài thao tác đơn giản. Quản lý lịch họp tập trung, bao gồm thông tin chi tiết về thời gian, địa điểm, người tham gia và nội dung cuộc họp. Nhận thông báo về các cuộc họp sắp diễn ra để không bỏ lỡ bất kỳ cuộc họp nào. Tham gia các cuộc họp trực tuyến và trực tiếp một cách thuân tiên thông qua tính năng tích hợp cuộc gọi video.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan về đề tài

1.1.1 Lí do chọn đề tài

Ngày nay, điện thoại di động trở thành thiết bị không thể thiếu trong cuộc sống, thay thế cho máy tính truyền thống trong nhiều tác vụ. Nhu cầu họp hành ngày càng cao trong môi trường làm việc hiện đại, đòi hỏi một công cụ hỗ trợ hiệu quả và tiện lợi. Do đó, ứng dụng tạo lịch họp trên điện thoại di động ra đời nhằm đáp ứng nhu cầu này.

Úng dụng cần có những tính năng cơ bản như tạo lịch họp, mời tham dự, xác nhận tham dự, nhắc nhở. Ngoài ra, một số tính năng nâng cao như lặp lại, lịch trình, tích hợp lịch, chia sẻ tài liệu, ghi chép biên bản, bình luận, thăm dò ý kiến, video họp cũng cần được tích hợp để tăng hiệu quả sử dụng. Giao diện ứng dụng cần đơn giản, dễ sử dụng, tối ưu cho màn hình điện thoại di động và hỗ trợ đa ngôn ngữ. Bảo mật thông tin người dùng và cuộc họp là yếu tố quan trọng cần được đảm bảo.

Úng dụng tạo lịch họp trên điện thoại di động mang lại nhiều lợi ích như tiết kiệm thời gian, tăng hiệu quả họp, cải thiện giao tiếp và thể hiện sự chuyên nghiệp. Một số ứng dụng phổ biến hiện nay là Google Calendar, Microsoft Outlook, Doodle, Zoom và Slack.

Với tiềm năng phát triển to lớn và sự hỗ trợ của các công nghệ mới như trí tuệ nhân tạo và học máy, ứng dụng tạo lịch họp trên điện thoại di động hứa hẹn sẽ trở thành công cụ hữu ích cho cá nhân và tổ chức trong tương lai.

1.1.2 Mục đích

Trong thời đại số, việc quản lý thời gian và công việc hiệu quả trở nên quan trọng hơn bao giờ hết. Ứng dụng tạo lịch họp trên điện thoại di động ra đời như một giải pháp thông minh, giúp người dùng tối ưu hóa quá trình lên kế hoạch và tổ chức các cuộc họp. Với tính năng đa dạng như tạo cuộc họp lặp lại, tích hợp với các phần mềm quản lý dự án, và chia sẻ tài liệu dễ dàng, ứng dụng không chỉ tiết kiệm thời gian mà còn đảm bảo tất cả các hoạt động diễn ra một cách chuyên nghiệp và hiệu quả. Nhờ đó, các doanh nghiệp có thể giảm thiểu

thời gian tổ chức họp, nâng cao hiệu quả làm việc nhóm và tạo ra một môi trường làm việc chuyên nghiệp. Với giao diện trực quan và dễ sử dụng, ứng dụng có thể được truy cập mọi lúc mọi nơi, giúp người dùng luôn nắm bắt được lịch trình của mình.

1.1.2 Mục tiêu

Ứng dụng tập trung giải quyết các vấn đề sau đây và hướng tới các đối tượng người dùng cu thể:

Quản lý lịch họp: Ứng dụng giúp người quản lý sắp xếp, theo dõi và cập nhật lịch họp một cách dễ dàng và hiệu quả

Thông báo và nhắc nhở: Ứng dụng gửi thông báo và nhắc nhở cho các thành viên về thời gian và địa điểm của các cuộc họp, đảm bảo mọi người đều được thông báo kịp thời.

Theo dõi và cập nhật kết quả cuộc họp: Ứng dụng cho phép cập nhật và lưu trữ kết quả các cuộc họp, giúp dễ dàng theo dõi tiến trình và kết quả công việc.

Hỗ trợ việc chia sẻ thông tin cuộc họp: Giúp người dùng có thể chia sẻ các thông tin cuộc họp với nhau.

1.1.3 Phương pháp tiến hành

Phương pháp phát triển ứng dụng được tiến hành theo hướng phân tích và thiết kế hướng đối tượng (Object-Oriented Analysis and Design - OOAD). Quá trình phát triển gồm các bước chính sau:

Phân tích yêu cầu: Thu thập và phân tích các yêu cầu từ người dùng để xác định các chức năng cần có của ứng dụng.

Thiết kế hệ thống: Thiết kế kiến trúc tổng thể của ứng dụng, bao gồm thiết kế cơ sở dữ liệu, thiết kế giao diện người dùng, và các module chức năng.

Phát triển ứng dụng: Lập trình các chức năng của ứng dụng theo thiết kế đã được xác định, đảm bảo tích hợp đầy đủ các tính năng.

Kiểm thử và triển khai: Thực hiện kiểm thử ứng dụng để đảm bảo hoạt động ổn định và không có lỗi, sau đó triển khai lên các nền tảng di động để người dùng có thể tải về và sử dụng.

Bảo trì và nâng cấp: Liên tục cập nhật và nâng cấp ứng dụng dựa trên phản hồi từ người dùng và yêu cầu mới.

Ứng dụng tạo lịch họp trên điện thoại di động không chỉ giúp người dùng quản lý lịch họp một cách hiệu quả mà còn nâng cao khả năng phối hợp và làm việc nhóm trong các tổ chức.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT ĐỀ TÀI

2.1 Tổng quan về công nghệ thông tin và quản lý công việc

2.1.1 Khái niệm công nghệ thông tin và quản lý công việc

Sự phát triển không ngừng của công nghệ thông tin (IT) đã tạo nên một cuộc cách mạng trong cách thức chúng ta làm việc và quản lý công việc. Công nghệ thông tin là một lĩnh vực bao trùm tất cả các khía cạnh liên quan đến việc sử dụng máy tính, phần mềm, mạng và các thiết bị để lưu trữ, xử lý, truyền tải và bảo vệ thông tin. Trong bối cảnh này, ứng dụng công nghệ thông tin vào quản lý công việc giúp tối ưu hóa hiệu quả và năng suất làm việc.

Quản lý công việc (task management) là quá trình theo dõi, lập kế hoạch, tổ chức và quản lý các nhiệm vụ và dự án từ khi bắt đầu đến khi hoàn thành. Việc ứng dụng công nghệ thông tin vào quản lý công việc giúp tự động hóa các quy trình, tăng cường sự phối hợp giữa các thành viên, và cải thiện khả năng giám sát và đánh giá hiệu quả công việc.

2.1.2 Sự kết hợp giữa công nghệ thông tin và quản lý công việc

Sự kết hợp giữa CNTT và quản lý công việc đã tạo ra một cuộc cách mạng trong cách thức chúng ta làm việc. Các công cụ và phần mềm quản lý công việc dựa trên CNTT giúp tự động hóa nhiều quy trình, nâng cao hiệu suất làm việc, và tạo điều kiện thuận lợi cho sự hợp tác giữa các thành viên trong nhóm.

2.1.3 Các hình thức ứng dụng công nghệ thông tin trong quản lý công việc

- Úng dụng quản lý dự án: Các phần mềm như Trello, Asana, Microsoft Project giúp lên kế hoạch, theo dõi tiến độ, phân công nhiệm vụ và quản lý tài nguyên của dự án.
- Úng dụng quản lý tác vụ: Các ứng dụng như Todoist, Google Tasks, Microsoft To-Do giúp cá nhân hóa việc quản lý công việc hàng ngày, tạo danh sách việc cần làm, đặt nhắc nhở và theo dõi tiến độ.

- Công cụ giao tiếp và hợp tác: Các nền tảng như Slack, Microsoft Teams, Zoom giúp các nhóm làm việc kết nối, chia sẻ thông tin, tổ chức cuộc họp trực tuyến và làm việc cùng nhau một cách hiệu quả.
- Hệ thống quản lý tài liệu: Các công cụ như Google Drive, Dropbox giúp lưu trữ, chia sẻ và đồng bộ hóa tài liệu một cách an toàn và tiện lợi.
- Trí tuệ nhân tạo (AI) và máy học: Các công cụ AI được tích hợp vào các phần mềm quản lý công việc để tự động hóa các tác vụ lặp đi lặp lại, dự đoán rủi ro và đưa ra các đề xuất tối ưu hóa.

2.1.4 Lợi ích của việc ứng dụng công nghệ thông tin trong quản lý công việc

- Tăng năng suất: Tự động hóa các quy trình, giảm thiểu thời gian thực hiện các tác vụ thủ công.
- Cải thiện sự phối hợp: Tạo điều kiện cho các thành viên trong nhóm làm việc cùng nhau một cách hiệu quả, bất kể khoảng cách địa lý.
- Nâng cao tính minh bạch: Tất cả các thông tin về dự án và công việc được lưu trữ và chia sẻ một cách minh bạch, giúp mọi người hiểu rõ về tiến độ và trách nhiệm của mình.
- Đưa ra quyết định tốt hơn: Các dữ liệu được thu thập và phân tích giúp người quản
 lý đưa ra các quyết định dựa trên cơ sở khoa học.
- Tiết kiệm chi phí: Giảm thiểu chi phí in ấn, giấy tờ và các nguồn lực khác.

2.1.5 Thách thức và giải pháp

- Thách thức:
- + Bảo mật thông tin: Áp dụng các biện pháp bảo mật mạnh mẽ để bảo vệ dữ liệu nhạy cảm.
- + Chi phí đầu tư: Đầu tư vào các công cụ và phần mềm quản lý công việc có thể đòi hỏi chi phí ban đầu lớn.
- + Kháng cự thay đổi: Một số người dùng có thể gặp khó khăn trong việc thích nghi với công nghê mới.

+ Phụ thuộc vào công nghệ: Sự cố về hệ thống hoặc mạng có thể gây gián đoạn công việc.

Giải pháp:

- + Chọn lựa công cụ phù hợp: Lựa chọn các công cụ phù hợp với quy mô và nhu cầu của tổ chức.
- Đào tạo người dùng: Tổ chức các khóa đào tạo để giúp người dùng làm quen với các công cụ mới.
- + Đảm bảo tính bảo mật: Áp dụng các biện pháp bảo mật mạnh mẽ như mã hóa dữ liệu, xác thực hai yếu tố.
- Xây dựng kế hoạch dự phòng: Có kế hoạch dự phòng để đối phó với các tình huống khẩn cấp

2.2 Giới thiệu về lập trình Android

2.2.1 Tổng quan về android

Android là hệ điều hành trên điện thoại di động (và hiện nay là cả trên một số đầu phát HD, HD Player, TV) phát triển bởi Google và dựa trên nền tảng Linux. Lập trình android là một lập trình ứng dụng di động phổ biến. Trước đây, Android được phát triển bởi công ty liên hợp Android (sau đó được Google mua lại vào năm 2005).

Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đính tạo nên một chuẩn mở cho điện thoại di động trong tương lai.

2.2.1 Kiến trúc cơ bản của hệ điều hành Android

Android gồm 5 phần chính sau được chứa trong 4 lớp:

- Nhân Linux: Đây là nhân nền tảng mà hệ điều hành Android dựa vào nó để phát triển. Đâu là lớp chứa tất cả các thiết bị giao tiếp ở mức thấp dùng để điều khiển các phần cứng khác trên thiết bị Android.
- Thư viện: Chứa tất cả các mã cái mà cung cấp cấp những tính năng chính của hệ điều hành Android, đôi với ví dụ này thì SQLite là thư viện cung cấp việc hộ trợ làm việc với database dùng để chứa dữ liệu. Hoặc Webkit là thư viện cung cấp những tính năng cho trình duyệt Web.
- Android runtime: Là tầng cùng với lớp thư viện Android runtime cung cấp một tập các thư viện cốt lỗi để cho phép các lập trình viên phát triển viết ứng dụng bằng việc sử dụng ngôn ngữ lập trình Java. Android Runtime bao gốm máy ảo Dalvik(ở các version < 4.4, hiện tài là phiên bản máy ảo ART được cho là mạnh mẽ hơn trong việc xử lý biên dịch). Là cái để điều khiển mọi hoạt động của ứng dụng Android chạy trên nó(máy ảo Dalvik sẽ biên dịch ứng dụng để nó có thể chạy(thực thi) được , tương tự như các ứng dụng được biên dịch trên máy ảo Java vậy). Ngoài ra máy ảo còn giúp tối ưu năng lượng pin cũng như CPU của thiết bị Android</p>
- Android framework: Là phần thể hiện các khả năng khác nhau của Android(kết nối, thông báo, truy xuất dữ liệu) cho nhà phát triển ứng dụng, chúng có thể được tạo ra để sử dụng trong các ứng dụng của họ.
- Application: Tầng ứng dụng là tầng bạn có thể tìm thấy chuyển các thiết bị Android như Contact, trình duyệt... Và mọi ứng dụng bạn viết đều nằm trên tầng này.

2.3 Giới thiệu về Java

2.3.1 Tổng quan về Java

Java là một ngôn ngữ lập trình hướng đối tượng thuần túy, được phát triển bởi Sun Microsystems vào năm 1995 và hiện thuộc sở hữu của Oracle Corporation. Java được thiết kế để có cú pháp đơn giản, dễ học và dễ sử dụng. Một trong những đặc điểm nổi bật của Java là khả năng "viết một lần, chạy ở mọi nơi" (write once, run anywhere), tức là mã Java đã biên dịch có thể chạy trên mọi nền tảng hỗ trợ Java mà không cần biên dịch lại.



Hình 2.1 Java

Java nổi bật với các tính năng như:

- Hướng đối tượng: Java cho phép lập trình viên xây dựng ứng dụng bằng cách sử dụng các đối tượng và lớp, giúp mã nguồn có cấu trúc rõ ràng và dễ bảo trì.
- Bảo mật cao: Java có các cơ chế bảo mật mạnh mẽ, giúp bảo vệ ứng dụng khỏi các
 lỗ hổng bảo mật như tràn bộ đệm và lỗi trỏ NULL.
- Quản lý bộ nhớ tự động: Java sử dụng cơ chế quản lý bộ nhớ tự động (garbage collection), giúp giảm thiểu các lỗi liên quan đến quản lý bộ nhớ như rò rỉ bộ nhớ.
- Khả năng tương thích đa nền tảng: Mã Java được biên dịch thành bytecode và chạy trên Java Virtual Machine (JVM), giúp ứng dụng Java có thể chạy trên nhiều nền tảng khác nhau.

2.3.2 Ưu nhược điểm của Java

- ❖ Có rất nhiều lợi thế khi sử dụng Java làm ngôn ngữ lập trình, bao gồm:
- Đơn giản: Java là ngôn ngữ lập trình dễ học và dễ hiểu. Ngoài ra, Java còn loại bỏ được tất cả các tính năng phức tạp của C và C++ như con trỏ, cấu trúc, liên kết và làm cho việc triển khai mã dễ dàng hơn.
- Hướng đối tượng (OOP): Một trong những ưu điểm chính của Java là nó là ngôn ngữ lập trình hướng đối tượng. Chúng ta đều biết rằng ngôn ngữ lập trình thủ tục rất phức tạp và khó thực hiện. Rõ ràng việc sử dụng khái niệm OOP làm cho Java dễ triển khai hơn và an toàn hơn nhiều. Nó cũng giúp duy trì đoạn mã lớn bằng cách chia chúng thành các đoạn nhỏ hơn được đặt tên.
- Bảo mật: Các ngôn ngữ như C và C++ sử dụng con trỏ, cho phép truy cập vào vị trí bộ nhớ. Đây là một rủi ro bảo mật vì con trỏ có thể dẫn đến truy cập bộ nhớ trái phép. Sun Microsystems đã loại bỏ tính năng này trên Java khiến nó trở thành một ngôn ngữ bảo mật tốt. Ngoài ra, Java cũng sử dụng các khái niệm OOP như đóng gói, trừu tượng hóa, kế thừa giúp tăng tính bảo mật và ngăn chặn truy cập trái phép đối với người dùng.
- Không phụ thuộc vào nền tảng: Java tuân theo tính năng của "WORA" (Viết một lần, chạy mọi nơi). Các chương trình Java được viết trong một hệ thống có thể chạy trên bất kỳ hệ thống nào khác có Java trong đó. Khả năng tương thích của Java không phụ thuộc vào hệ điều hành hoặc phần cứng, điều này làm cho Java trở nên độc lập với nền tảng và siêu linh hoạt.
- Hỗ trợ tính năng di động: Java là một ngôn ngữ có tính di động cao. Điều này là do Java độc lập với nền tảng cũng như không yêu cầu bất kỳ phần cứng đặc biệt nào để chạy. Điều này làm cho Java tương thích với hầu hết mọi thiết bị có thể.
- Hỗ trợ thu gom rác tự động: Trong C hoặc C++, các lập trình viên phải giải phóng dung lượng thông qua chương trình. Còn trong Java, JVM tự động quản lý bộ nhớ. Bất cứ khi nào có một đối tượng không tham chiếu đến bất kỳ lớp nào và cần hủy

- đăng ký, JVM sẽ tự động loại bỏ chúng khỏi chương trình, vì vậy các lập trình viên không cần phải viết thêm bất kỳ mã nào.
- Hỗ trợ đa luồng: Luồng là đơn vị nhỏ nhất của một quá trình. Để đạt được mức sử dụng CPU tối đa, đa luồng là một thành phần quan trọng. Java là ngôn ngữ lập trình hỗ trợ đa luồng. Nó chia sẻ một bộ nhớ chung để tăng hiệu quả và hiệu suất của ứng dụng, do đó chúng ta có thể chạy nhiều luồng cùng một lúc bằng Java.
- Ôn định: Java nhận được các bản cập nhật thường xuyên để loại bỏ các lỗi. Điều này làm cho Java trở thành một trong những ngôn ngữ lập trình ổn định nhất hiện có. Hầu như tất cả các lỗi đều được loại bỏ ngay lập tức thông qua các bản cập nhật. Đó là lý do tại sao việc cập nhật Java thường xuyên là rất quan trọng.
- Cung cấp chiến lược cấp phát bộ nhớ hiệu quả: Java chia bộ nhớ chủ yếu thành hai phần: vùng heap và vùng stack. JVM phân bổ bộ nhớ từ một trong hai phần tùy thuộc vào yêu cầu. Điều này giúp quản lý bộ nhớ hiệu quả.
- Cộng đồng hỗ trợ lớn: Java đã có từ lâu, do đó nó đã xây dựng được một cộng đồng các nhà phát triển lớn hơn bất kỳ ngôn ngữ lập trình nào khác trên thế giới. Với sự giúp đỡ của cộng đồng này và Oracle, Java đã trở thành một ngôn ngữ rất ổn định qua các thời đại. Không giống như các ngôn ngữ lập trình mới như C# hoặc R, Java có hầu hết các truy vấn được giải quyết trong StackOverflow, giúp các nhà phát triển gỡ lỗi mã của họ.
- Java cũng có những nhược điểm riêng, chẳng hạn như:
- Hiệu suất chậm và kém: Java tiêu tốn nhiều bộ nhớ hơn so với các ngôn ngữ lập trình gốc như C và C++. Java cũng chậm hơn so với chúng, điều này là do công việc bổ sung của trình thông dịch để chuyển đổi mã thành ngôn ngữ máy. JVM thực hiện các chức năng phụ trợ khác nhau làm giảm tốc độ của chương trình. Vì Java hỗ trợ thu gom rác tự động nên nó chạy liên tục trong phần backend, gây cản trở hiệu suất.
- GUI kém: Trình tạo GUI trong Java được nhận xét là kém và không thể xây dựng giao diện người dùng phức tạp. Có nhiều framework trong Java để tạo GUI như Swing, JavaFX, SWT, JSF,...nhưng những framework này chưa được phát triển đủ

- để xây dựng các GUI phức tạp. Trong khi đó, các ngôn ngữ hiện đại như Python, R, C#,...có trình tạo GUI tốt hơn.
- Không có phương tiện dự phòng: Java hoàn toàn không có tính năng sao lưu dữ liệu của người dùng. Nó chủ yếu tập trung vào việc lưu trữ dữ liệu, nhưng chúng không được bảo vệ bằng phương tiện sao lưu.
- Giấy phép thương mại trả phí: Oracle đã thông báo vào tháng 1 năm 2019 rằng họ sẽ tính phí cho giấy phép thương mại của Java 11 trở lên. Điều này gây ra rất nhiều lo lắng cho các lập trình viên vì họ phải trả phí dựa trên số lượng hệ thống để nhận các bản cập nhật. Mục đích chung của Java vẫn là miễn phí sử dụng, nhưng để phát triển và truy cập tất cả các tính năng của Java thì bạn phải trả phí.
- Mã dài dòng và phức tạp: Java có nhiều cú pháp dài dòng và phức tạp. Đôi khi, thật khó để nhớ những cú pháp phức tạp đó. Nhiều lập trình viên thích Python hoặc C++ hơn Java, vì chúng có cú pháp đơn giản hơn. Ví dụ:

Để có một đoạn input đơn giản, lập trình viên Java phải viết 2 dòng mã:

- BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

- int num = Integer.parseInt(br.readLine());

Trong khi đó, bạn có thể viết điều tương tự bằng ngôn ngữ C chỉ trong 1 dòng:

Chúng ta có thể thấy mã Java đôi khi có thể hơi phức tạp vì lý do này.

Song, tựu chung lại, Java vẫn là ngôn ngữ lập trình tuyệt vời và đáng để học hỏi. Từ máy tính xách tay đến trung tâm dữ liệu, bảng điều khiển trò chơi đến siêu máy tính khoa học, điện thoại di động đến Internet, Java có mặt ở khắp mọi nơi.

Dựa trên dữ liệu gần đây, Java được 2,6% trang web sử dụng làm ngôn ngữ lập trình phía máy chủ. Mặc dù ngôn ngữ này dường như không được nhiều trang web sử dụng, nhưng các trang web được viết bằng Java đều là các trang web có lưu lượng truy cập cao, chẳng hạn như LinkedIn, Spotify, Netflix, Airbnb, Salesforce.com, Chase, Indeed.com, Pinterest,...Điều này chứng tỏ được tính phổ biến và hiệu quả của nó.

Những công ty nhỏ hơn có thể chọn bắt đầu xây dựng chương trình bằng một ngôn ngữ khác, nhưng khi trang web ngày càng phát triển và đòi hỏi tốc độ cũng như độ ổn định cao hơn thì họ thường chuyển sang lập trình bằng Java. Twitter chính là một minh chứng hoàn hảo.

2.3.3 Java trong lập trình ứng dụng

Java là một ngôn ngữ lập trình phổ biến và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau. Dưới đây là một số lĩnh vực mà Java thường được sử dụng:

- Phát triển ứng dụng doanh nghiệp: Java là ngôn ngữ chính trong phát triển các ứng dụng doanh nghiệp lớn, nhờ vào các framework mạnh mẽ như Spring và Java EE.
- Phát triển ứng dụng web: Với sự hỗ trợ của các framework như Spring MVC và JavaServer Faces (JSF), Java là lựa chọn phổ biến cho việc xây dựng các ứng dụng web manh mẽ và linh hoat.
- Phát triển ứng dụng di động: Java là ngôn ngữ chính thức đầu tiên được sử dụng để phát triển ứng dụng Android, với hàng triệu ứng dụng Android được viết bằng Java.
- Phát triển ứng dụng desktop: Java Swing và JavaFX là hai bộ công cụ phổ biến để phát triển các ứng dụng desktop với giao diện đồ họa.
- Phát triển hệ thống nhúng: Java cũng được sử dụng trong các hệ thống nhúng, nhờ vào khả năng đa nền tảng và hiệu suất cao.

2.3.4 Lợi ích của việc sử dụng Java trong lập trình ứng dụng

- Đa nền tảng: Mã Java đã biên dịch có thể chạy trên mọi nền tảng hỗ trợ JVM, giúp ứng dụng Java có khả năng tương thích cao.
- Bảo mật: Java có các cơ chế bảo mật mạnh mẽ, giúp bảo vệ ứng dụng khỏi các lỗ hồng bảo mật.
- Hiệu suất cao: JVM sử dụng các kỹ thuật như Just-In-Time (JIT) compilation để tối ưu hóa hiệu suất của mã Java.

- Cộng đồng phát triển lớn: Java có một cộng đồng phát triển lớn và năng động, cung cấp nhiều tài liệu, ví dụ, sách và hướng dẫn trực tuyến giúp lập trình viên dễ dàng học hỏi và giải quyết các vấn đề gặp phải.
- Thư viện phong phú: Java cung cấp một loạt các thư viện và framework giúp lập
 trình viên nhanh chóng phát triển các ứng dụng phức tạp.

2.4 Giới thiệu về Node.js

2.4.1 Tổng quan về Node.js

Node.js là một nền tảng mã nguồn mở, đa nền tảng được phát triển bởi Ryan Dahl vào năm 2009. Nó cho phép chạy mã JavaScript bên ngoài trình duyệt web. Node.js được xây dựng trên công cụ JavaScript V8 của Google Chrome, giúp nó thực thi mã JavaScript một cách nhanh chóng và hiệu quả.

Node.js được thiết kế để xây dựng các ứng dụng mạng có khả năng mở rộng cao, cho phép xử lý nhiều kết nối đồng thời với hiệu suất cao. Một trong những điểm mạnh chính của Node.js là kiến trúc hướng sự kiện và mô hình bất đồng bộ (asynchronous), giúp tối ưu hóa việc sử dụng tài nguyên và nâng cao hiệu suất của ứng dụng.

2.4.2 Các tính năng nổi bật của Node.js

- Mô hình bất đồng bộ và hướng sự kiện: Node.js sử dụng mô hình lập trình bất đồng bộ và hướng sự kiện, cho phép xử lý nhiều yêu cầu cùng một lúc mà không bị chặn (non-blocking). Điều này giúp tăng cường khả năng chịu tải và hiệu suất của ứng dụng.
- Nhanh chóng và hiệu quả: Nhờ vào công cụ V8 JavaScript của Google, Node.js có thể thực thi mã JavaScript với tốc độ nhanh chóng và hiệu quả cao.
- Đơn luồng nhưng mạnh mẽ: Mặc dù Node.js chạy trên một luồng duy nhất (single-threaded), nhưng nó có thể xử lý hàng ngàn kết nối đồng thời nhờ vào mô hình sự kiên bất đồng bô.

- NPM (Node Package Manager): NPM là hệ thống quản lý gói cho Node.js, cung cấp hàng nghìn thư viện và module mã nguồn mở giúp lập trình viên dễ dàng mở rộng và phát triển ứng dụng.
- Khả năng mở rộng dễ dàng: Node.js hỗ trợ việc mở rộng ứng dụng bằng cách sử dụng các kỹ thuật như clustering, cho phép chạy nhiều instance của Node.js trên các CPU khác nhau.

2.4.3 Node js làm việc như thế nào?

Node.js hoạt động dựa trên một số nguyên tắc cơ bản giúp nó hiệu quả trong việc xử lý các ứng dụng có nhiều hoạt động nhập/xuất (I/O) mà không bị chặn, đồng thời giảm đáng kể sự phức tạp trong quản lý các luồng thực thi. Dưới đây là một số thành phần chính giải thích cách thức hoạt động của Node.js:

❖ Kiến trúc Non-blocking I/O và Event-Driven:

Node.js sử dụng một mô hình non-blocking I/O (input/output) và event-driven, nghĩa là các hoạt động như đọc file, truy vấn cơ sở dữ liệu, hoặc giao tiếp mạng được thực hiện mà không chặn tiến trình chính. Điều này cho phép xử lý nhiều yêu cầu cùng lúc mà không cần tạo nhiều luồng (thread), giúp giảm bớt chi phí liên quan đến quản lý luồng và tối ưu hóa hiệu suất.

Khi một hoạt động I/O được khởi tạo, nó sẽ được gửi đến thực thi trong hệ thống hoặc cơ sở dữ liệu mà không làm chậm tiến trình chính. Sau khi hoạt động hoàn tất, một sự kiện sẽ được phát đi và xử lý bằng các hàm gọi lại (callback).

❖ V8 JavaScript Engine

Node.js được xây dựng trên động cơ JavaScript V8 của Google Chrome, đây là một động cơ rất nhanh cho phép biên dịch mã JavaScript thành mã máy để thực thi trực tiếp trên phần cứng, làm tăng hiệu suất thực thi.

Single-Threaded

Mặc dù Node.js hoạt động trên một luồng duy nhất cho logic ứng dụng của người dùng, nó vẫn sử dụng nhiều luồng ở tầng thấp hơn thông qua thư viện libuv để xử lý các hoạt động

I/O. Tuy nhiên, những chi tiết này được ẩn giấu khỏi người dùng, giúp việc lập trình đơn giản hơn mà vẫn đảm bảo hiệu suất.

Event Loop

Trái tim của Node.js là "event loop". Đây là vòng lặp sự kiện mà ở đó Node.js tiếp tục lắng nghe sự kiện và thực hiện các hàm gọi lại khi một sự kiện được kích hoạt. Vòng lặp sự kiện cho phép Node.js xử lý hàng nghìn kết nối đồng thời mà không cần phải tạo ra chi phí quản lý luồng.

Trigger Callback

Khi thao tác I/O hoàn tất, hệ điều hành thông báo cho Node.js, và Node.js sau đó thực thi hàm callback tương ứng để xử lý kết quả hoặc tiếp tục xử lý logic.

❖ NPM (Node Package Manager)

NPM là hệ thống quản lý gói cho Node.js, cho phép các nhà phát triển dễ dàng chia sẻ và sử dụng mã nguồn từ nhau. NPM là một trong những kho lưu trữ mã nguồn mở lớn nhất thế giới và chứa hàng ngàn module có thể được tích hợp vào ứng dụng của bạn.

Tổng hợp lại, Node.js mang đến một mô hình hiệu quả và mạnh mẽ cho các ứng dụng web và máy chủ, nhờ khả năng xử lý đồng thời nhiều hoạt động I/O mà không bị chặn, và qua đó tối ưu hóa việc sử dụng tài nguyên và cải thiện hiệu suất.

2.4.4 NPM: hệ thống quản lý gói cho Node.js

Khi thảo luận về Node js thì một điều chắc chắn không nên bỏ qua là xây dựng package quản lý sử dụng các cộng cụ NPM mà mặc định với mọi cài đặt Node js. Ý tưởng của môđun NPM là khá tương tự như Ruby-Gems: một tập hợp các hàm có sẵn có thể sử dụng được, thành phần tái sử dụng, tập hợp các cài đặt dễ dàng thông qua kho lưu trữ trực tuyến với các phiên bản quản lý khác nhau.

Danh sách các mô-đun có thể tìm trên web NPM package hoặc có thể truy cập bằng cách sử dụng công cụ NPM CLI sẽ tự động cài đặt với Node js.

Một số các module NPM phổ biến nhất hiện nay là:

- expressjs.com: Express.js, một Sinatra-inspired web framework khá phát triển của
 Node.js, chứa rất nhiều các ứng dụng chuẩn của Node.js ngày nay.
- Connect: Connect là một mở rộng của HTTP server framework cho Node.js, cung cấp một bộ sưu tập của hiệu suất cao "plugins" được biết đến như là trung gian; phục vụ như một nền tảng cơ sở cho Express
- socket.io and sockjs: Hai thành phần Server-side websockets components nổi tiếng nhất hiện nay.
- Jade: Một trong những engines mẫu, lấy cảm hứng từ HAML, một phần mặc định trong Express.js.
- mongo and mongojs: MongoDB hàm bao để cung cấp các API cho cơ sở dữ liệu đối tượng trong MongoDB Node.js
- redis: thư viện Redis client.
- coffee-script: CoffeeScript trình biên dịch cho phép developers viết các chương trình Node.js của họ dùng Coffee.
- underscore (lodash, lazy): Thư viện tiện ích phổ biến nhất trong JavaScript, package được sử dụng với Node.js, cũng như hai đối tác của mình, hứa hẹn hiệu suất tốt hơn bằng cách lấy một cách tiếp cận thực hiện hơi khác nhau.
- forever: Có lẽ là tiện ích phổ biến nhất để đảm bảo rằng một kịch bản nút cho chạy liên tục. Giữ quá trình Node.js của bạn lên trong sản xuất đối mặt với bất kỳ thất bại không ngờ tới.

2.4.5 Node.js trong phát triển ứng dụng

Node.js là lựa chọn phổ biến cho việc phát triển các ứng dụng web và API nhờ vào tính năng hiệu suất cao và khả năng mở rộng. Dưới đây là một số lĩnh vực mà Node.js thường được sử dụng:

Phát triển ứng dụng web: Với các framework như Express.js, Koa.js và Hapi.js,
 Node.js giúp lập trình viên dễ dàng xây dựng các ứng dụng web mạnh mẽ và linh hoat.

- Phát triển API: Node.js là một lựa chọn lý tưởng cho việc phát triển các RESTful API và GraphQL API, nhờ vào khả năng xử lý đồng thời nhiều yêu cầu và tích hợp dễ dàng với các dịch vụ web khác.
- Phát triển ứng dụng thời gian thực: Node.js rất phù hợp cho các ứng dụng yêu cầu xử lý thời gian thực, chẳng hạn như chat ứng dụng, ứng dụng game trực tuyến và các hệ thống giám sát.
- Úng dụng I/O cao: Các ứng dụng yêu cầu xử lý nhiều I/O, như ứng dụng stream video, xử lý file lớn và ứng dụng mạng, đều được hưởng lợi từ mô hình I/O không đồng bộ của Node.js.

2.4.6 Lợi ích của việc sử dụng Node.js

- Hiệu suất cao: Node.js cho phép xử lý nhiều kết nối đồng thời mà không bị chặn, giúp nâng cao hiệu suất và khả năng chịu tải của ứng dụng.
- Tiết kiệm tài nguyên: Với mô hình bất đồng bộ, Node.js tối ưu hóa việc sử dụng tài nguyên hệ thống, giúp giảm thiểu chi phí và tối ưu hóa hiệu suất.
- Dễ học và sử dụng: Đối với các lập trình viên đã quen thuộc với JavaScript, việc học và sử dụng Node.js trở nên dễ dàng và nhanh chóng.
- Cộng đồng mạnh mẽ: Node.js có một cộng đồng phát triển lớn và năng động, cung cấp nhiều tài liệu, module và thư viện hữu ích giúp lập trình viên dễ dàng phát triển và mở rộng ứng dụng.
- NPM: Hệ thống quản lý gói NPM cung cấp hàng nghìn module mã nguồn mở, giúp lập trình viên dễ dàng tích hợp và sử dụng các chức năng bổ sung trong ứng dụng.

2.4.7 Cách cài đặt Node.js

Để cài đặt Node.js, bạn có thể làm theo các bước sau:

- Tải Node.js từ trang chủ:
- Truy cập vào trang web chính thức của Node.js: https://nodejs.org.
- Tải xuống phiên bản Node.js phù hợp với hệ điều hành của bạn (Windows, macOS, hoặc Linux).

+ Cài đặt Node.js trên Windows:

Mở tệp cài đặt đã tải xuống (ví dụ: node-v14.17.0-x64.msi).

Làm theo các bước hướng dẫn của trình cài đặt.

Sau khi cài đặt xong, mở Command Prompt và nhập node -v để kiểm tra phiên bản Node.js đã cài đặt.

+ Cài đặt Node.js trên macOS:

Mở tệp cài đặt đã tải xuống (ví dụ: node-v14.17.0.pkg).

Làm theo các bước hướng dẫn của trình cài đặt.

Sau khi cài đặt xong, mở Terminal và nhập node -v để kiểm tra phiên bản Node.js đã cài đặt.

+ Cài đặt Node.js trên Linux:

Mở Terminal và nhập các lệnh sau để cài đặt Node.js từ NodeSource:

```
curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Hình 2.2 Cài Node.js

Sau khi cài đặt xong, nhập node -v để kiểm tra phiên bản Node.js đã cài đặt.

+ Cài đặt NPM:

NPM (Node Package Manager) thường đi kèm với Node.js. Để kiểm tra phiên bản NPM, bạn nhập npm -v trong Command Prompt hoặc Terminal.

2.5 Giới thiệu về MySQL

2.5.1 Tổng quan về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển bởi MySQL AB, sau này thuộc sở hữu của Sun Microsystems và hiện tại là một phần của Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để quản lý và truy xuất dữ liệu trong cơ sở dữ liệu.

MySQL nổi bật với khả năng mở rộng, hiệu suất cao và tính dễ sử dụng, là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới, đặc biệt trong các ứng dụng web và dịch vụ trực tuyến.

2.5.2 Các tính năng nổi bật của MySQL

- Mã nguồn mở: MySQL là phần mềm mã nguồn mở, có thể được tải xuống, sử dụng và chỉnh sửa miễn phí theo giấy phép GPL (General Public License).
- Hiệu suất cao: MySQL tối ưu hóa việc thực hiện các truy vấn và xử lý dữ liệu, hỗ trợ nhiều cơ chế lưu trữ và chỉ mục giúp cải thiện hiệu suất.
- Khả năng mở rộng: MySQL có khả năng mở rộng linh hoạt, hỗ trợ các hệ thống từ nhỏ đến lớn, từ các ứng dụng cá nhân đến các ứng dụng doanh nghiệp quy mô lớn.
- Độ tin cậy: MySQL cung cấp các tính năng sao lưu và phục hồi dữ liệu, quản lý giao dịch, và khả năng xử lý lỗi để đảm bảo độ tin cậy của hệ thống.
- Bảo mật: MySQL hỗ trợ nhiều cơ chế bảo mật bao gồm mã hóa dữ liệu, xác thực
 người dùng và phân quyền truy cập để bảo vệ dữ liệu khỏi các truy cập trái phép.
- Dễ dàng tích hợp: MySQL có thể dễ dàng tích hợp với các ngôn ngữ lập trình và công nghệ khác, như PHP, Java, Python và các framework như Laravel, Django, và Spring.

2.4.3 MySQL trong phát triển ứng dụng

MySQL là một lựa chọn phổ biến cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng web và doanh nghiệp. Dưới đây là một số lĩnh vực mà MySQL thường được sử dụng:

- Úng dụng web: MySQL thường được sử dụng để lưu trữ dữ liệu trong các ứng dụng web, bao gồm các trang web thương mại điện tử, hệ thống quản lý nội dung (CMS), và các dịch vụ trực tuyến.
- Úng dụng doanh nghiệp: Các ứng dụng quản lý doanh nghiệp, hệ thống ERP, CRM
 và các công cụ phân tích dữ liệu đều sử dụng MySQL để quản lý và lưu trữ dữ liệu.
- Dịch vụ trực tuyến: MySQL cung cấp nền tảng vững chắc cho các dịch vụ trực tuyến
 như diễn đàn, mạng xã hội, và các ứng dụng chia sẻ nội dung.

2.5.4 Lợi ích của việc sử dụng MySQL

Miễn phí và mã nguồn mở: MySQL là phần mềm mã nguồn mở miễn phí, giúp giảm chi phí và cung cấp khả năng tùy chỉnh theo nhu cầu của người dùng.

- Hiệu suất tốt: MySQL cung cấp hiệu suất cao với khả năng xử lý nhanh chóng các truy vấn và giao dịch.
- Độ tin cậy cao: MySQL hỗ trợ các cơ chế bảo vệ dữ liệu và phục hồi, giúp đảm bảo tính toàn vẹn và độ tin cậy của dữ liệu.
- Cộng đồng lớn: MySQL có một cộng đồng phát triển lớn và năng động, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ kỹ thuật.
- Tích hợp dễ dàng: MySQL dễ dàng tích hợp với nhiều công nghệ và ngôn ngữ lập trình, giúp phát triển ứng dụng nhanh chóng và hiệu quả.

2.5.5 Cách cài đặt MySQL

Để cài đặt MySQL, bạn có thể làm theo các bước sau:

- Tải MySQL từ trang chủ:
- Truy cập vào trang web chính thức của MySQL: https://dev.mysql.com/downloads/.
- Chọn phiên bản MySQL phù hợp với hệ điều hành của bạn (Windows, macOS, hoặc Linux) và tải xuống.
- + Cài đặt MySQL trên Windows:
- Mở tệp cài đặt đã tải xuống (ví dụ: mysql-installer-community-8.0.26.0.msi).
- Làm theo các bước hướng dẫn của trình cài đặt, chọn các thành phần cần thiết và cấu hình thông tin kết nối cơ sở dữ liệu.
- Sau khi cài đặt xong, mở MySQL Workbench (hoặc công cụ quản lý MySQL khác)
 để kết nối và quản lý cơ sở dữ liệu.

2.6 Giới thiệu về SDK của Điện thoại Di động

2.6.1 Tổng quan về SDK

SDK (Software Development Kit) là một tập hợp các công cụ, thư viện, tài liệu, và hướng dẫn mà các nhà phát triển sử dụng để xây dựng ứng dụng cho một nền tảng cụ thể. SDK cung cấp các API, các công cụ lập trình, và môi trường cần thiết để phát triển phần mềm cho các thiết bị di động, như smartphone và tablet.

2.6.2 Các Thành Phần Chính của SDK

- API: Các giao diện lập trình ứng dụng (API) cung cấp các phương thức và lớp để truy cập các tính năng và dịch vụ của hệ điều hành hoặc nền tảng di động. Ví dụ, các API của Android cho phép truy cập vào các dịch vụ như GPS, camera, và cảm biến.
- Thư Viện: Bộ thư viện chứa các mã nguồn, hàm và lớp đã được lập trình sẵn để thực hiện các chức năng cụ thể, như xử lý đồ họa, kết nối mạng, và lưu trữ dữ liệu.
- Trình Biên Dịch và Công Cụ Phát Triển: Các công cụ giúp biên dịch mã nguồn thành mã máy có thể thực thi trên thiết bị di động. Ví dụ, Android SDK bao gồm trình biên dịch dx để chuyển đổi mã Java thành bytecode cho máy ảo Dalvik.
- Môi Trường Phát Triển Tích Hợp (IDE): Một số SDK cung cấp môi trường phát triển tích hợp, như Android Studio cho Android hoặc Xcode cho iOS, giúp nhà phát triển viết, kiểm tra, và gỡ lỗi mã nguồn dễ dàng hơn.
- Tài Liệu và Hướng Dẫn: SDK thường đi kèm với tài liệu chi tiết, ví dụ như hướng dẫn sử dụng API, các ví dụ mã nguồn, và tài liệu kỹ thuật để hỗ trợ các nhà phát triển trong việc sử dụng SDK.
- Công Cụ Gỡ Lỗi và Kiểm Tra: Các công cụ giúp kiểm tra và gỡ lỗi ứng dụng trên thiết bị thật hoặc máy ảo. Ví dụ, Android Studio cung cấp công cụ Debugger để theo dõi và sửa lỗi mã nguồn.
- Mẫu Ứng Dụng và Ví Dụ: SDK thường đi kèm với các mẫu ứng dụng và ví dụ mã nguồn để giúp các nhà phát triển bắt đầu nhanh chóng với các tính năng cơ bản.

2.6.3 Các SDK Chính Cho Di Động

- Android SDK:
- Tổng Quan: Cung cấp công cụ và thư viện để phát triển ứng dụng cho hệ điều hành Android của Google.
- Thành Phần Chính: Android Studio (IDE), Android API, Android Emulator, công cụ gỡ lỗi, và tài liệu chi tiết.
- Tính Năng: Hỗ trợ phát triển ứng dụng với Java, Kotlin, và C++. Cung cấp khả năng truy cập các dịch vụ như GPS, camera, và cảm biến.
- iOS SDK:
- Tổng Quan: Cung cấp công cụ và thư viện để phát triển ứng dụng cho hệ điều hành iOS của Apple.
- Thành Phần Chính: Xcode (IDE), Cocoa Touch (API), iOS Simulator, công cụ gỡ lỗi, và tài liệu chi tiết.
- Tính Năng: Hỗ trợ phát triển ứng dụng với Swift và Objective-C. Cung cấp khả năng truy cập các dịch vụ như GPS, camera, và cảm biến.
- Cross-Platform SDK:
- React Native: Cung cấp công cụ để phát triển ứng dụng di động bằng JavaScript và React. Hỗ trợ cả Android và iOS.
- Flutter: Cung cấp công cụ để phát triển ứng dụng di động bằng Dart. Hỗ trợ cả Android và iOS.
- Xamarin: Cung cấp công cụ để phát triển ứng dụng di động bằng C#. Hỗ trợ cả Android và iOS.
- Cài Đặt Android SDK:

Bước 1: Tải xuống và cài đặt Android Studio từ trang web chính thức của Google.

Bước 2: Trong quá trình cài đặt, Android Studio sẽ tự động cài đặt Android SDK và các công cụ cần thiết.

Bước 3: Sau khi cài đặt, bạn có thể cấu hình và cập nhật SDK thông qua Android Studio bằng cách vào File > Settings > Appearance & Behavior > System Settings > Android SDK.

- Cài Đặt iOS SDK:
- Bước 1: Tải xuống và cài đặt Xcode từ Mac App Store.
- Bước 2: Sau khi cài đặt Xcode, SDK iOS sẽ được cài đặt cùng với Xcode.
- Bước 3: Bạn có thể sử dụng Xcode để quản lý và cập nhật SDK iOS qua Xcode > Preferences > Components.
 - Cài Đặt SDK Cross-Platform:
 - React Native:
- Bước 1: Cài đặt Node.js từ nodejs.org.
- Bước 2: Cài đặt React Native CLI bằng lệnh npm install -g react-native-cli.
- Bước 3: Khởi tạo dự án mới với react-native init ProjectName.
 - Flutter:
- Bước 1: Tải xuống Flutter SDK từ trang web chính thức.
- Bước 2: Giải nén và thêm đường dẫn đến thư mục Flutter vào biến môi trường PATH.
- Bước 3: Chạy lệnh flutter doctor để kiểm tra và cài đặt các công cụ cần thiết.
 - Xamarin:
- Bước 1: Cài đặt Visual Studio với các tùy chọn Xamarin.
- Bước 2: Visual Studio sẽ tự động cài đặt SDK Xamarin cùng với các công cụ cần thiết.

CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

3.1 Khảo sát hệ thống ứng dụng tạo lịch họp trên điện thoại di động

Ứng dụng tạo lịch họp trên điện thoại di động được tổ chức như sau:

- Người dùng: Úng dụng cho phép nhiều người dùng đăng ký và sử dụng, mỗi người dùng có thông tin bao gồm mã người dùng (user_id), tên đăng nhập (username), mật khẩu (password), địa chỉ email (email), họ và tên đầy đủ (full_name), và vai trò (role). Vai trò có thể là 'manager' hoặc 'member', với giá trị mặc định là 'member'. Thời điểm tạo và cập nhật thông tin người dùng cũng được ghi lại.
- Cuộc họp: Mỗi cuộc họp được lưu trữ với thông tin bao gồm mã cuộc họp (meeting_id), tiêu đề (title), ngày diễn ra (meeting_date), thời gian bắt đầu (start_time), thời gian kết thúc (end_time), địa điểm (location), nội dung chính (agenda), tài liệu liên quan (documents), kết quả cuộc họp (result), và thời gian dự kiến cho cuộc họp tiếp theo (next_meeting_time). Thời điểm tạo và cập nhật cuộc họp cũng được ghi lại.
- Người tham gia cuộc họp: Mỗi người tham gia cuộc họp được ghi nhận với mã người tham gia (meeting_participant_id), mã cuộc họp (meeting_id), mã người dùng (user_id), vai trò của người tham gia (role), và trạng thái tham dự (attendance_status). Thời điểm tạo và cập nhật thông tin người tham gia cũng được ghi lại.
- Kết quả cuộc họp: Kết quả cuộc họp bao gồm mã kết quả (result_id), mã cuộc họp (meeting_id), chi tiết kết quả (result), và các yêu cầu mới sau cuộc họp (new_requirements). Thời điểm tạo và cập nhật kết quả cuộc họp cũng được ghi lại.
- Tóm tắt cuộc họp: Các bản tóm tắt cuộc họp bao gồm mã tóm tắt (summary_id), mã cuộc họp (meeting_id), kết luận cuộc họp (conclusion), và mã người điều phối cuộc họp (coordinator_id). Thời điểm tạo và cập nhật tóm tắt cũng được ghi lại.
- Phê duyệt tóm tắt: Thông tin phê duyệt các bản tóm tắt bao gồm mã phê duyệt (approval_id), mã tóm tắt (summary id), mã người dùng (user_id), trạng thái phê

- duyệt (approval_status), và gọi ý xem xét lại (review_suggestion). Thời điểm tạo và cập nhật thông tin phê duyệt cũng được ghi lại.
- Nhắc nhở cuộc họp: Các nhắc nhở cuộc họp bao gồm mã nhắc nhở (reminder_id), mã cuộc họp (meeting_id), thời gian nhắc nhở (reminder_time), mô tả nhắc nhở (reminder_description), mã người dùng nhận nhắc nhở (user_id), và trạng thái nhắc nhở (status). Thời điểm tạo và cập nhật nhắc nhở cũng được ghi lại.

3.2 Các chức năng của hệ thống.

3.2.1 Yêu cầu chức năng của hệ thống:

- ❖ Đối tượng tham gia hệ thống:
- Người quản lý cuộc họp.
- Thành viên tham gia cuộc họp.
- * Các chức năng của người quản lý (Manager):
- Quản lý thông tin người dùng: Xem thông tin người dùng
- Tạo và quản lý cuộc họp: Tạo cuộc họp mới với các thông tin như tiêu đề, ngày, thời gian, địa điểm, nội dung chính, tài liệu liên quan.
- Quản lý người tham gia cuộc họp
- Quản lý kết quả và yêu cầu mới từ cuộc họp:
- Tạo và cập nhật kết quả cuộc họp và các yêu cầu mới.
- Quản lý tóm tắt cuộc họp: Tạo và cập nhật các bản tóm tắt cuộc họp và phê duyệt các bản tóm tắt.
- Quản lý nhắc nhở cuộc họp
- ❖ Các chức năng của thành viên (Member):
- Xem thông tin cuộc họp: Xem các thông tin chi tiết về cuộc họp, bao gồm tiêu đề,
 ngày, thời gian, địa điểm, và nội dung chính.

- Tham gia và quản lý cuộc họp: Đưa ra ý kiến về các kết luận cuộc họp (tick vào phần phê duyệt kết luận). Đề xuất xem xét lại các kết luận nếu cần thiết.

3.3 Bối cảnh của hệ thống

3.3.1 Định nghĩa vấn đề mà đề tài sẽ giải quyết

Mục tiêu của đề tài là phát triển một ứng dụng di động giúp người dùng quản lý và tổ chức các cuộc họp hiệu quả. Hiện tại, quản lý lịch họp chủ yếu dựa vào các phương pháp thủ công hoặc công cụ không chuyên biệt. Ứng dụng di động sẽ giải quyết các vấn đề hiện tại bằng cách cung cấp một nền tảng tích hợp để:

- Lên lịch cuộc họp, bao gồm các thông tin như tiêu đề, thời gian, địa điểm và tài liệu liên quan.
- Theo dõi và quản lý danh sách người tham gia.
- Gửi thông báo và nhắc nhở về cuộc họp cho các thành viên.
- Ghi chú và theo dõi các điểm chính và quyết định trong cuộc họp.
- Tạo báo cáo và phân tích kết quả cuộc họp.

3.3.2 Hiện trạng trước khi sử dụng phần mềm:

Trước khi có phần mềm, quy trình quản lý lịch họp diễn ra như sau:

- ❖ Người dùng (Nhân viên, Quản lý):
- Hiện trạng: Sử dụng các công cụ như email, lịch giấy hoặc các phần mềm lịch cá nhân để tổ chức và quản lý cuộc họp. Việc tạo và gửi lời mời họp, cũng như cập nhật thông tin về cuộc họp và người tham gia, thường diễn ra qua email hoặc thông báo qua tin nhắn.
- Vấn đề: Quy trình này thường gây khó khăn trong việc theo dõi các thay đổi, nhắc nhở và cập nhật thông tin cho tất cả các bên liên quan. Thiếu khả năng tích hợp giữa các công cụ và thiếu tính năng tự động hóa dẫn đến sự thiếu chính xác và hiệu quả kém.

- ❖ Người tham gia cuộc họp:
- Hiện trạng: Nhận thông báo cuộc họp qua email hoặc tin nhắn, phải thường xuyên kiểm tra và cập nhật thông tin về cuộc họp bằng tay.
- Vấn đề: Khó khăn trong việc theo dõi lịch trình và thông tin cuộc họp. Không có cách dễ dàng để thêm, xóa hoặc cập nhật thông tin tham gia và ghi chú cuộc họp.
- Quản lý (Người tổ chức cuộc họp):
- Hiện trạng: Tạo và quản lý lịch họp bằng các công cụ thủ công hoặc phần mềm lịch cá nhân. Phải tự thực hiện việc gửi thông báo, theo dõi sự tham gia và quản lý tài liệu.
- Vấn đề: Thiếu khả năng theo dõi và quản lý cuộc họp một cách tập trung và tự động,
 dẫn đến việc quản lý kém hiệu quả và khả năng theo dõi không đầy đủ.

3.3.3 Đề xuất giải pháp của đề tài

Khi so sánh việc thiết lập cuộc họp bằng ứng dụng quản lý cuộc họp với các phương pháp tổ chức họp truyền thống như gửi email, ứng dụng này mang lại nhiều ưu điểm vượt trội. Dưới đây là ba ưu điểm chính của việc sử dụng ứng dụng quản lý cuộc họp:

- 1. Đảm Bảo Nhất Quán về Nội Dung Họp
- Chia Sẻ Công Khai Ý Kiến & Tài Liệu: Ứng dụng quản lý cuộc họp cho phép các thành viên dễ dàng truy cập vào tài liệu, ý kiến, và thông tin liên quan đến cuộc họp từ một nguồn chung, thường là trên nền tảng web hoặc ứng dụng di động. Điều này giúp đảm bảo rằng mọi người đều có thông tin nhất quán và cập nhật nhất về cuộc họp.
- Truy Cập Tài Liệu và Dẫn Chứng: Tài liệu và thông tin dẫn chứng có thể được chia sẻ công khai và dễ dàng truy cập qua ứng dụng, giúp giảm thiểu sự thiếu sót thông tin và đảm bảo moi thành viên đều có thể tham khảo tài liêu cần thiết.

- Lịch Sử và Ghi Chép: Ứng dụng lưu trữ toàn bộ thông tin cuộc họp, bao gồm biên bản, quyết định, và ý kiến, giúp việc tra cứu và quản lý thông tin trở nên dễ dàng và chính xác hơn so với việc sử dụng email hoặc phương pháp truyền thống khác.

2. Liên Kết Chuỗi Các Cuộc Họp

- Theo Dõi Tiến Độ và Mục Tiêu: Ứng dụng giúp liên kết các cuộc họp với nhau, theo dõi tiến độ và các mục tiêu đã được thiết lập. Bạn có thể dễ dàng xem lịch sử các cuộc họp và các kết quả đạt được, điều này giúp duy trì sự liên kết và tiếp nối giữa các cuộc họp để đạt được mục tiêu cuối cùng.
- Quản Lý Nhiệm Vụ và Hành Động: Các cuộc họp có thể được liên kết với các nhiệm vụ cụ thể, giúp theo dõi và quản lý tiến độ của các công việc đã được thảo luận và phân công trong các cuộc họp trước đó.
- Tạo Lịch Trình và Nhắc Nhỏ: Ứng dụng có thể tự động lên lịch các cuộc họp tiếp theo và gửi nhắc nhỏ cho các thành viên, đảm bảo rằng các cuộc họp tiếp theo diễn ra đúng thời gian và không bị bỏ lỡ.

3. Tự Động Hóa Việc Nhắc Nhở Đi Họp

- Nhắc Nhở Tự Động: Ứng dụng có thể tự động gửi nhắc nhở qua email, SMS, hoặc thông báo trên ứng dụng để đảm bảo các thành viên nhớ tham dự cuộc họp. Điều này giúp giảm thiểu sự quên lãng và tăng cường tính chính xác của lịch trình cuộc họp.
- Tùy Chỉnh Thông Báo: Các thông báo nhắc nhở có thể được tùy chỉnh theo yêu cầu, bao gồm việc gửi thông báo trước khi cuộc họp bắt đầu, gửi nhắc nhở cho các nhiệm vụ đã được phân công, và thông báo các cập nhật quan trọng.
- Giảm Thiểu Công Sức Quản Lý Thủ Công: Việc tự động hóa các thông báo và nhắc nhở giúp giảm thiểu công sức quản lý và tổ chức cuộc họp, cho phép người tổ chức tập trung vào nội dung và chất lượng cuộc họp thay vì quản lý các chi tiết nhỏ.

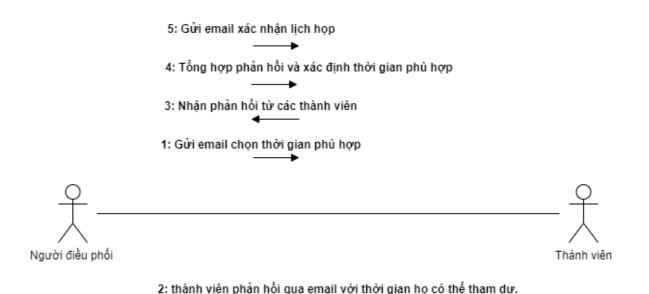
Những ưu điểm này giúp tối ưu hóa quy trình tổ chức cuộc họp, nâng cao hiệu quả làm việc và cải thiện sự hợp tác giữa các thành viên trong tổ chức.

- ❖ Một số tình huống chính (use case) trong bối cảnh này (5 tình huống)
- Lập lịch họp: Phần mềm cần cung cấp giao diện đơn giản cho người dùng để họ có thể dễ dàng tạo và quản lý lịch trình họp. Yêu cầu nghiệp vụ bao gồm khả năng đồng bộ hóa với lịch cá nhân của các thành viên, cung cấp thông tin chi tiết về thời gian, địa điểm và mục đích của cuộc họp.
- Quản lý danh sách tham dự: Phần mềm cần hỗ trợ việc quản lý danh sách tham dự bằng cách cho phép người dùng thêm, xóa và cập nhật thông tin thành viên. Yêu cầu nghiệp vụ bao gồm tính năng gửi mời tự động và theo dõi trạng thái tham dự.
- Gửi kết quả họp: Phần mềm cần hỗ trợ tự động tạo và gửi kết quả họp và biên bản ghi chép cho các thành viên liên quan, đảm bảo tính bảo mật và tính toàn vẹn của thông tin.
- Đánh giá cuộc họp: Cung cấp giao diện để người dùng có thể đánh giá cuộc họp và cung cấp phản hồi. Phần mềm cần có tính năng phân tích phản hồi để đưa ra các đề xuất cải tiến.
- Gửi nhắc nhỏ: Tạo cơ chế tự động để gửi nhắc nhỏ cho các thành viên về các cuộc họp sắp tới và các nhiệm vụ quan trọng.

3.3.3.1 Use Case lập lịch họp

• Khi chưa có PM:

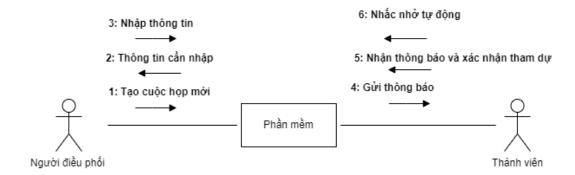
- 1. Người điều phối cuộc họp gửi email đến các thành viên để chọn thời gian phù hợp.
- 2. Nhận phản hồi từ các thành viên qua email.
- 3. Tổng hợp phản hồi và xác định thời gian phù hợp.
- 4. Gửi email xác nhận lịch họp cho các thành viên.
- Lược đồ cộng tác chi tiết:



Sơ đồ 3.1 Lược đồ công tác lập lịch họp

• Khi có PM:

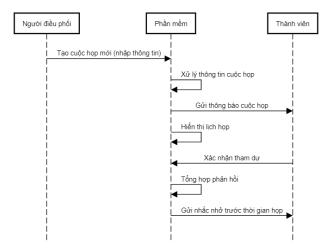
- 1. Người điều phối sử dụng PM để tạo cuộc họp mới và nhập thông tin cuộc họp (thời gian, địa điểm, mục đích).
- 2. PM gửi thông báo đến các thành viên và hiển thị lịch họp trong hệ thống.
- 3. Các thành viên xác nhận tham dự trực tiếp trên PM.
- 4. PM tự động tổng hợp phản hồi và gửi nhắc nhở tới các thành viên trước thời gian họp.
- Lược đồ cộng tác chi tiết



Sơ đồ 3.2 Lược đồ cộng tác lập lịch họp khi có phần mềm

Mô tả bằng lời:

Người điều phối cần tạo một cuộc họp mới. Họ mở ứng dụng PM, nhập thông tin về cuộc họp (thời gian, địa điểm, mục đích) và thêm danh sách người tham gia. Hệ thống PM sẽ gửi thông báo đến các thành viên và yêu cầu họ xác nhận tham dự.



Sơ đồ 3.3 Sơ đồ tuần tự lập lịch họp

3.3.3.2 Use Case quản lý danh sách tham dự

Khi chưa có PM:

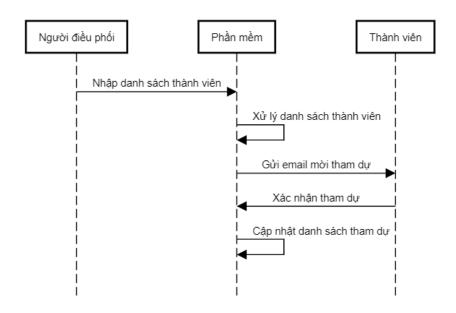
- 1. Người điều phối gửi email mời tham dự đến các thành viên.
- 2. Các thành viên trả lời email xác nhận tham dự.
- 3. Người điều phối cập nhật danh sách tham dự thủ công trong file Excel hoặc tài liệu tương tự.

• Khi có PM:

- 1. Người điều phối nhập danh sách thành viên vào hệ thống PM.
- 2. PM tự động gửi email mời tham dự đến các thành viên.
- 3. Các thành viên xác nhận tham dự trực tiếp trên PM.
- 4. PM tự động cập nhật danh sách tham dự

Mô tả bằng lời:

Người điều phối nhập danh sách các thành viên vào hệ thống PM. Hệ thống tự động gửi email mời tham dự đến các thành viên. Các thành viên xác nhận tham dự trực tiếp trên PM, và PM cập nhật danh sách tham dự tự động



Sơ đồ 3.4 Sơ đồ tuần tự quản lý danh sách tham dự

3.3.3.3 Use Case gửi kết quả họp

• Khi chưa có PM:

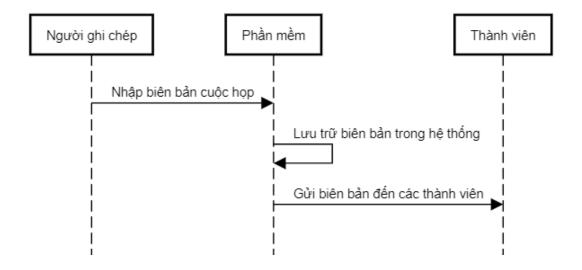
- 1. Người ghi chép viết biên bản cuộc họp.
- 2. Gửi email biên bản đến các thành viên.
- 3. Các thành viên lưu trữ biên bản trong hệ thống riêng của họ.

• Khi có PM:

- 1. Người ghi chép nhập biên bản cuộc họp vào hệ thống PM.
- 2. PM tự động gửi biên bản đến các thành viên và lưu trữ biên bản trong hệ thống.

• Mô tả bằng lời:

Sau cuộc họp, người ghi chép nhập biên bản cuộc họp vào hệ thống PM. Hệ thống PM tự động gửi biên bản đến các thành viên và lưu trữ biên bản trong hệ thống



Sơ đồ 3.5 Sơ đồ tuần tự gửi kết quả cuộc họp

3.3.3.4 Use Case đánh giá cuộc họp

• Khi chưa có PM:

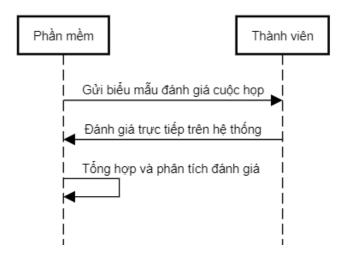
- 1. Người điều phối gửi email khảo sát đánh giá cuộc họp đến các thành viên.
- 2. Các thành viên trả lời email với đánh giá của họ.
- 3. Người điều phối tổng hợp đánh giá thủ công.

• Khi có PM:

- 1. PM tự động gửi biểu mẫu đánh giá cuộc họp đến các thành viên.
- 2. Các thành viên đánh giá trực tiếp trên hệ thống PM.
- 3. PM tự động tổng hợp và phân tích đánh giá.

• Mô tả bằng lời:

Hệ thống PM tự động gửi biểu mẫu đánh giá cuộc họp đến các thành viên. Các thành viên đánh giá trực tiếp trên hệ thống PM. Hệ thống tự động tổng hợp và phân tích đánh giá.



Sơ đồ 3.6 Sơ đồ tuần tự gửi kết đánh giá cuộc họp

3.3.3.5 Use Case gửi nhắc nhở

• Khi chưa có PM:

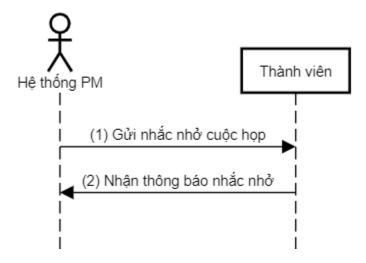
- 1. Người điều phối gửi email nhắc nhở cuộc họp đến các thành viên.
- 2. Các thành viên kiểm tra email để nắm bắt thông tin nhắc nhở.

• Khi có PM:

- 1. PM tự động gửi nhắc nhở đến các thành viên theo thời gian đã được cấu hình.
- 2. Các thành viên nhận thông báo nhắc nhở trên hệ thống PM hoặc qua email/sms

• Mô tả bằng lời:

Hệ thống PM tự động gửi nhắc nhỏ đến các thành viên theo thời gian đã được cấu hình. Các thành viên nhận thông báo nhắc nhỏ trên hệ thống PM hoặc qua email/sms



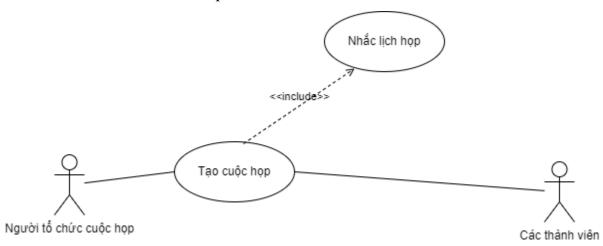
Sơ đồ 3.7 Sơ đồ tuần tự gửi nhắc nhở

3.4 Định nghĩa các tương tác cần thiết trên PM

3.4.1. Lược đồ Use Case

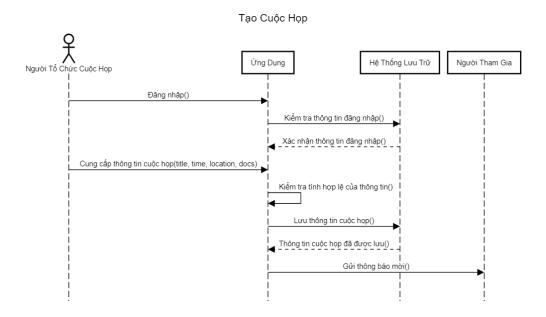
3.4.1.1 Người tổ chức cuộc họp (Meeting Organizer)

❖ Use Case 1: Tạo Cuộc họp



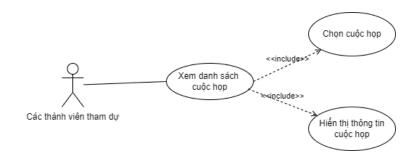
Sơ đồ 3.8. Sơ đồ Use Case 1: tao cuộc họp

- + **ID:** UC01
- + **Ý Nghĩa:** Người tổ chức cuộc họp muốn lên lịch một cuộc họp mới với các thông tin chi tiết như tiêu đề, thời gian, địa điểm và tài liệu liên quan...
- + **Actor:** Người Tổ Chức Cuộc Họp
- + Điều Kiện: Người tổ chức đã đăng nhập vào ứng dụng.
- + Tương Tác:
 - 1. Người Tổ Chức Cuộc Họp cung cấp thông tin cuộc họp (tiêu đề, thời gian, địa điểm, tài liệu, ...) cho ứng dụng.
 - 2. Ứng dụng kiểm tra tính hợp lệ của thông tin và lưu cuộc họp vào hệ thống.
 - 3. Ứng dụng gửi thông báo mời đến các người tham gia được chỉ định.



Sơ đồ 3.9 Sơ đồ tuần tự tạo cuộc họp

Use Case 2: Xem danh sách cuộc họp



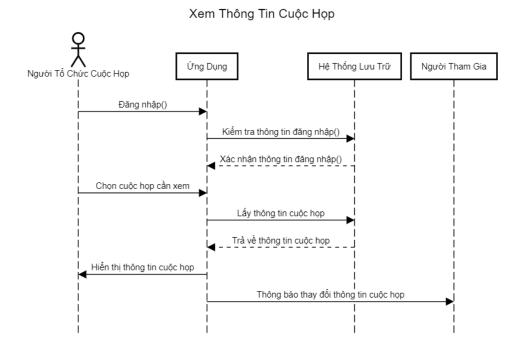
Sơ đồ 3.10 Sơ đồ Use Case 2: xem danh sách cuộc họp

- + **ID:** UC02
- + **Ý Nghĩa:** Người tổ chức muốn xem thông tin của một cuộc họp đã được tạo trước đó.
- + **Actor:** Người Tổ Chức Cuộc Họp

 Điều Kiện: Người tổ chức đã đăng nhập vào ứng dụng và có quyền chỉnh xem thông tin cuộc họp.

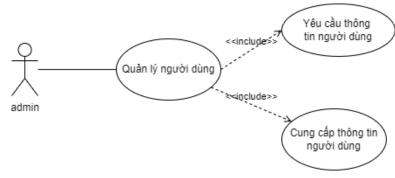
+ Tương Tác:

- 1. Người Tổ Chức Cuộc Họp chọn cuộc họp cần xem.
- 2. Úng dụng cập nhật thông tin cuộc họp và thông báo thay đổi đến người tham gia.



Sơ đồ 3.11 Sơ đồ tuần tự xem thông tin cuộc họp

Use Case 3: Quản lý người dùng

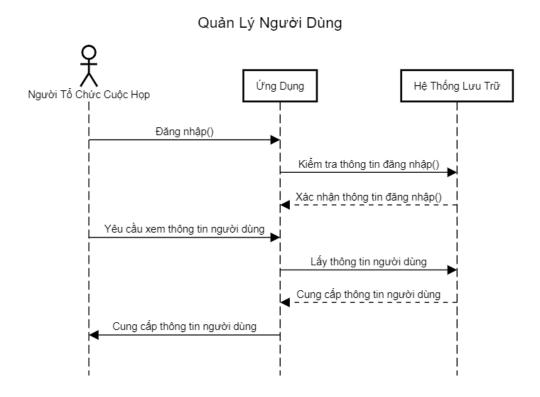


Sơ đồ 3.12 Sơ đồ Use Case 3: Quản lý người dung

- + **ID:** UC03
- + Ý Nghĩa: Người tổ chức muốn muốn xem và quản lý thông tin người dung
- + **Actor:** Người Tổ Chức Cuộc Họp
- + Điều Kiện: Người tổ chức đã đăng nhập vào ứng dụng và có quyền truy cập báo cáo.

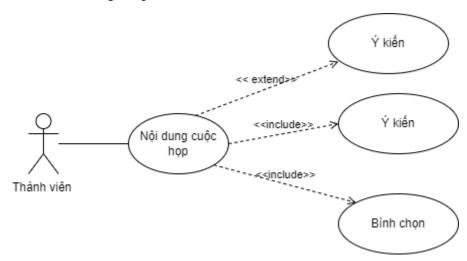
+ Tương Tác:

- 1. Người tổ chức yêu cầu xem thông tin người dùng từ ứng dụng.
- 2. Úng dụng cung cấp thông tin người dùng cho người tổ chức.



Sơ đồ 3.13 Sơ đồ tuần tự quản lý người dung

Use Case 4: Nội Dung Họp

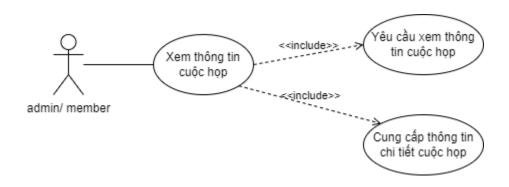


Sơ đồ 3.14 Sơ đồ tuần tự nội dung họp

- + ID: UC04
- + Ý Nghĩa: Use Case này liên quan đến việc xử lý nội dung cuộc họp, bao gồm việc nêu ý kiến, bình chọn, và đưa ra kết luận. Tính năng này cho phép tất cả các thành viên tham gia cuộc họp có thể đóng góp ý kiến, xem và bình chọn các đề xuất, và cuối cùng đưa ra kết luận và xác nhận thông qua hệ thống PM.
- + Actors:
- Chủ trì (Meeting Organizer): Người điều hành và kết luận cuộc họp.
- Các Thành Viên (Participants): Những người tham gia cuộc họp, có thể đưa ra ý kiến và bình chon.
- + Các Bước Chính:
- Chủ trì khởi xướng nội dung cuộc họp, yêu cầu các thành viên nêu ý kiến và tài liệu dẫn chứng.
- Các Thành Viên sử dụng hệ thống PM để gửi ý kiến, bình chọn, và kèm theo tài liệu dẫn chứng
- Chủ trì tổng hợp các ý kiến và kết quả bình chọn.
- Chủ trì đưa ra kết luận cuộc họp.
- Các Thành Viên ký tên xác nhận kết luận cuộc họp.

3.4.1.2 Người Tham Gia Cuộc Họp (Meeting Participant)

❖ Use Case 5: Xem Thông Tin Cuộc Họp



Sơ đồ 3.14 Sơ đồ Use Case 5: Xem Thông Tin Cuộc Họp

- + **ID:** UC05
- + **Ý Nghĩa:** Người tham gia muốn xem thông tin chi tiết của cuộc họp mà họ đã được mời.
- + **Actor:** Người Tham Gia Cuộc Họp
- + Điều Kiện: Người tham gia đã nhận được lời mời và đã đăng nhập vào ứng dụng.
- + Tương Tác:
 - 1. Người Tham Gia Cuộc Họp yêu cầu xem thông tin cuộc họp từ ứng dụng.
 - 2. Úng dụng cung cấp thông tin chi tiết về cuộc họp.

Người Tham Gia Cuộc Họp Dăng nhập() Kiểm tra thông tin đăng nhập() Yêu cầu xem thông tin cuộc họp Lấy thông tin cuộc họp Cung cấp thông tin chi tiết cuộc họp

Xem Thông Tin Cuộc Họp

Sơ đồ 3.15 Sơ đồ tuần tự xem thông tin cuộc họp

3.5 Định nghĩa yêu cầu và ràng buộc đối với phần mềm

3.5.1 Yêu cầu từ môi trường nghiệp vụ (business)

UseCase	Req-ID	Nội dung yêu cầu	Stack-Holder
U01	B01.1	Quản lý cần tạo cuộc họp với các thông tin chi tiết	Quản lý
		như thời gian, địa điểm, vấn đề cần thảo luận, tài	
		liệu họp, kết quả cần đạt được, và các thành viên	
		tham dự.	
U01	B01.2	Quản lý cần cập nhật kết quả của cuộc họp, yêu cầu	Quản lý
		phát sinh và lên kế hoạch cho cuộc họp tiếp theo.	
U02	B02.1	Thành viên cần xem chi tiết cuộc họp để hiểu rõ các	Thành viên
		vấn đề và thông tin liên quan.	

U02	B02.2	Thành viên cần bổ sung ý kiến và nhận xét về các	Thành viên
		vấn đề trong cuộc họp.	
U03	B03.1	Ứng dụng cần hỗ trợ chia sẻ lịch họp với tất cả thành	Quản lý và
		viên tham gia để đảm bảo họ được thông báo đầy	Thành viên
		đủ.	
U03	B03.2	Ứng dụng cần gửi thông báo nhắc nhở lịch họp qua	Quản lý và
		ứng dụng để đảm bảo các thành viên không quên	Thành viên
		cuộc họp.	

Bảng 3.1 Yêu cầu từ môi trường nghiệp vụ

3.5.2 Yêu cầu từ môi trường vận hành (operation)

3.5.2.1 Các yêu cầu chức năng:

UseCase	ID	Nội dung yêu cầu	Stack-Holder
U01	F01.01	Cuộc họp chỉ được tạo khi tất cả thông tin cần thiết được	Quản lý
		cung cấp (thời gian, địa điểm, thành viên, tài liệu, v.v.).	
U01	F01.02	Quản lý cần có khả năng cập nhật thông tin cuộc họp,	Quản lý
		bao gồm kết quả và các yêu cầu phát sinh.	
U02	F02.01	Thành viên có thể xem chi tiết cuộc họp, bao gồm các	Thành viên
		vấn đề cần thảo luận và tài liệu liên quan.	
U02	F02.02	Thành viên có thể bổ sung ý kiến và nhận xét về cuộc	Thành viên
		họp.	
U03	F03.01	Hệ thống phải cho phép chia sẻ lịch họp qua các kênh	Quản lý
		khác nhau như email hoặc thông báo trong ứng dụng.	
U03	F03.02	Ứng dụng cần gửi thông báo nhắc nhở lịch họp tự động	Quản lý
		dựa trên thời gian đã được cấu hình trước đó.	

Bảng 3.2 Các yêu cầu chức năng

3.5.2.1 Các yêu cầu chất lượng:

UseCase	ID	Nội dung yêu cầu	Stack-Holder
U01	NF01.01	Hiệu suất hệ thống: Thời gian phản hồi ≤ 2 giây cho	Quản lý
		mỗi thao tác và hệ thống cần có khả năng xử lý khoảng	
		50 cuộc họp/ngày.	
U01	NF01.02	Dễ bảo trì: Hệ thống cần có khả năng tái sử dụng cao,	Quản lý
		dễ dàng sửa đổi và kiểm thử.	
U02	NF02.01	Độ tin cậy: Hệ thống cần đảm bảo tính chính xác trong	Thành viên
		việc hiển thị thông tin cuộc họp và nhận xét của thành	
		viên.	
U03	NF03.01	Bảo mật: Hệ thống cần đảm bảo thông tin cuộc họp và	Quản lý và
		nhận xét được bảo vệ an toàn và chỉ có người dùng	Thành viên
		được phép truy cập.	

Bảng 3.3 Các yêu cầu chất lượng

3.5.3 Yêu cầu từ môi trường phát triển (development)

3.5.3.1 Ràng buộc về phương pháp/công nghệ:

ID	Đối tượng	Nội dung yêu cầu	Stack-Holder
TD01	Software	Phát triển ứng dụng sử dụng Android SDK cho môi	Dev team
		trường di động.	

TD02	Software	Sử dụng công cụ Android Studio để viết mã và quản lý	Dev team
		dự án.	
TD03	Software	Áp dụng kiến trúc MVVM (Model-View-ViewModel)	Dev team
		để xây dựng ứng dụng.	

Bảng 3.4 Ràng buộc về phương pháp/công nghệ

3.5.3.2 Ràng buộc về kỹ thuật/công nghệ:

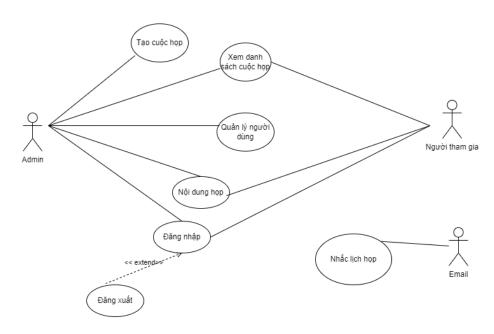
ID	Đối tượng	Nội dung yêu cầu	Stack-Holder
TA01	Application	Hệ thống cần chạy trên các thiết bị Android với hệ điều	Khách hàng
		hành từ Android 8.0 trở lên.	
TA02	Application	Yêu cầu cấu hình thiết bị: CPU 8 nhân, RAM 4GB, bộ	Khách hàng
		nhớ trong 64GB.	
TA03	Network	Ứng dụng cần hỗ trợ kết nối Internet qua Wi-Fi và dữ	Khách hàng
		liệu di động.	

Bảng 3.5 Ràng buộc về phương pháp/công nghệ

CHƯƠNG 4: THIẾT KẾ PHẦN MỀM

4.1 Lược đồ usecase

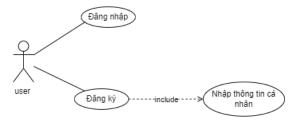
4.1.1 Use case tổng quát



Sơ đồ 4.1 Use case tổng quát

- + Người Quản Lý có thể thực hiện các chức năng: Tạo Cuộc Họp, Cập Nhật Kết Quả Cuộc Họp, Yêu Cầu Phát Sinh, Tạo Cuộc Họp Tiếp Theo, và Chia Sẻ Lịch Họp.
- + Thành Viên có thể thực hiện các chức năng: Xem Chi Tiết Cuộc Họp, Bổ Sung Ý Kiến, và Chia Sẻ Lịch Họp.
- + Hệ Thống sẽ tự động thực hiện chức năng Nhắc Lịch Họp.

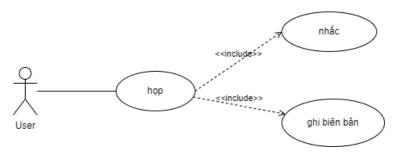
4.1.2 Use case đăng nhập, đăng ký



Sơ đồ 4.2 Use case đăng nhập, đăng ký

- + Người dùng sẽ điền thông tin cá nhân để đăng ký vào hệ thống.
- + Người dùng sẽ nhập thông tin và mật khẩu để đăng nhập vào hệ thống, người có quyền manager sẽ vào trang chủ admin còn người có quyền member sẽ vào trang người dùng.

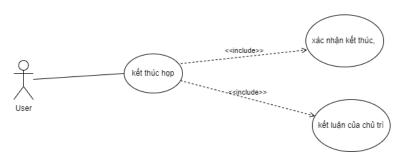
4.1.3 Use case hop



Sơ đồ 4.3 Use case họp

Người dùng sẽ được nhắc nhỏ khi cuộc họp sắp xảy ra. Và sẽ được điền kết luận của cuộc họp

4.1.4 Use case kết thúc cuộc họp

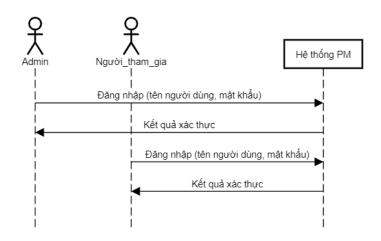


Sơ đồ 4.4 Use case kết thúc cuộc họp

Người dùng sẽ xác nhận kết thúc cuộc họp, và Manager sẽ đưa ra kết luận của cuộc họp cho người dùng tick vào có đồng ý hay không nếu không đồng ý thì sẽ nêu ý kiến.

4.2 Sơ đồ tuần tự

4.2.1 Sơ đồ tuần tự đăng nhập



Sơ đồ 4.5 Sơ đồ tuần tự đăng nhập

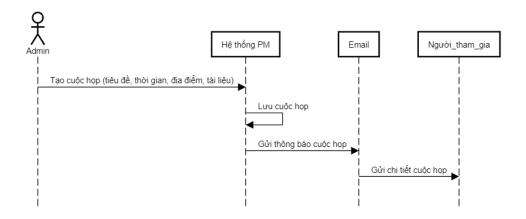
- * Actors:
- + Admin (người quản trị hệ thống)
- + Người_tham_gia (thành viên tham dự cuộc họp)

Tương tác:

- + Admin -> PM: Admin gửi thông tin đăng nhập tới hệ thống PM.
- + PM -> Admin: Hệ thống PM trả về kết quả xác thực cho Admin.
- + Người_tham_gia -> PM: Người tham gia gửi thông tin đăng nhập tới hệ thống PM.
- + PM -> Người_tham_gia: Hệ thống PM trả về kết quả xác thực cho Người tham gia.

Mục đích: Đảm bảo cả Admin và Người tham gia đều có thể đăng nhập vào hệ thống với thông tin hợp lệ.

4.2.2 Sơ đồ tuần tự tạo cuộc họp



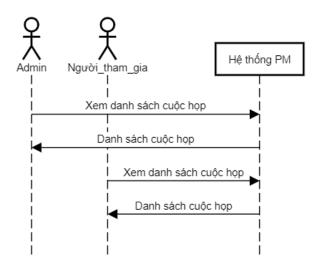
Sơ đồ 4. 6 Sơ đồ tuần tự tạo cuộc họp

Giải thích:

- * Actor: Admin (người quản trị).
- Participants:
- + Hệ thống PM (phần mềm quản lý).
- + Email (dịch vụ gửi email thông báo).
- * Tương tác:
- + Admin -> PM: Admin gửi thông tin tạo cuộc họp (tiêu đề, thời gian, địa điểm, tài liệu) tới hệ thống PM.
- + PM -> PM: Hệ thống PM lưu thông tin cuộc họp vào cơ sở dữ liệu.
- + PM -> Email: Hệ thống PM yêu cầu dịch vụ email gửi thông báo về cuộc họp.
- + Email -> Người_tham_gia: Dịch vụ email gửi chi tiết cuộc họp đến các người tham gia.

Mục đích: Tạo và thông báo cuộc họp đến các thành viên.

4.2.3 Sơ đồ tuần tự xem danh sách cuộc họp

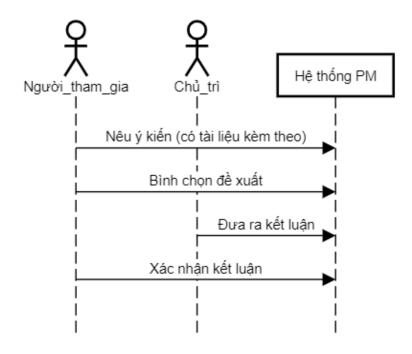


Sơ đồ 4.7 Sơ đồ tuần tự xem danh sách cuộc họp

- **Actors:**
- + Admin: Người quản trị hệ thống.
- + Người_tham_gia: Thành viên tham dự cuộc họp.
- Participant:
- + Hệ thống PM: Hệ thống quản lý cuộc họp.
- Tương tác:
- + Admin -> PM: Admin yêu cầu xem danh sách cuộc họp từ hệ thống PM.
- + PM -> Admin: Hệ thống PM gửi danh sách cuộc họp cho Admin.
- + Người_tham_gia -> PM: Người tham gia yêu cầu xem danh sách cuộc họp từ hệ thống PM.
- + PM -> Người_tham_gia: Hệ thống PM gửi danh sách cuộc họp cho Người tham gia.

Mục đích: Đảm bảo rằng cả Admin và Người tham gia đều có thể truy cập và xem danh sách cuộc họp từ hệ thống PM.

4.2.4 Sơ đồ tuần tự nội dung họp



Sơ đồ 4.8 Sơ đồ tuần tự nội dung họp

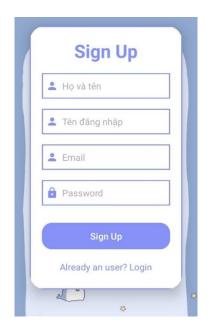
- **Actors:**
- + Người_tham_gia (thành viên tham gia).
- + Chủ_trì (người chủ trì cuộc họp).
- Participant: Hệ thống PM.
- * Tương tác:
- + Người_tham_gia -> PM: Người tham gia gửi ý kiến và tài liệu kèm theo lên hệ thống PM.
- + Người_tham_gia -> PM: Người tham gia thực hiện bình chọn các đề xuất trong cuộc họp.
- + Chủ_trì -> PM: Chủ trì cuộc họp đưa ra kết luận cuối cùng.
- + Người_tham_gia -> PM: Các thành viên xác nhận kết luận của cuộc họp.

Mục đích: Đảm bảo tất cả các thành viên có thể đóng góp ý kiến, bình chọn và xác nhận kết luận cuối cùng của cuộc họp thông qua hệ thống.

4.3 Thiết kế các thành phần của phần mềm

4.3.1 Đăng ký

4.3.1.1 Form: Đăng Ký



Hình 4.1Đăng Ký

- ❖ Giao tiếp:
- + Actor: Người dùng mới
- + Vai trò: Người dùng muốn tạo tài khoản mới.
- ❖ Các Control:
- Control 1: Họ và tên
 - + Inputs: Họ và tên của người dùng.
 - + Outputs: Họ và tên được nhập vào form.
- Control 2: Tên đăng nhập
 - + Inputs: Tên đăng nhập mới của người dùng.
 - + Outputs: Tên đăng nhập được nhập vào form.

- Control 3: Gmail
 - + Inputs: Gmail mới của người dùng.
 - + Outputs: Gmail được nhập vào form.
- Control 4: Mât khẩu
 - + Inputs: Mật khẩu mới của người dùng.
 - + Outputs: Mật khẩu được nhập vào form.
- Control 5: Submit Button
 - + Inputs: Dữ liệu từ các input fields (username, password, confirmPassword)
 - + Outputs: Gửi dữ liệu đến API để xử lý đăng ký và nhận kết quả trả về (thành công hoặc lỗi)
- ❖ Xử lý:
 - Dùng API: registerUser để gửi dữ liệu đăng ký từ form đến server.
 - Kiểm tra dữ liệu: Xác thực tính hợp lệ của dữ liệu đầu
 - Xử lý API: Gửi yêu cầu POST tới /signup để thực hiện đăng ký.

4.3.1.2 API/Service: Đăng Ký

- ❖ Tên API/Service: registerUser
- Inputs: Full Name: full_name, Email: email, Username: username, Password: password
- Outputs:

Thành công: Thông tin người dùng mới (JSON), ví dụ: { "user_id": 1, "username": "user1", "email": "user1@example.com" }.

Lỗi: Thông báo lỗi nếu đăng ký không thành công, ví dụ: { "error": "Signup failed" }.

Xử lý:

Kiểm tra dữ liệu:

Xác thực tính hợp lệ của dữ liệu đầu vào:

- Đảm bảo email có định dạng hợp lệ.
- Đảm bảo mật khẩu đủ mạnh và không trùng lặp với mật khẩu đã đăng ký.
- Kiểm tra tên đăng nhập đã tồn tại hay chưa.

Truy xuất DB qua DBMS:

- Gọi User.create(req.body) để lưu thông tin người dùng vào cơ sở dữ liệu.
 Gọi các dịch vụ từ actors:
 - Tạo tài khoản mới cho người dùng, gửi thông tin về thành công hay lỗi đến client.

4.3.1.3 DBMS Procedure: Đăng Ký Người Dùng

- ❖ Tên Procedure: SP_User_Register
- ❖ Inputs: Full Name: @full_name, Email: @email, Username: @username, Password: @password
- Outputs:

Thông báo xác thực và lưu trữ thành công hoặc lỗi

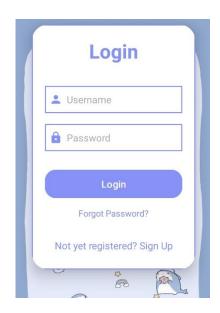
❖ Tables: users: Bảng lưu trữ thông tin người dùng mới, bao gồm các cột user_id, username, password, email, full_name, role, created_at, và updated_at.

```
1 • USE meeting_scheduler;
      DELIMITER //
 3 ● ○ CREATE PROCEDURE SP_User_Register(
         IN Username VARCHAR(100), IN Password VARCHAR(255), IN Email VARCHAR(255), IN FullName VARCHAR(255),
          IN Role VARCHAR(20))
 DECLARE user_exists INT;
          -- Kiểm tra nếu vai trò có giá trị hợp lệ, nếu không thì gán giá trị mặc định là 'member'
          IF Role IS NULL OR (Role <> 'manager' AND Role <> 'member') THEN
9
10
           SET Role = 'member';
11
          END IF;
12
          -- Kiểm tra nếu tên người dùng hoặc email đã tồn tại
13
          SELECT COUNT(*) INTO user exists
14
          FROM users
          WHERE username = Username OR email = Email;
15
16 ⊝
        IF user exists > 0 THEN
             SIGNAL SQLSTATE '45000' SET MESSAGE TEXT = 'Username or Email already exists';
17
         ELSE -- Thêm người dùng mới
18
19
           INSERT INTO users (username, password, email, full_name, role, created_at, updated_at)
20
              VALUES (Username, Password, Email, FullName, Role, NOW(), NOW());
21
              -- Trả về thông báo thành công
22
              SELECT 'User registered successfully' AS Message;
23
         END IF;
24
     END //
```

Hình 4.2 SP_User_Register

4.3.2 Đăng nhập

4.3.2.1 Form: Đăng nhập người dùng



Hình 4. 3 Đăng nhập người dùng

❖ Tên Form: loginForm

❖ Giao tiếp:

+ Actor: User (Người dùng)

+ Vai trò: Người dùng có vai trò đăng nhập vào hệ thống.

❖ Các Control:

- Control: Email

+ Inputs: Email nhập từ form

+ Outputs: Email được gửi đến API để xác thực người dùng.

- Control: Mật khẩu

+ Inputs: Mật khẩu nhập từ form

+ Outputs: Mật khẩu được gửi đến API để xác thực người dùng.

* Xử lý: Sử dụng API checkLogin để gửi dữ liệu đăng nhập (email và password) đến server để xác thực người dùng.

4.3.2.2 API/Service: Đăng nhập người dung

- ❖ Tên API/Service: checkLogin
- ❖ Inputs: email (Email người dùng), npassword (Mật khẩu người dùng)
- Outputs: Xác nhận đăng nhập thành công hoặc lỗi nếu không thể xác thực người dùng.
- Xử lý:
- Kiểm tra dữ liệu: Xác thực dữ liệu đầu vào từ form (email và mật khẩu).
- Truy xuất DB qua DBMS: Sử dụng Sequelize để gọi User.findone({ where: { email, password } }) để kiểm tra xem người dùng có tồn tại trong cơ sở dữ liệu với email và mật khẩu cung cấp không.
- Gọi các dịch vụ từ actors: Xác thực người dùng và phản hồi kết quả về client.

4.3.2.3 DBMS Procedure: Đăng nhập

- ❖ Tên Procedure: SP_User_Login
- ❖ Inputs: p_email (Email người dùng), p_password (Mật khẩu người dùng)
- Outputs: Xác nhận đăng nhập thành công hoặc thông báo lỗi nếu không thể xác thực người dùng.
- * Tables: Bång users được truy xuất.

```
1 • USE meeting_scheduler;
      DELIMITER //
 3 • ○ CREATE PROCEDURE SP_User_Login(
        IN Email VARCHAR(255), IN Password VARCHAR(255))
 6
         DECLARE user_count INT;
          -- Tìm người dùng với email và mật khẩu
 7
 8
         SELECT COUNT(*) INTO user_count
         WHERE email = Email AND password = Password;
11
        IF user_count = 0 THEN
12 🖨
             SIGNAL SQLSTATE '45000' SET MESSAGE TEXT = 'Invalid email or password';
13
14
15
              -- Trả về thông tin người dùng
             SELECT user_id, username, email, full_name, role
16
17
             FROM users
18
             WHERE email = Email AND password = Password;
20
       END //
21
      DELIMITER;
22
```

Hình 4.4 SP_User_Login

4.3.3 Trang chủ admin



Hình 4.5 Trang chủ admin

Admin đăng nhập vào bằng tài khoản của mình và có thể thực hiện được các chức năng bằng cách nhấp vào các form đang hiện.

4.3.4 Tạo cuộc họp

4.3.4.1 Form: Tạo cuộc họp



Hình 4. 6 Tạo cuộc họp



Hình 4.7 Tạo cuộc họp

- ❖ Giao tiếp: Cho actor Người Quản Lý, với vai trò là Người Tạo Cuộc Họp.
- ❖ Các control:
- Control: Title
 - + Inputs: Dữ liệu từ trường Title.
 - + Outputs: Giá trị Title được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Title.
- Control: Meeting Date
 - + Inputs: Dữ liệu từ trường Meeting Date.
 - + Outputs: Giá trị Meeting Date được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Meeting Date.
- Control: Start Time

- + Inputs: Dữ liệu từ trường Start Time.
- + Outputs: Giá trị Start Time được lưu trữ.
- + Xử lý: Kiểm tra tính hợp lệ của Start Time.
- Control: End Time
 - + Inputs: Dữ liệu từ trường End Time.
 - + Outputs: Giá trị End Time được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của End Time.
- Control: Location
 - + Inputs: Dữ liệu từ trường Location.
 - + Outputs: Giá trị Location được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Location.
- Control: Agenda
 - + Inputs: Dữ liệu từ trường Agenda.
 - + Outputs: Giá trị Agenda được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Agenda.
- Control: Documents
 - + Inputs: Dữ liệu từ trường Documents.
 - + Outputs: Giá trị Documents được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Documents.
- Control: Result
 - + Inputs: Dữ liệu từ trường Result.
 - + Outputs: Giá trị Result được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Result.
- Control: Next Meeting Time
 - + Inputs: Dữ liệu từ trường Next Meeting Time.
 - + Outputs: Giá trị Next Meeting Time được lưu trữ.
 - + Xử lý: Kiểm tra tính hợp lệ của Next Meeting Time.

4.3.4.2 API/Service: Tạo cuộc họp

- ❖ Tên API/Service: createMeeting
- Inputs: Title, Meeting Date, Start Time, End Time, Location, Agenda, Documents, Result, Next Meeting Time
- Outputs: Xác thực thành công hoặc lỗi
- ❖ Xử lý:

Kiểm tra dữ liệu: Xác thực thông tin cuộc họp.

Truy xuất DB qua DBMS: Gọi stored procedure để lưu thông tin cuộc họp mới vào cơ sở dữ liêu.

4.3.4.3 DBMS Procedure: Tạo cuộc họp

- ❖ Tên Procedure: SP_Create_Meeting
- Inputs: Title, Meeting Date, Start Time, End Time, Location, Agenda, Documents, Result, Next Meeting Time
- Outputs: Thông báo thành công hoặc lỗi.

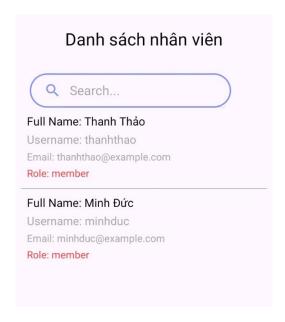
Tables: Bång meetings.

```
1 • USE meeting_scheduler;
                              DELIMITER //
    3 ● ○ CREATE PROCEDURE SP_Create_Meeting(
                            IN p_title VARCHAR(255), IN p_meeting_date DATE, IN p_start_time TIME, IN p_end_time TIME,
                             IN p_location VARCHAR(255), IN p_agenda TEXT, IN p_documents TEXT, IN p_result TEXT,
     6
                             IN p_next_meeting_time DATETIME
   8 ⊝ BEGIN
   9
                           -- Thêm cuộc họp mới vào bảng meetings
                                       INSERT INTO meetings (title, meeting date, start time, end time, location, agenda, documents, result,
10
                                                             next_meeting_time,created_at,updated_at)
11
12
13
                                               p\_title, p\_meeting\_date, p\_start\_time, p\_end\_time, p\_location, p\_agenda, p\_documents, p\_result, p\_next\_meeting\_time, p\_title, p\_meeting\_date, p\_start\_time, p\_end\_time, p\_location, p\_agenda, p\_documents, p\_result, p\_next\_meeting\_time, p\_title, p\_meeting\_time, p\_me
14
                                                            NOW(),
                                                                NOW()
15
16
                                               );
17
                         END //
18
19
                               DELIMITER;
20
```

Hình 4.8 SP_Create_Meeting

4.3.5 Xem danh sách nhân viên

4.3.5.1 Form: danh sách nhân viên



Hình 4.9 danh sách nhân viên

- ❖ Giao tiếp:
- Actor: Người Quản Lý hoặc Nhân Viên có quyền xem danh sách nhân viên
- Vai trò: Xem thông tin chi tiết về các nhân viên trong hệ thống
- ❖ Các Control:
- Control: Employee List Display
- Inputs: Yêu cầu từ giao diện để hiển thị danh sách nhân viên.
- Outputs: Danh sách nhân viên được hiển thị trên giao diện.
- Xử lý:

Gọi API getAllusers để lấy dữ liệu danh sách nhân viên từ server. Hiển thị danh sách nhân viên trên giao diện.

4.3.5.2 API/Service: Get All Users

- ❖ Tên API/Service: getAllUsers
- ❖ Inputs: Không có tham số đầu vào.
- Outputs: Danh sách nhân viên (bao gồm các trường thông tin như user_id, username, password, email, full_name, role, created_at, updated_at).
- ❖ Xử lý:
- Kiểm tra quyền truy cập của người dùng để đảm bảo họ có quyền xem danh sách nhân viên.
- Gửi yêu cầu đến endpoint API /api/user/users để lấy danh sách nhân viên.
- Nhận dữ liệu từ server và trả về cho giao diện để hiển thị.

4.3.5.3 DBMS Procedure: Lấy danh sách nhân viên

- Stored Procedure SP_Get_Employee_List thực hiện truy vấn dữ liệu từ bảng users để lấy danh sách nhân viên.
- ❖ Kết quả được trả về dưới dạng danh sách nhân viên với các thông tin chi tiết.
- * Table: bång User

```
USE meeting_scheduler;
       DELIMITER //
      CREATE PROCEDURE SP_Get_Employee_List()
           -- Chọn dữ liệu từ bảng users
 5
           SELECT
 6
7
               `user_id`,
               `username`,
               `password`,
9
10
               `email`,
11
               `full_name`,
12
               `role`,
               `created_at`,
13
14
               `updated at`
15
           FROM `users`;
16
       END //
17
18
       DELIMITER;
19
```

Hình 4.10 SP_Get_Employee_List

4.3.6 Xem danh sách cuộc họp và chi tiết cuộc họp

4.3.6.1 Form: danh sách cuộc họp và chi tiết cuộc họp



Hình 4. 11 danh sách cuộc họp



Hình 4.12 Chi tiết cuộc họp

Mannager và Admin có thể xem danh sách các cuộc họp. Muốn xem chi tiết cuộc họp chỉ cần nhấp vào cuộc họp đó sẽ chuyển sang trang chi tiết cuộc họp. Màu xanh là cuộc họp chưa diễn ra, màu đỏ là cuộc họp đã xảy ra.

- ❖ Giao tiếp:
- Actor: Người Quản Lý hoặc Nhân Viên có quyền xem danh sách cuộc họp.
- Vai trò: Xem thông tin chi tiết về các cuộc họp trong hệ thống.
- ❖ Các Control:
- Control: Meeting List Display
- Inputs: Yêu cầu từ giao diện để hiển thị danh sách các cuộc họp.
- Outputs: Danh sách các cuộc họp được hiển thị trên giao diện.
- ❖ Xử lý:
- Gọi API getMeetings để lấy dữ liệu danh sách các cuộc họp từ server.
- Hiển thị danh sách các cuộc họp trên giao diện.

4.3.6.2 API/Service: Get Meetings

- ❖ Tên API/Service: getMeetings
- ❖ Inputs: Không có tham số đầu vào.
- Outputs: Danh sách các cuộc họp (bao gồm các trường thông tin như meeting_id, title, meeting_date, start_time, end_time, location, agenda, documents, result, next_meeting_time, created_at, updated_at).
- ❖ Xử lý:
- Kiểm tra quyền truy cập của người dùng để đảm bảo họ có quyền xem danh sách các cuộc họp.
- Gửi yêu cầu đến endpoint API /api/meetings để lấy danh sách các cuộc họp.
- Nhận dữ liệu từ server và trả về cho giao diện để hiển thị.

4.3.6.3 DBMS Procedure: Lấy danh sách chi tiết cuộc họp và chi tiết từng cuộc họp

❖ Tên Procedure: SP Get Meeting List

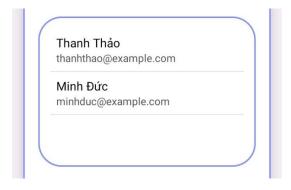
- ❖ Inputs: Không có tham số đầu vào.
- Outputs: Danh sách các cuộc họp với các thông tin chi tiết.
- * Tables: Bång meetings, user, meetingparticipants.

```
USE meeting_scheduler;
       DELIMITER //_
      CREATE PROCEDURE SP_Get_Meeting_List()
           -- Lấy danh sách các cuộc họp
           SELECT
               `meeting_id`,
              `title`,
 8
               `meeting_date`,
 9
               `start_time`,
10
               `end_time`,
11
               `location`,
12
               `agenda`,
13
               `documents`,
14
               `result`,
15
16
               `next_meeting_time`,
17
               `created_at`,
18
               `updated_at`
19
           FROM `meetings`;
20
     END //
21
       DELIMITER;
```

Hình 4.13 SP_Get_Meeting_List

4.3.7 Danh sách người tham gia trong một cuộc họp

4.3.7.1 Form: Danh sách người tham gia một cuộc họp



Hình 4.14 Danh sách người tham gia một cuộc họp

- ❖ Giao tiếp:
- Actor: Người Quản Lý hoặc Nhân Viên có quyền xem danh sách người tham gia cuộc họp.
- Vai trò: Xem thông tin chi tiết về các người tham gia cuộc họp trong hệ thống.
- ❖ Các Control:
- Control: Participant List Display
- Inputs: Yêu cầu từ giao diện để hiển thị danh sách người tham gia cuộc họp.
- Outputs: Danh sách người tham gia cuộc họp được hiển thị trên giao diện.
- Xử lý:
- Gọi API getMeetingParticipants để lấy dữ liệu danh sách người tham gia từ server.
- Hiển thị danh sách người tham gia trên giao diện.

4.3.7.2 API/Service: Get Meeting Participants

- ❖ Tên API/Service: getMeetingParticipants
- ❖ Inputs: Không có tham số đầu vào.
- ❖ Outputs: Danh sách người tham gia cuộc họp (bao gồm các trường thông tin như participant_id, meeting_id, participant_name, email, role, user_id, attendance status, created at, updated at).
- ❖ Xử lý:
- Kiểm tra quyền truy cập của người dùng để đảm bảo họ có quyền xem danh sách người tham gia cuộc họp.
- Gửi yêu cầu đến endpoint API /api/meetingparticipants để lấy danh sách người tham gia cuộc họp.
- Nhận dữ liệu từ server và trả về cho giao diện để hiển thị.

4.3.7.3 DBMS Procedure: Danh sách người tham gia

- ❖ Tên Procedure: SP_Get_Participants_By_Meeting
- ❖ Inputs: Không có tham số đầu vào.

- Outputs: Danh sách người tham gia cuộc họp với các thông tin chi tiết.
- * Tables: Bång meetingparticipants, meetings, uses.

```
USE meeting scheduler;
       DELIMITER //
2
3
      CREATE PROCEDURE SP_Get_Participants_By_Meeting(IN p_meeting_id INT)
4 •
           -- Lấy danh sách người tham gia của cuộc họp với meeting_id được chỉ định
6
7
           SELECT
8
               `participant_id`,
9
              `meeting_id`,
               `participant_name`,
10
11
               `email`,
               `role`,
12
              `user id`,
13
               `attendance_status`,
14
15
               `created_at`,
               `updated_at`
16
17
           FROM `meeting_scheduler`.`meetingparticipants`
           WHERE `meeting_id` = p_meeting_id;
18
19
      END //
20
       DELIMITER;
21
22
```

Hình 4.15 SP_Get_Participants_By_Meeting

4.3.8 Xem kết quả cuộc họp

4.3.8.1 Form: Kết quả cuộc họp với phê duyệt



Hình 4. 16 Kết quả cuộc họp với phê duyệt

- Giao tiếp:
- Actor: Người Quản Lý hoặc Nhân Viên có quyền xem và phê duyệt kết quả cuộc họp.
- Vai trò: Xem kết quả cuộc họp và chọn trạng thái phê duyệt (đồng ý hoặc không đồng ý).
- ❖ Các Control:
- Control: Meeting Results Approval
- Inputs: Yêu cầu từ giao diện để hiển thị kết quả cuộc họp và lựa chọn phê duyệt.
- Outputs: Kết quả cuộc họp và trạng thái phê duyệt được lưu trữ.
- ❖ Xử lý:
- Gọi API getMeetingResults để lấy dữ liệu kết quả cuộc họp từ server.
- Hiển thị kết quả cuộc họp và tùy chọn phê duyệt trên giao diện.
- Gửi trạng thái phê duyệt và (nếu cần) gọi ý xem xét lại đến server.

4.3.8.2 API/Service: Kết quả cuộc họp với Phê duyệt

- ❖ Tên API/Service: getMeetingResults
- ❖ Inputs: meeting_id (ID của cuộc họp cần lấy kết quả)
- Outputs: Danh sách kết quả cuộc họp (bao gồm các trường thông tin như result_id, meeting_id, result, new_requirements, created_at, updated_at).
- Xử lý:
- Kiểm tra quyền truy cập của người dùng để đảm bảo họ có quyền xem kết quả cuộc họp.
- Gửi yêu cầu đến endpoint API /api/meetingresults với tham số meeting_id để lấy kết quả cuộc họp.
- Nhận dữ liệu từ server và trả về cho giao diện để hiển thị.
- Endpoint API để Phê duyệt Kết quả:
- ❖ Tên API/Service: approveMeetingResults

Inputs: result_id (ID của kết quả cuộc họp cần phê duyệt), user_id (ID của người phê duyệt)

approval_status (Trạng thái phê duyệt, ví dụ: "approved" hoặc "request_review"), review_suggestion (Gợi ý xem xét lại nếu không đồng ý)

- Outputs: Xác nhận việc lưu trữ trạng thái phê duyệt.
- ❖ Xử lý:
- Kiểm tra quyền truy cập của người dùng.
- Gửi yêu cầu đến endpoint API /api/summaryapproval để lưu trạng thái phê duyệt và gọi ý (nếu có).
- Cập nhật trạng thái phê duyệt trong cơ sở dữ liệu.

4.3.8.3 DBMS Procedure: Phê duyệt Kết quả cuộc họp

- ❖ Tên Procedure: SP_Approve_Meeting_Result
- ❖ Inputs: p_result_id (ID của kết quả cuộc họp cần phê duyệt), p_user_id (ID của người phê duyệt), p_approval_status (Trạng thái phê duyệt, ví dụ: "approved" hoặc "request_review"), p_review_suggestion (Gợi ý xem xét lại nếu không đồng ý)
- ❖ Outputs: Xác nhận việc lưu trữ trạng thái phê duyệt.
- * Tables: summaryapproval

```
1 • USE meeting_scheduler;
     DELIMITER //
3 ● ○ CREATE PROCEDURE SP_Approve_Meeting_Result(
         IN p_result_id INT,IN p_user_id INT,IN p_approval_status ENUM('approved', 'request_review'),
5
          IN p_review_suggestion TEXT
6
          INSERT INTO `meeting scheduler`.`summaryapproval` (
9
             `summary_id`,`user_id`,`approval_status`,`review_suggestion`,`created_at`,`updated_at`
10
         VALUES (
11 Ó
12
             p_result_id,
13
             p_user_id,
14
             p_approval_status,
             p review suggestion,
             CURRENT TIMESTAMP,
              CURRENT TIMESTAMP
        );
     END //
19
      DELIMITER ;
```

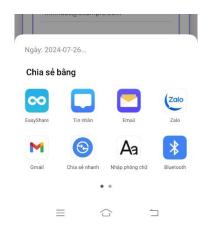
Hình 4. 17 SP_Approve_Meeting_Result

4.3.9 Chia sẽ thông tin cuộc họp

4.3.9.1 Form: Chia sẻ thông tin cuộc họp



Hình 4.18 Chia sẻ thông tin cuộc họp



Hình 4.19 Chia sẻ thông tin cuộc họp

- Giao tiếp:
- Actor: Người Quản Lý hoặc Nhân Viên có quyền chia sẻ thông tin cuộc họp.
- Vai trò: Chia sẻ thông tin cuộc họp với các cá nhân hoặc nhóm qua các phương thức như email, tin nhắn, hoặc nền tảng nội bộ.
- ❖ Các Control:
- Control: Meeting Sharing Interface

- Inputs: Yêu cầu từ giao diện để chia sẻ thông tin cuộc họp và lựa chọn phương thức chia sẻ.
- Outputs: Xác nhận việc chia sẻ thông tin cuộc họp thành công và thông báo cho người dùng.
- ❖ Xử lý:
- Sử dụng mã nguồn hiện tại để chia sẻ thông tin qua email.
- Hiển thị thông báo xác nhận việc chia sẻ thông tin thành công hoặc thất bại trên giao diện người dùng.

4.4 Thiết kế cơ sở dữ liệu và ràng buộc

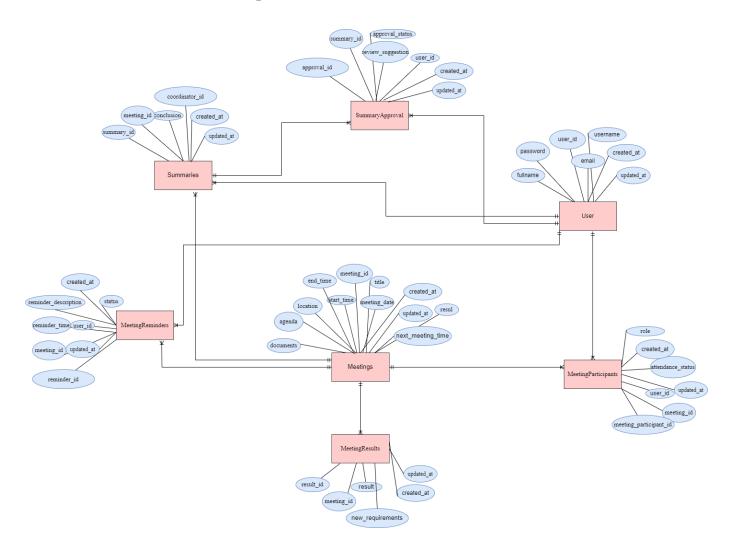
4.4.1 Thiết lập mô hình thực thể

4.4.1.1 Xác định các thực thể

- Users (<u>user_id (PK)</u>, username, password, email, full_name, role, created_at, updated_at)
- Meetings (<u>meeting id (PK)</u>, title, meeting_date, start_time, end_time, location, agenda, documents, result, next_meeting_time, created_at, updated_at)
- MeetingParticipants (<u>participant_id (PK)</u>, meeting_id (FK to Meetings), participant_name, email, role, user_id (FK to Users), attendance_status, created_at, updated_at)
- MeetingReminders (<u>reminder_id (PK)</u>, meeting_id (FK to Meetings),
 reminder_time, reminder_description, user_id (FK to Users), status, created_at,
 updated_at)
- MeetingResults (<u>result_id (PK)</u>, meeting_id (FK to Meetings), result,
 new_requirements, created_at, updated_at)
- Summaries (summary id (PK), meeting_id (FK to Meetings), conclusion,
 coordinator_id (FK to Users), created_at, updated_at

SummaryApproval (<u>approval id (PK)</u>, summary_id (FK to Summaries), user_id
 (FK to Users), approval_status, review_suggestion, created_at, updated_at)

4.4.1.2 Mô hình thực thể kết hợp ERD



Sơ đồ 4.9 ERD

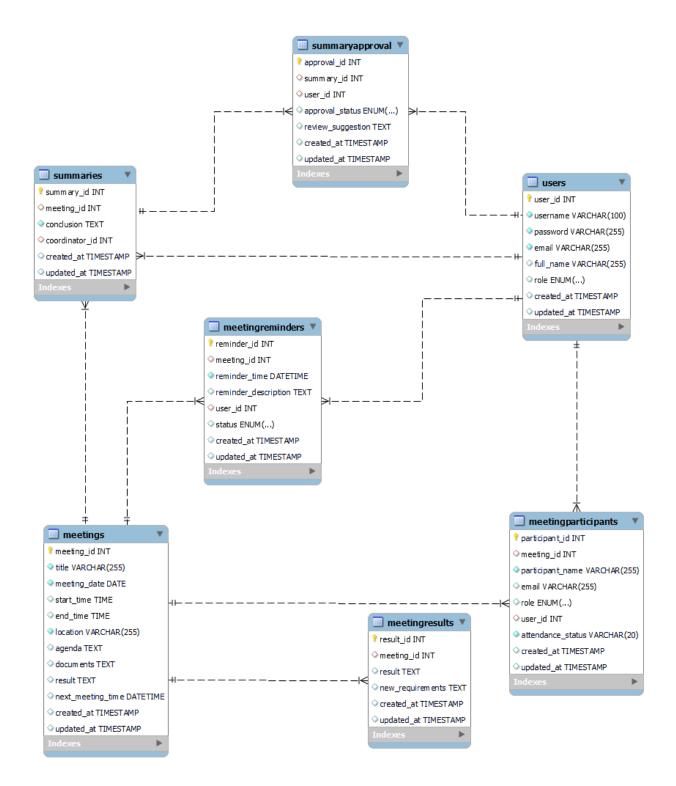
4.4.1.2 Mô hình thực thể dạng chuẩn 3

 Users (user_id (PK), username, password, email, full_name, role, created_at, updated_at)

- Meetings (meeting_id (PK), title, meeting_date, start_time, end_time, location, agenda, documents, result, next_meeting_time, created_at, updated_at)
- MeetingParticipants (participant_id (PK), meeting_id (FK to Meetings), participant_name, email, role, user_id (FK to Users), attendance_status, created_at, updated_at)
- MeetingReminders (reminder_id (PK), meeting_id (FK to Meetings),
 reminder_time, reminder_description, user_id (FK to Users), status, created_at,
 updated_at)
- MeetingResults (result_id (PK), meeting_id (FK to Meetings), result,
 new_requirements, created_at, updated_at)
- Summaries(summary_id (PK), meeting_id (FK to Meetings), conclusion,
 coordinator_id (FK to Users), created_at, updated_at
- SummaryApproval(approval_id (PK), summary_id (FK to Summaries), user_id
 (FK to Users), approval_status, review_suggestion, created_at, updated_at)
- * Ràng buộc (Constraints)
- Primary Key (PK): Đảm bảo tính duy nhất của mỗi bản ghi trong bảng.
- Foreign Key (FK): Đảm bảo rằng một giá trị trong bảng con phải tồn tại trong bảng cha, duy trì tính toàn vẹn của dữ liệu.

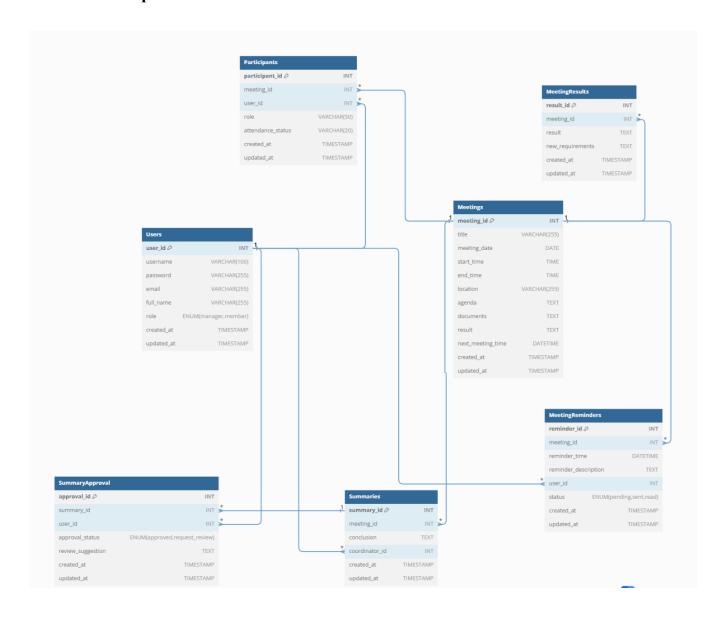
4.4.2 Thiết kế cơ sở dữ liệu

4.4.2.1 Mô hình diagram



Sơ đồ 4.10 Diagram

4.4.2.2 Sơ đồ lớp



Sơ đồ 4.11 Sơ đồ lớp

4.4.2.3 Chi tiết các bảng

Dưới đây là cấu trúc chi tiết của từng bảng trong cơ sở dữ liệu meeting_scheduler:

♣ Bảng Users: Lưu trữ thông tin người dùng hệ thống.

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	user_id	INT	Khóa chính, số tự động tăng	X	X		X
2	username	VARCHAR(10 0)	Tên đăng nhập				X
3	password	VARCHAR(25 5)	Mật khẩu người dùng				X
4	email	VARCHAR(25 5)	Địa chỉ email				X
5	full_name	VARCHAR(25 5)	Họ và tên đầy đủ				
6	role	ENUM('manag er', 'member')	Vai trò người dùng				
7	created_at	TIMESTAMP	Thời điểm tạo bản ghi				
8	updated_at	TIMESTAMP	Thời điểm cập nhật bản ghi				

Bång 4.1 Bång Users

♣ Bảng Meetings: Lưu trữ thông tin về các cuộc họp.

STT	Tên trường	Kiểu	Diễn giải	Khóa duy	Khóa chính	Khóa ngoại	Not Null
				nhất			
1	meeting_id	INT	Khóa chính, số	X	X		X
			tự động tăng				
2	title	VARCHAR(25	Tiêu đề cuộc họp				X
		5)					
3	meeting_d	DATE	Ngày diễn ra				X
	ate		cuộc họp				

4	start_time	TIME	Thời gian bắt đầu		
5	end_time	TIME	Thời gian kết thúc		
6	location	VARCHAR(25 5)	Địa điểm tổ chức cuộc họp		X
7	agenda	TEXT	Nội dung chính của cuộc họp		
8	documents	TEXT	Các tài liệu liên quan		
9	result	TEXT	Kết quả của cuộc họp		
10	next_meeti ng_time	DATETIME	Thời gian dự kiến cho cuộc họp tiếp theo		
11	created_at	TIMESTAMP	Thời điểm tạo bản ghi		
12	updated_at	TIMESTAMP	Thời điểm cập nhật bản ghi		

Bång 4.2 Bång Meetings

♣ Bảng MeetingParticipants: Bảng này lưu thông tin người tham gia cuộc họp.

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	meeting_participant_id	INT	Khóa chính, số tự động tăng	X	X		
2	meeting_id	INT	ID cuộc họp			X	X
3	user_id	INT	ID người dùng			X	X

4	role	VARCHAR(50)	Vai		X
			trò		
			của		
			người		
			tham		
			gia		
5	attendance_status	VARCHAR(20)	Trạng		X
			thái		
			tham		
			dự		
6	created_at	TIMESTAMP	Thời		
			gian		
			tạo		
			bản		
			ghi		
7	updated_at	TIMESTAMP	Thời		
			gian		
			cập		
			nhật		
			bản		
			ghi		

Bång 4.3 Bång MeetingParticipants

♣ Bảng MeetingResults: Lưu trữ kết quả và yêu cầu mới từ các cuộc họp

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	result_id	INT	Khóa chính, số tự động tăng	X	X		
2	meeting_id	INT	ID cuộc họp			X	
3	result	TEXT	Kết quả chi tiết của cuộc họp				
4	new_require ments	TEXT	Các yêu cầu mới sau cuộc họp				
5	created_at	TIMESTAMP	Thời gian tạo bản ghi				

6	updated_at	TIMESTAMP	Thời gian cập nhật		
			bản ghi		

Bång 4.4 Bång MeetingResults

♣ Bảng Summaries: Bảng này lưu các bản tóm tắt cuộc họp.

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not Null
1	summary_id	INT	Khóa chính, số tự động tăng	X	X		
2	meeting_id	INT	ID cuộc họp			X	X
3	conclusion	TEXT	Kết luận cuộc họp				X
4	coordinator_ id	INT	ID người điều phối cuộc họp			X	
5	created_at	TIMESTAMP	Thời gian tạo bản ghi				
6	updated_at	TIMESTAMP	Thời gian cập nhật bản ghi				

Bång 4.5 Bång Summaries

♣ Bảng SummaryApproval: Bảng này lưu thông tin phê duyệt các bản tóm tắt.

TT	Tên trường	Kiểu	Diễn	Khóa	Khóa	Khóa	Not
			giải	duy	chính	ngoại	Null
				nhất			
1	approval_id	INT	Khóa	X	X		
			chính,				
			số tự				
			động				
			tăng				
2	summary_id	INT	ID bản			X	X
			tóm tắt				
3	user_id	INT	ID			X	X
			người				
			dùng				

4	approval_status	ENUM('approved',	Trạng		
		'request_review')	thái phê		
			duyệt		
5	review_suggestion	TEXT	Gợi ý		
			xem xét		
			lại (nếu		
			có)		
6	created_at	TIMESTAMP	Thời		
			gian tạo		
			bản ghi		
7	updated_at	TIMESTAMP	Thời		
			gian		
			cập		
			nhật		
			bản ghi		

Bảng 4.6 Bảng SummaryApproval

♣ Bảng MeetingReminders: Quản lý nhắc nhở cho các cuộc họp

STT	Tên trường	Kiểu	Diễn	Khóa	Khóa	Khóa	Not
			giải	duy nhất	chính	ngoại	Null
1	reminder_id	INT	Khóa chính, số tự động tăng	X	X		
2	meeting_id	INT	ID cuộc họp			X	X
3	reminder_time	DATETIME	Thời gian nhắc nhở				X
4	reminder_description	TEXT	Mô tả nhắc nhở				
5	user_id	INT	ID người dùng nhận			X	

			nhắc		
			nhở		
6	status	ENUM('pending',	Trạng		
		'sent', 'read')	thái		
			nhắc		
			nhở		
7	created_at	TIMESTAMP	Thời		
			gian		
			tạo		
			bản		
			ghi		
8	updated_at	TIMESTAMP	Thời		
			gian		
			cập		
			nhật		
			bản		
			ghi		

Bång 4.7 Bång MeetingReminders

CHƯƠNG 5. KẾT LUẬN - HƯỚNG PHÁT TRIỀN

* Kết luận:

Ứng dụng tạo lịch họp trên thiết bị di động là một giải pháp quan trọng trong bối cảnh công việc ngày càng yêu cầu sự linh hoạt và hiệu quả. Đề tài này, tuy không phải là hoàn toàn mới, nhưng mang đến giá trị thiết thực trong việc quản lý lịch họp và phối hợp công việc cho các tổ chức và cá nhân. Ứng dụng đã được triển khai thành công với những chức năng chính như tạo và quản lý lịch họp, theo dõi kết quả, và gửi nhắc nhở tự động, từ đó giúp người dùng dễ dàng sắp xếp và quản lý thời gian hiệu quả hơn.

❖ Một số kết quả đạt được:

- Quản lý lịch họp: Úng dụng cho phép người dùng tạo cuộc họp, thêm thông tin chi tiết như thời gian, địa điểm, và tài liệu liên quan.
- Theo dõi kết quả và phê duyệt: Tính năng ghi lại kết quả cuộc họp và quản lý phê duyệt của các thành viên giúp đảm bảo tất cả các ý kiến và quyết định quan trọng được ghi nhận đầy đủ.
- Gửi nhắc nhở tự động: Nhắc nhở cuộc họp giúp giảm thiểu khả năng quên và đảm bảo tất cả các thành viên đều được thông báo kịp thời.

Hướng phát triển

- 1. Tối ưu hóa hiệu suất:
- + Tối ưu hóa tốc độ: Cải thiện tốc độ xử lý và phản hồi của ứng dụng để mang lại trải nghiệm mượt mà hơn, giảm thiểu thời gian chờ đợi của người dùng.
- + Quản lý bộ nhớ: Giảm mức tiêu tốn bộ nhớ và tài nguyên hệ thống để ứng dụng hoạt động hiệu quả trên các thiết bị có cấu hình thấp hơn.
- 2. Tính năng thông minh và tự động:
- + Gợi ý thông minh: Phát triển các thuật toán gợi ý thông minh giúp người dùng dễ dàng sắp xếp lịch họp dựa trên thói quen và lịch sử hoạt động.

- + Nhắc nhở thông minh: Tích hợp các nhắc nhở tự động dựa trên lịch trình của người dùng và các sự kiện quan trọng.
- 3. Mở rộng khả năng tương thích và tích hợp:
- + Tương thích với các nền tảng khác: Đảm bảo ứng dụng hoạt động ổn định trên nhiều hệ điều hành và các thiết bị có kích thước màn hình khác nhau.
- + Tích hợp công cụ quản lý dự án: Phát triển các tính năng tích hợp với các công cụ quản lý dự án phổ biến để nâng cao hiệu quả làm việc nhóm.
- 4. Tăng cường tính bảo mật:
- + Cải thiện bảo mật dữ liệu: Thực hiện các biện pháp bảo mật nâng cao để bảo vệ thông tin người dùng và ngăn chặn các cuộc tấn công mạng.
- + Xác thực và phân quyền: Cung cấp các phương thức xác thực mạnh mẽ và kiểm soát quyền truy cập chi tiết để đảm bảo chỉ những người dùng được phép mới có thể truy cập vào các tính năng và dữ liệu nhay cảm.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

- 1. Nguyễn Văn Cường, "Lập trình ứng dụng Android với Java", Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội, 2018.
- 2. Trần Thị Lan, "Java và ứng dụng trong phát triển phần mềm", Nhà xuất bản Giáo dục, Hà Nội, 2016.
- 3. Hoàng Minh Tuấn, "Node.js và xây dựng ứng dụng web", Nhà xuất bản Bưu điện, Hà Nội, 2020.
- 4. Nguyễn Trung Hiếu, "Hệ quản trị cơ sở dữ liệu MySQL: Cơ bản và nâng cao", Nhà xuất bản Đại học Quốc gia Hà Nội, 2019.

Tiếng Anh:

- 1. Biggs, J., & Collett, D. (2014), "Android Programming: The Big Nerd Ranch Guide", Big Nerd Ranch Guides, Atlanta.
- Schildt, H. (2018), "Java: The Complete Reference", McGraw-Hill Education, New York.
- 3. Tilkov, S. & Vinoski, S. (2010), "Node.js: Real-Time Web Application Development", O'Reilly Media, Sebastopol.
- 4. Korth, H.F., & Silberschatz, A. (2019), "Database System Concepts", McGraw-Hill Education, New York.
- 5. Maughan, D. (2021), "Full-Stack JavaScript: Learn Backbone.js, Node.js and MongoDB", O'Reilly Media, Sebastopol.

Danh mục các Website tham khảo:

- 1. Android Developer Documentation: https://developer.android.com/
- 2. Java Tutorials: https://docs.oracle.com/javase/tutorial/
- 3. Node.js Official Website: https://nodejs.org/

- 4. MySQL Documentation: https://dev.mysql.com/doc/
- 5. W3Schools: https://www.w3schools.com/