**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ- VIỄN THÔNG**

****

PBL1: LẬP TRÌNH

ĐỀ TÀI: QUẢN LÍ DỊCH VỤ LƯU TRÚ

GVHD: NGUYỄN DUY NHẬT VIỄN

SINH VIÊN THỰC HIỆN: TRẦN LÊ LONG VŨ 22KTMT1

MỤC LỤC

[MỤC LỤC 2](#_Toc73522281)

[MỞ ĐẦU 3](#_Toc73522283)

[1. TỔNG QUAN ĐỀ TÀI 4](#_Toc73522284)

[2. CƠ SỞ LÝ THUYẾT 4](#_Toc73522285)

[2.1. Ý tưởng 4](#_Toc73522286)

[2.2. Cơ sở lý thuyết 4](#_Toc73522287)

[3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 5](#_Toc73522288)

[3.1. Phát biểu bài toán 7](#_Toc73522289)

[3.2. Cấu trúc dữ liệu 8](#_Toc73522290)

[3.3. Thuật toán 9](#_Toc73522291)

[4. CHƯƠNG TRÌNH VÀ KẾT QUẢ 13](#_Toc73522292)

[4.1. Tổ chức chương trình 13](#_Toc73522293)

[4.2. Ngôn ngữ cài đặt 13](#_Toc73522294)

[4.3. Kết quả 14](#_Toc73522295)

[4.3.1. Giao diện chính của chương trình 14](#_Toc73522296)

[4.3.2. Kết quả thực thi của chương trình 14](#_Toc73522297)

[4.3.3. Nhận xét đánh giá 17](#_Toc73522298)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 17](#_Toc73522299)

[5.1. Kết luận 17](#_Toc73522300)

[5.2. Hướng phát triển 18](#_Toc73522301)

**Lời nói đầu:**

-Trong thời đại số hóa và công nghệ hiện nay, việc quản lí những cơ sở lưu trú là một thách thức với các người chủ. Từ việc ghi chép bằng tay và xử lí thông tin trên giấy, chuyển đổi sang sử dụng công nghệ thông tin để thu thập, lưu trữ và quản lí các cơ sở lưu trú một cách nhanh chóng và chính xác. Báo cáo này nhằm tập trung vào ứng dụng quản lí một dịch vụ lưu trú , một công cụ thông minh và tiện ích, đã mang lại những lợi ích vượt trội cho khách hàng và người quản lí. Trên cơ sở này, báo cáo sẽ giới thiệu và phân tích những ưu điểm và tiềm năng của ứng dụng quản lí dịch vụ lưu trú.

**Mục đích thực hiện :**

- Mục đích của đề tài này là tìm hiểu và đánh giá ứng dụng quản lí dịch vụ lưu trú, đồng thời phân tích những ưu điểm và tiềm năng của nó trong việc cải thiện quá trình quản lí và nâng cao chất lượng . Bằng cách nghiên cứu các hệ thống quản lí dịch vụ lưu trú hiện có và tìm hiểu về công nghệ ứng dụng, chúng ta sẽ có cái nhìn tổng quan về sự phát triển và tiềm năng của ứng dụng này trong lĩnh vực kinh doanh.

**Đối tượng và phương pháp nghiên cứu:**

-Để nghiên cứu về vấn đề ứng dụng quản lí dịch vụ lưu trú một cách toàn diện, chúng ta xác định đối tượng nghiên cứu là việc giúp người dùng quản lý một cách hiệu quả và nhanh chóng. Tạo ra một ứng dụng thuận tiện và thân thiện với người dùng, từ đó nâng cao hiệu suất và khả năng quản lý của chủ trong việc ghi nhận và xử lí thông tin khách hàng.

**1. TỔNG QUAN ĐỀ TÀI**

ĐỀ TÀI : QUẢN LÍ DỊCH VỤ LƯU TRÚ THEO TỪNG PHÒNG

Mô tả chung đề tài:

-Xây dựng một ứng dụng cho phép quản với các chức năng sau:

+ đặt phòng (điền thông tin)

+ hiển thị thông tin

+sửa thông tin

+ Thoát

- thông tin sẽ được lưu trên file

- Chức năng xem thông tin: Nhập vào số phòng và hiển thị thông tin của phòng đó

- Chức năng sửa thông tin : nhập vào số phòng và cho phép sửa thông tin của phòng đó

**2. CƠ SỞ LÝ THUYẾT**

2.1 ý tưởng

- Tạo ra sự tiện lợi và chính xác trong việc quản lý thông tin của khách hàng, giúp chủ theo dõi một cách dễ dàng hơn và nhanh chóng hơn.

- Giảm thiểu việc sai sót trong việc ghi nhận thông tin khách hàng, giúp đảm bảo tính chính xác và bảo mật.

- giúp người chủ dễ dàng hơn trong việc kiểm soát thông tin.

2.2 cơ sở lý thuyết

-sử dụng file để lưu thông tin của khách hàng

-sử dụng các thuật toán để thực hiện các chức năng như hiện thị thông tin, sửa thông tin.

**3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THỰC TOÁN**

3.1 phát biểu bài toán:

-chức năng đặt phòng : có thể nhập vào số phòng muốn ở và nhập vào thông tin cá nhân để xác nhận đặt phòng

-chức năng hiển thị thông tin: khi nhập vào số phòng thì sẽ hiện ra thông tin của khách hàng phòng đó.

-chức năng sửa thông tin : cho phép sủa thông tin của khách hàng khi nhập vào số phòng muốn sửa

3.2 cấu trúc dữ liệu:

* File được sử dụng để lưu trữ các dữ liệu cần thiết của chương trình để ngay cả khi ngừng chạy dữ liệu cũng không bị mất hoặc dùng để lưu trữ dữ liệu đầu vào lớn và sử dụng lại nó mỗi khi cần chạy chương trình.
* Có hai kiểu file là file văn bản (text files) và file nhị phân (binary files). Đồ án chủ yếu thực hiện thao tác với file nhị phân.

**-Định nghĩa :**

File nhị phân là file chứa nội dung không nhất thiết phải là ASCII text.

Xét ví dụ sau :

ofstream file("num.dat");   
short x = 1297;   
file << x; Dòng lệnh cuối cùng của ví dụ trên sẽ ghi nội dung của x vào file. Và chúng được lưu vào ở dạng kí tự '1', '2', '9', '7'.  
Thực sự là con số 1297 không hề được lưu trong bộ nhớ. Nó đã được định dạng thành 1 số nhị phân, và chiếm 2 byte trong bộ nhớ máy tính.  
Vì x kiểu short nó sẽ được lưu như sau :  
00000101 | 00010001  
Và đây chính là dữ liệu nguyên thuỷ được lưu trong bộ nhớ. Để làm được như vậy chúng ta sẽ có cú pháp như sau :

 file.open("stuff.dat", ios::out | ios::binary);

**-Vào/ ra trên file nhị phân**

Vào ra tệp tin nhị phân bằng: read và write

**+Write**  
-Hàm write dùng để ghi 1 file stream ở định dạng nhị nhận. Dạng tổng quát của hàm write như sau :

fileObject.write(address, size);

**Lưu ý :**  
  
-fileObject là tên của đối tượng file stream.  
-address là địa chỉ đầu tiên của 1 vùng nhớ được ghi vào file. **Đối số này có thể là địa chỉ của 1 kí tự hoặc là con trỏ tới kiểu char.**  
-size là số lượng byte của vùng nhớ mà nó được write. Đối số này bắt buộc phải là kiểu integer( số nguyên dương )  
Ví dụ :

char letter = 'A';   
file.write(&letter, sizeof(letter));

-Đối thứ nhất ở đây là địa chỉ của biến letter. Và đối này sẽ nói cho hàm write biết rằng dữ liệu được ghi vào file ở đâu trong vùng nhớ.   
-Đối thứ 2 sẽ là kíck thước của biến letter, và đối này sẽ nói cho hàm write biết số lượng byte của dữ liệu sẽ ghi vào file. Bởi vì dữ sẽ được lưu khác nhau trên tuỳ hệ thống khác nhau, nên cách tốt nhất là chúng ta dùng toán tử sizeof để quyết định số bytes được ghi. Và sau khi hàm này được thực hiện, nội dung của biến letter sẽ được khi vào file nhị phân của đối tượng "file"đó.  
  
Chúng ta xem tiếp 1 ví dụ khác

char data[] = {'A', 'B', 'C', 'D'};   
file.write(data, sizeof(data));

Trong ví dụ này thì đối thứ 1 là tên của mãng (data). Vì khi ta truyền tham số là tên của mãng thì tức là ta đã truyền con trỏ trỏ tới vị trí đầu tiên của mãng. Đối thứ 2 cũng có ý nghĩa tượng tự như ví dụ 1. Và sau khi gặp này thực hiện thì nội dung của mãng sẽ được ghi vào file nhị phân tương ứng với đối tượng file.  
**+Read:**  
Hàm read thì sẽ dùng đọc vào số dữ liệu nhị phân từ file vào bộ nhớ máy tính. Dạng tổng quát là :  
        fileObject.read(address, size);  
-Ở đây fileObject là tên của đối tượng file stream.  
-address là địa chỉ đầu tiên mà vùng nhớ mà dữ liệu được đọc vào được lưu. Và đối này có thể là địa chỉ của 1 kí tự hay 1 con trỏ tới kiểu char.  
-size cũng là số lượng byte trong bộ nhớ được đọc vào từ file. Và đối này bắt buộc cũng phải là số kiểu integer ( nguyên dương )  
Ví dụ :

char letter;   
file.read(&letter, sizeof(letter));

Hoặc :

char data[4];  
file.read(data, sizeof(data));

Chương trình sau sẽ minh hoạ cách sử dụng 2 hàm read and write :

// Ghi các nội dung của mảng vào tập tin.

cout << "Viết các ký tự để các tập tin. \ n";

file.write (dữ liệu, sizeof (dữ liệu));

// Đóng tệp.

file.close ();

// Mở file cho đầu vào trong chế độ nhị phân.

file.open ("test.dat", ios :: in | ios :: binary);

// Đọc nội dung của tập tin vào mảng.

cout << "Bây giờ đọc lại dữ liệu vào bộ nhớ. \ n";

file.read (dữ liệu, sizeof (dữ liệu));

// Hiển thị nội dung của mảng.

for (int count = 0; count <SIZE; đếm ++)

cout << dữ liệu [count] << "";

cout << endl;

// Đóng tệp.

file.close ();

return 0;

}

Nếu chúng ta muốn ghi các kiểu khác vào file nhị phân thì ta phải dùng cú pháp có 1 tí đặt biệt sau đây.

reinterpret\_cast<dataType>(value)

Ở cú pháp trên thì dataType sẽ là kiểu dữ liệu mà chúng ta muốn ép về, và value sẽ là giá trị mà chúng ta muốn ép.

Ví dụ :

int x = 65;   
file.write(reinterpret\_cast<char \*>(&x), sizeof(x));

Đối với mảng thì :

const int SIZE = 10;   
int numbers[SIZE] = {1,2,3,4,5,6,7,8,9,10};   
file.write(reinterpret\_cast<charr \*>(numbers), sizeof(numbers));

Ví dụ :

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

const int SIZE = 10;

fstream file;

int numbers[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

// Open the file for output in binary mode.

file.open("numbers.dat", ios::out | ios::binary);

// Write the contents of the array to the file.

cout << "Writing the data to the file.\n";

file.write(reinterpret\_cast<char \*>(numbers), sizeof(numbers));

// Close the file.

file.close();

// Open the file for input in binary mode.

file.open("numbers.dat", ios::in | ios::binary);

// Read the contents of the file into the array.

cout << "Now reading the data back into memory.\n";

file.read(reinterpret\_cast<char \*>(numbers), sizeof(numbers));

// Display the contents of the array.

for (int count = 0; count < SIZE; count++)

cout << numbers[count] << " ";

cout << endl;

// Close the file.

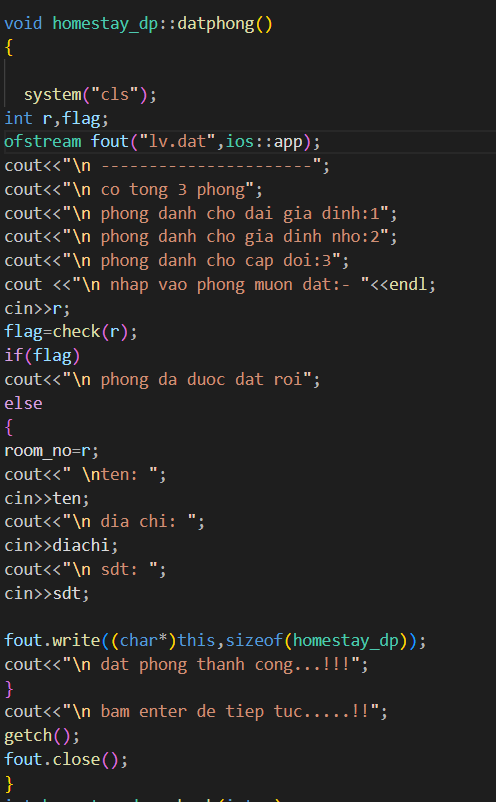
file.close();

return 0;

}

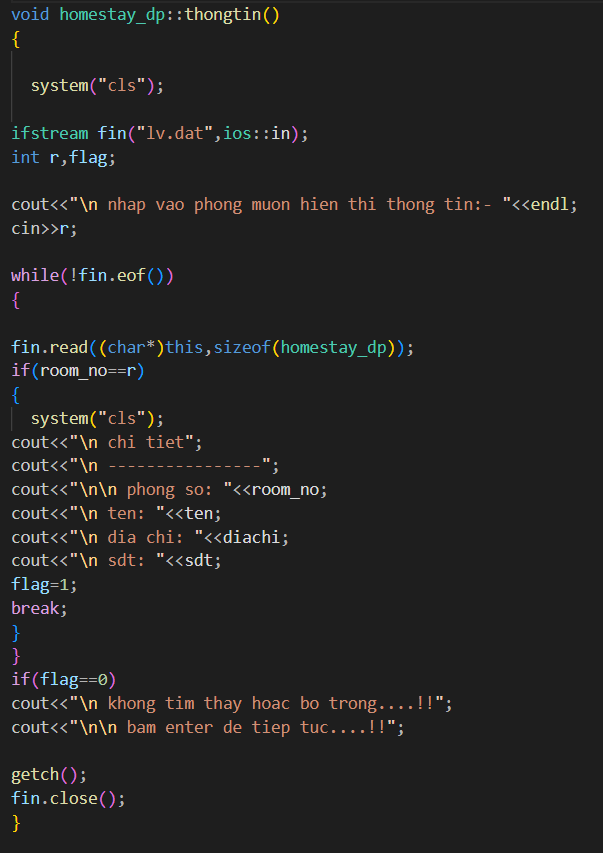
**3.3 THUẬT TOÁN**

-chức năng đặt phòng:

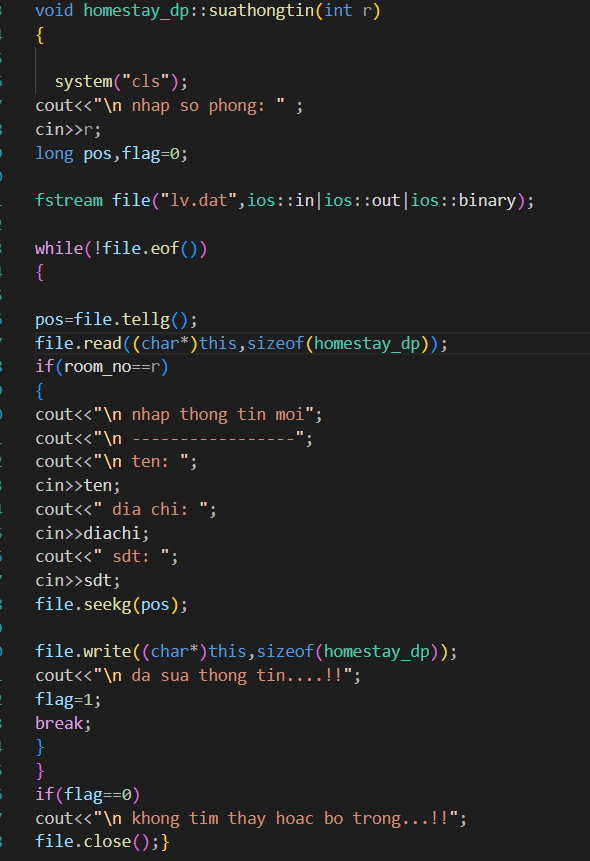


-

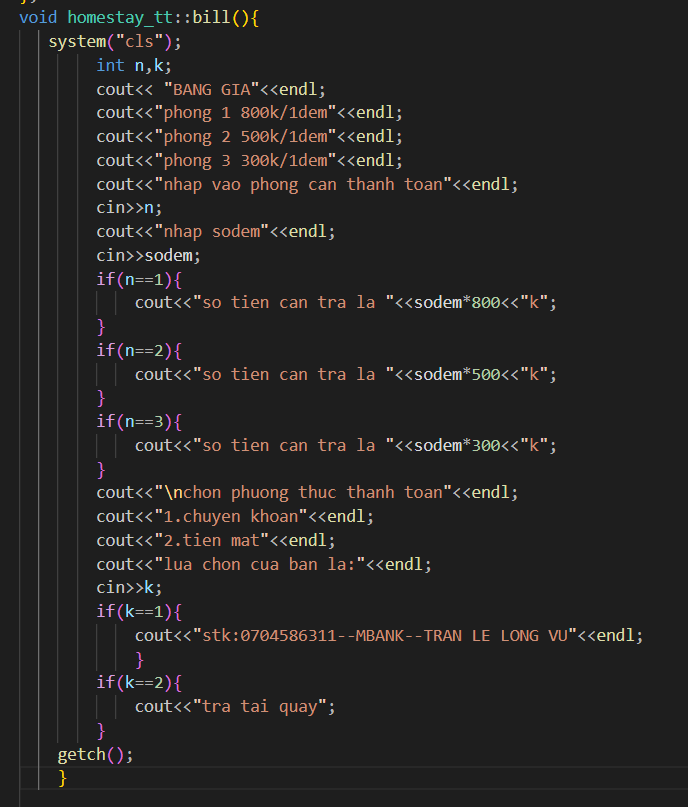
Chức năng hiển thị thông tin:



Chức năng sủa thông tin:



Chức năng thanh toán:



**4 chương trình và kết quả**

4.1 tổ chức chương trình

-chương trình quản lý dv lưu trú được xây dựng trên cấu trúc sau:

Main

Thanh toán

Sửa thông tin

Hiển thị thông tin

Đặt phòng

Exit

-Người dùng sẽ tiếp cận với hàm chính của chương trình, chức năng sẽ hiển thị trong hàm "menu". Menu này bao gồm 4 chức năng chính:đặt phòng , hiển thị thông tin, sửa thông tin, thanh toán. Người dùng có thể lựa chọn chức năng mà họ muốn thực hiện, và chương trình sẽ tiến hành thực hiện chức năng tương ứng.

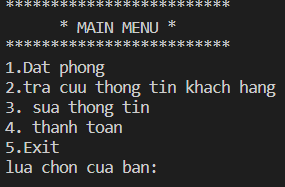
-Với cấu trúc này, chương trình quản lí dịch vụ lưu trú được tổ chức một cách rõ ràng và dễ hiểu. Người dùng có thể tương tác với các chức năng thông qua menu và lựa chọn lưu trữ dữ liệu một cách linh hoạt. Chương trình đảm bảo tính toàn vẹn và đáng tin cậy của dữ liệu, đồng thời cung cấp cho người chủ sự tiện lợi trong quá trình quản lý khách hàng

4.2 Ngôn ngữ cài đặt

Ứng dụng quản lý điểm sinh viên được chúng tôi xây dựng bằng sử dụng ngôn ngữ lập trình C/C++, một ngôn ngữ mạnh lập trình bậc cao, mang lại khả năng cao cho việc phát triển và triển khai. Để thực hiện và thử nghiệm ứng dụng, chúng tôi đã lựa chọn Visual studio code một môi trường phát triển tích hợp (IDE) được sử dụng rộng rãi. Visual studio code không chỉ cung cấp một giao diện thân thiện và tiện ích, mà còn hỗ trợ nhiều tính năng quan trọng để giúp đơn giản hóa quá trình phát triển và kiểm thử ứng dụng của chúng tôi.

4.3 kết quả.

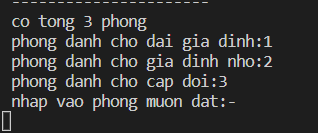
4.3.1 giao diện chính của chương trình



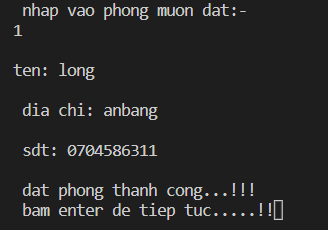
4.3.2 kết quả thực thi chương trình:

-Khi ta lựa chọn chức năng đặt phòng:

+Giao diện sẽ hiện ra cho khách hàng:



+sau khi chọn phòng muốn đặt:



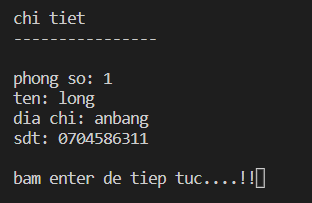
Sẽ lần lượt nhập vào các thông tin để xác nhận đặt phòng và sau đó bấm enter để trở lại menu.

-Khi ta chọn chức năng tra cứu thông tin:

+Giao diện sẽ hiện ra :

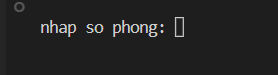


+sau khi nhập vào phòng muốn hiển thị:

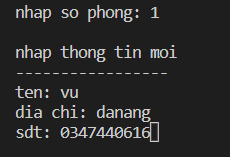


-khi ta lựa chọn chức năng sửa thông tin :

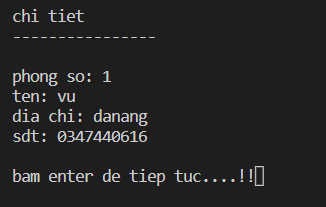
+giao diện sẽ hiện ra :



+sau khi nhập vào số phòng:

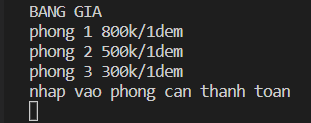


+Cho phép nhập vào thông tin mới thay thế thông tin cũ và sau đó chọn chức năng hiển thị thông tin sẽ hiện thị thông tin mới

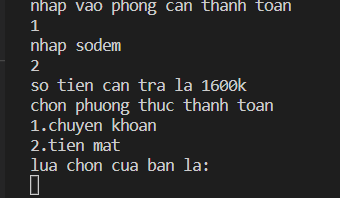


-khi chọn chức năng thanh toán:

+giao diện sẽ hiện bảng giá:



+ sau khi nhập vào phòng cần thanh toán và số đêm sẽ hiện ra số tiền cần trả và tiếp theo chọn phương thức thanh toán và thanh toán



-khi chọn chức năng Exit sẽ thoát ra khỏi chương trình.

4.3.3 Nhận xét và đánh giá

-Chúng tôi đã thành công xây dựng ứng dụng quản lý dịch vụ lưu trú với sự hoàn thiện cơ bản. Mặc dù chúng tôi còn nhận thấy một số khuyết điểm liên quan đến cấu trúc và giao diện, nhưng chúng tôi cam kết không ngừng nỗ lực để nâng cao chất lượng và tạo ra trải nghiệm tuyệt vời cho người dùng.

- Ứng dụng của chúng tôi được thiết kế với mục tiêu cung cấp cho người dùng một giao diện thuận tiện và dễ sử dụng. Chúng tôi đặt sự quan tâm đặc biệt vào việc tạo ra một giao diện dễ sử dụng cho người dùng , nhằm mang lại trải nghiệm tốt nhất cho người dùng.

- Chúng tôi lắng nghe phản hồi của người dùng và sẵn sàng thay đổi và cải tiến để đáp ứng nhu cầu ngày càng cao của họ. Chúng tôi luôn đặt trọng điểm vào việc nâng cao chất lượng và tính thân thiện của ứng dụng, nhằm đáp ứng tốt nhất nhu cầu và mong muốn của người dùng.

**5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

5.1 Kết luận.

Qua quá trình nguyên cứu, chúng tôi đã tạo ra một ứng dụng đáng tin cậy và tiện lợi, giúp quản lý dịch vụ lưu trú trở nên dễ dàng và hiệu quả.

Ứng dụng của chúng tôi cung cấp các chức năng cơ bản như đặt phòng, hiển thị thông tin, sửa thông tin , thanh toán. Chúng tôi đã thiết kế giao diện người dùng thân thiện và trực quan, giúp người dùng dễ dàng tương tác và sử dụng ứng dụng một cách thuận tiện.

Tổng kết lại, đồ án lập trình xây dựng ứng dụng quản lý dịch vụ lưu trú là một thành công ban đầu và đã đạt được mục tiêu đề ra.

5.2 Hướng phát triển

- Tối ưu cấu trúc và tăng tốc xử lý để cải thiện hiệu suất làm việc.

- Đẩy mạnh phát triển giao diện và cải thiện đồ họa của ứng dụng.

- Mở rộng và bổ sung thêm nhiều chức năng cho ứng dụng.