

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN
MÔN LẬP TRÌNH MẠNG CĂN BẢN**

ĐỀ TÀI: “Ứng dụng quản lí nhân viên”

Giảng viên hướng dẫn: Trần Hồng Nghi

Sinh viên thực hiện:

Đặng Chí Thịnh	–	22521399
Nguyễn Tuấn Phát	–	22521076
Trần Hoàng Tuấn Kiệt	–	22520724
Diệp Tấn Phát	–	22521066

LỜI CẢM ƠN

Trước hết, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến toàn thể giảng viên và nhân viên của Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM đã tạo điều kiện cho chúng em tham gia môn học **Lập trình mạng căn bản**, từ đó giúp chúng em có được những kiến thức cơ bản cần thiết để thực hiện đề tài này.

Đặc biệt, chúng em muốn gửi lời cảm ơn chân thành đến cô Trần Hồng Nghi – giảng viên lý thuyết của môn **Lập trình mạng căn bản**. Những kiến thức mà cô chia sẻ cùng với sự tận tâm và những lời khuyên quý báu đã giúp chúng em tiến bộ qua từng bước học tập.

Trải qua một học kỳ dài, chúng em đã nhận thấy cô **Trần Hồng Nghi** là một người cô thân thiện và chu đáo với sinh viên. Phong cách giảng dạy của cô không chỉ giúp chúng em dễ dàng tiếp thu kiến thức mà còn áp dụng vào cuộc sống hàng ngày một cách hiệu quả. Môn học này không chỉ đem lại niềm vui và sự hấp dẫn mà còn là nền tảng quan trọng giúp chúng em tiến xa hơn trong học tập và sự nghiệp sau này.

Sau quãng thời gian dài thực hiện đề tài, chúng em đã cùng nhau tìm hiểu, nghiên cứu và thảo luận để hoàn thiện công việc một cách cẩn thận nhất. Dù với ít kinh nghiệm và kỹ năng, cùng với việc làm việc nhóm lần đầu tiên, chắc chắn không tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý từ cô để có thể hoàn thiện hơn qua đề tài này.

Nhóm 9 – lớp NT106.022.ANTT, xin chúc cô sức khỏe dồi dào, thành công và hạnh phúc trong cuộc sống.

Chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng 6 năm 2024

Nhóm sinh viên thực hiện

[illegible]

MỤC LỤC

LỜI CẢM ƠN	1
NHẬN XÉT CỦA GIẢNG VIÊN	2
CHƯƠNG I: TỔNG QUAN ĐỀ TÀI.....	4
1. Giới thiệu đề tài	4
2. Mục tiêu đề tài	4
3. Công nghệ sử dụng.....	4
CHƯƠNG II: CƠ SỞ LÝ THUYẾT.....	5
1. Tổng quan về C# và Winform.....	5
2. Lập trình mạng với C# WinForm	5
3. Firebase	5
CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	6
1. Phân rã chức năng & Use Case.....	6
2. Thiết kế database.....	7
3. Thiết kế giao diện UI.....	8
CHƯƠNG IV: HIỆN THỰC ĐỀ TÀI	14
1. Xác thực người dùng.....	14
2. Trong giao diện chính (USER).....	15
3. Trong giao diện chính (ADMIN)	17
CHƯƠNG V: THỰC NGHIỆM ĐỀ TÀI	19
1. Giao diện	19
2. Chức năng	19
3. Hiệu suất	19
4. Bảo mật	19
5. Tương thích.....	19
CHƯƠNG VI: KẾT LUẬN, HƯỚNG PHÁT TRIỂN	20
1. Đánh giá chung.....	20
2. Nhược điểm	20
3. Ưu điểm	20
4. Hướng phát triển	20

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

1. Giới thiệu đề tài

- Trong thời đại công nghệ số ngày nay, việc quản lý nhân viên trở nên cực kỳ quan trọng đối với mọi doanh nghiệp và tổ chức. Để đáp ứng nhu cầu này một cách hiệu quả và hiện đại, việc sử dụng ứng dụng quản lý nhân viên trở thành một phương tiện không thể thiếu.

- Đề tài này nhằm vào việc nghiên cứu, phát triển và triển khai một hệ thống phần mềm nhằm quản lý thông tin và hoạt động của nhân viên trong một tổ chức. Điều này bao gồm việc tổ chức, quản lý hồ sơ nhân viên, quản lý vấn đề liên quan đến lịch làm việc và nhiệm vụ, cũng như tạo điều kiện tốt nhất cho sự tương tác và giao tiếp giữa các thành viên trong tổ chức.

2. Mục tiêu đề tài

- Tìm hiểu về cách xây dựng một chương trình, ứng dụng trên nền tảng WindowForm. Giao diện phần mềm thân thiện và dễ dàng sử dụng.

- Tập trung vào nghiên cứu, phát triển và triển khai một hệ thống phần mềm nhằm cải thiện quản lý nhân sự trong môi trường doanh nghiệp.

- Nhằm mang lại giải pháp hiệu quả và linh hoạt cho các tổ chức trong việc quản lý nguồn nhân lực, góp phần vào sự thành công và thịnh vượng của doanh nghiệp trong môi trường kinh doanh ngày nay.

3. Công nghệ sử dụng

- Ngôn ngữ: C# (Visual Studio)

- Database: Firebase (Realtime Database)

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

1. Tổng quan về C# và Winform:

- Là ngôn ngữ lập trình thuần hướng đối tượng được phát triển bởi Microsoft.
- Được tạo ra dựa trên nền tảng của C++ và Java nên có ưu điểm cũng như sự cân bằng giữa các NNLT này.
- C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo 1 ứng dụng Window Forms hay WPT, trở nên rất dễ dàng.

2. Lập trình mạng với C# WinForm:

C# hỗ trợ các công nghệ lập trình dành cho lập trình mạng như:

- Socket:
 - Tạo, quản lý kết nối mạng, API.
 - Là một điểm cuối để giao tiếp giữa hai máy tính. Nó cho phép các ứng dụng gửi và nhận dữ liệu qua mạng.
 - Sockets có thể được sử dụng cho cả giao thức TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).
- TCP/IP và UDP:
 - Cung cấp 2 giao thức truyền dữ liệu phổ biến, TCP/IP cho độ tin cậy và UDP cho độ nhanh chóng.
- WebSocket:
 - Cho phép giao tiếp real-time giữa client-server, có ích cho chẳng hạn như ứng dụng chat, trò chơi trực tuyến, và các ứng dụng web tương tác.
- Threading:
 - Giúp xử lý nhiều kết nối mạng cùng lúc trên máy chủ, tận dụng tối đa tài nguyên của hệ thống.

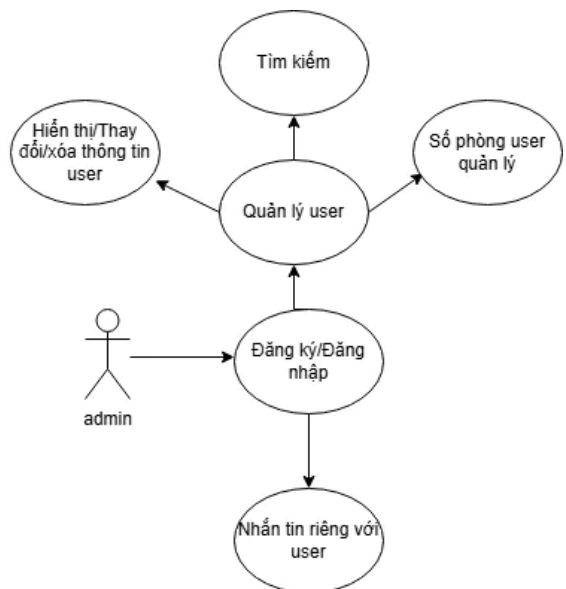
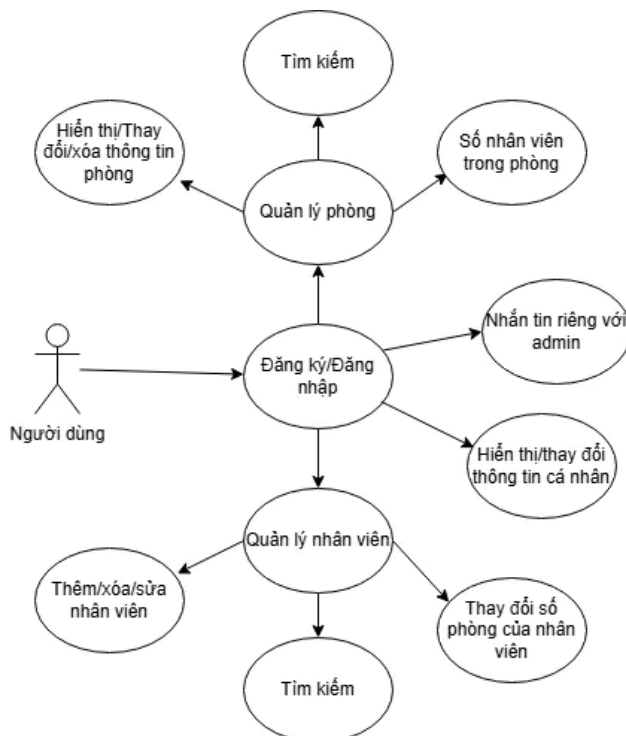
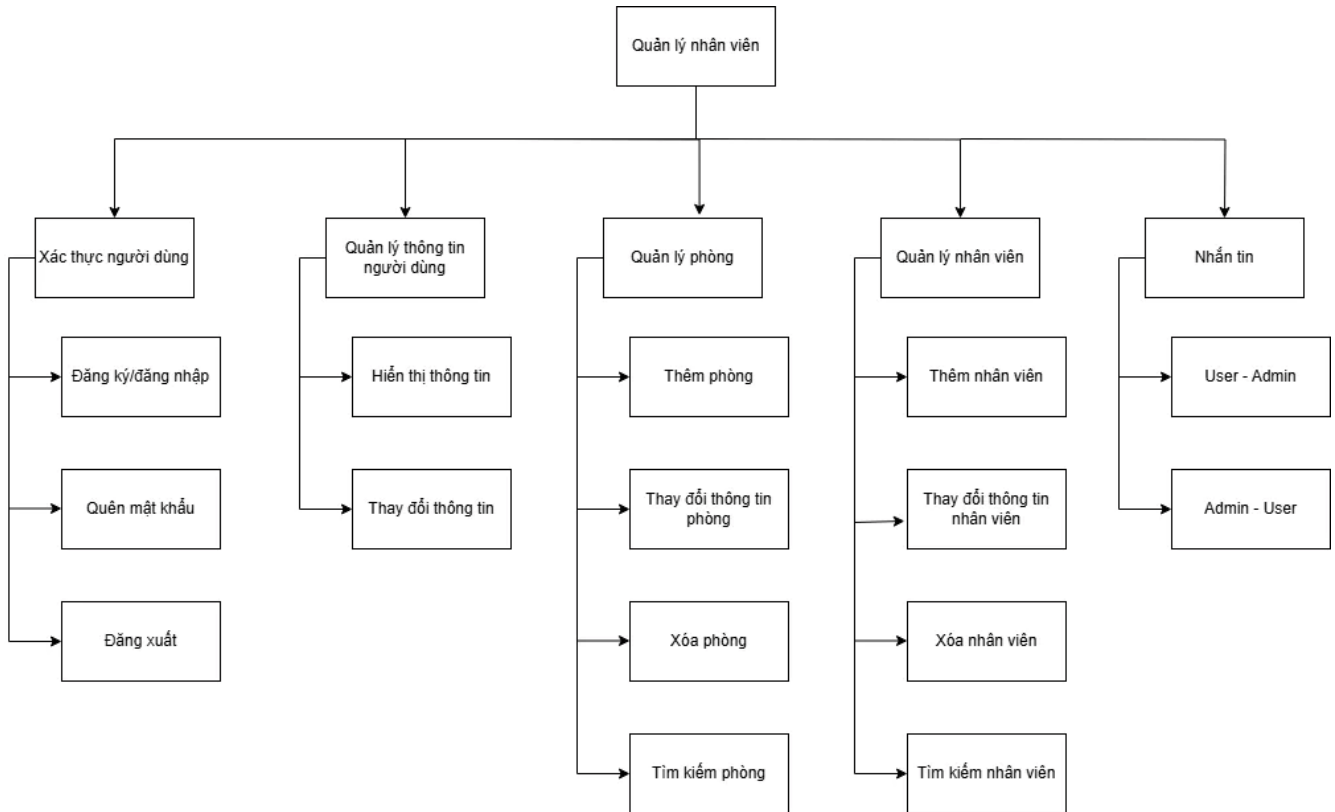
3. Firebase:

Firebase là một nền tảng đa năng, nó cung cấp rất nhiều dịch vụ khác nhau cho người dùng.

- Real-time Database: Dịch vụ này được lưu trữ trực tiếp trên iCloud.
- Authentication: cung cấp cho ứng dụng của bạn một số phương pháp xác thực thông qua email, mật khẩu, số điện thoại, tài khoản Google, tài khoản Facebook...
- Firebase cloud messaging: cho phép người dùng xây dựng ứng dụng chat và đẩy thông báo tới nhiều thiết bị khác nhau như web, Android, iOS...
- Firebase data query: đơn giản hóa quá trình lấy dữ liệu thay vì phải thông qua các câu lệnh SQL phức tạp.

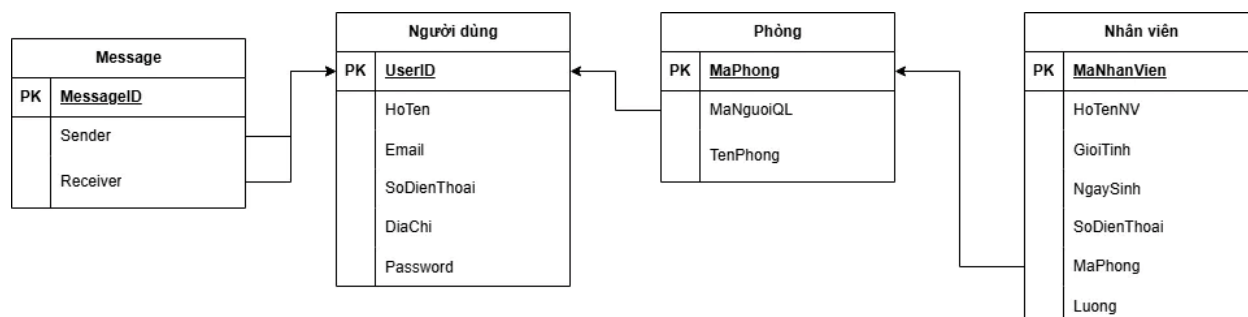
CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. Phân rã chức năng & Use Case



2. Thiết kế database

Sơ đồ cơ sở dữ liệu



- Đặc tả các đối tượng:

a. Người quản lý (người dùng):

STT	Thuộc tính	Trường dữ liệu	Kiểu dữ liệu	Mô tả
1	UserID	Mã người quản lý	varchar(20)	Khóa chính , viết liền, không dấu
2	HoTen	Họ tên người quản lý	varchar(30)	
3	Email	Email	varchar(30)	Phải có @
4	SoDienThoai	Số điện thoại	varchar(10)	
5	DiaChi	Địa chỉ	varchar(30)	
6	Passowrd	Mật khẩu của account	varchar(30)	độ dài ít nhất 8, có ít nhất một chữ viết hoa, một ký tự đặc biệt

b. Phòng:

STT	Thuộc tính	Trường dữ liệu	Kiểu dữ liệu	Mô tả
1	MaPhong	Mã phòng	varchar(8)	Khóa chính , viết liền không dấu
2	TenPhong	Tên phòng	varchar(30)	
3	MaNguoiQL	Mã người quản lý	varchar(10)	Khóa ngoại , tham chiếu đến User(UserID)

c. Nhân viên:

STT	Thuộc tính	Trường dữ liệu	Kiểu dữ liệu	Mô tả
1	MaNhanVien	Mã nhân viên	varchar(20)	Khóa chính
2	HoTenNV	Họ tên nhân viên	varchar(30)	
3	GioiTinh	Giới tính (nam/nữ)	boolean	1 = nam, 0 = nữ
4	NgaySinh	Ngày/tháng/năm sinh	smalldatetime	
5	SoDienThoai	Số điện thoại	varchar(10)	
6	MaPhong	Mã phòng mà nhân viên công tác	varchar(8)	Khóa ngoại , tham chiếu đến Phong(MaPhong)
7	Luong	Lương của nhân viên	int	

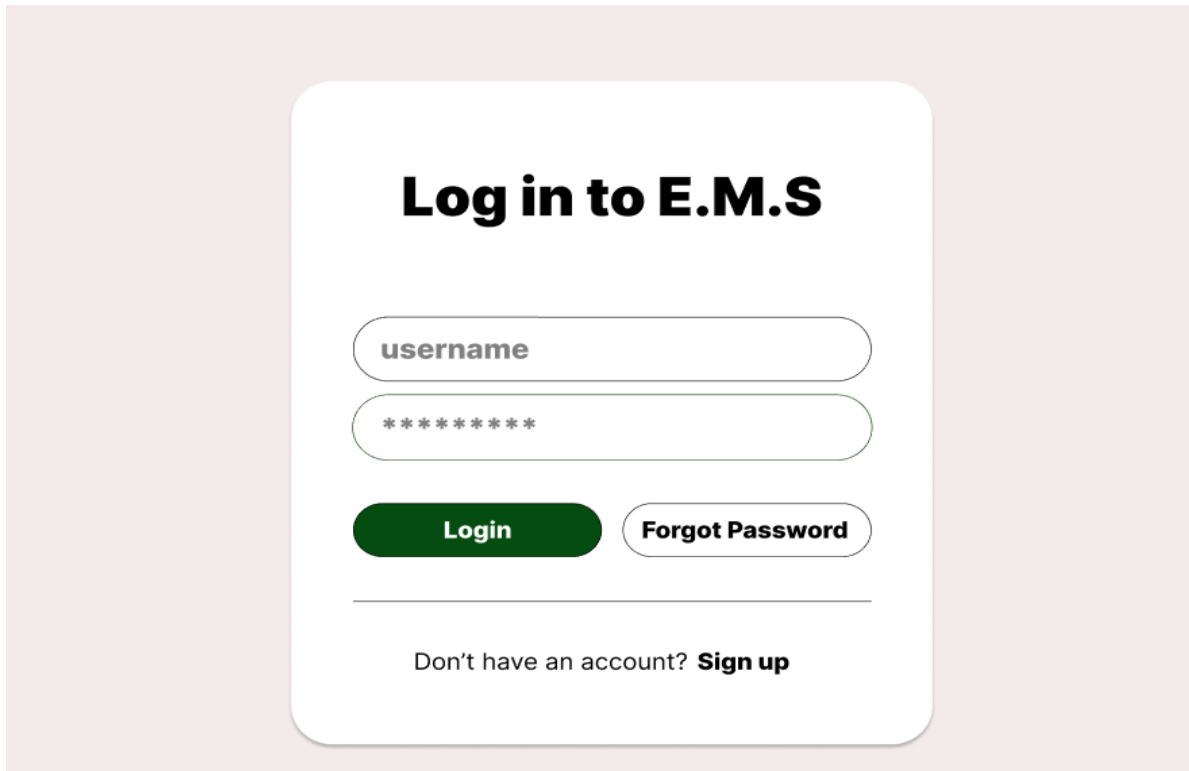
d. Message:

STT	Thuộc tính	Trường dữ liệu	Kiểu dữ liệu	Mô tả
1	MessageID	Mã message	varchar(10)	Khóa chính
2	Sender	Người gửi	varchar(10)	Khóa ngoại , tham chiếu đến User(UserID)
3	Receiver	Người nhận	varchar(10)	Khóa ngoại , tham chiếu đến User(UserID)

3. Thiết kế giao diện UI

STT	Tên giao diện	Mô tả giao diện
1	Login page	Trang để đăng nhập
2	Register page	Trang để đăng ký tài khoản
3	Forgot-password page	Trang đổi mật khẩu khi quên mật khẩu
4	Logout page	Đăng xuất
5	Profile page	Hiển thị thông tin cá nhân của người dùng
6	Room page	Hiển thị thông tin phòng ban
7	Employee page	Hiển thị thông tin nhân viên
8	User-chat page	Giao diện chat bên phía user
9	Admin-user page	Hiển thị thông tin quản lý user
10	Admin-chat page	Giao diện chat bên phía admin

III.3.1 Login page



The login page features a white rounded rectangle centered on a light pink background. At the top, the text "Log in to E.M.S" is displayed in a large, bold, black font. Below this, there are two input fields: the first is labeled "username" and the second is filled with asterisks "*****". Under the input fields are two buttons: a dark green "Login" button and a white "Forgot Password" button with a dark green border. A horizontal line separates these buttons from the text "Don't have an account? Sign up", where "Sign up" is in bold.

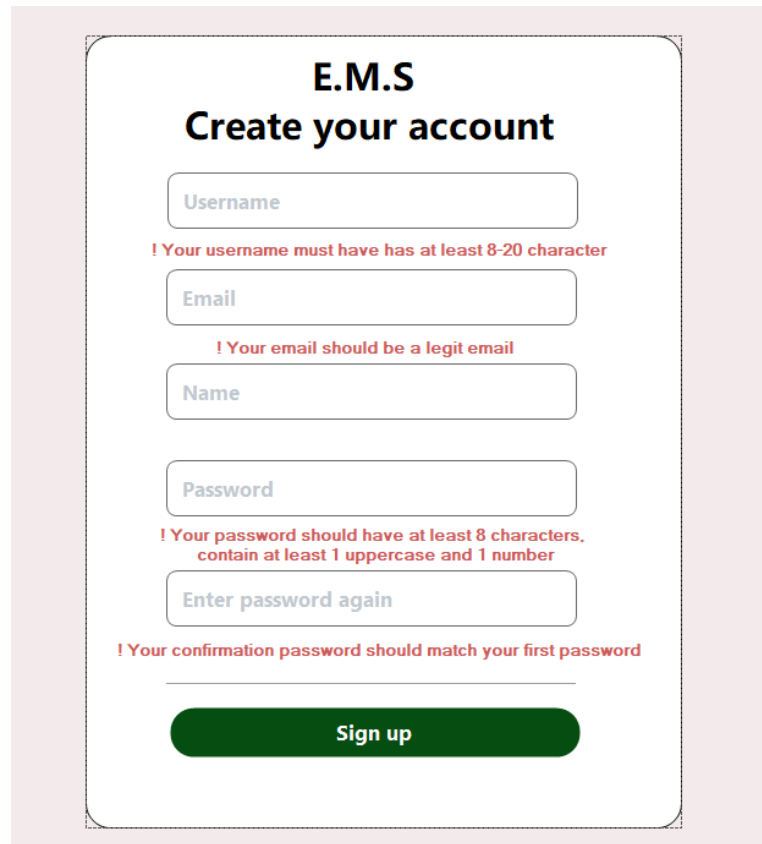
Log in to E.M.S

username

Login Forgot Password

Don't have an account? **Sign up**

III.3.2 Register page



The register page is a white rounded rectangle on a light pink background. It is titled "E.M.S Create your account" in bold black text. The form contains five input fields: "Username", "Email", "Name", "Password", and "Enter password again". Each input field is followed by a red error message: "Your username must have has at least 8-20 character", "Your email should be a legit email", and "Your password should have at least 8 characters, contain at least 1 uppercase and 1 number". A final red message "Your confirmation password should match your first password" is placed below the last two fields. A dark green "Sign up" button is at the bottom.

E.M.S
Create your account

Username

! Your username must have has at least 8-20 character

Email

! Your email should be a legit email

Name

Password

! Your password should have at least 8 characters, contain at least 1 uppercase and 1 number

Enter password again

! Your confirmation password should match your first password

Sign up

III.3.3 Forgot-password page

Change account password

Email


Send confirmation code

Confirmation code

New password

Confirm

III.3.4 Logout page



Are you sure to logout?

Cancel

Okay

III.3.5 Profile page

Employee Management System

Room


Employee


Chat

My profile

Logout

E.M.S

 a



Username: a

Email:

Name:

Password: (Enter your password to change your information)

Confirm

Change password

III.3.6 Room page

Employee Management System

Room


Employee

Chat

My profile

Logout

E.M.S

 a

List of Room

AddUpdateDelete

Search by room ID:

Reload

RoomID	RoomName	Number of employees
1	Phòng kế hoạch	0
2	Phòng kế toán	0
3	Phòng kinh doanh	2
4	Phòng hành chính	0
5	Phòng dụng cụ	0
6	aaaa	0
7	dsdsdsd	0

III.3.7 Employee page

Employee Management System

Room


Employee

Chat

My profile

Logout

E.M.S

 a

List of Employee

Search by employee ID:

AddUpdateDeleteReload

ID	Name	Gender	Email	Birth Day	Phone	Salary	Room ID
11	AAA	Male	2252@a	6/1/2024 11:...	12345	0	3
22	dasdasda	Male	2@2	6/13/2024 11:...	1234	0	3

III.3.8 User-chat page

Employee Management System

Room


Employee

Chat

My profile

Logout

E.M.S

 a

LOLOLOLOLOLOLOL

from thinh dang

dãdã
hi admin
hello
hello
ok
good

SEND

III.3.9 Admin-user page

The screenshot shows the 'Admin main form' window. The left sidebar is dark green with the text 'Employee Management System' and buttons for 'User', 'Server', and 'Logout'. The top header is dark blue with 'E.M.S - Admin' and a user profile for 'admin'. The main content area is titled 'List of User' and features a search bar, 'Update' and 'Delete' buttons, and a table of users.

Username	Full name	Email	Rooms managed
a	a	a@gmail.com	7
admin			0
b	b	b@gmail.com	0
lmaolmao	lmaolmao	a@gmail.com	2

III.3.10 Admin-chat page

The screenshot shows the 'Admin-chat' page. The layout is similar to the previous page, but the main content area is a chat interface. It includes a search bar with 'a' selected, a list of chat messages, and a 'SEND' button.

Chat messages:

- hi admin
- hello
- hello
- from thinh dang
- ok
- good

Input field: [] SEND

CHƯƠNG 4: HIỆN THỰC ĐỀ TÀI

1. Xác thực người dùng:

1.1 Đăng ký tài khoản:

- User nhập vào Username, email, name, password, password 2nd. Khi nhập đúng các regex của các trường thì mới có thể gửi request để đăng ký tài khoản:

- + username: 8-20 chữ cái
- + email: phải là một email hợp lệ
- + password: ít nhất 8 kí tự, chứa ít nhất một chữ in hoa, một kí tự đặc biệt
- + password 2nd : phải giống password

- Sau khi nhấn gửi, server sẽ nhận các thông tin và bắt đầu kiểm tra các thông tin, server sẽ kiểm tra username và email đã nhập đã tồn tại trên database chưa, nếu chưa thì sẽ gửi một email bằng giao thức **SMTP** chứa code xác thực vào email đã nhập. Nếu nhập đúng code thì sẽ tạo một tài khoản user mới với các thông tin đã nhập, sau đó sẽ chuyển qua trang **đăng nhập**.

1.2 Đăng nhập tài khoản:

- User sẽ nhập vào username và password, nếu nhập đúng thì sẽ được chuyển tiếp sang trang home của tài khoản.

- Nếu user quên mật khẩu thì có thể bấm vào button “**Forgot Password**” để chuyển qua trang **quên mật khẩu** để bắt đầu quá trình đổi mật khẩu.

1.3 Quên mật khẩu:

- Đầu tiên user sẽ nhập vào email của tài khoản muốn đổi mật khẩu, nếu email có tồn tại trong database thì server sẽ gửi một email bằng giao thức **SMTP** chứa mã code xác thực vào email đã nhập.

- Sau đó người dùng nhập vào mật khẩu mới muốn đổi vào cùng với mã code xác thực nhận được trong email. Nếu password thỏa regex giống như **đăng ký tài khoản** và đúng mã code thì sẽ thay đổi mật khẩu của email đó và chuyển về lại trang **đăng nhập**.

2. Trong giao diện chính (USER):

2.1 Đăng xuất (Logout):

- Khi bấm vào nút “**Logout**” thì sẽ chuyển qua trang mới để xác nhận đăng xuất. Nếu xác nhận thì sẽ thoát hết các trang hiện tại và chuyển về lại trang **đăng nhập**.

2.2 User profile:

- *Hiển thị thông tin* của nhân viên ở đây, gồm **username, name, email**. User chỉ có thể thay đổi name, email ở đây. Để thay đổi 2 thông tin trên thì user cần nhập vào đúng password của tài khoản.

- *Khi muốn đổi mật khẩu*, sẽ chuyển sang trang “**Forgot Password**” và thao tác giống như trên (1.3) để đổi mật khẩu.

2.3 Quản lý room:

- *Hiển thị thông tin* của những phòng ban mà user quản lý, thông tin hiển thị bao gồm: **mã phòng, tên phòng, số lượng nhân viên trong phòng**.

- *Thêm phòng:*

+ Khi bấm vào nút **Add** sẽ popup một form mới giúp thêm phòng, thông tin cần nhập: **mã phòng, tên phòng**.

+ Khi xác nhận, server sẽ kiểm tra mã phòng đó đã tồn tại trên server hay chưa, nếu chưa thì sẽ set **số lượng nhân viên trong phòng = 0** và thêm room đó lên database và đóng form.

- *Update phòng:*

+ Khi chọn một row phòng trên listView và bấm nút **Update** sẽ popup một form mới với các thông tin hiển thị của phòng đó (trừ số lượng nhân viên) trên form, người dùng sẽ thay đổi các thông tin của phòng theo ý muốn.

+ Khi bấm xác nhận server sẽ kiểm tra mã phòng mới trên database, nếu không tồn tại thì sẽ thay đổi thông tin và đóng form, còn không thì sẽ báo lỗi.

- *Xóa phòng:* Khi chọn một row phòng trên listView và bấm nút **Delete** sẽ popup một messageBox xác nhận muốn thay đổi hay không. Nếu chọn có thì server sẽ bắt đầu xóa room đó trên database, nếu không thì sẽ đóng form.

- *Tìm kiếm:* Nhập vào roomID, sẽ hiển thị những phòng có roomID giống với roomID đã nhập lên listView.

- *Reload:* Tải lại data các phòng của user.

2.4 Quản lý employee:

- *Hiển thị thông tin* của những nhân viên mà user quản lý, thông tin hiển thị bao gồm: **mã nhân viên, họ tên, giới tính, email, ngày sinh, SĐT, lương và mã phòng mà nhân viên đó đang công tác.**

- *Thêm nhân viên:*

+ Khi bấm vào nút **Add** sẽ popup một form mới giúp thêm phòng, thông tin cần nhập: **mã nhân viên, họ tên, giới tính, email, ngày sinh, SĐT, mã phòng.** Mã phòng ở đây chọn trong comboBox chứa các **mã phòng** mà user quản lý.

+ Khi xác nhận, server sẽ kiểm tra mã nhân viên đó đã tồn tại trên server hay chưa, nếu chưa thì sẽ set **lương = 0** và thêm nhân viên đó lên database và đóng form.

- *Update nhân viên:*

+ Khi chọn một row nhân viên trên listView và bấm nút **Update** sẽ popup một form mới với các thông tin hiển thị của nhân viên đó trên form, người dùng sẽ thay đổi các thông tin của nhân viên theo ý muốn.

+ Khi bấm xác nhận server sẽ kiểm tra mã nhân viên mới trên database, nếu không tồn tại thì sẽ thay đổi thông tin và đóng form, còn không thì sẽ báo lỗi.

- *Xóa nhân viên:* Khi chọn một row nhân viên trên listView và bấm nút **Delete** sẽ popup một messageBox xác nhận muốn thay đổi hay không. Nếu chọn có thì server sẽ bắt đầu xóa nhân viên đó trên database, nếu không thì sẽ đóng form.

- *Tìm kiếm:* Nhập vào employeeID, sẽ hiển thị những nhân viên có employeeID giống với employeeID đã nhập lên listView.

- *Reload:* Tải lại data các nhân viên của user.

2.5 Chat với admin:

- User chỉ có thể chat được khi bên phía admin mở kết nối server, nếu không thì user chỉ có thể xem lại những tin nhắn cũ trước đó của mình với admin.

- *Load message cũ:* Sẽ tìm trên database những message có **sender** là user hiện tại và **receiver** là admin và ngược lại. Sau đó chuyển message đó lên **FlowLayoutPanel** để hiển thị các message:

+ *message từ user:* background xanh nhẹ, nằm ở phía bên *phải* của panel.

+ *message từ admin:* background xám, nằm ở phía bên *trái* của panel.

- Khi đã kết nối, user và admin có thể gửi tin nhắn **realtime** cho nhau qua giao thức **TCP**:

+ *Gửi:* chuyển message đã nhập trên messageTextbox thành bytes và chuyển lên server, message chuyển lên có format “**{receiver}:{sender}:{content}**” với **receiver** ở đây là “*admin*”, **sender** là *username của user đang gửi message*. Sau đó sẽ thêm

message đó lên **FlowLayoutPanel** để hiển thị tin nhắn, message đó sẽ có background xanh nhẹ và nằm ở phía bên *phải* của panel. Đồng thời sẽ push message đó lên **database** với các thông tin: **content, sender, receiver**.

+ *Nhận*: chuyển message nhận được từ bytes về lại string và kiểm tra xem có phải đúng là message do *admin* gửi cho mình hay không qua việc split message để lấy thông tin receiver-sender. Nếu đúng là message gửi cho mình thì sẽ thêm vào **FlowLayoutPanel** để hiển thị tin nhắn, message đó sẽ có background xám và nằm bên *trái* của panel.

3. Trong giao diện chính (ADMIN):

- Người dùng chỉ có thể vào trang của admin khi đăng nhập bằng tài khoản **admin**.

3.1 Đăng xuất (Logout):

- Khi bấm vào nút “**Logout**” thì sẽ chuyển qua trang mới để xác nhận đăng xuất. Nếu xác nhận thì sẽ thoát hết các trang hiện tại và chuyển về lại trang **đăng nhập**.

3.2 Quản lý user:

- *Hiển thị thông tin* của tất cả user tồn tại, thông tin hiển thị bao gồm: **username, name, email, số phòng mà user quản lý**.

- *Update user*:

+ Khi chọn một row user trên listView và bấm nút **Update** sẽ popup một form mới với các thông tin hiển thị của user (*có cả password*) đó trên form, người dùng sẽ thay đổi các thông tin của user theo ý muốn (*bao gồm cả password của user đó, nhưng không thể thay đổi username*).

+ Khi bấm xác nhận server sẽ kiểm tra username, email mới trên database, nếu không tồn tại thì sẽ thay đổi thông tin và đóng form, còn không thì sẽ báo lỗi.

- *Xóa user*: Khi chọn một row user trên listView và bấm nút **Delete** sẽ popup một messageBox xác nhận muốn thay đổi hay không. Nếu chọn có thì server sẽ bắt đầu xóa user đó trên database cùng toàn bộ room/employee mà user đó quản lý, nếu không thì sẽ đóng form.

- *Tìm kiếm*: Nhập vào username, sẽ hiển thị những user có username giống với username đã nhập lên listView.

3.3 Chat với từng user:

- Khi bấm vào nút **Server**, admin sẽ mở kết nối cho phép các user tham gia vào để chat.

- *Chọn user muốn chat*: admin chọn user mà mình muốn chat trên comboBox trên cùng của form, comboBox sẽ chứa tất cả username của các user trong database (*trừ admin*).

- *Load message cũ mỗi khi chọn user trên comboBox*: Sẽ tìm trên database những message có **sender** là username của user đang chat và **receiver** là admin và ngược lại. Sau đó chuyển message đó lên **FlowLayoutPanel** để hiển thị các message:

+ *message từ user*: background xám, nằm ở phía bên *trái* của panel.

+ *message từ admin*: background xanh nhẹ, nằm ở phía bên *phải* của panel.

- Khi đã kết nối, user và admin có thể gửi tin nhắn **realtime** cho nhau qua giao thức **TCP**:

+ *Gửi*: chuyển message đã nhập trên messageTextbox thành bytes và chuyển lên server, message chuyển lên có format “**{receiver}:{sender}:{content}**” với **receiver** ở đây là *username trên comboBox*, **sender** là “*admin*”. Sau đó sẽ thêm message đó lên **FlowLayoutPanel** để hiển thị tin nhắn, message đó sẽ có background xanh nhẹ và nằm ở phía bên *phải* của panel. Đồng thời sẽ push message đó lên **database** với các thông tin: **content, sender, receiver**.

+ *Nhận*: chuyển message nhận được từ bytes về lại string và kiểm tra xem có phải đúng là message do *user đang hỗ trợ* gửi cho mình hay không qua việc split message để lấy thông tin receiver-sender. Nếu đúng là message gửi cho mình thì sẽ thêm vào **FlowLayoutPanel** để hiển thị tin nhắn, message đó sẽ có background xám và nằm bên *trái* của panel.

CHƯƠNG V: THỰC NGHIỆM ĐỀ TÀI

1. Giao diện

- Tính thẩm mỹ: Giao diện đẹp mắt, hài hòa với màu sắc và font chữ dễ đọc.
- Tính trực quan: Các yếu tố giao diện dễ hiểu, dễ sử dụng với các biểu tượng và nhãn rõ ràng.
- Khả năng điều hướng: Người dùng dễ dàng di chuyển giữa các trang và chức năng khác nhau, sử dụng menu và nút điều hướng hợp lý.
- Phản hồi người dùng: Ứng dụng cung cấp thông báo rõ ràng cho người dùng sau khi thực hiện các thao tác (thành công hoặc lỗi).

2. Chức năng

- Đầy đủ chức năng: Các chức năng yêu cầu (thêm, sửa, xóa, tìm kiếm) hoạt động đúng và đầy đủ.
- Tính chính xác: Chức năng thực hiện đúng theo yêu cầu và logic.
- Tính ổn định: Ứng dụng hoạt động ổn định, không bị lỗi hay treo khi thực hiện các thao tác thông thường.
- Khả năng mở rộng: Ứng dụng dễ dàng mở rộng và thêm mới chức năng trong tương lai.

3. Hiệu suất

- Thời gian tải dữ liệu: Thời gian tải và hiển thị danh sách nhân viên nhanh chóng.
- Phản hồi người dùng: Ứng dụng phản hồi nhanh chóng khi người dùng thực hiện các thao tác như thêm, sửa, xóa, tìm kiếm.
- Sử dụng tài nguyên: Ứng dụng sử dụng tài nguyên hệ thống (CPU, RAM) một cách hiệu quả.

4. Bảo mật

- Bảo mật dữ liệu trong quá trình truyền nhận chưa được chú trọng.

5. Tương thích

- Hệ điều hành: Ứng dụng hoạt động tốt trên các phiên bản Windows
- Độ phân giải màn hình: Giao diện hiển thị tốt trên các độ phân giải màn hình khác nhau.

*** Video demo:** <https://youtu.be/IGu4-oQ-2bw>

CHƯƠNG 6: KẾT LUẬN, HƯỚNG PHÁT TRIỂN

1. Đánh giá chung:

Sau thời gian hoàn thành đồ án vừa qua, bọn em xin tổng hợp lại những gì bọn em đã tìm hiểu được:

- Tiếp thu được nhiều kiến thức về C#, .NET, Window Form.
- Set up và quản lý các thông tin trên Firebase, cũng như truy xuất, thêm vào và cập nhật các thông tin.
- Chức năng chat giữa các người dùng, gửi tin nhắn qua lại với tin nhắn được lưu lại trong Firebase.
- Ứng dụng đáp ứng đủ các chức năng thêm và xóa các nhân viên, cập nhật thông tin của các trưởng phòng, ...

2. Ưu điểm:

- Dễ dàng sử dụng, thân thiện với người dùng.
- Đáp ứng nhu cầu của một chương trình quản lý nhân viên, với các chức năng cần thiết.
- Các chức năng rõ ràng, giúp người dùng biết cần phải nhập thông tin gì vào.
- Tốc độ truy xuất nhanh khi sử dụng Firebase.

3. Nhược điểm:

- Hơi ít tính năng, thời gian response của app hơi chậm.
- Giao diện chưa được bắt mắt.

4. Hướng phát triển:

- Sử dụng webserver khác hoặc tự build để tăng tốc độ response của server.
- Trạng thái online/offline: Hiện thị trạng thái online/offline của nhân viên để dễ dàng nhận biết sự sẵn sàng của người nhận.
- Phát triển nhóm chat: Hỗ trợ chat nhóm để các phòng ban hoặc nhóm dự án có thể trao đổi thông tin nhanh chóng.
- Thông báo: Tích hợp hệ thống thông báo (notification) để nhân viên không bỏ lỡ các thông tin quan trọng.