

**ĐẠI HỌC CẦN THƠ**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**NIÊN LUẬN CHUYÊN NGÀNH**  
**CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI**  
**XÂY DỰNG ỨNG DỤNG TÌM ĐƯỜNG ĐI TRONG**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN**  
**THÔNG**

**Giảng viên hướng dẫn:**  
**ThS. Trần Minh Tân**

**Sinh viên thực hiện:**  
**Họ tên: Nguyễn Quốc Trạng**  
**MSSV: B1910467**  
**Khóa: 45**

Cần Thơ, 11/2022

## LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn Trường Công nghệ Thông tin & Truyền Thông đã tạo điều kiện tốt nhất cho em học tập. Đặc biệt, em xin cảm ơn Thầy, Cô đã dạy dỗ, truyền đạt những kiến thức quý báu, những lời khuyên chân thành trong quá trình giảng dạy em. Trong quá học em đã có nhiều kiến thức bổ ích, đây chắc chắn là những kiến thức quý báu và là hành trang để em có thể vững bước trên con đường học tập sắp tới.

Em xin cảm ơn Thầy Trần Minh Tân đã giúp đỡ, hướng dẫn em trong suốt thời gian em làm Niên luận để đạt được kết quả tốt nhất có thể. Do vốn kiến thức hạn chế, mặc dù em đã cố gắng nhưng chắc chắn bài niên luận khó tránh khỏi sai sót và nhiều chỗ chưa chính xác, kính mong thầy xem xét và góp ý để em rút kinh nghiệm, hoàn thiện hơn trong những bài sắp tới.

Em xin chân thành cảm ơn!

[illegible]

**Giáo viên hướng dẫn**

**ThS. Trần Minh Tân**

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>1</b>
<b>NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN.....</b>	<b>2</b>
<b>MỤC LỤC .....</b>	<b>3</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>4</b>
<b>DANH MỤC BẢNG .....</b>	<b>5</b>
<b>DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT.....</b>	<b>6</b>
<b>PHẦN TỔNG QUAN .....</b>	<b>7</b>
1. GIỚI THIỆU ĐỀ TÀI.....	7
2. MỤC TIÊU ĐỀ TÀI.....	7
3. HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN .....	7
3.1. Hướng giải quyết.....	7
3.2. Kế hoạch thực hiện.....	7
<b>PHẦN NỘI DUNG .....</b>	<b>8</b>
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....</b>	<b>8</b>
1. CÁC CÔNG NGHỆ, THUẬT TOÁN SỬ DỤNG TRONG ĐỀ TÀI.....	8
2. MÔ TẢ CHỨC NĂNG .....	8
<b>CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU .....</b>	<b>9</b>
1. KẾT QUẢ NGHIÊN CỨU VÀ VẬN DỤNG THUẬT TOÁN .....	9
1.1. Sơ đồ khối xử lý .....	9
1.2. Sơ đồ khoa .....	10
1.3. Các mô hình dữ liệu sử dụng trong thuật toán.....	12
1.4. Các hàm, thư viện sử dụng trong thuật toán.....	12
1.4.1. Hàm dijkstra .....	12
1.4.2. Hàm nameToCode.....	12
1.4.3. Hàm codeToName.....	13
1.4.4. Hàm searchRoom .....	13
1.4.5. Hàm distance .....	13
1.4.6. Hàm testTwoPoint.....	14
1.4.7. Hàm getPoint.....	14
1.4.8. Hàm testThreePoint.....	15
1.4.9. Hàm route.....	17
2. THIẾT KẾ GIAO DIỆN ỨNG DỤNG .....	18
2.1. Giao diện trang chào mừng .....	18
2.2. Giao diện trang tìm kiếm đường đi.....	19
2.3. Giao diện trang kết quả tìm kiếm .....	20
<b>CHƯƠNG 3: KIỂM THỬ .....</b>	<b>24</b>
3. MỤC TIÊU KIỂM THỬ .....	24
4. KỊCH BẢN KIỂM THỬ .....	24
5. KẾT QUẢ KIỂM THỬ .....	25
<b>CHƯƠNG 4: KẾT LUẬN – ĐÁNH GIÁ.....</b>	<b>26</b>
6. KẾT QUẢ ĐẠT ĐƯỢC.....	26
7. HẠN CHẾ, NGUYÊN NHÂN .....	26
8. HƯỚNG PHÁT TRIỂN .....	26
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>27</b>

## DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ khối xử lý của ứng dụng .....	9
Hình 2: Sơ đồ tầng trật của trường Công nghệ Thông Tin và Truyền Thông.....	10
Hình 3: Sơ đồ tầng lầu của trường Công nghệ Thông Tin và Truyền Thông .....	11
Hình 4: Ví dụ chạy thuật toán bằng sơ đồ .....	17
Hình 5: Giao diện trang chào mừng ứng dụng .....	18
Hình 6: Giao diện trang tìm kiếm đường đi .....	19
Hình 7: Giao diện trang kết quả tìm kiếm qua hai tầng (a) .....	20
Hình 8: Giao diện trang kết quả tìm kiếm qua hai tầng (b).....	21
Hình 9: Giao diện trang kết quả tìm kiếm qua hai tầng (c) .....	22
Hình 10: Giao diện trang kết quả tìm kiếm trong một tầng .....	23

## DANH MỤC BẢNG

Bảng 1: Bảng mô tả hàm dijkstra .....	12
Bảng 2: Bảng mô tả hàm nameToCode.....	12
Bảng 3: Bảng mô tả hàm codeToName.....	13
Bảng 4: Bảng mô tả hàm searchRoom .....	13
Bảng 5: Bảng mô tả hàm distance .....	13
Bảng 6: Bảng mô tả hàm testTwoPoint.....	14
Bảng 7: Bảng mô tả hàm getPoint.....	14
Bảng 8: Bảng mô tả hàm testThreePoint.....	15
Bảng 9: Bảng mô tả hàm route .....	17
Bảng 10: Bảng kịch bản kiểm thử chức năng.....	24
Bảng 11: Bảng kết quả kiểm thử chức năng.....	25

**DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

Từ viết tắt	Tên đầy đủ	Mô tả
SDK	Software Development Kit	Là các công cụ và phần mềm dùng để phát triển ứng dụng thông qua một nền tảng nhất định
GPS	Global Positioning System	Là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo

## PHẦN TỔNG QUAN

### 1. GIỚI THIỆU ĐỀ TÀI

Nhằm tăng cường chuyển đổi số để phát triển Trường Đại học Cần Thơ theo hướng đại học thông minh. Thêm vào đó, nhận thấy các bạn sinh viên từ các khoa, trường khác khi vào Trường Công nghệ Thông Tin & Truyền Thông vẫn còn lúng túng trong việc tìm phòng học, thực hành. Các bạn sinh viên khóa mới của trường vẫn chưa quen với các phòng trong trường.

Nhận thấy những vấn đề trên, “Ứng dụng tìm đường trong Trường Công nghệ Thông Tin & Truyền Thông” được ra đời nhằm giúp các bạn có thể dễ dàng tìm kiếm đường đi đến phòng mong muốn. Nhằm một phần giúp Trường Đại học Cần Thơ tăng cường chuyển đổi số.

### 2. MỤC TIÊU ĐỀ TÀI

Mục tiêu chính của đề tài là xây dựng được ứng dụng tìm đường đi đến vị trí phòng học trong Trường Công nghệ Thông Tin & Truyền Thông, nhằm đáp ứng nhu cầu tìm đường đi một cách nhanh chóng.

### 3. HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN

#### 3.1. Hướng giải quyết

- Nghiên cứu công nghệ đa nền tảng Flutter và môi trường phát triển công nghệ này.
- Xây dựng giao diện cho Ứng dụng bằng Flutter.
- Nghiên cứu được thuật toán chỉ đường trong một khu vực.

#### 3.2. Kế hoạch thực hiện

**Tuần 1 – 10:** Tìm hiểu về đề tài, tìm hiểu về Flutter, Dart.

**Tuần 10 - 15:** Thiết kế và cài đặt ứng dụng.

**Tuần 16:** Kiểm thử, sửa lỗi và hoàn thành báo cáo.



## PHẦN NỘI DUNG

### CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

#### 1. CÁC CÔNG NGHỆ, THUẬT TOÁN SỬ DỤNG TRONG ĐỀ TÀI

- **Flutter:** Flutter là một SDK phát triển ứng dụng di động nguồn mở được tạo ra bởi Google. Nó được sử dụng để phát triển ứng dụng cho Android và iOS, cũng là phương thức chính để tạo ứng dụng cho Google Fuchsia.

- **Dart:** Dart là một ngôn ngữ lập trình web do Google phát triển. Nó được chính thức công bố tại Hội thảo GOTO tại Wayback Machine ngày 10-12 tháng 10 năm 2011 tại Aarhus. Mục đích của Dart không phải để thay thế JavaScript như là ngôn ngữ kịch bản chính bên trong trình duyệt web, mà là cung cấp sự lựa chọn hiện đại hơn.

- **Dijkstra:** Thuật toán Dijkstra, mang tên của nhà khoa học máy tính người Hà Lan Edsger Dijkstra vào năm 1956 và ấn bản năm 1959, là một thuật toán giải quyết bài toán đường đi ngắn nhất từ một đỉnh đến các đỉnh còn lại của đồ thị có hướng không có cạnh mang trọng số không âm. Thuật toán thường được sử dụng trong định tuyến với một chương trình con trong các thuật toán đồ thị hay trong công nghệ Hệ thống định vị toàn cầu (GPS).

#### 2. MÔ TẢ CHỨC NĂNG

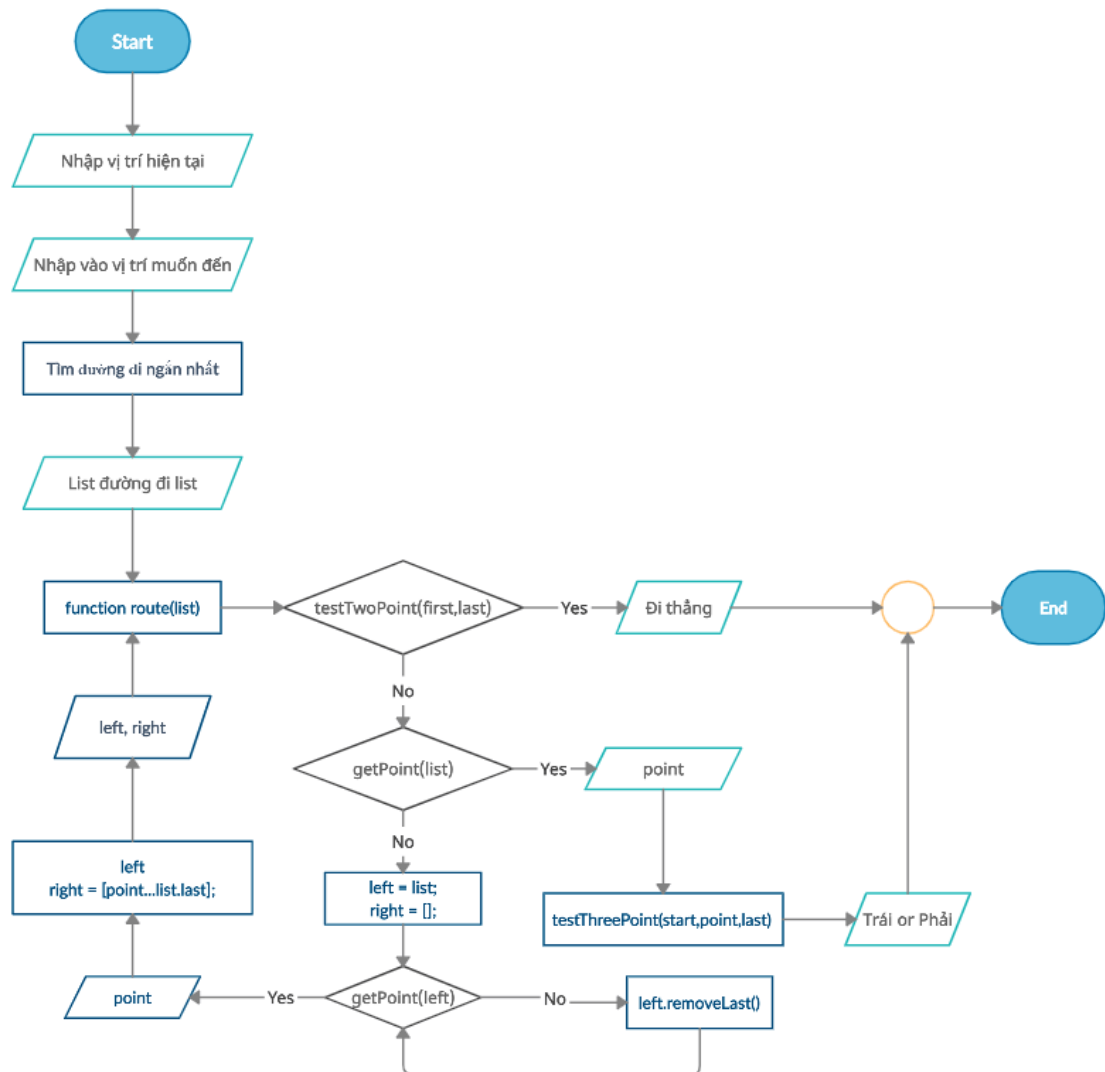
**Chức năng tìm đường đi:** Người dùng nhập vào vị trí hiện tại đang đứng trong trường và vị trí muốn đến trong trường. Ứng dụng sẽ hiển thị ra giao diện chỉ đường và hướng dẫn đường đi.

## CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU

### 1. KẾT QUẢ NGHIÊN CỨU VÀ VẬN DỤNG THUẬT TOÁN

#### 1.1. Sơ đồ khối xử lý

- Sơ đồ mô tả quá trình các bước xử lý của ứng dụng và thuật toán

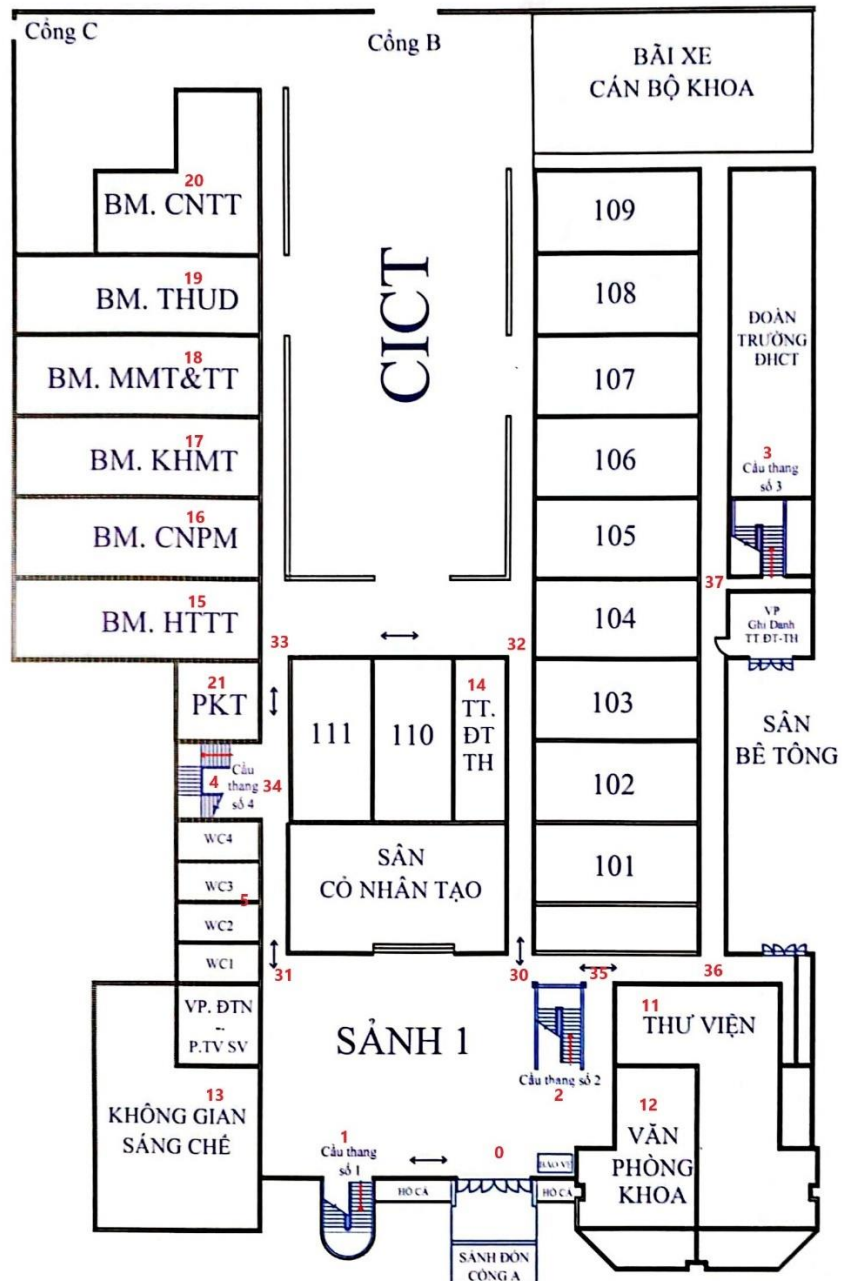


Hình 1: Sơ đồ khối xử lý của ứng dụng

## 1.2. Sơ đồ khoa

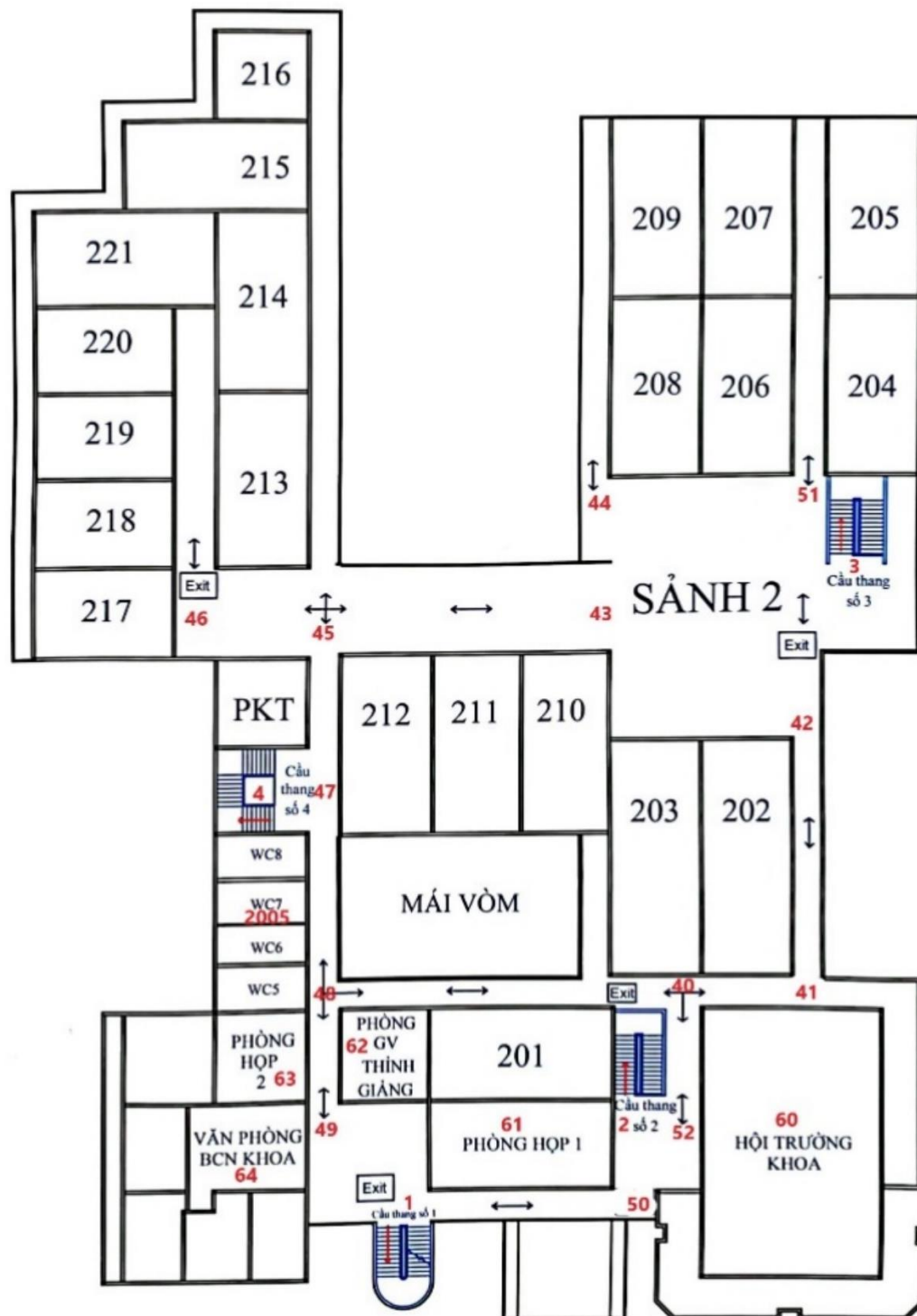
- Sử dụng 2 ảnh sơ đồ khoa;
- **Số đánh dấu trên mỗi phòng học** là những điểm để set tọa độ tìm đường đi cho thuật toán.

# TẦNG TRỆT



Hình 2: Sơ đồ tầng trệt của trường Công nghệ Thông Tin và Truyền Thông

# SƠ ĐỒ TẦNG LẦU



Hình 3: Sơ đồ tầng lầu của trường Công nghệ Thông Tin và Truyền Thông

### 1.3. Các mô hình dữ liệu sử dụng trong thuật toán

- Bằng cách đánh số các phòng, các ngã rẽ, các địa điểm quan trọng trong trường như hình trên. Sau đó đo đạc khoảng cách giữa các điểm đã đánh dấu và ghi nhận lại vào Map\_graph.
- Map\_graph chứa mỗi điểm và khoảng cách giữa chúng để tìm đường đi ngắn nhất từ một điểm đến các điểm bất kỳ.
- Map\_offset\_1 chứa tọa độ của các phòng và các điểm nối ở tầng trệt dựa trên sơ đồ phòng học ở trên.
- Map\_offset\_2 chứa tọa độ của các phòng và các điểm nối ở tầng 2 dựa trên sơ đồ phòng học ở trên.

### 1.4. Các hàm, thư viện sử dụng trong thuật toán

#### 1.4.1. Hàm dijkstra

**Bảng 1: Bảng mô tả hàm dijkstra**

Input	int from (default = 0)
	int to (required)
Output	List
Mô tả	Hàm sử dụng thuật toán dijkstra để tìm đường đi ngắn nhất từ một phòng bất kỳ đến một phòng bất kỳ khác dựa trên Map_graph

#### 1.4.2. Hàm nameToCode

**Bảng 2: Bảng mô tả hàm nameToCode**

Input	String title
Output	int
Mô tả	Hàm nhận tham số đầu vào là một chuỗi chứa tên của phòng mà người dùng tìm kiếm và trả về mã số của phòng đó để sử dụng trong thuật toán.

### 1.4.3. Hàm *codeToName*

**Bảng 3: Bảng mô tả hàm *codeToName***

Input	int code
Output	String
Mô tả	Hàm nhận tham số đầu vào là một số nguyên chứa mã của phòng và trả về tên phòng đó để hiển thị ra màn hình

### 1.4.4. Hàm *searchRoom*

**Bảng 4: Bảng mô tả hàm *searchRoom***

Input	String from (default = '0')
	String to (required)
Output	List
Mô tả	Hàm sử dụng thư viện <i>TiengViet</i> để xóa dấu tiếng việt cho 2 chuỗi truyền vào. Sau đó chuyển từ chuỗi đã xóa dấu sang mã phòng và tìm đường đi ngắn nhất từ <i>from</i> đến <i>to</i>

### 1.4.5. Hàm *distance*

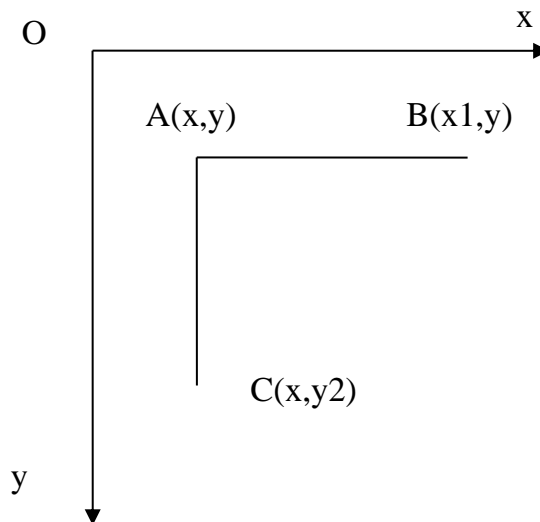
**Bảng 5: Bảng mô tả hàm *distance***

Input	Offset a
	Offset b
Output	Double
Mô tả	Hàm nhận vào tọa độ hai điểm và trả ra khoảng cách của hai điểm đó

#### 1.4.6. Hàm testTwoPoint

**Bảng 6: Bảng mô tả hàm testTwoPoint**

Input	int first – vị trí điểm đầu tiên trong Map
	int last – vị trí điểm thứ hai trong Map
	Map offset – Map chứa tọa độ của 2 điểm hiện tại
Output	List? – Chứa hướng dẫn đường đi hoặc null
Mô tả	<p>Hàm nhận vào vị trí 2 điểm, sau đó dựa vào Map offset lấy tọa độ hai điểm đó ra.</p> <p>Kiểm tra hai điểm có nằm trên 1 đường thẳng hay không:</p> <ul style="list-style-type: none"> <li>- Nếu hai điểm có cùng tọa độ Ox và khác tọa độ Oy</li> <li>- Nếu hai điểm có cùng tọa độ Oy và khác tọa độ Ox</li> <li>- Nếu hai điểm nằm cạnh nhau trong sơ đồ phòng</li> </ul>



#### 1.4.7. Hàm getPoint

**Bảng 7: Bảng mô tả hàm getPoint**

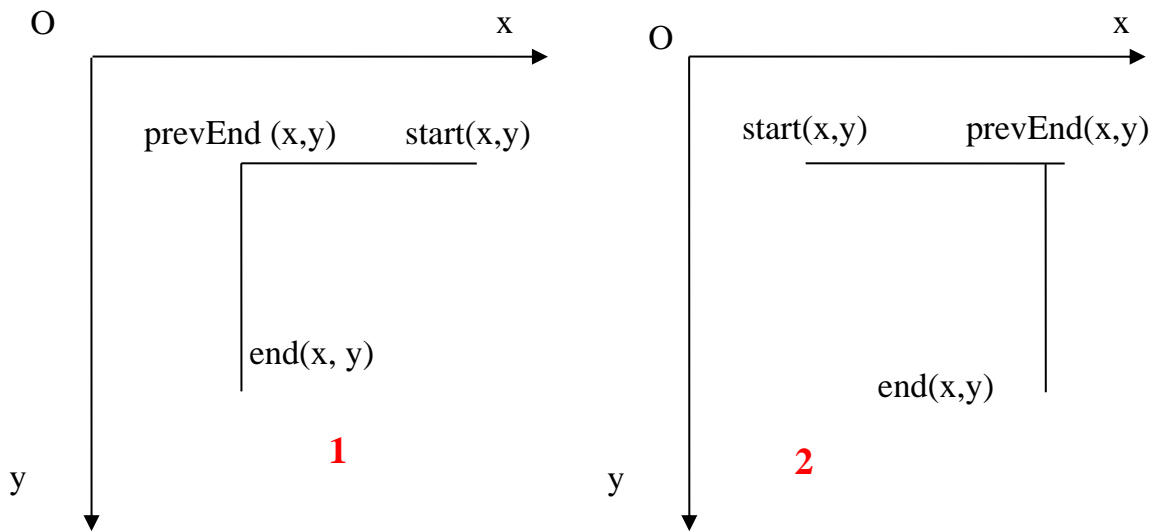
Input	List listPoint – chứa danh sách các điểm trên đường đi ngắn nhất
	Map offset – Map chứa tọa độ của 2 điểm hiện tại
Output	int? – Chứa điểm mà tại đó rẽ phải hay rẽ trái hoặc null
Mô tả	Hàm nhận vào một List các điểm, thực hiện kiểm tra xem trong danh sách đường đi có tồn tại một điểm mà tại đó chỉ rẽ một lần là tới đích

#### 1.4.8. Hàm testThreePoint

**Bảng 8: Bảng mô tả hàm testThreePoint**

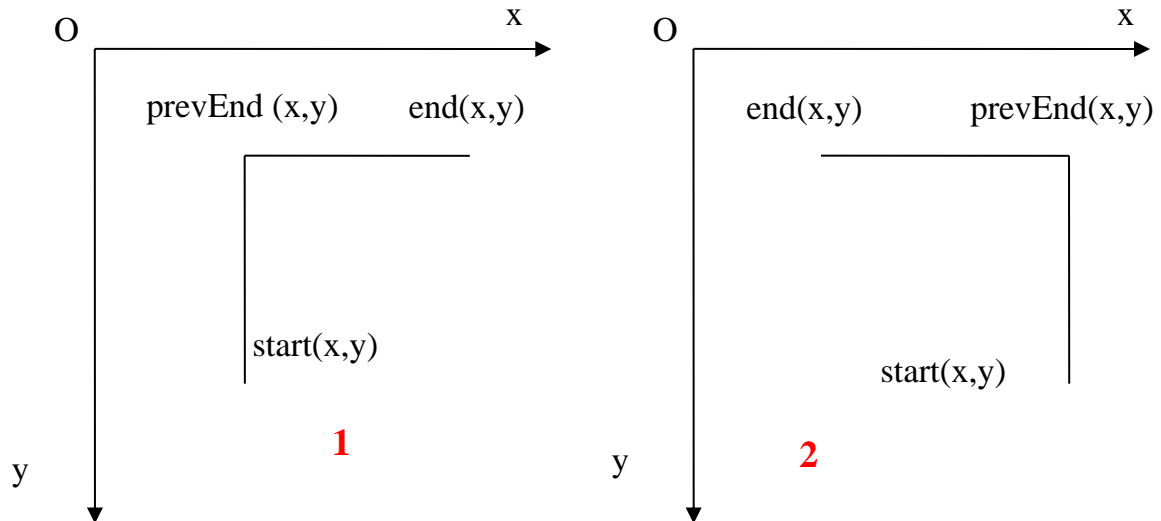
Input	int first
	int mid
	int last
	Map offset – Map chứa tọa độ của 3 điểm hiện tại
Output	List? – Chứa hướng dẫn đường đi hoặc null
Mô tả	Trả về hướng dẫn rẽ phải hoặc trái

- Giải thích hàm testThreePoint:
- Kiểm tra nếu điểm đầu và điểm giữa có nằm trên 1 đường thẳng hay không.
- Nếu không thì dừng lại, nếu có thì tiếp tục:



- Nếu hai điểm start và prevEnd có cùng tọa độ y và hai điểm prevEnd và end có cùng tọa độ x hoặc hai điểm start và prevEnd cùng tọa độ và hai điểm prevEnd và end nằm trên cùng một đường thẳng thì:
  - Nếu tọa độ x của start lớn hơn tọa độ x của prevEnd ta sẽ được như hình 1 ở trên. Sau đó so sánh tọa độ y của prevEnd và end để xác định rẽ phải hay rẽ trái để đến được end.
  - Nếu tọa độ x của start nhỏ hơn tọa độ x của prevEnd ta sẽ được như hình 2 ở trên. Sau đó so sánh tọa độ y của prevEnd và end để xác định rẽ phải hay rẽ trái để đến được end.

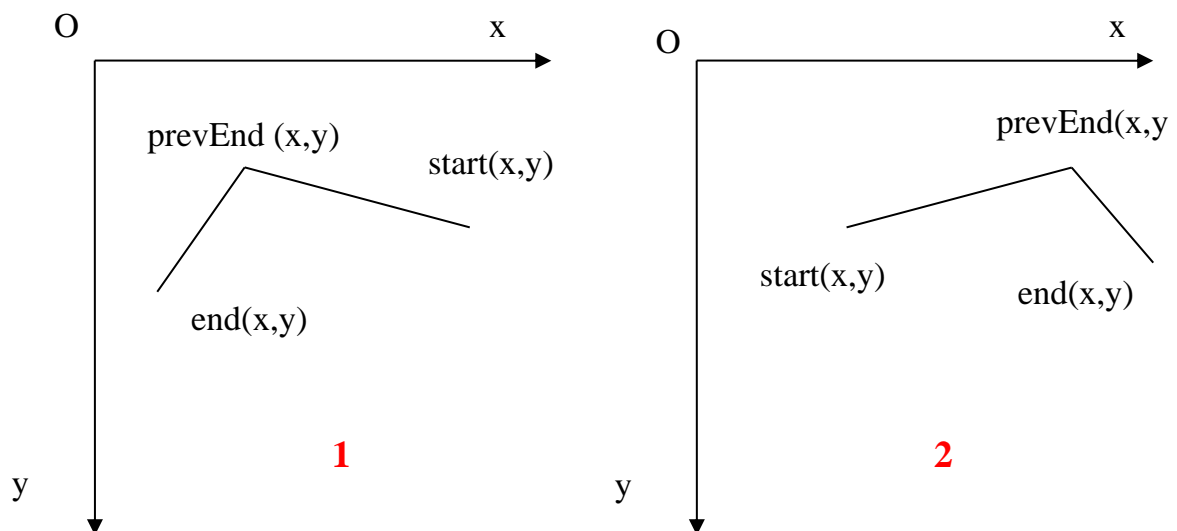




- Nếu hai điểm  $start$  và  $prevEnd$  có cùng tọa độ  $x$  và hai điểm  $prevEnd$  và  $end$  có cùng tọa độ  $y$  hoặc hai điểm  $start$  và  $prevEnd$  cùng tọa độ và hai điểm  $prevEnd$  và  $end$  nằm trên cùng một đường thẳng thì:

- Nếu tọa độ  $y$  của  $start$  nhỏ hơn tọa độ  $y$  của  $prevEnd$  ta sẽ được như hình 1 ở trên. Sau đó so sánh tọa độ  $x$  của  $prevEnd$  và  $end$  để xác định rẽ phải hay rẽ trái để đến được  $end$ .

- Nếu tọa độ  $y$  của  $start$  lớn hơn tọa độ  $y$  của  $prevEnd$  ta sẽ được như hình 2 ở trên. Sau đó so sánh tọa độ  $x$  của  $prevEnd$  và  $end$  để xác định rẽ phải hay rẽ trái để đến được  $end$ .



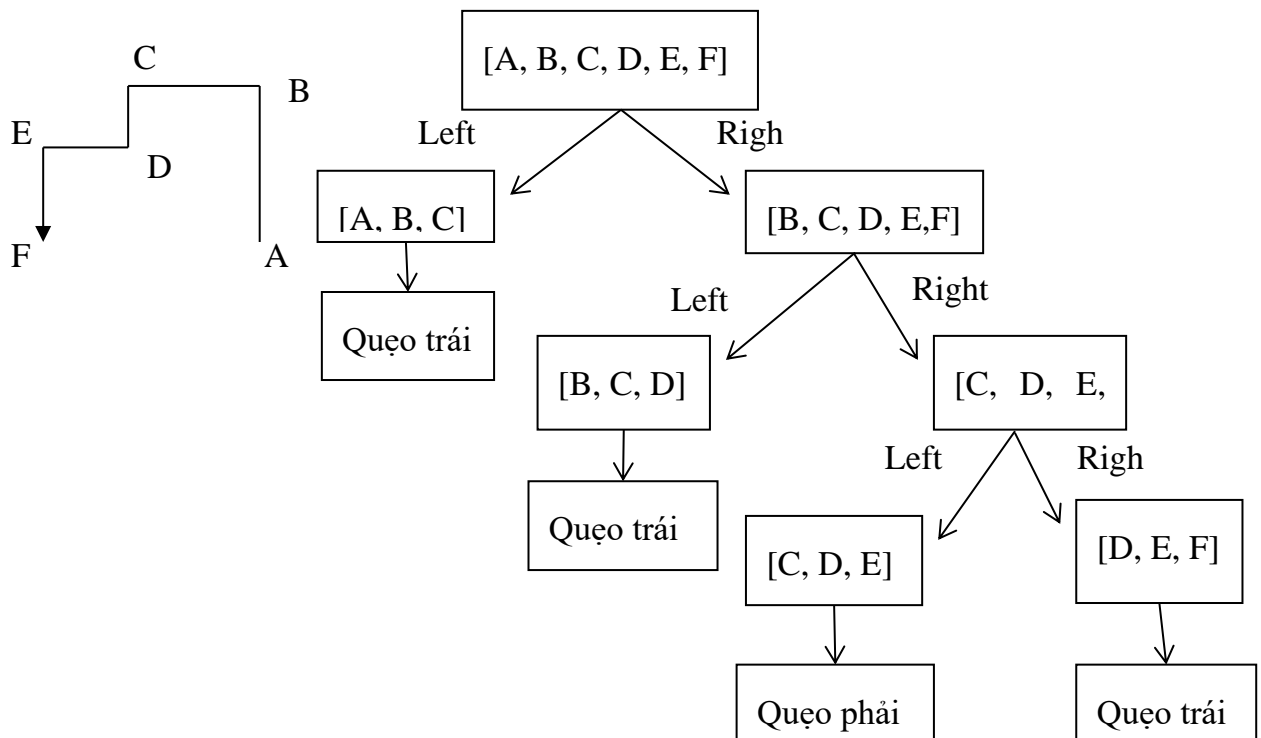
- Nếu hai điểm  $prevEnd$  và  $end$  nằm trên cùng 1 đường thẳng
- Nếu tọa độ  $x$  của  $start$  lớn hơn tọa độ  $x$  của  $prevEnd$  sẽ tương tự như hình 1 ở trên, sau đó dựa trên tọa độ  $y$  của  $prevEnd$  và  $end$  để xác định sẽ rẽ trái hay phải
- Nếu tọa độ  $x$  của  $start$  nhỏ hơn tọa độ  $x$  của  $prevEnd$  sẽ tương tự như hình 2 ở trên, sau đó dựa trên tọa độ  $y$  của  $prevEnd$  và  $end$  để xác định sẽ rẽ trái hay phải

#### 1.4.9. Hàm route

**Bảng 9: Bảng mô tả hàm route**

Input	List listDijkstra – chứa danh sách các điểm trên đường đi ngắn nhất
	Map offset – Map chứa tọa độ hiện tại
Output	List – Chứa thông tin hướng dẫn đường đi
Mô tả	Trả về thông tin hướng dẫn đường đi dựa trên đường đi ngắn nhất hiện tại

- Sử dụng hàm testTwoPoint để kiểm tra điểm đầu và điểm cuối của danh sách đường đi ngắn nhất có nằm trên 1 đường thẳng đi được hay không, nếu được trả về đi thẳng.
- Ngược lại, sử dụng hàm getPoint để kiểm tra xem danh sách đường đi có tồn tại một điểm mà tại điểm đó chỉ cần rẽ một lần là tới được điểm đích mong muốn.
- Nếu có thì sử dụng hàm testThreePoint để lấy ra hướng dẫn là rẽ phải hay trái.
- Ngược lại nếu không tồn tại điểm đó, chia danh sách đường đi ra làm 2 mảng left và right:
- Mảng left chứa các điểm bằng cách giảm các phần tử của mảng ban đầu đến khi nào getPoint có giá trị. Thực hiện tìm đường mảng left bằng hàm route
- Mảng right chứa các điểm từ điểm getPoint trả lại đến hết danh sách ban đầu. Thực hiện tìm đường mảng right bằng hàm route



**Hình 4: Ví dụ chạy thuật toán bằng sơ đồ**

## 2. THIẾT KẾ GIAO DIỆN ỨNG DỤNG

### 2.1. Giao diện trang chào mừng

**Mục đích:** hiển thị giao diện trang chào mừng khi người dùng mở ứng dụng.

**Giao diện:**



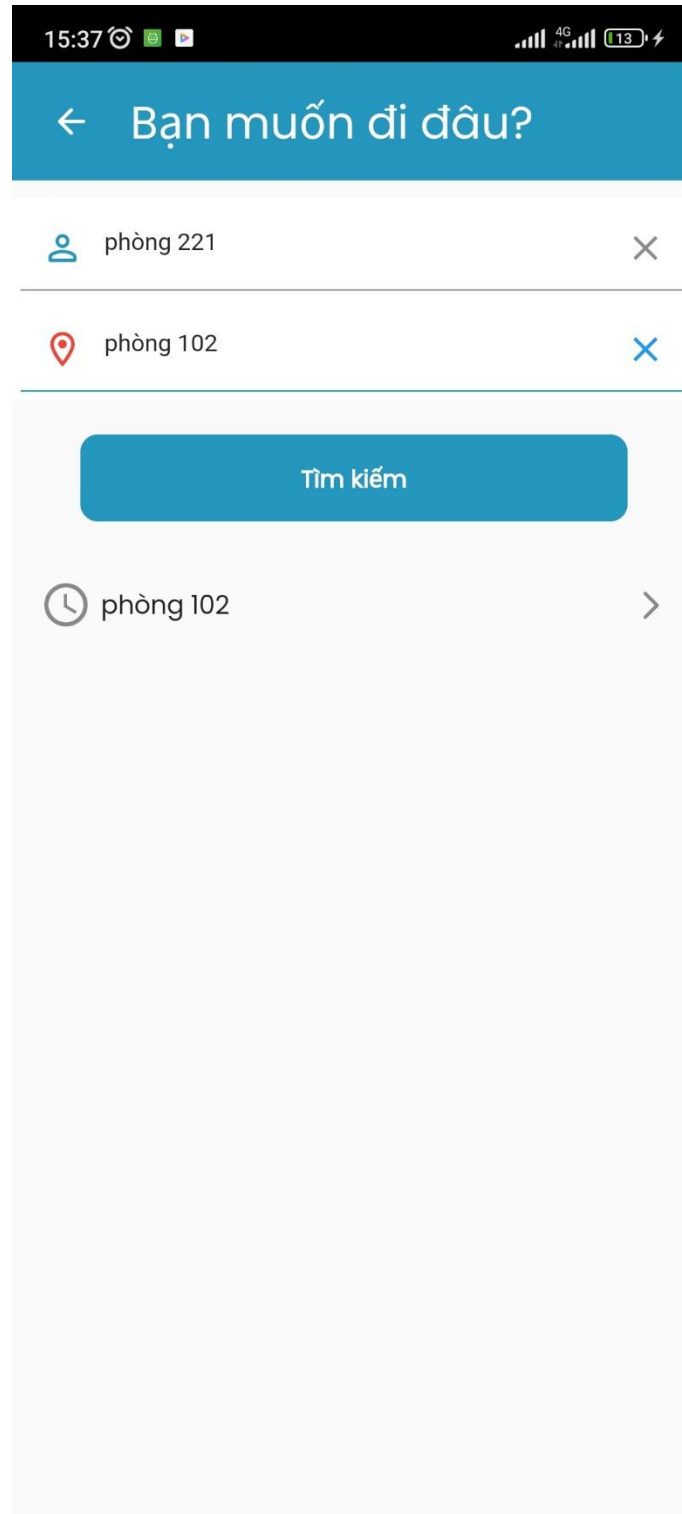
*Hình 5: Giao diện trang chào mừng ứng dụng*

## 2.2. Giao diện trang tìm kiếm đường đi

**Mục đích:** Cho phép người dùng nhập vào vị trí hiện tại và vị trí muốn đi đến để tìm kiếm.

**Mô tả:** Khi người dùng nhập dữ liệu vào ô tìm kiếm và nhấn tìm kiếm thì dữ liệu sẽ được chuyển qua trang kết quả để xử lý và hiển thị ra.

**Giao diện:**



The screenshot displays a mobile application interface for route finding. At the top, the status bar shows the time 15:37, signal strength, 4G network, and a battery level of 13%. The app's title bar is blue with a white back arrow and the text "Bạn muốn đi đâu?". Below the title bar, there are two input fields. The first field has a person icon and the text "phòng 221", with a close button (X) to its right. The second field has a location pin icon and the text "phòng 102", also with a close button (X) to its right. Below these fields is a large blue button labeled "Tìm kiếm". At the bottom, there is a list item with a clock icon, the text "phòng 102", and a right-pointing arrow (>).

**Hình 6:** Giao diện trang tìm kiếm đường đi

### 2.3. Giao diện trang kết quả tìm kiếm

**Mục đích:** Hiển thị giao diện đường đi và hướng dẫn đường đi cho người dùng xem

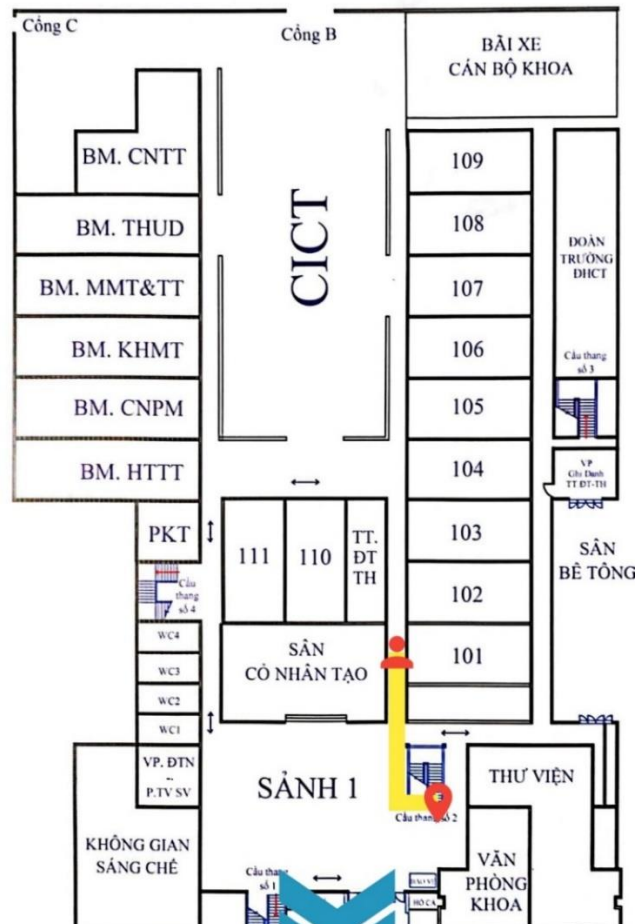
**Mô tả:** Nhận dữ liệu từ trang tìm kiếm và xử lý để hiển thị hướng dẫn đường đi cho người dùng. Sử dụng Map tọa độ để vẽ đường đi và sử dụng hàm route để hiển thị hướng dẫn đường đi.

- Khi tìm kiếm các phòng giữa hai tầng khác nhau, màn hình sẽ hiển thị hai hình ảnh của hai tầng và chứa hướng dẫn đường đi cụ thể giữa các tầng

**Giao diện:**



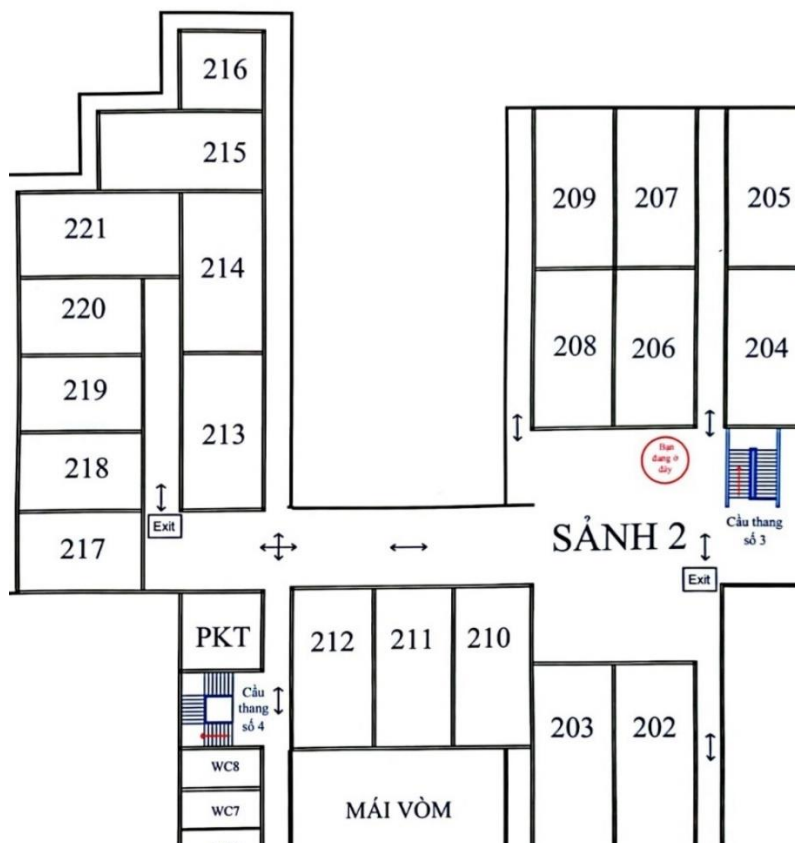
### TẦNG TRỆT



Hình 7: Giao diện trang kết quả tìm kiếm qua hai tầng (a)



## SƠ ĐỒ TẦNG LẦU



Hình 8: Giao diện trang kết quả tìm kiếm qua hai tầng (b)



## SƠ ĐỒ TẦNG LẦU

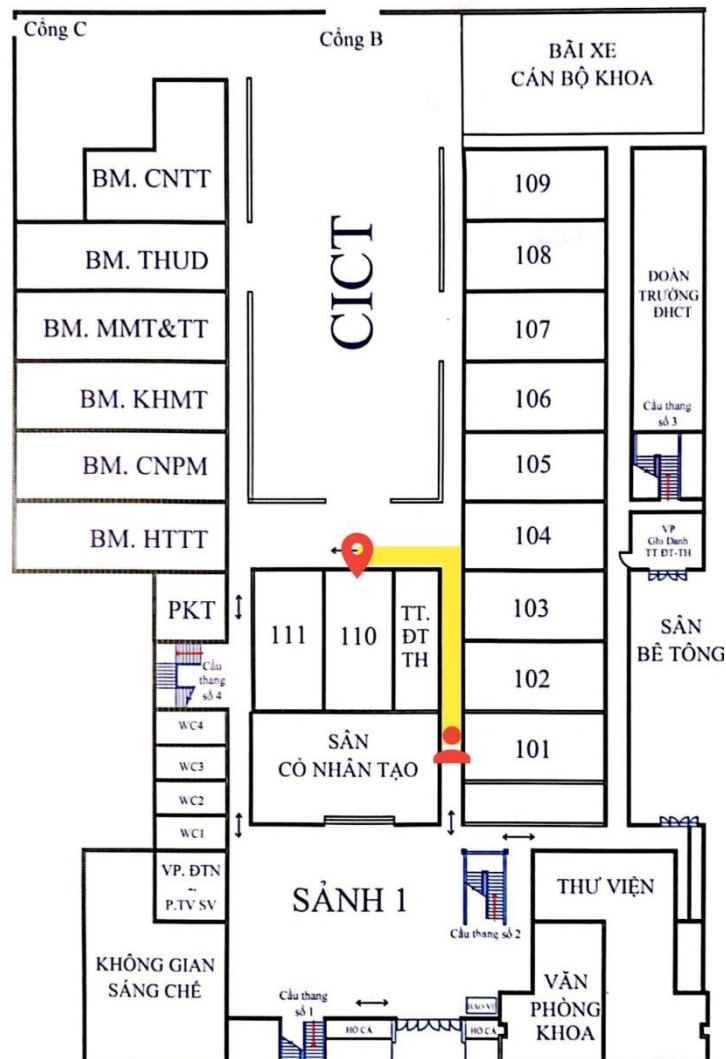


Hình 9: Giao diện trang kết quả tìm kiếm qua hai tầng (c)

- Khi tìm kiếm trong 1 tầng, trang sẽ hiển thị một hình ảnh và hướng dẫn đường đi đến đó.



## TẦNG TRỆT



Hình 10: Giao diện trang kết quả tìm kiếm trong một tầng



## CHƯƠNG 3: KIỂM THỬ

### 3. MỤC TIÊU KIỂM THỬ

- Tiến hành thực hiện kiểm thử nhằm để kiểm tra độ hoàn thiện của chức năng, tìm ra các lỗi phát sinh trong quá trình sử dụng và khắc phục sớm nhất có thể
- Mục tiêu của kiểm thử bao gồm:
  - o Tìm các lỗi phát sinh do người lập trình tạo ra.
  - o Đảm bảo kết quả cuối cùng đáp ứng các yêu cầu người dùng.
  - o Tuân theo kịch bản kiểm thử.

### 4. KỊCH BẢN KIỂM THỬ

**Bảng 10: Bảng kịch bản kiểm thử chức năng**

STT	Trường hợp	Ngày kiểm thử
1	Từ sảnh đến phòng 110	26/11/2022
2	Từ thư viện đến khoa Công nghệ thông tin	26/11/2022
3	Từ phòng 201 đến phòng 221	26/11/2022
4	Từ phòng 221 đến phòng 207	26/11/2022
5	Từ phòng 110 đến phòng 201	26/11/2022
6	Từ phòng 109 đến phòng 209	26/11/2022
7	Từ phòng 203 đến nhà vệ sinh tầng 2	26/11/2022
8	Từ phòng 206 đến nhà vệ sinh tầng 1	26/11/2022
9	Từ phòng 203 đến cầu thang số 4	26/11/2022
10	Từ phòng 221 đến hội trường khoa	26/11/2022
11	Từ sảnh khoa đến phòng 207	26/11/2022
12	Từ khoa hệ thống thông tin đến phòng họp 1	26/11/2022

## 5. KẾT QUẢ KIỂM THỬ

**Bảng 11: Bảng kết quả kiểm thử chức năng**

STT	Trường hợp	Kết quả mong đợi	Kết quả thực tế	Kết quả
1	Từ sảnh đến phòng 110	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
2	Từ thư viện đến khoa Công nghệ thông tin	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
3	Từ phòng 201 đến phòng 221	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
4	Từ phòng 221 đến phòng 207	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
5	Từ phòng 110 đến phòng 201	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
6	Từ phòng 109 đến phòng 209	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
7	Từ phòng 203 đến nhà vệ sinh tầng 2	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
8	Từ phòng 206 đến nhà vệ sinh tầng 1	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
9	Từ phòng 203 đến cầu thang số 4	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
10	Từ phòng 221 đến hội trường khoa	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
11	Từ sảnh khoa đến phòng 207	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công
12	Từ khoa hệ thống thông tin đến phòng họp 1	Hiện thị đường đi và hướng dẫn	Hiện thị đường đi và hướng dẫn	Thành công

## **CHƯƠNG 4: KẾT LUẬN – ĐÁNH GIÁ**

### **1. KẾT QUẢ ĐẠT ĐƯỢC**

- Sau quá trình tìm hiểu về Flutter, đã có thể phát triển thành một ứng dụng hoàn chỉnh.
- Ứng dụng đã đạt được mục tiêu đề ra ban đầu.
- Một phần đáp ứng được lý do thực hiện đề tài đã nêu ra.

### **2. HẠN CHẾ, NGUYÊN NHÂN**

Vì thời gian có hạn, ứng dụng hiện tại vẫn chưa hoàn chỉnh ở các điểm sau:

- Chức năng còn tương đối ít và đơn giản.
- Thuật toán vẫn chưa được tối ưu để sử dụng trên phạm vi lớn hơn.
- Giao diện ứng dụng còn khá đơn giản.
- Vẫn còn một vài lỗi phát sinh.

### **3. HƯỚNG PHÁT TRIỂN**

- Để ứng dụng hoạt động tốt hơn cần có một hướng phát triển trong tương lai. Các mục tiêu được đề ra để cải thiện và phát triển ứng dụng:
  - Xây dựng ứng dụng kèm thêm các chức năng hiển thị các tin trong trường, Cho phép xem lịch thực hành của các phòng,...
  - Tối ưu hóa thuật toán để có thể phát triển ở phạm vi rộng hơn

## TÀI LIỆU THAM KHẢO

- [1]. *Flutter documentation*. Flutter. (n.d.). Retrieved November 01, 2022, from <https://docs.flutter.dev/>
- [2]. Bài giảng môn Phát triển ứng dụng di động, Thầy Bùi Võ Quốc Bảo