UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2022/2023

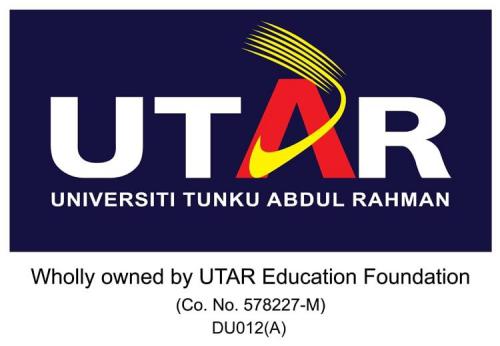
JUNE 2022 TRIMESTER

**ASSIGNMENT**

**REPORT**

**Candidate is required to fill in ALL the information below:**

|  |  |  |  |
| --- | --- | --- | --- |
| Group Leader: (*Name, ID, Programme*) | **Tin Hui Hui, 2103451, SE** | | |
| Group member 1: (*Name, ID, Programme*) | **Choo Jia Zheng, 2101998, SE** | | |
| Group member 2: (*Name, ID, Programme*) | **Emily Wee Earn Xin, 2103552, SE** | | |
| Group member 3: (*Name, ID, Programme*) | **Loke Weng Yan, 2103237, SE** | | |
| Group member 4: (*Name, ID, Programme*) | **Mooi Chee Ying, 2103124, SE** | | |
| Faculty /Institute/ Centre: | **LKC FES** | Group number: | **G1** |
| Course Code: | **UECS1004/UECS1104** | Course Description: | **Programming and Problem Solving** |

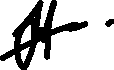


**DECLARATION STATEMENT**

We hereby solemnly and sincerely declare and confirm that we have read, understood and shall abide and comply with all laws, rules, regulations, guidelines and lawful instruction of the University and its staff in relation to the commencement of any assessment / examination during my programme of study in Universiti Tunku Abdul Rahman.

We hereby declare that our submission for all assessment / examination during our programme of study in the University shall be based on our original work, not plagiarised from any source(s) except for citations and quotations which have been duly acknowledged. We are fully aware that students who are suspected of violating this pledge are liable to be referred to the Examination Disciplinary Committee of the University.

We further certify that we have read and understand the above guidelines and agree to abide by the above declarations. The group leader signs this declaration statement on behalf of any and all group members.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group Leader Name: Tin Hui Hui

Student’s I.C. / Passport No.: 020513-01-0908

Date: 8 September 2022

### TABLE OF CONTENTS

**DECLARATION .................................................................................................... 1**

**TABLE OF CONTENTS ....................................................................................... 2**

### CHAPTER

**1 INTRODUCTION TO SOLUTION DESIGN................................................ 3**

1.1 Introduction/Background ............................................................ 3

1.2 System Roles/Users .................................................................... 4

1.3 System Modules .......................................................................... 5

1.3.1 *Module 1: Main* ............................................................ 5

1.3.2 *Module 2: Resident Page* ............................................. 5

1.3.3 *Module 3: Resident Registration* .................................. 5

1.3.4 *Module 4: Parcel Registration* ..................................... 6

1.3.5 *Module 5: Parcel Collection* ........................................ 6

1.4 Structure Chart ........................................................................... 7

1.5 Flow Chart/Pseudocode .............................................................. 8

1.6 Special Issues/Constraint ............................................................. 22

**2 C++ PROGRAM ............................................................................................. 23**

**3 SAMPLE OUTPUT ........................................................................................ 38**

3.1 *Module 1: Main* ............................................................ 38

3.2 *Module 2: Resident Page* ............................................. 40

3.3 *Module 3: Resident Registration* .................................. 41

3.4 *Module 4: Parcel Registration* ..................................... 46

3.5 *Module 5: Parcel Collection* ........................................ 49

**4 SAMPLE INPUT ............................................................................................ 52**

4.1 *Text File 1* – Resident\_List.txt................................................... 52

4.2 *Text File 2* – pin.txt………….................................................... 56

## 1 INTRODUCTION TO SOLUTION DESIGN

**1.1 Introduction/Background**

Since residents in Boulevard Service Apartment have faced a problem collecting their parcel by finding it from a stack of parcels, the management of Boulevard Service Apartment intends to develop a service of community parcel lockers for its residents to enhance the convenient lifestyle of community living.

The system was developed using the programming language C++. In this system, four main sections are developed, which are resident page, resident registration, parcel registration, and parcel collection. The information of registered residents will be stored in a text file so that the system can retrieve their information to do validation when collecting the parcel.

Each apartment unit is required to register their apartment unit number, owner’s full name, contact number, and password to login to the system so that residents can receive the one-time PIN code when a parcel is delivered to a locker. In addition, the parcel will be registered to the resident based on their apartment unit number and a one-time PIN code will be generated. The courier will inform the recipient of the delivery status. Thus, residents who want to collect a parcel need to check the one-time PIN code in the system and enter the one-time PIN code at the locker station.

**1.2 System Roles/Users**

The system roles in this system are recipient and courier. Recipients can register their information in the system and collect the parcel by using the PIN code generated by the system after the courier informs them of the delivery status. If the recipient has registered before, they are not required to register again. Besides, couriers can register the parcel with the specific resident’s unit number and send the delivery status to the recipient after registration.

**1.3 System Modules**

**1.3.1 Module 1: Main**

In the main module, the program will display the name of the system and a welcoming greeting to the user, who are the courier and resident. In this module, the user will be able to choose whether they are a courier or resident, or even to exit the program. If the user inputs an option other than these, the program will display a notification to the user to inform them of the wrong option entered and let them try again. If the user chooses the courier option, the program will execute the parcel registration module. However, if the user chooses the resident option, the program will execute the resident page.

**1.3.2 Module 2: Resident Page**

This resident page module will be executed only when the user chooses the resident option in the main module. In the resident page module, the program will ask the user to choose whether to register, collect a parcel, or exit from the resident page module to the main module. If the user inputs an option other than these, the program will display a notification to the user to inform them of the wrong option entered and let them try again. If the user chooses the registration option, the program will execute the resident registration module. However, if the user chooses the parcel collection option, the program will execute the parcel collection module.

**1.3.3 Module 3: Resident Registration**

The resident registration will be executed when the user chooses the resident option in the main module. In the resident page module, the program will ask the user to choose whether to register, collect a parcel, or exit from resident page to the main module. If the user chooses the register option, the program will display resident registration module. The user should input the information such as unit number, name, password, and phone number. The limit number of password only 15 and valid phone number can up to 10. There will be a display show the user want to edit or not. After that, there is a display showing the resident registration is successful and display again the information that input by user.

**1.3.4 Module 4: Parcel Registration**

The resident registration module will be executed when the user chooses the courier option in the main module. In this module, the program will display a welcome message to the user. Then, there is a prompt for the user to enter the apartment unit number and the resident’s contact number. After that, there is a display showing the parcel registration was successful. Next, the module will ask the user whether there is another parcel to deliver. If the user needs to deliver another parcel, the process repeats. If not, the loop is broken and a thank you message is generated.

Besides, this module generates a 6-digit PIN code for the resident’s parcel collection. The pin code is generated by setting the lower boundary to be 100000 and the upper boundary to be 999999 and a 6-digit random number will be generated within that range of digits.

Afterwards, a text file containing the apartment unit number and PIN code is updated. At the end, a thank you message is displayed to the user.

**1.3.5 Module 5: Parcel Collection**

The parcel collection module will be executed only when the user chooses the parcel collection option in the resident page module. In this parcel collection module, the program will request the apartment unit number and password from the user for verification. If the verification fails, the program will display a notification and ask the user to key in the option again.

If the verification succeeds, the program will display whether there is any available parcel or not. If the parcel is available, the one-time pin code will be displayed on the screen. The user needs to key in the one-time pin code in the locker station in order to collect their parcel. If the parcel is not available, the notification will be displayed on the screen.

**1.4 Structure Chart**

Diagram

Description automatically generated

**1.5 Flow Chart/Pseudocode**

**1.5.1 Module 1: Main**

**Diagram

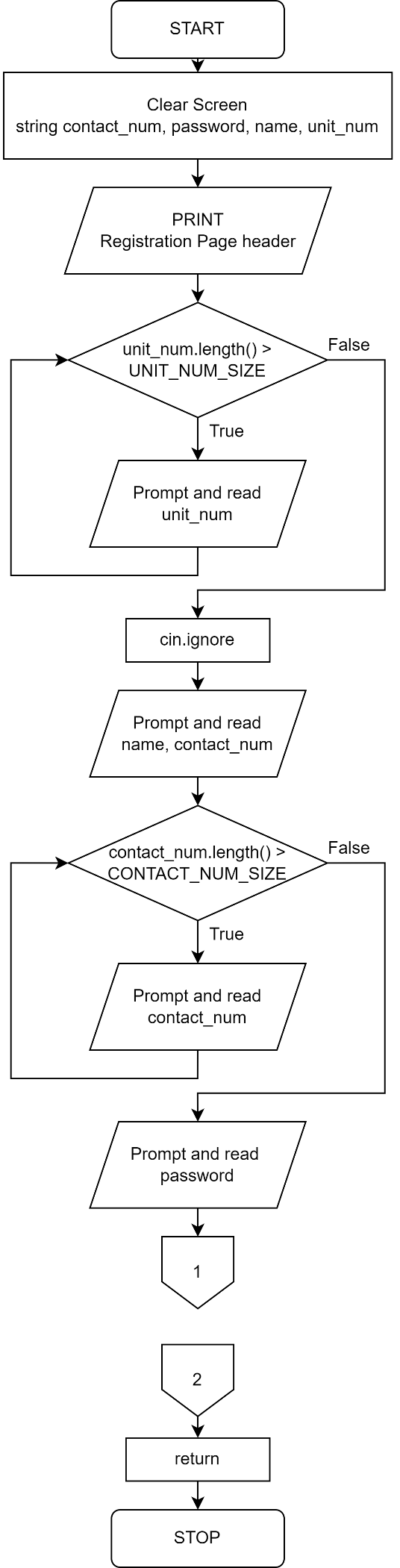
Description automatically generated**

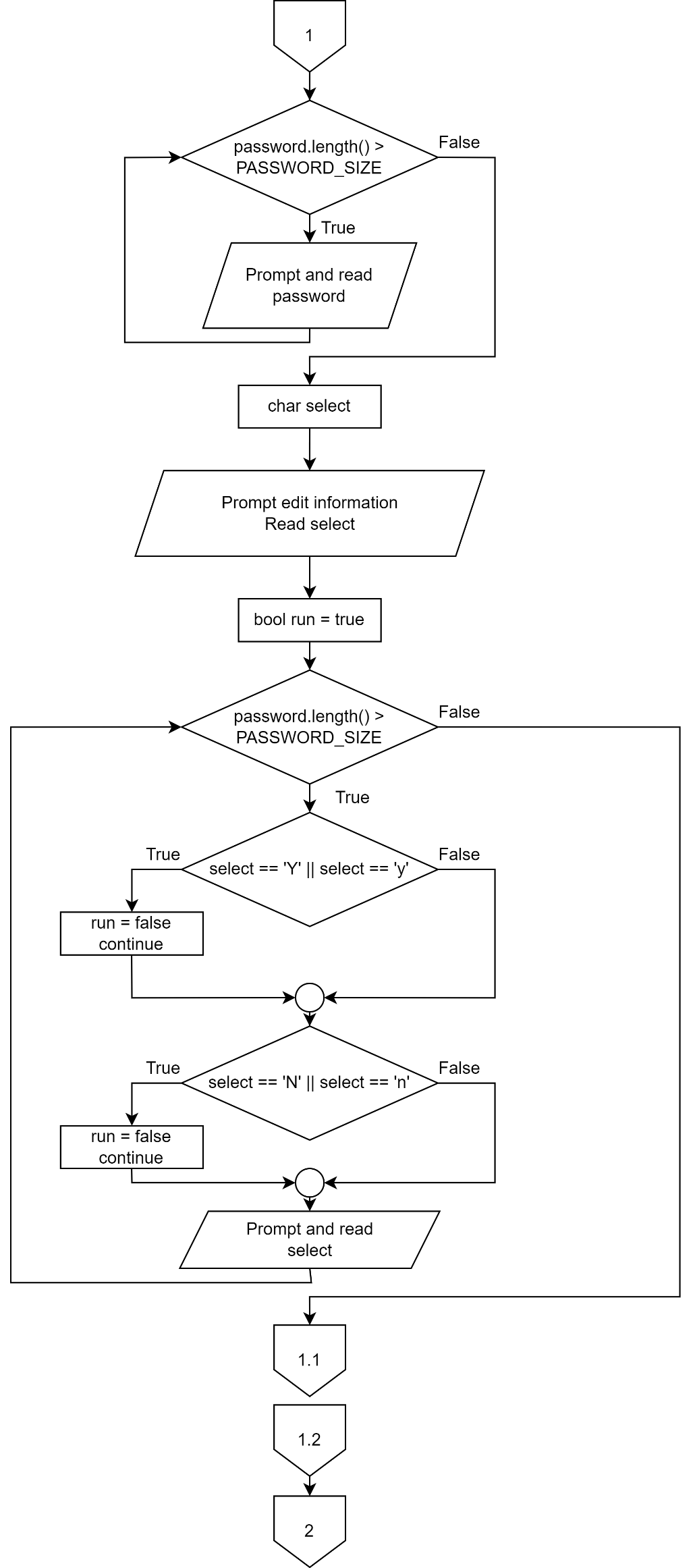
**1.5.2 Module 2: Resident Page**

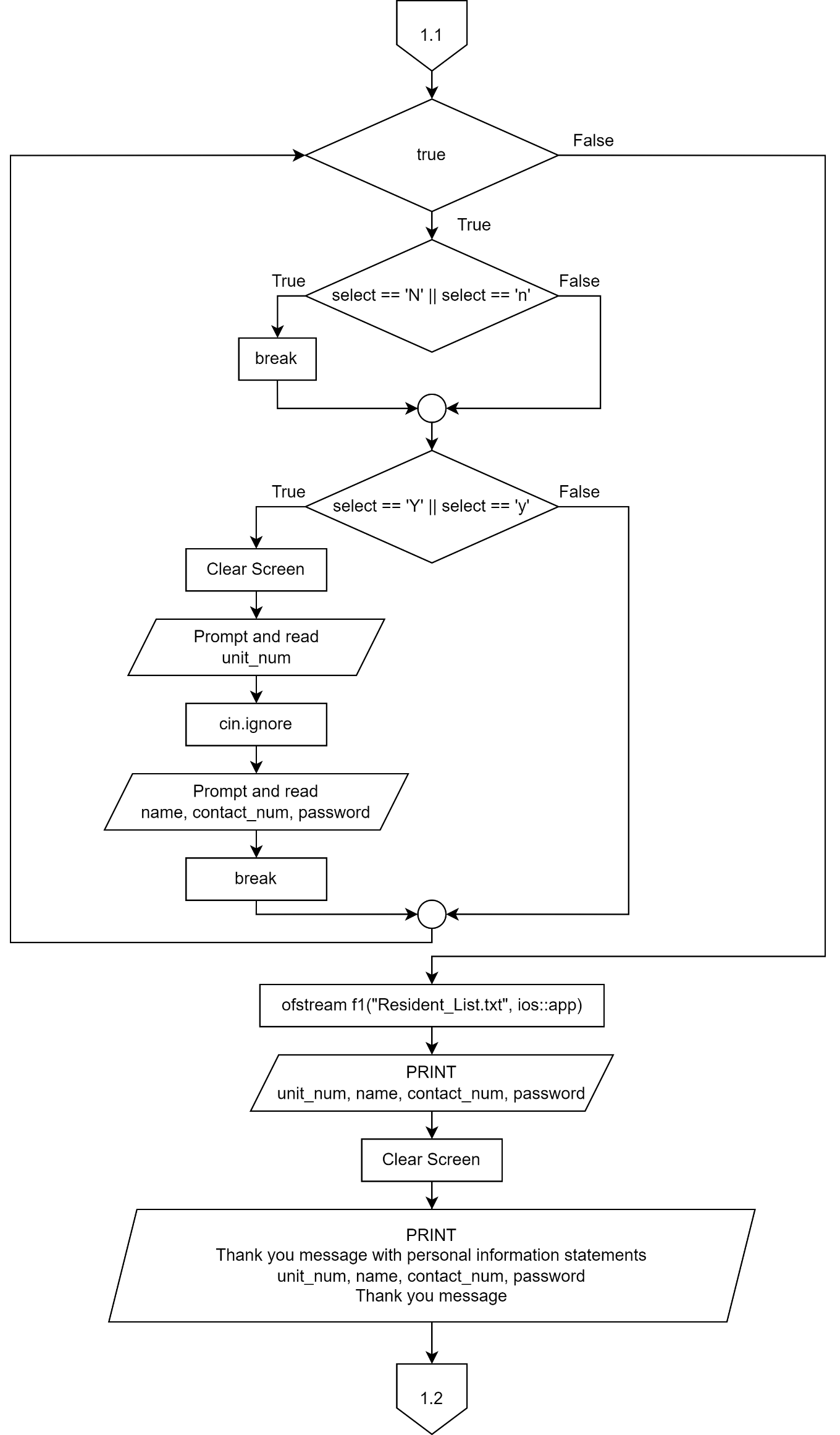
Diagram

Description automatically generated

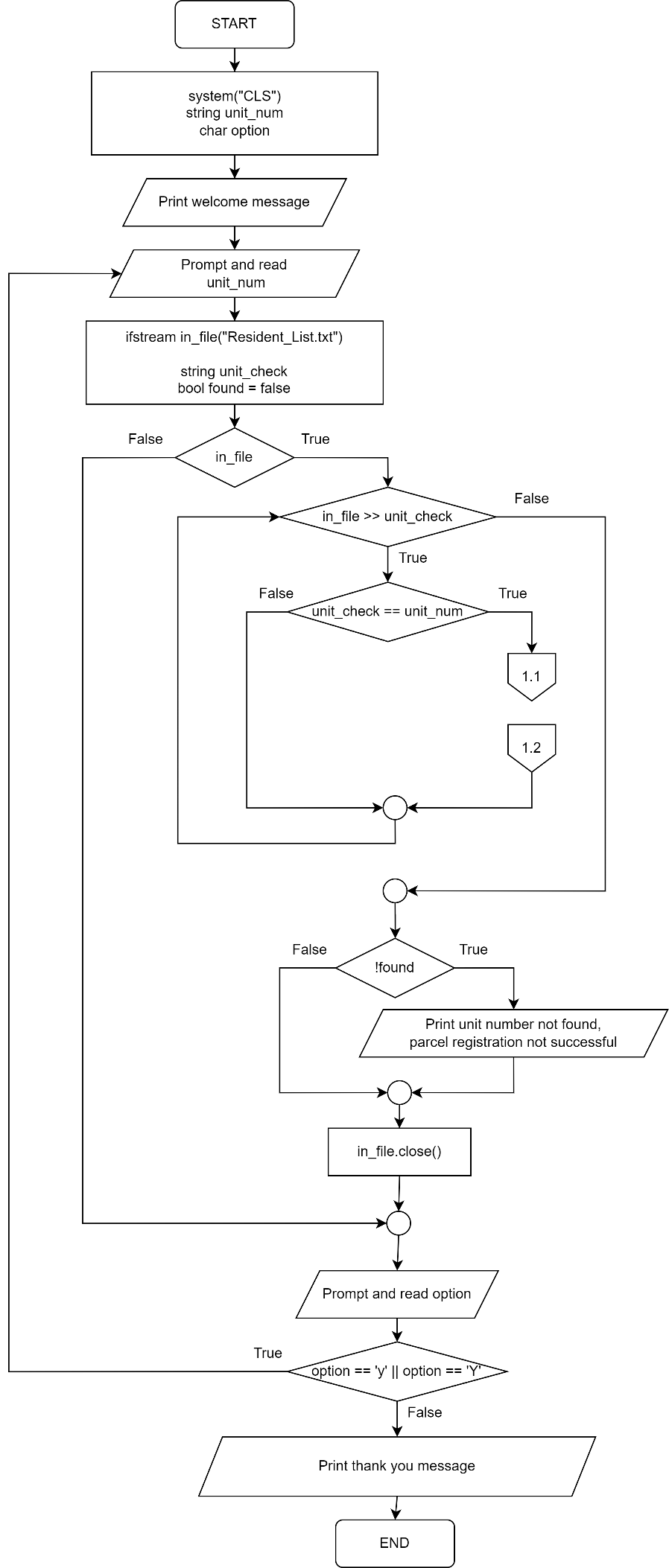
**1.5.3 Module 3: Resident Registration**

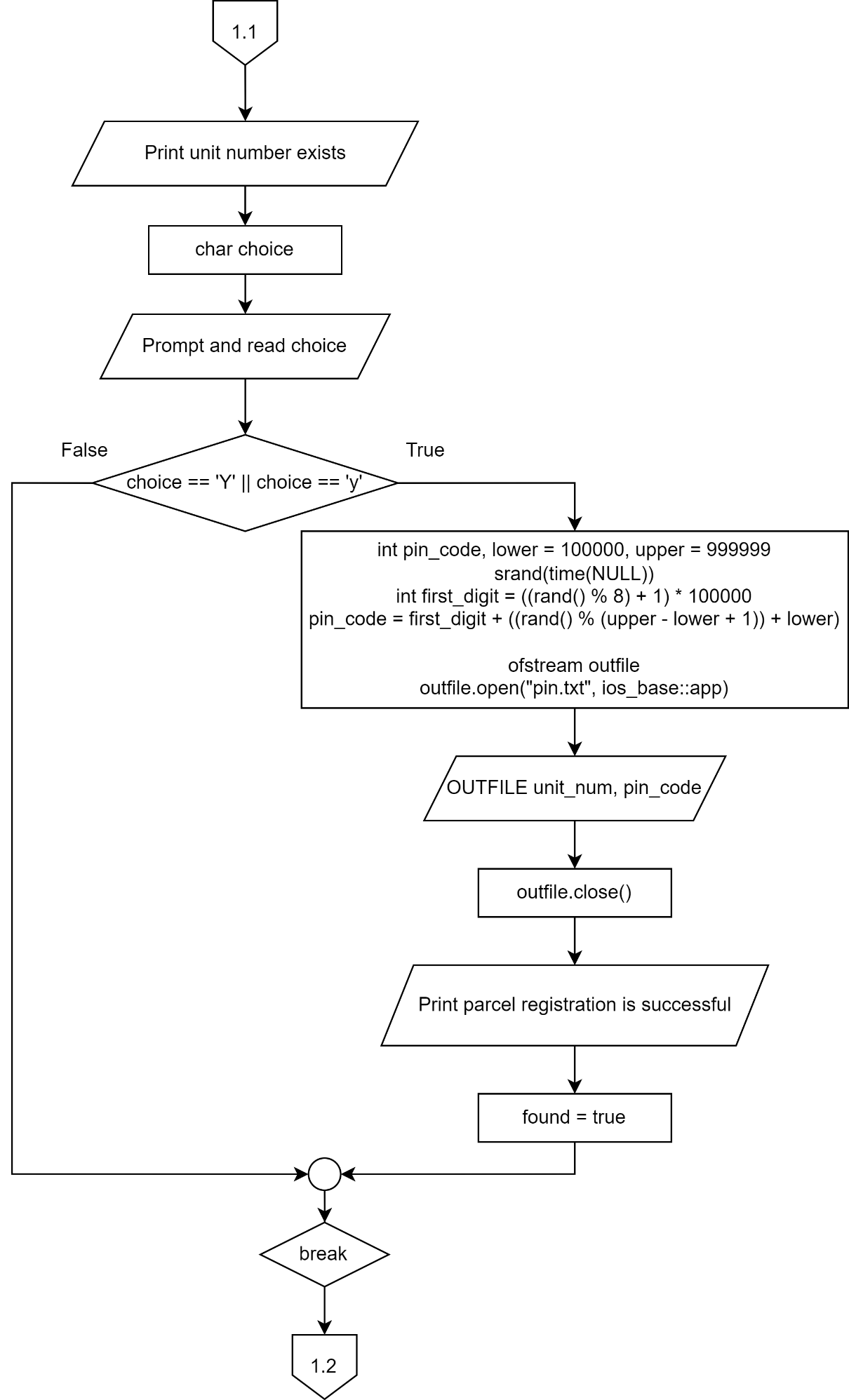






**1.5.4 Module 4: Parcel Registration**





**1.5.5 Module 5: Parcel Collection**

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

**1.6 Special Issues/Constraint**

In the section of resident registration of the system, residents must register with their apartment unit number, owner’s name, contact number, and set a password with a limit of 15 characters.

Besides, the system will validate the unit number entered by the courier in the parcel registration section. If the unit number entered does not exist, the parcel registration will not be successful. If the unit number entered exists, a PIN code with a 6-digit random number will be generated by setting the lower boundary to be 100000 and upper boundary to be 999999.

Other than that, the system will let the residents confirm the unit number and password entered. Also, If the unit number entered is not recorded in the system, residents need to register their apartment unit number, name, contact number, and password in the resident registration section. The system will validate the unit number and password entered with the information stored in the text file to ensure the password corresponds to the unit number entered in the parcel collection section before displaying the PIN code to the resident.

## 2 C++ PROGRAM

/\*

This is a C++ application to handle a community parcel locker system for the residents of Boulevard Service Apartment.

This application can be used by couriers to deliver parcels, and residents to collect their parcel.

Couriers can register parcels for delivery by entering residents' details.

This application reads the residents' details from a text file containing the residents' unit number, name,

contact number, and password, then generates a PIN code for the residents' parcel collection.

\*/

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <cstdlib>

#include <fstream>

#include <ctime>

#include <cstring>

#include <string>

using namespace std;

#define UNIT\_NUM\_SIZE 5

#define NAME\_SIZE 30

#define CONTACT\_NUM\_SIZE 15

#define PASSWORD\_SIZE 15

typedef struct

{

char unit\_num[5];

char name[30];

char contact\_num[15];

char password[15];

} RESIDENT;

typedef struct

{

string unit\_num2;

string pincode;

} PINCODELST;

// Function prototypes

void resident\_page();

void resident\_regis();

void parcel\_regis();

void parcel\_collect();

/\*

The main function displays the main menu of the application, then prompts the user, asking who they are to proceed.

The user can choose whether they are a courier, a resident, or they can exit the application.

\*/

int main(void)

{

bool loop = true;

while (loop)

{

system("CLS");

char option;

cout << "\t\t==================================================\n" << endl;

cout << "\t\t Boulevard Service Apartment \n" << endl;

cout << "\t\t==================================================\n" << endl;

cout << "\t\t Welcome to Boulevard Service Apartment " << endl;;

cout << "\t\t Community Parcel Locker System \n" << endl;

cout << "\tWho are you? " << endl;

cout << "\t1. Courier\t2. Resident\t3. Exit application\n" << endl;

cout << "\tEnter option : ";

cin >> option;

if (option == '3')

{

cout << "\n\tThank you for using our application!";

loop = false;

}

else

{

while (option != '1' && option != '2')

{

cout << "\tWrong option. Please try again.\n\tEnter option : ";

cin >> option;

}

if (option == '1')

{

parcel\_regis();

}

else

{

resident\_page();

}

system("pause");

}

}

return 0;

}

/\*

The resident\_page function displays the resident page.

The function prompts whether the user wants to register, collect their parcel, or exit the page.

\*/

void resident\_page()

{

system("CLS");

char resident\_op;

cout << "\tWelcome, resident\n" << endl;

cout << "\tWhat can we do to help you?\n" << endl;

cout << "\t1. Registration\n\t2. Parcel Collection\n\t3. Exit\n" << endl;

cout << "\tEnter option : ";

cin >> resident\_op;

while (resident\_op != '1' && resident\_op != '2' && resident\_op != '3')

{

cout << "\tWrong option. Please try again.\n\tEnter option : ";

cin >> resident\_op;

}

/\*

Switch statement allows the user to make options in the resident page.

When user inputs '1': resident\_regis function is called for resident registration.

When user inputs '2': parcel\_collect function is called for parcel collection.

\*/

switch (resident\_op)

{

case '1':

resident\_regis();

break;

case '2':

parcel\_collect();

break;

default:

cout << "\tPlease press 'Enter' to exit this page.\n";

}

system("pause");

return;

}

/\*

The resident\_regis function is to let users register for parcel collection.

The function gets the users' unit number, name, contact number and password, then saves them into a text file.

\*/

void resident\_regis()

{

system("cls");

string contact\_num;

string password;

string name;

string unit\_num;

cout << "\t\t\t\tRegistration Page\t\t\n\n";

cout << "\tEnter apartment unit number : ";

cin >> unit\_num;

// If the length of unit number is more than 5

while (unit\_num.length() > UNIT\_NUM\_SIZE)

{

cout << "\tInvalid apartment unit number! Please key in again...";

cout << "\n\tEnter apartment unit number : ";

cin >> unit\_num;

}

cin.ignore(); // Ignores or clears one or more characters from the input buffer

cout << "\n\tEnter name : ";

getline(cin, name);

cout << "\n\tEnter contact number : ";

cin >> contact\_num;

// If the length of contact number is more than 15

while (contact\_num.length() > CONTACT\_NUM\_SIZE)

{

cout << "\tInvalid contact number! Please key in again...";

cout << "\n\tEnter contact number : ";

cin >> contact\_num;

}

cout << "\n\tEnter password (max 15) : ";

cin >> password;

// If the length of password is more than 15

while (password.length() > PASSWORD\_SIZE)

{

cout << "\tInvalid password. Password too long.Please key in again...";

cout << "\n\tEnter password (max 15) : ";

cin >> password;

}

char select;

cout << "\n\tDo you want to edit the information?"; // Lets the user choose if they want to edit the information or not

cout << "\n\tyes or no (Y or N): ";

cin >> select;

bool run = true;

while (run)

{ // If the user does not input 'y' or 'n', the prompt keeps displaying until user types the correct option

if (select == 'Y' || select == 'y')

{

run = false;

continue;

}

if (select == 'N' || select == 'n')

{

run = false;

continue;

}

cout << "\n\tInvalid\n\tEnter Y or N only: ";

cin >> select;

}

while (true)

{

if (select == 'N' || select == 'n')

{ //If the user inputs 'N', the system will break and goes to the register successfully message

break;

}

if (select == 'Y' || select == 'y')

{ //If the user inputs 'Y', the system will keep running

system("cls");

cout << "\tPlease re-enter the information!";

cout << "\n\tEnter apartment unit number : ";

cin >> unit\_num;

cin.ignore(); // Ignore or clears one or more characters from the input buffer

cout << "\n\tEnter name : ";

getline(cin, name);

cout << "\n\tEnter contact number : ";

cin >> contact\_num;

cout << "\n\tEnter password (max 15) : ";

cin >> password;

break;

}

}

ofstream f1("Resident\_List.txt", ios::app);

f1 << unit\_num << "\n" << name << "\n" << contact\_num << "\n" << password << "\n"; // Input information will be saved into text file

system("cls");

cout << "\n\tThank you";

cout << "\n\tYour registration was successful! Your personal information is stated below.\n";

cout << "\n\tApartment unit number : " << unit\_num;

cout << "\n\tName : " << name;

cout << "\n\tContact number : " << contact\_num;

cout << "\n\tPassword : " << password;

cout << "\n\tThank you for using Boulevard Service Apartment Community Parcel Locker System!";

cout << "\n\tPlease press Enter to exit this page.";

return;

}

/\*

The parcel\_regis function is for couriers to register and deliver parcels.

The function gets the unit number and checks with the text file to see whether it exists.

\*/

void parcel\_regis()

{

system("CLS");

string unit\_num;

char option;

cout << "\tWelcome to the Parcel Registration Page!\n";

do

{

cout << "\n\tPlease enter apartment unit number : ";

cin >> unit\_num;

ifstream in\_file("Resident\_List.txt");

string unit\_check;

bool found = false;

if (in\_file)

{

while (in\_file >> unit\_check)

{

if (unit\_check == unit\_num)

{

// Lets courier know the unit number exists

cout << "\n\t\tUNIT NUMBER EXISTS!";

char choice;

// Prompts confirmation of the parcel delivery

cout << "\n\tConfirmation for parcel delivery to " << unit\_num << " (Y/N) : ";

cin >> choice;

if (choice == 'Y' || choice == 'y')

{

// 6-digit PIN code generation with random numbers

int pin\_code, lower = 100000, upper = 999999;

srand(time(NULL));

int first\_digit = ((rand() % 8) + 1) \* 100000;

pin\_code = first\_digit + ((rand() % (upper - lower + 1)) + lower);

ofstream outfile;

outfile.open("pin.txt", ios\_base::app);

outfile << unit\_num << "\n" << pin\_code << endl;

outfile.close();

cout << "\n\tParcel registration is SUCCESSFUL!\n";

found = true;

break;

}

else

{

break;

}

}

}

if (!found)

{

// Lets courier know parcel registration is unsuccessful

cout << "\n\t\tUNIT NUMBER NOT FOUND!";

cout << "\n\tParcel registration is NOT SUCCESSFUL!\n";

}

in\_file.close();

}

// Asks courier if there is another parcel to deliver

cout << "\n\tDo you have another parcel to deliver (Y/N) : ";

cin >> option;

} while (option == 'y' || option == 'Y');

cout << "\n\tThank you for using the Boulevard Service Apartment Community Parcel Locker System!\n";

}

/\*

The parcel\_collect function allows residents to collect their parcels.

The resident enters their unit number, password, then receives their PIN code.

\*/

void parcel\_collect()

{

system("CLS");

ifstream in\_file("Resident\_List.txt", ios::in);

if (!in\_file)

cout << "Error opening file" << endl;

else

{

RESIDENT residents[10000];

int count = -1;

in\_file.getline(residents[++count].unit\_num, UNIT\_NUM\_SIZE);

while (in\_file)

{

in\_file.getline(residents[count].name, NAME\_SIZE);

in\_file.getline(residents[count].contact\_num, CONTACT\_NUM\_SIZE);

in\_file.getline(residents[count].password, PASSWORD\_SIZE);

in\_file.getline(residents[++count].unit\_num, UNIT\_NUM\_SIZE);

}

in\_file.close();

string unit\_num\_input, password\_input;

string name;

char check;

cout << "\tWelcome! Sir/Madam!\n" << endl;

cout << "\tPlease enter your apartment unit number : ";

cin >> unit\_num\_input;

cout << "\tPlease enter your password : ";

cin >> password\_input;

bool loop = true;

while (loop)

{

// Confirmation for correct data input

cout << "\n\tYour entered apartment unit number : " << unit\_num\_input << endl;

cout << "\tYour entered password : " << password\_input << endl;

cout << "\tPlease make sure your entered data are correct. (Y for Yes, N for no) : ";

cin >> check;

if (check == 'Y' || check == 'y')

loop = false;

else

{

if (check == 'N' || check == 'n') // Enter data again if previous entry is wrong

{

cout << "\n\tPlease enter your apartment unit number : ";

cin >> unit\_num\_input;

cout << "\tPlease enter your password : ";

cin >> password\_input;

}

else

{

cout << "\n\tWrong option. Only Y, y, N, n are allowed. Please re-enter again." << endl;

cout << "\tPlease make sure your entered data are correct. (Y for Yes, N for no)" << endl;

cin >> check;

}

}

}

bool loop2 = true;

while (loop2)

{

bool acc\_exist = false;

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

{

if (residents[i].unit\_num == unit\_num\_input)

{

acc\_exist = true;

if (residents[i].password == password\_input)

{

ifstream in\_file2("pin.txt", ios::in);

if (!in\_file2)

cout << "\n\tError opening text file.\n";

else

{

PINCODELST pincode[50];

int index = -1;

in\_file2 >> pincode[++index].unit\_num2;

while (in\_file2)

{

in\_file2 >> pincode[index].pincode;

in\_file2 >> pincode[++index].unit\_num2;

}

in\_file2.close();

bool found = false;

for (int j = 0; j <= index; j++)

{

if (pincode[j].unit\_num2 == unit\_num\_input)

{

// Prompts user that there is a parcel in their locker, with their PIN code.

cout << "\n\tYou have a parcel in the locker. Please collect it as soon as possible.\n";

cout << "\tYour one-time PIN code : " << pincode[j].pincode << endl;

cout << "\tYou are only allowed to view the pin code once.\n";

cout << "\tPlease remember your pin code before you exit from this page.\n";

cout << "\n\n";

int indexs = j;

for (int k = indexs; k < index - 1; k++)

{

pincode[k].unit\_num2 = pincode[k + 1].unit\_num2;

pincode[k].pincode = pincode[k + 1].pincode;

}

ofstream out\_file("pin.txt", ios\_base::out);

if (!out\_file)

cout << "Error opening output file.\n";

else

{

for (int l = 0; l < index - 1; l++)

{

out\_file << pincode[l].unit\_num2 << endl;

out\_file << pincode[l].pincode << endl;

}

}

out\_file.close();

found = true;

break;

}

}

if (found == false)

{

// Prompts user that there is no parcel in their locker.

cout << "\n\tThe apartment unit number of " << unit\_num\_input << " don't have a parcel currently.\n\n";

}

}

}

}

}

if (acc\_exist == false)

{

// Prompts user that they do not have an account.

cout << "\n\tThe apartment unit number of " << unit\_num\_input << " don't have an account.\n";

cout << "\tPlease register an account first.\n\n";

}

cout << "\tThank you for using Boulevard Service Apartment Community Parcel Locker System!" << endl;

cout << "\tPlease press 'Enter' to exit this page.\n\n";

loop2 = false;

}

}

return;

}

## 3 SAMPLE OUTPUT

* 1. **Module 1: Main**

Text

Description automatically generatedThe program’s main menu for the Boulevard Service Apartment locker system. Inputting ‘1’ leads to the parcel registration function, inputting ‘2’ leads to the resident page function, and inputting ‘3’ exits the application.

*Figure 3.1.1: Main Menu with 3 Different Functions*

**3.1.1 Main – Wrong Input**

If the character of choice entered by the user is incorrect, an invalid statement will be displayed. Users can re-enter another character that is valid in the next try.

Assumption: User inputs only characters that are integers.

Text

Description automatically generated

*Figure 3.1.2: Wrong Input Character Entered by User*

**3.2 Module 2: Resident Page**

Entering option ‘2’ to select the “Resident” in the Main Menu.

Text

Description automatically generated

*Figure 3.2.1: User selecting the resident page function*

After pressing the ‘Enter’ key, the resident can enter the option of either of these three listed functions.

Text

Description automatically generated

*Figure 3.2.2: Displaying three different functions*

* 1. **Module 3: Resident Registration**

Entering option ‘1’ to select the “Registration” in the resident page.

**Text

Description automatically generated**

*Figure 3.2.3 User selecting the resident registration function.*

After pressing the ‘Enter’ key, resident can enter their apartment unit number.

*Text

Description automatically generated*

*Figure 3.2.4 User entering the apartment unit number*

Text

Description automatically generatedThe program prompts the resident to enter their name.

*Figure 3.2.5 User entering the unit owner’s name*

The resident enters their contact number for registration.

**Text

Description automatically generated***Figure 3.2.6 User entering the unit owner’s contact number*

The resident enters a password with a limit of 15 characters to complete the registration.

**Text

Description automatically generated***Figure 3.2.7 User entering the password*

The resident enters ‘N’ to quit the edit section.

**Text

Description automatically generated**

*Figure 3.2.8 Prompt to edit the information entered*

The information entered by the resident for registration is displayed on the screen.

**Text

Description automatically generated**

*Figure 3.2.9. Program displays the information user entered*

* + 1. **User Edits Information**

The resident enters ‘Y’ to enter the edit information section.

**Text

Description automatically generated**

*Figure 3.2.1.1 Prompt to edit information entered*

Resident can re-enter the information in the edit information section.

**Text

Description automatically generated**

*Figure 3.2.1.2 Re-enter information*

The edited information is successfully saved and displayed to the resident.

**Text

Description automatically generated**

*Figure 3.2.1.3 Program displays edited information*

* 1. **Module 4: Parcel Registration**

If the user identity is a courier, the user can enter the character ‘1’ to enter the parcel registration function.

**Text

Description automatically generated**

*Figure 3.3.1: User entering option ‘1’*

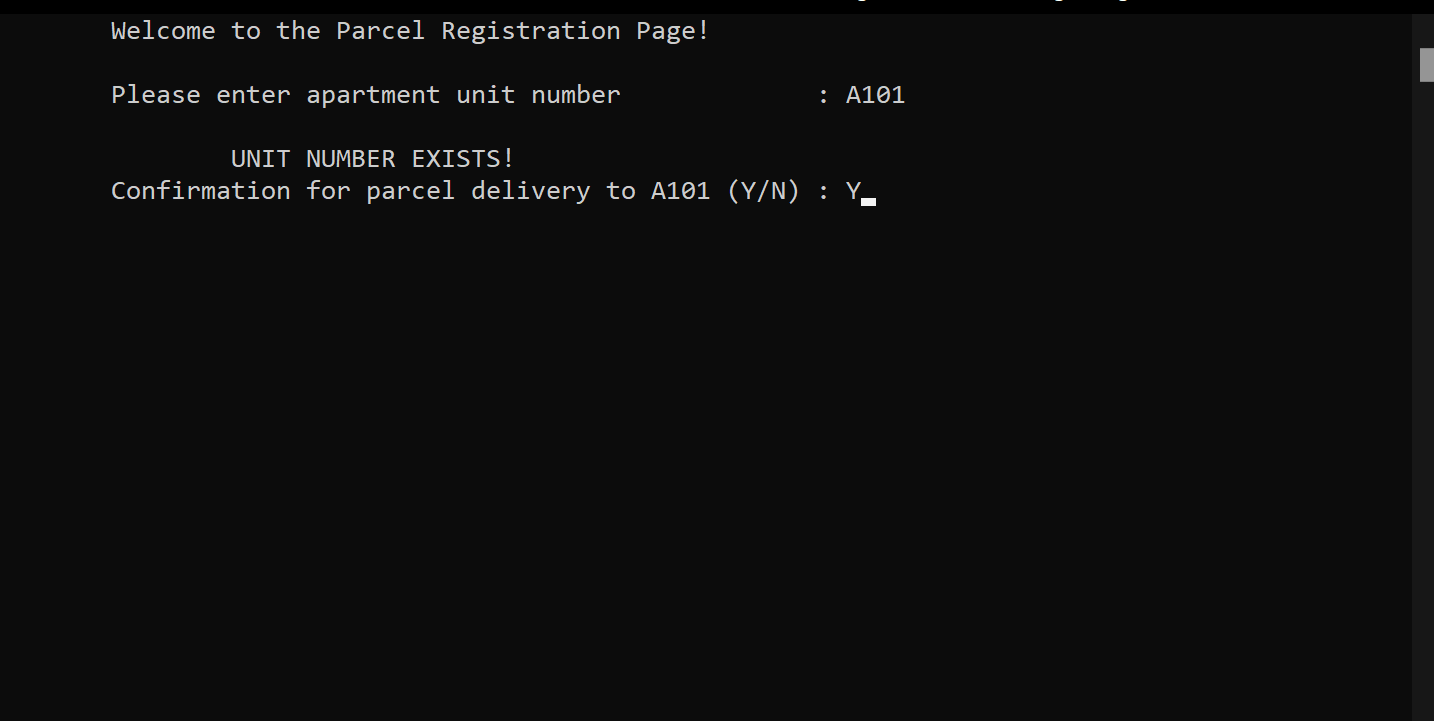
Courier enters the unit number which the parcel need be deliver.

**Text

Description automatically generated**

*Figure 3.3.2 User entering apartment unit number*

If the unit number entered by the courier is exists, the system will send a message to get the confirmation of the courier for delivering the parcel to the unit number entered.

****

*Figure 3.3.3 Confirmation of delivering parcel to entered apartment unit*

After the first parcel registration is completed, the system will ask if there are any more parcels to be delivered. Enter ‘N’ to quit this section.

Text

Description automatically generated

*Figure 3.3.4 Option to deliver another parcel*

The parcel registration is successful.

*Text

Description automatically generated*

*Figure 3.3.5 Parcel registration successful*

* 1. **Module 5: Parcel Collection**

This is the starting of the parcel collection page.

Text

Description automatically generated

*Figure 3.4.1 Parcel collection page*

The resident enters the apartment unit number to check whether there is a parcel in the locker.

Text

Description automatically generated

*Figure 3.4.2 Entering apartment unit number*

Text

Description automatically generatedThe resident enters the password for the entered unit number.

*Figure 3.4.3 Entering password*

Text

Description automatically generatedAfter entering the unit number and password, the system will send a confirmation message to resident.

*Figure 3.4.4 Confirmation of entered information*

If the unit number entered by the resident has a parcel, the system will display the one-time PIN code to the resident for the parcel collection.Text

Description automatically generated

*Figure 3.4.5 Program displaying one-time PIN code*

## 4 SAMPLE INPUT

**4.1 Text File 1 – Resident\_List.txt**

* This text file contains the residents’ information in the following order: Unit number, name, contact number, password.

A101

GOH MEI HUA

0120406400

goh101

A103

CHAN XIAO HUA

0120094124

xh103

A104

LEE GOU ZHI

0129846134

lgz104

A106

LEE ZHAO CHAI

0128463166

lcz106

A115

LIEW YUAN BAO

0124643135

45ad5cz

A201

LIEW HUAN FA

0124547945

456sda

A205

WONG FA MING

0152461215

4fs5dad

A206

WONG CHUI HUA

0127194345

paasp4

A207

GOH MEI LING

0187464478

6sfd4sfd

A208

CHAN XIAO MING

0124954656

cxming7894

A303

TI MEI XIN

0168789455

wtyn484

A306

MOOI MEI YING

0123463245

awa7ada

A308

LOKE JIA MING

0130812132

ljiam2132

A501

TIN HUI LING

0127497521

tnl7846

A502

CHOO WEE YING

0124141245

as5d4

A505

LOKE CHOO TING

0125551212

34afas6

A516

LOW AI LING

0123456712

4a24asf

A601

HONG ONG FA

0126014542

sf4hof

A602

ONG HUAT FAT

0129456021

ohf8845

A604

TANG YONG LAM

0126145042

as451asA222

A222

Chin Po Yan

0123456789

cpy123

**4.2 Text File 2 – pin.txt**

* This text file contains the PIN codes generated for parcel collection.

A106

521638

A207

519024

A208

827681

A501

215365

A505

802002

A516

711705

A602

720885

A101

807438

A101

210769