

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

— * —

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BỘ MÔN KỸ THUẬT MÁY TÍNH

**XÂY DỰNG ỨNG DỤNG
BÁCH KHOA OPEN DAY**

Sinh viên thực hiện: **Lê Mạnh Hùng**

Lớp CNTT 1.1 - K59

Giáo viên hướng dẫn: **ThS. Nguyễn Đức Tiến**

HÀ NỘI 05-2019

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Lê Mạnh Hùng

Điện thoại liên lạc: 034-823-4347

Email: hactieuho96@gmail.com

Lớp: Kỹ thuật máy tính K59

Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: C.ty TNHH Công nghệ và Truyền thông MDC

Thời gian làm ĐATN: Từ ngày 20/02/2019 đến 28/05/2019

2. Mục đích nội dung của ĐATN

- Quản lý các chương trình tham quan trường Đại học Bách khoa Hà Nội (Bách khoa Open Day).
- Xây dựng một ứng dụng trên nền tảng Android cho phép quản lý chương trình tham quan.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu việc lập trình ứng dụng trên Android Studio.
- Tìm hiểu về mô hình lập trình client – server
- Khảo sát quá trình mời học sinh tới tham dự OpenDay và các vấn đề phát sinh
- Tìm hiểu và tích hợp với các dịch vụ của Google như Map, Authentication.. SocketIO...
- Kiểm thử chương trình trên thiết bị Android có kết nối Internet.
- Tổng kết và đánh giá.

4. Lời cam đoan của sinh viên:

Tôi - *Lê Mạnh Hùng* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS Nguyễn Đức Tiến*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

	<i>Hà Nội, ngày 24 tháng 05 năm 2019</i> Tác giả ĐATN <i>Lê Mạnh Hùng</i>
--	---

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

	<i>Hà Nội, ngày 24 tháng 05 năm 2018</i> Giáo viên hướng dẫn <i>ThS Nguyễn Đức Tiến</i>
--	---

LỜI CẢM ƠN

Để có thể hoàn thành đồ án tốt nghiệp này, em xin gửi lời cảm ơn chân thành nhất tới tập thể các thầy giáo, cô giáo trường Đại học Bách khoa Hà Nội nói chung, viện Công nghệ thông tin và truyền thông, bộ môn Kỹ thuật máy tính, đã đào tạo cho em những kiến thức, những kinh nghiệm quý báu trong suốt thời gian học tập và rèn luyện.

Em xin gửi lời cảm ơn tới thầy giáo, ThS. Nguyễn Đức Tiến - Giảng viên bộ môn Kỹ thuật máy tính, viện Công nghệ thông tin và truyền thông, trường Đại học Bách khoa Hà Nội đã tận tình hướng dẫn em trong quá trình làm đồ án tốt nghiệp.

Tiếp theo, em xin gửi lời cảm ơn chân thành tới toàn thể các anh chị trong công ty TNHH Công nghệ và Truyền Thông MDC đã giúp đỡ rất nhiệt tình và tạo mọi điều kiện tốt nhất trong suốt quá trình em làm và hoàn thành đồ án tốt nghiệp tại công ty.

Cuối cùng là lời cảm ơn chân thành nhất tới gia đình, bạn bè đã luôn ở bên động viên, góp ý và tạo mọi điều kiện tốt nhất để tôi có thể hoàn thành đồ án tốt nghiệp này.

Tuy nhiên, do thời gian và kiến thức có hạn nên đồ án này chắc chắn không tránh khỏi những thiếu sót, em rất mong được sự đóng góp ý kiến của các thầy, các cô và toàn thể các bạn. Em xin chân thành cảm ơn.

Hà Nội ngày 23 tháng 5 năm 2019

Sinh viên: **Lê Mạnh Hùng**

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Mục tiêu chính của đồ án là xây dựng một ứng dụng quản lý các tour tham quan với tên gọi Bách khoa Open Day trên nền tảng Android. Ứng dụng này giúp cho người dùng chỉ với chiếc smartphone có kết nối internet trên tay là có thể tổ chức, đăng ký tham dự một chuyến tham quan trường ĐH Bách khoa. Đây cũng là một dự án quan trọng với nhà trường và các khoa viện.

Nội dung chính của đồ án gồm các phần:

Phần I: Đặt vấn đề và định hướng giải pháp

Phần này tập trung đề cập đến nền tảng và công nghệ phát triển ứng dụng cho đồ án, gồm các phần chính sau:

- + Tìm hiểu công nghệ lập trình ứng dụng Android.
- + Tìm hiểu về ngôn ngữ Java, xml, sql, Javascript – NodeJs.
- + Tìm hiểu về phương pháp kéo thả giao diện trên Android Studio.
- + Tìm hiểu cơ chế giao tiếp giữa client – server.

Phần II: Phân tích và thiết kế hệ thống

Phần này tập trung vào việc khảo sát, phân tích và thiết kế chương trình Bách khoa Open Day:

- + Phần đầu là giới thiệu, khảo sát hệ thống.
- + Phần giữa tập trung vào việc phân tích, đặc tả các chức năng hệ thống.
- + Phần cuối là việc thiết kế hệ thống bao gồm: kiến trúc hệ thống, thiết kế lớp, thiết kế cơ sở dữ liệu và thiết kế mã nguồn chương trình.

Phần III: Cài đặt và triển khai hệ thống

Phần đầu của sẽ tập trung mô tả các tính năng và giao diện của sản phẩm.

Phần tiếp theo sẽ mô tả giao diện của chương trình, kết quả kiểm thử và cách thức triển khai, phân phối ứng dụng đến tay người dùng.

Phần IV: Tổng kết và đánh giá

Phần này tập trung vào việc đánh giá kết quả đạt được, đưa ra phương hướng phát triển cho hệ thống sau khi kết thúc đồ án và phần kết luận.

LỜI NÓI ĐẦU

Làm sao để chọn đúng ngành phù hợp với bản thân? Câu hỏi tưởng dễ nhưng lại khá hóc búa đối với những bạn trẻ, nhất là những bạn vừa tốt nghiệp THPT vẫn đang còn loay hoay với vòng tròn Nghề-ngành-trường. Bởi nếu chọn sai, cái mất đi không chỉ là 4 năm đại học “sống hoài sống phí” mà tương lai vì thế cũng bị ảnh hưởng.

Từ nhu cầu trên, và mong muốn mang lại một sản phẩm có ý nghĩa và có tính ứng dụng cao, đã dẫn em đến ý tưởng xây dựng một chương trình giúp kết nối các bạn học sinh và trường ĐHBKHN giúp các bạn học sinh có cơ hội trải nghiệm môi trường đại học. SmartPhone, một thiết bị thiết yếu mà mỗi bạn học sinh ai cũng sở hữu ít nhất 1 cái, sẽ trở thành nền tảng thích hợp nhất để mang những sự kiện, hoạt động của nhà trường đến với các bạn học sinh.

Với những lý do trên, em quyết định chọn đề án tốt nghiệp của mình là “**Xây dựng ứng dụng Bách khoa Open Day**”, ứng dụng được lấy tên là “**BKOD**”. Mỗi người sử dụng, với chiếc SmartPhone trên tay, chỉ với yêu cầu có kết nối internet, là có thể xem các sự kiện, hoạt động mà nhà trường tổ chức. Qua đó, các bạn có thể đăng ký tham dự và quản lý các thành viên của mình một cách thuận tiện và trực quan nhất.

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	2
LỜI CẢM ƠN.....	3
TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP	4
LỜI NÓI ĐẦU	5
MỤC LỤC.....	6
DANH MỤC BẢNG.....	9
DANH MỤC HÌNH VẼ	10
DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ.....	11
PHẦN I: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP	12
1.1. Tổng quan	12
1.1.1. Các vấn đề, khó khăn hiện tại	12
1.1.2. Mục tiêu cần đạt được.....	12
1.1.3. Lựa chọn và định hướng thiết kế	12
1.2. Các phần cứng sử dụng trong hệ thống	12
1.2.1. Smartphone	12
1.2.2. Server	12
1.3. Các công cụ để tích hợp	13
1.3.1. Node.js	13
1.3.2. Cơ sở dữ liệu Sql Server 2017	13
1.4. Mô hình giao tiếp client-server.....	13
1.4.1. Socket.....	13
1.5. Ứng dụng các kỹ thuật vào bài toán đã đề ra	13
PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	14
2.1. Các chức năng	14
2.2. Usecase	14
2.2.1. Xác định các Actor	14
2.2.2. Xác định các Use Case	14
2.2.3. Vẽ bản vẽ Use Case	16
2.3. Thuật toán cốt lõi.....	18
2.3.1. Xác định màu sắc của các marker.....	18
2.3.2. Xác định màu sắc của các route.....	18
2.3.3. Vẽ route nối giữa đường với cửa vào của toà nhà	18
2.3.4. Định dạng thời gian	18
2.3.5. Thuật toán mã hoá mật khẩu trên client.....	19
2.4. Thiết kế cơ sở dữ liệu	20
2.5. Biểu đồ trình tự.....	25
2.5.1. Đăng ký tài khoản	25
2.5.2. Đăng nhập	25
2.5.3. Nhắn tin.....	26
2.6. Thiết kế giao diện/layout	27
2.6.1. Màu đặc trưng	27
2.6.2. Giao diện progress bar	27
2.6.3. Giao diện đăng nhập	28
2.6.4. Giao diện sau khi đăng nhập (trang chủ)	29

2.6.5.	Giao diện tab danh sách tour	30
2.6.6.	Giao diện tab danh sách tin tức.....	31
2.6.7.	Giao diện tab thông tin cá nhân	32
2.6.8.	Giao diện tab danh sách hội thoại.....	33
2.6.9.	Giao diện tab liên hệ	34
2.6.10.	Giao diện 1 tour.....	35
2.6.11.	Giao diện tab bản đồ tour	36
2.6.12.	Giao diện tab thông tin tour.....	40
2.6.13.	Giao diện tab hình ảnh bản đồ tour	41
2.6.14.	Giao diện thông tin của chặng.....	42
2.6.15.	Giao diện nhấn tin	43
2.7.	Mô hình tổng thể các thiết bị trong hệ thống	44
2.8.	Các giao thức	45
2.8.1.	Gửi thông tin cài đặt	45
2.8.2.	Đăng nhập	45
2.8.3.	Lấy danh sách tin tức	46
2.8.4.	Lấy danh sách tour.....	47
2.8.5.	Đăng xuất.....	48
2.8.6.	Toạ độ của thành viên.....	48
2.8.7.	Lấy danh sách hội thoại	49
2.8.8.	Lấy danh sách tư vấn viên	49
2.8.9.	Nhấn tin.....	50
2.8.10.	Kết nối lại	50
2.9.	Mô hình tích hợp phần cứng/phần mềm trong hệ thống	51
PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG.....		53
3.1.	Sản phẩm	53
3.1.1.	Database.....	53
3.1.2.	Client.....	53
3.1.3.	Server config.....	57
3.1.4.	Server	57
3.2.	Các tính năng	58
3.2.1.	Đăng nhập	58
3.2.2.	Xem danh sách tour	58
3.2.3.	Xem danh sách tin tức	58
3.2.4.	Xem thông tin cá nhân	58
3.2.5.	Xem danh sách tin nhắn.....	58
3.2.6.	Xem danh sách tư vấn viên.....	58
3.2.7.	Xem bản đồ tour	59
3.2.8.	Xem thông tin tour.....	59
3.2.9.	Xem hình ảnh minh hoạ tour	59
3.2.10.	Xem thông tin của chặng.....	59
3.2.11.	Nhấn tin với các thành viên trong tour.....	59
3.2.12.	Nhấn tin với các tư vấn viên.....	59
3.3.	Các giao diện cụ thể.....	60
3.3.1.	Giao diện progress bar	60
3.3.2.	Giao diện đăng nhập	60
3.3.3.	Giao diện tab danh sách tour	61

3.3.4.	Giao diện tab danh sách tin tức.....	61
3.3.5.	Giao diện tab thông tin người dùng	62
3.3.6.	Giao diện tab danh sách hội thoại.....	62
3.3.7.	Giao diện tab liên hệ	63
3.3.8.	Giao diện tab bản đồ tour.....	63
3.3.9.	Giao diện tab thông tin tour	64
3.3.10.	Giao diện tab hình ảnh minh họa tour	65
3.3.11.	Giao diện thông tin chặng	65
3.3.12.	Giao diện nhấn tin	66
3.4.	Hướng dẫn cài đặt cụ thể	66
3.4.1.	Cài đặt hệ quản trị cơ sở dữ liệu Microsoft Sql Server	66
3.4.2.	Cấu hình công cho Microsoft Sql Server.....	68
3.4.3.	Khởi tạo cơ sở dữ liệu.....	70
3.4.4.	Cài đặt Node.js.....	72
3.4.5.	Cài đặt ứng dụng phía server	74
3.4.6.	Cài đặt ứng dụng phía client	75
PHẦN IV: TỔNG KẾT VÀ ĐÁNH GIÁ		85
4.1.	Kết quả đạt được.....	85
4.2.	Phương hướng phát triển cho đồ án.....	85
4.3.	Kết luận.....	85
TÀI LIỆU THAM KHẢO		86
PHỤ LỤC.....		87
PHỤ LỤC 1: Hướng dẫn mở commandline trên các hệ điều hành.....		87
PHỤ LỤC 2: Định hướng thiết kế giao diện cho phiên bản tiếp theo		88

DANH MỤC BẢNG

Bảng 1. Bảng User – Danh sách tài khoản.....	21
Bảng 2. Bảng Tour – Danh sách tour.....	21
Bảng 3. Bảng Tour2Member – Liên kết tour và danh sách thành viên	22
Bảng 4. Bảng Message – Danh sách tin nhắn trong 1 cuộc hội thoại.....	22
Bảng 5. Bảng Timesheet – Danh sách các chặng tham quan của tour.....	22
Bảng 6. Bảng Tour2Timesheet – Liên kết tour và chặng	22
Bảng 7. Bảng Classroom – Danh sách các phòng học.....	23
Bảng 8. Bảng Manager – Danh sách các người quản lý phòng học theo từng khung giờ.....	23
Bảng 9. Bảng Timesheet2Classroom – Liên kết chặng, phòng học và người quản lý	23
Bảng 10. Bảng Building – Danh sách các toà nhà của trường.....	23
Bảng 11. Bảng Building2Classroom – Liên kết toà nhà và các phòng học trong đó	24
Bảng 12. Bảng News – Danh sách tin tức.....	24
Bảng 13. Bảng Setting – Cài đặt	24

DANH MỤC HÌNH VẼ

Hình 1. Bản vẽ biểu đồ phân rã chức năng	14
Hình 2. Bản vẽ Use Case cho Actor “User”	16
Hình 3. Bản vẽ Use Case cho Actor “Counselor”	17
Hình 4. Bản vẽ Use Case cho Actor “Admin”	17
Hình 5. Sơ đồ quan hệ thực thể	20
Hình 6. Bản vẽ biểu đồ trình tự đăng ký tài khoản	25
Hình 7. Bản vẽ biểu đồ trình tự đăng ký tạo tour	25
Hình 8. Bản vẽ biểu đồ trình tự thêm thành viên vào tour	26
Hình 9. Bản vẽ giao diện progress bar	27
Hình 10. Bản vẽ giao diện đăng nhập	28
Hình 11. Bản vẽ giao diện sau khi đăng nhập (trang chủ)	29
Hình 12. Bản vẽ giao diện tab danh sách tour	30
Hình 13. Bản vẽ giao diện 1 tour item	30
Hình 14. Bản vẽ giao diện tab danh sách tin tức	31
Hình 15. Bản vẽ giao diện 1 news item	31
Hình 16. Bản vẽ giao diện tab thông tin cá nhân	32
Hình 17. Bản vẽ giao diện tab danh sách hội thoại	33
Hình 18. Bản vẽ giao diện 1 conversation item	33
Hình 19. Bản vẽ giao diện tab liên hệ	34
Hình 20. Bản vẽ giao diện 1 contact item	34
Hình 21. Bản vẽ giao diện 1 tour	35
Hình 22. Bản vẽ giao diện bản đồ tour	36
Hình 23. Bản vẽ giao diện các marker	37
Hình 24. Bản vẽ giao diện các route	38
Hình 25. Bản vẽ minh hoạ các route khi biết 1 route màu xanh da trời	39
Hình 26. Bản vẽ minh hoạ các route khi biết 1 route màu đỏ	39
Hình 27. Bản vẽ minh hoạ các route dịch đi 1 khoảng để không bị trùng với route khác	40
Hình 28. Bản vẽ giao diện bản đồ tour	40
Hình 29. Bản vẽ giao diện bản đồ tour	41
Hình 30. Bản vẽ giao diện thông tin của chặng	42
Hình 31. Bản vẽ giao diện nhấn tin	43
Hình 32. Mô hình tổng thể các thiết bị trong hệ thống	44

DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Số thứ tự	Từ viết tắt	Ý nghĩa
1	Client	Máy khách. Là thiết bị sử dụng dịch vụ được cung cấp bởi server
2	Server	Máy chủ cung cấp dịch vụ
3	ĐHBK	Đại học Bách khoa
4	ĐHBKHN	Đại học Bách khoa Hà Nội
5	HUST	Hanoi University of Science and Technology
6	sdt	Số điện thoại
7	SDK	Software Development Kit – Bộ công cụ phát triển phần mềm
8	CSDL	Cơ sở dữ liệu

PHẦN I: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

1.1. Tổng quan

1.1.1. Các vấn đề, khó khăn hiện tại

- Khuôn viên trường ĐHBKHN rất rộng, dễ dàng gây nhầm lẫn cho những bạn học sinh lần đầu tới tham quan.
- Việc quản lý các bạn học sinh trong 1 tour gặp nhiều khó khăn. Các bạn học sinh tuổi còn trẻ nên rất năng động, thường tách đoàn đi riêng.
- Các bạn học sinh không nhớ được lộ trình, không biết đường đi và các địa điểm trong tour.
- Khi có thắc mắc, các bạn học sinh không biết hỏi ai.

1.1.2. Mục tiêu cần đạt được

- Ứng dụng giúp các bạn học sinh định vị chính xác được vị trí của mình, vị trí các chặng và đường đi giữa các chặng.
- Ứng dụng có tốc độ phản hồi cao, ít tốn pin và dữ liệu mạng.
- Các tương tác thời gian thực có độ trễ thấp (nhắn tin, cập nhật vị trí trên bản đồ).
- Định vị vị trí chính xác.
- An toàn dữ liệu, toàn vẹn thông tin.
- Thông báo khi có thành viên rời khỏi khu vực tham quan.

1.1.3. Lựa chọn và định hướng thiết kế

- Tốc độ phản hồi cao → Sử dụng Android Studio để lập trình ứng dụng phía client.
- Các tương tác thời gian thực có độ trễ thấp → Sử dụng giao tiếp socket giữa client và server.
- Định vị vị trí chính xác → Sử dụng api Google Maps của Google để định vị.
- An toàn dữ liệu, toàn vẹn thông tin → Sử dụng hệ quản trị cơ sở dữ liệu Sql Server.

1.2. Các phần cứng sử dụng trong hệ thống

1.2.1. Smartphone

- Smartphone chạy Android (Samsung, Huawei, Oppo,...).

1.2.2. Server

- Sử dụng máy chủ của viện CNTT-TT.

1.3. Các công cụ để tích hợp

1.3.1. Node.js

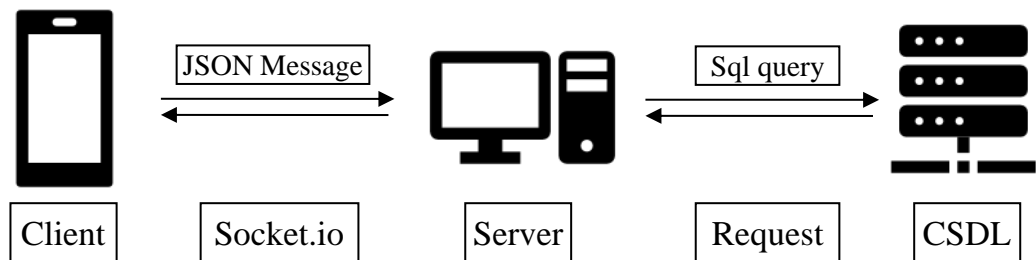
- Máy chủ cần cài đặt chương trình Node.js và một số các module cần thiết để chạy ứng dụng phía server (Hướng dẫn cài đặt xem mục 3.4.4 Cài đặt Node.js).

1.3.2. Cơ sở dữ liệu Sql Server 2017

- Nơi lưu trữ cơ sở dữ liệu cần cài đặt chương trình Microsoft Sql Server để ứng dụng phía server tương tác được với cơ sở dữ liệu (Hướng dẫn cài đặt xem mục Cài đặt hệ quản trị cơ sở dữ liệu Microsoft Sql Server).

1.4. Mô hình giao tiếp client-server

1.4.1. Socket



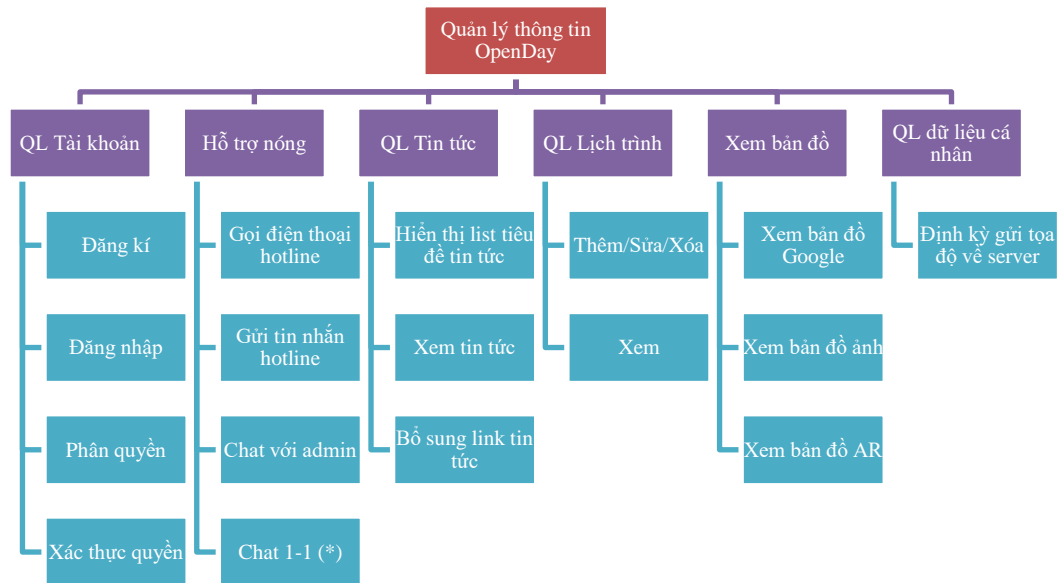
- Giao tiếp giữa client và server thông qua việc lắng nghe các socket.
- Ứng dụng phía client sẽ tích hợp thư viện `socket.io-client`, phiên bản mới nhất đến thời điểm hoàn thành đồ án này là 1.0.0 còn nhiều hạn chế (Mục Kết quả đạt được).
- Ứng dụng phía server sẽ tích hợp thư viện `socket.io`, phiên bản mới nhất đến thời điểm hoàn thành đồ án này là 2.2.0 và sẽ liên tục được cập nhật.
- Client và server sẽ trao đổi qua lại thông qua các chuỗi được chuẩn hoá gọi là các JSON message.
- Server giao tiếp với cơ sở dữ liệu thông qua các yêu cầu truy vấn gọi là các request. Dữ liệu server request lên cơ sở dữ liệu là các chuỗi chuẩn hoá gọi là các sql query.

1.5. Ứng dụng các kỹ thuật vào bài toán đã đề ra

- Android Studio: Dùng để lập trình client, giao diện người dùng cho khách hàng
- Api Google Maps: Dùng để quản lý lộ trình, hiển thị vị trí các thành viên trên bản đồ.
- Sql Server 2017: Dùng để lưu trữ dữ liệu danh sách thành viên, danh sách tour, tin nhắn, người sử dụng, các địa điểm,...

PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Các chức năng



Hình 1. Bản vẽ biểu đồ phân rã chức năng

2.2. Usecase

2.2.1. Xác định các Actor

- Actor được dùng để chỉ người sử dụng hoặc một đối tượng nào đó bên ngoài tương tác với hệ thống.
- Xem xét các mục tiêu của phần mềm, ta nhận thấy hệ thống có các Actor sau:
 - + **Học sinh cấp 2, 3, giáo viên, phụ huynh (User):** Những người sử dụng phần mềm để quản lý / được quản lý khi tham quan trường.
 - + **Tư vấn viên (Counselor):** Nhân viên giải đáp thắc mắc cho các User.
 - + **Quản trị viên (Admin):** Người quản trị hệ thống, phân quyền.

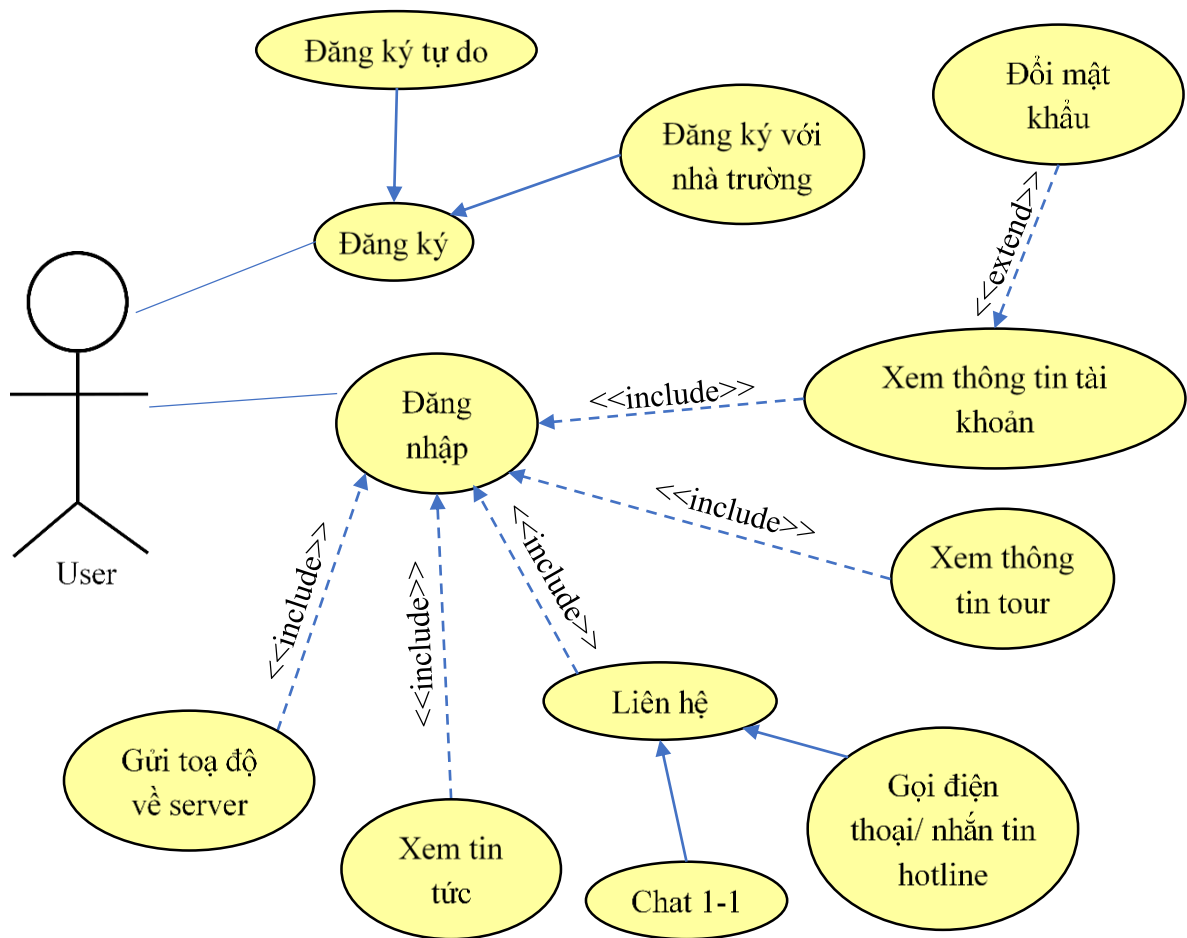
2.2.2. Xác định các Use Case

- Use Case là ca sử dụng của hệ thống, giúp chúng ta biết được ai sử dụng hệ thống, hệ thống có những chức năng gì. Xác định được use case, bạn sẽ hiểu được yêu cầu của hệ thống cần xây dựng.

- Để xác định được các Use Case, bạn cần xem xét các Actor sử dụng chức năng nào của hệ thống:
 - + [User](#):
 - Quản lý tài khoản:
 - Đăng ký
 - Đăng nhập
 - Xem thông tin tài khoản
 - Xem thông tin tour:
 - Lịch trình:
 - Lịch trình text (mốc thời gian, địa điểm)
 - Lịch trình Google Maps
 - Điểm đến tiếp theo trên màn hình điện thoại (sử dụng công nghệ AR)
 - Danh sách thành viên, trưởng đoàn:
 - Xem thông tin thành viên
 - Xem toạ độ các thành viên trong tour
 - Liên hệ
 - Gọi điện thoại/gửi tin nhắn vào số hotline
 - Chat (1-1):
 - Với các thành viên trong tour
 - Với [Counselor](#)
 - Xem tin tức
 - Tự động gửi toạ độ bản đồ lên server
 - + [Counselor](#)
 - [Xem thông tin tour](#)
 - [Chat \(1-1\)](#) (giải đáp thắc mắc):
 - Với [User](#)
 - [Nghe điện thoại, nhận tin nhắn hotline](#) (giải đáp thắc mắc)
 - + [Admin](#)
 - Quản lý các tài khoản [User](#), [Counselor](#)
 - [Xem thông tin tour](#)
 - Phân quyền cho tài khoản [User](#), [Counselor](#)

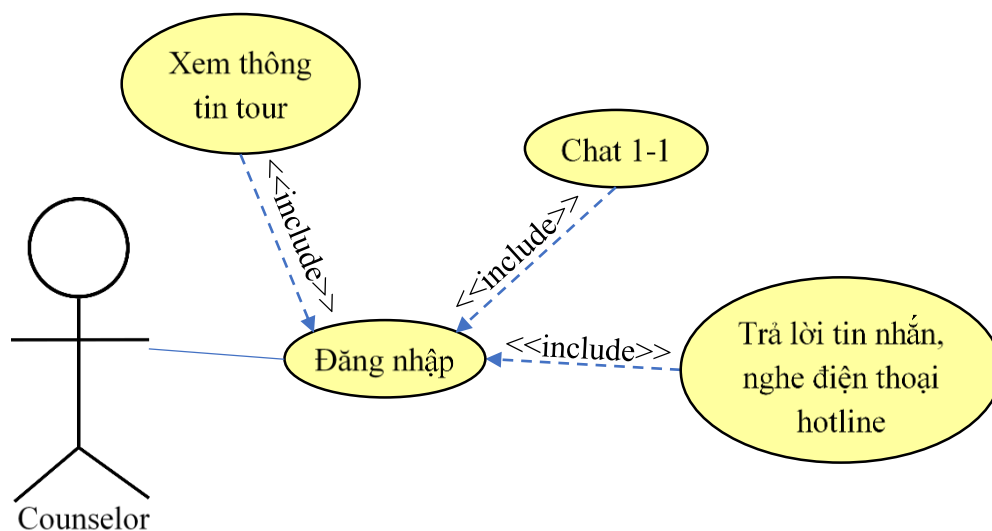
2.2.3. Vẽ bản vẽ Use Case

- User



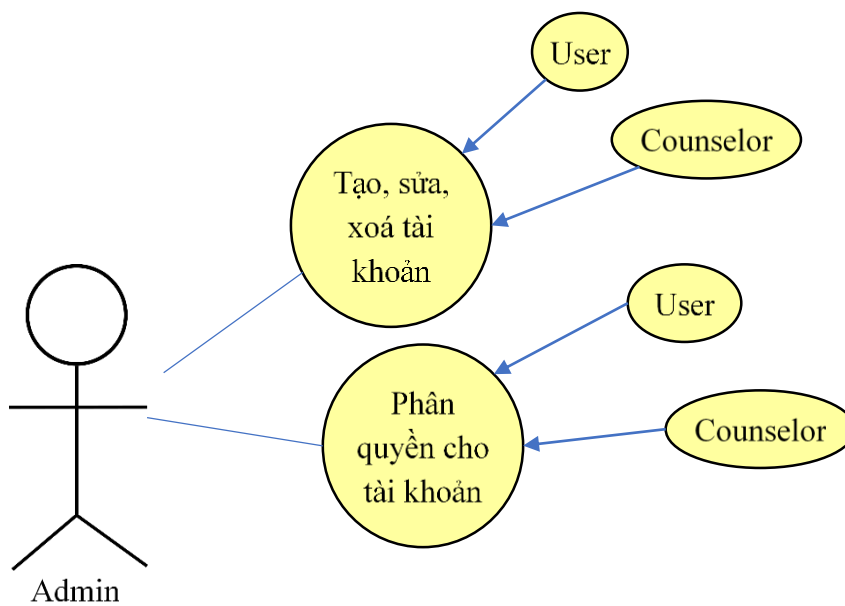
Hình 2. Bản vẽ Use Case cho Actor “User”

- Counselor



Hình 3. Bản vẽ Use Case cho Actor “Counselor”

- Admin



Hình 4. Bản vẽ Use Case cho Actor “Admin”

2.3. Thuật toán cốt lõi

2.3.1. Xác định màu sắc của các marker

- Xem mục *Giao diện tab bản đồ tour*.

2.3.2. Xác định màu sắc của các route

- Xem mục *Giao diện tab bản đồ tour*.

2.3.3. Vẽ route nối giữa đường với cửa vào của toà nhà

- Google map api chỉ hỗ trợ tìm đường đi trên trục đường cái, không hỗ trợ tìm đường đi trên những lối đi nhỏ trong khuôn viên trường.
- Các toà nhà của Bách khoa khá to, tìm được nhà chưa chắc đã tìm được lối vào. → Vẽ thêm các nét đứt để chỉ dẫn lối vào.
- Xem mục *Giao diện tab bản đồ tour*.

2.3.4. Định dạng thời gian

- Định dạng ngày mặc định khi truy vấn từ cơ sở dữ liệu là:
`2019-04-26T06:30:45.000Z` (Đầy đủ cả ngày, tháng, năm, giờ, phút, giây, tích tắc).
- Khi truy vấn vào bảng tour thì chỉ cần gửi bộ dữ liệu bao gồm ngày, tháng, năm về client thôi. Còn khi truy vấn vào bảng timesheet thì chỉ cần bộ dữ liệu bao gồm giờ, phút, giây.
- Định dạng lại ngày, tháng, năm theo dạng: `DD-MM-YYYY` bằng thuật toán:
 - + Dùng hàm lấy ngày từ chuỗi định dạng mặc định (ví dụ ta được chuỗi ngày là `26`).
 - + Chèn thêm ký tự `0` trước ký tự của chuỗi ngày, ta được chuỗi ngày là `026`.
 - + Lấy ra 2 ký tự cuối cùng của chuỗi ngày, ta được chuỗi ngày cuối cùng là `26` (tương tự với trường hợp chuỗi ngày có 1 ký tự là `6` thì kết quả sau 3 bước trên sẽ là `06`).
 - + Tương tự với chuỗi tháng (ví dụ ta có chuỗi tháng là `4` thì kết quả sau 3 bước trên sẽ là `04`). Lưu ý là hàm lấy chuỗi tháng trả về kết quả giá trị của tháng bị giảm đi 1 đơn vị nên phải tăng 1 đơn vị để bù vào.
 - + Chuỗi năm giữ nguyên
 - + Ghép chuỗi ngày, chuỗi tháng và chuỗi năm lại và chèn ký tự `-` ở giữa, ta được định dạng ngày gửi về client như sau: `26-04-2019`.

- + Đoạn code minh hoạ trên server (Node.js):

```
${String("0" + time.getUTCDate()).slice(-2)}-  
${String("0" + (time.getUTCMonth()+1)).slice(-2)}-  
${ time.getUTCFullYear() }
```

- Định dạng lại giờ, phút, giây theo dạng: hh:mm:ss bằng thuật toán:
 - + Dùng hàm lấy giờ từ chuỗi định dạng mặc định (ví dụ ta được chuỗi giờ là 6).
 - + Chèn thêm ký tự 0 trước ký tự của chuỗi giờ, ta được chuỗi giờ là 06.
 - + Lấy ra 2 ký tự cuối cùng của chuỗi giờ, ta được chuỗi giờ cuối cùng là 06 (trùng tự với trường hợp chuỗi giờ có 2 ký tự là 16 thì kết quả sau 3 bước trên sẽ là 16).
 - + Tương tự với chuỗi phút và chuỗi giây.
 - + Ghép chuỗi giờ, chuỗi phút và chuỗi giây lại và chèn ký tự : ở giữa, ta được định dạng giờ gửi về client như sau: 06:30:45.
 - + Đoạn code minh hoạ trên server (Node.js):

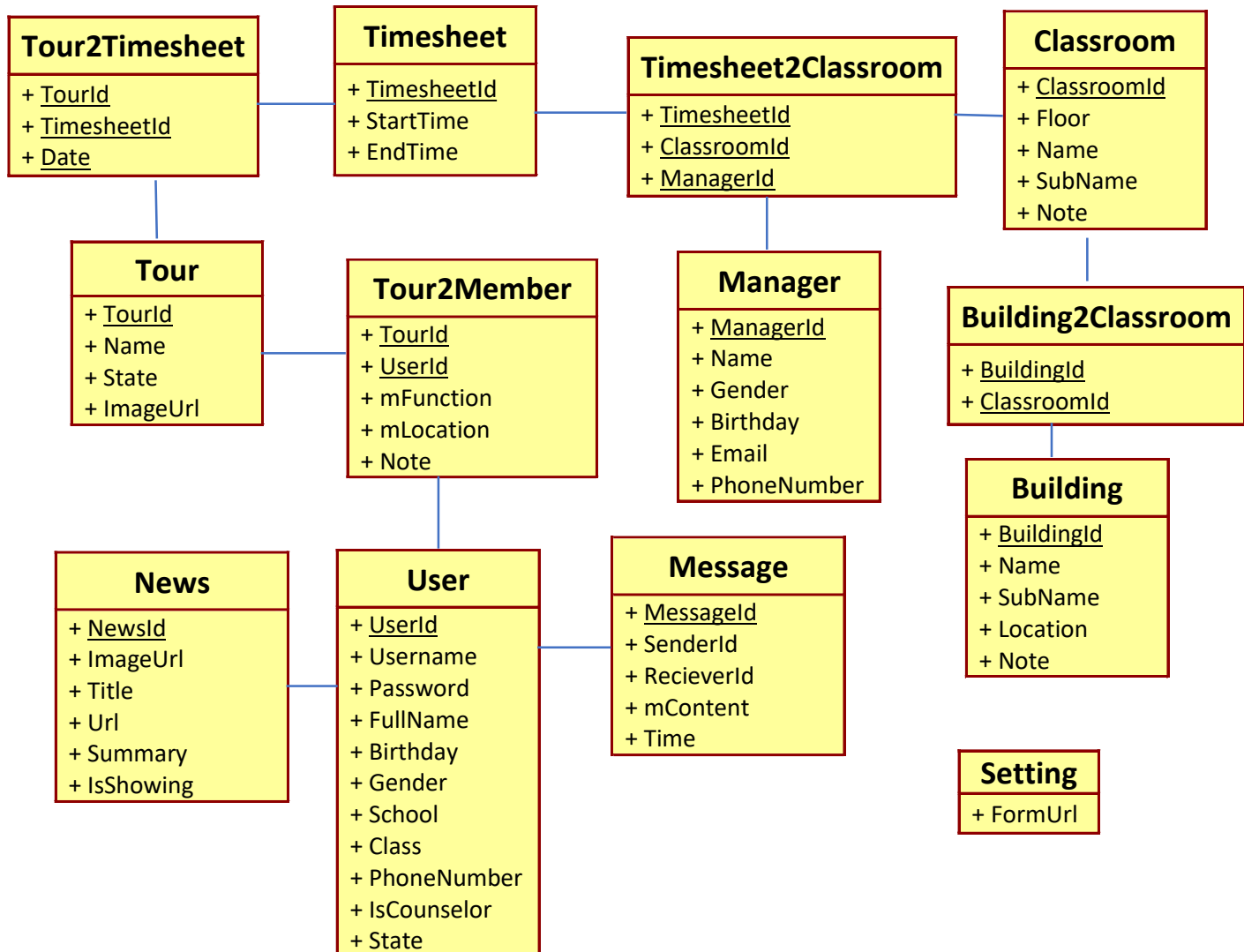
```
${String("0" + time.getUTCHours()).slice(-2)}:  
${String("0" + (time.getUTCMinutes()).slice(-2))}:  
${String("0" + (time.getUTCSeconds()).slice(-2))}
```

2.3.5. Thuật toán mã hoá mật khẩu trên client

- Sử dụng giải thuật băm SHA256.
- Băm kết hợp với tên đăng nhập và 1 private key là BKODv1Habvietio.
- Đoạn code minh hoạ trên client (Android Studio):

```
sha256(password + username + "BKODv1Habvietio");
```

2.4. Thiết kế cơ sở dữ liệu



Hình 5. Sơ đồ quan hệ thực thể

- *Đặc tả các bảng dữ liệu có trong cơ sở dữ liệu:*

Tên trường	Kiểu dữ liệu	Ghi chú
<u>UserId</u>	INT	ID người dùng, tăng dần và bắt buộc.
Username	VARCHAR(40)	Tên đăng nhập, duy nhất và bắt buộc.
Password	VARCHAR(40)	Mật khẩu đăng nhập, bắt buộc. Mặc định là 1234.
Fullname	NVARCHAR(50)	Họ tên đầy đủ, bắt buộc. Mặc định là Lê Mạnh Hùng.
Birthday	DATE	Ngày sinh, bắt buộc. Mặc định là 14/06/2002.
Gender	TINYINT	Giới tính (0 là không tiết lộ, 1 là nam, 2 là nữ, 3 giới tính thứ 3), bắt buộc. Mặc định là 1.
School	NVARCHAR(100)	Tên trường người dùng đang theo học.
Class	NVARCHAR(10)	Tên lớp người dùng đang theo học.
PhoneNumber	VARCHAR(15)	Sđt của người người dùng.
IsCounselor	BIT	Đánh dấu có phải là tư vấn viên không (0 là không phải, 1 là tư vấn viên, mặc định là 0).
State	TINYINT	Trạng thái (0 là offline, 1 là online, 2 là busy, 3 là hidden), bắt buộc. Mặc định là 0.

Bảng 1. Bảng User – Danh sách tài khoản

Tên trường	Kiểu dữ liệu	Ghi chú
<u>TourId</u>	INT	ID của tour, tăng dần và bắt buộc.
Name	NVARCHAR(100)	Tên của tour, bắt buộc.
State	TINYINT	Trạng thái hiển thị của tour (0 là ẩn, 1 hiển thị). Mặc định là 0.
ImageUrl	VARCHAR(300)	Đường dẫn ảnh minh họa tour.
Date	DATE	Ngày hoạt động của chặng, bắt buộc. Mặc định là 01/06/2019.
MapImageUrl	VARCHAR(300)	Đường dẫn ảnh bản đồ tour.

Bảng 2. Bảng Tour – Danh sách tour

Tên trường	Kiểu dữ liệu	Ghi chú
<u>TourId</u>	INT	ID của tour, khoá ngoại và bắt buộc.
<u>UserId</u>	INT	ID người dùng, khoá ngoại và bắt buộc.
mFunction	TINYINT	Vai trò trong tour (1 là trưởng nhóm, 2 là phó nhóm, 3 là phụ huynh, 0 là thành viên bình thường), bắt buộc. Mặc định là 0.
mLocation	GEOMETRY	Toạ độ trên Google Maps.
Note	NVARCHAR(100)	Ghi chú.

Bảng 3. Bảng Tour2Member – Liên kết tour và danh sách thành viên

Tên trường	Kiểu dữ liệu	Ghi chú
<u>MessageId</u>	INT	ID tin nhắn, tăng dần và bắt buộc.
<u>SenderId</u>	INT	Id người nhắn, bắt buộc.
<u>RecieverId</u>	INT	Id người nhận, bắt buộc.
mContent	NVARCHAR(500)	Nội dung tin nhắn, bắt buộc. Mặc định là “Hello”
Time	DATETIME	Thời điểm nhắn tin, bắt buộc. Mặc định là 12:00:00 01/06/2019.

Bảng 4. Bảng Message – Danh sách tin nhắn trong 1 cuộc hội thoại

Tên trường	Kiểu dữ liệu	Ghi chú
<u>TimesheetId</u>	INT	ID của chặng, tăng dần và bắt buộc.
StartTime	TIME	Giờ bắt đầu chặng, bắt buộc.
EndTime	TIME	Giờ kết thúc chặng, bắt buộc.

Bảng 5. Bảng Timesheet – Danh sách các chặng tham quan của tour

Tên trường	Kiểu dữ liệu	Ghi chú
<u>TourId</u>	INT	ID của tour, khoá ngoại và bắt buộc.
<u>TimesheetId</u>	INT	ID của chặng, khoá ngoại và bắt buộc.

Bảng 6. Bảng Tour2Timesheet – Liên kết tour và chặng

Tên trường	Kiểu dữ liệu	Ghi chú
<u>ClassroomId</u>	INT	ID phòng học, tăng dần và bắt buộc.
Floor	TINYINT	Tầng của phòng học, bắt buộc. Mặc định là 1
Name	VARCHAR(40)	Tên phòng học, bắt buộc. Mặc định là “101”
SubName	VARCHAR(40)	Tên khác của phòng học.
Note	VARCHAR(100)	Ghi chú.

Bảng 7. Bảng Classroom – Danh sách các phòng học

Tên trường	Kiểu dữ liệu	Ghi chú
<u>ManagerId</u>	INT	ID người quản lý phòng học trong khung giờ, tăng dần và bắt buộc.
Name	NVARCHAR(40)	Tên người quản lý, bắt buộc. Mặc định là “Trần Minh Thành”
Gender	TINYINT	Giới tính(0 là không tiết lộ, 1 là nam, 2 là nữ, 3 giới tính thứ 3), bắt buộc. Mặc định là 1.
Birthday	DATE	Ngày sinh, bắt buộc. Mặc định là 21/04/1990.
Email	VARCHAR(100)	Email của người quản lý, bắt buộc.
PhoneNumber	VARCHAR(15)	Sđt của người quản lý, bắt buộc.

Bảng 8. Bảng Manager – Danh sách các người quản lý phòng học theo từng khung giờ

Tên trường	Kiểu dữ liệu	Ghi chú
<u>TimesheetId</u>	INT	ID của chặng, khoá ngoại và bắt buộc.
<u>ClassroomId</u>	INT	ID phòng học, khoá ngoại và bắt buộc.
<u>ManagerId</u>	INT	ID người quản lý phòng học trong khung giờ, khoá ngoại và bắt buộc.

Bảng 9. Bảng Timesheet2Classroom – Liên kết chặng, phòng học và người quản lý

Tên trường	Kiểu dữ liệu	Ghi chú
<u>BuildingId</u>	INT	ID toà nhà, tăng dần và bắt buộc.
Name	NVARCHAR(40)	Tên toà nhà, duy nhất và bắt buộc. Mặc định là “D3”
SubName	NVARCHAR(40)	Tên khác của toà nhà.
Location	GEOMETRY	Toạ độ toà nhà trên Google Maps, bắt buộc. Mặc định là toạ độ nhà D3: 'POINT(21.004797 105.845266)'
Note	NVARCHAR(100)	Ghi chú

Bảng 10. Bảng Building – Danh sách các toà nhà của trường

Tên trường	Kiểu dữ liệu	Ghi chú
<u>BuildingId</u>	INT	ID toà nhà, khoá ngoại và bắt buộc.
<u>ClassroomId</u>	INT	ID phòng học, khoá ngoại và bắt buộc.

Bảng 11. Bảng Building2Classroom – Liên kết toà nhà và các phòng học trong đó

Tên trường	Kiểu dữ liệu	Ghi chú
<u>NewsId</u>	INT	ID tin tức, tăng dần và bắt buộc.
ImageUrl	VARCHAR(300)	Đường dẫn ảnh minh hoạ tin tức.
Title	NVARCHAR(100)	Tiêu đề tin tức, bắt buộc.
Url	NVARCHAR(300)	Đường dẫn đến trang tin tức, bắt buộc.
Summary	NVARCHAR(100)	Tóm tắt nội dung tin tức, bắt buộc.
IsShowing	BIT	Đánh dấu có hiển thị không (0 là không hiển thị, 1 là hiển thị), bắt buộc.

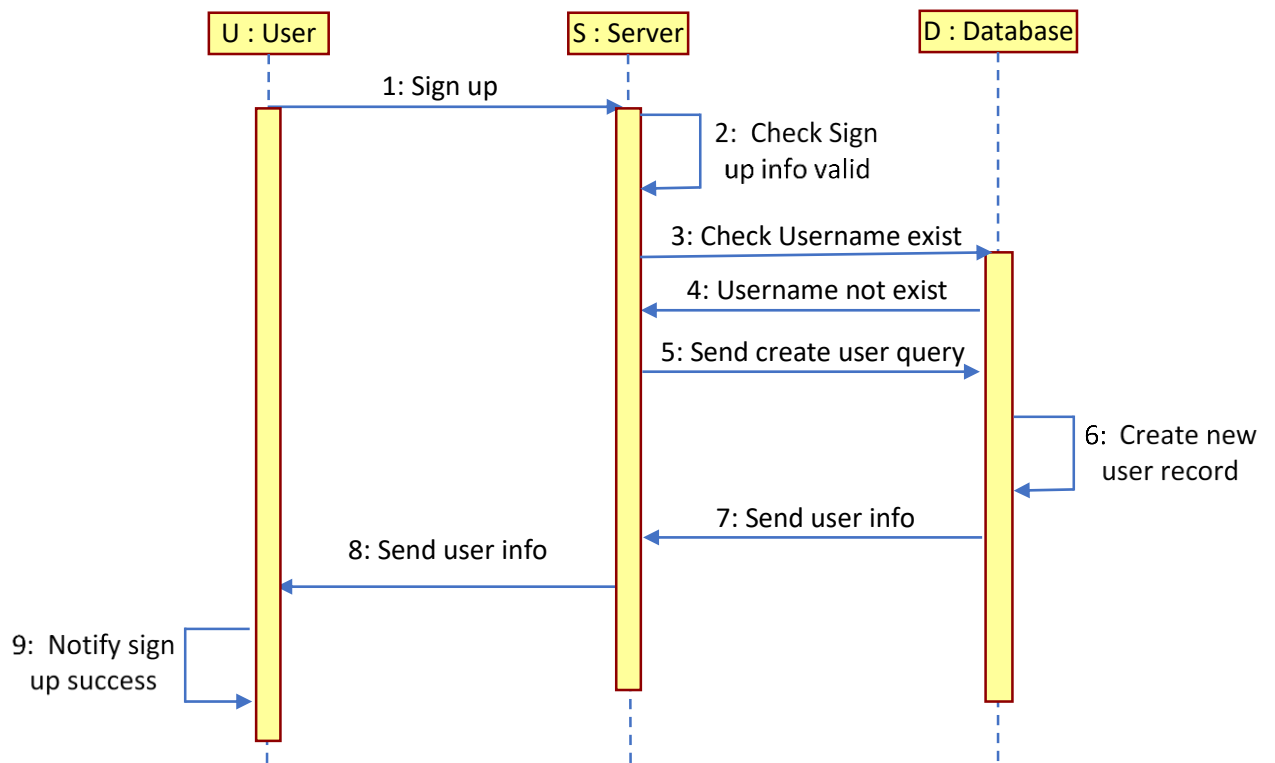
Bảng 12. Bảng News – Danh sách tin tức

Tên trường	Kiểu dữ liệu	Ghi chú
FormUrl	VARCHAR(300)	Đường dẫn form đăng ký thông tin người dùng để tạo tài khoản.

Bảng 13. Bảng Setting – Cài đặt

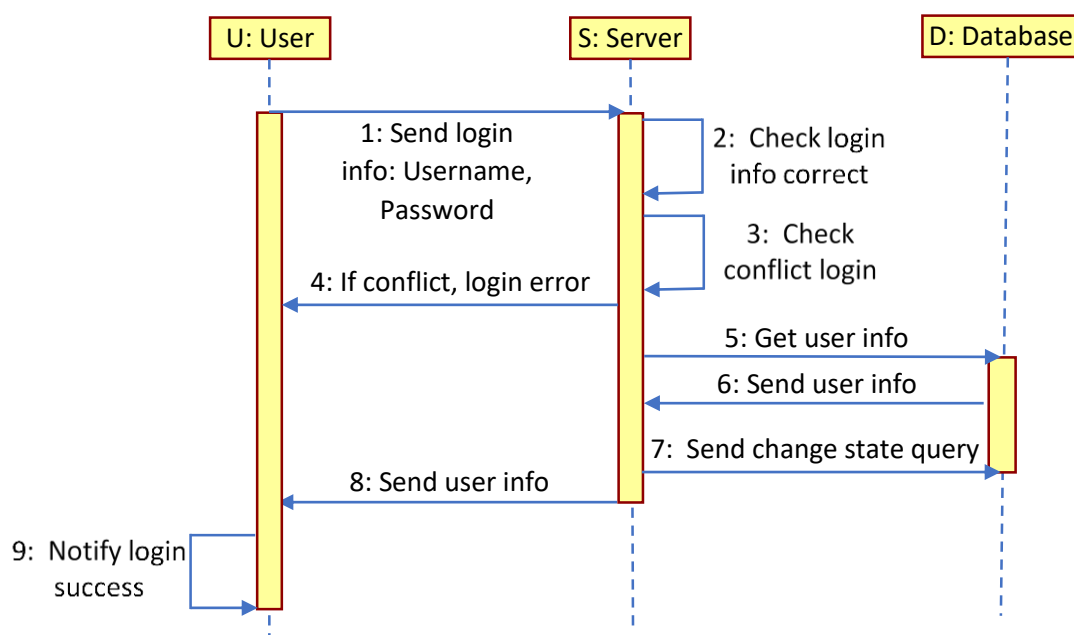
2.5. Biểu đồ trình tự

2.5.1. Đăng ký tài khoản



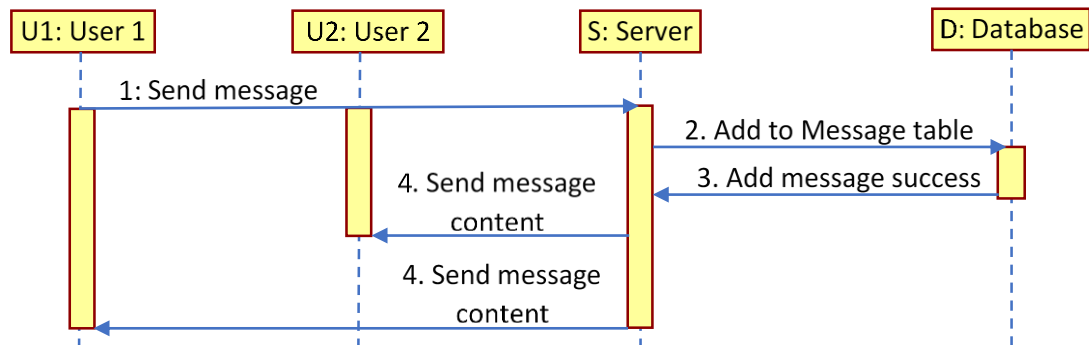
Hình 6. Bản vẽ biểu đồ trình tự đăng ký tài khoản

2.5.2. Đăng nhập



Hình 7. Bản vẽ biểu đồ trình tự đăng nhập

2.5.3. Nhắn tin



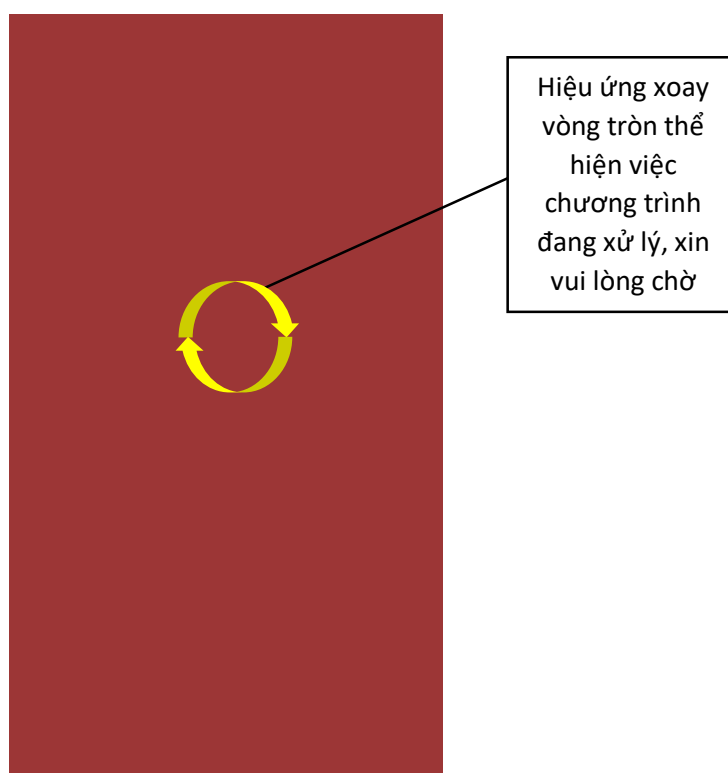
Hình 8. Bản vẽ biểu đồ trình tự nhắn tin

2.6. Thiết kế giao diện/layout

2.6.1. Màu đặc trưng

- Logo Bách khoa có 3 màu trắng, đỏ, vàng. Phần mềm cũng sử dụng 3 màu đặc trưng. Các thiết kế sẽ dựa trên trang <https://www.hust.edu.vn/brand-identity>

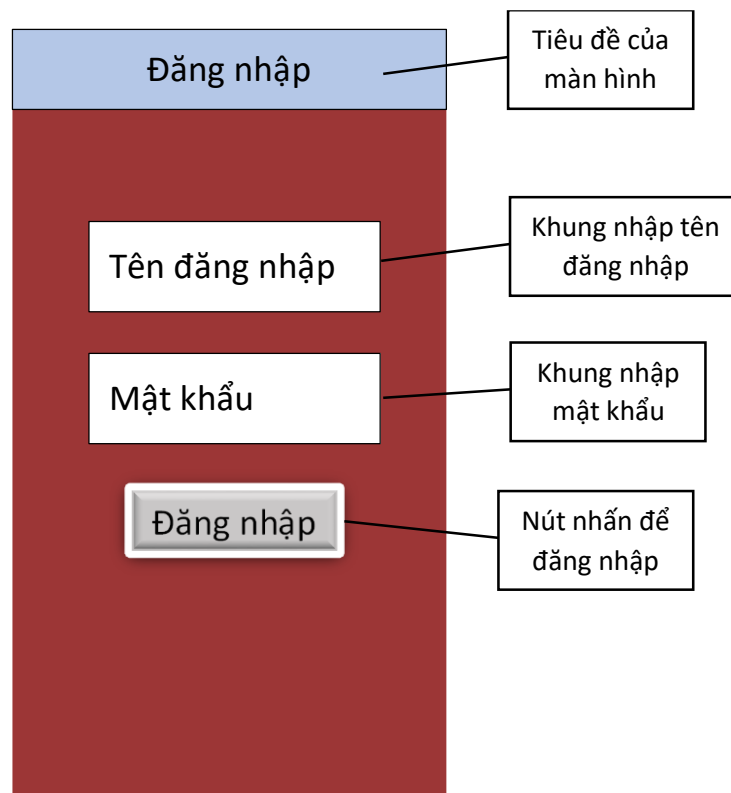
2.6.2. Giao diện progress bar



Hình 9. Bản vẽ giao diện progress bar

- Đây là giao diện hiện lên khi ứng dụng đợi phản hồi mất một thời gian dài.

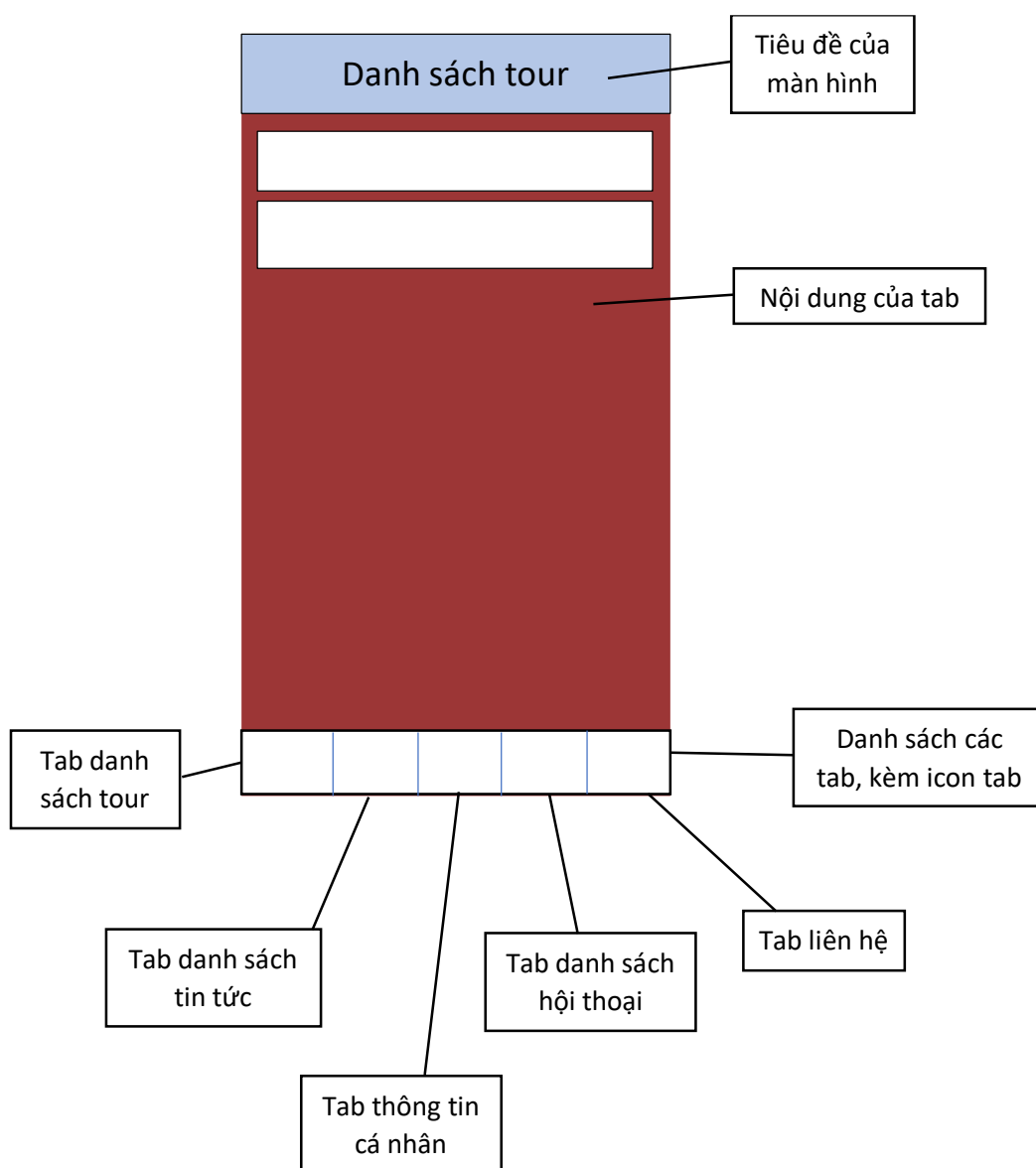
2.6.3. Giao diện đăng nhập



Hình 10. Bản vẽ giao diện đăng nhập

- Người dùng nhập tên đăng nhập và mật khẩu vào khung, sau đó nhấn vào nút đăng nhập, client sẽ gửi JSON message thông tin đăng nhập lên server, đồng thời hiển thị progress bar thông báo đợi phản hồi từ server (mục [2.10.1](#)).

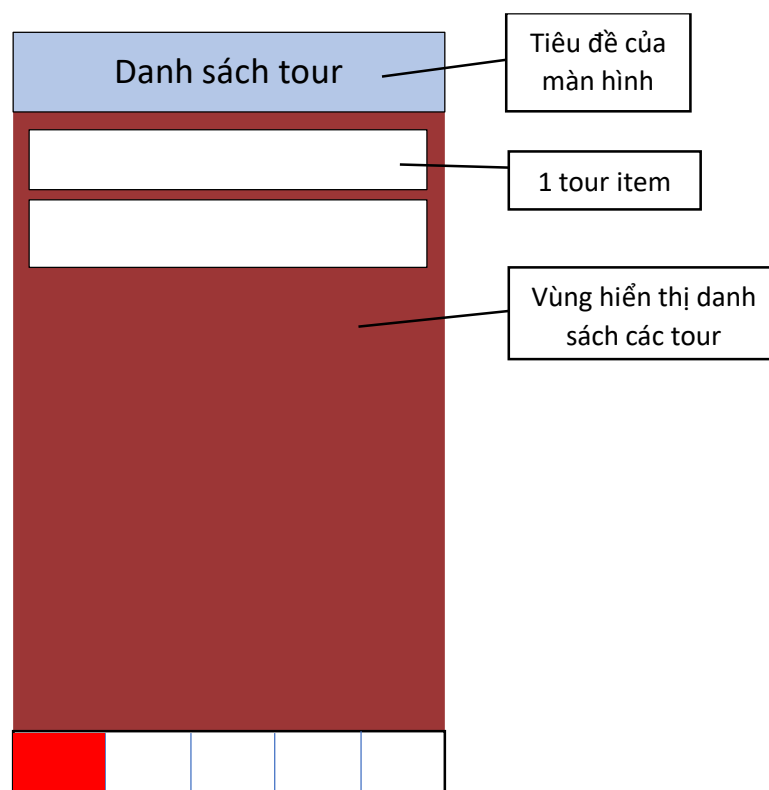
2.6.4. Giao diện sau khi đăng nhập (trang chủ)



Hình 11. Bản vẽ giao diện sau khi đăng nhập (trang chủ)

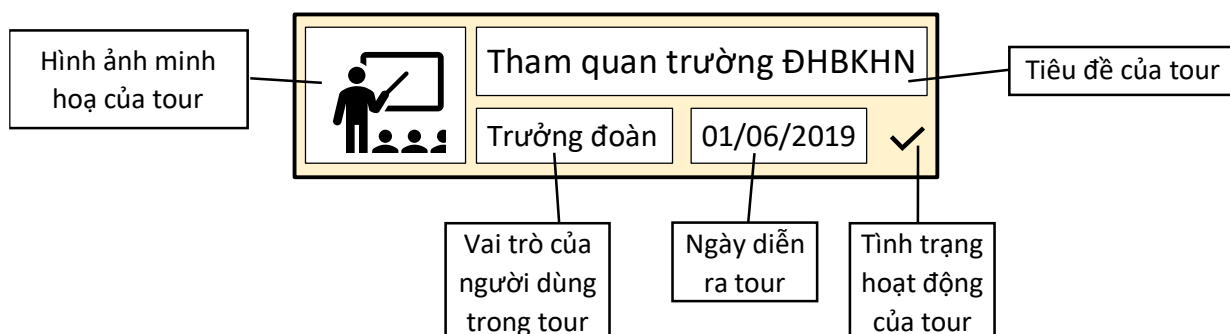
- Sau khi đăng nhập thành công, ứng dụng sẽ hiển thị giao diện trang chủ gồm 5 tab chính, người dùng có thể nhấn vào icon của tab hoặc vuốt qua trái/phải để chuyển qua lại giữa các tab.
- Ứng dụng đang hiển thị tab nào thì icon của tab đó sẽ chuyển sang màu đỏ, icon của các tab khác sẽ chuyển thành màu xám.
- Tab mặc định khi khởi tạo giao diện trang chủ là tab đầu tiên – tab danh sách tour.

2.6.5. Giao diện tab danh sách tour



Hình 12. Bản vẽ giao diện tab danh sách tour

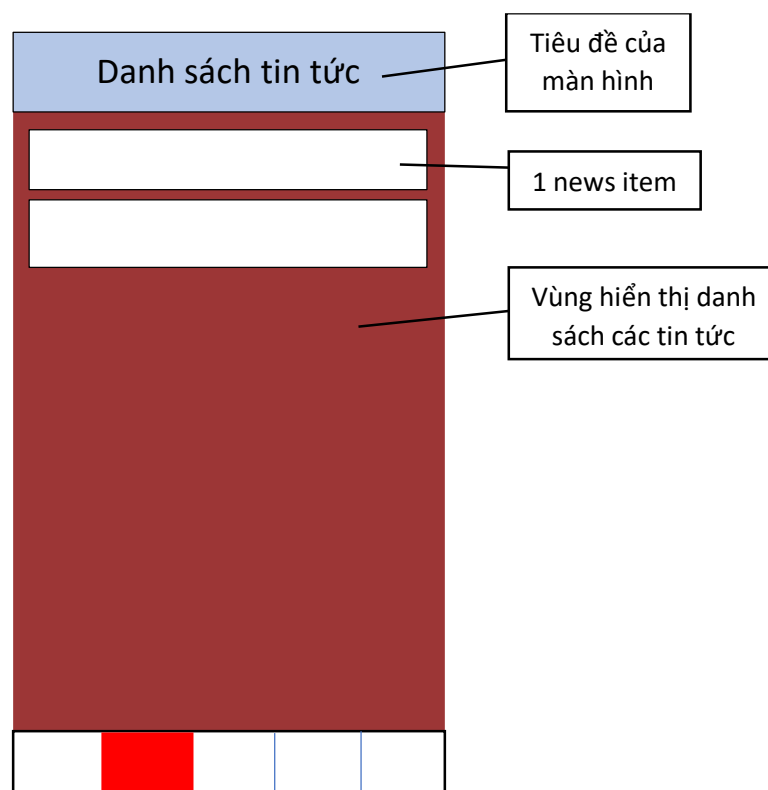
- Giao diện 1 tour item:



Hình 13. Bản vẽ giao diện 1 tour item

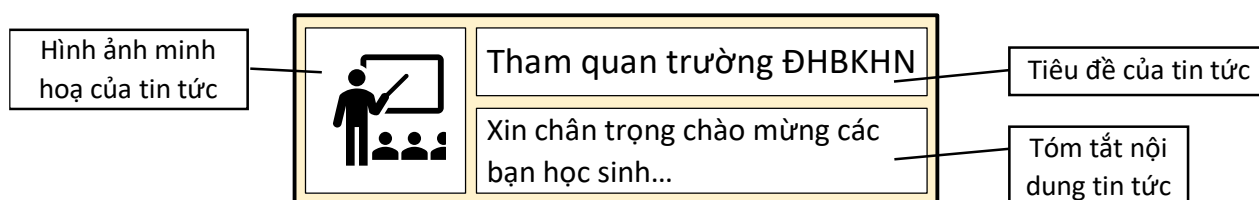
- Tab đầu tiên trong giao diện trang chủ chứa danh sách các tour đã tham gia, đang tham gia và sẽ tham gia của người dùng.
- Từng tour sẽ được hiển thị các thông tin chính trên 1 mục riêng biệt, gọi là 1 tour item.
- Khi người dùng nhấn vào tour item, ứng dụng sẽ chuyển sang giao diện của 1 tour (mục [2.7.8](#)).

2.6.6. Giao diện tab danh sách tin tức



Hình 14. Bản vẽ giao diện tab danh sách tin tức

- Giao diện 1 news item:



Hình 15. Bản vẽ giao diện 1 news item

- Tab thứ 2 trong giao diện trang chủ chứa danh sách các tin tức mà nhà trường đã đăng tải.
- Từng tin tức sẽ được hiển thị các thông tin chính trên 1 mục riêng biệt, gọi là 1 news item.
- Khi người dùng nhấn vào tour item, ứng dụng sẽ gọi trình duyệt web để mở ra đường link của tin tức.

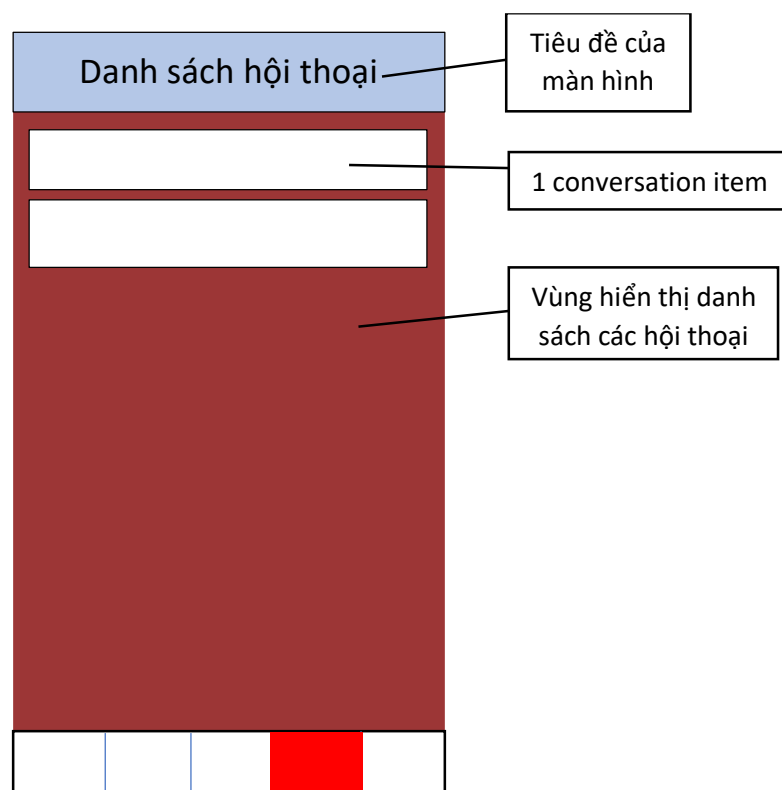
2.6.7. Giao diện tab thông tin cá nhân

The diagram illustrates the 'Thông tin cá nhân' (Personal Information) tab interface. It features a blue header bar with the title 'Thông tin cá nhân'. Below the header are four white input fields for personal information. A 'Đăng xuất' (Logout) button is located below the input fields. At the bottom is a tab bar with five items, the third of which is highlighted in red. Callout boxes identify the title bar, the list of input fields, the display area, the logout button, and the tab bar.

Hình 16. Bản vẽ giao diện tab thông tin cá nhân

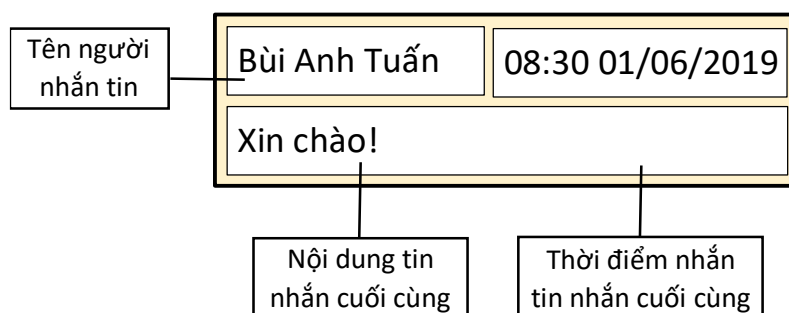
- Tab thứ 3 trong giao diện trang chủ chứa thông tin cá nhân của người dùng.
- Các thông tin cá nhân được hiển thị bao gồm:
 - + Tên đăng nhập
 - + Email
 - + Họ tên đầy đủ
 - + Giới tính
 - + Ngày tháng năm sinh
 - + Trường học
 - + Số điện thoại
 - + Nút đăng xuất
- Khi nhấn nút đăng xuất, ứng dụng sẽ hiển thị 1 dialog hỏi người dùng có muốn đăng xuất không, có 2 lựa chọn là có hoặc không. Nếu người dùng nhấn có, ứng dụng sẽ hiển thị progress bar và gửi JSON message yêu cầu đăng xuất lên server (mục [2.10.4](#)).

2.6.8. Giao diện tab danh sách hội thoại



Hình 17. Bản vẽ giao diện tab danh sách hội thoại

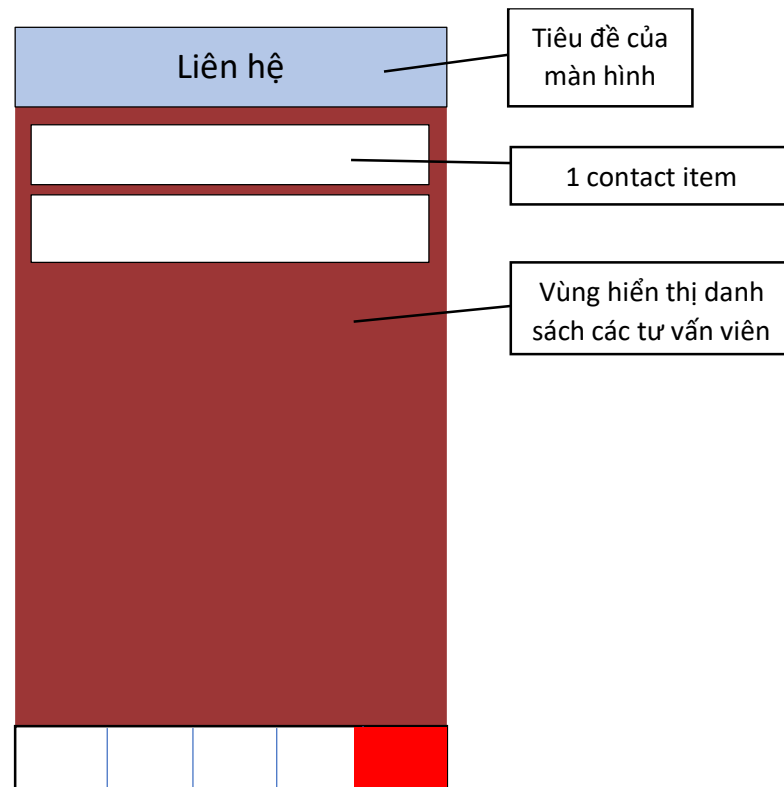
- Giao diện 1 conversation item:



Hình 18. Bản vẽ giao diện 1 conversation item

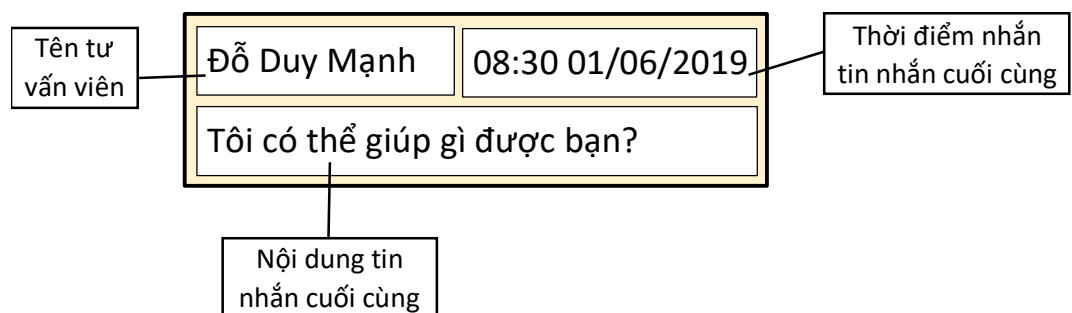
- Tab thứ 4 trong giao diện trang chủ chứa danh sách các đoạn hội thoại với những người dùng khác.
- Từng đoạn hội thoại sẽ được hiển thị các thông tin chính trên 1 mục riêng biệt, gọi là 1 conversation item.
- Khi người dùng nhấn vào conversation item, ứng dụng sẽ chuyển sang giao diện của 1 đoạn hội thoại (mục [2.7.8](#)).

2.6.9. Giao diện tab liên hệ



Hình 19. Bản vẽ giao diện tab liên hệ

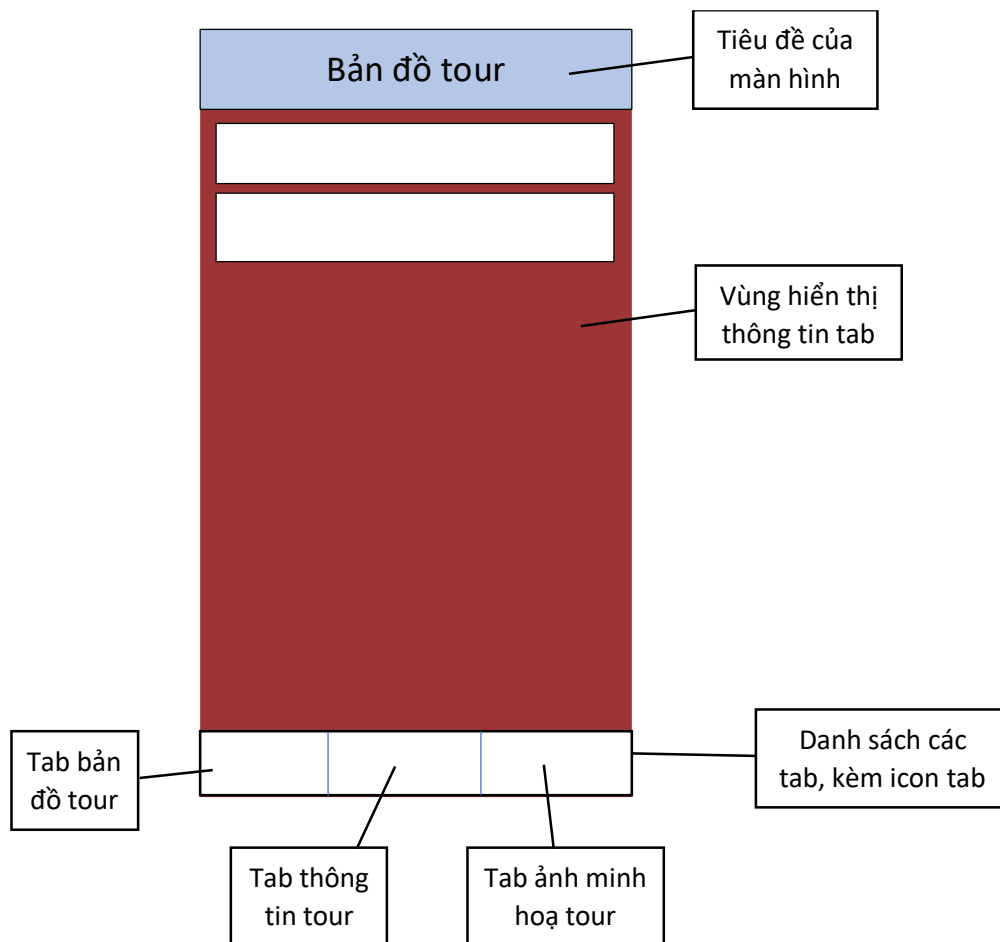
- Giao diện 1 contact item:



Hình 20. Bản vẽ giao diện 1 contact item

- Tab thứ 5 trong giao diện trang chủ chứa danh sách các đoạn hội thoại với những tư vấn viên.
- Từng đoạn hội thoại sẽ được hiển thị các thông tin chính trên 1 mục riêng biệt, gọi là 1 contact item.
- Khi người dùng nhấn vào contact item, ứng dụng sẽ chuyển sang giao diện của 1 đoạn hội thoại (mục [2.7.8](#)).

2.6.10. Giao diện 1 tour



Hình 21. Bản vẽ giao diện 1 tour

- Sau khi nhấn vào 1 tour item, ứng dụng sẽ hiển thị giao diện của 1 tour gồm 3 tab chính, người dùng có thể nhấn vào icon của tab hoặc vuốt qua trái/phải để chuyển qua lại giữa các tab.
- Ứng dụng đang hiển thị tab nào thì icon của tab đó sẽ chuyển sang màu đỏ, icon của các tab khác sẽ chuyển thành màu xám.
- Tab mặc định khi khởi tạo giao diện trang chủ là tab đầu tiên – tab bản đồ tour.

2.6.11. Giao diện tab bản đồ tour

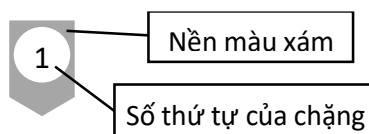


Hình 22. Bản vẽ giao diện bản đồ tour

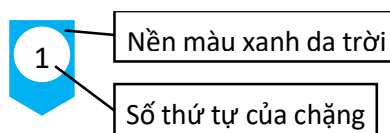
- Tab đầu tiên trong giao diện của 1 tour hiển thị bản đồ của tour đó.
- Các thành phần của tour hiển thị trên bản đồ:

+ Marker:

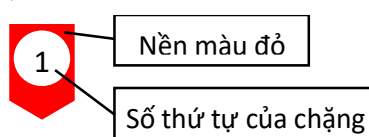
- Công dụng: Dùng để đánh dấu vị trí chặng
- Hình dạng:
 - Chặng đã đi qua:



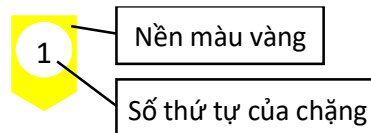
- Chặng đang đi qua:



- Chặng chuẩn bị tới:

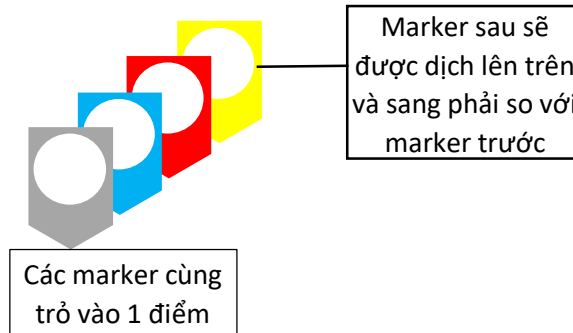


- Các chặng còn lại chưa đi qua:

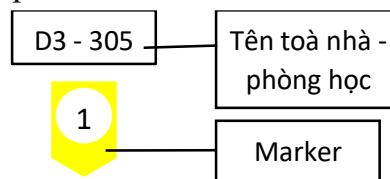


Hình 23. Bản vẽ giao diện các marker

- Nếu có 2 marker trùng nhau thì marker sau sẽ được dịch đi 0.00005 độ về phía Đông và 0.00005 độ về phía Bắc:



- Sự kiện khi nhấn vào: Hiện thị ra tên của toà nhà - phòng học mà chặng đó đi qua:

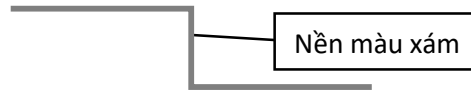


- Sự kiện khi nhấn vào tên toà nhà – phòng học (gọi là title của chặng): Hiện thị [giao diện thông tin của chặng](#) (tên toà nhà, phòng học, tầng, tên người quản lý, email, số điện thoại người quản lý,... - mục [2.7.11](#))
- Thuật toán dùng để xác định màu sắc:
 - Với mỗi chặng, có 1 thuộc tính chứa thời điểm bắt đầu chặng và thời điểm kết thúc chặng, gọi là startTime và endTime.
 - Trên client dùng 1 hàm lấy thời điểm hiện tại, gọi là currentTime
 - Tiến hành duyệt từng chặng, với mỗi chặng, so sánh currentTime với startTime và endTime:
 - Nếu currentTime trước startTime thì tô cho marker màu xám.
 - Nếu currentTime sau startTime và trước endTime thì tô cho marker màu xanh da trời.
 - Nếu currentTime sau endTime thì tô màu cho chặng đầu tiên thoả mãn điều kiện này màu đỏ, những chặng sau đó màu vàng.

+ Route:

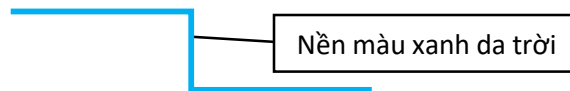
- Công dụng: Dùng để chỉ dẫn đường đi giữa các chặng
- Hình dạng:

- Chặng vừa đi qua:

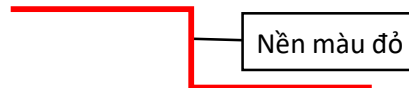


- Các chặng đã đi qua trước chặng vừa đi qua sẽ không được vẽ trên bản đồ.

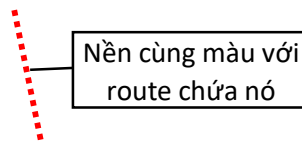
- Chặng đang đi qua:



- Chặng sắp đi qua:



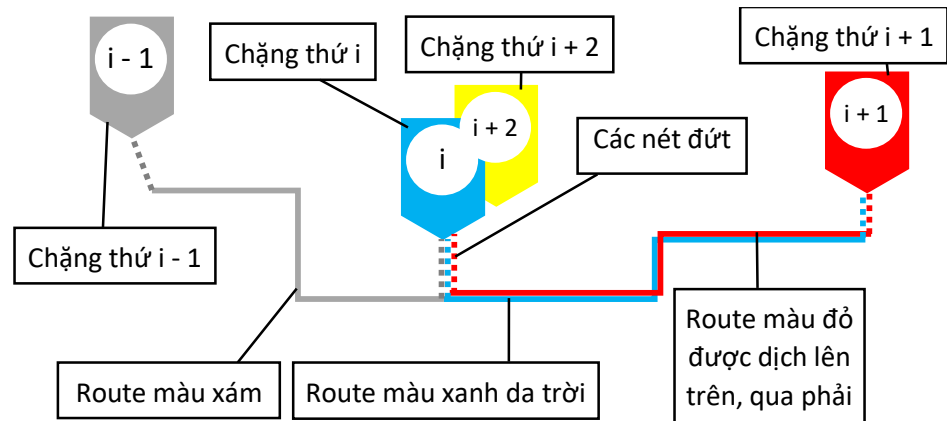
- Vì Google map api không thể chỉ đường chi tiết đến mức toà nhà nên cần vẽ thêm đường đi vào trong toà nhà bằng nét đứt:



Hình 24. Bản vẽ giao diện các route

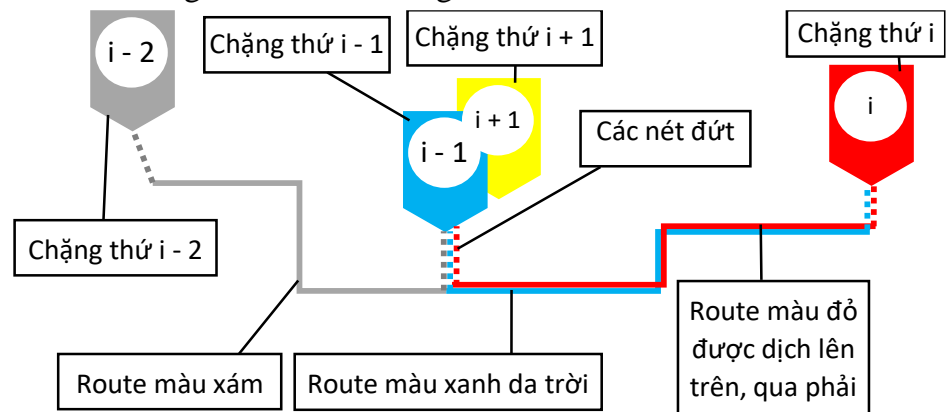
- Các chặng chưa đi qua sau chặng sắp đi qua sẽ không được vẽ trên bản đồ.
- Thuật toán dùng để xác định màu sắc:
 - Để vẽ được route cần có một chặng nguồn và 1 chặng đích. Duyệt danh sách chặng, gán chặng thứ i là chặng nguồn, chặng thứ $i + 1$ là chặng đích. Thời điểm bắt đầu chặng nguồn là `srcStartTime`, thời điểm kết thúc chặng nguồn là `srcEndTime`, thời điểm bắt đầu chặng đích là `dstStartTime`.
 - Trên client dùng 1 hàm lấy thời điểm hiện tại, gọi là `currentTime`
 - Nếu `currentTime` nằm giữa `srcStartTime` và `srcEndTime` thì vẽ đường đi giữa chặng thứ i và chặng thứ $i + 1$ màu xanh, thoát khỏi vòng lặp.
 - Tiếp tục kiểm tra i để vẽ nốt 2 route còn lại.
 - Nếu chặng thứ $i + 1$ chưa phải chặng cuối cùng thì vẽ route giữa chặng thứ $i + 1$ và chặng thứ $i + 2$ màu đỏ.

- Nếu chặng thứ i không phải chặng đầu tiên thì vẽ route giữa chặng thứ $i - 1$ và chặng thứ i màu xám.



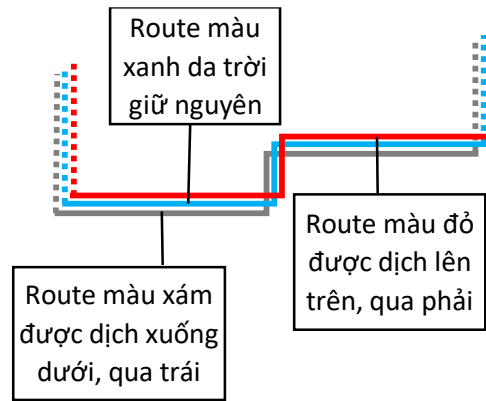
Hình 25. Bản vẽ minh họa các route khi biết 1 route màu xanh da trời

- Nếu currentTime nằm giữa srcEndTime và dstStartTime thì vẽ đường đi giữa chặng thứ i và chặng thứ $i + 1$ màu đỏ, thoát khỏi vòng lặp.
 - Tiếp tục kiểm tra i để vẽ nốt 2 route còn lại.
 - Nếu chặng thứ i không phải chặng đầu tiên thì vẽ route giữa chặng thứ $i - 1$ và chặng thứ i màu xanh da trời.
 - Nếu chặng thứ $i - 1$ không phải chặng đầu tiên thì vẽ route giữa chặng thứ $i - 2$ và chặng thứ $i - 1$ màu xám.



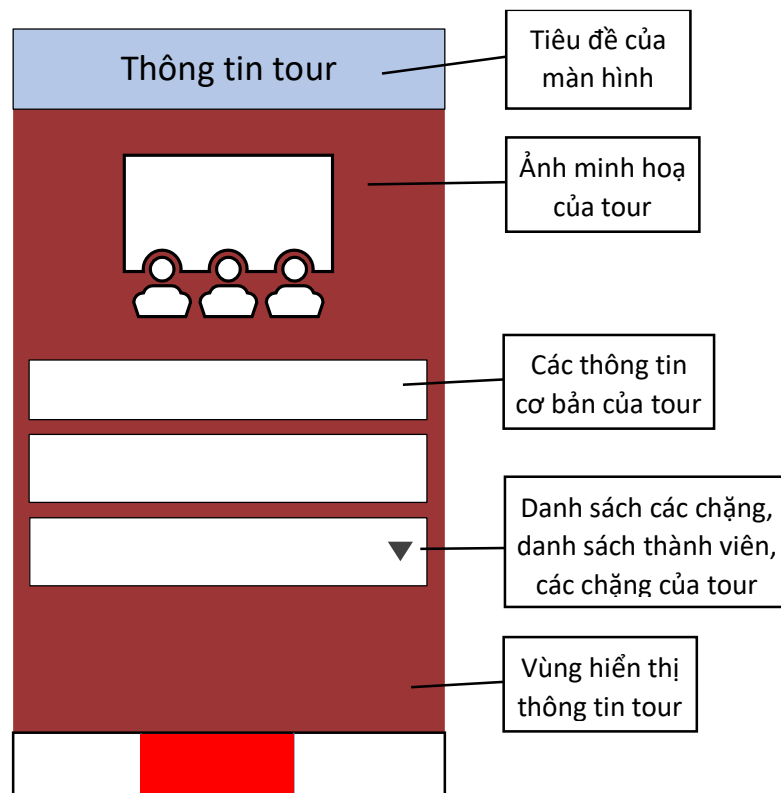
Hình 26. Bản vẽ minh họa các route khi biết 1 route màu đỏ

- Để tránh trường hợp các route đè lên nhau thì route màu xám sẽ được dịch 0.00001 độ về hướng Tây, 0.00001 về hướng Nam; route màu đỏ sẽ được dịch 0.00001 độ về hướng Đông, 0.00001 về hướng Bắc; route màu xanh da trời giữ nguyên:



Hình 27. Bản vẽ minh họa các route dịch đi 1 khoảng để không bị trùng với route khác

2.6.12. Giao diện tab thông tin tour

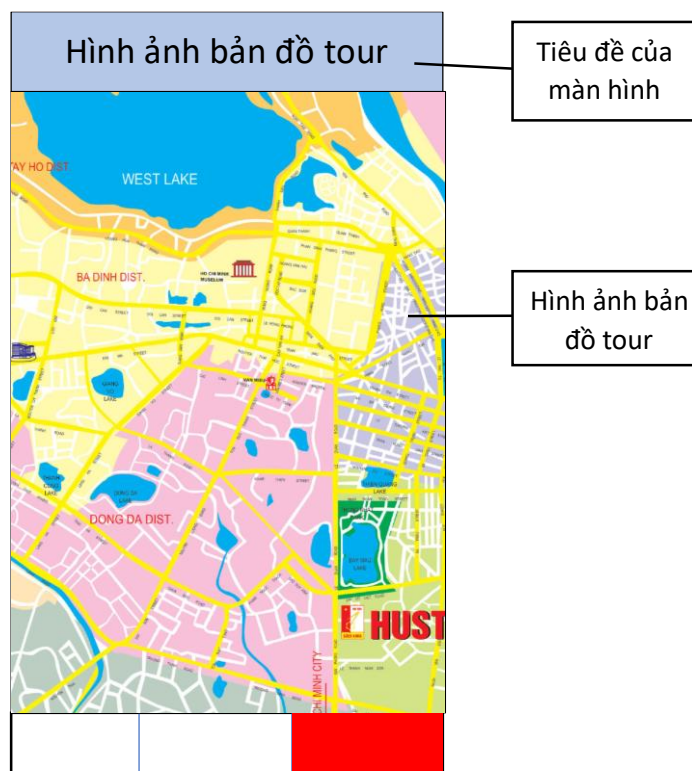


Hình 28. Bản vẽ giao diện bản đồ tour

- Tab thứ 2 trong giao diện của 1 tour hiển thị thông tin của tour đó.
- Các thông tin sẽ được hiển thị trong tab thông tin tour:
 - + Ảnh minh họa của tour
 - + Tên của tour
 - + Trạng thái của tour (đã trôi qua, đang diễn ra, chưa diễn ra)
 - + Vai trò của người dùng trong tour (trưởng đoàn, phó đoàn, phụ huynh, học sinh – thành viên bình thường)
 - + Ngày diễn ra tour

- + Danh sách thành viên trong tour
- + Danh sách các chặng trong tour

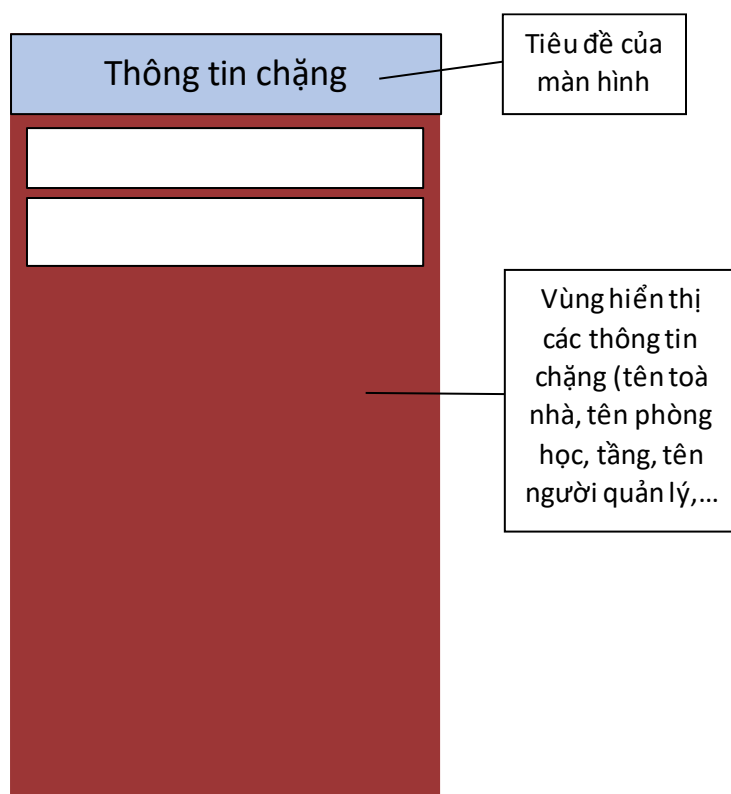
2.6.13. Giao diện tab hình ảnh bản đồ tour



Hình 29. Bản vẽ giao diện bản đồ tour

- Tab thứ 2 trong giao diện của 1 tour hiển thị thông tin của tour đó.
- Các thông tin sẽ được hiển thị trong tab thông tin tour:
 - + Ảnh minh họa của tour
 - + Tên của tour
 - + Trạng thái của tour (đã trôi qua, đang diễn ra, chưa diễn ra)
 - + Vai trò của người dùng trong tour (trưởng đoàn, phó đoàn, phụ huynh, học sinh – thành viên bình thường)
 - + Ngày diễn ra tour
 - + Danh sách thành viên trong tour
 - + Danh sách các chặng trong tour

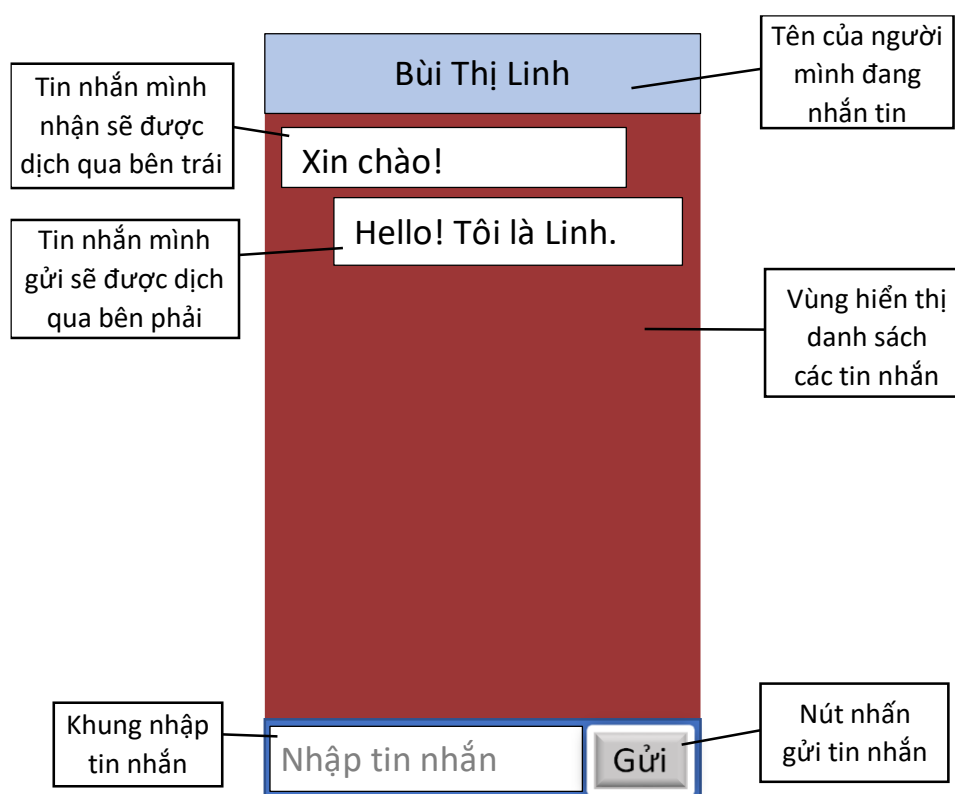
2.6.14. Giao diện thông tin của chặng



Hình 30. Bản vẽ giao diện thông tin của chặng

- Sau khi nhấn vào title của chặng hoặc 1 timesheet item trong tab thông tin tour của giao diện 1 tour, ứng dụng sẽ hiển thị giao diện thông tin của 1 chặng bao gồm các thông tin: thời gian của chặng (thời điểm bắt đầu – thời điểm kết thúc), tên của toà nhà, tên của phòng học, tầng của phòng học, tên người quản lý, số điện thoại của người quản lý, email của người quản lý.

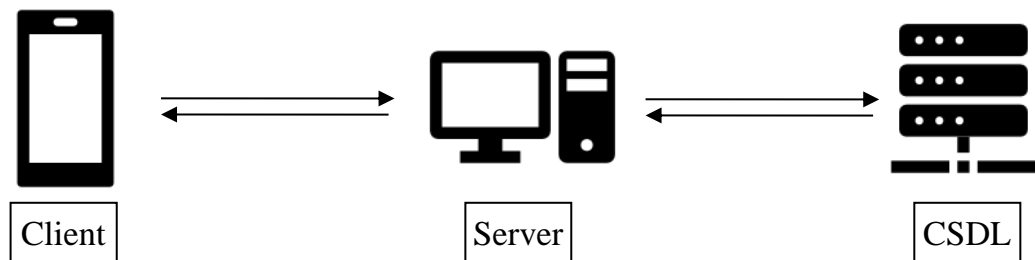
2.6.15. Giao diện nhắn tin



Hình 31. Bản vẽ giao diện nhắn tin

- Sau khi nhấn vào tên của 1 thành viên trong tour, nhấn vào 1 đoạn hội thoại trong giao diện danh sách hội thoại hoặc nhấn vào 1 tư vấn viên trong giao diện danh sách tư vấn viên, ứng dụng sẽ hiển thị giao diện nhắn tin với người mình nhấn vào.
- Người dùng nhập tin nhắn cần gửi vào khung nhập tin nhắn, sau khi người dùng nhấn vào nút gửi, ứng dụng sẽ gửi message nhắn tin lên server, server sẽ thêm vào cơ sở dữ liệu vào gửi kết quả về cho client. Chi tiết giao thức xem mục Nhắn tin.

2.7. Mô hình tổng thể các thiết bị trong hệ thống



Client: Thiết bị điện thoại di động của người dùng (học sinh, phụ huynh, trưởng đoàn, phó đoàn,...). Thiết bị cần cài đặt hệ điều hành Android.

Server: Máy chủ cung cấp dịch vụ, xử lý các yêu cầu từ phía client. Server cần chạy trên 1 thiết bị ổn định, hoạt động liên tục 24/7. Thiết bị cần hỗ trợ cài đặt Node.js.

CSDL: Cơ sở dữ liệu, nơi lưu trữ tất cả dữ liệu của người dùng. Thiết bị cài đặt cơ sở dữ liệu cần hỗ trợ Microsoft Sql Server.

Hình 32. Mô hình tổng thể các thiết bị trong hệ thống

2.8. Các giao thức

- Client giao tiếp với server thông qua các JSON message được gửi qua Socket
- + Cấu trúc các gói tin JSON:

```
{ "COMMAND": "LOGIN",  
  "USERNAME": "a@a.a",  
  "PASSWORD": "oj25#%4nf3" }
```

- Key đầu tiên là `COMMAND`, cho biết mục đích của gói tin
- Các key tiếp theo chứa thông tin của gói tin
- Các key và value của key `COMMAND` phải viết hoa toàn bộ
- Server cũng giao tiếp với client thông qua các JSON message gửi qua Socket
- Server giao tiếp với csdl thông qua các câu lệnh truy vấn sql

2.8.1. Gửi thông tin cài đặt

- Server tự động gửi thông tin cài đặt cho mỗi client kết nối đến:

```
{ "COMMAND": "SETTING",  
  "FORM_URL": "http://bit.ly/DangKyTraiNghiemHUST" (Địa chỉ form  
  đăng ký trải nghiệm HUST) }
```

2.8.2. Đăng nhập

- Client gửi message đăng nhập lên server:

```
{ "COMMAND": "LOGIN",  
  "USERNAME": "a@a.a" (Địa chỉ email của người đăng nhập),  
  "PASSWORD": "oj25#%4nf3" (Mật khẩu của người đăng nhập) }
```

- Server kiểm tra thông tin đăng nhập trong csdl:

```
SELECT * FROM [User] WHERE Username = 'a@a.a' AND Password =  
'oj25#%4nf3'
```

- Nếu tìm thấy bản ghi thì gửi message thông tin tài khoản về client:

```
{ "COMMAND": "LOGIN",  
  "RESULT": "success",  
  "USER_ID": 6 (UserId của người đăng nhập),  
  "USERNAME": "a@a.a" (Địa chỉ email của người đăng nhập),  
  "FULLNAME": "Lê Mạnh Hùng" (Họ tên đầy đủ của người đăng nhập),  
  "BIRTHDAY": "2000-21-07" (Ngày sinh của người đăng nhập),  
  "GENDER": "1" (Giới tính của người đăng nhập),  
  "SCHOOL": "" (Trường người đăng nhập đang học),  
  "CLASS": "" (Lớp người đăng nhập đang học),  
  "IS_COUNSELOR": 0 (Đánh dấu có phải tư vấn viên không),  
  "STATE": 1 (Trạng thái đăng nhập của người đăng nhập) }
```

- Đồng thời cập nhật trạng thái đăng nhập là đang online:

```
UPDATE [User] SET State = 1 WHERE UserId = 6
```

- Nếu không tìm thấy bản ghi nào thì gửi message lỗi đăng nhập về client:

```
{ "COMMAND": "LOGIN",  
  "RESULT": "fail",  
  "REASON": "Lý_do_đăng_nhập_thất_bại" }
```

+ REASON có thể là:

- WRONG_EMAIL_PASSWORD nếu sai email hoặc mật khẩu.
- UNKNOWN_ERROR nếu gặp lỗi không xác định.
- CONFLICT_ERROR nếu có ai đó đang dùng tài khoản này.

- Đồng thời truy vấn danh sách thành viên trong những tour có sự tham gia của mình (User Id là 3):

```
SELECT DISTINCT UserId FROM Tour2Member  
WHERE TourId IN (SELECT TourId FROM Tour2Member WHERE UserId = 3)
```

- Kiểm tra trong danh sách socket kết nối tới server, nếu socket nào có UserId nằm trong danh sách UserId vừa truy vấn thì gửi message thông báo mình vừa online cho socket này:

```
{ "COMMAND": "MEMBER_ONLINE",  
  "USER_ID": 3 }
```

- Client nhận được message sẽ refresh danh sách thành viên

2.8.3. Lấy danh sách tin tức

- Client gửi yêu cầu danh sách tin tức lên server:

```
{ "COMMAND": "GET_NEWS_LIST" }
```

- Server truy vấn danh sách tin tức còn hoạt động trong csdl:

```
SELECT * FROM News WHERE IsShowing = 'True'
```

- Sau đó Server gửi danh sách tin tức về client:

```
{ "COMMAND": "GET_NEWS_LIST", "NEWS": [] }
```

+ Các phần tử của mảng NEWS:

```
{ "NEWS_ID": 3,  
  "IMAGE_URL": "https://ảnh_minh_hoạ_tin_tức.png",  
  "TITLE": "Bách khoa Open Day 2019" (Tiêu đề tin tức),  
  "URL": "https://liên_kết_tới_trang_tin_tức",  
  "SUMMARY": "Tóm tắt nội dung tin tức" }
```

2.8.4. Lấy danh sách tour

- Client gửi yêu cầu danh sách tin tức lên server:

```
{"COMMAND": "GET_TOUR_LIST",  
"USERID": 1 (UserId của người muốn lấy danh sách tour)}
```

- Server truy vấn danh sách tour trong csdl:

```
SELECT * FROM News WHERE IsShowing = 'True'
```

- Sau đó Server gửi danh sách tour về client:

```
{"COMMAND": "GET_TOUR_LIST", "TOURS": []}
```

- + Các phần tử của mảng TOURS:

```
{"TOUR_ID": 3,  
"NAME": "Tên của tour",  
"IMAGE_URL": "https://ảnh_minh_hoạ_tour.png",  
"STATE": 1 (Trạng thái hoạt động của tour),  
"MAP_IMAGE_URL": "https://hình_ảnh_bản_đồ_tour.png",  
"FUNCTION": 1 (Vai trò của mình trong tour),  
"MEMBERS": [], (Danh sách các thành viên trong tour)  
"TIMESHEETS": [] (Danh sách các chặng của tour)}
```

- Các phần tử của mảng MEMBERS:

```
{"USERID": 6 (UserId của thành viên),  
"FUNCTION": 0 (Vai trò của thành viên trong tour),  
"LATITUDE": 21.004645 (Vĩ độ của thành viên trên Google Map),  
"LONGITUDE": 105.845266 (Kinh độ của thành viên trên Google Map),  
"NOTE": "Ghi chú về thành viên",  
"USERNAME": "a@a.a" (Địa chỉ email của thành viên),  
"FULLNAME": "Lê Mạnh Hùng" (Họ tên đầy đủ của thành viên),  
"BIRTHDAY": "2000-21-07" (Ngày sinh của thành viên),  
"GENDER": "1" (Giới tính của thành viên),  
"SCHOOL": "" (Trường của thành viên),  
"CLASS": "" (Lớp của thành viên đang học),  
"IS_COUNSELOR": 0 (Đánh dấu thành viên có phải tư vấn viên không),  
"STATE": 1 (Trạng thái online của thành viên)}
```

- Các phần tử của mảng `TIMESHEETS`:

```
{ "TIMESHEET_ID": 3 (Id của chặng trong tour),  
  "START_TIME": 07:45:00 (Thời điểm bắt đầu chặng),  
  "END_TIME": 08:15:00 (Thời điểm kết thúc chặng),  
  "CLASSROOM_ID": 3 (Id của lớp học mà chặng sẽ tham quan),  
  "CLASSROOM_FLOOR": 3 (Tầng của lớp học),  
  "CLASSROOM_NAME": 305 (Tên của lớp học),  
  "CLASSROOM_SUB_NAME": D3-305 (Tên đầy đủ/ tên khác của lớp học),  
  "CLASSROOM_NOTE": "Phòng thí nghiệm" (Ghi chú của lớp học),  
  "BUILDING_NAME": "D3" (Tên toà nhà chứa phòng học),  
  "BUILDING_SUB_NAME": "" (Tên khác/ tên đầy đủ của toà nhà),  
  "LATITUDE": 21.004645 (Vĩ độ của toà nhà trên Google Map),  
  "LONGITUDE": 105.845266 (Kinh độ của toà nhà trên Google Map),  
  "BUILDING_NOTE": "" (Chú thích về toà nhà)  
  "MANAGERS": [] (Danh sách người quản lý lớp học) }
```

- Các phần tử của mảng `MANAGERS`:

```
{ "MANAGER_ID": 3 (Id của người quản lý phòng học),  
  "NAME": "Lê Thanh Nghị" (Tên người quản lý),  
  "GENDER": 1 (Giới tính của người quản lý),  
  "BIRTHDAY": "1990-03-12" (Ngày sinh của người quản lý),  
  "EMAIL": (Email của người quản lý),  
  "PHONE_NUMBER": (Số điện thoại của người quản lý) }
```

2.8.5. Đăng xuất

- Client gửi message đăng xuất lên server:

```
{ "COMMAND": "LOGOUT" }
```

- Server tiến hành đăng xuất cho UserId đã liên kết với Socket trước đó:

```
UPDATE [User] SET State = 0 WHERE UserId = socket.UserId
```

- Nếu đăng xuất thành công thì gửi thông báo về cho client:

```
{ "COMMAND": "LOGOUT",  
  "RESULT": "success" }
```

- Nếu đăng xuất thất bại thì gửi thông báo về cho client:

```
{ "COMMAND": "LOGOUT",  
  "RESULT": "fail",  
  "REASON": "Lý_doĐăngXuấtThấtBại" }
```

- + `REASON` có thể là:

- `DATABASE_ERROR` nếu lỗi csdl.
- `UNKNOWN_ERROR` nếu gặp lỗi không xác định.

2.8.6. Tọa độ của thành viên

- Client gửi message tọa độ hiện tại của mình lên server:

```
{ "COMMAND": "MEMBER_CURENT_LOCATION", "LATITUDE": 21.404645,  
  "LONGITUDE": 105.245266 }
```


- Server tìm kiếm trong csdl danh sách các member có user id đính kèm socket và gán tọa độ:

```
UPDATE Tour2Member set mLocation = 'POINT(21.404645 105.245266)'  
WHERE UserId = 3
```

- Server tìm các tour có mặt user này và gửi vị trí của user này cho tất cả các user còn lại:

```
SELECT UserId FROM Tour2Member WHERE TourId IN (SELECT TourId FROM  
Tour2Member WHERE UserId = 3)
```

- Message gửi cho các user còn lại:

```
{"COMMAND": "MEMBER_LOCATION_CHANGE",  
"USER_ID": 3,  
"LATITUDE": 21.004645 (Vĩ độ của thành viên vừa thay đổi),  
"LONGITUDE": 105.845266 (Kinh độ của thành viên vừa thay đổi)}
```

2.8.7. Lấy danh sách hội thoại

- Client có User Id là 3 gửi yêu cầu danh sách hội thoại lên server:

```
{"COMMAND": "GET_CONVERSATIONS_LIST"}
```

- Server truy vấn danh sách hội thoại có mặt Client này:

```
SELECT * FROM [Message] WHERE SenderId = 3 OR RecieverId = 3 ORDER  
BY [Time] ASC;
```

- Server gửi danh sách hội thoại về cho Client:

```
{"COMMAND": "CONVERSATIONS_LIST",  
"CONVERSATIONS_LIST": [] (Danh sách hội thoại)}
```

+ Các phần tử của mảng `CONVERSATIONS_LIST`:

```
{"PARTNER_ID": 2 (Id của người nhắn tin với User này),  
"PARTNER_NAME": "Vũ Thanh Tùng" (Tên của người nhắn tin với User  
này),  
"MESSAGES": [] (Danh sách tin nhắn 2 người đã nhắn với nhau)}
```

- Các phần tử của mảng `MESSAGES`:

```
{"IS_SENDER": 1 (Đánh dấu User này là người gửi hay nhận tin nhắn  
(0 là người nhận, 1 là người gửi),  
"TIME": "2019-05-21 17:05:30.230" (Thời gian gửi tin nhắn),  
"CONTENT": "Hello" (Nội dung của tin nhắn)}
```

2.8.8. Lấy danh sách tư vấn viên

- Client gửi yêu cầu danh sách tư vấn viên lên server:

```
{"COMMAND": "GET_COUNSELORS_LIST"}
```

- Server truy vấn danh sách User có trường `IsCounselor` là 1 (hoặc true):

```
SELECT UserId, Username, Fullname, Gender, State FROM [User] WHERE  
IsCounselor = '1' ORDER BY UserId ASC;
```

- Server gửi danh sách tư vấn viên về cho Client:

```
{ "COMMAND": "COUNSELORS_LIST",  
  "COUNSELORS_LIST": [] (Danh sách tư vấn viên) }
```

- + Các phần tử của mảng COUNSELORS_LIST:

```
{ "USER_ID": 2 (Id của người nhắn tin với User này),  
  "USERNAME": "thanhvt@gmail.com" (Tên đăng nhập của tư vấn viên),  
  "FULLNAME": "Vũ Thị Thanh" (Tên đầy đủ của tư vấn viên),  
  "GENDER": "1" (Giới tính của tư vấn viên),  
  "STATE": "1" (Trạng thái đăng nhập của tư vấn viên) }
```

2.8.9. Nhắn tin

- Client có User Id là 2 gửi tin nhắn lên server:

```
{ "COMMAND": "MEMBER_MESSAGE",  
  "SENDER_NAME": "Bùi Anh Tuấn" (Tên của người gửi tin nhắn),  
  "RECIEVER_ID": 3 (User Id của người mình muốn gửi tin nhắn),  
  "RECIEVER_NAME": "Trần Ngọc Trinh" (Tên của người nhận tin nhắn),  
  "CONTENT": "Hello" (Nội dung tin nhắn) }
```

- Server thêm 1 tin nhắn vào cơ sở dữ liệu (người gửi có User Id là 2):

```
INSERT INTO [Message] (SenderId, RecieverId, mContent, [Time])  
VALUES (2, 3, 'Hello', getdate());
```

- Server gửi tin nhắn này cho cả 2 nếu không gặp lỗi gì:

```
{ "COMMAND": "MEMBER_MESSAGE",  
  "RESULT": "success",  
  "SENDER_ID": 2 (User Id của người gửi),  
  "SENDER_NAME": "Bùi Anh Tuấn" (Tên của người gửi),  
  "RECIEVER_ID": 3 (User Id của người nhận),  
  "RECIEVER_NAME": "Trần Ngọc Trinh" (Tên của người nhận),  
  "CONTENT": "Hello" (Nội dung tin nhắn),  
  "TIME": '2016-06-01 12:00:00' (Thời gian gửi tin nhắn) }
```

- Server gửi tin nhắn này cho cả 2 nếu gặp lỗi:

```
{ "COMMAND": "MEMBER_MESSAGE",  
  "RESULT": "fail",  
  "REASON": "UNKNOWN_ERROR" (Mã lỗi) }
```

- Sau đó cả người gửi và người nhận mới update tin nhắn mới trên màn hình.

2.8.10. Kết nối lại

- Khi client không hoạt động 1 thời gian, kết nối socket đến server sẽ bị ngắt (timeout).
- Khi client hoạt động trở lại, nếu đã đăng nhập cần kết nối lại đến server để sử dụng các tính năng của 1 người dùng đã đăng nhập.

- Nội dung message yêu cầu kết nối lại như sau:

```
{ "COMMAND": "RECONNECT",  
  "USER_ID": 3 (User Id của người dùng muốn kết nối lại) }
```

- Server gán user id của người dùng vào kết nối socket hiện tại, đoạn code minh họa trên server (Node.js) như sau:

```
socket.UserId = message.USER_ID;
```

- Server gửi tin nhắn thông báo người dùng kết nối lại thành công nếu không gặp lỗi gì:

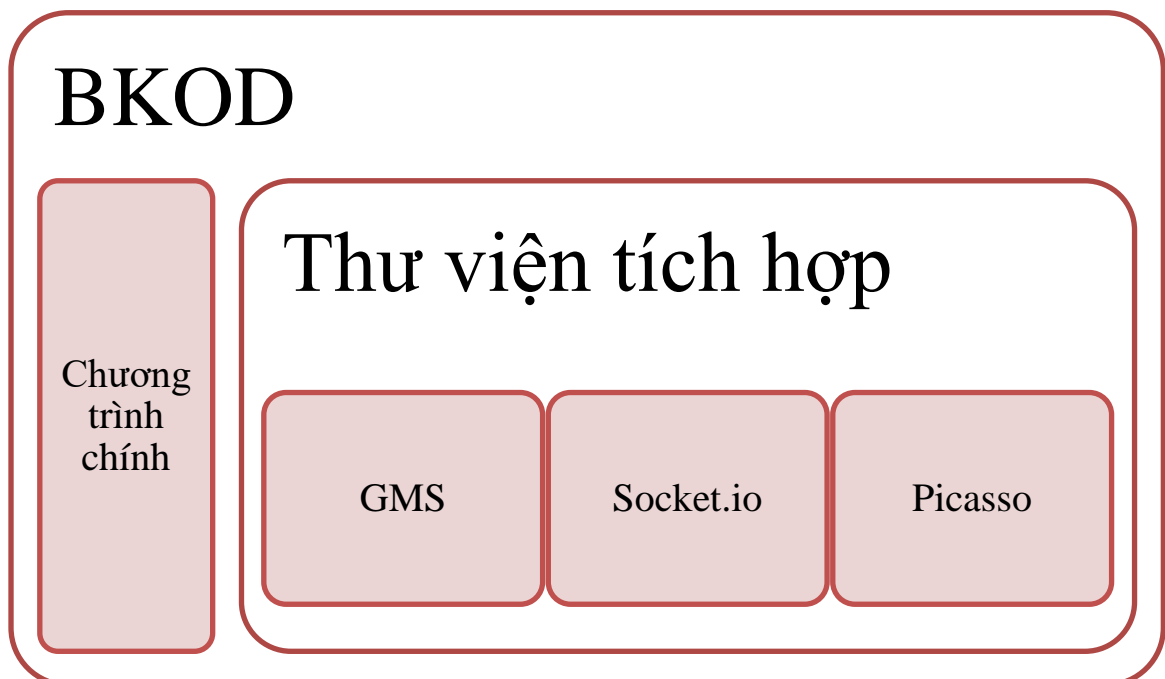
```
{ "COMMAND": "RECONNECT",  
  "RESULT": "success" }
```

- Nếu có lỗi thì nội dung message sẽ như sau:

```
{ "COMMAND": "RECONNECT",  
  "RESULT": "fail" }
```

- Client phân tích message từ server, nếu `RESULT` là `fail` thì client gửi lại yêu cầu kết nối lại lên server.

2.9. Mô hình tích hợp phần cứng/phần mềm trong hệ thống



- Ngoài những thành phần giao diện mặc định của Android ra, chương trình còn tích hợp thêm các thư viện ngoài như sau:

- + **GMS:** Thư viện hỗ trợ các tính năng liên quan đến bản đồ và định vị của Google. Tên đầy đủ: `com.google.android.gms:play-services-maps:16.1.0`.
- + **Socket.io:** Thư viện hỗ trợ giao tiếp giữa client và server. Tên đầy đủ: `io.socket:socket.io-client:1.0.0`.
- + **Picasso:** Dùng để tải hình ảnh khi có đường dẫn của ảnh. Tên đầy đủ: `com.squareup.picasso:picasso:2.5.0`.
- **Một số thư viện khác:**
 - + **AR:** Thư viện hỗ trợ góc nhìn thực tế tăng cường cho client, đang phát triển.

PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

3.1. Sản phẩm

3.1.1. Database

- Cơ sở dữ liệu chạy trên Microsoft SQL Server, bao gồm:
 - + 1 cơ sở dữ liệu tên là BKOD:
 - 13 bảng, chi tiết các bảng xem mục Thiết kế cơ sở dữ liệu:
 - Bảng User: Danh sách tài khoản.
 - Bảng Tour: Danh sách tour.
 - Bảng Tour2Member: Liên kết tour và danh sách thành viên.
 - Bảng Message: Danh sách tin nhắn trong 1 cuộc hội thoại.
 - Bảng Timesheet: Danh sách các chặng tham quan của tour.
 - Bảng Tour2Timesheet: Liên kết tour và chặng.
 - Bảng Classroom: Danh sách các phòng học.
 - Bảng Manager: Danh sách các người quản lý phòng học theo từng khung giờ.
 - Bảng Timesheet2Classroom: Liên kết chặng, phòng học và người quản lý.
 - Bảng Building: Danh sách các toà nhà của trường.
 - Bảng Building2Classroom: Liên kết toà nhà và các phòng học trong đó.
 - Bảng News: Danh sách tin tức.
 - Bảng Setting: Cài đặt.

3.1.2. Client

- 1 ứng dụng Android, bao gồm:
 - + 1 file cài đặt .apk.
 - + 1 thư mục project tại đường dẫn `Source\BKOD-AndroidClient`, bao gồm:
 - 1 thư mục mã nguồn tại đường dẫn `Source\BKOD-AndroidClient\app\src\main\java`, bao gồm:
 - Thư mục `Classes`, chứa các lớp hỗ trợ, bao gồm:
 - File `NDSpinner.java`: Hỗ trợ các thao tác chọn các mục trong danh sách xổ xuống (danh sách thành viên, danh sách chặng trong 1 tour,...).
 - File `UserInfo.java`: Chứa các thông tin người dùng.

- File `ZoomableImageView.java`: Hỗ trợ thao tác thu phóng hình ảnh ở tab hình ảnh minh họa tour.
- Thư mục `com\example\administrator\bkod_androidclient`, chứa các lớp giao diện quan trọng, bao gồm:
 - Thư mục `Adapter`, chứa các lớp hỗ trợ hiển thị giao diện các mục trong danh sách, bao gồm:
 - ❖ File `ConversationsAdapter.java`: Hiển thị giao diện các mục trong danh sách hội thoại.
 - ❖ File `CounselorsAdapter.java`: Hiển thị giao diện các mục trong danh sách tư vấn viên.
 - ❖ File `MessageAdapter.java`: Hiển thị giao diện các mục trong danh sách tin nhắn.
 - ❖ File `NewsAdapter.java`: Hiển thị giao diện các mục trong danh sách tin tức.
 - ❖ File `TourAdapter.java`: Hiển thị giao diện các mục trong danh sách tour.
 - ❖ File `TourMemberAdapter.java`: Hiển thị giao diện các mục trong danh sách thành viên trong tour.
 - ❖ File `TourTimesheetAdapter.java`: Hiển thị giao diện các mục trong danh sách chặng trong tour.
 - Thư mục `model`, chứa các lớp dữ liệu cho các lớp Adapter và các lớp khởi tạo các tab, bao gồm:
 - ❖ File `ClassroomManager.java`: Chứa các thông tin về người quản lý trong 1 chặng.
 - ❖ File `Conversation.java`: Chứa các thông tin về đoạn hội thoại.
 - ❖ File `Counselor.java`: Chứa các thông tin về tư vấn viên.
 - ❖ File `HomeContactTab.java`: Khởi tạo tab liên hệ trong giao diện trang chủ.
 - ❖ File `HomeConversationsTab.java`: Khởi tạo tab danh sách hội thoại trong giao diện trang chủ.
 - ❖ File `HomeNewsTab.java`: Khởi tạo tab danh sách tin tức trong giao diện trang chủ.
 - ❖ File `HomeTourTab.java`: Khởi tạo tab danh sách tour trong giao diện trang chủ.
 - ❖ File `HomeUserTab.java`: Khởi tạo tab thông tin cá nhân trong giao diện trang chủ.

- ❖ File `MemberMessage.java`: Chứa các thông tin về tin nhắn.
- ❖ File `News.java`: Chứa các thông tin về tin tức.
- ❖ File `Tour.java`: Chứa các thông tin về tour.
- ❖ File `TourInfoTab.java`: Khởi tạo tab thông tin tour trong giao diện 1 tour.
- ❖ File `TourMapImageTab.java`: Khởi tạo tab hình ảnh minh hoạ tour trong giao diện 1 tour.
- ❖ File `TourMapTab.java`: Khởi tạo tab bản đồ tour trong giao diện 1 tour.
- ❖ File `TourMember.java`: Chứa các thông tin về thành viên trong tour.
- ❖ File `TourTimesheet.java`: Chứa các thông tin về chặng trong tour.
- File `BaseActivity.java`: Lớp cha của tất cả các lớp giao diện.
- File `HomeTabsActivity.java`: Quản lý giao diện trang chủ và tất cả những tab trong đó.
- File `LoginActivity.java`: Quản lý giao diện đăng nhập.
- File `MessageActivity.java`: Quản lý giao diện nhắn tin.
- File `SignUpActivity.java`: Quản lý giao diện đăng ký.
- File `TimesheetInfoActivity.java`: Quản lý giao diện thông tin chặng.
- File `TourTabsActivity.java`: Quản lý giao diện 1 tour và tất cả những tab trong đó.
- Thư mục `Manager`, chứa các lớp quản lý các tiến trình xử lý:
 - File `ActivityManager.java`: Quản lý tất cả các lớp giao diện.
 - File `ConversationManager.java`: Quản lý việc gửi và nhận tin nhắn, nhận danh sách hội thoại và danh sách tư vấn viên từ server.
 - File `HomeManager.java`: Quản lý giao diện trang chủ, xử lý danh sách tour nhận được từ server.
 - File `JsonManager.java`: Chuyển đổi đối tượng chuỗi sang JSON.
 - File `LocationManager.java`: Xử lý việc thay đổi vị trí của thành viên trong tour.
 - File `LoginManager.java`: Xử lý kết quả đăng nhập từ server.
 - File `LogoutManager.java`: Xử lý kết quả đăng xuất từ server.

- File `NewsManager.java`: Xử lý danh sách tin tức từ server.
- File `OnlineManager.java`: Quản lý giao kết nối đến server, gửi và nhận message giữa client và server.
- File `SignUpManager.java`: Xử lý kết quả đăng ký từ server.
- File `TourInfoManager.java`: Sắp xếp danh sách thành viên theo thứ tự vai trò trong tour.
- 1 thư mục giao diện tại đường dẫn
`Source\BKOD-AndroidClient\app\src\main\res\layout`, bao gồm:
 - File `activity_home_tabs.xml`: Giao diện trang chủ.
 - File `activity_login.xml`: Giao diện đăng nhập.
 - File `activity_message.xml`: Giao diện nhắn tin.
 - File `activity_sign_up.xml`: Giao diện đăng ký.
 - File `activity_timesheet_info.xml`: Giao diện thông tin chặng trong 1 tour.
 - File `activity_tour_tabs.xml`: Giao diện 1 tour.
 - File `conversation_item_layout.xml`: Giao diện 1 mục trong danh sách hội thoại.
 - File `counselor_item_layout.xml`: Giao diện 1 mục trong danh sách tư vấn viên.
 - File `full_custom_action_bar_layout.xml`: Giao diện thanh tiêu đề.
 - File `map_number_marker_layout.xml`: Giao diện marker.
 - File `message_item_layout.xml`: Giao diện 1 mục trong danh sách tin nhắn.
 - File `news_item_layout.xml`: Giao diện 1 mục trong danh sách tin tức.
 - File `tab_home_contact.xml`: Giao diện tab liên hệ trong giao diện trang chủ.
 - File `tab_home_conversations.xml`: Giao diện tab danh sách hội thoại trong giao diện trang chủ.
 - File `tab_home_news.xml`: Giao diện tab tin tức trong giao diện trang chủ.
 - File `tab_home_tour.xml`: Giao diện tab danh sách tour trong giao diện trang chủ.
 - File `tab_home_user.xml`: Giao diện tab thông tin cá nhân trong giao diện trang chủ.
 - File `tab_tour_info.xml`: Giao diện tab thông tin tour trong giao diện 1 tour.

- File `tab_tour_map.xml`: Giao diện tab bản đồ tour trong giao diện 1 tour.
- File `tab_tour_map_image.xml`: Giao diện tab hình ảnh minh họa tour trong giao diện 1 tour.
- File `tour_item_layout.xml`: Giao diện 1 mục trong danh sách tour.
- File `tour_member_item_layout.xml`: Giao diện 1 mục trong danh sách thành viên của tour (xỏ xuống).
- File `tour_timesheet_item_layout.xml`: Giao diện 1 mục trong danh sách chặng của tour (xỏ xuống).

3.1.3. Server config

- 1 thư mục `ServerConfig`, bao gồm:
 - + File mã nguồn `index.js`: Dùng để chạy dịch vụ cung cấp địa chỉ server cho client kết nối đến.
 - + File `index.json`: Dùng để thay thế file mã nguồn để client lấy trực tiếp địa chỉ server.
 - + File `push.cmd`: Mô tả cách chạy file mã nguồn `index.js` trên `Heroku`.
 - + Chi tiết cách đẩy xem mục Cài đặt ứng dụng phía client.

3.1.4. Server

- 1 ứng dụng Node.js, bao gồm:
 - + 11 file mã nguồn .js, bao gồm:
 - Thư mục `config`, chứa file `serverConfig.js`: dùng để cấu hình cổng cho server và địa chỉ cơ sở dữ liệu. Chi tiết cách cấu hình xem mục Cài đặt ứng dụng phía server.
 - File `commonFunction.js`: Chứa các hàm dùng chung như in ra thông báo trên màn hình.
 - File `conversation.js`: Quản lý việc truy vấn danh sách hội thoại trong cơ sở dữ liệu và gửi về client.
 - File `counselor.js`: Quản lý việc truy vấn danh sách tư vấn viên trong cơ sở dữ liệu và gửi về client.
 - File `login.js`: Quản lý việc truy vấn thông tin đăng nhập trong cơ sở dữ liệu và gửi kết quả đăng nhập về client.
 - File `logout.js`: Quản lý việc đăng xuất cho client và gửi về client.
 - File `memberLocation.js`: Quản lý việc cập nhật vị trí của thành viên trong cơ sở dữ liệu và gửi về cho các thành viên khác.

- File `news.js`: Quản lý việc truy vấn danh sách tin tức trong cơ sở dữ liệu và gửi về client.
 - File `server.js`: Quản lý các kết nối tới client.
 - File `signUp.js`: Quản lý việc thêm 1 user mới trong cơ sở dữ liệu khi client đăng ký và gửi kết quả đăng ký về client.
 - File `tour.js`: Quản lý việc truy vấn danh sách tour trong cơ sở dữ liệu và gửi về client.
- + Thư mục `node_modules`: Các module cần thiết, có thể cài đặt thủ công. Xem mục Cài đặt ứng dụng phía server.

3.2. Các tính năng

3.2.1. Đăng nhập

- Người dùng sử dụng 1 thiết bị chạy hệ điều hành Android mở ứng dụng lên, nhập tên tài khoản và mật khẩu vào, nhấn vào nút đăng nhập.
- Sau khi đăng nhập thành công, người dùng có thể sử dụng các tính năng dưới đây.

3.2.2. Xem danh sách tour

- Ở tab đầu tiên trong giao diện trang chủ, người dùng có thể xem danh sách các tour đã tham gia, đang tham gia và sẽ tham gia.

3.2.3. Xem danh sách tin tức

- Ở tab thứ 2 trong giao diện trang chủ, người dùng có thể xem danh sách các tin tức mà nhà trường đã đăng.

3.2.4. Xem thông tin cá nhân

- Ở tab thứ 3 trong giao diện trang chủ, người dùng có thể xem thông tin cá nhân của mình.

3.2.5. Xem danh sách tin nhắn

- Ở tab thứ 4 trong giao diện trang chủ, người dùng có thể xem danh sách các tin nhắn mà mình đã gửi hoặc đã nhận.

3.2.6. Xem danh sách tư vấn viên

- Ở tab thứ 5 trong giao diện trang chủ, người dùng có thể xem danh sách các tư vấn viên.

3.2.7. Xem bản đồ tour

- Ở tab thứ nhất trong giao diện 1 tour, người dùng có thể xem bản đồ của tour với các marker, route chỉ đường.

3.2.8. Xem thông tin tour

- Ở tab thứ 2 trong giao diện 1 tour, người dùng có thể xem các thông tin của tour.

3.2.9. Xem hình ảnh minh họa tour

- Ở tab thứ 3 trong giao diện 1 tour, người dùng có thể xem hình ảnh minh họa của tour, hoặc bản đồ tổng thể.

3.2.10. Xem thông tin của chặng

- Khi nhấn vào 1 chặng trong 1 tour, người dùng có thể xem các thông tin của chặng đó.

3.2.11. Nhắn tin với các thành viên trong tour

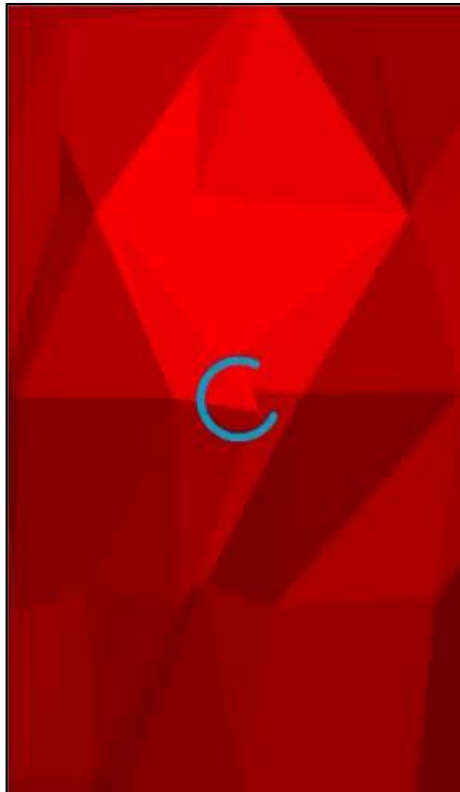
- Khi nhấn vào 1 thành viên trong 1 tour, người dùng có thể nhắn tin với thành viên đó.

3.2.12. Nhắn tin với các tư vấn viên

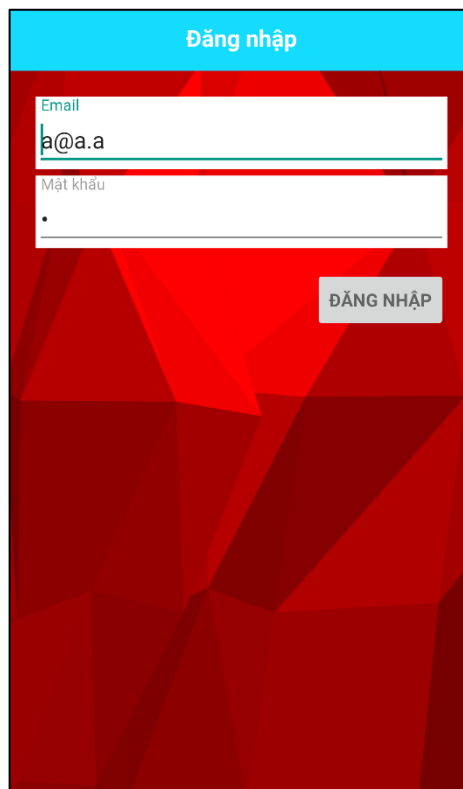
- Khi nhấn vào 1 tư vấn viên trong danh sách tư vấn viên, người dùng có thể nhắn tin với tư vấn viên đó.

3.3. Các giao diện cụ thể

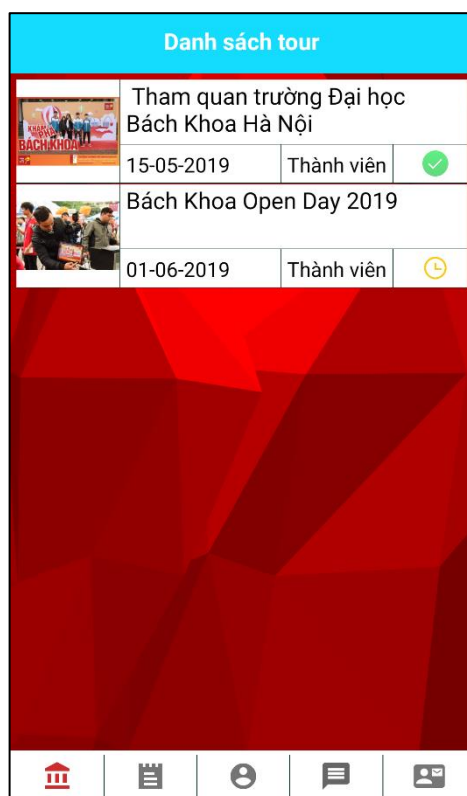
3.3.1. Giao diện progress bar



3.3.2. Giao diện đăng nhập

The image displays a login form on a red background with a low-poly pattern. At the top, there is a blue header bar with the text 'Đăng nhập' (Login) in white. Below the header, there are two white input fields. The first field is labeled 'Email' and contains the text 'a@a.a'. The second field is labeled 'Mật khẩu' (Password) and contains a single dot. To the right of the password field, there is a grey button with the text 'ĐĂNG NHẬP' (Login) in white.

3.3.3. Giao diện tab danh sách tour








3.3.4. Giao diện tab danh sách tin tức



3.3.5. Giao diện tab thông tin người dùng

Thông tin cá nhân	
Email	a@a.a
Họ tên đầy đủ	Nguyễn Tử Quảng
Giới tính	Nam
Trường	Trường THPT Phan Huy Chú
Lớp	12A4
Số điện thoại	0367456534

ĐĂNG XUẤT

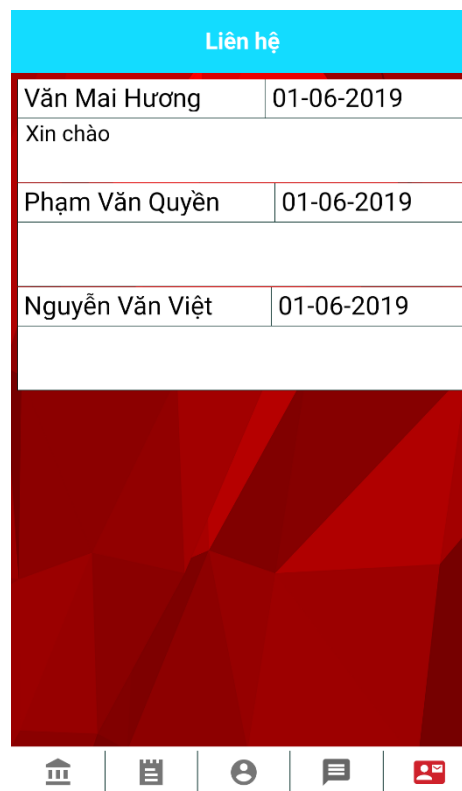


3.3.6. Giao diện tab danh sách hội thoại

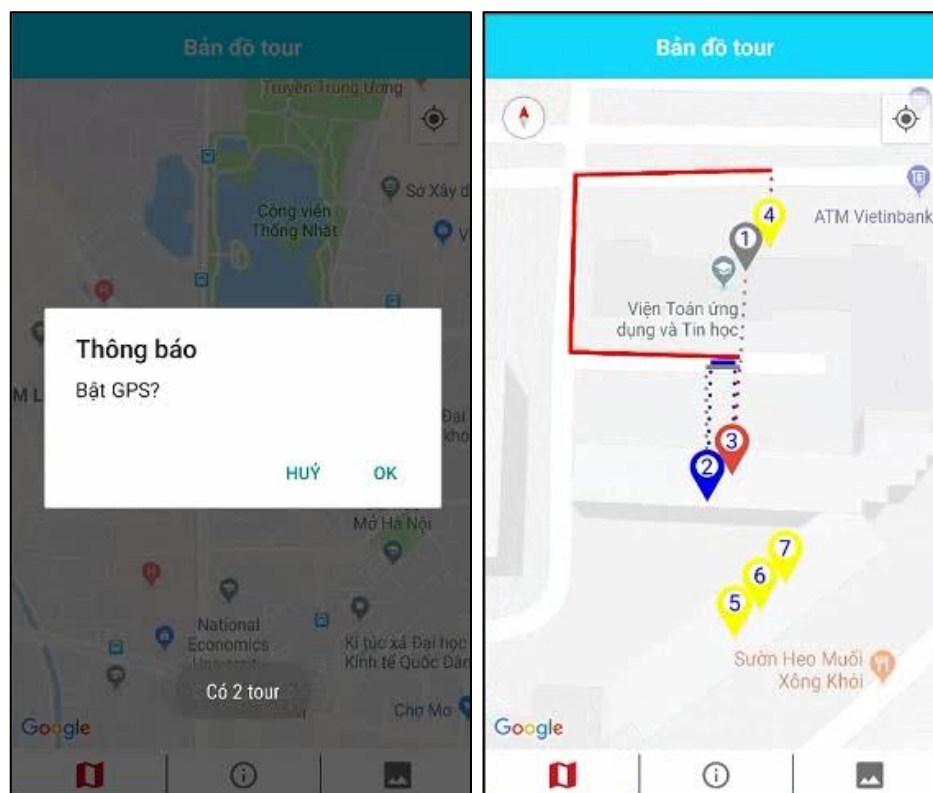
Danh sách hội thoại	
Văn Mai Hương	01-06-2019
Xin chào	
Lê Mạnh Hùng	01-06-2019
Chào bạn	
Nguyễn Thị Thanh	01-06-2019
Mình học lớp 12A4	
Phạm Văn Quyền	01-06-2019
Nguyễn Văn Việt	01-06-2019



3.3.7. Giao diện tab liên hệ




3.3.8. Giao diện tab bản đồ tour




3.3.9. Giao diện tab thông tin tour

Thông tin tour



Tên của tour	Tham quan trường Đại học Bách Khoa Hà Nội
Trạng thái tour	Đang hoạt động ✔
Vai trò trong tour	Thành viên
Ngày diễn ra tour	15-05-2019
Danh sách thành viên	


Thông tin tour



Trưởng đoàn	Phan Văn Tài Em
Phó đoàn	Nguyễn Thị Linh
Thành viên	Lê Công Vinh
Thành viên	Lê Than Nghị
Thành viên	Nguyễn Văn An
Thành viên	Trần Duy Hưng
Thành viên	Nguyễn Thị Thanh
Thành viên	Nguyễn Tử Quảng
Thành viên	Lê Mạnh Hùng
Phụ huynh	Nguyễn Xuân Bắc


Danh sách chặng
 10:00:00 - 11:00:00 D3-202 Chưa bật GPS

Thông tin tour



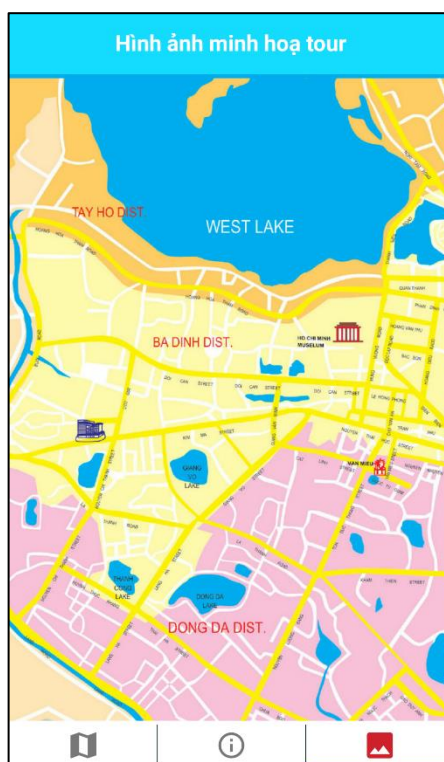
10:00:00 - 11:00:00	D3-202	Chưa bật GPS
11:00:00 - 12:00:00	D5-202	Chưa bật GPS
12:00:00 - 13:00:00	D5-203	Chưa bật GPS
14:00:00 - 15:00:00	D3-204	Chưa bật GPS
15:00:00 - 16:00:00	D7-104	Chưa bật GPS
16:00:00 - 17:00:00	D7-201	Chưa bật GPS
17:00:00 - 18:00:00	D7-202	Chưa bật GPS

Thông tin tour



	Nội	
Trạng thái tour	Đang hoạt động ✔	
Vai trò trong tour	Thành viên	
10:00:00 - 11:00:00	D3-202	6873m
11:00:00 - 12:00:00	D5-202	6886m
12:00:00 - 13:00:00	D5-203	6889m
14:00:00 - 15:00:00	D3-204	6875m
15:00:00 - 16:00:00	D7-104	6903m
16:00:00 - 17:00:00	D7-201	6906m
17:00:00 - 18:00:00	D7-202	6908m

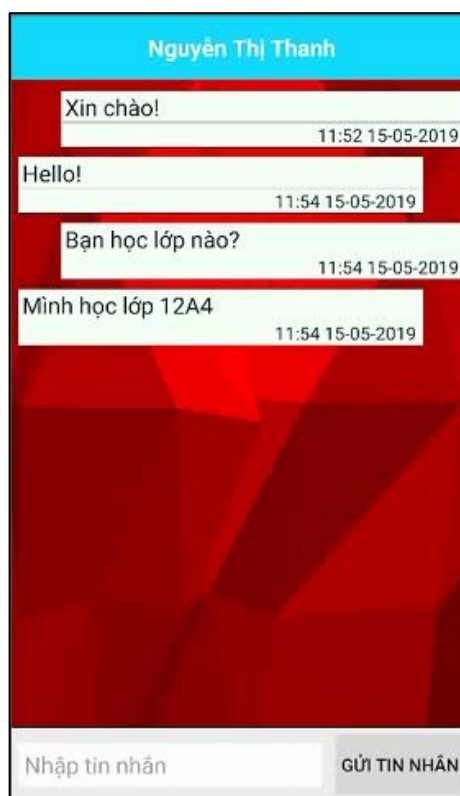
3.3.10. Giao diện tab hình ảnh minh họa tour



3.3.11. Giao diện thông tin chặng

Thông tin chặng	
Thời gian	10:00:00 - 11:00:00
Tên toà nhà	D3
Tên phòng học	202
Tầng	2
Tên người quản lý	Trần Thị Thanh Thuỷ
Email	thuyttt234685@gmail.com
Số điện thoại	thuyttt234685@gmail.com

3.3.12. Giao diện nhắn tin



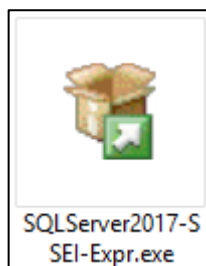
3.4. Hướng dẫn cài đặt cụ thể

3.4.1. Cài đặt hệ quản trị cơ sở dữ liệu Microsoft Sql Server

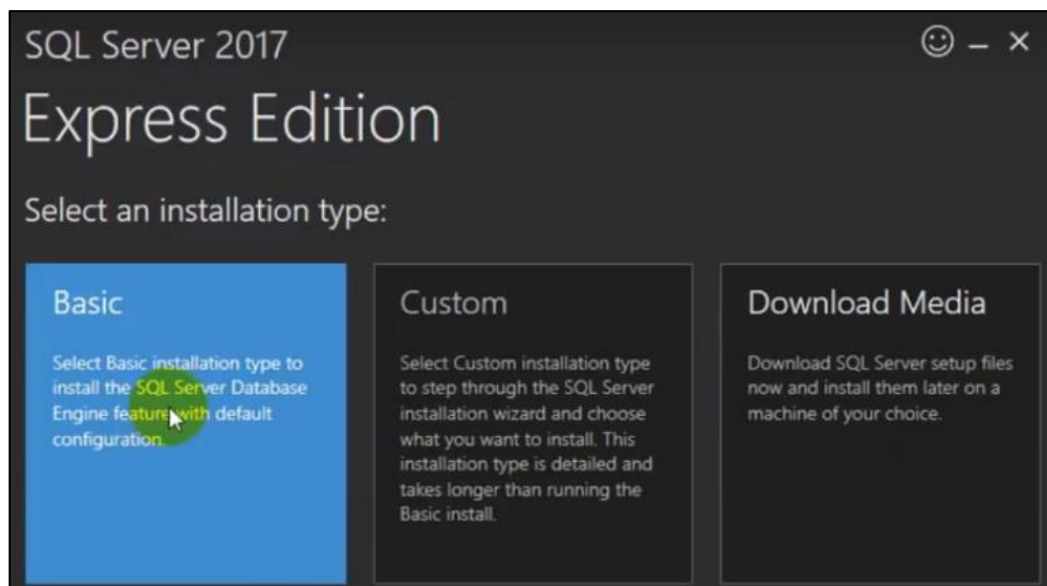
- Thiết bị cài đặt Sql Server cần chạy hệ điều hành Windows hoặc Windows Server. Nếu đã cài đặt rồi thì có thể bỏ qua bước này.
- Truy cập đường dẫn <https://www.microsoft.com/en-us/sql-server/sql-server-editions-express> để tải về file cài đặt.
- Trong trường hợp đường dẫn không tồn tại thì tìm kiếm từ khoá: `Microsoft Sql Server Express Edition`, nhấn vào kết quả đầu tiên.
- Nhấn vào nút `Download now` để tải phần mềm về và tiến hành cài đặt:



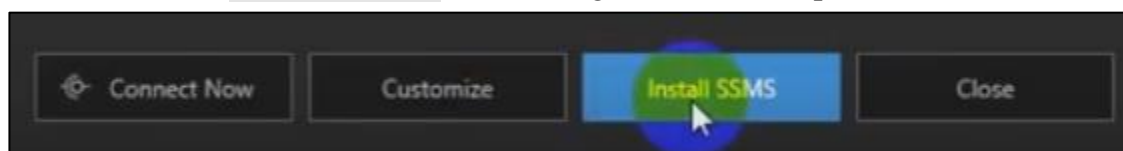
- Nhấn vào file .exe vừa tải về để cài đặt:



- Chọn mục Basic:



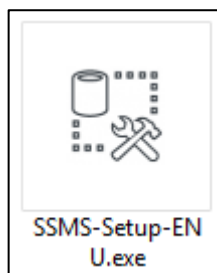
- Nhấn Next, Accept, Install và đợi 1 thời gian để quá trình cài đặt kết thúc.
- Nhấn vào Install SSMS để cài đặt giao diện cho Sql Server:



- Có thể nhấn Close sau khi nhấn vào Install SSMS.
- Tải về file cài đặt SSMS:



- Nhấn vào file .exe vừa tải về để cài đặt:



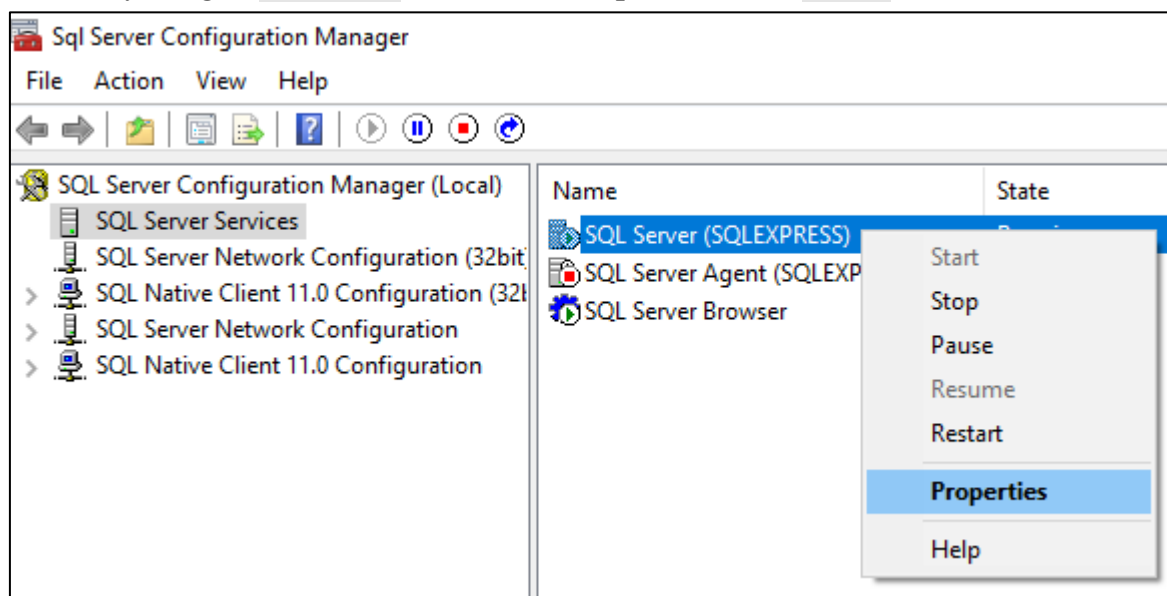
- Nhấn Install và đợi 1 thời gian để quá trình cài đặt kết thúc.

3.4.2. Cấu hình cổng cho Microsoft Sql Server

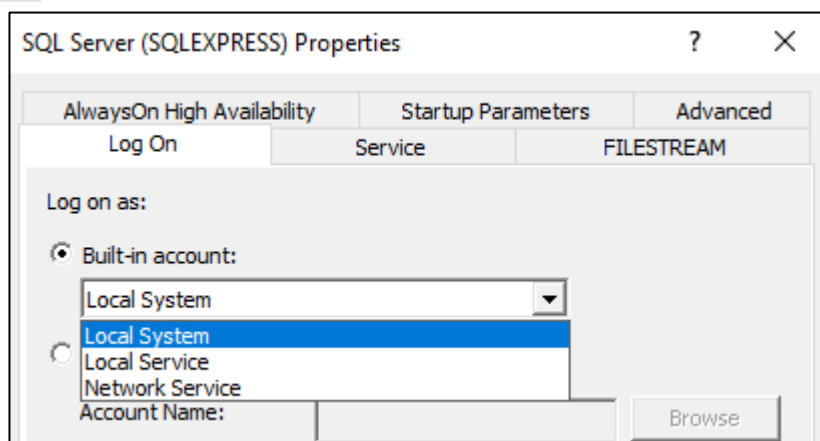
- Nhấn vào biểu tượng Start trên Windows, tìm kiếm từ khoá **SQL Server Configuration Manager**, nhấn vào kết quả tìm được:



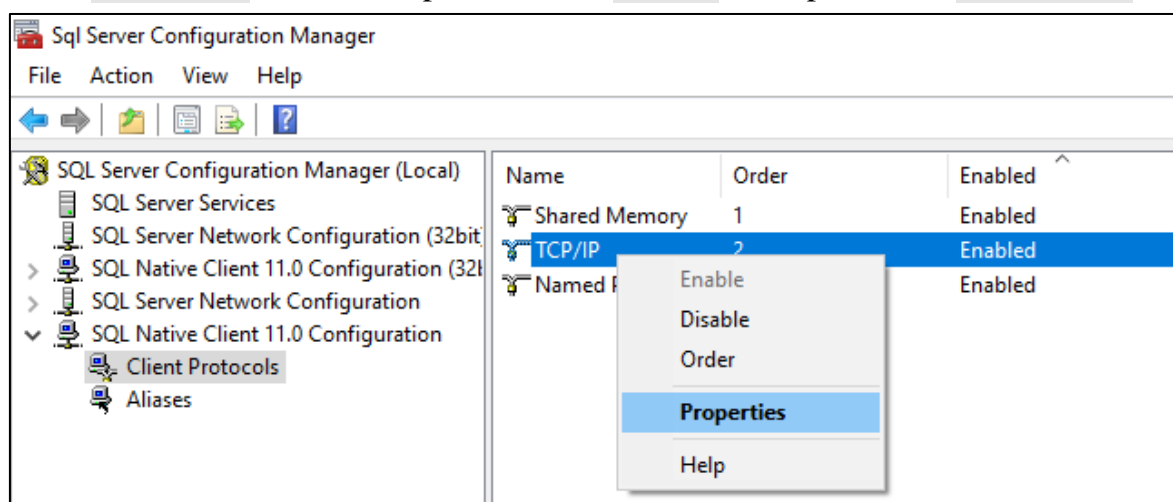
- Ở tab bên trái chọn **SQL Server Services**, nhấn chuột phải vào mục **SQL Server (SQLEXPRESS)** ở bên phải, chọn **Properties** (Nếu State của mục này đang là **Disabled** thì nhấn chuột phải và chọn **Start**):



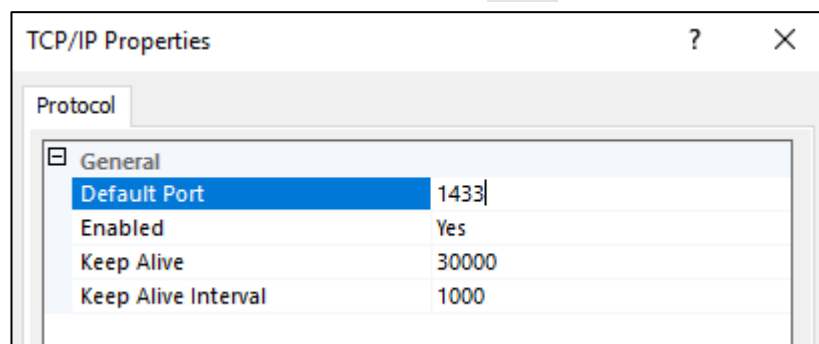
- Ở cửa sổ **Properties**, tab **Log On**, chọn **Built-in account** là **Local System**:



- Nhấn **Ok** để đóng tab.
- Chuyển qua tab **Sql Native Client 11.0 Configuration** → **Client Protocols**, nhấn chuột phải vào mục **TCP/IP** ở bên phải, chọn **Properties**:



- Chọn cổng cho Sql Server, mặc định là **1433**:



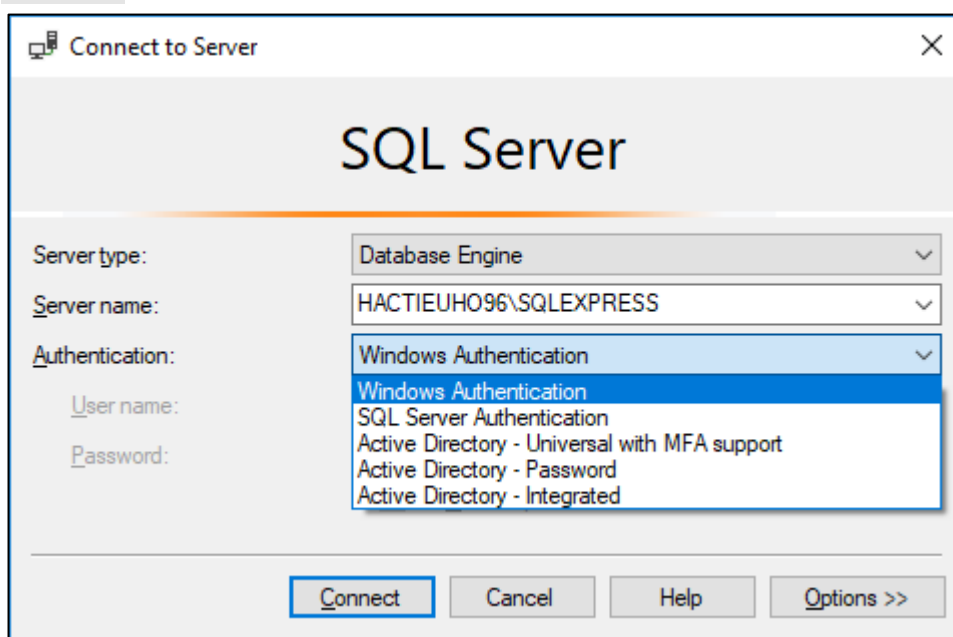
- Nhấn **Ok** để đóng cửa sổ, sau bước này có thể tắt luôn cửa sổ **SQL Server Configuration Manager**.

3.4.3. Khởi tạo cơ sở dữ liệu

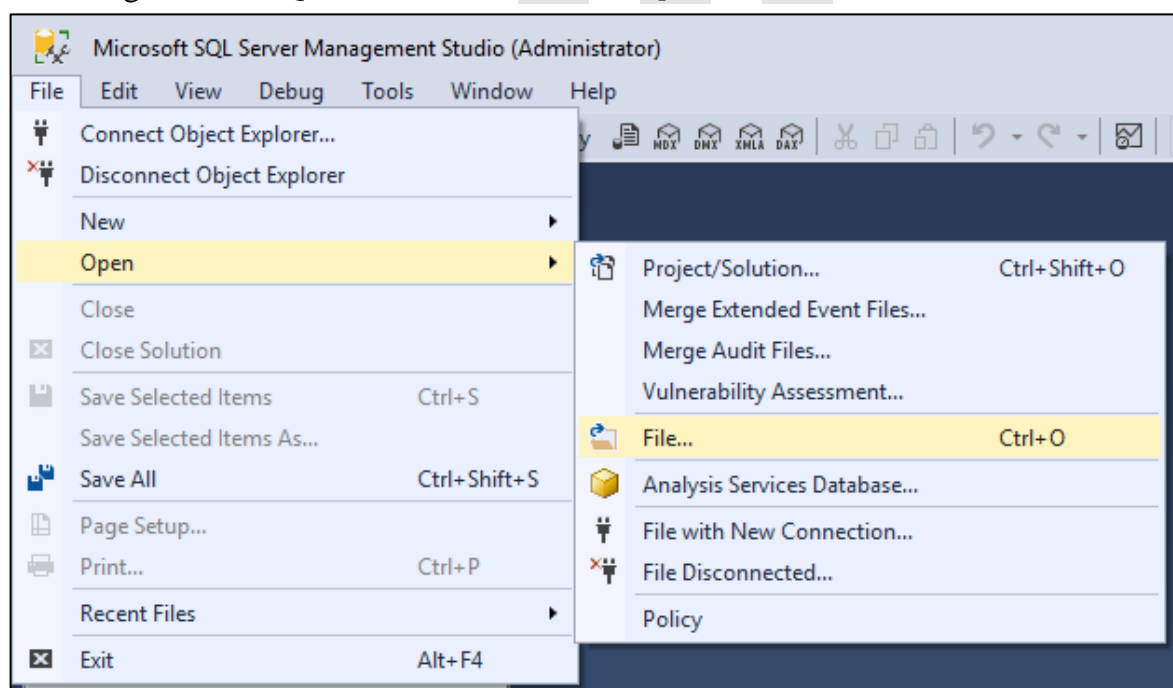
- Nhấn vào biểu tượng Start trên Windows, tìm kiếm từ khoá `SQL Server Management Studio`, nhấn vào kết quả tìm được:



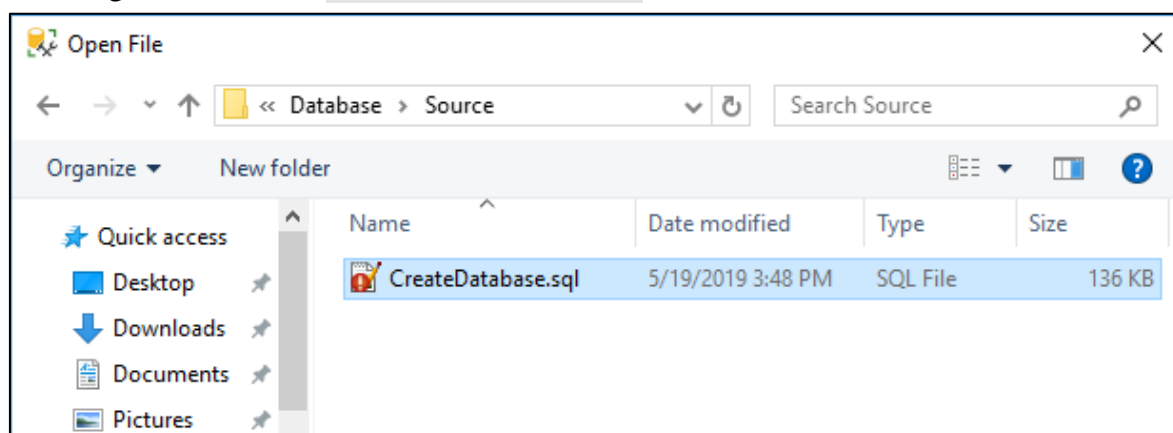
- Ở mục `Authentication` chọn `Windows Authentication`, sau đó nhấn `Connect`:



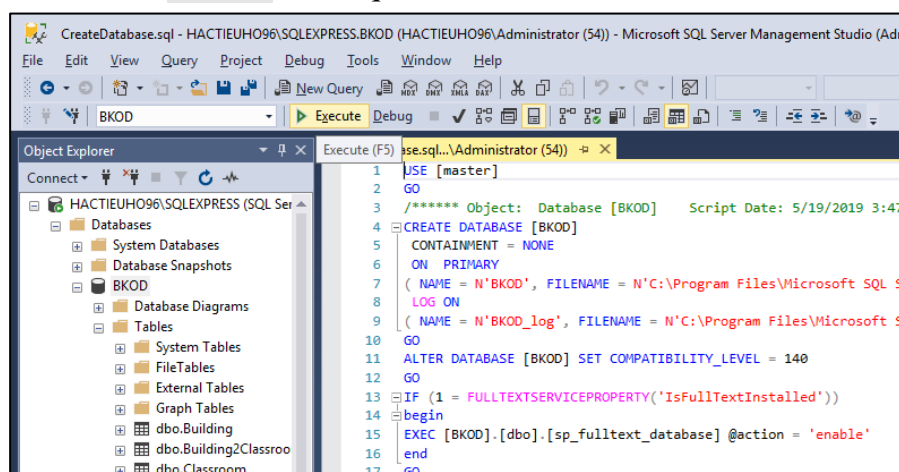
- Ở giao diện SQL Server, chọn **File** → **Open** → **File**:



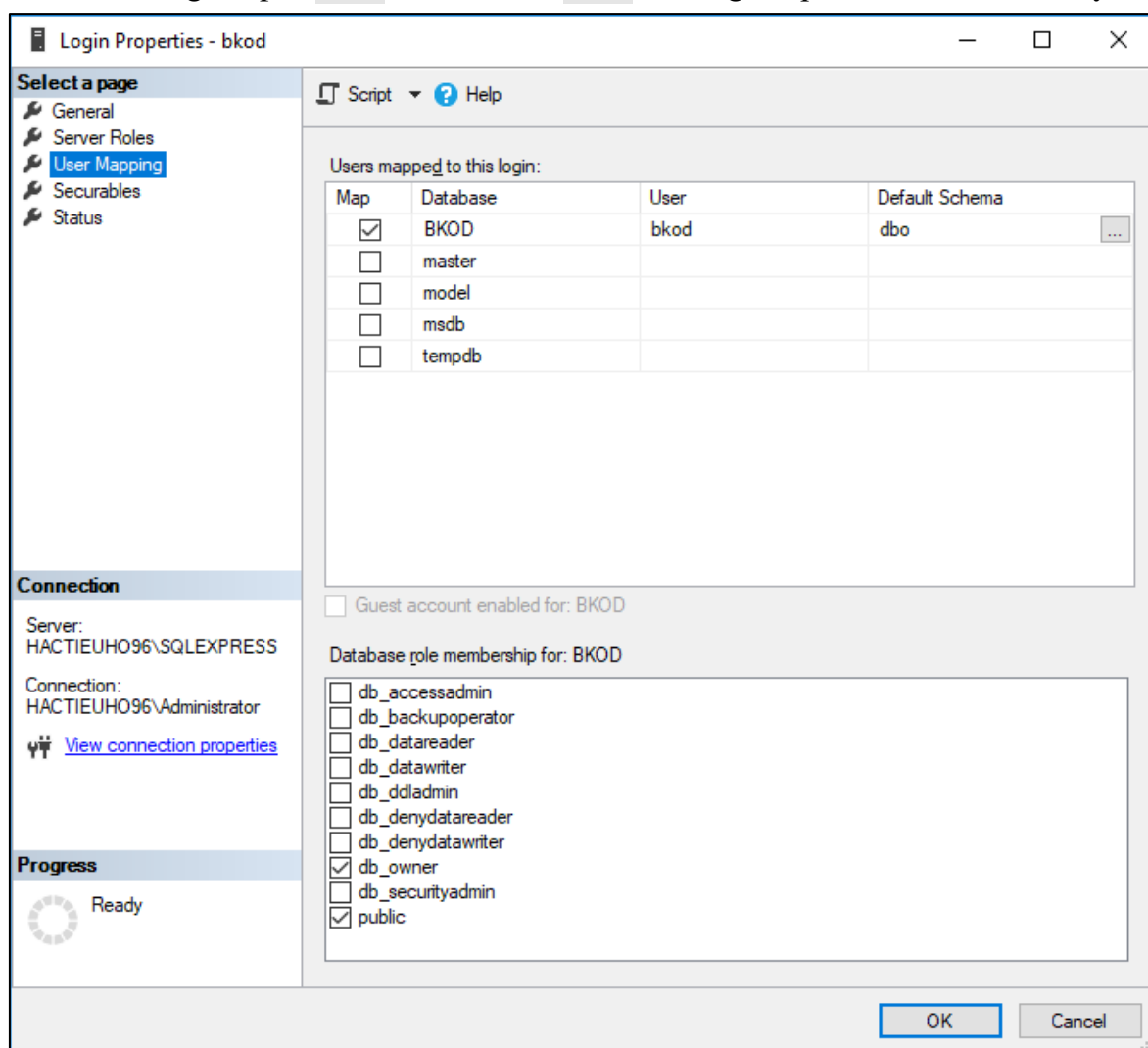
- Chuyển đến đường dẫn **Source/Database/Source** trong thư mục mã nguồn, chọn file **CreateDatabase.sql**:



- Nhấn vào nút **Execute** trên Sql Server:



- Chương trình sẽ chạy đoạn mã và sinh ra 1 cơ sở dữ liệu tên là BKOD nếu không bị trùng tên với cơ sở dữ liệu nào đã có trước đó và sinh ra 1 user với tên đăng nhập là `bkod`, mật khẩu là `bkod` để đăng nhập vào cơ sở dữ liệu này:

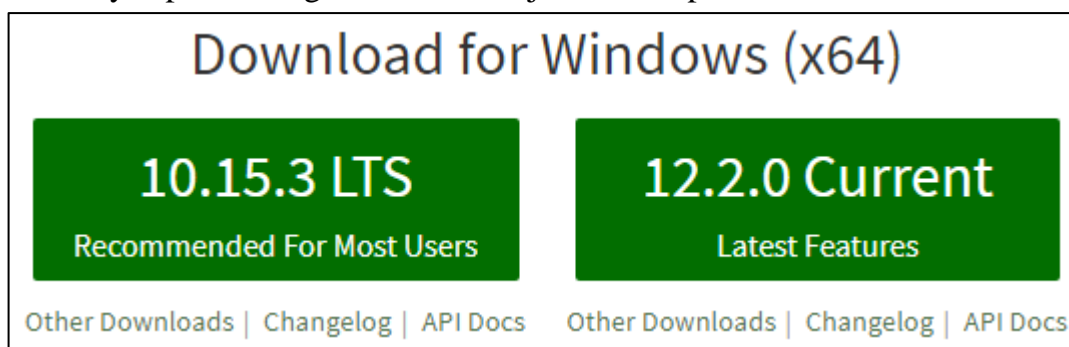


- Nếu Sql Server thông báo lỗi ở bước tạo user bkod, có thể tạo thủ công.

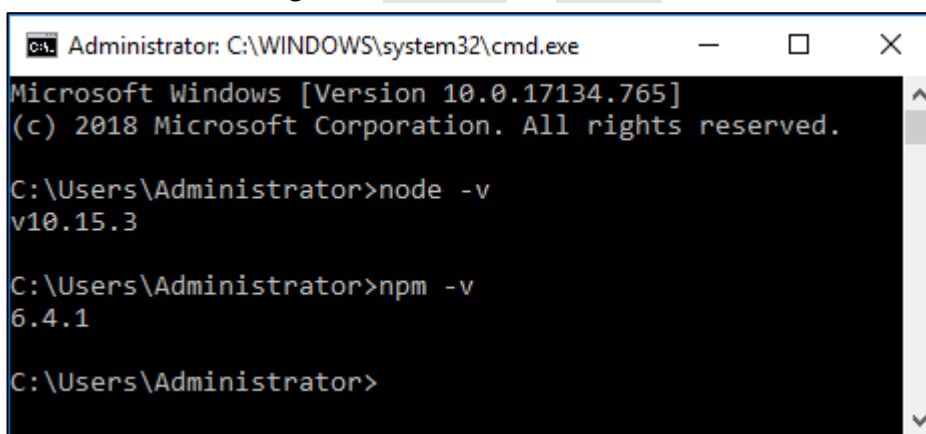
3.4.4. Cài đặt Node.js

- Thiết bị cần chạy các hệ điều hành hỗ trợ cài đặt Node.js (Windows, Windows Server, Linux,...). Nếu đã cài đặt rồi thì có thể bỏ qua bước này.

- Truy cập vào trang chủ của Node.js và tải về phiên bản LTS mới nhất:



- Tiến hành cài đặt như những phần mềm bình thường (Next → Accept → Next → Next → Next → Install → Finish).
- Kiểm tra phiên bản Node.js trên Command Prompt (Windows) hoặc Terminal (Linux) bằng lệnh `node -v` và `npm -v`:

The image shows a Windows Command Prompt window titled "Administrator: C:\WINDOWS\system32\cmd.exe". The window displays the following text: "Microsoft Windows [Version 10.0.17134.765] (c) 2018 Microsoft Corporation. All rights reserved. C:\Users\Administrator>node -v v10.15.3 C:\Users\Administrator>npm -v 6.4.1 C:\Users\Administrator>".

3.4.5. Cài đặt ứng dụng phía server

- Copy thư mục `Server` trong thư mục chứa mã nguồn vào bộ nhớ của thiết bị chạy server:

HDD 256 (D:) > 20182 > DoAnTotNghiep > Source > Server >				
Name	Date modified	Type	Size	
.idea	5/17/2019 4:38 PM	File folder		
config	4/18/2019 9:12 AM	File folder		
node_modules	3/12/2019 5:27 PM	File folder		
commonFunction.js	3/10/2019 2:28 PM	JS File	1 KB	
conversation.js	5/6/2019 11:39 AM	JS File	13 KB	
counselor.js	5/13/2019 2:41 PM	JS File	3 KB	
login.js	5/15/2019 11:28 AM	JS File	10 KB	
logout.js	3/28/2019 6:14 PM	JS File	2 KB	
memberLocation.js	4/2/2019 11:40 AM	JS File	4 KB	
news.js	3/21/2019 12:23 PM	JS File	3 KB	
server.js	5/15/2019 10:33 AM	JS File	6 KB	
signUp.js	4/20/2019 5:26 PM	JS File	4 KB	
tour.js	5/6/2019 12:17 PM	JS File	13 KB	

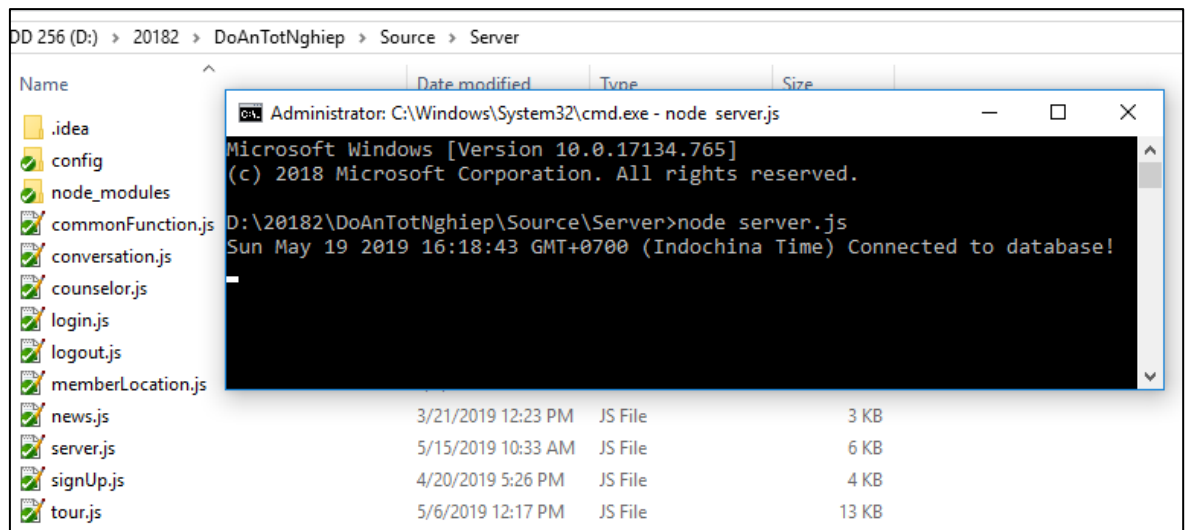
- Mở file `serverConfig.js` trong thư mục `config` để cấu hình server bằng trình chỉnh sửa văn bản:

HDD 256 (D:) > 20182 > DoAnTotNghiep > Source > Server > config				
Name	Date modified	Type	Size	
serverConfig.js	4/18/2019 9:12 AM	JS File	1 KB	

- Nội dung file `serverConfig.js` như sau:

```
D:\20182\DoAnTotNghiep\Source\Server\config\serverConfig.js - No
File Edit Search View Encoding Language Settings Tools
serverConfig.js
1 module.exports = {
2   port: 28146,
3   database: {
4     connectionLimit: 20,
5     user: 'bkod',
6     password: 'bkod',
7     // server: 'system.techlinkvn.com',
8     server: 'localhost',
9     database: 'BKOD',
10    port: 1433
11  }
12 };
```

- + `port: 28146`: Đây là cổng dành cho server trên thiết bị. Cổng này sẽ được mở cho các client kết nối đến. Có thể thay đổi cổng `28146` bằng bất kỳ cổng nào hợp lệ, chưa dùng và không bị tường lửa chặn.
- + `database` → `user: 'bkod'`: Tên đăng nhập vào database đã tạo ở mục Khởi tạo cơ sở dữ liệu.
- + `database` → `password: 'bkod'`: Mật khẩu đăng nhập vào database đã tạo ở mục Khởi tạo cơ sở dữ liệu.
- + `database` → `server: 'system.techlinkvn.com'`: Địa chỉ của nơi lưu cơ sở dữ liệu.
- + `database` → `port: 1433`: Cổng kết nối vào Sql Server đã cấu hình ở mục Cấu hình cổng cho Microsoft Sql Server.
- Sau khi chỉnh sửa xong, lưu file lại.
- Dùng commandline của thiết bị di chuyển vào thư mục `Server` trên bộ nhớ thiết bị chạy server, cài đặt các module cần thiết bằng cách chạy lệnh `npm install mssql express socket.io`:
- Dùng command line của thiết bị di chuyển vào thư mục `Server` trên bộ nhớ thiết bị chạy server, chạy lệnh `node server.js`:



- Khi command line hiện lên thông báo `Connected to database!` tức là server đã chạy bình thường và kết nối thành công tới cơ sở dữ liệu.

3.4.6. Cài đặt ứng dụng phía client

- Cấu hình địa chỉ server để client đọc được có 2 cách:
- Cách 1: Mở file `index.js` trong thư mục `ServerConfig` để cấu hình địa chỉ server mà client đọc được:

```
index.js
1  const express = require('express')
2  const app = express()
3  const port = process.env.PORT || '3000'
4
5  app.get('/serverConfig', (req, res) => res.send({"APP":"BKOD","IP":"http://192.168.15.46","PORT":"28146"}));
6  //app.get('/serverConfig', (req, res) => res.send({"APP":"BKOD","IP":"http://system.techlinkvn.com","PORT":"28146"}));
7  app.get('/serverConfigNDT', (req, res) => res.send({"APP":"BKOD","IP":"http://system.techlinkvn.com","PORT":"28146"}));
```

- + `'/serverConfig'`: Địa chỉ uri của địa chỉ file config.
- + `"IP": "Địa_chỉ_của_ứng_dụng_phía_server"`, xem mục Cài đặt ứng dụng phía server.
- + `"PORT": "Cổng_của_ứng_dụng_phía_server"`, xem mục Cài đặt ứng dụng phía server.
- Đẩy file `index.js` như một ứng dụng Node.js lên một server có hỗ trợ Node.js trên internet. Ở đây em chọn server của Heroku, chi tiết cách đẩy trong file `push.cmd` trong thư mục `ServerConfig`.
- Sau khi đẩy, lấy đường link của địa chỉ server bằng cách nối địa chỉ uri phía sau địa chỉ ứng dụng Node.js vừa đẩy lên Heroku, cụ thể là <https://bkod.herokuapp.com/serverConfig>:

- Cách 2: Đơn giản chỉ cần đẩy file `index.json` lên internet (không cần hỗ trợ Node.js):

- Lấy đường link của file này.
- Kết thúc bước cấu hình địa chỉ server.
- Tải `Android Studio` tại địa chỉ <https://developer.android.com/studio>.
- Đồ án này sử dụng `Android Studio` phiên bản 3.2.0 build 181.5014246. Để hạn chế lỗi xảy ra, nên sử dụng cùng phiên bản. Tải phiên bản này tại địa chỉ: <https://developer.android.com/studio/archive>:

^ Android Studio 3.2.0

September 24, 2018

Installers

Windows IDE only (64-bit): [android-studio-ide-181.5014246-windows.exe](#) (968031392 bytes)

Mac: [android-studio-ide-181.5014246-mac.dmg](#) (1036181525 bytes)

SHA-256 checksums

978673a7babf51a9dca67729213c178995c1039a496dc1dfccb4c095b842c753 android-studio-ide-181.5014246-windows.exe

a3499a64970bf97d95a3bb27ebe571a56cee77510fa8a6d4745d6fbc24d252e1 android-studio-ide-181.5014246-mac.dmg

Zip files

Windows (64-bit): [android-studio-ide-181.5014246-windows.zip](#) (1048288041 bytes)

Windows (32-bit): [android-studio-ide-181.5014246-windows32.zip](#) (1047760189 bytes)

Mac: [android-studio-ide-181.5014246-mac.zip](#) (1039641440 bytes)

Linux: [android-studio-ide-181.5014246-linux.zip](#) (1054392038 bytes)

SHA-256 checksums

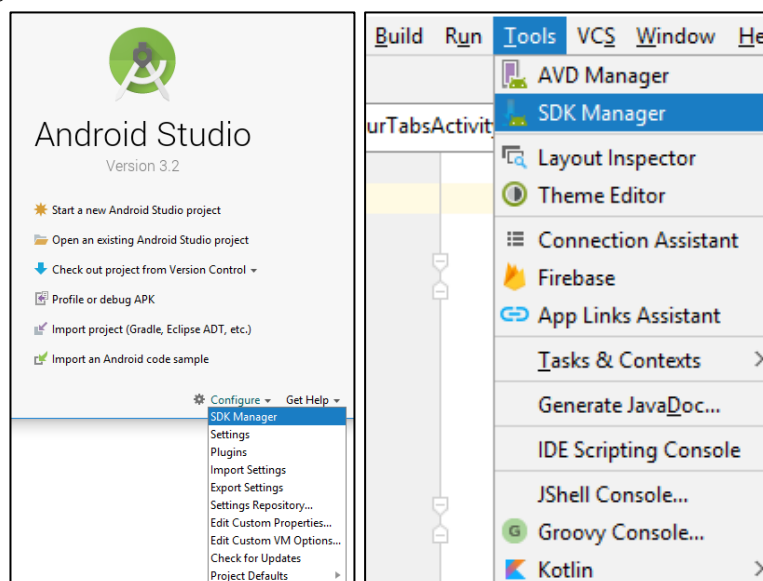
0a301ae52a41f0a55d10e78f8390b931abd6eb31af932a520738438d0d6bcfe9 android-studio-ide-181.5014246-windows.zip

e28a862663ae90cb9e329ce69cf431306b7933af10ed06e428a8c31900ed2ef0 android-studio-ide-181.5014246-windows32.zip

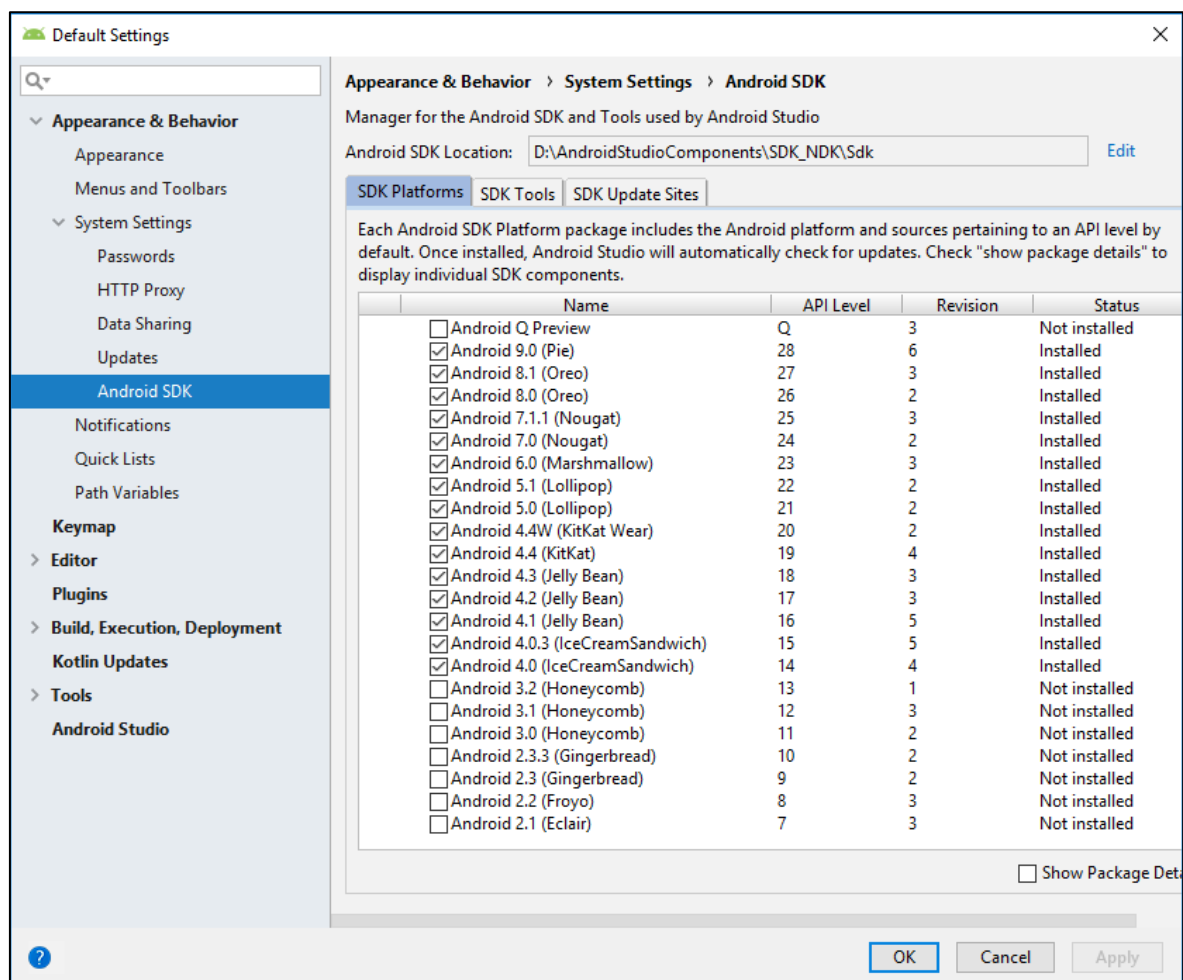
055f0665f345ba0d426fb5aae02f142e85e02f6ec5a8bba32c072928f85d0cac android-studio-ide-181.5014246-mac.zip

e671d48cad66589860c510871167309b88c3f1f5e22a691cba053764c11a2a6c android-studio-ide-181.5014246-linux.zip

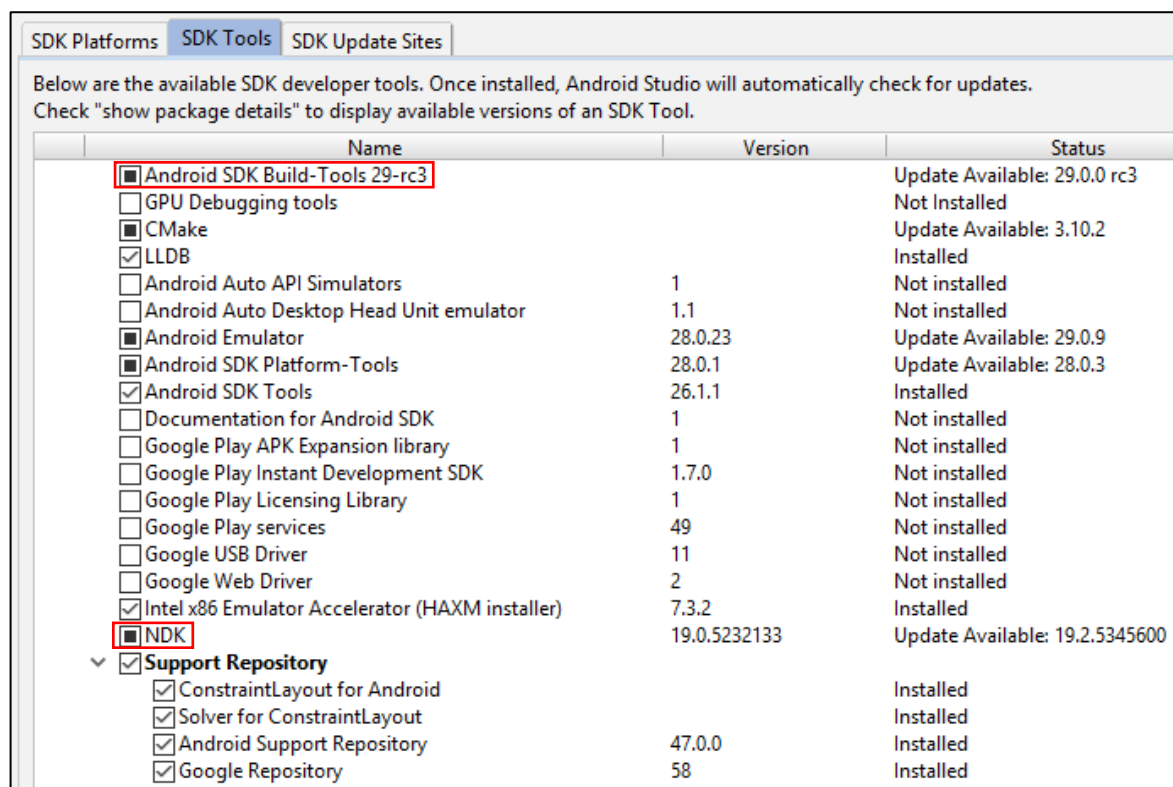
- Tiến hành giải nén/cài đặt Android Studio như bình thường.
- Sau khi cài đặt, mở Android Studio lên. Nhấn vào **Configure** → **SDK Manager** nếu chưa mở project nào, hoặc **Tools** → **SDK Manager** nếu đã mở 1 project.



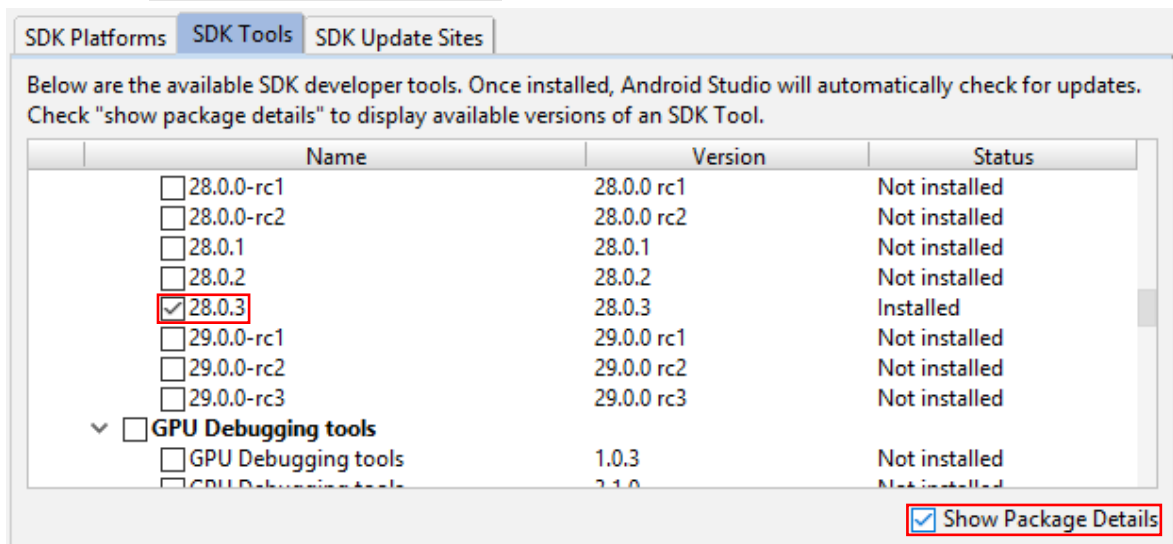
- Tải về các bản **SDK Platform** mà mình muốn ứng dụng chạy trên đó, hoặc có thể tải tất cả những **SDK Platform** thịnh hành:



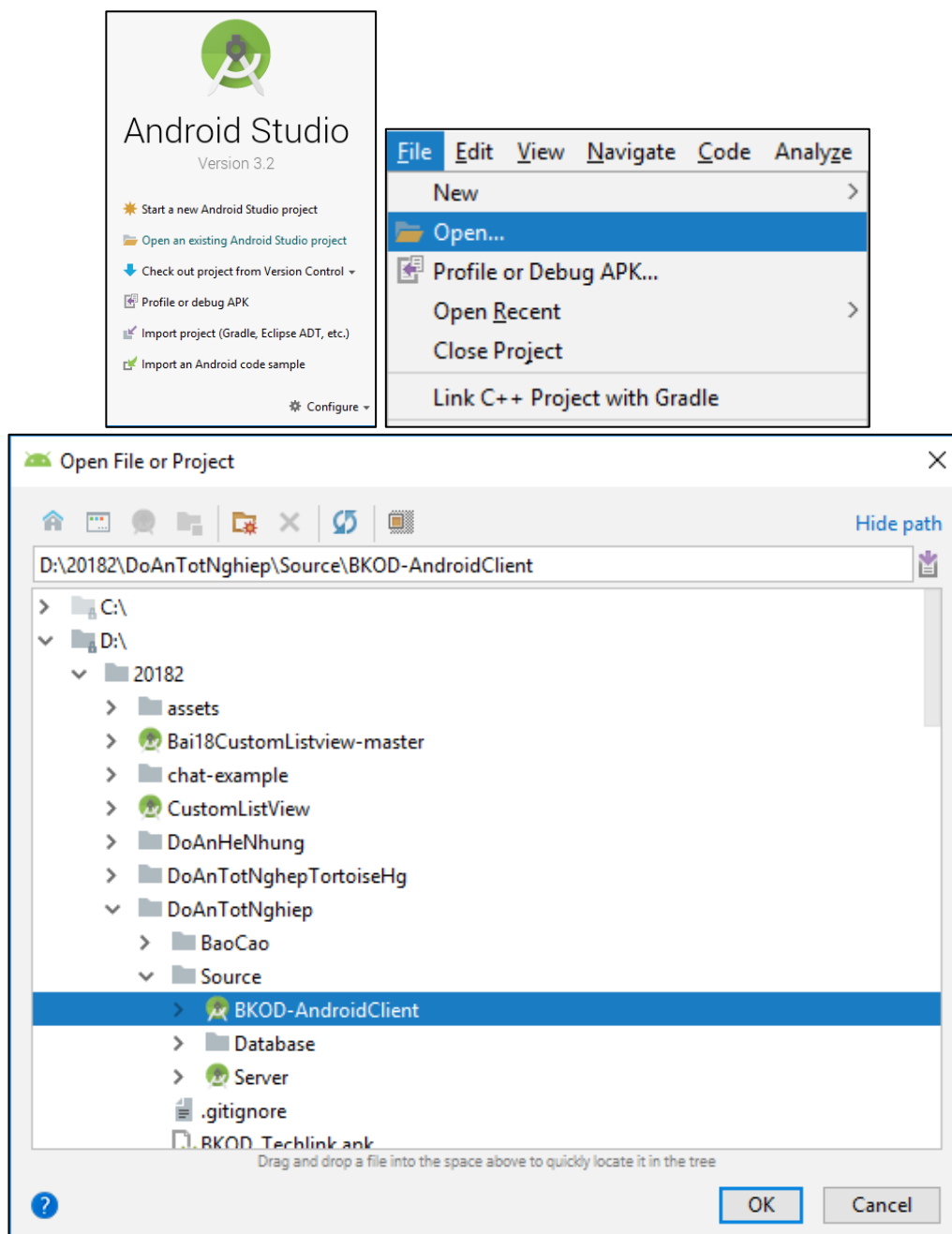
- Chuyển qua tab **SDK Tools**, cài đặt **Android SDK Build-Tools** và **NDK**, nhấn **OK** và chương trình sẽ tự động cài đặt phiên bản mới nhất:



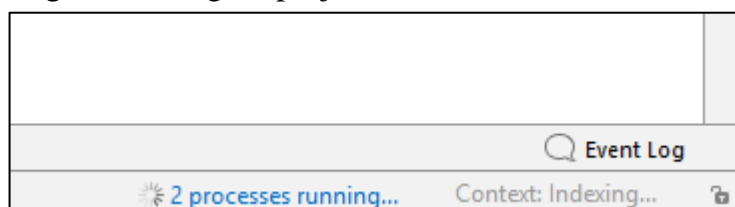
- Kiểm tra phiên bản Android SDK Build-Tools hiện tại bằng cách nhấn vào Show Package Details, ghi nhớ số hiệu phiên bản này:



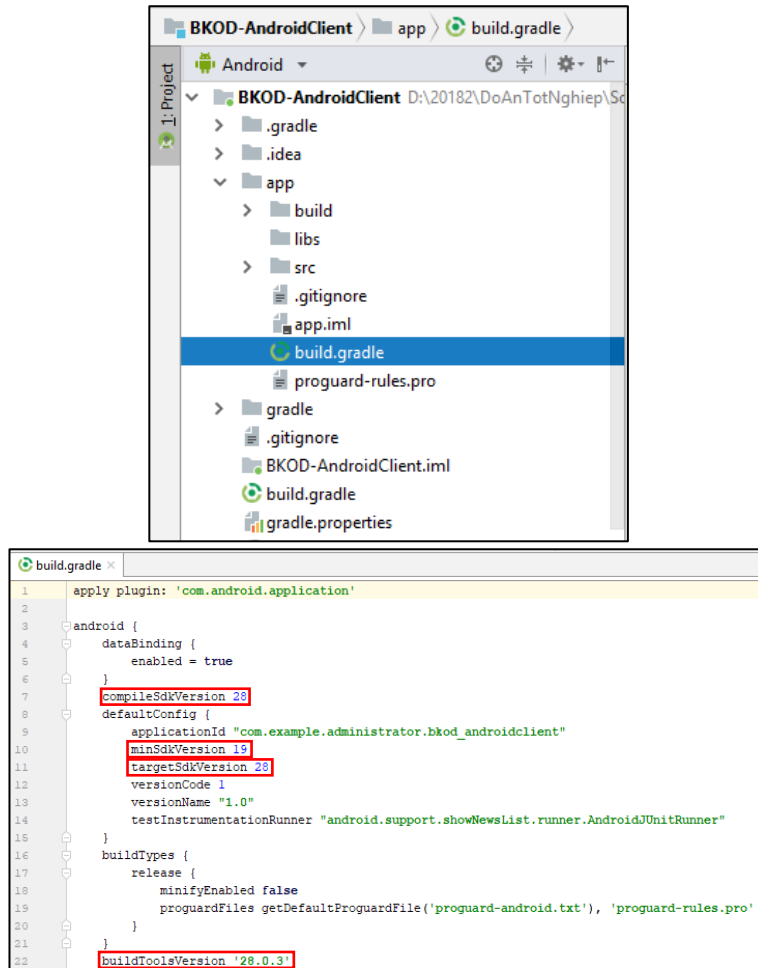
- Mở thư mục BKOD-AndroidClient trong thư mục Source bằng Android Studio. Nhấn Open an existing Android Studio project nếu chưa mở project nào hoặc File → Open... nếu đang mở 1 project khác:



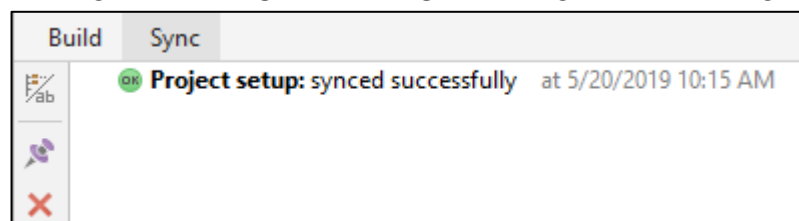
- Dợi 1 thời gian để đồng bộ project:



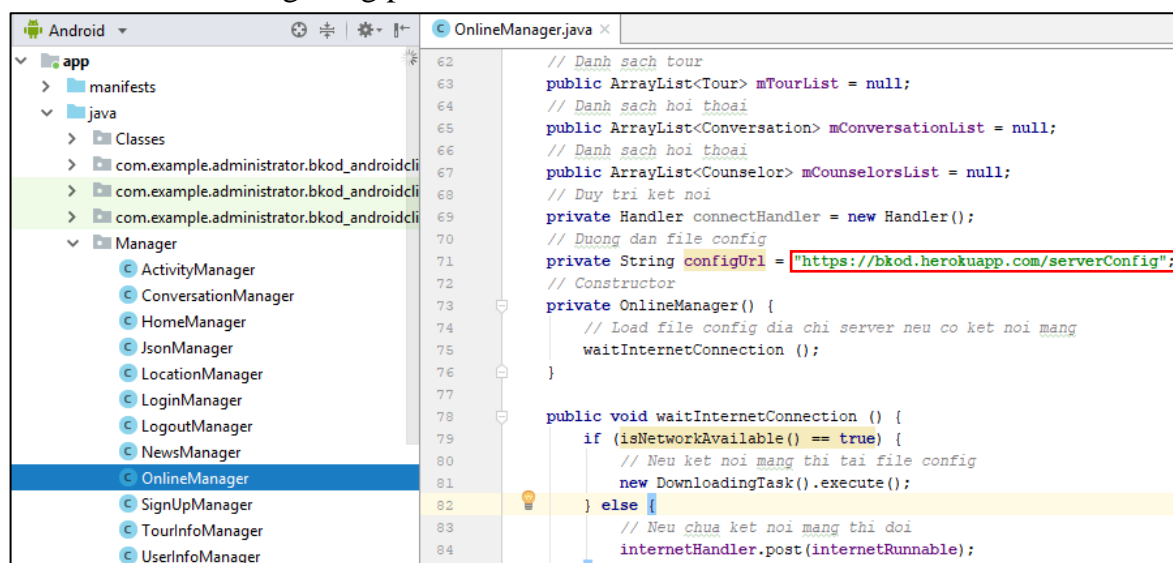
- Nếu đồng bộ bị lỗi thì mở file `build.gradle` trong thư mục `app` kiểm tra:



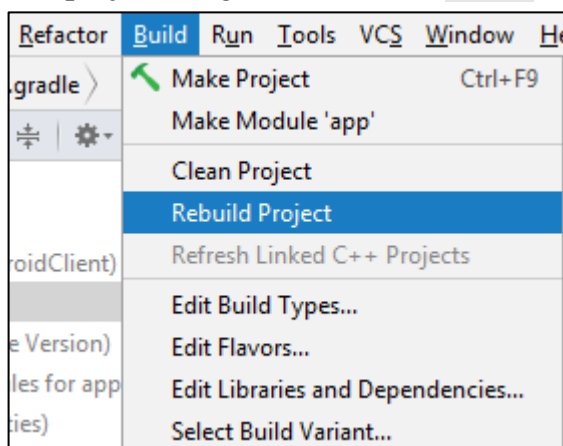
- Chú ý các dòng:
 - + `compileSdkVersion`: Phiên bản trình biên dịch, ở đây là 28 tương ứng với phiên bản cao nhất của SDK Platforms đã cài đặt phía trên, tương tự với `targetSdkVersion`.
 - + `minSdkVersion`: Phiên bản SDK Platforms thấp nhất đã cài đặt phía trên.
 - + `buildToolsVersion`: Phiên bản Android SDK Build-Tools đã ghi nhớ ở phía trên.
- Sau khi đồng bộ, chương trình thông báo đồng bộ thành công:



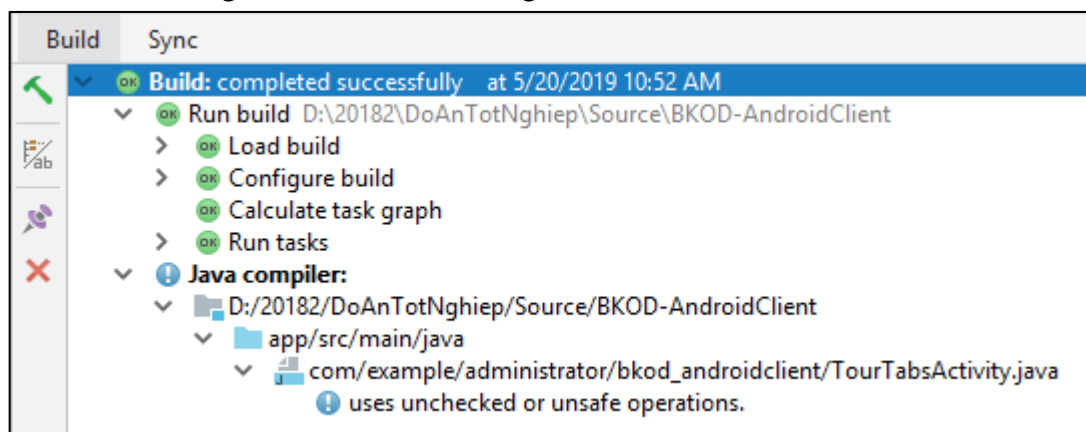
- Mở file `java` → `Manager` → `OnlineManager.java` để cấu hình địa chỉ server trên ứng dụng phía client:



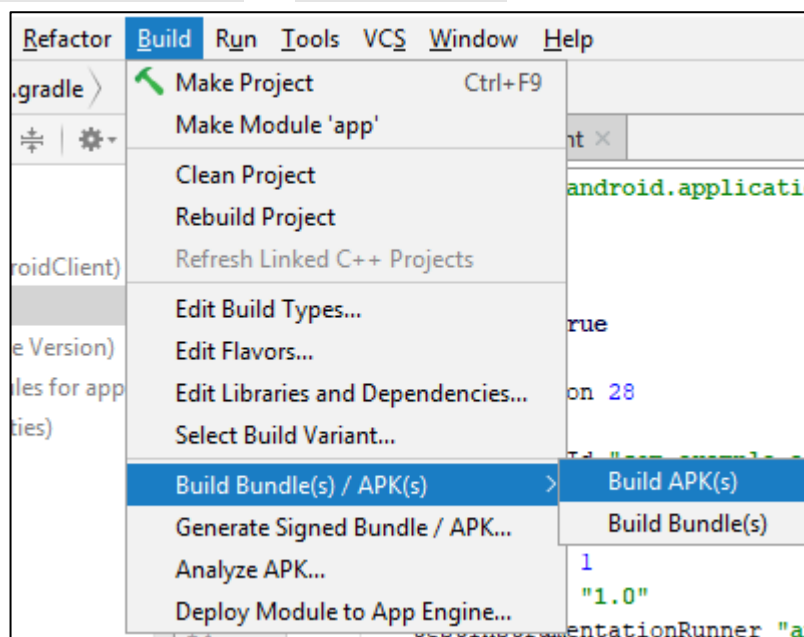
- Kéo xuống dòng 71, biến `configUrl` chứa đường dẫn file server config ở bước cấu hình địa chỉ server phía trên.
- Tiến hành build lại project bằng cách nhấn vào `Build` → `Rebuild Project`:



- Chương trình sẽ mất một thời gian để build lại project. Sau khi build, chương trình thông báo build thành công:

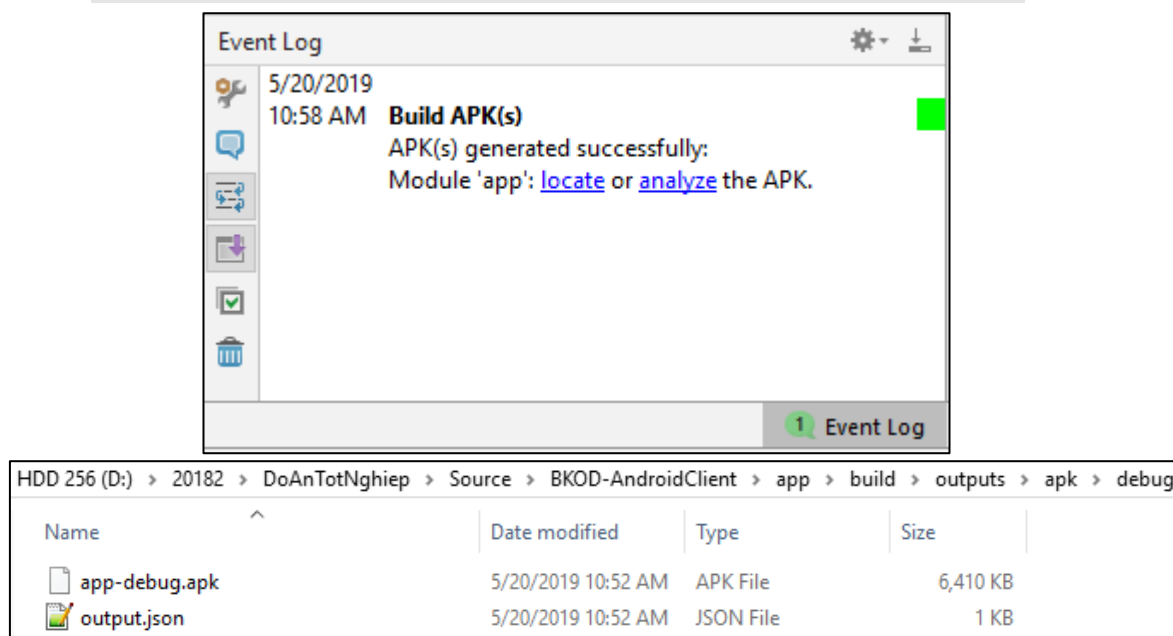


- Sinh ra file .apk để cài đặt lên điện thoại bằng cách nhấn vào Build → Build Bundle(s) / APK(s) → Build APK(s):

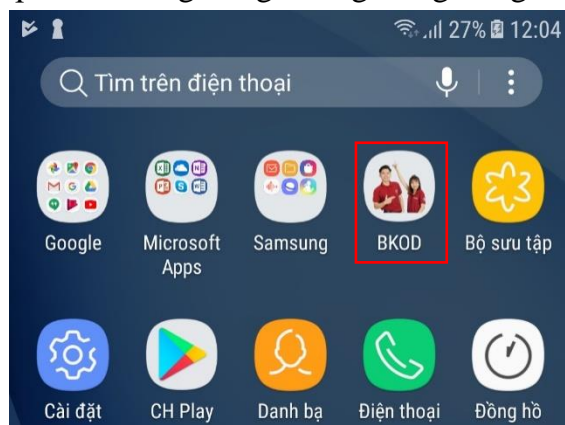


- Sau khi build xong, mở tab **Event Log**, nhấn vào **locate** ở thông báo build thành công, chương trình sẽ mở thư mục chứa file .apk hoặc có thể tìm đến thư mục này theo đường dẫn

Source\BKOD-AndroidClient\app\build\outputs\apk\debug:



- Copy file .apk này vào thiết bị chạy hệ điều hành Android và cài đặt (trước đó cần cho phép cài đặt ứng dụng không rõ nguồn gốc trên thiết bị):



- Khi mở ứng dụng lên, nếu kết nối thành công tới server, ứng dụng sẽ chuyển từ giao diện Progress bar sang giao diện đăng nhập.

PHẦN IV: TỔNG KẾT VÀ ĐÁNH GIÁ

4.1. Kết quả đạt được

- Ứng dụng đã hoàn thành được yêu cầu được đặt ra.
- Hoàn thành các chức năng, hạng mục như đã yêu cầu.
- Ứng dụng có giao diện đẹp và dễ sử dụng cũng như thân thiện cho người dùng.
- Kết nối tới server ổn định, ngay cả khi tắt màn hình.
- Tuy nhiên bên cạnh đó do trong khoảng thời gian hạn chế vẫn còn một số nhược điểm như:
 - + Các gói tin Json rất dễ bị bắt trên đường truyền.
 - + Giao diện còn hiện tượng giật, lag.
 - + Phiên bản socket.io client hiện tại chỉ hỗ trợ các dòng điện thoại chạy hệ điều hành Android phiên bản từ 8.1 (API level 27) trở xuống

4.2. Phương hướng phát triển cho đồ án

- Một số phương hướng phát triển sau khi hoàn thành đồ án:
 - + Tăng cường bảo mật bằng cách mã hoá các gói tin JSON giao tiếp giữa client và server.
 - + Phát triển trên các nền tảng di động khác cụ thể là IOS.
 - + Sử dụng một giao tiếp khác trên client hoặc nâng cấp thư viện `socket.io-client` để ứng dụng có thể chạy được trên hệ điều hành Android phiên bản 9.0 trở lên.
 - + Thiết kế giao diện bắt mắt hơn.
 - + Cài đặt thêm tính năng chat nhóm.
 - + Tích hợp AR API của Google Maps để sử dụng tính năng xem bản đồ thực tế tăng cường.

4.3. Kết luận

Đây là sản phẩm đầu tiên em viết bằng Android Studio nên còn nhiều thiếu sót, nhưng sẽ liên tục được cải thiện trong tương lai. Hi vọng đồ án này sẽ giúp ích cho các bạn học sinh đến tham quan trường và làm tiền đề cho các đồ án về Android, Node.js sau này. Một lần nữa, em xin cảm ơn ThS. Nguyễn Đức Tiến đã định hướng giúp em thực hiện đồ án này, cảm ơn Công ty TNHH Công nghệ và Truyền thông MDC đã hỗ trợ và giúp đỡ em rất nhiều trong quá trình làm đồ án.

TÀI LIỆU THAM KHẢO

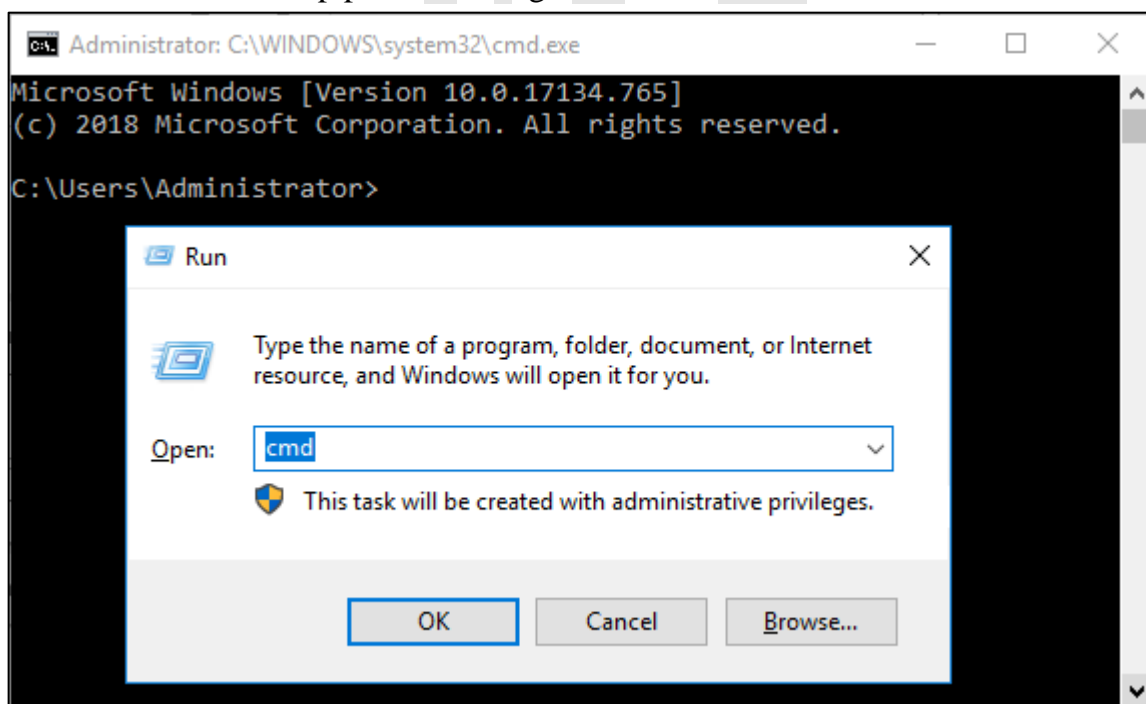
*** Danh mục website:**

1. Android: Custom Title Bar:
<https://stackoverflow.com/questions/3157522/android-custom-title-bar>
2. DATA BINDING TRONG ANDROID:
<https://viblo.asia/p/data-binding-trong-android-Do754EpJKM6>
3. Socket.IO-client Java:
<https://github.com/socketio/socket.io-client-java>
4. Android Spinner:
<https://stackoverflow.com/questions/5335306/how-can-i-get-an-event-in-android-spinner-when-the-current-selected-item-is-sele>
5. Sử dụng viewpager và Tablayout trong android:
<https://androidcoban.com/su-dung-viewpager-va-tablayout-trong-android.html>

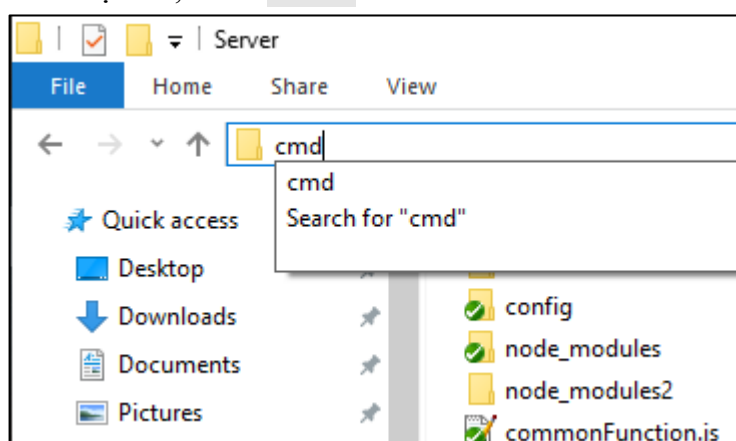
PHỤ LỤC

PHỤ LỤC 1: Hướng dẫn mở commandline trên các hệ điều hành

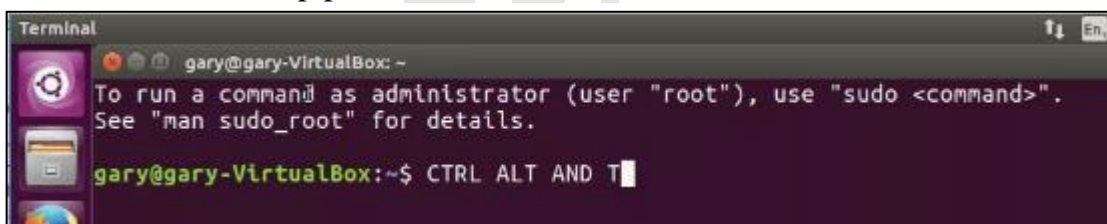
- Trên windows (Command prompt):
 - + Nhấn tổ hợp phím **Win** + **R**, gõ **cmd**, nhấn **enter**:



- + Để mở commandline trên 1 thư mục nhất định, gõ **cmd** vào thanh địa chỉ của thư mục đó, nhấn **enter**:

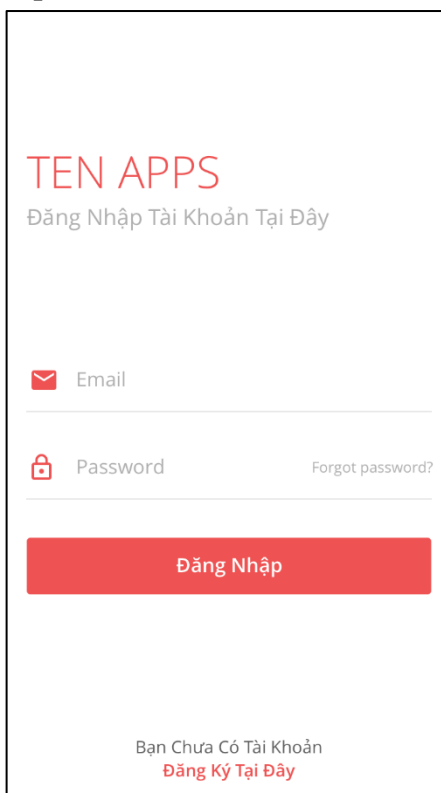


- Trên Ubuntu (Terminal):
 - + Nhấn tổ hợp phím `Ctrl + alt + T`:



PHỤ LỤC 2: Định hướng thiết kế giao diện cho phiên bản tiếp theo


- Giao diện đăng nhập:





- Giao diện đăng ký:

Tạo Tài Khoản

Điền Thông Tin Để Tạo Tài Khoản




 Email

 Mật Khẩu

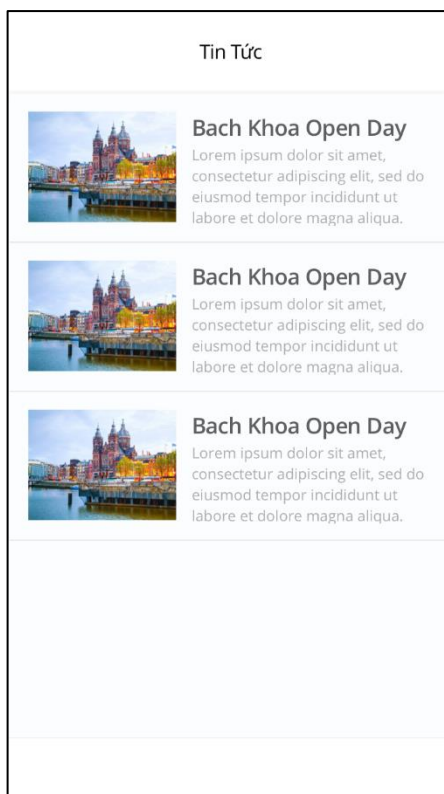
 Nhập Lại Mật Khẩu

Đăng Ký

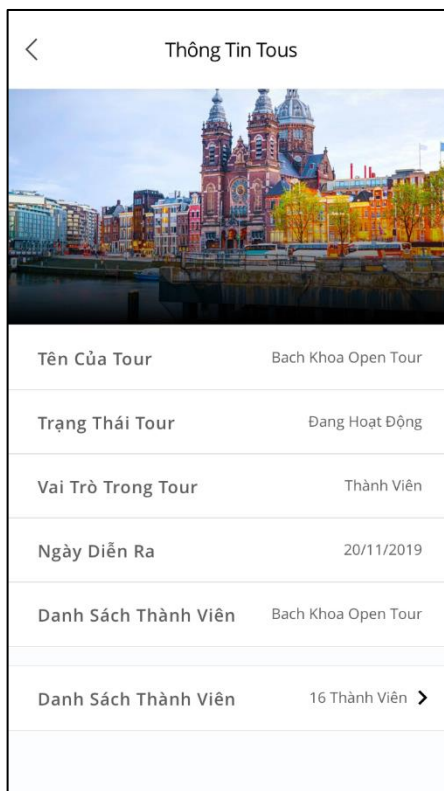
- Giao diện danh sách tour:

Danh Sách Tours	
	Why Do You Need To Join An Affiliate Marketing Network 20 Jan 2016 - Thành Viên - Chưa Diễn Ra
	Why Do You Need To Join An Affiliate Marketing Network 20 Jan 2016 - Thành Viên - Đang Diễn Ra
	Why Do You Need To Join An Affiliate Marketing Network 20 Jan 2016 - Thành Viên - Đã Diễn Ra

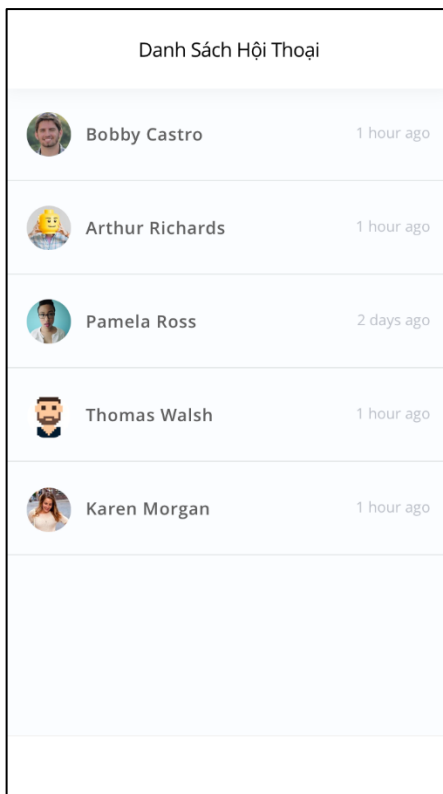
- Giao diện danh sách tin tức:



- Giao diện thông tin tour:



- Giao diện danh sách hội thoại:



- Giao diện nhắn tin:

