UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

W O R K S H O P 1 R E P O R T

Semester 1 Session 2022-2023

Name	NUR ATIQA IDAYU BINTI KAMARUZAMAN
Matric Numbers	B032110191
Course	BITS
Project Title	STUDENT HOSTEL MANAGEMENT SYSTEM
Supervisor	DATIN WIRA TS. NURAZLINA BINTI MD SANUSI
Evaluator	ASSOC. PROF DR. MOHD SANUSI AZMI

TABLE OF CONTENTS

PAGE

Table of Contents

89

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

IDE - Integrated Development Enviroment. In This System, Visual Studio Is The IDE Of Choice

SQL – Structured Query Language. This programming language used to interact with database

CHAPTER 1: INTRODUCTION

1.1 Introduction

A hostel is the accommodation for the students to live during their studies and their study place is away from their home. As an undergraduate student who is enrolled in college/university, living in a hostel that is provided by the university is a big advantage for the student. It will save the cost for the student because a university/ college hostel has cheaper fee than renting a house or room nearby the university. However, the hostel is only for the students that meet the requirements such as it must be in a first and second-year student and have taken part in a lot of college activities. It contains a lot of rooms from different blocks and levels. Some of the furniture was provided for the students to use such as bed, locker, and study table. Each block of the hostel has been managed by their own residential fellow that is responsible to keep the information and activities that will be carried out by the hostel.

In every semester, the students must apply if they want to live in the hostel or live outside the hostel. When they are chosen to live in the hostel, they need to come and do registration at the hostel office. Therefore, this system is developed to ease the process of the registration of the students, manage the students and room item information effectively. The students will also be able to update their own information and send report to the hostel management without going to the hostel office. This system also will help the hostel management to retrieve the report from the student within a short time and the hostel problem will be solve without waiting for a long time.

1.2 Problem Statements

- The current hostel management record the information of registration manually and it takes a long time to process the data.
- The process of finding available room for the students was not efficient and the staff need to refer it from the list of the students registered.
- The students do not have a lot of time to go to hostel office to send report of their room inventory and to register their electrical appliances.
- The security of the current system is very weak because anyone will be able to view the student's information and it will cause data leaking.

1.3 Objectives

1. To develop and design a system that will ease the registration of the student in the hostel.
2. To manage and secure the record of the student and their room report.
3. To optimize the time for searching available room for the student.
4. To easily generate a report of the students in the hostel.

1.4 Scope

1. Target User

- Hostel Staff (Admin)
  - -Log in Admin
  - -Search available room
  - -Register and Remove student
  - -Update and print student & room inventory report
  - -Print list of students
- Student (User)
  - -Log in User
  - -Update student information
  - -Register electrical appliances
  - -Update room inventory report

2. **System module**

- Login / logout module

  - Admin and Student can login and logout the system.

- Registration module

  - Admin can make or remove the registration of the students.

  - Student can register to bring their electrical appliances to the hostel.

- Room item report module

  -Students can send report to the admin if there is any issue about the room item.

- List of student module

  -Admin can generate report list of students based on faculty, block, or level.

1.5 Summary

This chapter explains the specifics of the project's introduction, which includes the problem statements that serve as the project's background and describe the purpose and context for the development of the system, the problems that the system addresses, the objectives, which are the goals that had to be attained during the system's development, and the scopes, which identify the system's intended audience. The project's problem analysis section, which includes a thorough description of the issue and a structured chart, will be covered in more detail in the following chapter.

CHAPTER 2: PROBLEM ANALYSIS

2.1 Introduction

The thorough description of the problem, which includes problem description and problem decomposition, will be covered in more detail in this chapter. This chapter also provides a system flow chart, which is essential for comprehending the issue and outlining potential fixes.

2.2 Problem Description

The first problem is manual student registration. Before the development of this system, the hostel staffs use manual method of registering the students. It takes a lot of time to register the students manually, and the registration process become slower. **The second problem is inefficiency of finding available room.** The hostel staff used to print the list of rooms and they will find the available room from the list. This process was not efficient because they need to check the list of the room one by one to find the available room. **The third problem is difficulty of finding right time for the student to register electrical appliance.** Previously, they needed to go to the hostel office to register, but it is hard for them to find the right time to go to the office because they are busy, and the office is only open at office hour. **Fourth problem is the security of the current system is very weak.** The data of the students and the room item was stored in a hardcopy file. It tend to be lost and the data can easily be stolen by other people.

2.3 Problem Decomposition

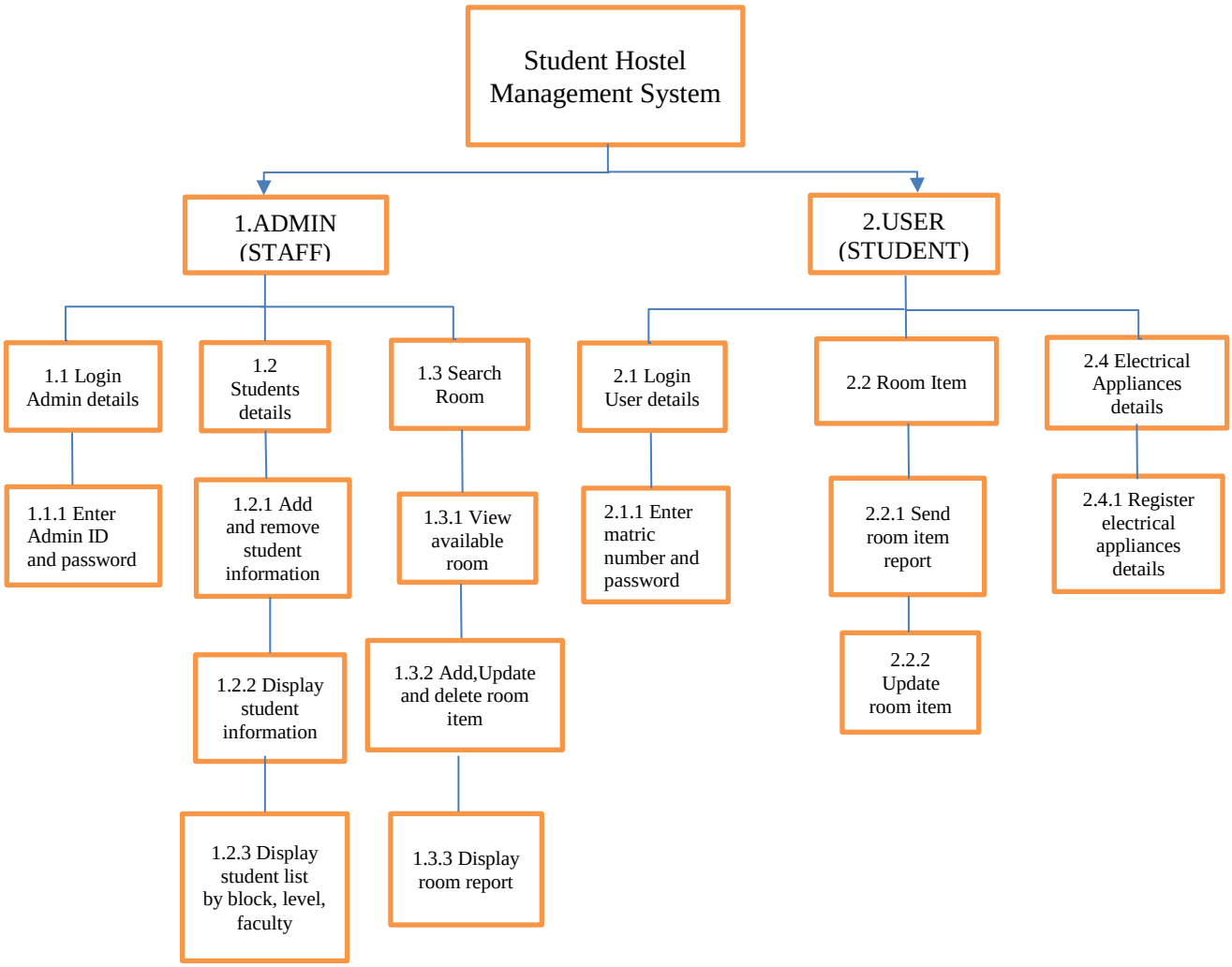
- The registration of student was not efficiently managed because the students must write their information on the provided list manually. More time is needed along the registration process since they need to write it on paper.

- The students can choose which room that they want to live, and the hostel staff must find the available room for the student from the registration list. This will also take longer time for the registration and also cause human error because there is a risk that the staff will overlook the available room for the students.

- The hostel needs to keep the student's electrical appliances record and their room inventory in different files separately. To retrieve the information, the students need to go to the hostel office that only open on office hour and students does not have so much time since they are busy to do a lot of tasks and activities. The room inventory information will become late to update since the staff need to wait for the student to send the report.

- The data of the students and the hostel was not secured and anyone in the hostel office can see the report and easily remove or manipulate the data.

2.4 Structured Chart



2.5 Summary

This chapter covers the system's problem analysis, which comprises problem description—where the problem is broken down into smaller, more manageable chunks—problem decomposition, where we offer a solution—and structured chart—which shows each module and its associated subfunctions. The relationships between the modules are depicted in this structured chart using a tree structure. The system design will be the subject of the following chapter.

CHAPTER 3: DESIGN

3.1 Introduction

This chapter include the design of the system that covers flowchart, pseudocode, the Entity Relationship Diagram (ERD), data dictionary of the system and the interface design of the system.

3.2 Flowchart

The system's flowchart is broken up into modules. This is due to the large number of sub flowcharts (functions) that illustrate the flow of the function within the module. The flowcharts can be found below:

3.2.1 Main()

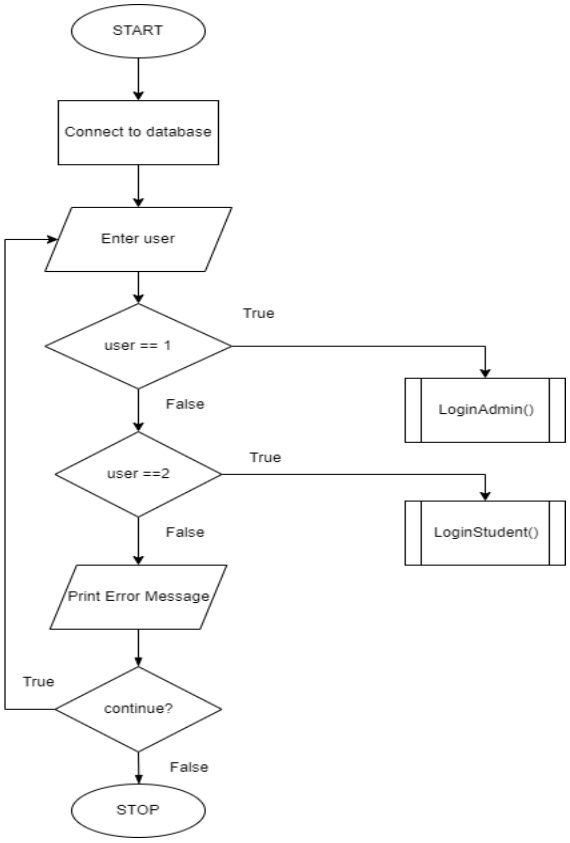
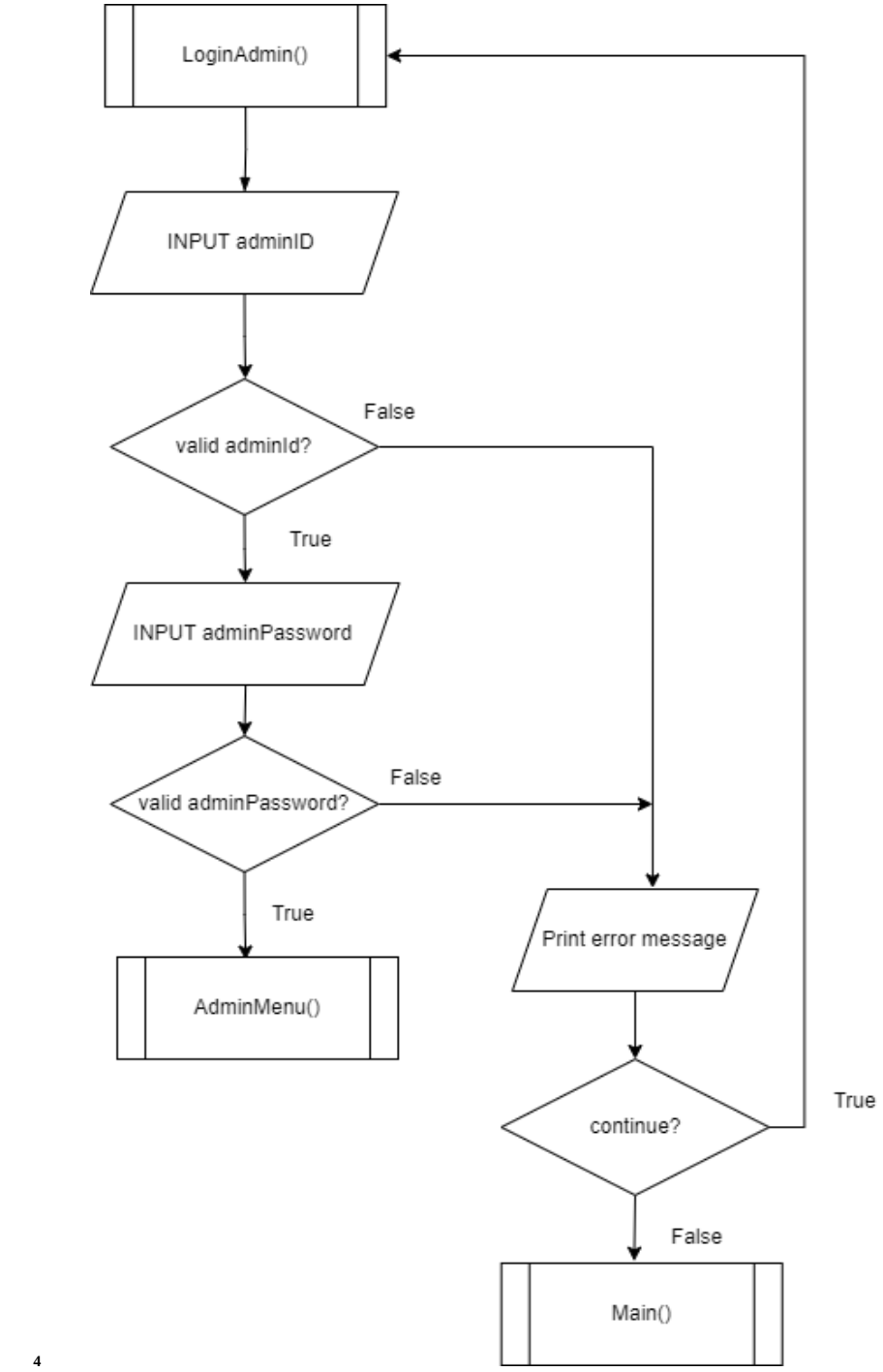


Figure 3.2.1.1: Flowchart of main() function

3.2.3 Login Admin module



4

Figure 3.2.2.1: Flowchart of LoginAdmin() function

3.2.3 AdminMenu modul

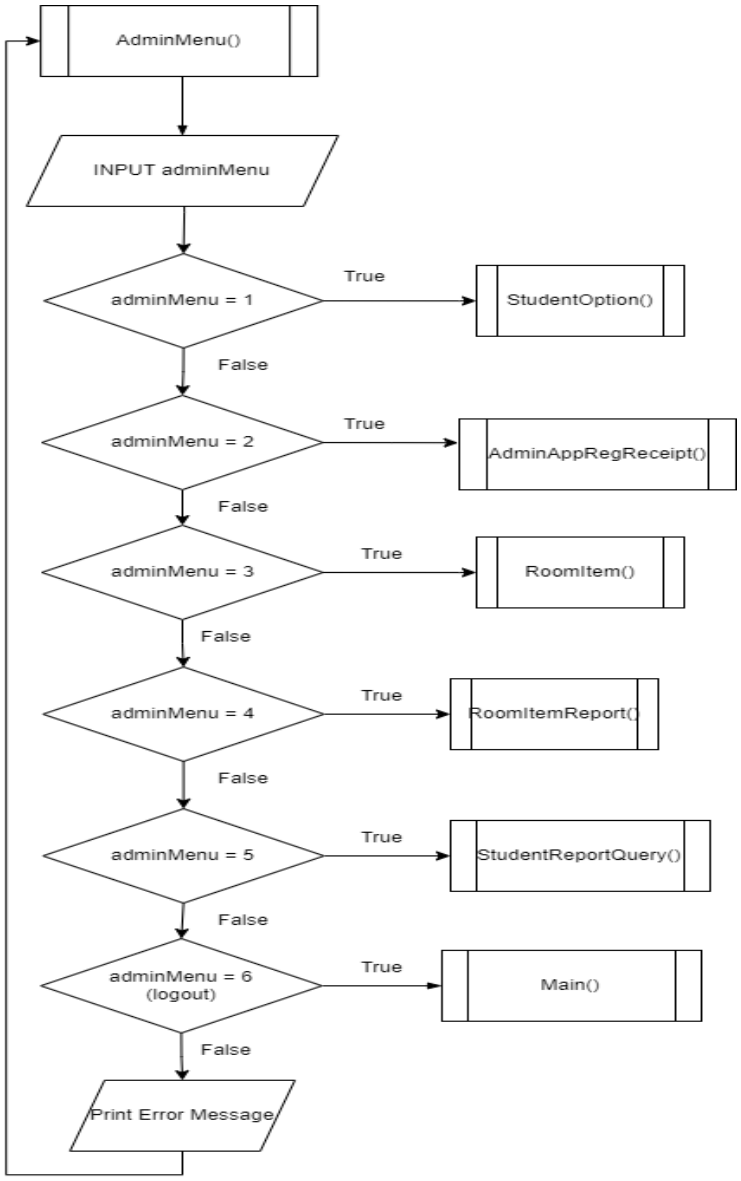


Figure 3.2.3.1: Flowchart of AdminMenu() function

3.2.4 StudentOption()

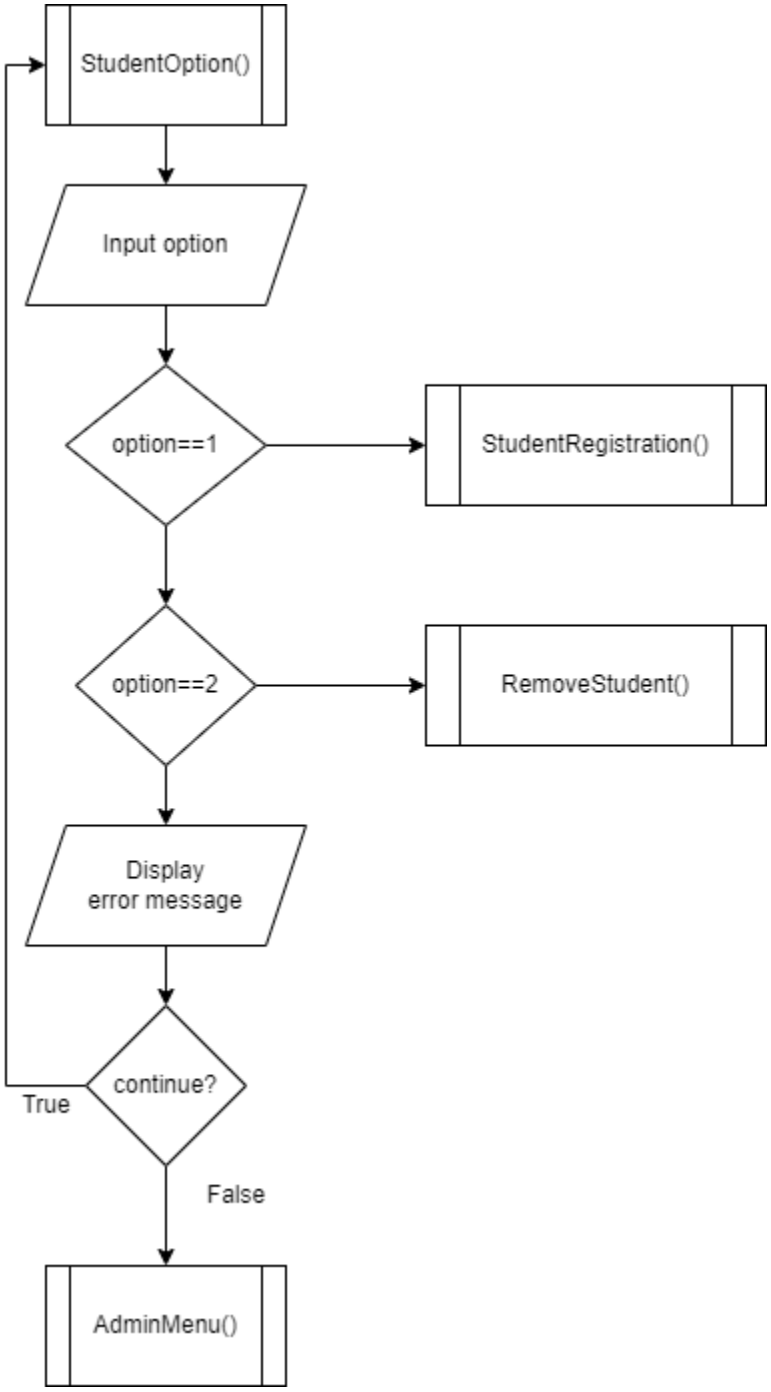


Figure 3.2.4.1: Flowchart of StudentOption() function

3.2.5 RegisterStudent()

Figure 3.2.5.1 Flowchart Of RegisterStudent() function

3.2.7 RemoveStudent()

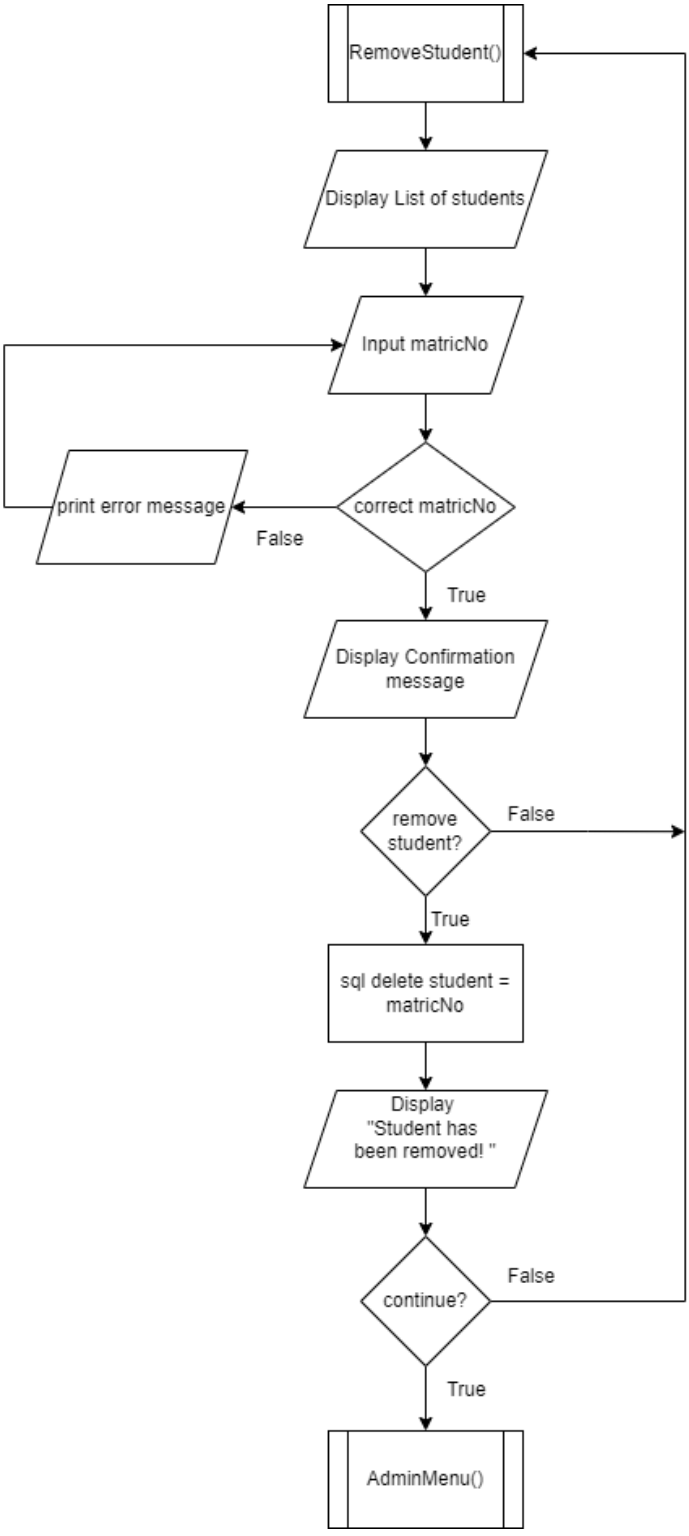


Figure 3.2.7.1 Flowchart Of RemoveStudent() function

3.2.8 AdminAppRegReceipt()

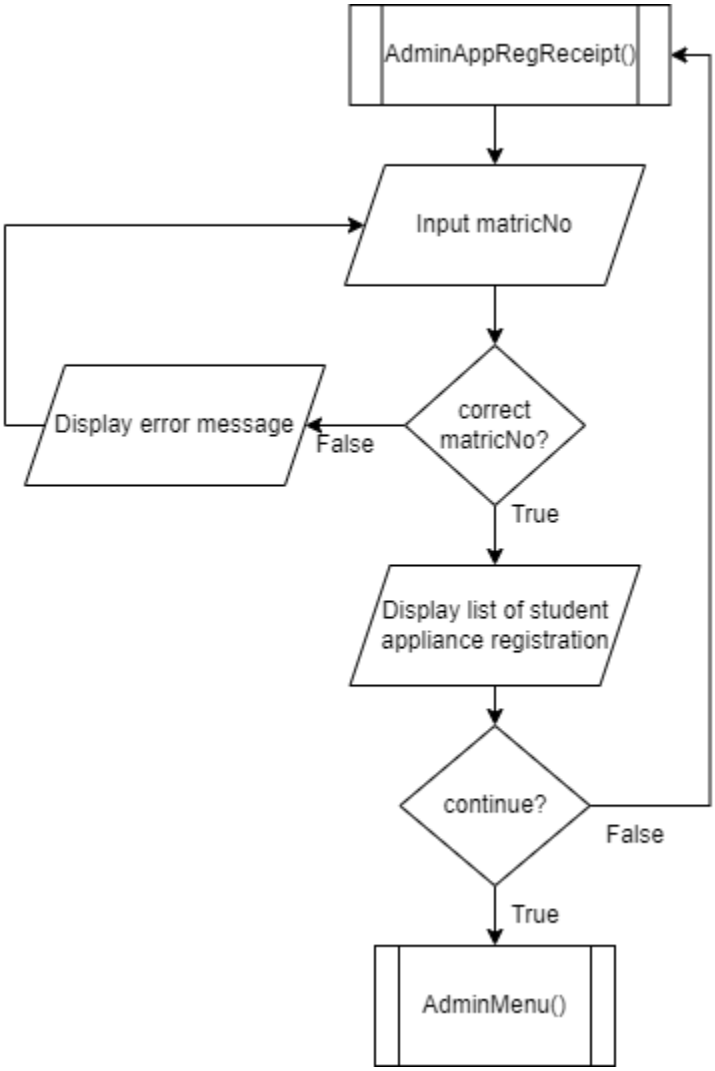


Figure 3.2.8.1 Flowchart Of AdminAppRegReceipt() function

3.2.9 RoomItem()

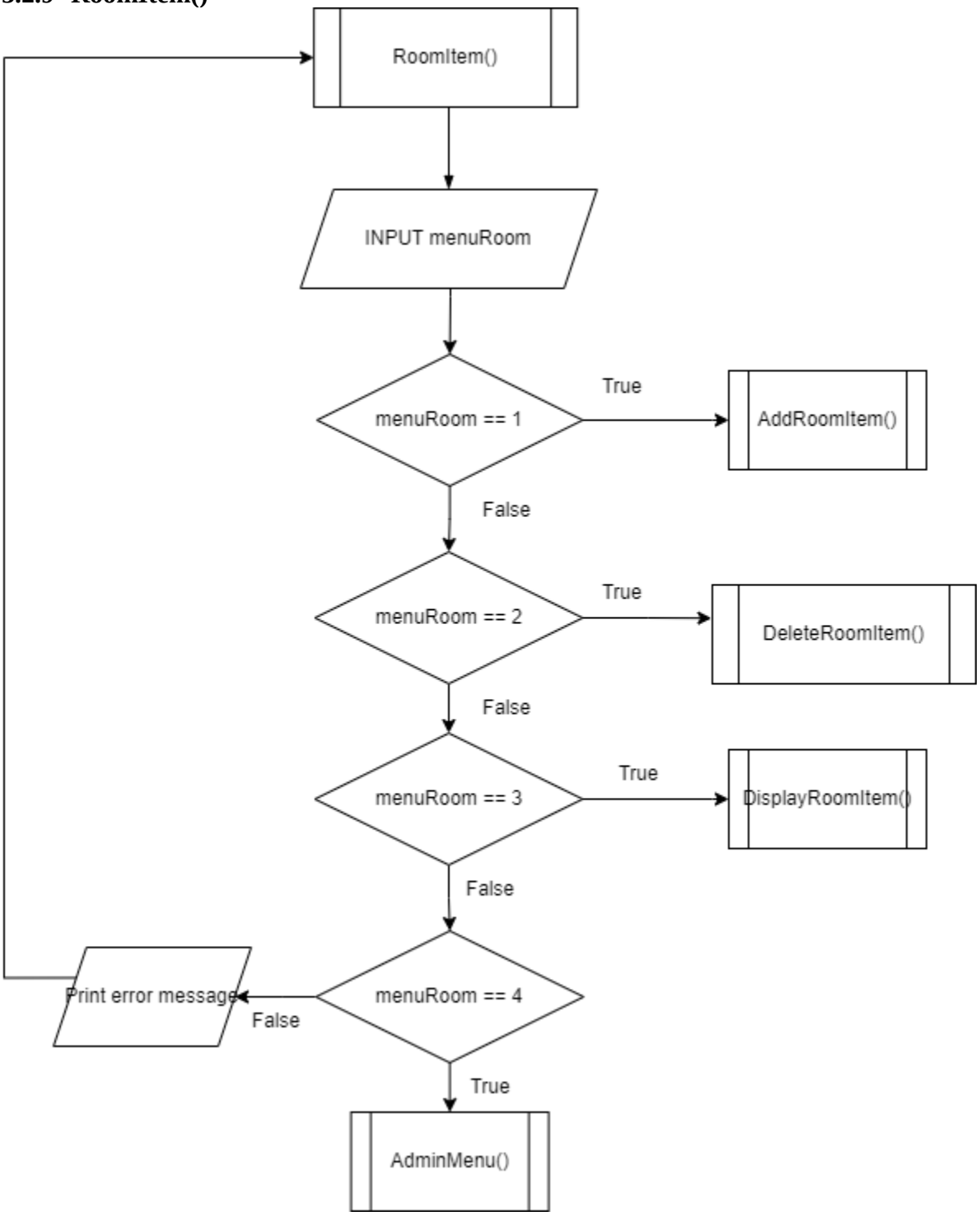


Figure 3.2.9.1 Flowchart Of RoomItem() function

3.2.10 AddRoomItem()

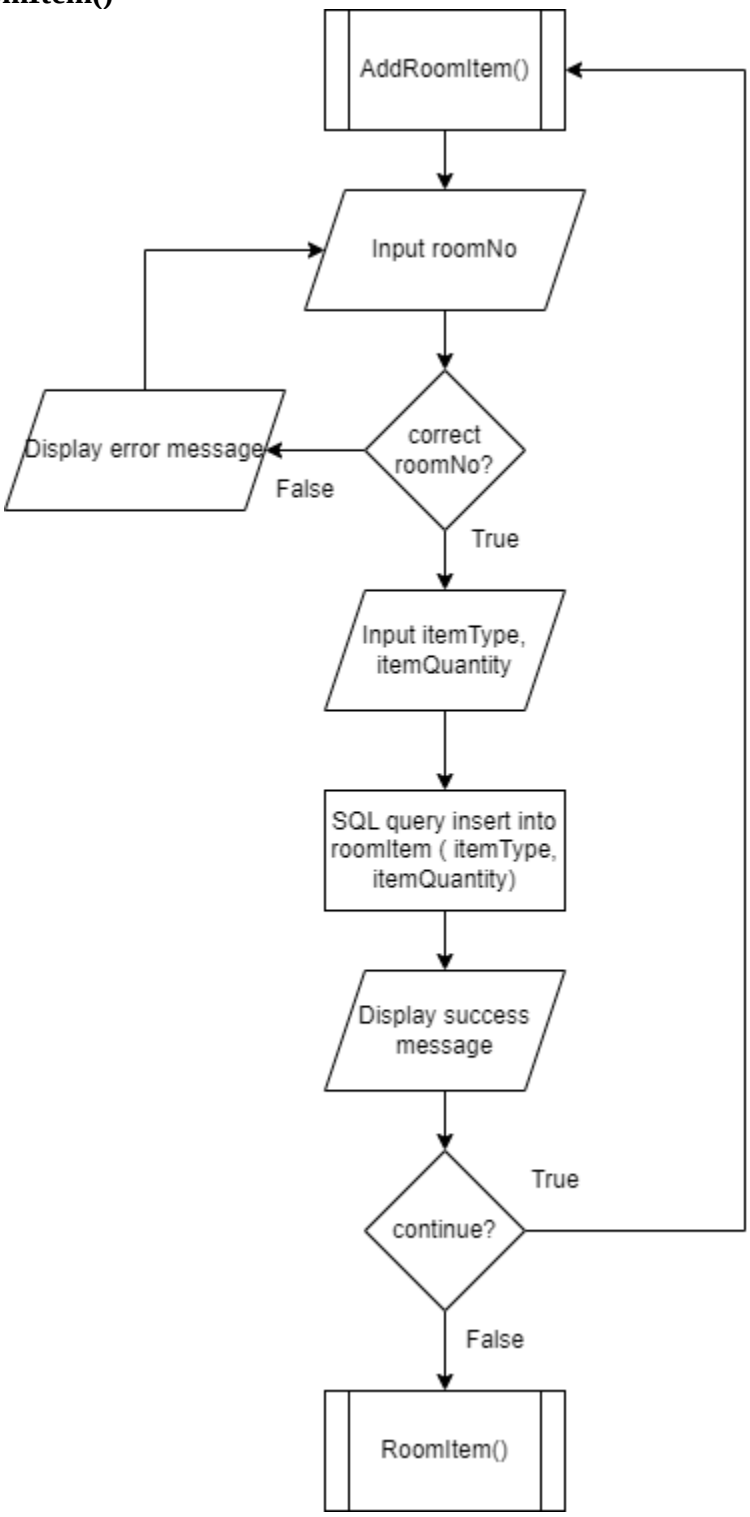


Figure 3.2.10 Flowchart Of AddRoomItem() function

3.2.11 DisplayRoomItem()

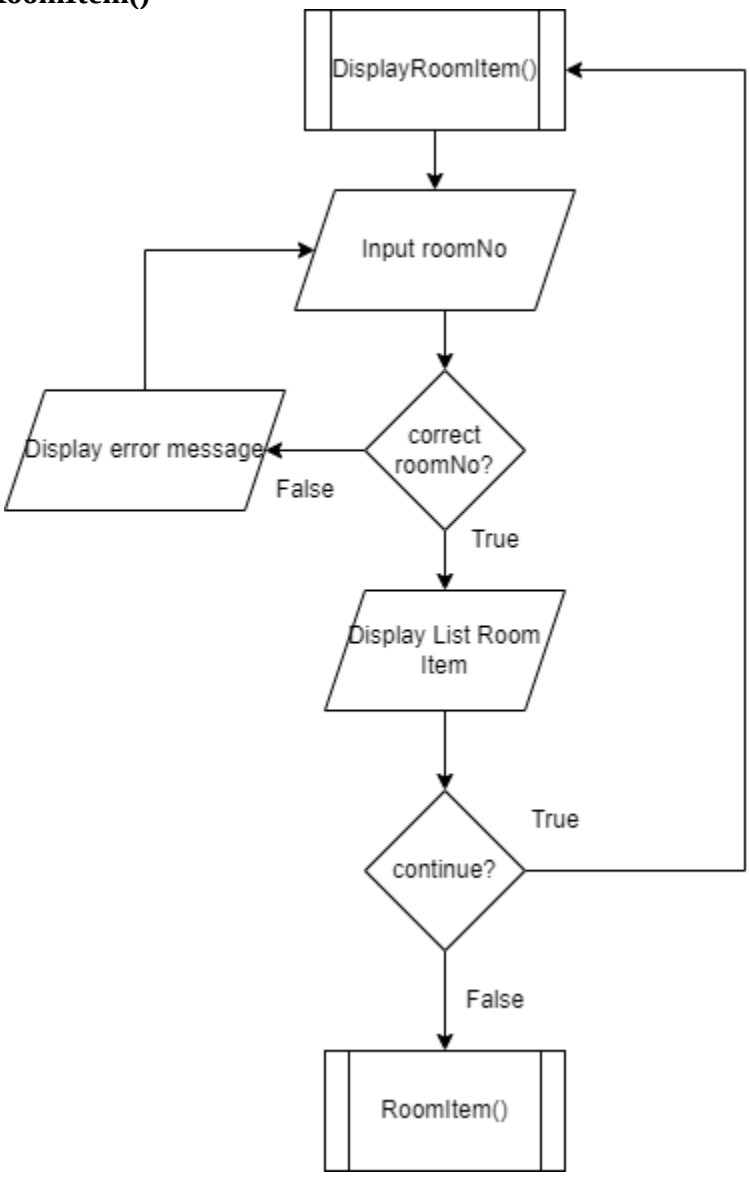


Figure 3.2.11.1 Flowchart Of DisplayRoomItem() function

3.2.12 DeleteRoomItem()

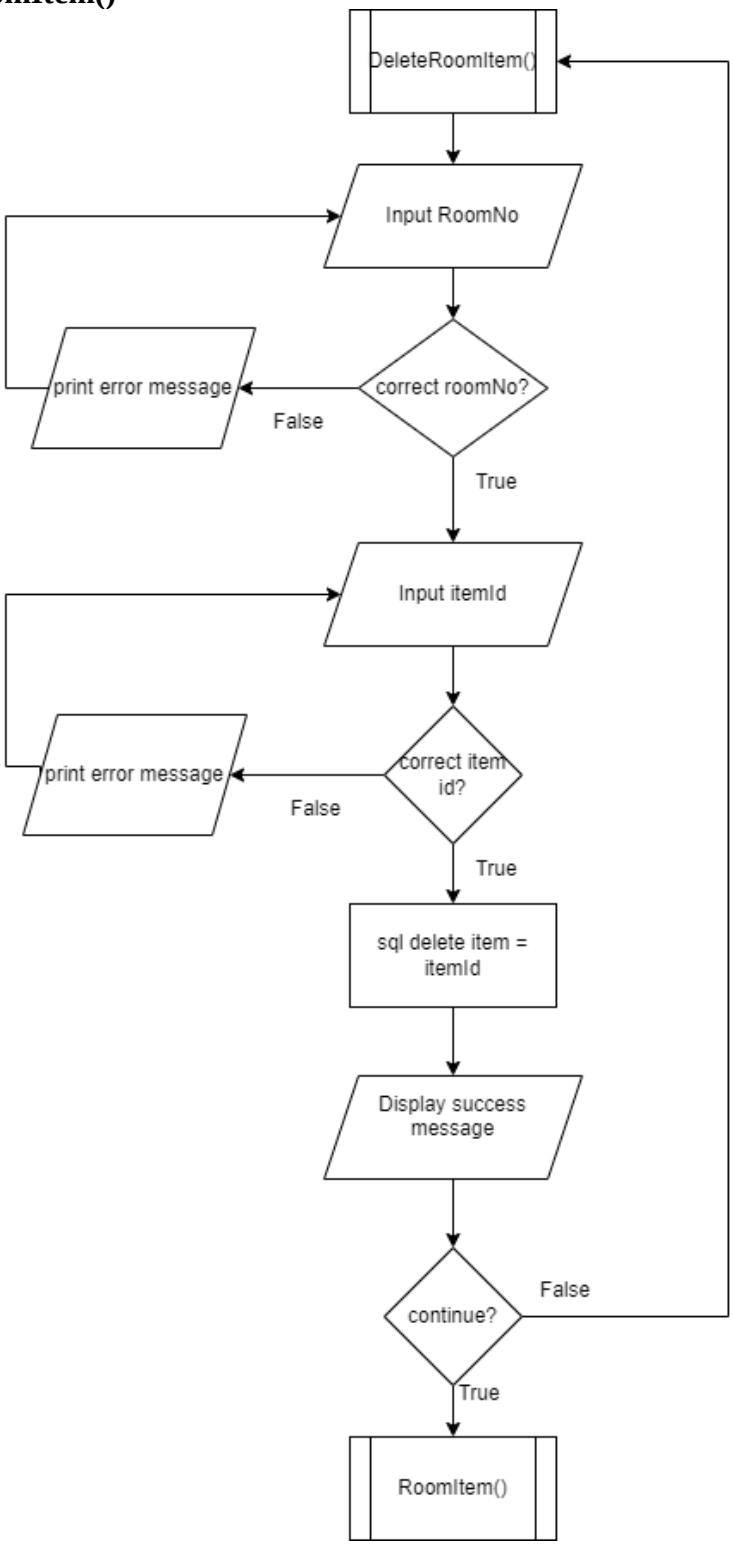


Figure 3.2.12.1 Flowchart Of DeleteRoomItem() function

3.2.13 RoomItemReport()

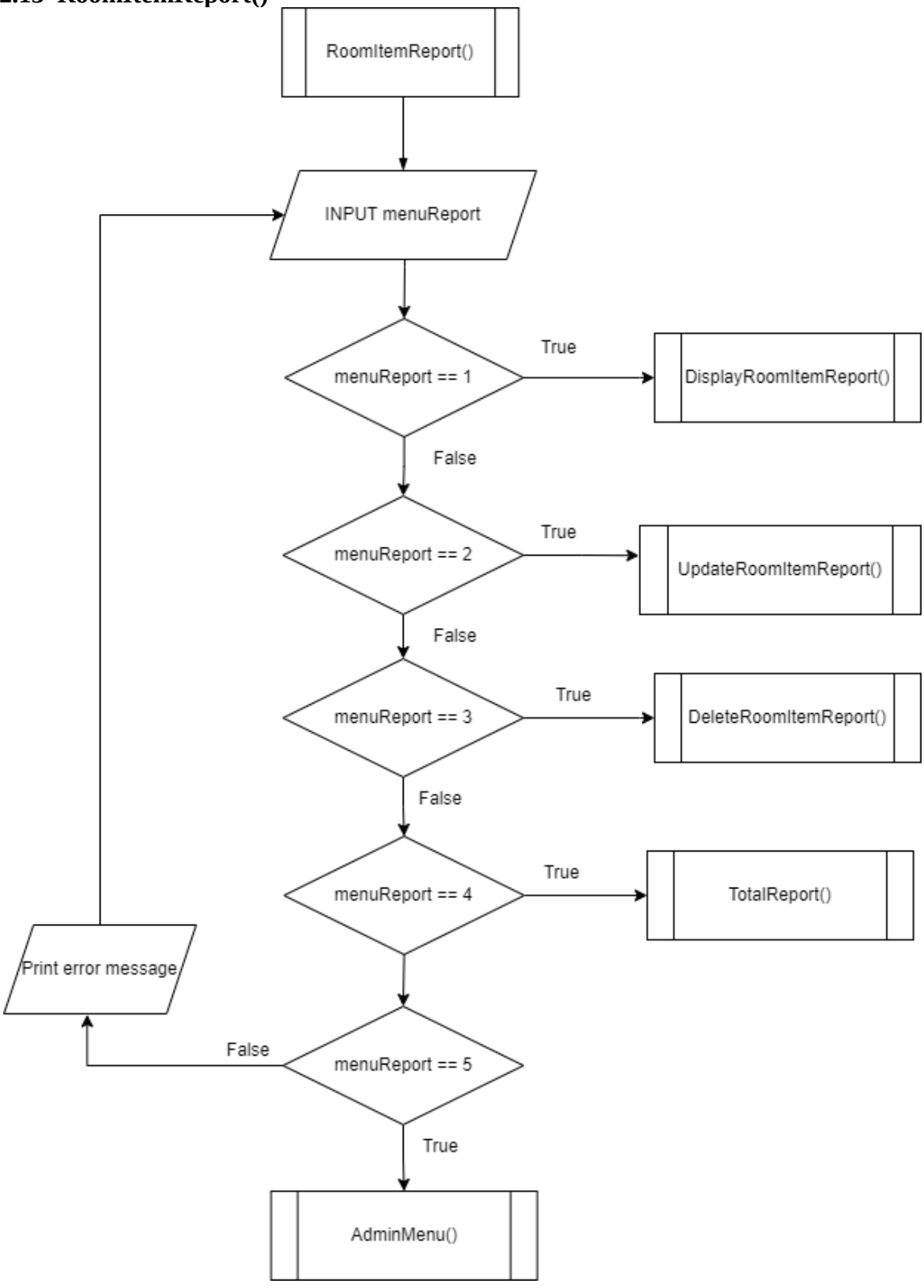


Figure 3.2.13.1 Flowchart Of RoomItemReport() function

3.2.14 DisplayRoomItemReport()

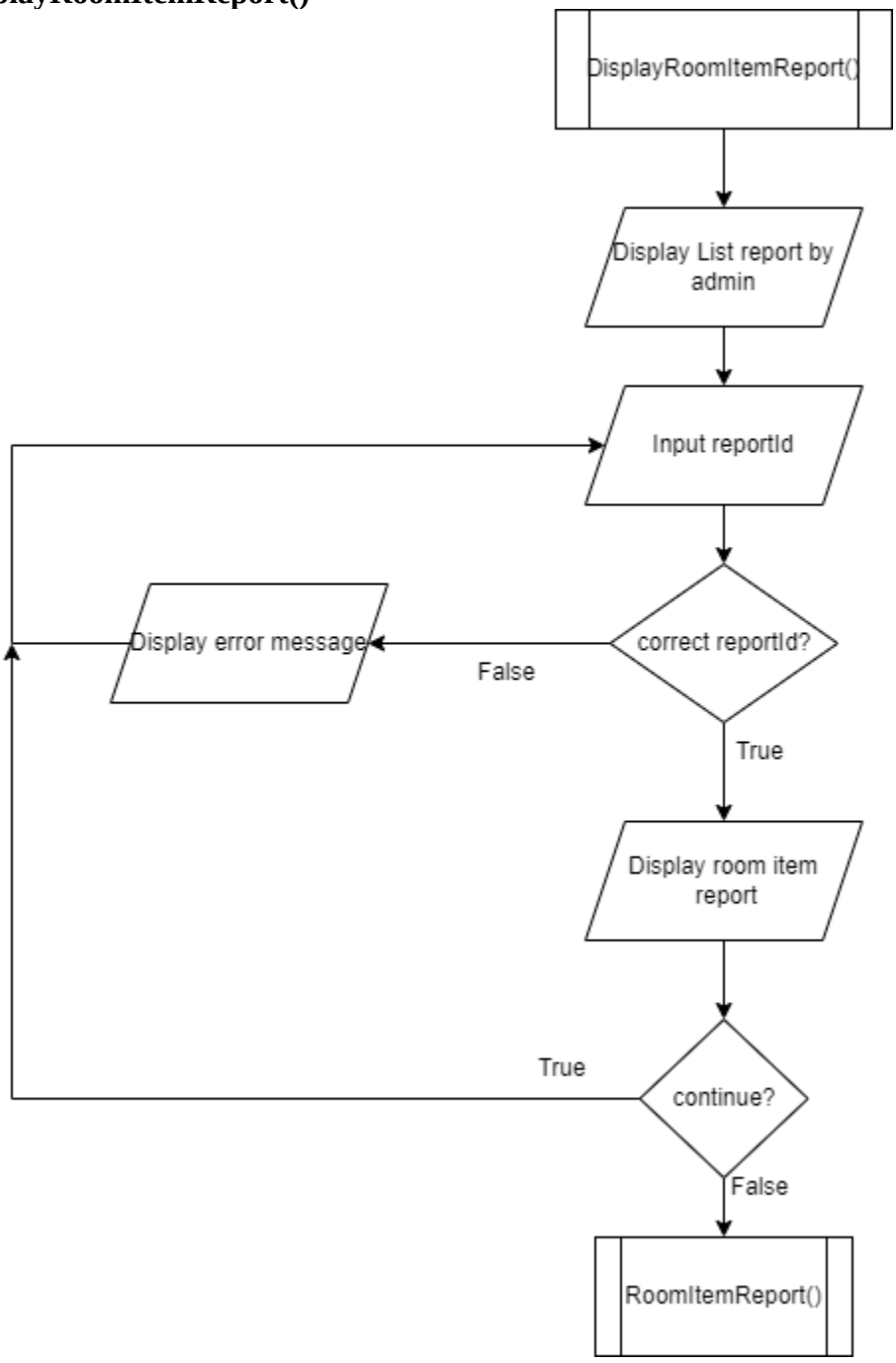


Figure 3.2.14.1 Flowchart Of DisplayRoomItemReport() function

3.2.15 UpdateRoomItemReport()

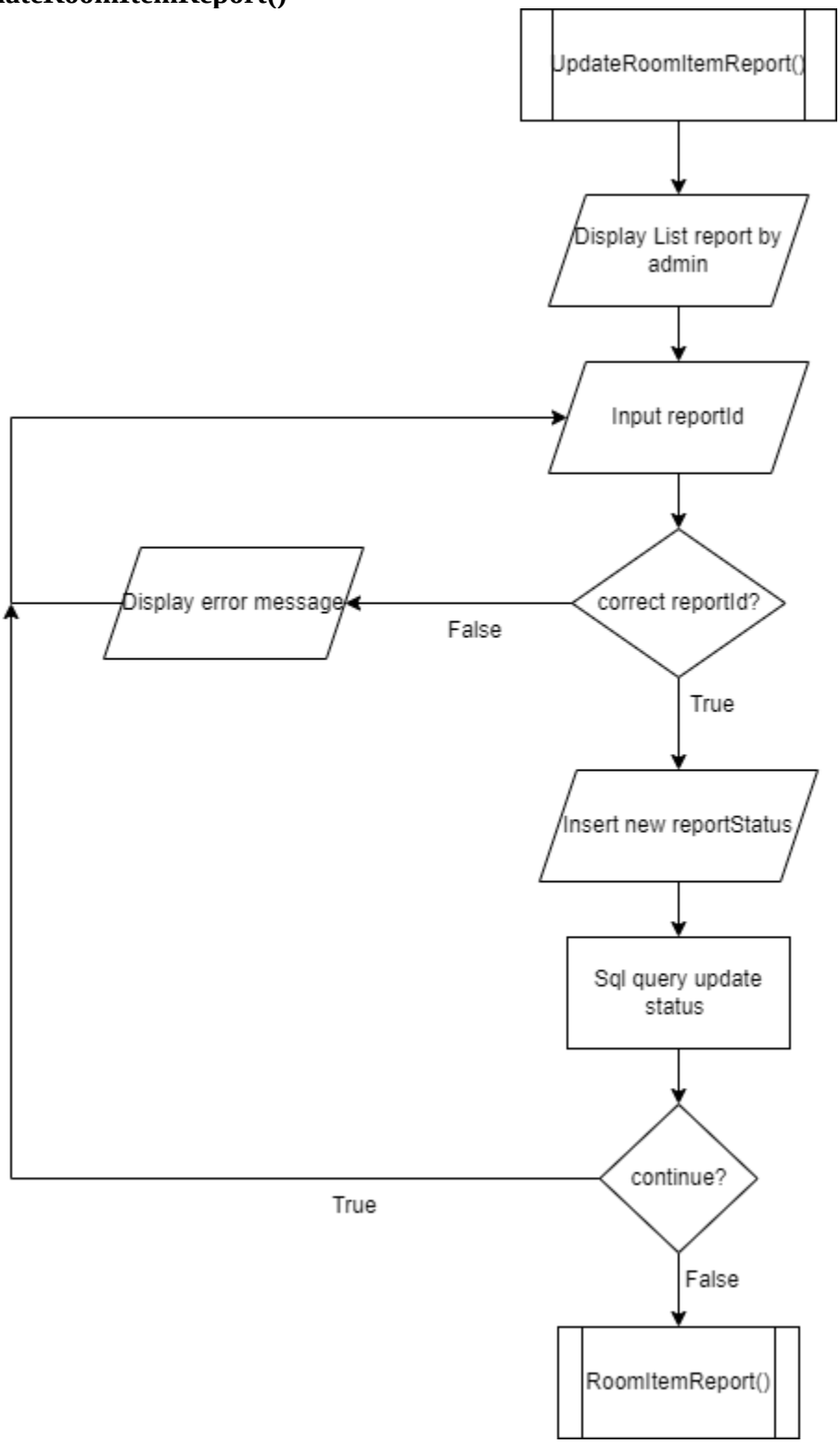


Figure 3.2.15.1 Flowchart Of UpdateRoomItemReport() function

3.2.16 DeleteRoomItemReport()

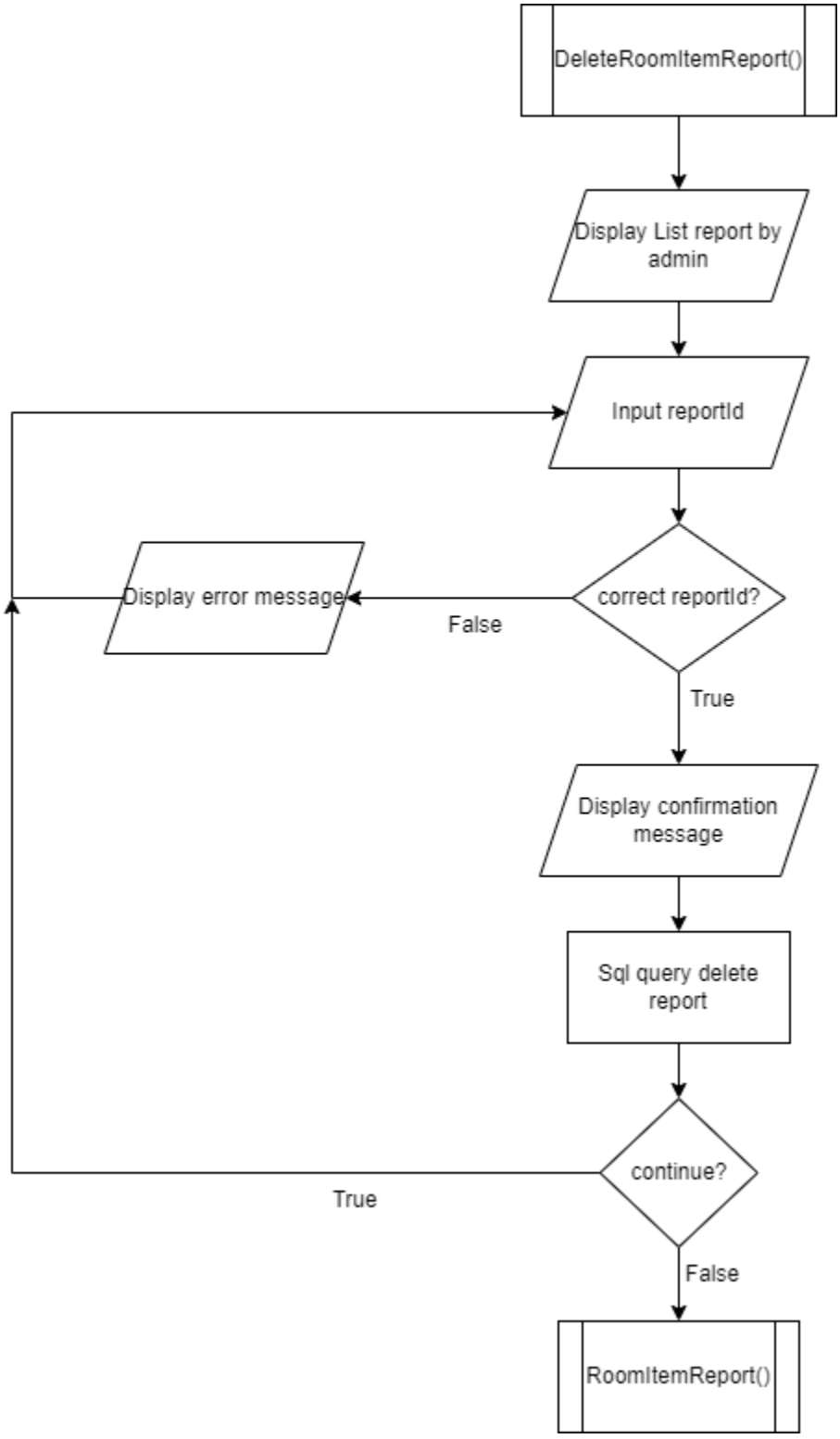


Figure 3.2.16.1 Flowchart Of DeleteRoomItemReport() function

3.2.17 TotalReport()

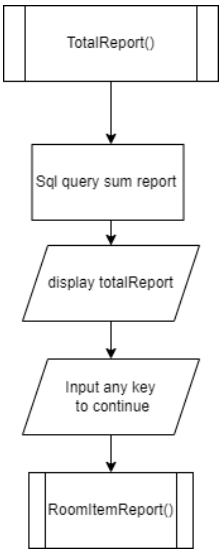


Figure 3.2.17.1 Flowchart Of TotalReport() function

3.2.18 StudentReportQuery()

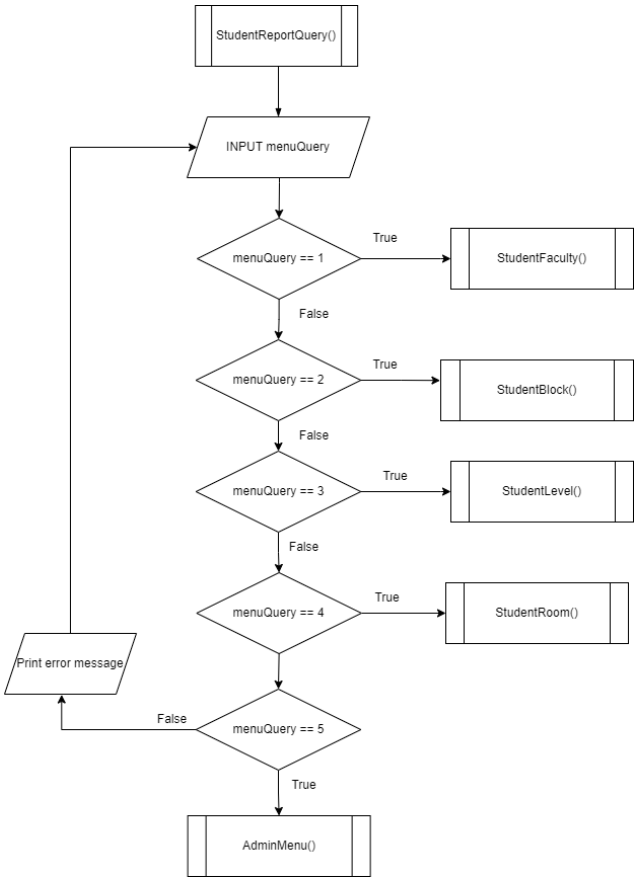


Figure 3.2.18.1 Flowchart Of StudentReportQuery() function

3.2.19 StudentFaculty()

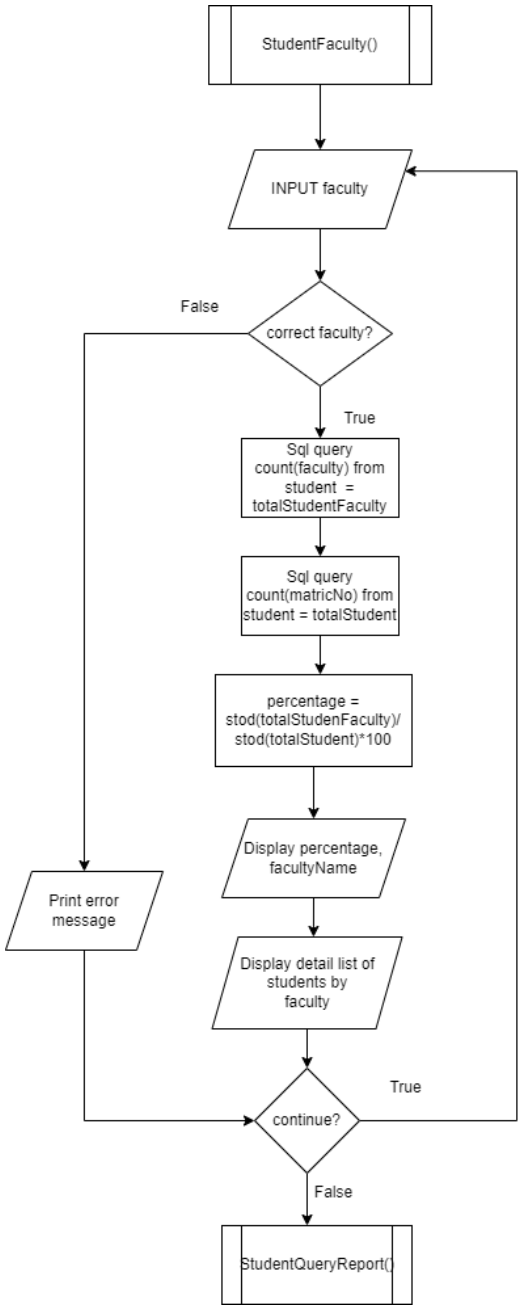


Figure 3.2.19.1 Flowchart Of StudentFaculty() function

3.2.20 StudentBlock()

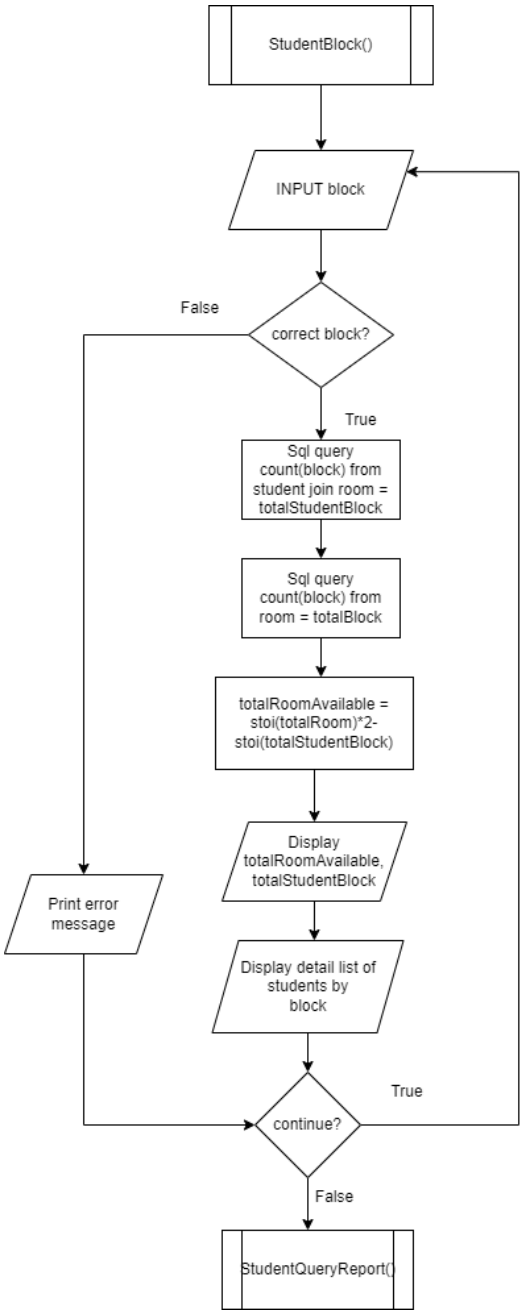


Figure 3.2.20.1 Flowchart Of StudentBlock() function

3.2.21 StudentLevel()

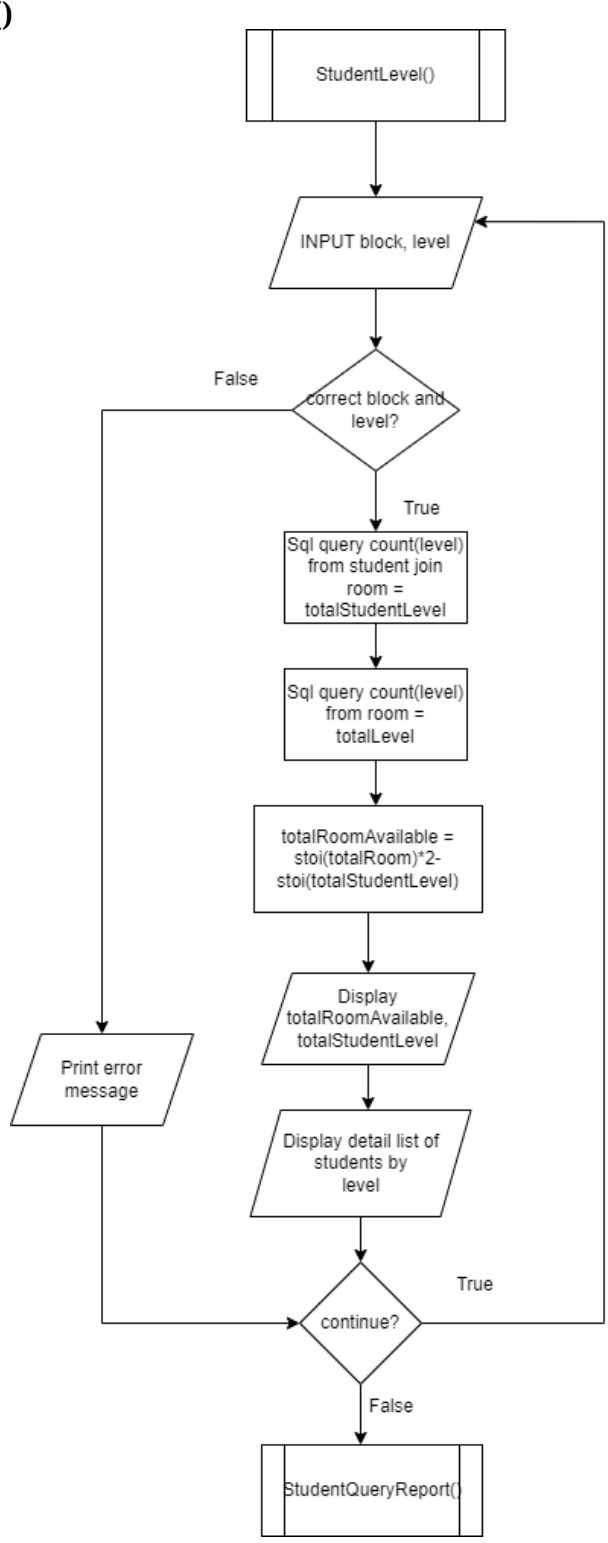


Figure 3.2.21.1 Flowchart Of StudentLevel() function

3.2.22 StudentRoom()

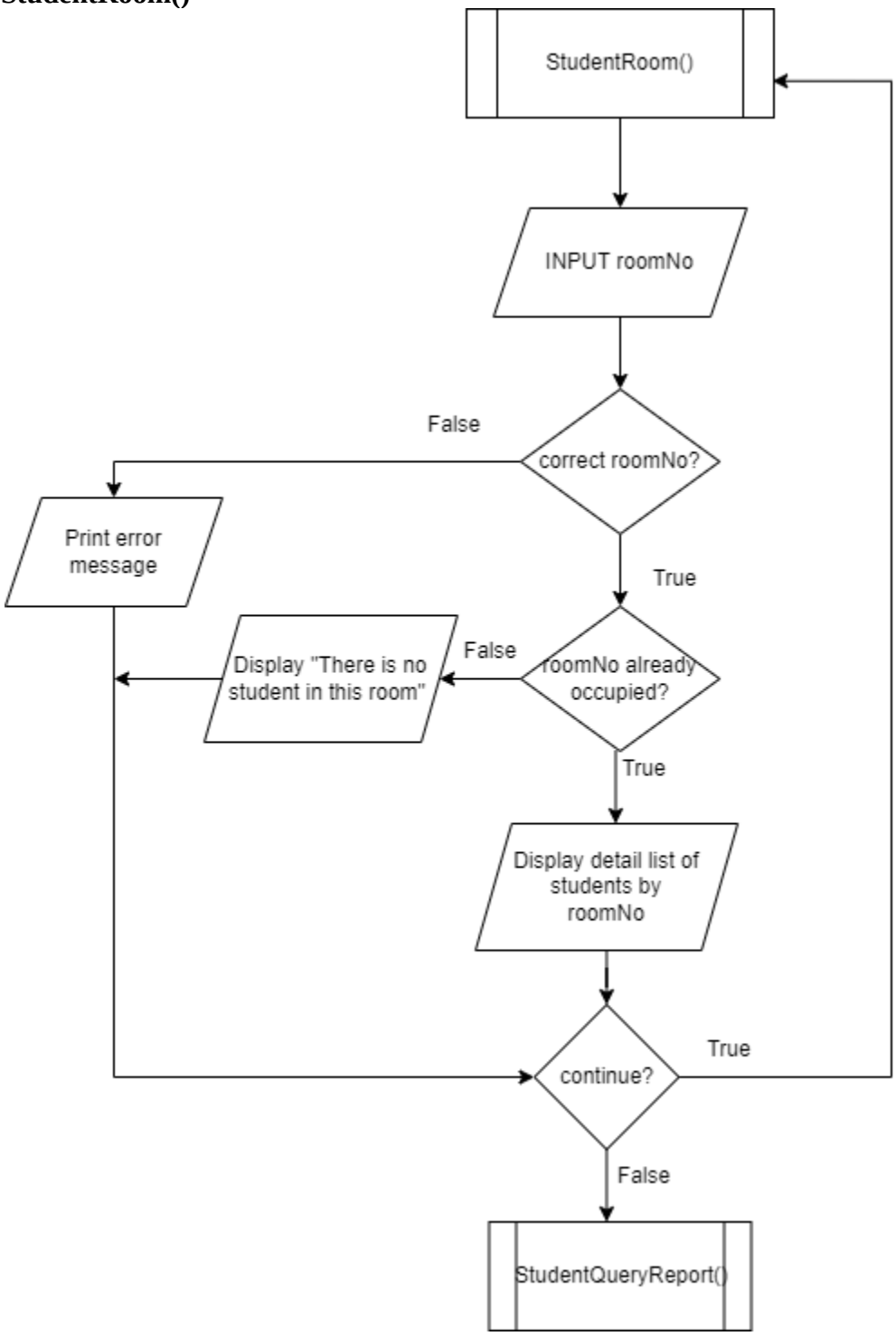


Figure 3.2.22.1 Flowchart Of StudentRoom() function

3.2.23 LoginStudent()

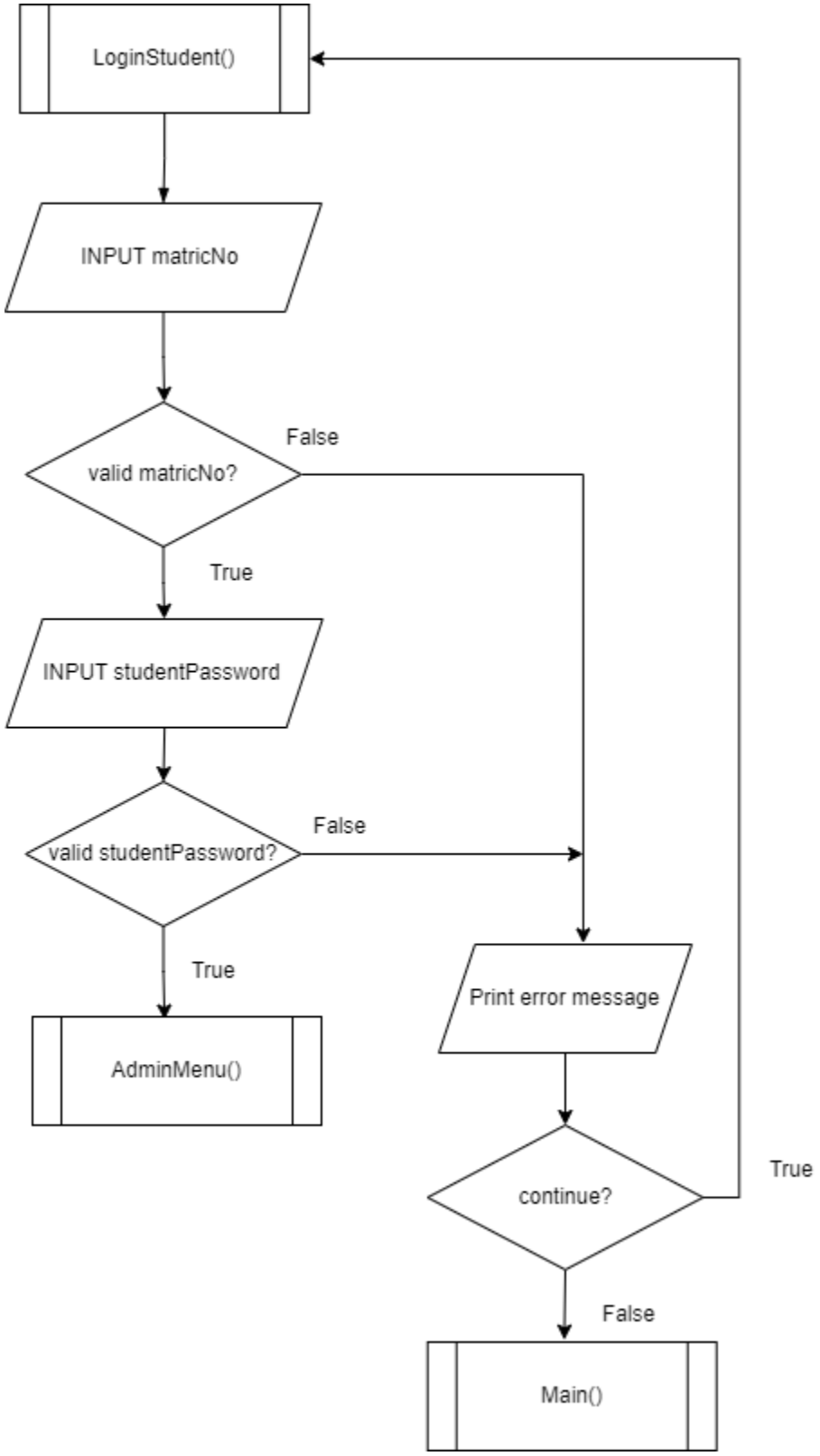


Figure 3.2.23.1: Flowchart of LoginStudent() function

3.2.24 StudentMenu()

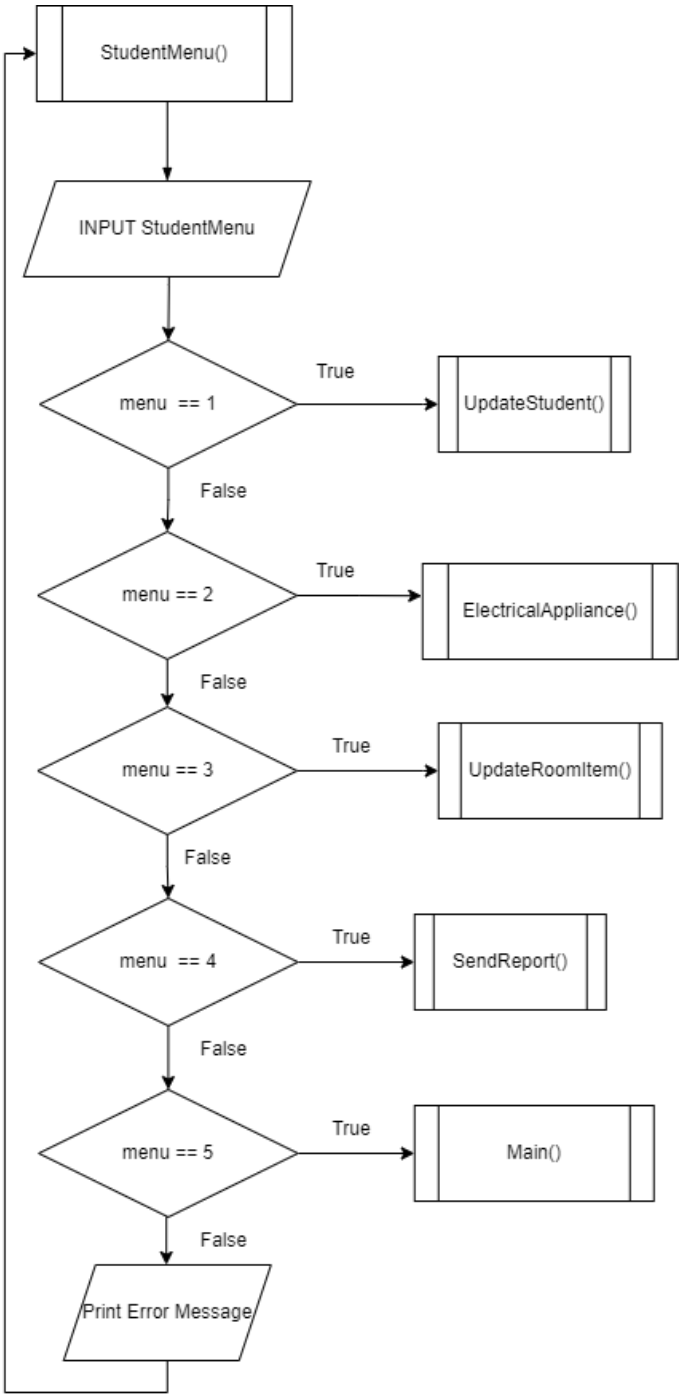


Figure 3.2.24.1: Flowchart of StudentMenu() function

3.2.25 UpdateStudent()

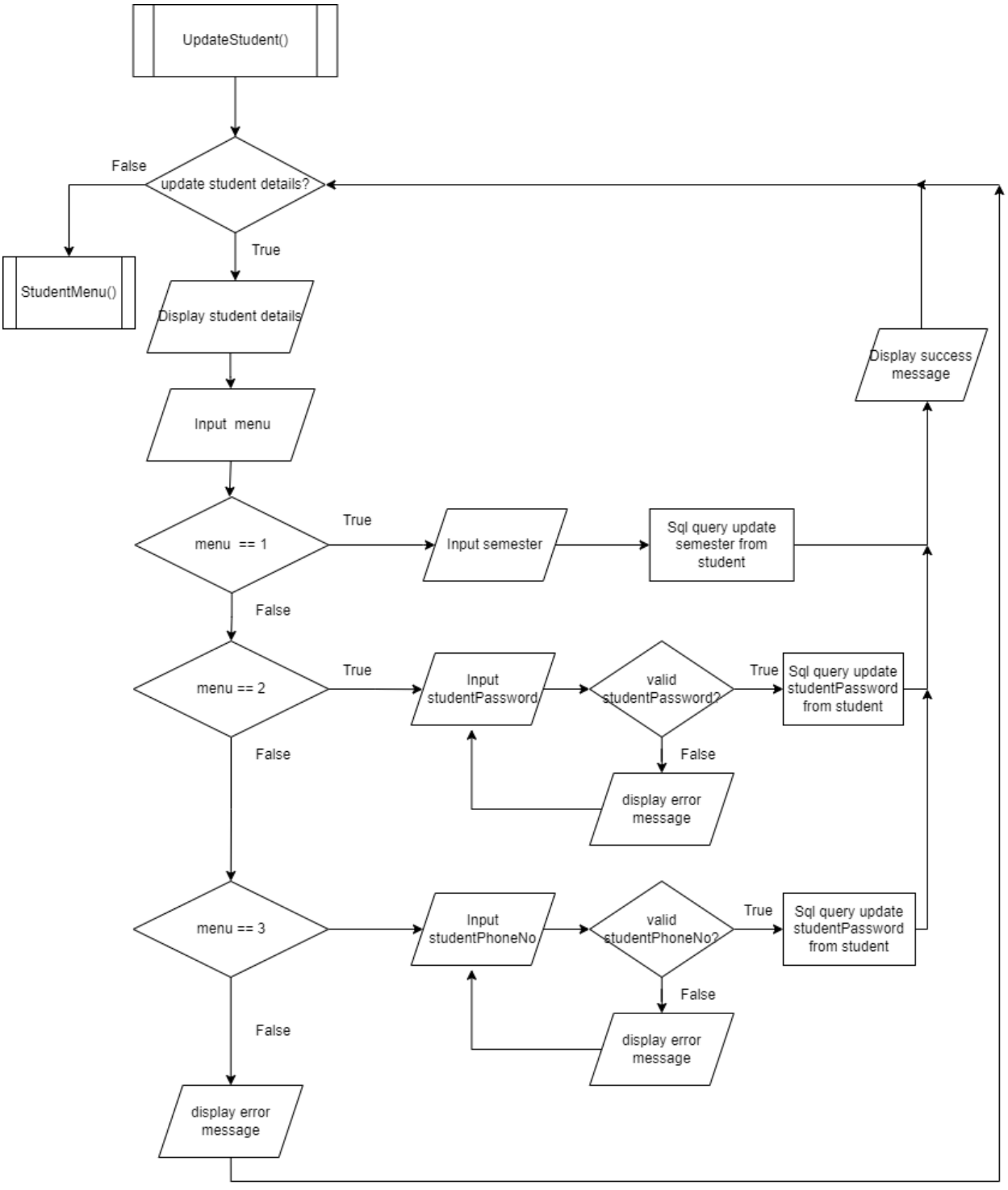


Figure 3.2.25.1 Flowchart Of UpdateStudent() function

3.2.26 ElectricalAppliance()

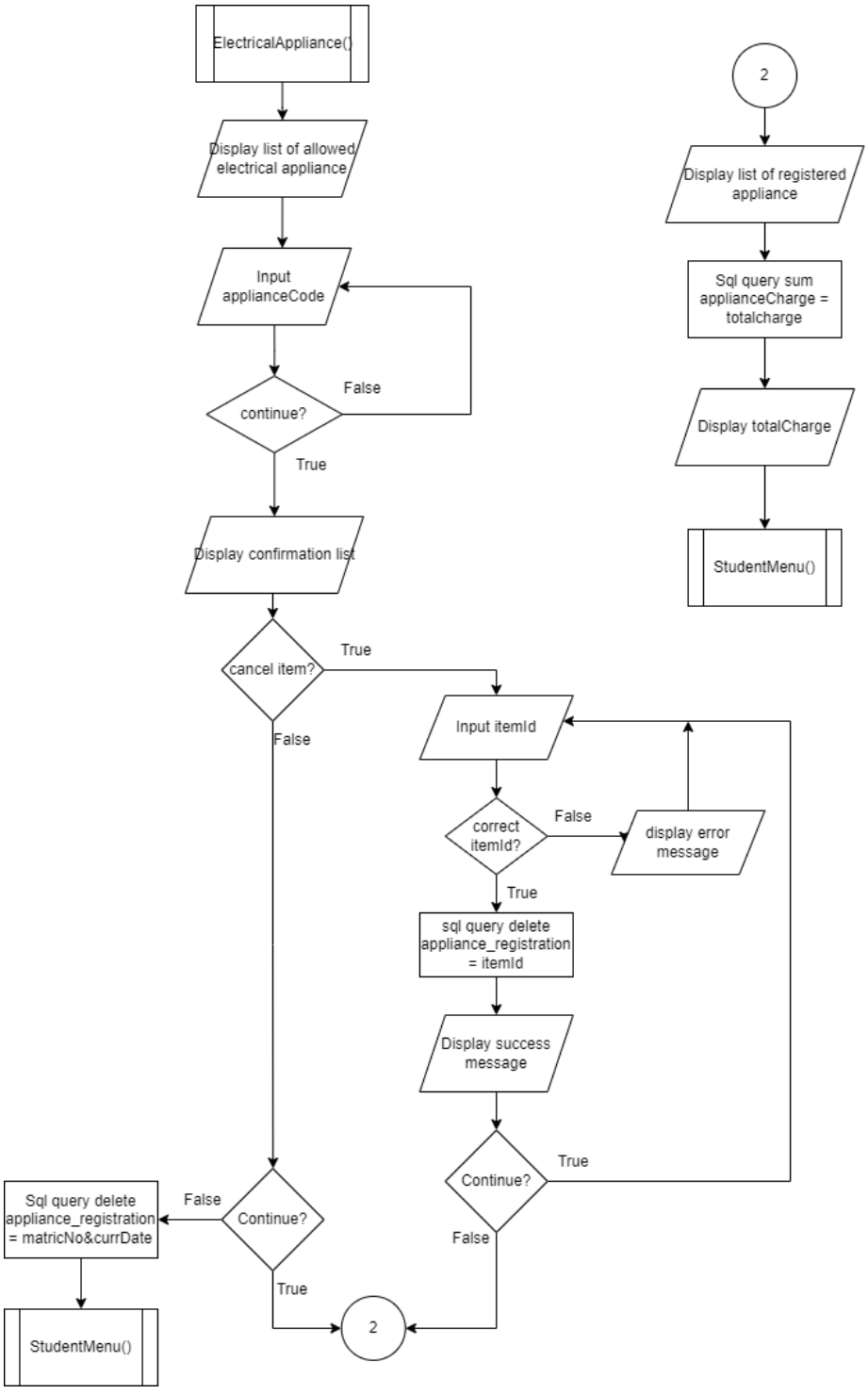


Figure 3.2.26.1 Flowchart Of ElectricalAppliance() function

3.2.27 UpdateRoomItem()

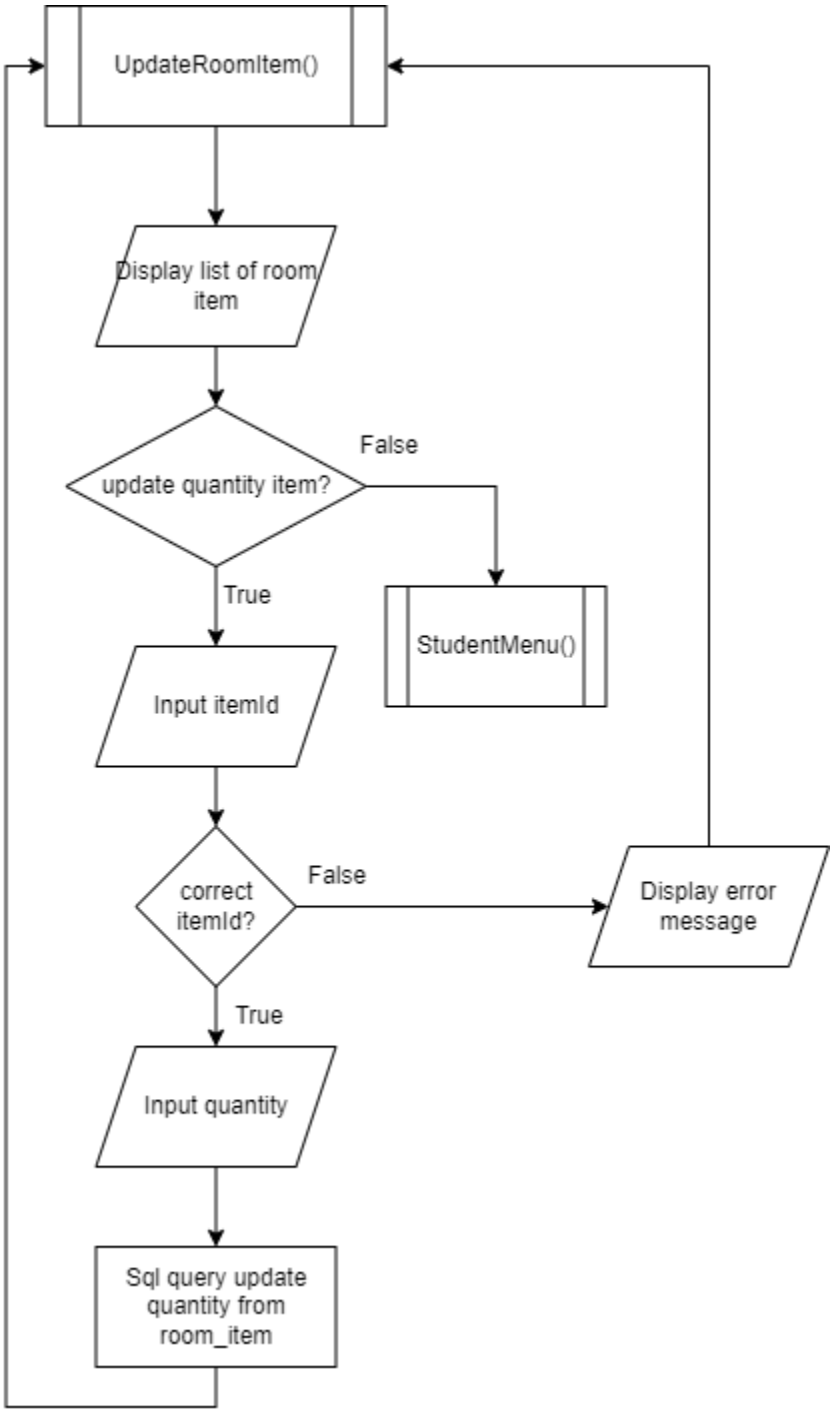


Figure 3.2.27.1 Flowchart Of UpdateRoomItem() function

3.2.28 SendReport()

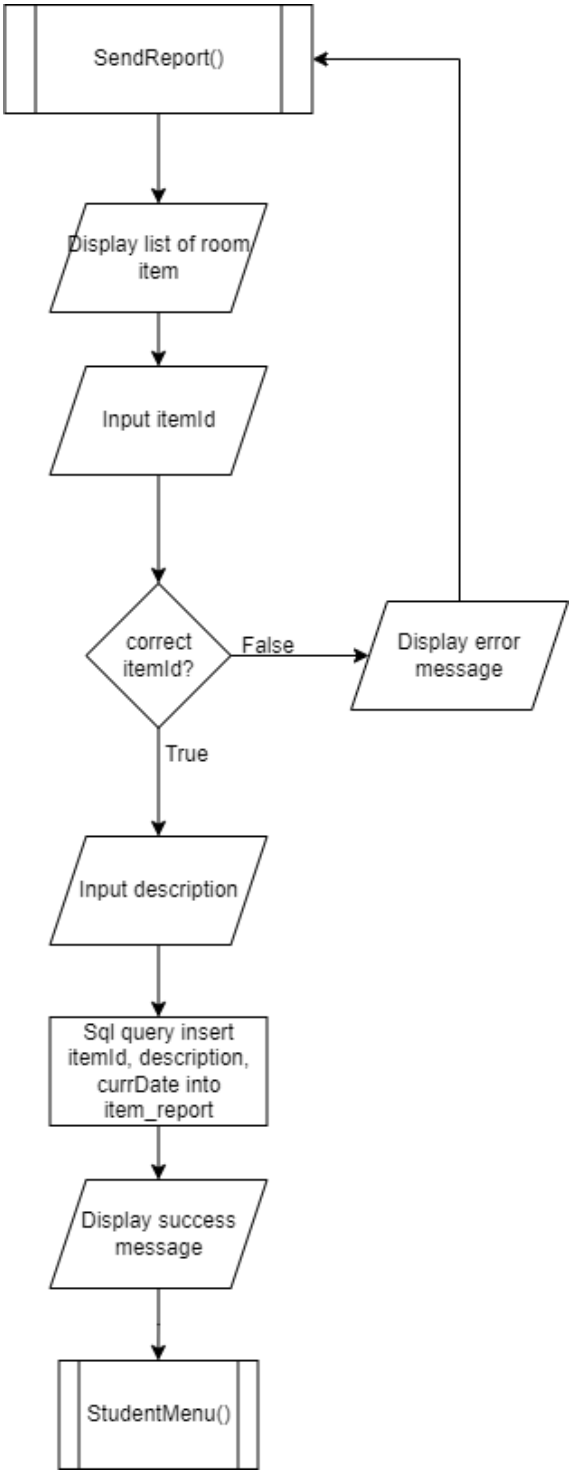


Figure 3.2.28.1 Flowchart Of SendReport() function

3.3 Entity Relationship Diagram

The relationship between the database's tables is depicted in this diagram.

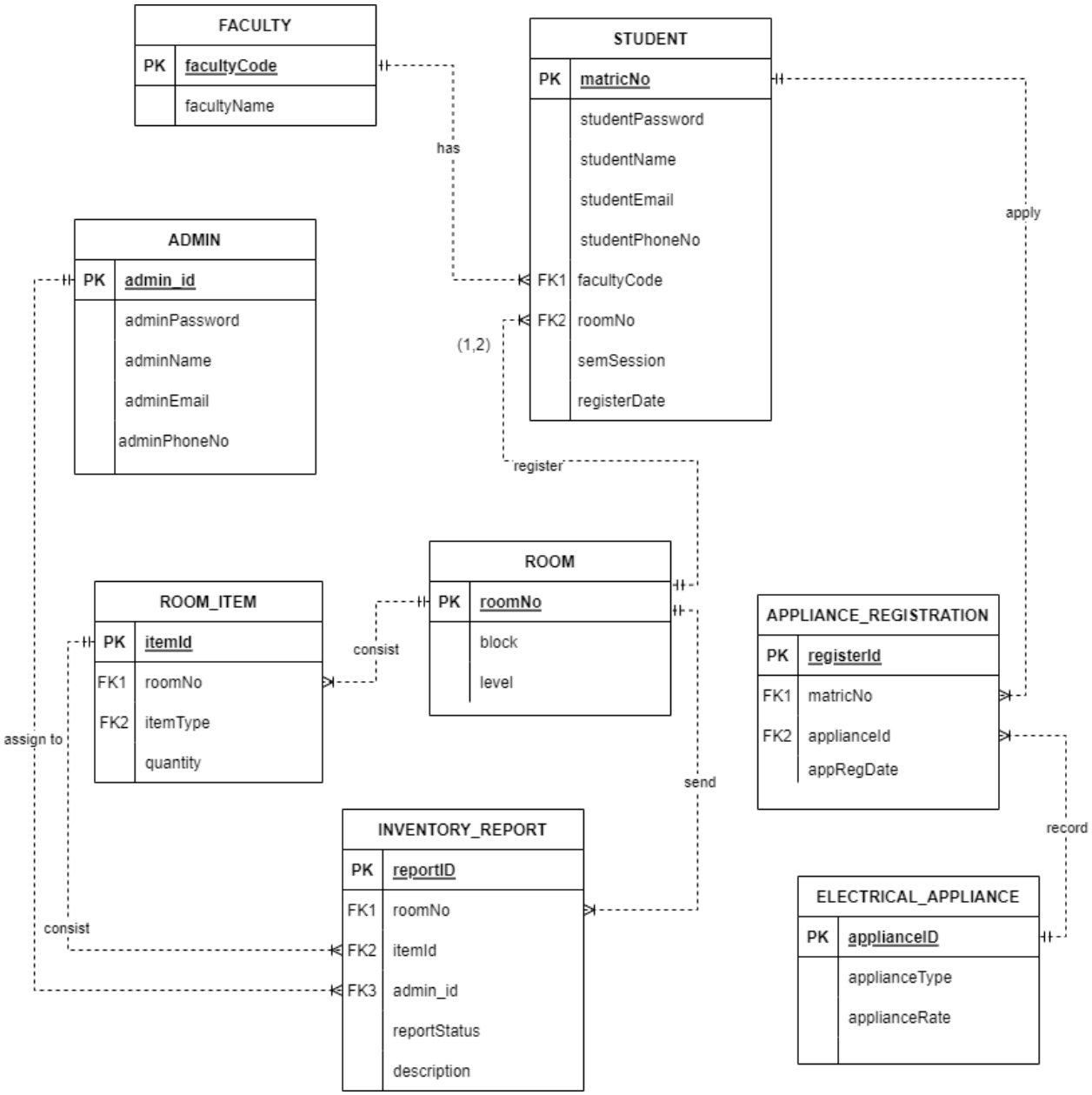


Figure 4.1.1 Entity Relationship Diagram

The business rule for the entity relationship are :

- One room can be registered by maximum 2 students.
- A faculty has many students.
- Each student can apply for many electrical appliance registrations.
- Each electrical appliance registration recorded many electrical appliances.
- A room can send many inventories report.
- Many inventories report consists of a room item.
- An admin is assigned to many inventories report.

3.4 Data Dictionary

3.4.1 ADMIN

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/ FK	FK TABLE
admin_id	varchar (10)	XXXX	YES	PK	
adminPassword	varchar (8)	XXXX	-	-	
adminName	varchar (50)	XXXX	-	-	
adminPhoneNo	varchar (11)	XXXX	-	-	
adminEmail	varchar (30)	XXX@gmail.com	-	-	

Table 3.4.1.1 Table of ADMIN data dictionary

3.4.2 STUDENT

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	FK TABLE
matricNo	varchar (10)	XXXX	YES	PK	
facultyCode	varchar (10)	XXXX	-	FK	FACULTY
studentPassword	varchar (8)	XXXX	-	-	
studentName	varchar (50)	XXXX	-	-	
studentEmail	varchar (30)	XXX@student.com	-	-	
studentPhoneNo	varchar (11)	XXXX	-	-	
semester	varchar (5)	XXX	-	-	
dateRegister	date	YYYY-MM-DD	-	-	
roomNo	varchar (10)	X-XX-XX	-	FK	ROOM

Table 3.4.2.1 Table of STUDENT data dictionary

3.4.3 ROOM

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	FK TABLE
roomNo	varchar (10)	XXXX	YES	PK	
block	varchar (1)	X	-	-	
level	varchar (1)	X	-	-	

Table 3.4.3.1 Table of STUDENT data dictionary

3.4.4 FACULTY

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	FK TABLE
facultyCode	varchar (5)	XXXX	YES	PK	
facultyName	varchar (100)	XXXX	-	-	

Table 3.4.4.1 Table of FACULTY data dictionary

3.4.5 ROOM_ITEM

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	FK TABLE
itemId	int (11)	99999	YES	PK	
roomNo	varchar (10)	X-XX-XX	-	FK	ROOM
itemType	varchar (30)	XXXX	-	-	
itemQuantity	int (5)	999	-	-	

Table 3.4.5.1 Table of ROOM_ITEM data dictionary

3.4.6 ELECTRICAL_APPLIANCE

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	PK TABLE
applianceCode	varchar (4)	XXXX	YES	PK	
applianceType	varchar (30)	XXXX	-	-	
applianceCharge	Decimal (10,2)	99.99	-	-	

Table 3.4.6.1 Table of STUDENT data dictionary

3.4.7 APPLIANCE_REGISTRATION

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	PK TABLE
registerId	int (7)	9999999	YES	PK	
applianceCode	varchar (4)	XXXX	-	-	
appRegDate	date	YYYY-MM-DD	-	-	
matricNo	varchar (10)	XXXX	-	FK	STUDENT

Table 3.4.7.1 Table of APPLIANCE_REGISRATION data dictionary

3.8.8 ITEM_REPORT

ATTRIBUTE / NAME	DATA TYPE AND SIZE	FORMAT	UNIQUE	PK/FK	PK TABLE
reportId	varchar (6)	XXXXXX	YES	PK	
admin_id	varchar (10)	XXXX	-	FK	ADMIN
roomNo	varchar (10)	XXXX	-	FK	ROOM
itemId	int (11)	99999	-	FK	ROOM_ITEM
description	varchar (30)	XXXX	-	-	
reportDate	date	YYYY-MM-DD			
status	varchar (30)	XXXX			

Table 3.4.8.1 Table of ITEM_REPORT data dictionary

3.5 System Interface Design

3.5.1 Main Menu

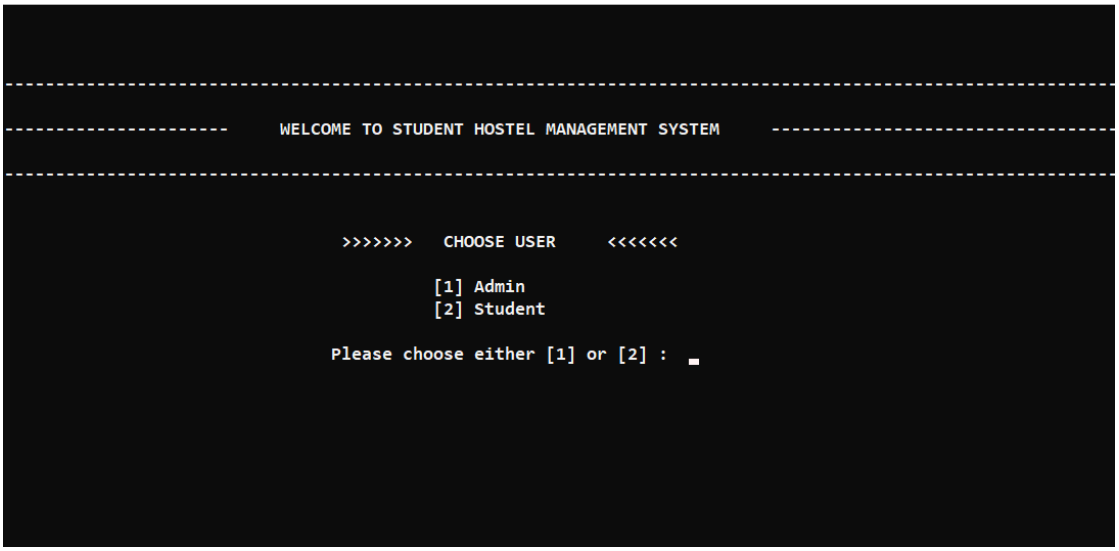


Figure 3.5.1.1 System Interface Main Menu

3.5.2 Admin Login

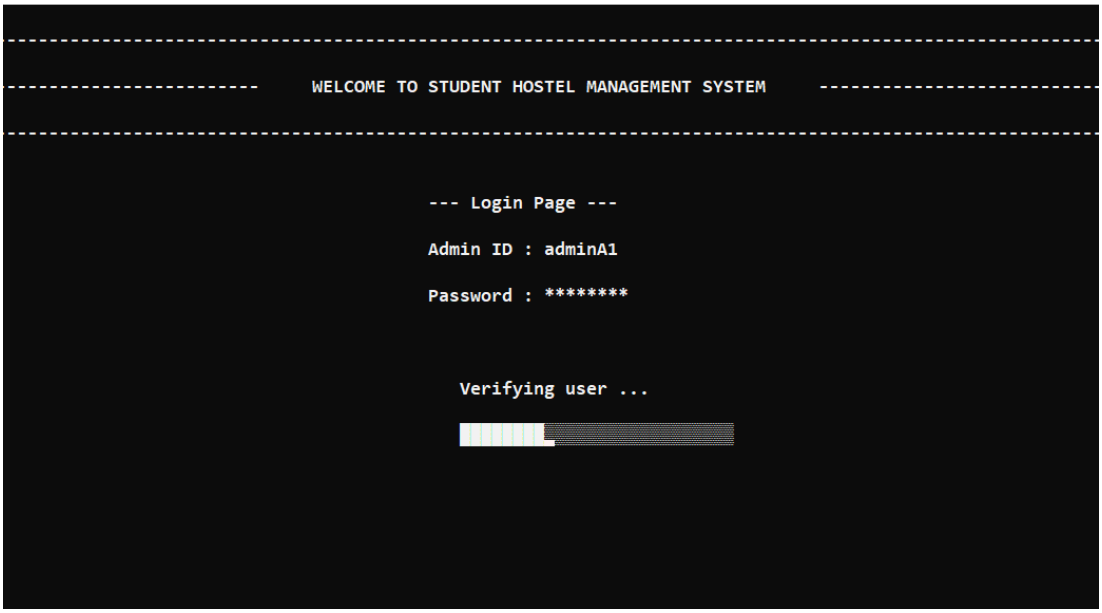


Figure 3.5.2.1 System Interface Login Student

3.5.3 Admin Menu

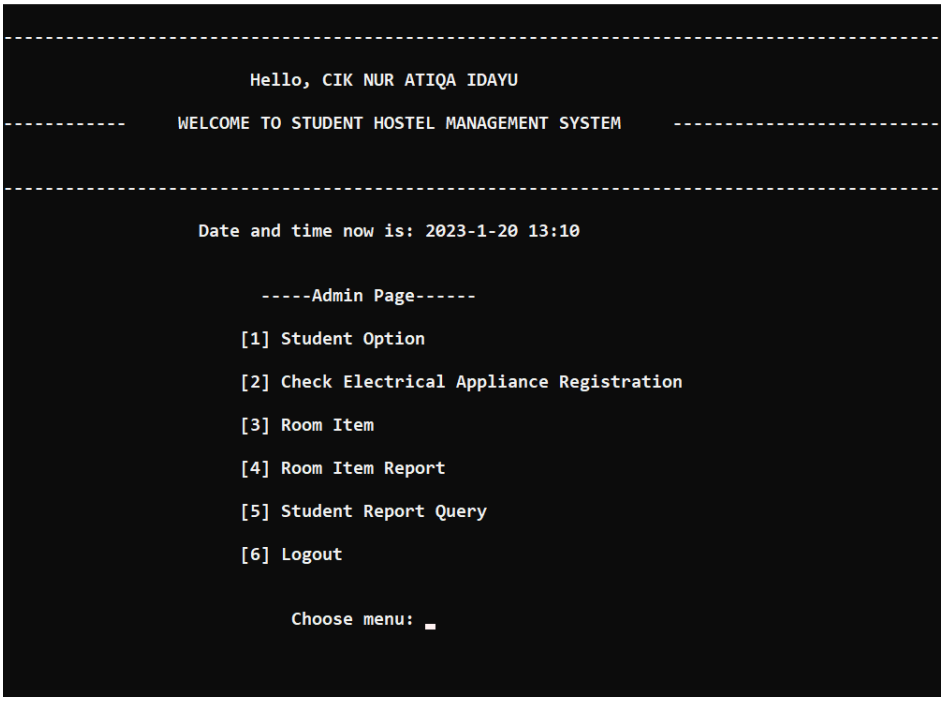


Figure 3.5.3.1 System Interface Admin Menu

3.5.4 Student Option

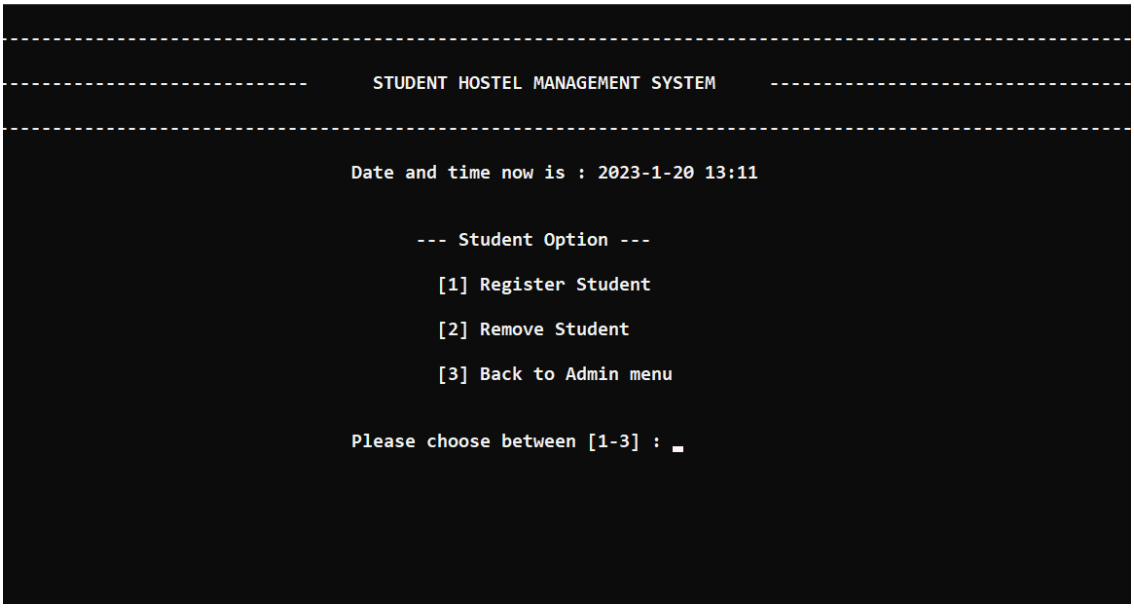


Figure 3.5.4.1 System Interface Student Option

3.5.5 Register Student

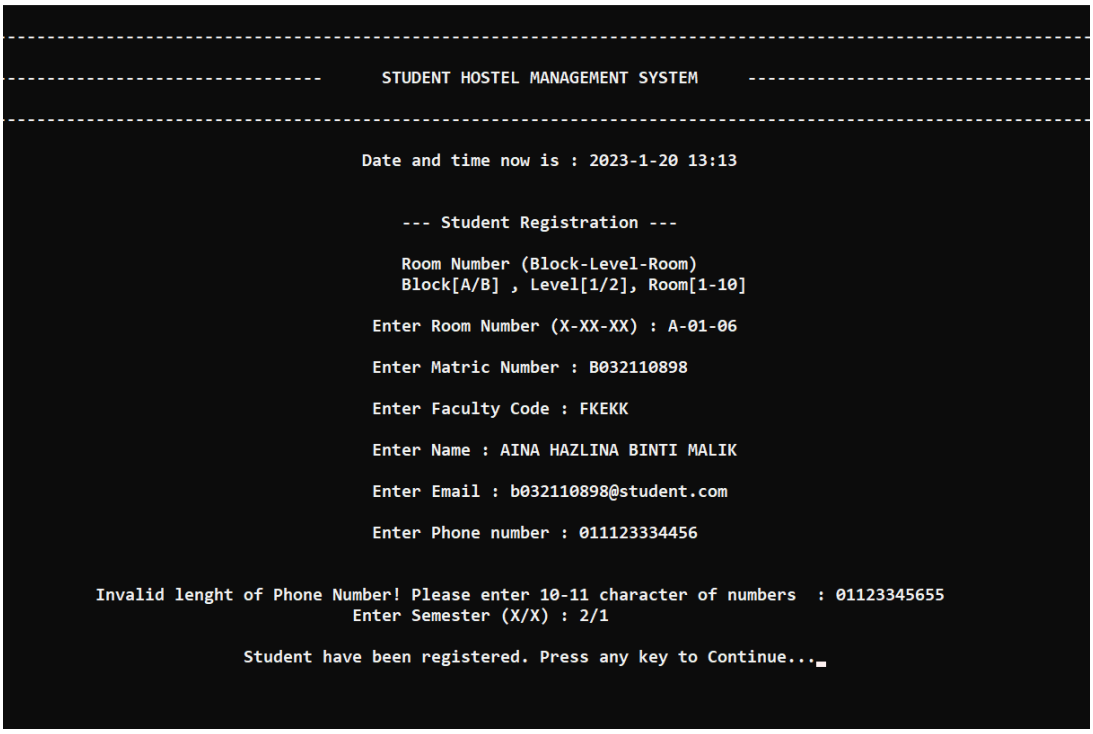
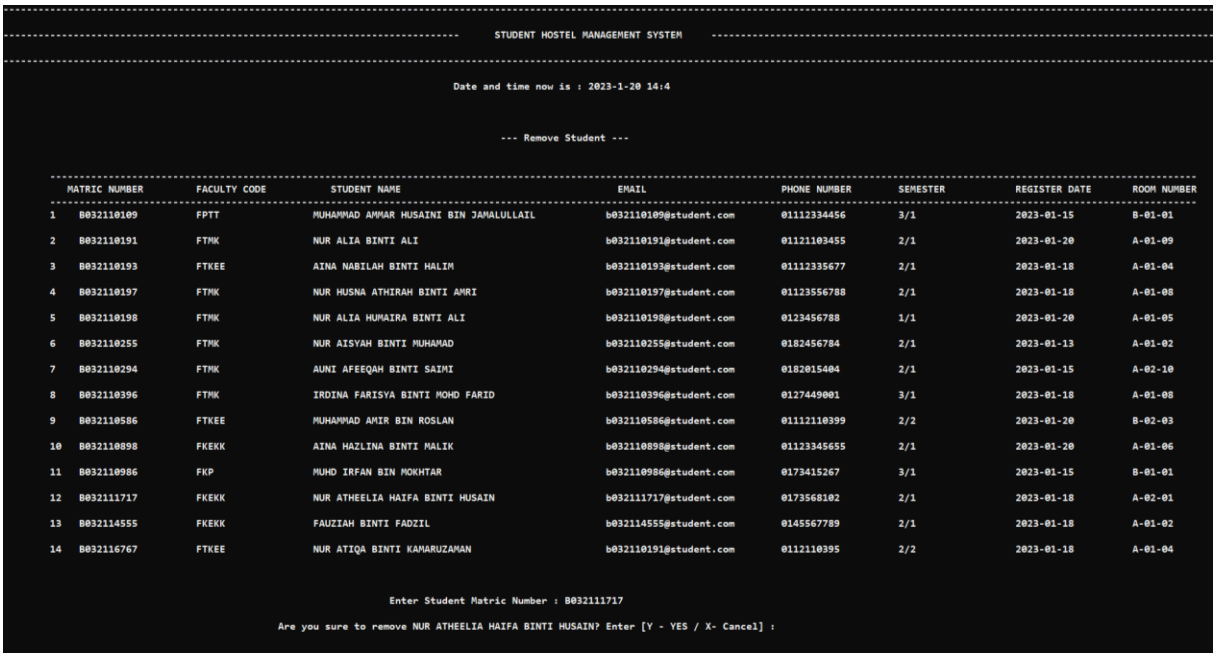


Figure 3.5.5.1 System Interface Register Student

3.5.6 Remove Student



	MATRIC NUMBER	FACULTY CODE	STUDENT NAME	EMAIL	PHONE NUMBER	SEMESTER	REGISTER DATE	ROOM NUMBER
1	B032110109	FPTT	MUHAMMAD AMMAR HUSAINI BIN JAMALULLAIL	b032110109@student.com	01112334456	3/1	2023-01-15	B-01-01
2	B032110191	FTMK	NUR ALIA BINTI ALI	b032110191@student.com	01121103455	2/1	2023-01-20	A-01-09
3	B032110193	FTKEE	AINA NABILAH BINTI HALIM	b032110193@student.com	01112335677	2/1	2023-01-18	A-01-04
4	B032110197	FTMK	NUR HUSNA ATHIRAH BINTI AMRI	b032110197@student.com	01123556788	2/1	2023-01-18	A-01-08
5	B032110198	FTMK	NUR ALIA HUMAIRA BINTI ALI	b032110198@student.com	0123456788	1/1	2023-01-20	A-01-05
6	B032110255	FTMK	NUR AISYAH BINTI MUHAMAD	b032110255@student.com	0182456784	2/1	2023-01-13	A-01-02
7	B032110294	FTMK	AUNI AFEEQAH BINTI SAIMI	b032110294@student.com	0182015404	2/1	2023-01-15	A-02-10
8	B032110396	FTMK	IRDINA FARISYA BINTI MOHD FARID	b032110396@student.com	0127449001	3/1	2023-01-18	A-01-08
9	B032110586	FTKEE	MUHAMMAD AMIR BIN ROSLAN	b032110586@student.com	01112110399	2/2	2023-01-20	B-02-03
10	B032110898	FKEKK	AINA HAZLINA BINTI MALIK	b032110898@student.com	01123345655	2/1	2023-01-20	A-01-06
11	B032110986	FKP	MUHD IRFAN BIN MOKHTAR	b032110986@student.com	0173415267	3/1	2023-01-15	B-01-01
12	B032111717	FKEKK	NUR ATHEELIA HAIFA BINTI HUSAIN	b032111717@student.com	0173568102	2/1	2023-01-18	A-02-01
13	B032114555	FKEKK	FAUZIAH BINTI FADZIL	b032114555@student.com	0145567789	2/1	2023-01-18	A-01-02
14	B032116767	FTKEE	NUR ATIQA BINTI KAMARUZAMAN	b032110191@student.com	0112110395	2/2	2023-01-18	A-01-04

Figure 3.5.6.1 System Interface Remove Student

3.5.7 Electrical Appliance Receipt

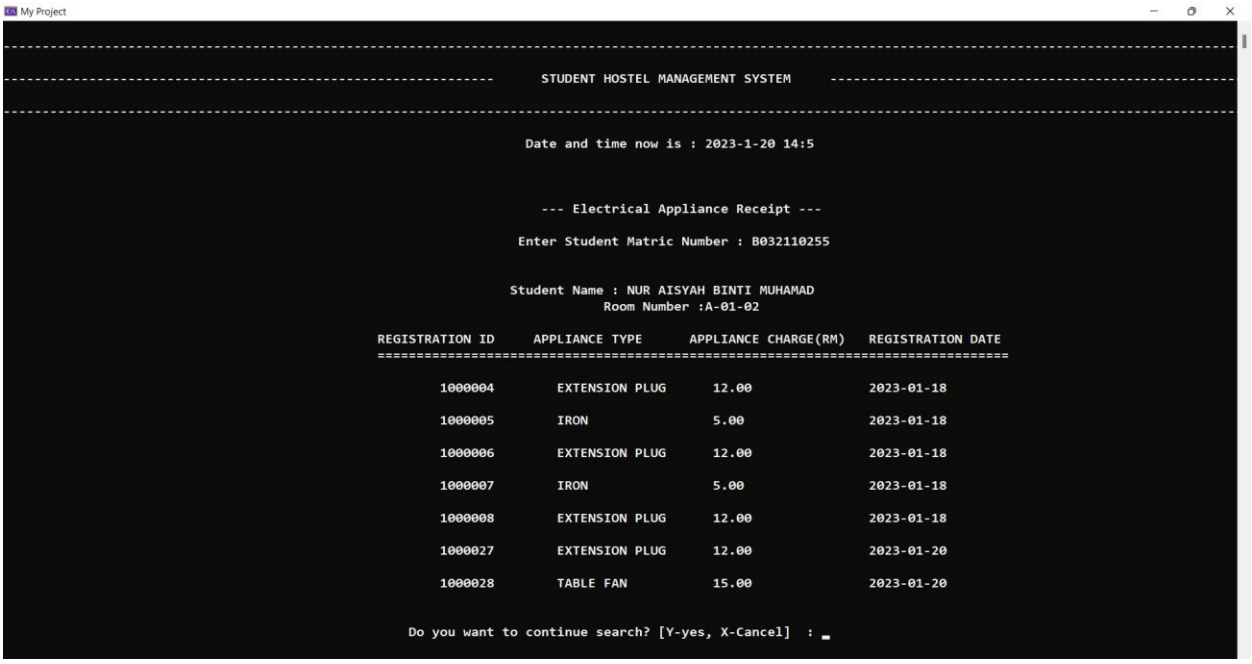


Figure 3.5.7.1 System Interface Electrical Appliance Receipt

3.5.8 Room Item Menu

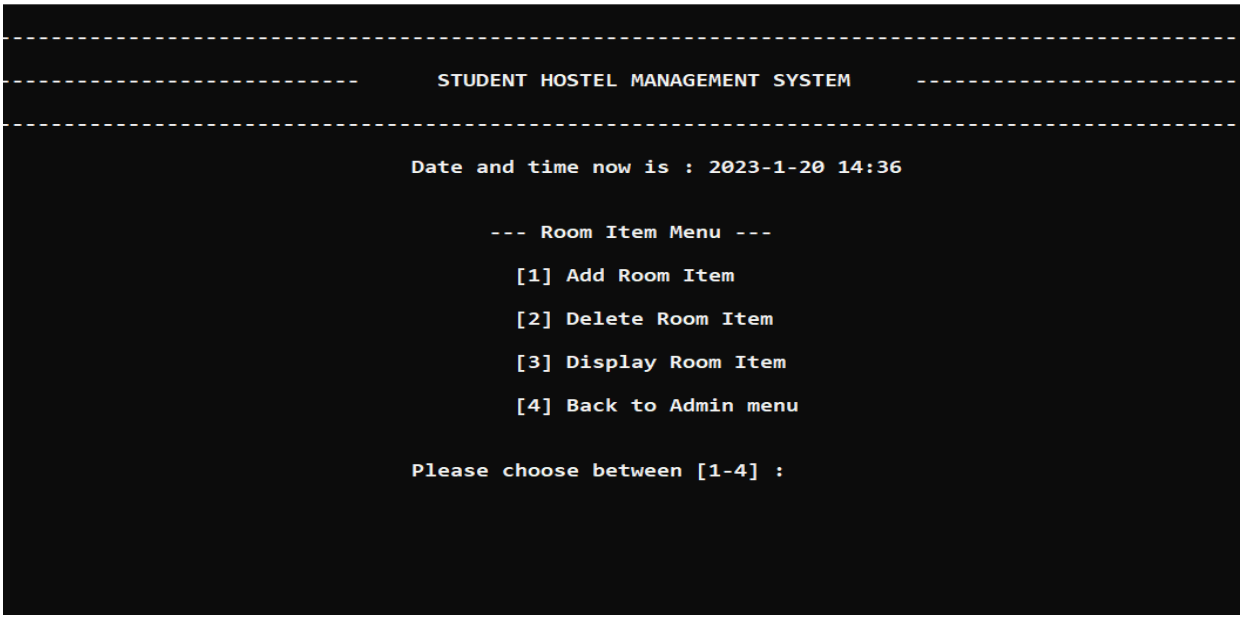


Figure 3.5.8.1 System Interface Room Item Menu

3.5.9 Add Room Item

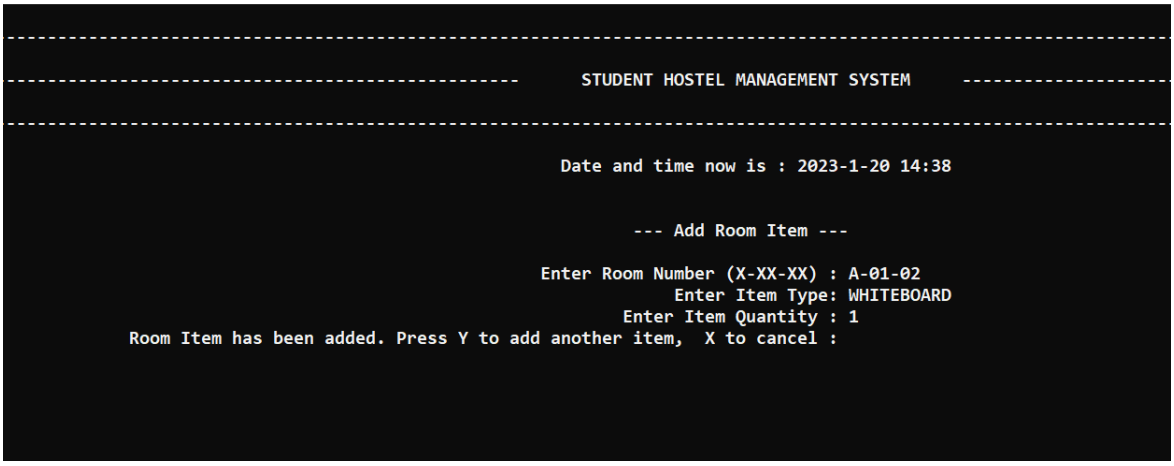


Figure 3.5.9.1 System Interface Add Room Item

3.5.10 Delete Room Item

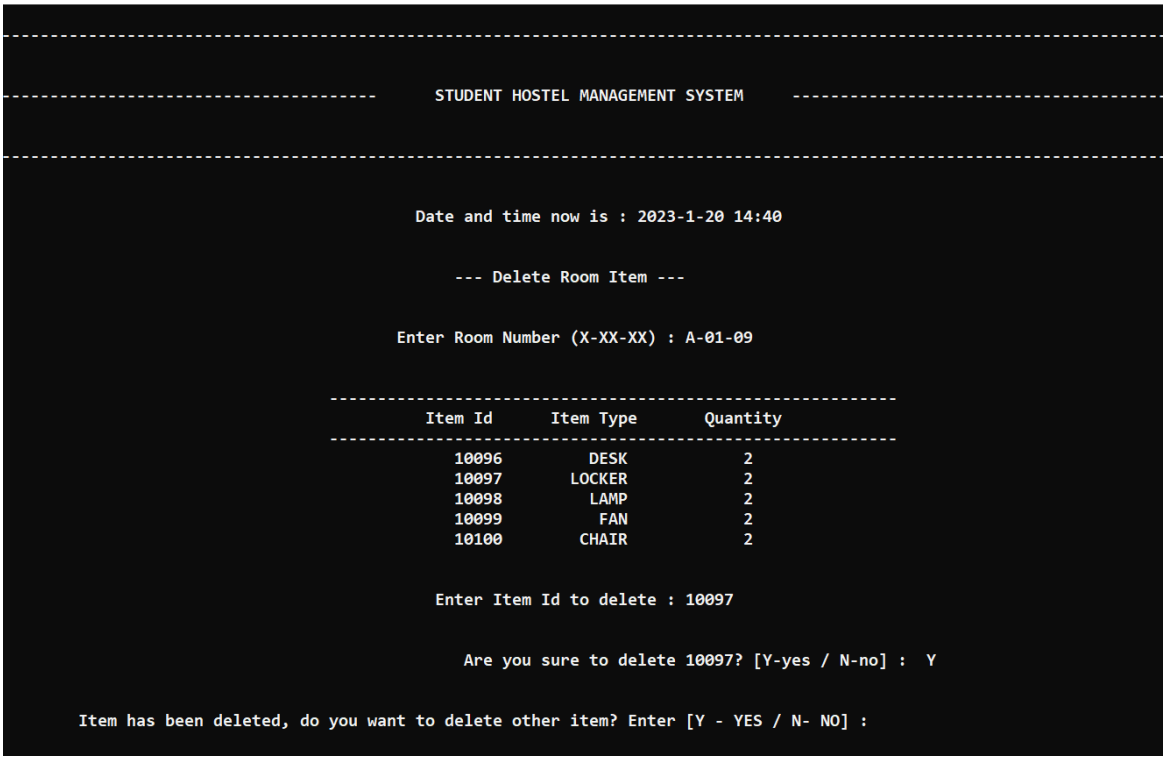


Figure 3.5.10.1 System Interface Delete Room Item

3.5.11 Display Room Item

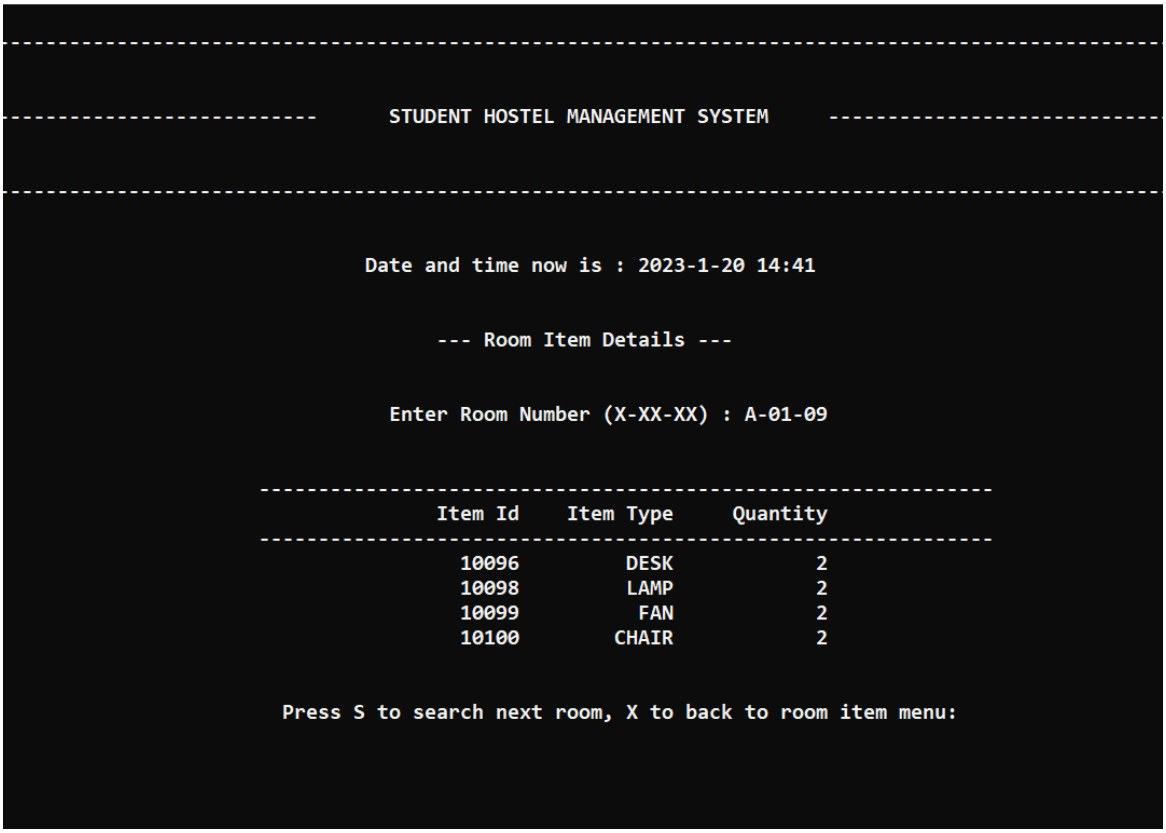


Figure 3.5.11.1 System Interface Display Room Item

3.5.12 Room Item Report

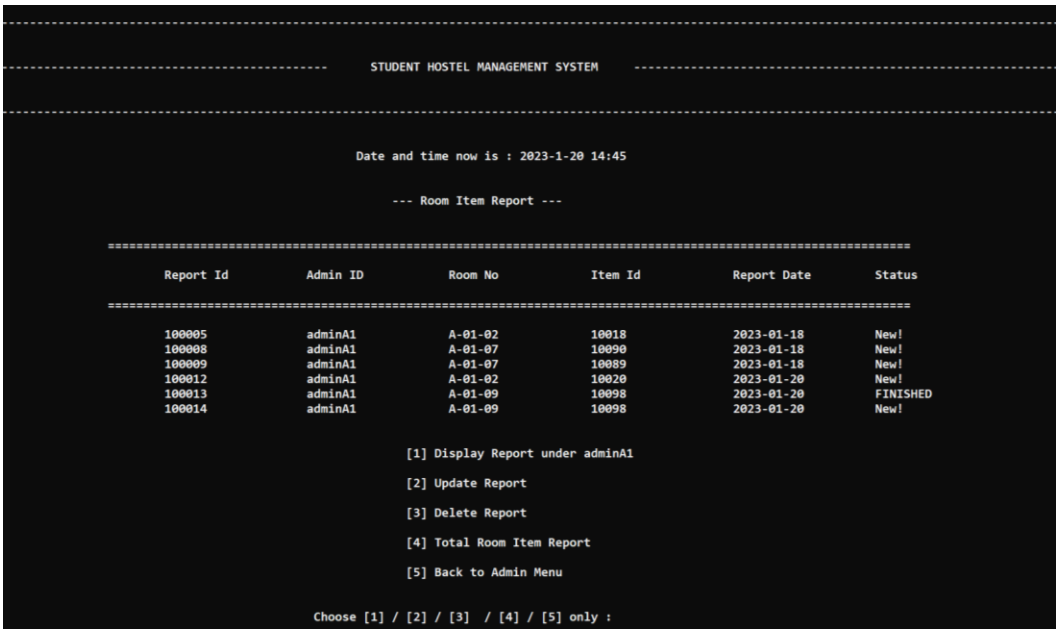


Figure 3.5.1.1 System Interface Room Item Report

3.5.13 Update Report

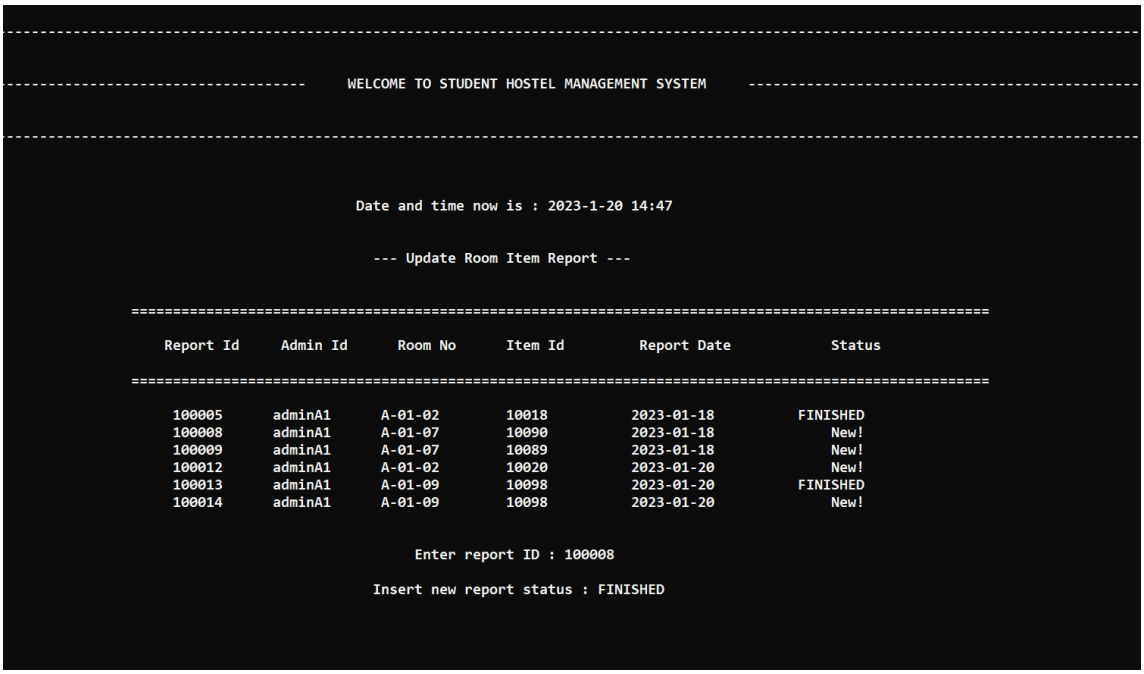


Figure 3.5.13.1 System Interface Update Report

3.5.14 Delete Item Report

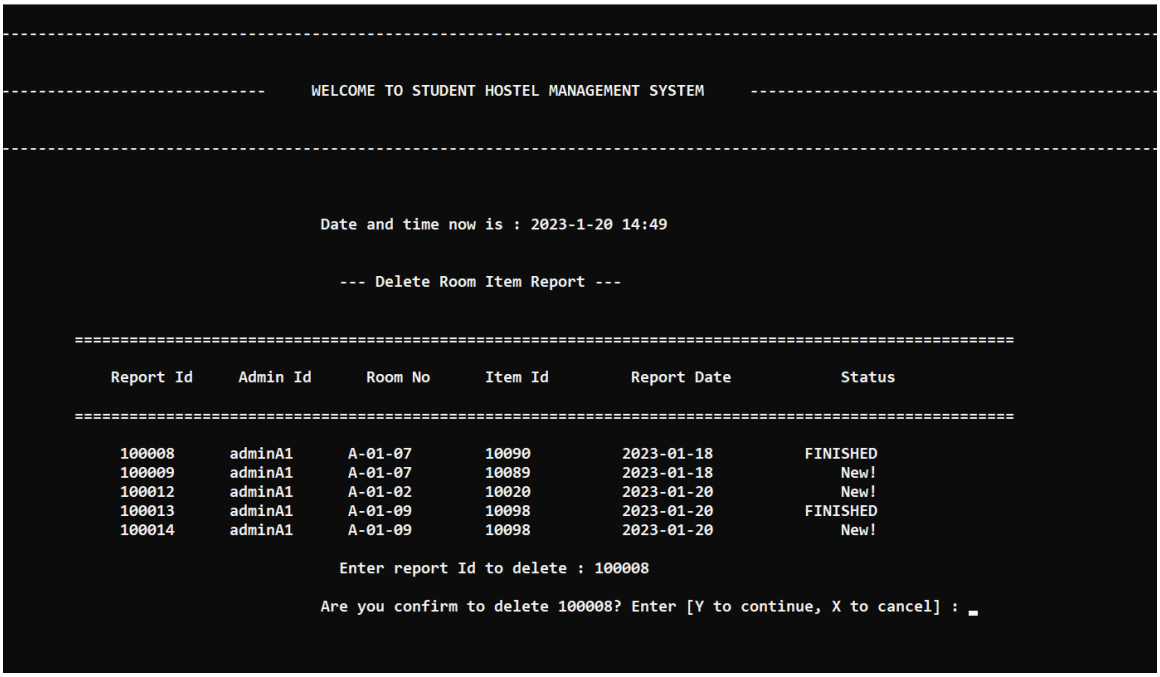


Figure 3.5.14.1 System Interface Delete Item Report

3.5.15 Total Report

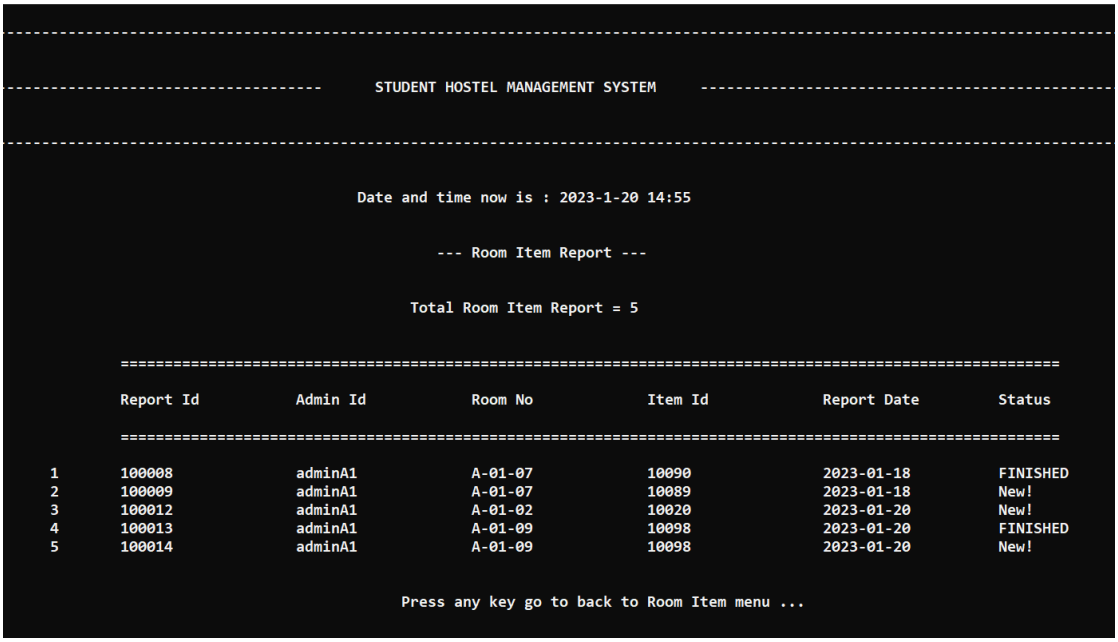


Figure 3.5.1.1 System Interface Total Report

3.5.16 Student Report Query

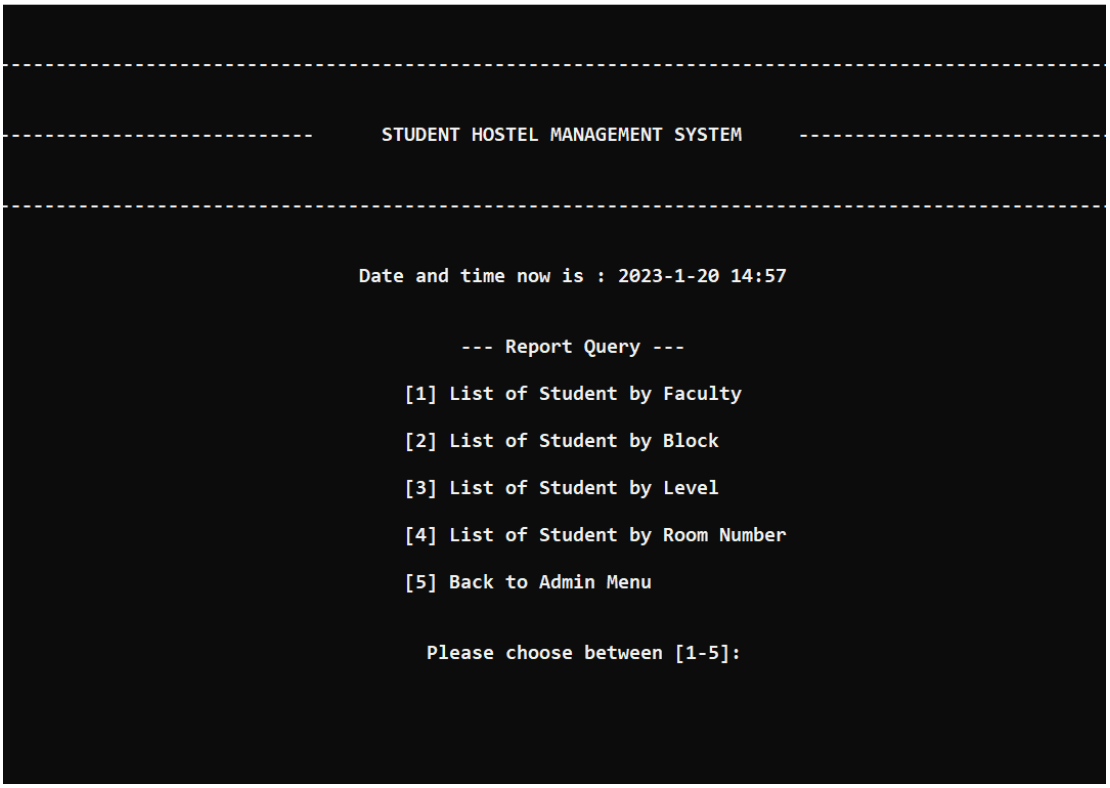


Figure 3.5.16.1 System Interface Student Report Query

3.5.17 List of Student by Faculty

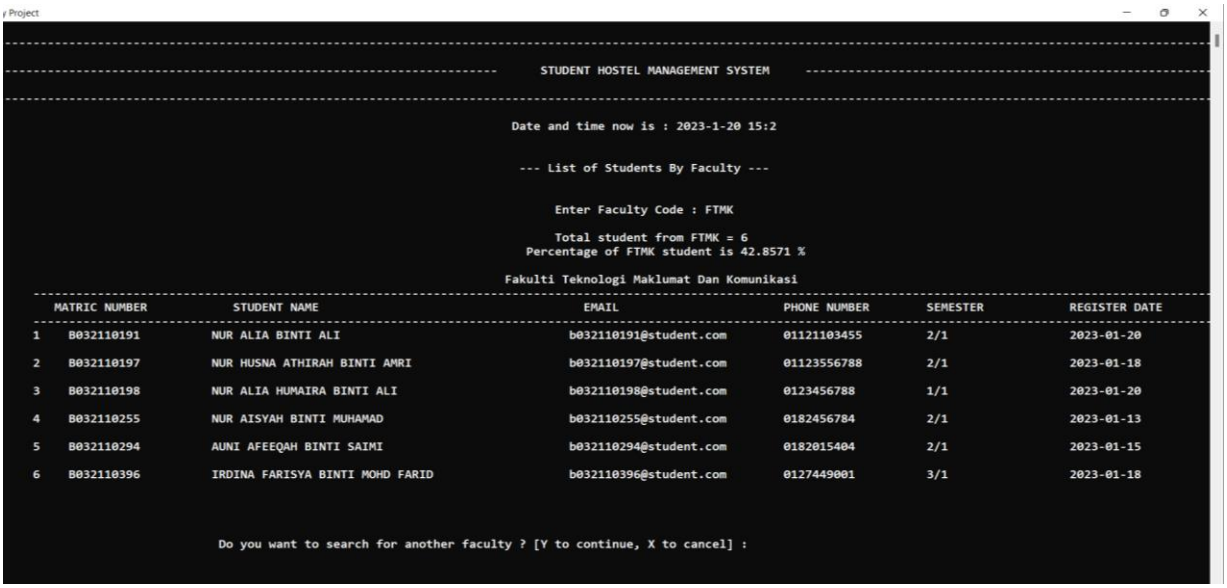


Figure 3.5.17.1 System Interface List Student by Faculty

3.5.18 List of Student by Block

Figure 3.5.18.1 System Interface List of student by Block

3.5.19.1 List of Student by Level

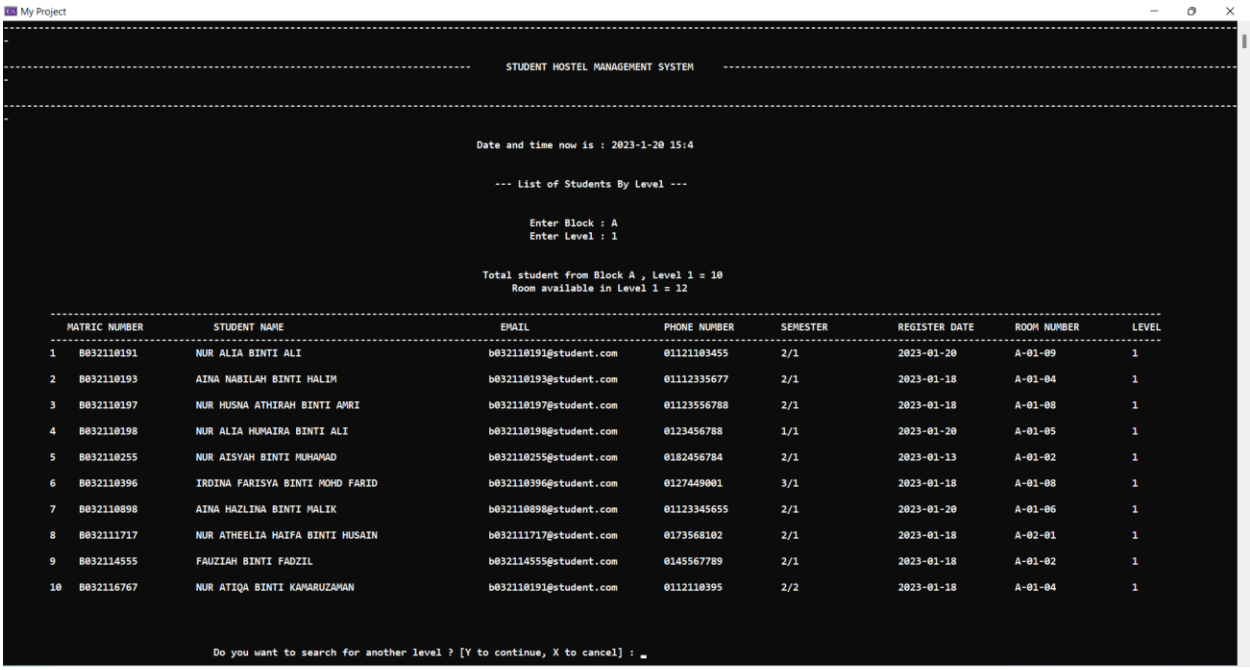


Figure 3.5.19.1 System Interface List of Students by level

3.5.20 List of Student by Room

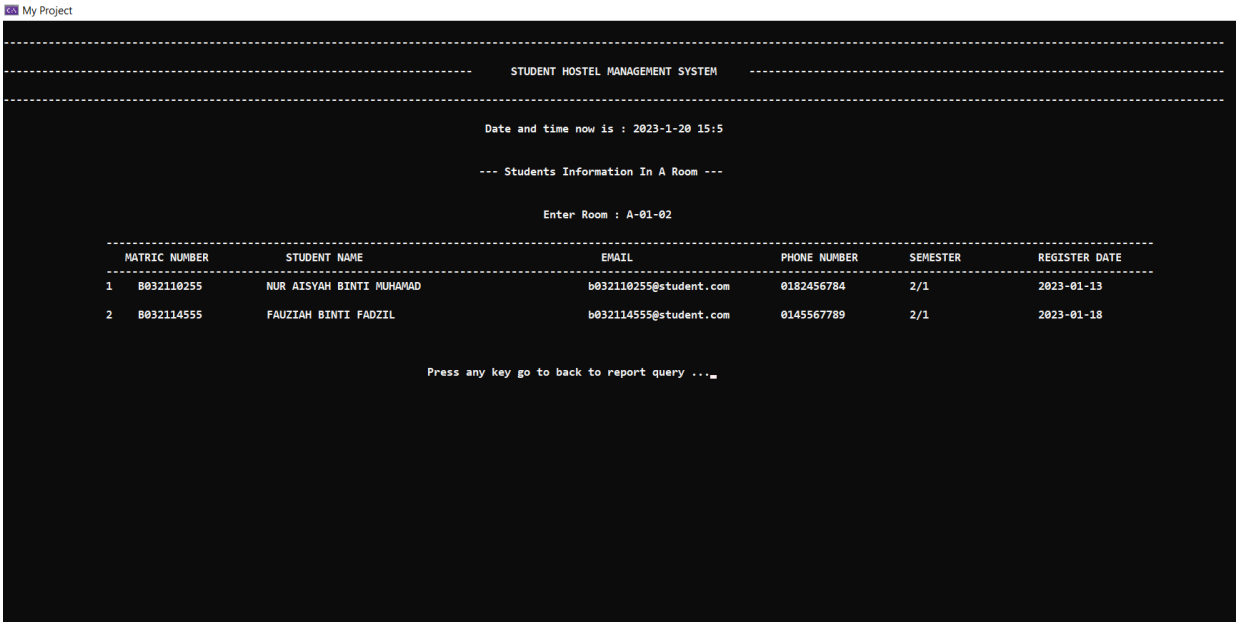


Figure 3.5.20.1 System Interface List of student by room

3.5.21 Login Student

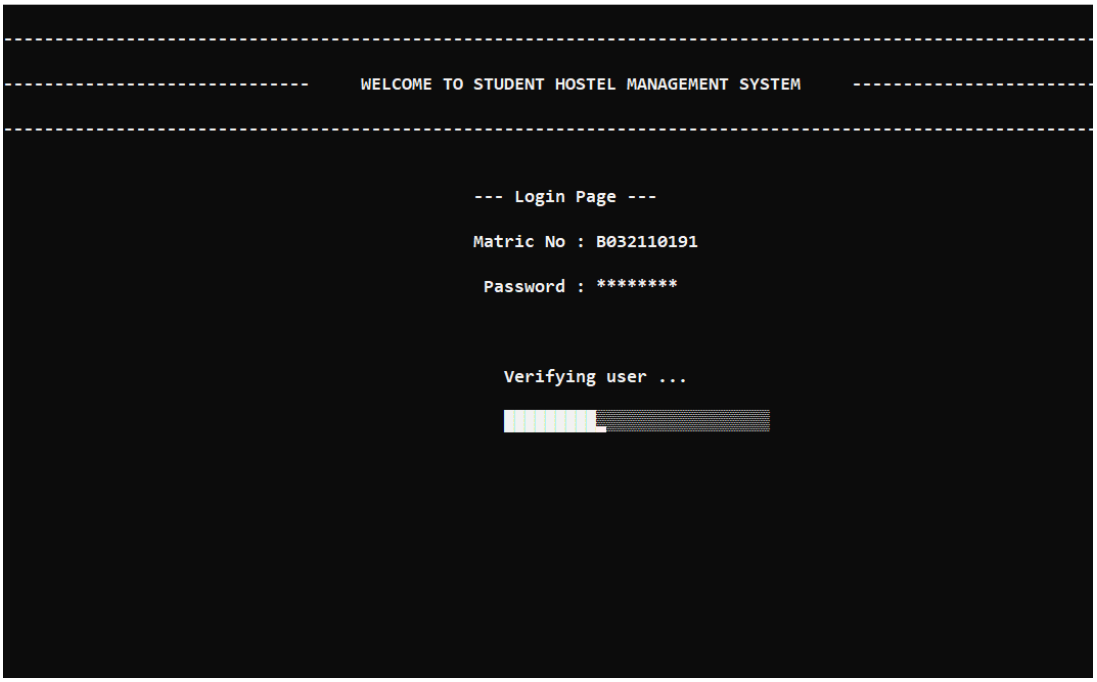


Figure 3.5.21.1 System Interface Login Student

3.5.22 Student Menu

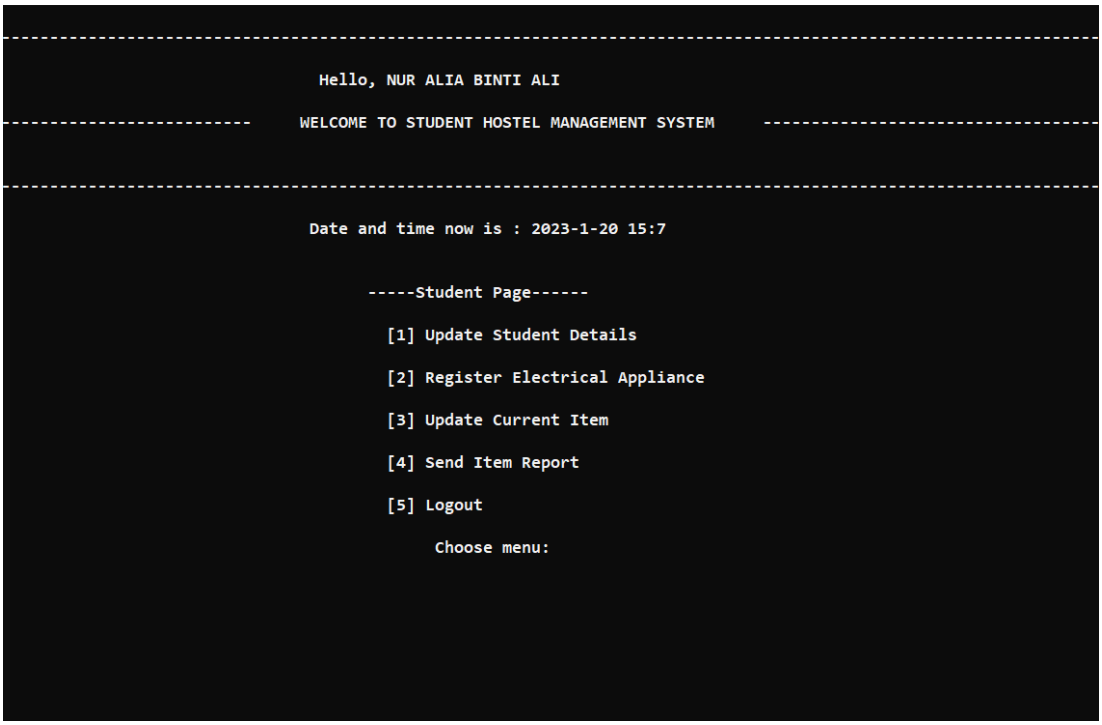


Figure 3.5.22.1 System Interface Student Menu

3.5.23 Update Student Details

Figure 3.5.23.1 System Interface Update Student Details

3.5.24 Register Electrical Appliance

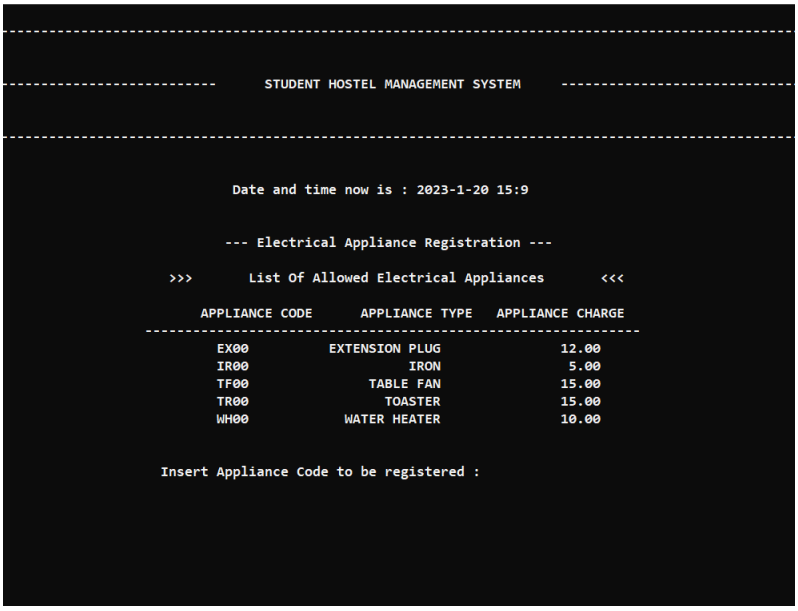


Figure 3.5.24.1 System Interface Register Electrical Appliance

3.5.25 Register Electrical Appliance Receipt

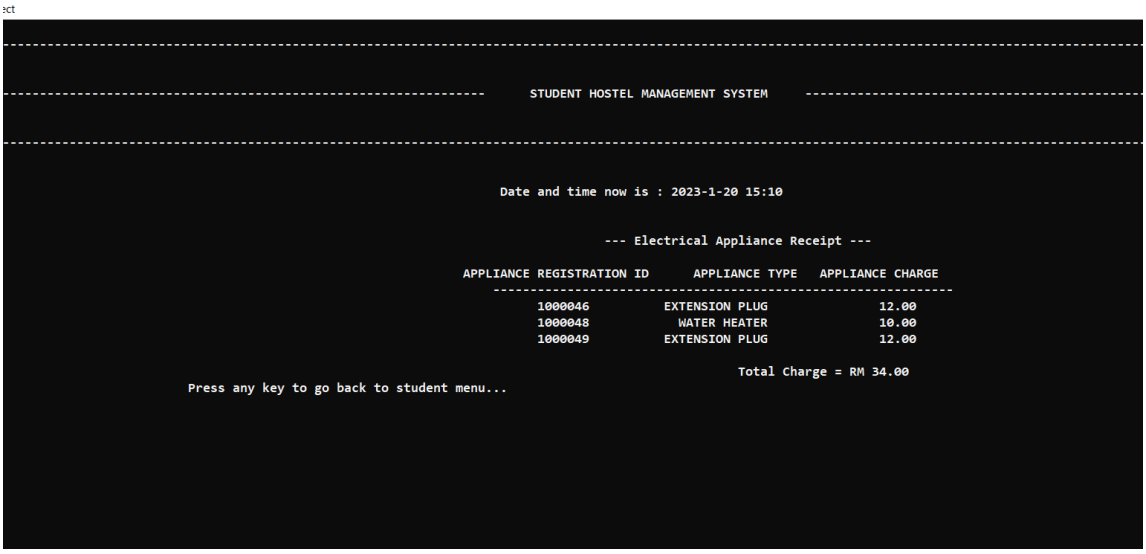


Figure 3.5.25.1 System Interface Register Electrical Appliance Receipt

3.4.26 Update Room Item

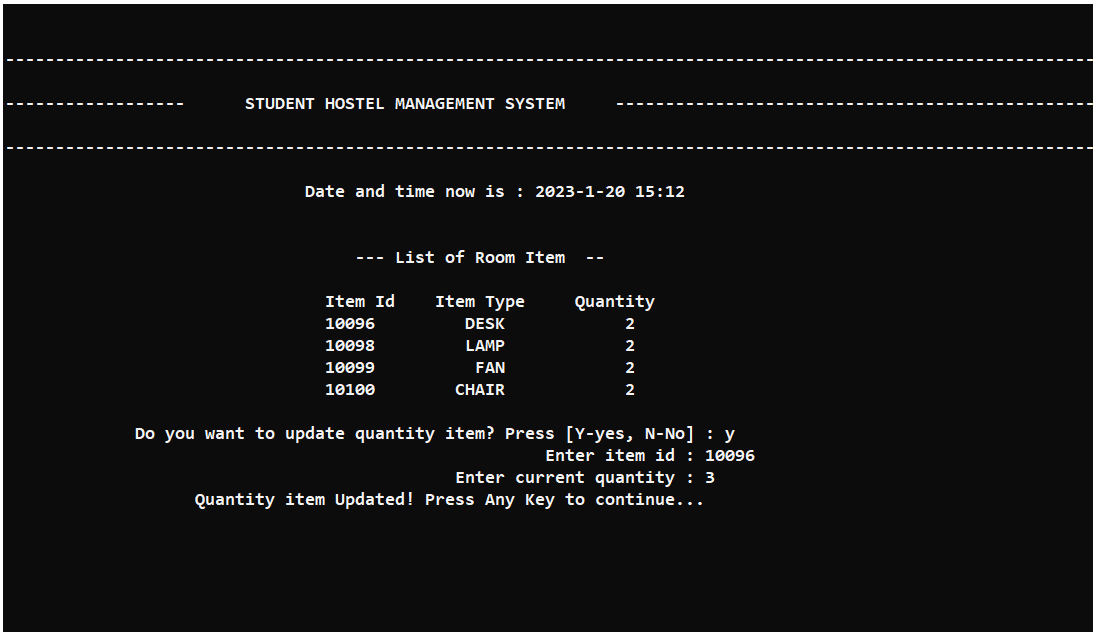


Figure 3.5.26.1 System Interface Update Room Item

4.4.27 Send Room Item Report

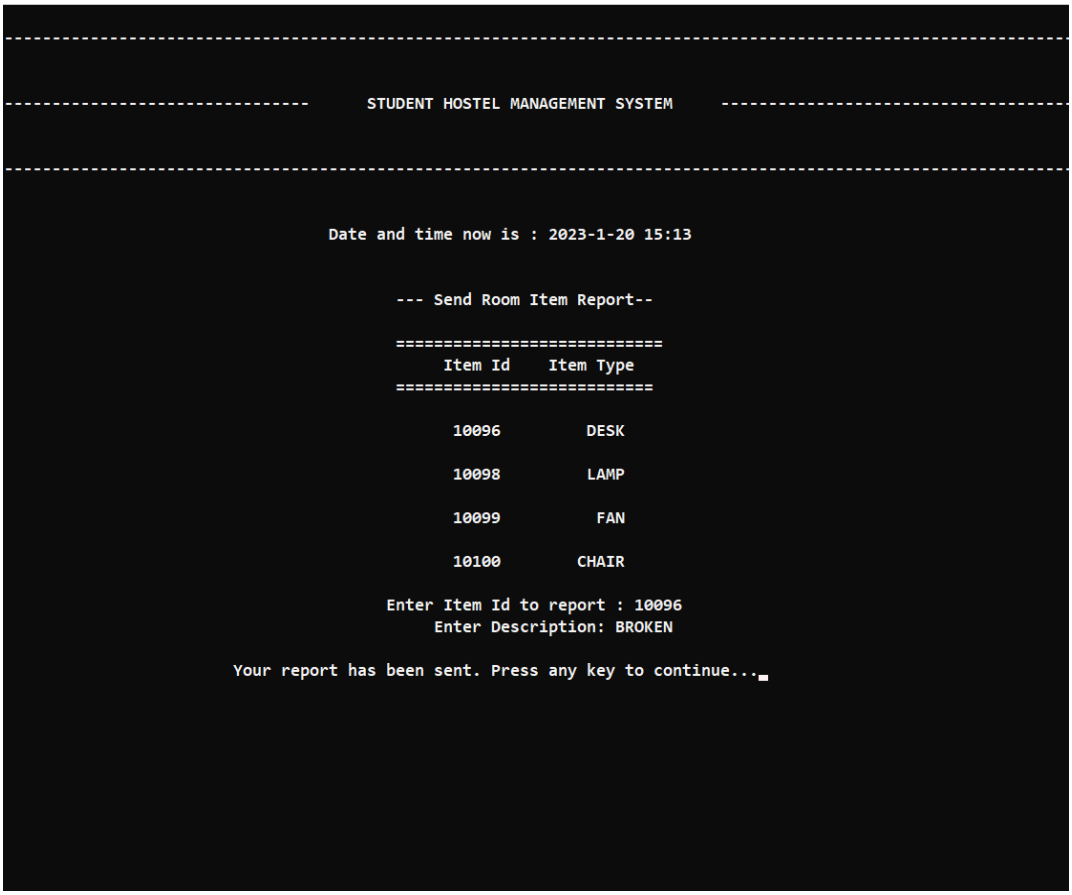


Figure 3.5.27.1 System Interface Main Menu

4.5 Summary

This chapter describes the system's design, which comprises of a flowchart, a system's pseudocode, and a system's user interface. The implementation of the system, including the programming approach used throughout system development and the techniques used to manage faults, will be covered in more detail in the following chapter.

CHAPTER 4: IMPLEMENTATION

4.1) Introduction

This chapter discusses how the system is implemented using software, including the programming technique used, a description of errors that have occurred, and a way for handling such issues.

4.2) Coding implementation

Programming technique used:

- Function

workshop1 (Global Scope)

```
1291    cout << setw(90) << "Date and time now is : " << currTime << endl;
1292    cout << endl;
1293    cout << endl;
1294
1295    cout << setw(95) << "--- Room Item Menu ---" << endl;
1296    cout << endl;
1297    cout << setw(93) << "[1] Add Room Item " << endl << endl;
1298    cout << setw(96) << "[2] Delete Room Item " << endl << endl;
1299    cout << setw(97) << "[3] Display Room Item " << endl << endl;
1300    cout << setw(98) << "[4] Back to Admin menu " << endl << endl;
1301
1302    cout  << endl;
1303    cout << setw(97) << "Please choose between [1-4] : ";
1304
1305    cin >> menuRoom;
1306
1307    switch (menuRoom)
1308    {
1309    case 1:
1310        AddRoomItem();
1311        break;
1312    case 2:
1313        DeleteRoomItem ();
1314        break;
1315    case 3:
1316        DisplayRoomItem();
1317        break;
1318    case 4:
1319        AdminMenu();
1320        break;
1321    /*case 4:
1322        AdminMenu();
1323        break;*/
1324    default:
1325        cout << setw(97) << "Please choose between 1 - 4. Press Any Key To Continue...";
1326        _getch();
1327        system("cls");
1328        RoomItem();
1329        break;
1330    }
1331 }
```

Figure 4.2.1 Example of function used in the system

- Array

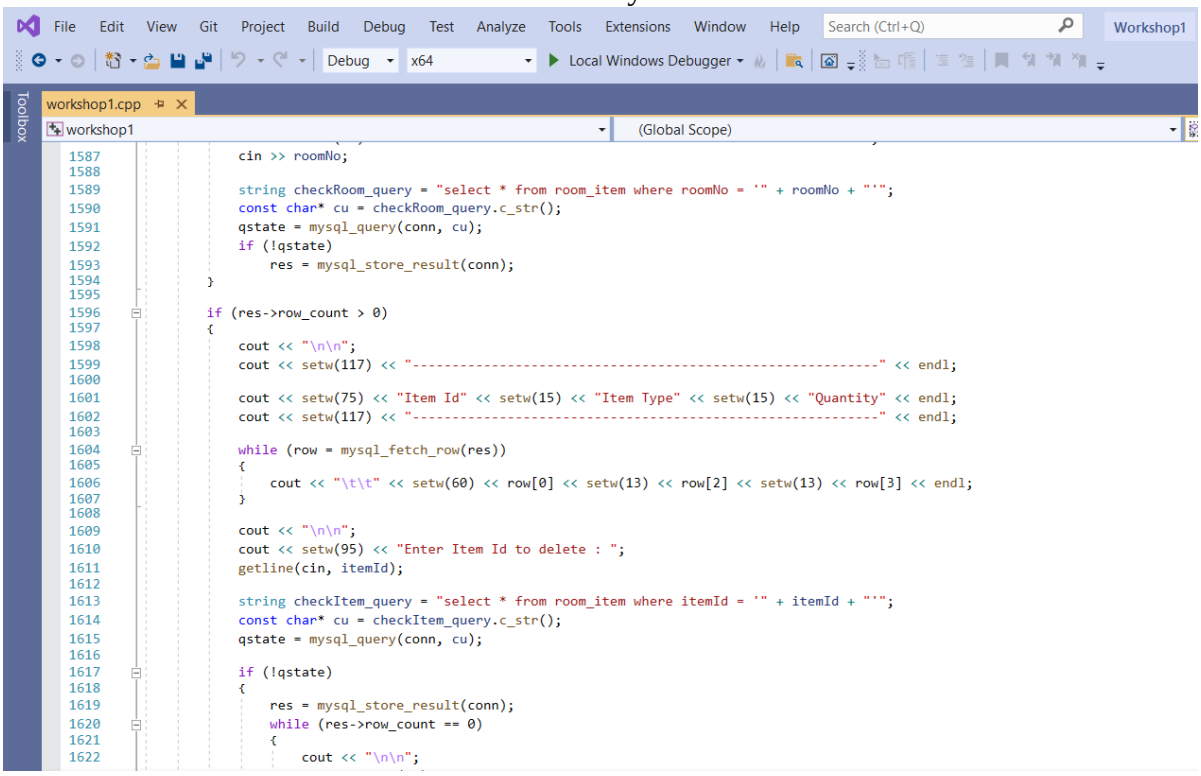


Figure 4.2.2 Example of array used in the system

- Data Manipulation Language in SQL

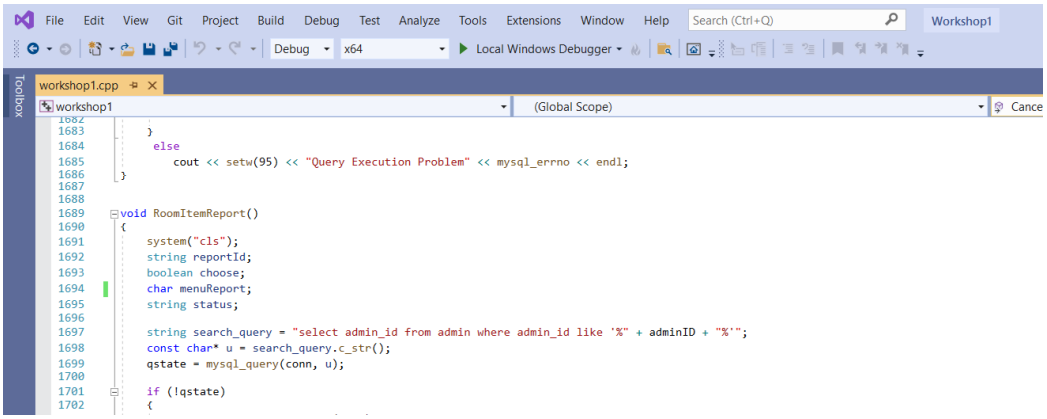


Figure 4.2.3 Example of Data Manipulation Language in SQL used in the system

- while loop

```
}

cout << endl;
cout << setw(109) << "Do you want to search for another level ? [Y to continue, X to cancel] : ";
cin >> choose;

while ((choose != 'y' && choose != 'Y') && (choose != 'x' && choose != 'X'))
{
    cout << setw(97) << "Wrong choice! Please choose either Y or X :  ";
    cin >> choose;
}
```

Figure 4.2.4 Example of while loop used in the system

- Calculation In SQL

```
2549    cin.ignore(1, '\n');
2550    getline(cin, facultyCode);
2551
2552    string totalStudFp_query = "select count(facultyCode) as totalStudentFaculty from student where facultyCode = '" + facultyCode + "'";
2553    const char* cuP = totalStudFp_query.c_str();
2554    qstate = mysql_query(conn, cuP);
2555
2556    if (!qstate)
2557    {
2558        res = mysql_store_result(conn);
2559        while (row = mysql_fetch_row(res))
2560        {
2561            totalStudentFaculty = row[0];
2562        }
2563    }
2564    else
2565    {
2566        cout << "Query Execution Problem!" << mysql_errno(conn) << endl;
2567    }
2568
2569    string totalStudF_query = "select count(matricNo) as totalStudent from student";
2570    const char* cu = totalStudF_query.c_str();
2571    qstate = mysql_query(conn, cu);
2572
2573    if (!qstate)
2574    {
2575        res = mysql_store_result(conn);
2576        while (row = mysql_fetch_row(res))
2577        {
2578            totalStudent = row[0];
2579        }
2580    }
2581    else
2582    {
2583        cout << "Query Execution Problem!" << mysql_errno(conn) << endl;
2584    }
2585
2586    studentPercentage = (stod(totalStudentFaculty)/stod(totalStudent))*100;
2587
```

Figure 4.2.5 Example of Calculation in SQL used in the system

- Join statement in SQL

```
cout << setw(106) << "Enter Block : ";
cin.ignore(1, '\n');
getline(cin, block);

string totalStudF_query = "select count(room.block) as totalStudentBlock from student INNER JOIN room On student.roomNo = room.roomNo where room.block = '" + block + "'";
const char* cuF = totalStudF_query.c_str();
qstate = mysql_query(conn, cuF);
```

Figure 4.2.6 Example of Join statement in SQL used in the system

4.3) Error handling

There are 3 types of error that available in this system:

1) Human error
   - Error that occurs when user wrongly enter the input
   - Method to handle this type of error:

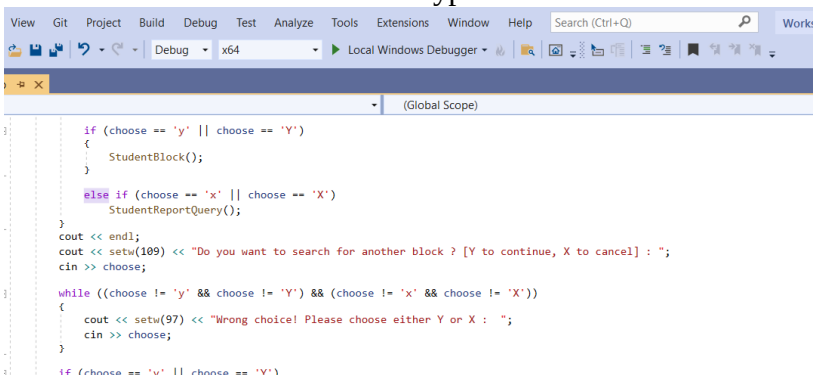


Figure 4.3.1 Method to handle careless error in the system

An error message will appear when a user enters a number that isn't present in the menu interface and will instruct them to only enter activity codes that are present in the menu's selection section.

2) System error

- This kind of error typically denotes any technical issue with the system that could prevent it from operating normally (typically on the SQL function of the system, as part from the a c++ problem that could be present and quickly noticed in the IDE).

Example:

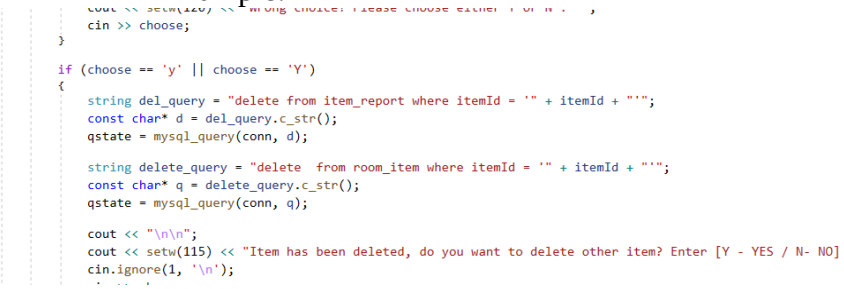
```
        StudentReportQuery();

    }

    else
    {
        cout << "Query Execution Problem!" << mysql_errno(conn) << endl;
    }
}
```

Figure 4.3.2 Method to handle system error in the system

The SQL query in this bit of code is used to select student information from the database. If the SQL query does not execute, "Query Execution Problem" and the appropriate error code will be displayed.

3) Logic error

- This kind of error affects the interface side of the system, where it affects the system's flow, but it has no impact on the technical side of the system.
- Example:



```
    cin >> choose;
}

if (choose == 'y' || choose == 'Y')
{
    string del_query = "delete from item_report where itemId = '" + itemId + "'";
    const char* d = del_query.c_str();
    qstate = mysql_query(conn, d);

    string delete_query = "delete  from room_item where itemId = '" + itemId + "'";
    const char* q = delete_query.c_str();
    qstate = mysql_query(conn, q);

    cout << "\n\n";
    cout << setw(115) << "Item has been deleted, do you want to delete other item? Enter [Y - YES / N- NO]
    cin.ignore(1, '\n');
```

Figure 4.3.3 Method to handle logic error in the system

If we want to delete room Item, we need to delete all the report of the item report for the item id first. This is because it might cause the logic error if there is still the report of the deleted room item.

4.4 Summary

The programming technique used to create the system and the technique for dealing with system failures are covered in this chapter. The conclusion of this report, which will include the system's limits and suggestions for future improvements, will be the system's final chapter.

CHAPTER 5: TESTING AND RESULTS

5.1 Introduction

This chapter will discuss the system's drawbacks and potential improvements.

5.2 Constraints of the system

- **Using xampp application**

There are some issues while using xampp to connect to the database if the xampp is mistakenly shut down. If the user does not have any backup of the database, all of the information will be lost.

- **The design interface are not systematic**

When running a module, some of it has no way to exit the function and it's hard for the user to use the system because they need to enter some details first before exit the module. Some of the issues will need the user to run the system again to use the system

5.3 Future improvement on the system

- **Use cloud database**

This particular innovation will create the opportunity for real-time collaborations where any database modifications may be viewed without needing to be saved in the local device of any user who wants to make some database changes.

- **Make the navigation key for the system**

For future improvement, it is better to put the function for navigate using keyboard for the user to easily go to menu or modules that they want.

5.4 Total Conclusion of the report

The hostel staff can easily manage the student and room item information with the use of this technology. Although this system achieves the desired results based on the objectives, there are also several weaknesses that must be fixed in order to increase the system's efficiency. May the Student Hostel Management System will give the big impact to the student hostel in managing the data of the hostel.

REFERENCES

W3Schools. “W3Schools Online Web Tutorials.” *W3schools.com*, W3Schools, 1998, www.w3schools.com/.

Ayub, N., Hassan, A.S.& Zahra Khozaei, Z. (2010). The Factors Predicting Students’ Satisfaction with University Hostels. A Case Study, University Sains, Malaysia

APPENDIX