



**FACULTY OF ARTS, COMPUTING & CREATIVE INDUSTRY UNIVERSITI
PENDIDIKAN SULTAN IDRIS**

MTD 3043: SYSTEMS ANALYSIS & DESIGN




Project Title	House Inventory of Residential College Management System		
Priority	Medium		
Start Date:	09-03-2020		
Finish Date:	31-07-2020		
Name of Members	Matric no	Position / Role	
AHMAD FAUZI BIN SUKARDI	D20181083618	Project Manager	
GILBERT ANAK GORDION	D20181083619	System Analyst	
NURIHDI BIN TAJUDIN	D20181083636	Developer 1	
NUR ATIKAH BINTI SHAPUDIN	D20181083611	Developer 2	
HASYA CAMELIA BINTI HASBI	D20181083621	Quality Assurance	

CONTENT OF THE PROJECT REPORT**TABLE OF CONTENT****1. Planning****• Introduction**

Some of existing system are semi-automated and many are manual to keep track of utilities in the residential college. Student still using the manual method by fill the form that being provide by the college management. We have found that the staffs first of all will distribute the utility form to the students before entering in computer system. The staffs in college management are using both ways to keep the record of house utility in residential college. Following this method is very time consuming and tedious. It has many drawbacks as there may be mistakes while recording the data.

So, in this project we trying to make house utility of residential college system which will help students to fill the online form of their house utility and staffs can keep record of utilities in systematic way.

• Problem statement

-  How will resolving the problem advance understanding of the topic?
-  What benefits will it have for future research?
-  Does the problem have direct or indirect consequences for society?

To gain a more complete understanding of how the best steps in solving hardware registration problems in public universities in the college administration division as well as can help develop a more robust theory of flexibility and accuracy in providing better and aesthetic service. Also to make sure that this system are Paperless, which is we want to reduce the use of paper for the forms . Minimize the inaccuracy. Sometime, the student that filled the inventory form may make a mistake by ticked at wrong table of inventory. Improve IT governances by manage the inventory by online, the system of management will smoothly being filled and more convenient. Data secure, the privacy of the users that have been stored will be safe. with this, all work between graduates and officers will be more efficient and faster.

- **Project Objective** (Objective of the project is to develop college house utility management system and)

- ✚ To provide house inventory form access to all necessary personnel (students/staffs)
- ✚ To provide registration of electrical stuffs form to students
- ✚ To provide damage complaint form to students
- ✚ To reduce labor of inventory management, centralize control and automation
- ✚ Develop an automated system that will be able to record, store, retrieve of inventory useful to management

- **Scope**

Project scope is how a project's goals and objectives are defined. So in this project scope lists those goals, what the deliverables will be, the tasks needed to make the deliverables, what the costs for that will be and the deadlines. It is, in fact, the parameters of the whole project. This system are targeting among student and officer because we all decided to create a system to ensure that this system is able to provide convenience to graduates and administrative officers of UPSI College, especially.

- **Feasibility study**

A Feasibility Study Report (FSR) is a formally documented output of feasibility study that summarizes results of the analysis and evaluations conducted to review the proposed solution and investigate project alternatives for the purpose of identifying if the project is really feasible, cost-effective and profitable. It describes and supports the most feasible solution applicable to the project.

The report gives a brief description of the project and some background information. Formally this document is the starting point for running the Pre-Charter Sub-Phase. In practice, it signifies that the sponsor can proceed with deciding on project investment and make necessary assignments to the project manager.

Finally, here is an outline for the nine parts of a feasibility report:

1. Executive summary

Name of Members	Position / Role
AHMAD FAUZI BIN SUKARDI	Project Manager
GILBERT ANAK GORDION	System Analyst
NURIHDI BIN TAJUDIN	Developer 1
NUR ATIKAH BINTI SHAPUDIN	Developer 2
HASYA CAMELIA BINTI HASBI	Quality Assurance

2. Description of product/service

House Inventory of Residential College Management System is a system to provide convenience to graduates and administrative officers of UPSI College, especially when registering their house inventory.

3. Technology considerations

We are using System Development Life Cycle (SDLC) model as our method to develop this system. It will have 5 phases which are planning, analysis, design, implementation, and maintenance.

4. Marketing strategy/ Product/service marketplace

We're aiming this system will going to use by student and among administrative officers of UPSI College as the conclusion for whole state education department of higher education institutions

6. Financial projections

Project Budget (Estimated)

Description	Amount
Fact-finding (interview session) 5 times interview x RM50.00	RM250.00
Hardware (server) 2 servers x RM4700	RM 9400.00
Software (open source)	RM 0.00
Quality Assurance (tester) 1 x 150	RM 150.00

- **Risks**

If this project is not pursuing it can cause the inefficient in management of utility form as the staff need to print and distribute, it will take much more time. So by this system the utility form can be access by students and staffs also no need to print and distribute the form.

The loss of data is possible to occur as student fill the form in paper it can easily lost, torn, or damaged. So by this system, the data is being input straight to the system safely.

2. Requirement analysis

- Analyze the results of your fact finding

We conduct research through Google form that we have provided. and the results of the study found that 87 percent of the 41 respondents stated that they experienced some problems when registering their hardware items manually some of them having issue when have to line up to submit the form.

3. Analysis

- Methodology

In the development of House Inventory of Residential College Management System, there are several methods that have been used to collect data based on the system that we will develop. Some of the method were used in this projects are questionnaire and survey, research, observations and documents reviews.

First method that be used for collected the data is by using Google Form. This method to get some feedback from students and college staffs at Sultan Idris Education University (UPSI). The respondents are required to answer some questions that are related to manual way of filling house inventory form that have been implement by residential colleges. Furthermore, residential colleges' staff will key in the data of house inventory form one by one. So with the system that we create, it will facilitate the colleges' staff and students to manage the house inventory form.

Second is research methods that also used in developing this project. Before developed the system, the system analyst will do some research about what are going to do. For example, system analyst will search what is suitable and what the important matters to put in the system about House Inventory of Residential College Management System. We have searched for materials on the internet, so that the information that have been collected can be applied while we developing this project. The research method aims to obtain reference material that can be used as a seconder source.

Observation method also can aid the developer to study and analyze on the weaknesses of existing method. Besides that, we also have surveyed students' opinion on the existing system. Through the observation, we find that the existing method was not systematic. That's one of the reasons of why we build the new system to provide a good service for students and colleges' staff.

Last but not least, we also used documents reviews method in developing our project. After receiving the feedback from the respondent, we will review and

checked the data one by one. From the data we can resolve the problems encountered to finish our project. It will give us the useful information from student and colleges' staff perspective about our system.

- **Techniques**

The techniques of system development that we used is Formal System Development Technique. The Formal Methods Model is an approach to Software Engineering that applies mathematical methods or techniques to the process of developing complex software systems. The approach uses a formal specification language to define each characteristic of the system. The language is very particular, and employs a unique syntax whose components includes objects, relations, and rules.

Formal system development follows the system development lifecycle in a step by step sequence. First system is investigated, then analyzed, designed and more. The advantage that we used this technique is that it allows for a large degree of management control. System control is important when development projects involve significant time and expense.

- **Tools**

Notepad : Notepad is a generic text editor included with all versions of [Microsoft Windows](#) that allows you to create, open, and read [plaintext files](#). If the file contains special formatting or is not a plaintext file, it cannot be read in Microsoft Notepad. The image is a small example of what the Microsoft Notepad may look like while running.

Excel : Excel is a [spreadsheet](#) application developed and published by [Microsoft](#). It is part of the [Microsoft Office](#) suite of productivity software. Excel organizes data in [columns](#) and [rows](#). Rows and columns intersect at a space called a [cell](#). Each cell can contain a single of data, such as text, a numerical value, or a [formula](#). We used excel to do the gantt chart and a gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time.

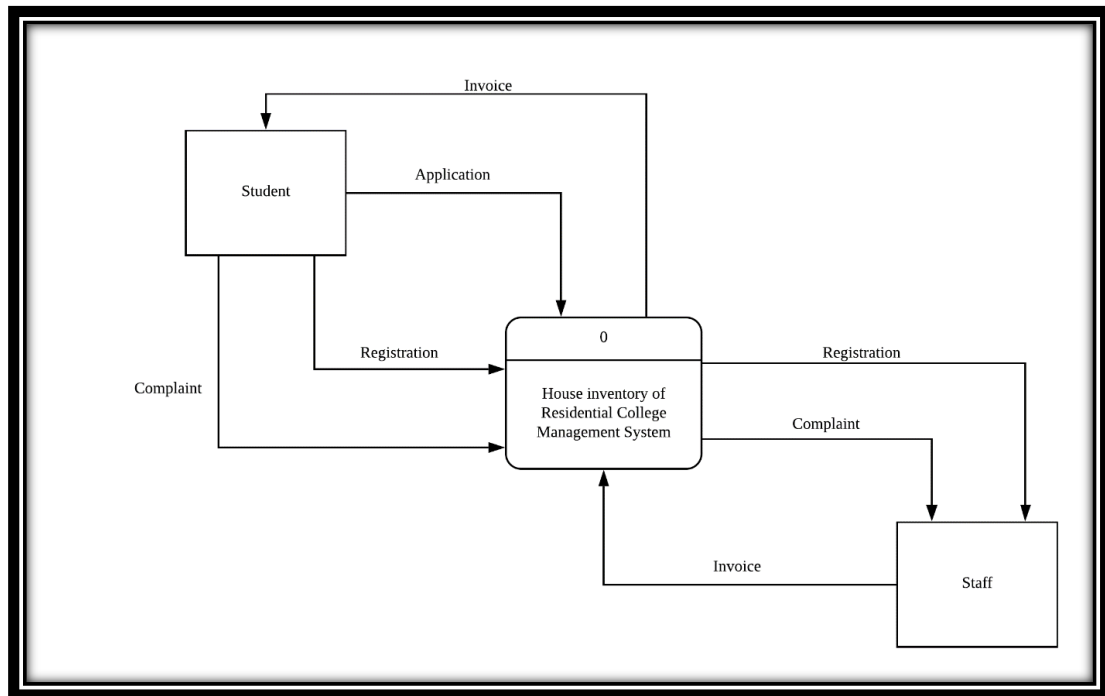
Lucid chart : Lucidchart is a web-based proprietary platform that is used to allow users to collaborate on drawing, revising and sharing charts and diagrams.

Lucidchart is the ultimate diagramming tool that provides businesses and professional individuals with a web-based, user friendly flowchart platform loaded with multiple features and capabilities to help them with their diagramming needs. Lucidchart is an ideal diagram tool for brainstorming and managing projects. The tool also works smoothly with popular web applications and business systems, including Google Apps. Lucidchart is so intuitive that it is used in many industries, including engineering, web design and development, and business sectors.

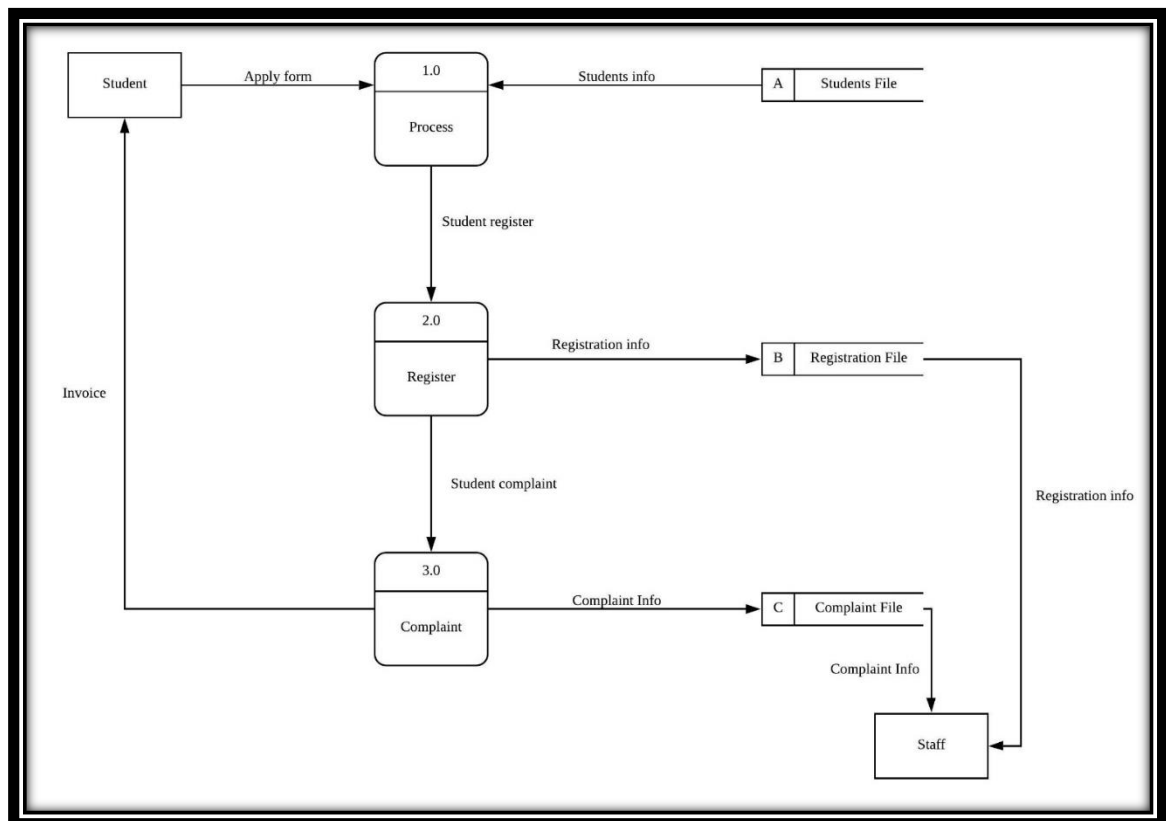
Adobe xd : Adobe XD is a vector-based user experience design tool for web apps and mobile apps, developed and published by Adobe Inc. It is available for macOS and Windows, although there are versions for iOS and Android to help preview the result of work directly on mobile devices. XD supports website wireframing and creating click-through prototypes. Adobe XD works for individual designers and developers wanting to create in a vector format. Adobe XD also works for teams that want to create user experiences and collaborate on the process. You can even build a design system that helps keep everyone on track with a single project.

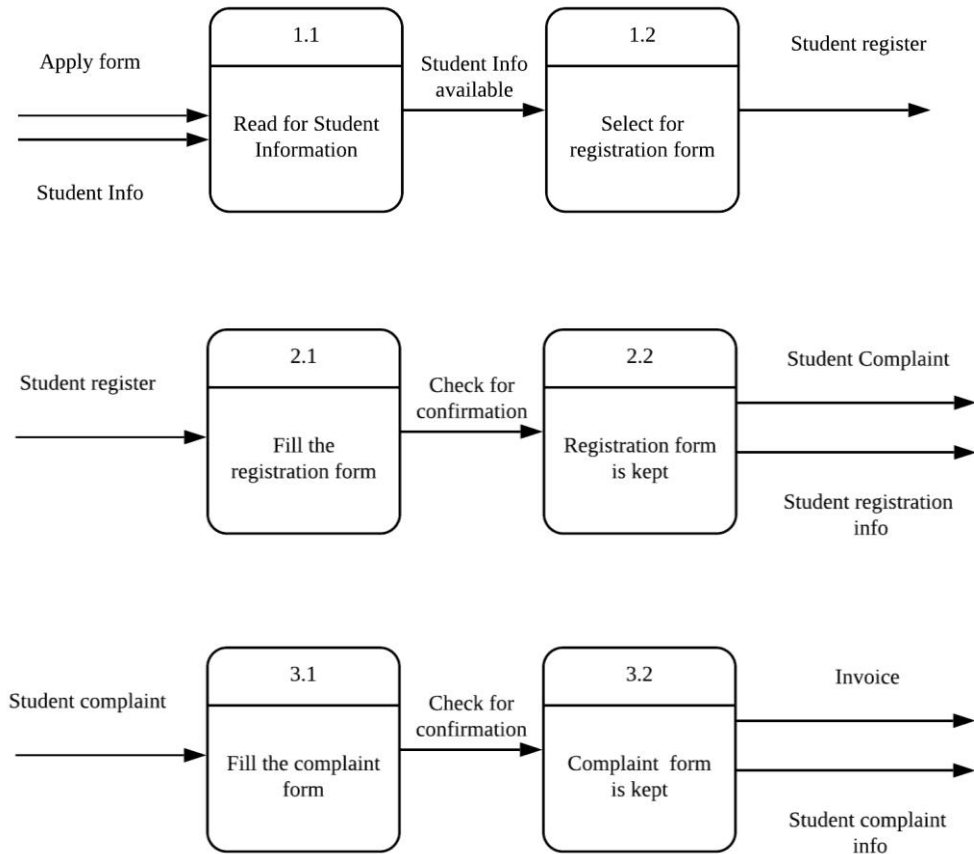
DevC++ : Dev-C++, developed by [Bloodshed Software](#), is a fully featured graphical IDE (Integrated Development Environment), which is able to create Windows or console-based C++ programs using the MinGW compiler system. We used DevC++ for running our systems coding.

- Context Diagram



- DFD 0, at least one DFD level 1



Level 1 DFD

- **Process Description Tools**

House Inventory of Residential College Management System is a system that provide house inventory form access to all necessary appliance for student and residential colleges' staffs. In order for the students to register in the system, they must fill in the form given. Student need to fill out the information such as name, matric number, and their house number. After they finished filled in the system, they can use the system to fill the inventory form, register any electrical stuffs and also can make any complaint regarding any damages in the house. The staffs need to log in to the system to can get the form directly from the system.

- **Structured English.**

If students fill inventory form after their got into college

THEN the college staff received the information about the students AND

If the students finished register in the system

THEN the students can register any electrical stuffs

IF students can make any complaint regarding any damages in the house

THEN staff can get the complaint directly from the system

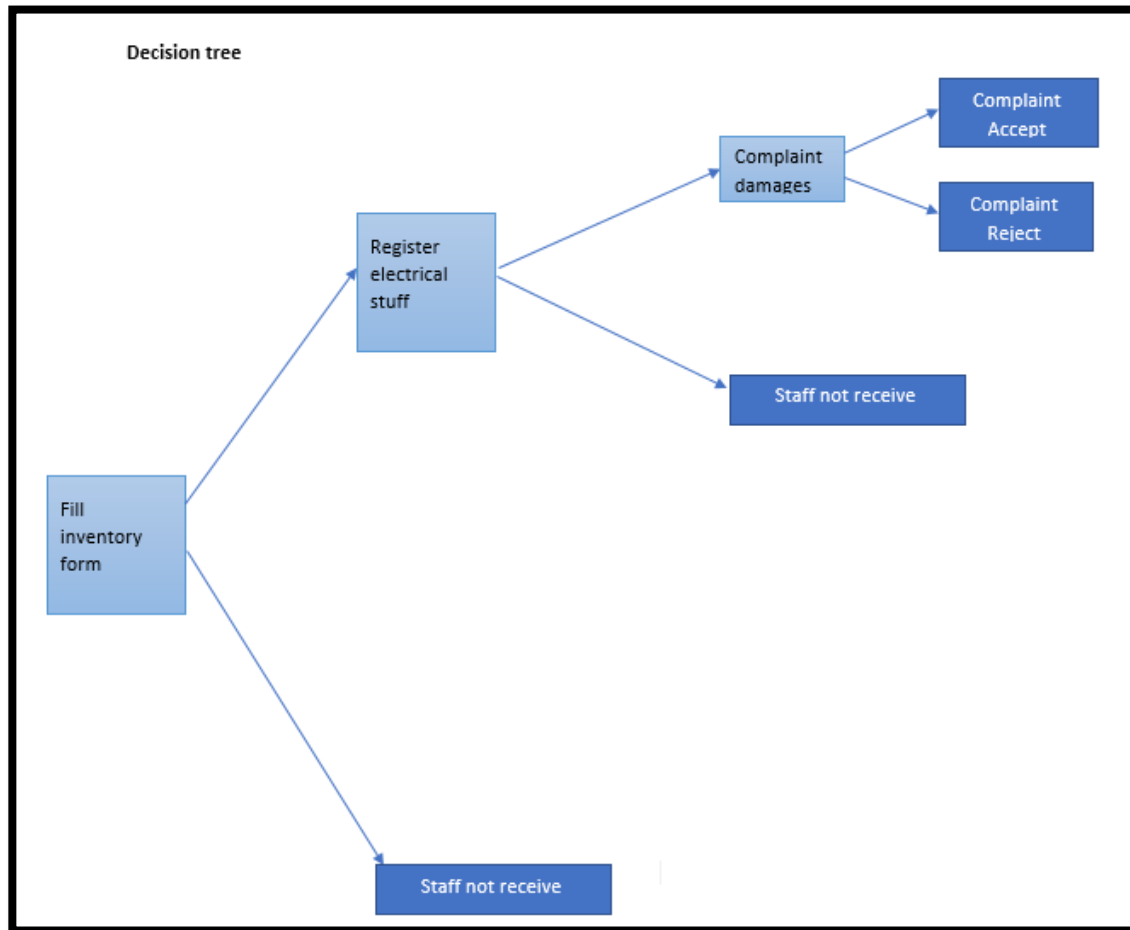
ELSE

No house inventory form

- **Decision table**

Condition	1	2	3	4	5	6	7	8
Fill inventory form	Y	Y	Y	Y	N	N	N	N
Register electrical stuff	Y	Y	N	N	Y	Y	N	N
Complaint damage	Y	N	Y	N	Y	N	Y	N
Staff receive the form	X	X	X	X				
Staff not received the form					X	X	X	X
Complaint accept	X	X			X		X	
Complaint reject			X	X		X		X

- *Object oriented modeling diagram (optional)**



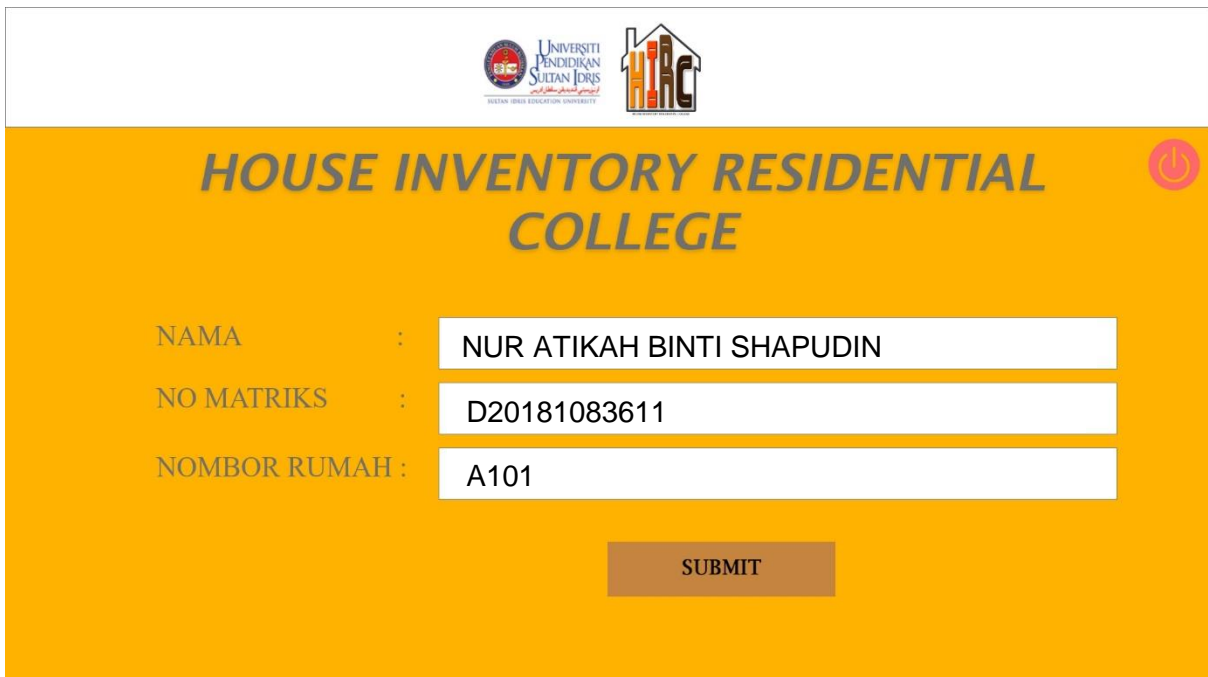
DESIGN

For the design, my group have been decided to use Adobe XD to design the interface of the system. We also develop the prototypes in the design. There are some functions regarding our system which are students can update their inventory form, register their electrical stuff and also can make complaint due to damage that occurred. Besides, staff also can use the system by using the system but they can login with their password to print out the inventory form and damage that have been reported by students.

This is our main interface of House Inventory Residential College system.



If as a student want to fill up for inventory form, student just need to click the Inventory Form button and it will lead to the next interface which required student to fill up the name, matrices number and house number section. And student just need to submit it.



Next, student will be displayed the list of inventory and need to fill in with Y (Yes) or N (No) for this field.




Sila tandakan Y/N

RUANG TAMU		TANDAS		
Pintu :	<input checked="" type="checkbox"/>	Pintu :	<input checked="" type="checkbox"/>	2-Paip Mandi : <input checked="" type="checkbox"/>
Sinki :	<input checked="" type="checkbox"/>	Lampu :	<input checked="" type="checkbox"/>	
Kipas :	<input checked="" type="checkbox"/>	2-Cermin :	<input checked="" type="checkbox"/>	
Lampu :	<input checked="" type="checkbox"/>	2-Sinki :	<input checked="" type="checkbox"/>	
2-Soket :	<input checked="" type="checkbox"/>	3-Pili Air :	<input checked="" type="checkbox"/>	
Rak Kasut :	<input checked="" type="checkbox"/>	2-Hos Air :	<input checked="" type="checkbox"/>	

SIMPAN

After student save your first section in inventory form, it will appeared another field for he or she to fill it.




Sila tandakan Y/N

BILIK					
Pintu :	<input checked="" type="checkbox"/>	2-Meja Belajar :	<input checked="" type="checkbox"/>	Tingkap :	<input checked="" type="checkbox"/>
Kipas :	<input checked="" type="checkbox"/>	2-Bangku Belajar :	<input checked="" type="checkbox"/>	Langsir :	<input checked="" type="checkbox"/>
2-Almari :	<input checked="" type="checkbox"/>	2-Rak Buku :	<input checked="" type="checkbox"/>	Barang Elektrik :	KETTLE
2-Tilam :	<input checked="" type="checkbox"/>	2-Lampu :	<input checked="" type="checkbox"/>		
2-Katil :	<input checked="" type="checkbox"/>	2-Soket :	<input checked="" type="checkbox"/>		
2-Bantal :	<input checked="" type="checkbox"/>	2-Suis Lampu dan Kipas :	<input checked="" type="checkbox"/>		

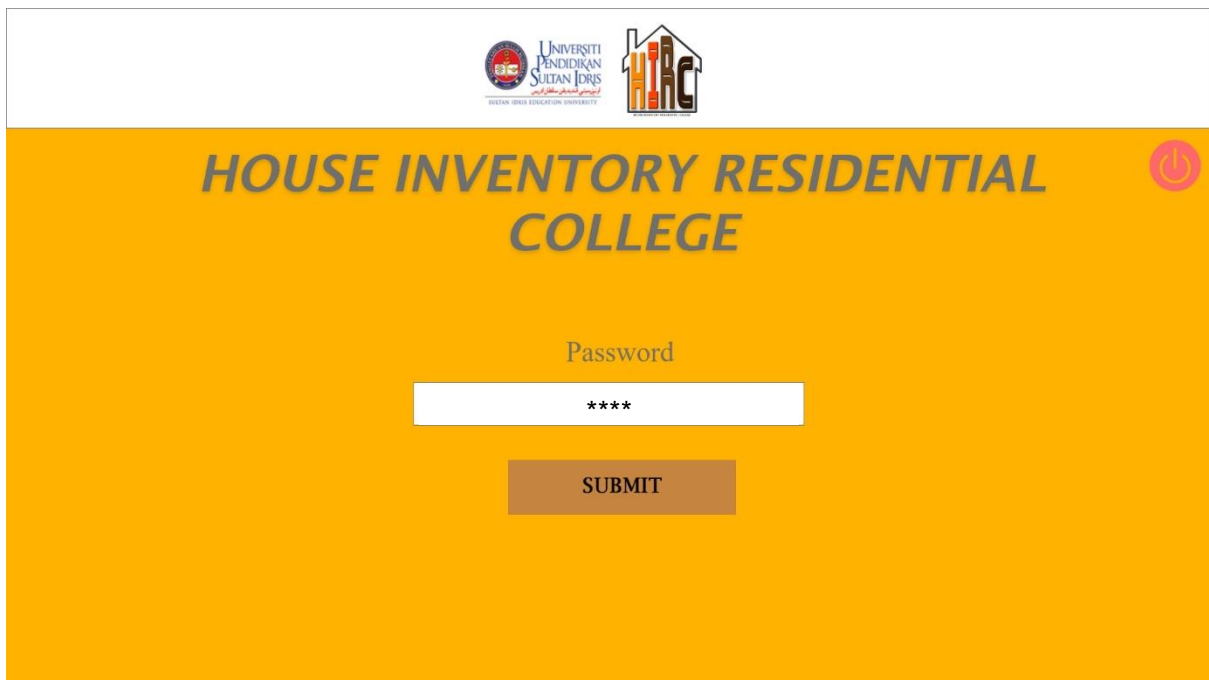
SIMPAN

Later, after student have done filling up the information about inventory form, if student want to make a report about damage, he or she just need to click the Complaint Damage button. Then, it will appear this interface for student to submit the damage.



The screenshot shows a web form titled "HOUSE INVENTORY RESIDENTIAL COLLEGE". At the top, there are logos for "UNIVERSITI PENDIDIKAN SULTAN IDRIS" and a house icon with the letters "HRC". The form has a yellow background. It contains four input fields with labels on the left: "NAMA" (filled with "NUR ATIKAH BINTI SHAPUDIN"), "NO MATRIKS" (filled with "D20181083611"), "NOMBOR RUMAH" (filled with "A101"), and "ADUAN ANDA" (filled with "PINTU TANDAS ROSAK"). A "SUBMIT" button is at the bottom right.

If as a staff, they can check the information regarding student's inventory form and complaint damage by click the Login button at the main interface of system. And it will lead to the login interface.



The screenshot shows a login interface titled "HOUSE INVENTORY RESIDENTIAL COLLEGE". It features the same logos as the previous form. Below the title, there is a "Password" label and a text input field containing five asterisks "*****". A "SUBMIT" button is located below the input field.

After staff have already done login process it required to choose whether staff want to print inventory report or damage complaint. Staff can also view their information without need to print it.



If staff choose whether ‘Cetak Borang Inventori Pelajar’ or ‘Cetak Borang Aduan Pelajar’, it will let it choose the house number that they need to view or print.










A101	A102	A103	A104
A105	A106	A107	A108
A109	A110	A111	A112







NAMA :	NUR ATIKAH BINTI SHAPUDIN
NO MATRIKS :	D20181083611
NOMBOR RUMAH :	A101

Inventory Form



NAMA	:	NUR ATIKAH BINTI SHAPUDIN	
NO MATRIKS	:	D20181083611	
NOMBOR RUMAH	:	A101	

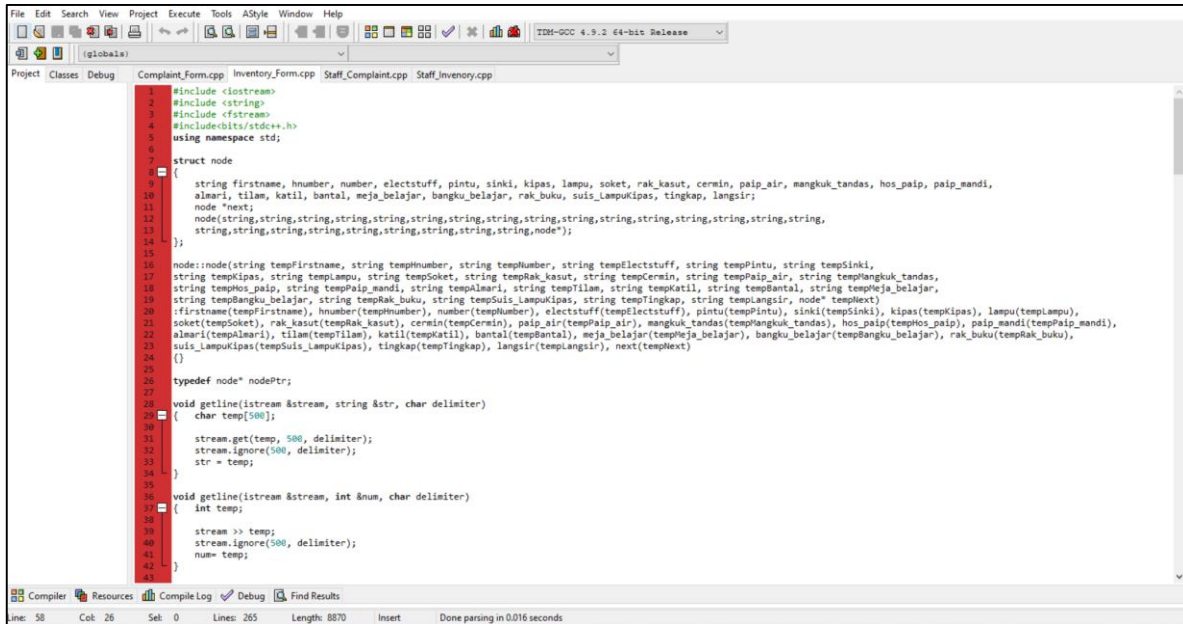
Complaint Damage

The blue light button that showed above will direct to staff for viewing the inventory form and complaint damage made by student. So, the data will be print out if staff need in paper.

Implementation and Coding

We divide the coding into four (4) segment; Inventory_Form, Complaint_Form, Staff_Inventroy and Staff_Complaint. There are two (2) databases; HIRC_DB and ComplaintDB. The coding will be shown below:

Inventory_Form

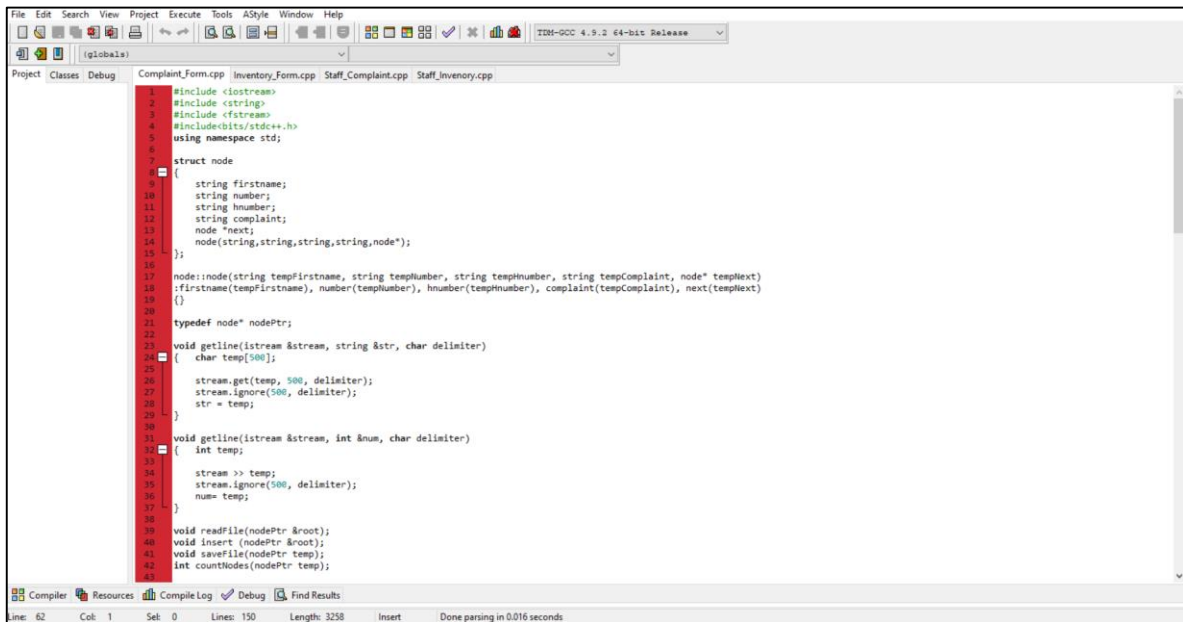


```

1 #include <iostream>
2 #include <string>
3 #include <fstream>
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 struct node
8 {
9     string firstname, hnumber, number, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
10     almar, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_Lampukipas, tingkap, langsir;
11     node *next;
12     node(string,string,string,string,string,string,string,string,string,string,string,string,string,string,node*);
13 }
14
15 node::node(string tempFirstname, string tempNumber, string tempNumber, string tempElectstuff, string tempPintu, string tempSinki,
16 string tempKipas, string tempLampu, string tempSoket, string tempRak_kasut, string tempCermin, string tempPaip_air, string tempMangkuk_tandas,
17 string tempHos_paip, string tempPaip_mandi, string tempAlmar, string tempTilam, string tempKatil, string tempBantal, string tempMeja_belajar,
18 string tempBangku_belajar, string tempRak_buku, string tempSuis_Lampukipas, string tempTingkap, string tempLangsir, node* tempnext)
19 {
20     firstname(tempFirstname), hnumber(tempNumber), number(tempNumber), electstuff(tempElectstuff), pintu(tempPintu), sinki(tempSinki), kipas(tempKipas), lampu(tempLampu),
21     soket(tempSoket), rak_kasut(tempRak_kasut), cermin(tempCermin), paip_air(tempPaip_air), mangkuk_tandas(tempMangkuk_tandas), hos_paip(tempHos_paip), paip_mandi(tempPaip_mandi),
22     almar(tempAlmar), tilam(tempTilam), katil(tempKatil), bantal(tempBantal), meja_belajar(tempMeja_belajar), bangku_belajar(tempBangku_belajar), rak_buku(tempRak_buku),
23     suis_Lampukipas(tempSuis_Lampukipas), tingkap(tempTingkap), langsir(tempLangsir), next(tempnext)
24 }
25
26 typedef node* nodePtr;
27
28 void getline(istream &stream, string &str, char delimiter)
29 {
30     char temp[500];
31
32     stream.get(temp, 500, delimiter);
33     stream.ignore(500, delimiter);
34     str = temp;
35 }
36
37 void getline(istream &stream, int &num, char delimiter)
38 {
39     int temp;
40
41     stream >> temp;
42     stream.ignore(500, delimiter);
43     num= temp;
44 }
45
46 int main()
47 {
48     // ...
49 }

```

Complaint_Form



```

1 #include <iostream>
2 #include <string>
3 #include <fstream>
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 struct node
8 {
9     string firstname;
10     string number;
11     string hnumber;
12     string complaint;
13     node *next;
14     node(string,string,string,string,node*);
15 }
16
17 node::node(string tempFirstname, string tempNumber, string tempNumber, string tempComplaint, node* tempnext)
18 {
19     firstname(tempFirstname), number(tempNumber), hnumber(tempNumber), complaint(tempComplaint), next(tempnext)
20 }
21
22 typedef node* nodePtr;
23
24 void getline(istream &stream, string &str, char delimiter)
25 {
26     char temp[500];
27
28     stream.get(temp, 500, delimiter);
29     stream.ignore(500, delimiter);
30     str = temp;
31 }
32
33 void getline(istream &stream, int &num, char delimiter)
34 {
35     int temp;
36
37     stream >> temp;
38     stream.ignore(500, delimiter);
39     num= temp;
40 }
41
42 void readFile(nodePtr &root);
43 void insert (nodePtr &root);
44 void saveFile(nodePtr temp);
45 int countNodes(nodePtr temp);
46
47 int main()
48 {
49     // ...
50 }

```

Staff_Inventory

The screenshot shows a C++ development environment with a project named "Complaint_Form.cpp". The main file being edited is "Staff_Inventory.cpp". The code defines a linked list structure for storing complaint information.

```

1 #include <iostream>
2 #include <conio.h>
3 #include <string>
4 #include <fstream>
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 struct node
9 {
10     string firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
11     almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_lampukipas, tingkap, langsir;
12     node *next;
13     node(string,string,string,string,string,string,string,string,string,string,string,string,string,node*){}
14 };
15
16 node::node(string tempFirstname, string tempNumber, string tempHnumber, string tempElectstuff, string tempPintu, string tempSinki,
17 string tempKipas, string tempLampu, string tempSoket, string tempRak_kasut, string tempCermin, string tempPaip_air, string tempMangkuk_tandas,
18 string tempHos_paip, string tempPaip_mandi, string tempAlmari, string tempTilam, string tempKatil, string tempBantal, string tempMeja_belajar,
19 string tempBangku_belajar, string tempRak_buku, string tempSuis_lampukipas, string tempTingkap, string tempLangsir, node* tempNext)
20 {
21     firstname(tempFirstname), number(tempNumber), hnumber(tempHnumber), electstuff(tempElectstuff), pintu(tempPintu), sinki(tempSinki), kipas(tempKipas),
22     lampu(tempLampu), soket(tempSoket), rak_kasut(tempRak_kasut), cermin(tempCermin), paip_air(tempPaip_air), mangkuk_tandas(tempMangkuk_tandas),
23     hos_paip(tempHos_paip), paip_mandi(tempPaip_mandi), almari(tempAlmari), tilam(tempTilam), katil(tempKatil), bantal(tempBantal), meja_belajar(tempMeja_belajar),
24     bangku_belajar(tempBangku_belajar), rak_buku(tempRak_buku), suis_lampukipas(tempSuis_lampukipas), tingkap(tempTingkap), langsir(tempLangsir), next(tempNext)
25 }
26
27 typedef node* nodePtr;
28
29 void getline(Istream &istream, string &str, char delimiter)
30 {
31     char temp[500];
32     istream.get(temp, 500, delimiter);
33     stream.ignore(500, delimiter);
34     str = temp;
35 }
36
37 void getline(Istream &istream, int &num, char delimiter)
38 {
39     int temp;
40     istream >> temp;
41     stream.ignore(500, delimiter);
42     num = temp;
43 }

```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing "Line: 53 Col: 9 Sel: 0 Lines: 231 Length: 7445 Insert Done parsing in 0.016 seconds".

Staff_Complaint

The screenshot displays the Visual Studio IDE with the following components:

- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help.
- Toolbar:** Includes icons for file operations (Open, Save, Print), building (Build, Rebuild), and debugging (Run, Breakpoints).
- Project Explorer:** Shows the project structure with files: Complaint_Form.cpp, Inventory_Form.cpp, Staff_Complaint.cpp, and Staff_Inventory.cpp.
- Code Editor:** Contains the source code for 'Complaint_Form.cpp'. The code is as follows:


```

1  #include <iostream>
2  #include <conio.h>
3  #include <string>
4  #include <fstream>
5  #include <bits/stdc++.h>
6  using namespace std;
7
8  struct node
9  {
10     string firstname;
11     string number;
12     string hnumber;
13     string complaint;
14     node *next;
15     node(string,string,string,string,node*);
16 };
17
18 node:node(string tempFirstname, string tempNumber, string tempHnumber, string tempComplaint, node* tempNext)
19 :firstname(tempFirstname), number(tempNumber), hnumber(tempHnumber), complaint(tempComplaint), next(tempNext)
20 {}
21
22 typedef node* nodePtr;
23
24 void getline(istream &stream, string &str, char delimiter)
25 {
26     char temp[500];
27     stream.get(temp, 500, delimiter);
28     stream.ignore(500, delimiter);
29     str = temp;
30 }
31
32 void getline(istream &stream, int &num, char delimiter)
33 {
34     int temp;
35     stream >> temp;
36     stream.ignore(500, delimiter);
37     num= temp;
38 }
39
40 void password();
41 void mainmenu();
42 void readFile(nodePtr &root);
43 void printList(nodePtr temp);
      
```
- Status Bar:** Shows 'Done parsing in 0.016 seconds'.

System testing

Inventory_Form

```
=====
|  House Inventory of Residential College Management System  |
=====

Menu: Pilih Pilihan

(1) Isi Borang Inventory Rumah
(2) Simpan Borang Inventory Rumah
(3) Keluar

Masukkan Pilihan Anda ---> 1

Nama: AHMAD FAUZI BIN SUKARDI
No.Matrik: D20181083618
Nombor Rumah: 61112
Inventori (Sila taip Y atau N)
-Ruang Tamu-
Pintu: Y
Sinki: Y
Kipas: Y
Lampu: Y
2-Soket: Y
```

```
2-Soket: Y
Rak Kasut: Y
-Tandas-
3-Pintu: Y
Lampu: Y
2-Cermin: Y
2-Sinki: Y
3-Pili Air: Y
2-Hos Air: Y
2-Paip Mandi: Y
-Bilik-
Pintu: Y
Kipas: Y
2-Almari: Y
2-Tilam: Y
```

```

2-Tilam: Y
2-Katil: Y
2-Bantal: Y
2-Meja Belajar: Y
2-Bangku Belajar: Y
2-Rak Buku: Y
2-Lampu: Y
2-Soket: Y
2-Suis Lampu dan Kipas: Y
Tingkap: Y
Langsir: Y
Barang Elektrik:Kettle, Iron, Toaster, Hair Dryer, Printer
    
```

```

=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Isi Borang Inventory Rumah
(2) Simpan Borang Inventory Rumah
(3) Keluar

Masukkan Pilihan Anda ---> 2
    
```

```

=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Isi Borang Inventory Rumah
(2) Simpan Borang Inventory Rumah
(3) Keluar

Masukkan Pilihan Anda --->
    
```

```

=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Isi Borang Inventory Rumah
(2) Simpan Borang Inventory Rumah
(3) Keluar

Masukkan Pilihan Anda ---> 3

-----
Process exited after 2.759 seconds with return value 0
Press any key to continue . . .
    
```

Complaint_Form

```
=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Borang Aduan
(2) Simpan Borang Aduan
(3) Keluar

Masukkan Pilihan Anda ---> 1

Nama: AHMAD FAUZI BIN SUKARDI
Nombor Matrik: D20181083618
Nombor Rumah : 61112
Aduan Anda: Lampu tandas rosak
```

```
=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Borang Aduan
(2) Simpan Borang Aduan
(3) Keluar

Masukkan Pilihan Anda ---> 2

=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Borang Aduan
(2) Simpan Borang Aduan
(3) Keluar

Masukkan Pilihan Anda --->
```

```
=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Borang Aduan
(2) Simpan Borang Aduan
(3) Keluar

Masukkan Pilihan Anda ---> 3

-----
Process exited after 122.9 seconds with return value 0
Press any key to continue . . .
```



```
=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Borang Aduan
(2) Simpan Borang Aduan
(3) Keluar

Masukkan Pilihan Anda ---> 3

-----
Process exited after 122.9 seconds with return value 0
Press any key to continue . . .
```

Staff_Inventory

```
=====
|   House Inventory of Residential College Management System   |
=====
Enter the password to Login:***
```

```
=====
|   House Inventory of Residential College Management System   |
=====

Menu: Pilih Pilihan

(1) Cetak Borang Inventori Pelajar
(2) Quit

Masukkan Pilihan Anda ---> 1

Name: AHMAD FAUZI BIN SUKARDI
House Number: D20181083618
Matrix Number: 61112
Inventori
  -Ruang Tamu-
Pintu: Y
Sinki: Y
Kipas: Y
Lampu: Y
2Soket: Y
Rak Kasut: Y
  -Tandas-
3-Pintu: Y
Lampu: Y
2-Cermin: Y
2-Sinki: Y
3-Pili air: Y
2-Hos air: Y
```

```
2-Hos air: Y
2-Paip mandi: Y
2-Mangkuk tandas: Y
  -Bilik-
Pintu: Y
Kipas: Y
2-Almari: Y
2-Tilam: Y
2-Katil: Y
2-Bantal: Y
2-Meja belajar: Y
2-Bangku belajar: Y
2-Rak buku: Y
2-Lampu: Y
2-Soket: Y
2-Suis Lampu dan Kipas: Y
2-Tingkap: Y
2-Langsir: Y
Barang Elektrik: Kettle, Iron, Toaster, Hair Dryer, Printer
```

Staff_Complaint

```
=====
| House Inventory of Residential College Management System |
=====
Masukkan Kata Laluan:****
```

```
=====
| House Inventory of Residential College Management System |
=====

Menu: Pilih Pilihan

(1) Cetak Borang Aduan Pelajar
(2) Quit

Masukkan Pilihan Anda ---> 1
Nama: AHMAD FAUZI BIN SUKARDI
Nombor Matrik: 61112
Nombor Rumah: D20181083618
Aduan anda: Lampu tandas rosak
```

Inventory_Form

```
#include <iostream>
#include <string>
#include <fstream>
#include <bits/stdc++.h>
using namespace std;

struct node
{
    string firstname, hnumber, number, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
    almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_LampuKipas, tingkap, langsir;
    node *next;
    node(string,string,string,string,string,string,string,string,string,string,string,string,string,string,string,
    string,string,string,string,string,string,string,string,node*);
};

node::node(string tempFirstname, string tempHnumber, string tempNumber, string tempElectstuff, string tempPintu, string tempSinki,
string tempKipas, string tempLampu, string tempSoket, string tempRak_kasut, string tempCermin, string tempPaip_air, string tempMangkuk_tandas,
string tempHos_paip, string tempPaip_mandi, string tempAlmari, string tempTilam, string tempKatil, string tempBantal, string tempMeja_belajar,
string tempBangku_belajar, string tempRak_buku, string tempSuis_LampuKipas, string tempTingkap, string tempLangsir, node* tempNext)
:firstname(tempFirstname), hnumber(tempHnumber), number(tempNumber), electstuff(tempElectstuff), pintu(tempPintu), sinki(tempSinki), kipas(tempKipas), lampu(tempLampu),
soket(tempSoket), rak_kasut(tempRak_kasut), cermin(tempCermin), paip_air(tempPaip_air), mangkuk_tandas(tempMangkuk_tandas), hos_paip(tempHos_paip), paip_mandi(tempPaip_mandi),
almari(tempAlmari), tilam(tempTilam), katil(tempKatil), bantal(tempBantal), meja_belajar(tempMeja_belajar), bangku_belajar(tempBangku_belajar), rak_buku(tempRak_buku),
suis_LampuKipas(tempSuis_LampuKipas), tingkap(tempTingkap), langsir(tempLangsir), next(tempNext)
{}
}
```

```
typedef node* nodePtr;

void getline(istream &stream, string &str, char delimiter)
{
    char temp[500];

    stream.get(temp, 500, delimiter);
    stream.ignore(500, delimiter);
    str = temp;
}

void getline(istream &stream, int &num, char delimiter)
{
    int temp;

    stream >> temp;
    stream.ignore(500, delimiter);
    num= temp;
}

void readFile(nodePtr &root);
void insert (nodePtr &root);
void saveFile(nodePtr temp);
int countNodes(nodePtr temp);
```

```
int main()
{
    int choice;
    nodePtr root = NULL;
    readFile(root);

    do
    {
        cout<<"\n\t\t=====";
        cout<<"\n\t\t|   House Inventory of Residential College Management System   |"<<endl;
        cout<<"\t\t=====";
        cout << "\n\n\t\tMenu: Pilih Pilihan\n\n";
        cout << "\t\t(1) Isi Borang Inventory Rumah\n";
        cout << "\t\t(2) Simpan Borang Inventory Rumah\n";
        cout << "\t\t(3) Keluar\n\n";
        cout << "\t\tMasukkan Pilihan Anda ---> ";

        cin >> choice;

        if (1 <= choice && choice <= 2)
        {
            switch (choice)
            {
                case 1:
                    insert(root);
                    break;
                case 2:
                    saveFile(root);
                    break;
                default:
                    cout << "Pilihan Tidak Sah. Sila Masukkan Semula\n\n";
                    break;
            }
        }
    }
    while (choice != 3);
    saveFile(root);
    return 0;
}
```

```
void readFile(nodePtr &root)
{
    int numPhones;
    string firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
    almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_LampuKipas, tingkap, langsir;
    ifstream infile ("HIRC_DB.txt", ios::in);
    infile >> numPhones;
    infile.ignore(500, '\n');
    for (int count = 0; count < numPhones; count++)
    {
        getline(infile, firstname, '\n');
        getline(infile, number, '\n');
        getline(infile, hnumber, '\n');
        getline(infile, electstuff, '\n');
        getline(infile, pintu, '\n');
        getline(infile, sinki, '\n');
        getline(infile, kipas, '\n');
        getline(infile, lampu, '\n');
        getline(infile, soket, '\n');
        getline(infile, rak_kasut, '\n');
        getline(infile, cermin, '\n');
        getline(infile, paip_air, '\n');
        getline(infile, mangkuk_tandas, '\n');
        getline(infile, hos_paip, '\n');
        getline(infile, paip_mandi, '\n');
        getline(infile, almari, '\n');
        getline(infile, tilam, '\n');
        getline(infile, katil, '\n');
        getline(infile, bantal, '\n');
        getline(infile, meja_belajar, '\n');
        getline(infile, bangku_belajar, '\n');
        getline(infile, rak_buku, '\n');
        getline(infile, suis_LampuKipas, '\n');
        getline(infile, tingkap, '\n');
        getline(infile, langsir, '\n');
        root = new node (firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
        almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_LampuKipas, tingkap, langsir, root);
    }
}
```

```

void insert (nodePtr &root)
{
    string firstname,number,hnumber,electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
    almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_Lampukipas, tingkap, langsir;

    cout << "\nNama: ";
    cin.ignore(500, '\n');
    getline(cin, firstname, '\n');
    cout << "\nNo.Matrik: ";
    getline(cin, number, '\n');
    cout << "\nNombor Rumah: ";
    getline(cin, hnumber, '\n');
    cout << "\nInventori (Sila taip Y atau N)";
    cout << "\n\t-Ruang Tamu-";
    cout << "\nPintu: ";
    getline(cin, pintu, '\n');
    cout << "\nSinki: ";
    getline(cin, sinki, '\n');
    cout << "\nKipas: ";
    getline(cin, kipas, '\n');
    cout << "\nLampu: ";
    getline(cin, lampu, '\n');
    cout << "\n2-Soket: ";
    getline(cin, soket, '\n');
    cout << "\nRak Kasut: ";
    getline(cin, rak_kasut, '\n');
    cout << "\n\t-Tandas-";
    cout << "\n3-Pintu: ";
    getline(cin, pintu, '\n');
    cout << "\nLampu: ";
    getline(cin, lampu, '\n');
    cout << "\n2-Cermin: ";
    getline(cin, cermin, '\n');
    cout << "\n2-Sinki: ";
    getline(cin, sinki, '\n');
    cout << "\n3-Pili Air: ";
    getline(cin, paip_air, '\n');
    cout << "\n2-Hos Air: ";
    getline(cin, hos_paip, '\n');
    cout << "\n2-Paip Mandi: ";
    getline(cin, paip_mandi, '\n');
    cout << "\n\t-Bilik-";
    cout << "\nPintu: ";
}

```

```

    cout << "\nPintu: ";
    getline(cin, pintu, '\n');
    cout << "\nKipas: ";
    getline(cin, kipas, '\n');
    cout << "\n2-Almari: ";
    getline(cin, almari, '\n');
    cout << "\n2-Tilam: ";
    getline(cin, tilam, '\n');
    cout << "\n2-Katil: ";
    getline(cin, katil, '\n');
    cout << "\n2-Bantal: ";
    getline(cin, bantal, '\n');
    cout << "\n2-Meja Belajar: ";
    getline(cin, meja_belajar, '\n');
    cout << "\n2-Bangku Belajar: ";
    getline(cin, bangku_belajar, '\n');
    cout << "\n2-Rak Buku: ";
    getline(cin, rak_buku, '\n');
    cout << "\n2-Lampu: ";
    getline(cin, lampu, '\n');
    cout << "\n2-Soket: ";
    getline(cin, soket, '\n');
    cout << "\n2-Suis Lampu dan Kipas: ";
    getline(cin, suis_Lampukipas, '\n');
    cout << "\nTingkap: ";
    getline(cin, tingkap, '\n');
    cout << "\nLangsir: ";
    getline(cin, langsir, '\n');
    cout << "\nBarang Elektrik: ";
    getline(cin, electstuff, '\n');

    root = new node (firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
    almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_Lampukipas, tingkap, langsir, root);
}

```

```

void saveFile(nodePtr temp)
{
    int count = countNodes(temp);
    ofstream outFile("inventory_saved.txt", ios::out);

    outFile << count << "\n";
    while (temp != NULL)
    {
        outFile << temp->firstname << "\n";
        outFile << temp->hnumber << "\n";
        outFile << temp->number << "\n";
        outFile << temp->pintu << "\n";
        outFile << temp->sinki << "\n";
        outFile << temp->kipas << "\n";
        outFile << temp->lampu << "\n";
        outFile << temp->soket << "\n";
        outFile << temp->rak_kasut << "\n";
        outFile << temp->pintu << "\n";
        outFile << temp->lampu << "\n";
        outFile << temp->cermin << "\n";
        outFile << temp->sinki << "\n";
        outFile << temp->paip_air << "\n";
        outFile << temp->hos_paip << "\n";
        outFile << temp->paip_mandi << "\n";
        outFile << temp->mangkuk_tandas << "\n";
        outFile << temp->pintu << "\n";
        outFile << temp->kipas << "\n";
        outFile << temp->almari << "\n";
        outFile << temp->tilam << "\n";
        outFile << temp->katil << "\n";
        outFile << temp->bantal << "\n";
        outFile << temp->meja_belajar << "\n";
        outFile << temp->bangku_belajar << "\n";
        outFile << temp->rak_buku << "\n";
        outFile << temp->lampu << "\n";
        outFile << temp->soket << "\n";
        outFile << temp->suis_LampuKipas << "\n";
        outFile << temp->tingkap << "\n";
        outFile << temp->langsir << "\n";
        outFile << temp->electstuff << "\n";
        temp = temp->next;
    }
    cout << "\n";
}

```

```

int countNodes(nodePtr temp)
{
    int countB = 0;
    while (temp != NULL)
    {
        countB++;
        temp = temp->next;
    }
    return countB;
}

```

Complaint_Form

```
#include <iostream>
#include <string>
#include <fstream>
#include <bits/stdc++.h>
using namespace std;

struct node
{
    string firstname;
    string number;
    string hnumber;
    string complaint;
    node *next;
    node(string,string,string,string,node*);
};

node::node(string tempFirstname, string tempNumber, string tempHNumber, string tempComplaint, node* tempNext)
:firstname(tempFirstname), number(tempNumber), hnumber(tempHNumber), complaint(tempComplaint), next(tempNext)
{}

typedef node* nodePtr;

void getline(istream &stream, string &str, char delimiter)
{
    char temp[500];

    stream.get(temp, 500, delimiter);
    stream.ignore(500, delimiter);
    str = temp;
}

void getline(istream &stream, int &num, char delimiter)
{
    int temp;

    stream >> temp;
    stream.ignore(500, delimiter);
    num= temp;
}

void readFile(nodePtr &root);
void insert (nodePtr &root);
void saveFile(nodePtr temp);
int countNodes(nodePtr temp);
```

```
int main()
{
    int choice;
    nodePtr root = NULL;
    readFile(root);

    do
    {
        cout<<"\n\t\t=====";
        cout<<"\n\t\t|   House Inventory of Residential College Management System   |"<<endl;
        cout<<"\t\t=====";
        cout << "\n\n\t\tMenu: Pilih Pilihan\n\n";
        cout << "\t\t(1) Borang Aduan\n";
        cout << "\t\t(2) Simpan Borang Aduan\n";
        cout << "\t\t(3) Keluar\n\n";
        cout << "\t\tMasukkan Pilihan Anda ---> ";

        cin >> choice;

        if (1 <= choice && choice <= 2)
        {
            switch (choice)
            {
                case 1:
                    insert(root);
                    break;
                case 2:
                    saveFile(root);
                    break;
                default:
                    cout << "Pilihan Tidak Sah. Sila Masukkan Semula\n\n";
                    break;
            }
        }
    }
    while (choice != 3);
    saveFile(root);
    return 0;
}
```



```
void readFile(nodePtr &root)
{
    int numPhones;
    string firstname, number, hnumber, complaint;
    ifstream infile ("ComplaintDB.txt", ios::in);
    infile >> numPhones;
    infile.ignore(500, '\n');
    for (int count = 0; count < numPhones; count++)
    {
        getline(infile, firstname, '\n');
        getline(infile, number, '\n');
        getline(infile, hnumber, '\n');
        getline(infile, complaint, '\n');

        root = new node (firstname, number, hnumber, complaint, root);
    }
}

void insert (nodePtr &root)
{
    string firstname, number, hnumber, complaint;

    cout << "\nNama: ";
    cin.ignore(500, '\n');
    getline(cin, firstname, '\n');
    cout << "\nNombor Matrik: ";
    getline(cin, number, '\n');
    cout << "\nNombor Rumah : ";
    getline(cin, hnumber, '\n');
    cout << "\nAduan Anda: ";
    getline(cin, complaint, '\n');

    root = new node (firstname, number, hnumber, complaint, root);
}
```

```
void saveFile(nodePtr temp)
{
    int count = countNodes(temp);
    ofstream outFile("complaint_saved.txt", ios::out);

    outFile << count << "\n";
    while (temp != NULL)
    {
        outFile << temp->firstname << "\n";
        outFile << temp->number << "\n";
        outFile << temp->hnumber << "\n";
        outFile << temp->complaint << "\n";
        temp = temp->next;
    }
    cout << "\n";
}

int countNodes(nodePtr temp)
{
    int countB = 0;
    while (temp != NULL)
    {
        countB++;
        temp = temp->next;
    }
    return countB;
}
```

Staff_Inventory

```
#include <iostream>
#include <conio.h>
#include <string>
#include <fstream>
#include <bits/stdc++.h>
using namespace std;

struct node
{
    string firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
    almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_LampuKipas, tingkap, langsir;
    node *next;
    node(string,string,string,string,string,string,string,string,string,string,string,string,string,
    string,string,string,string,string,string,string,string,string,string,node*);
};

node::node(string tempFirstname, string tempNumber, string tempHnumber, string tempElectstuff, string tempPintu, string tempSinki,
string tempKipas, string tempLampu, string tempSoket, string tempRak_kasut, string tempCermin, string tempPaip_air, string tempMangkuk_tandas,
string tempHos_paip, string tempPaip_mandi, string tempAlmari, string tempTilam, string tempKatil, string tempBantal, string tempMeja_belajar,
string tempBangku_belajar, string tempRak_buku, string tempSuis_LampuKipas, string tempTingkap, string tempLangsir, node* tempNext)
:firstname(tempFirstname), number(tempNumber), hnumber(tempHnumber), electstuff(tempElectstuff), pintu(tempPintu), sinki(tempSinki), kipas(tempKipas),
lampu(tempLampu), soket(tempSoket), rak_kasut(tempRak_kasut), cermin(tempCermin), paip_air(tempPaip_air), mangkuk_tandas(tempMangkuk_tandas),
hos_paip(tempHos_paip), paip_mandi(tempPaip_mandi), almari(tempAlmari), tilam(tempTilam), katil(tempKatil), bantal(tempBantal), meja_belajar(tempMeja_belajar),
bangku_belajar(tempBangku_belajar), rak_buku(tempRak_buku), suis_LampuKipas(tempSuis_LampuKipas), tingkap(tempTingkap), langsir(tempLangsir), next(tempNext)
{
}
```

```
typedef node* nodePtr;

void getline(istream &stream, string &str, char delimiter)
{   char temp[500];

    stream.get(temp, 500, delimiter);
    stream.ignore(500, delimiter);
    str = temp;
}

void getline(istream &stream, int &num, char delimiter)
{   int temp;

    stream >> temp;
    stream.ignore(500, delimiter);
    num= temp;
}

void password();
void mainmenu();
void readFile(nodePtr &root);
void printList(nodePtr temp);
int countNodes(nodePtr temp);
```

```

int main(){
password();
getch();
return 0;
}

void mainmenu()
{
    int choice;
    nodePtr root = NULL;
    readFile(root);

    do
    {
        cout<<"\n\t\t=====";
        cout<<"\n\t\t|   House Inventory of Residential College Management System   |"<<endl;
        cout<<"\t\t=====";
        cout << "\n\n\t\tMenu: Pilih Pilihan\n\n";
        cout << "\t\t(1) Cetak Borang Inventori Pelajar\n";
        cout << "\t\t(2) Quit\n\n";
        cout << "\t\tMasukkan Pilihan Anda ----> ";

        cin >> choice;

        if (1 <= choice && choice <= 2)
        {
            switch (choice)
            {
                case 1:
                    printList(root);
                    break;
                default:
                    cout << "Pilihan Tidak Sah. Sila Masukkan Semula\n\n";
                    break;
            }
        }
    } while (choice != 2);
    return;
}

```

```

void readFile(nodePtr &root)
{
    int numPhones;
    string firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
    almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_LampuKipas, tingkap, langsir;
    ifstream infile ("HRC_DB.txt", ios::in);
    infile >> numPhones;
    infile.ignore(500, '\n');
    for (int count = 0; count < numPhones; count++)
    {
        getline(infile, firstname, '\n');
        getline(infile, number, '\n');
        getline(infile, hnumber, '\n');
        getline(infile, electstuff, '\n');
        getline(infile, pintu, '\n');
        getline(infile, sinki, '\n');
        getline(infile, kipas, '\n');
        getline(infile, lampu, '\n');
        getline(infile, soket, '\n');
        getline(infile, rak_kasut, '\n');
        getline(infile, cermin, '\n');
        getline(infile, paip_air, '\n');
        getline(infile, mangkuk_tandas, '\n');
        getline(infile, hos_paip, '\n');
        getline(infile, paip_mandi, '\n');
        getline(infile, almari, '\n');
        getline(infile, tilam, '\n');
        getline(infile, katil, '\n');
        getline(infile, bantal, '\n');
        getline(infile, meja_belajar, '\n');
        getline(infile, bangku_belajar, '\n');
        getline(infile, rak_buku, '\n');
        getline(infile, suis_LampuKipas, '\n');
        getline(infile, tingkap, '\n');
        getline(infile, langsir, '\n');
        root = new node (firstname, number, hnumber, electstuff, pintu, sinki, kipas, lampu, soket, rak_kasut, cermin, paip_air, mangkuk_tandas, hos_paip, paip_mandi,
        almari, tilam, katil, bantal, meja_belajar, bangku_belajar, rak_buku, suis_LampuKipas, tingkap, langsir, root);
    }
}

```

```
void password(void){
system("cls");
char pass[10],ch,password[10]="1234";
int i=0;
cout<<"\n\t\t=====";
cout<<"\n\t\t|   House Inventory of Residential College Management System   |"<<endl;
cout<<"\t\t=====";
printf("\nEnter the password to Login:");
//scanf("%s",pass);
while(ch!=13)
{
ch=getch();

if(ch!=13 && ch!=8){
putch('*');
pass[i] = ch;
i++;
}
}
pass[i] = '\0';

if(strcmp(pass,password)==0){
    printf("Password Matched!!");
    printf("Loading ");
for(i=0;i<=6;i++)
    {
        printf(".");
    }
    system("cls");
    mainmenu();
}
else
printf("\nPassword is invalid!!\a");
printf("\nENTER correct password");
getch();
main();
}
```

```

void printList(nodePtr temp)
{
    while (temp != NULL)
    {
        cout << "\nName: " << temp->firstname;
        cout << "\nHouse Number: " << temp->hnumber;
        cout << "\nMatrix Number: " << temp->number;
        cout << "\nInventori";
        cout << "\n\t-Ruang Tamu-";
        cout << "\nPintu: " << temp->pintu;
        cout << "\nSinki: " << temp->sinki;
        cout << "\nKipas: " << temp->kipas;
        cout << "\nLampu: " << temp->lampu;
        cout << "\n2Soket: " << temp->soket;
        cout << "\nRak Kasut: " << temp->rak_kasut;
        cout << "\n\t-Tandas-";
        cout << "\n3-Pintu: " << temp->pintu;
        cout << "\nLampu: " << temp->lampu;
        cout << "\n2-Cermin: " << temp->cermin;
        cout << "\n2-Sinki: " << temp->sinki;
        cout << "\n3-Pili air: " << temp->paip_air;
        cout << "\n2-Hos air: " << temp->hos_paip;
        cout << "\n2-Paip mandi: " << temp->paip_mandi;
        cout << "\n2-Mangkuk tandas: " << temp->mangkuk_tandas;
        cout << "\n\t-Bilik-";
        cout << "\nPintu: " << temp->pintu;
        cout << "\nKipas: " << temp->kipas;
        cout << "\n2-Almari: " << temp->almari;
        cout << "\n2-Tilam: " << temp->tilam;
        cout << "\n2-Katil: " << temp->katil;
        cout << "\n2-Bantal: " << temp->bantal;
        cout << "\n2-Meja belajar: " << temp->meja_belajar;
        cout << "\n2-Bangku belajar: " << temp->bangku_belajar;
        cout << "\n2-Rak buku: " << temp->rak_buku;
        cout << "\n2-Lampu: " << temp->lampu;
        cout << "\n2-Soket: " << temp->soket;
        cout << "\n2-Suis Lampu dan Kipas: " << temp->suis_LampuKipas;
        cout << "\n2-Tingkap: " << temp->tingkap;
        cout << "\n2-Langsir: " << temp->langsir;
        cout << "\nBarang Elektrik: " << temp->electstuff;
        cout << "\n";
        temp = temp->next;
    }
    cout << "\n\n";
}

```

```

}

int countNodes(nodePtr temp)
{
    int countB = 0;
    while (temp != NULL)
    {
        countB++;
        temp = temp->next;
    }
    return countB;
}

```

Staff_Complaint

```
#include <iostream>
#include <conio.h>
#include <string>
#include <fstream>
#include <bits/stdc++.h>
using namespace std;

struct node
{
    string firstname;
    string number;
    string hnumber;
    string complaint;
    node *next;
    node(string,string,string,string,node*);
};

node::node(string tempFirstname, string tempNumber, string tempHnumber, string tempComplaint, node* tempNext)
:firstname(tempFirstname), number(tempNumber), hnumber(tempHnumber), complaint(tempComplaint), next(tempNext)
{}

typedef node* nodePtr;

void getline(istream &stream, string &str, char delimiter)
{
    char temp[500];

    stream.get(temp, 500, delimiter);
    stream.ignore(500, delimiter);
    str = temp;
}

void getline(istream &stream, int &num, char delimiter)
{
    int temp;

    stream >> temp;
    stream.ignore(500, delimiter);
    num= temp;
}
```



```

void password();
void mainmenu();
void readFile(nodePtr &root);
void printList(nodePtr temp);
int countNodes(nodePtr temp);

int main(){
password();
getch();
return 0;
}

void mainmenu()
{
    int choice;
    nodePtr root = NULL;
    readFile(root);

    do
    {
        cout<<"\n\t\t===== ";
        cout<<"\n\t\t|   House Inventory of Residential College Management System   |"<<endl;
        cout<<"\n\t\t===== ";
        cout << "\n\n\t\tMenu: Pilih Pilihan\n\n";
        cout << "\t\t(1) Cetak Borang Aduan Pelajar\n";
        cout << "\t\t(2) Quit\n\n";
        cout << "\t\tMasukkan Pilihan Anda ---> ";

        cin >> choice;

        if (1 <= choice && choice <= 2)
        {
            switch (choice)
            {
                case 1:
                    printList(root);
                    break;
                default:
                    cout << "Pilihan Tidak Sah. Sila Masukkan Semula\n\n";
                    break;
            }
        }
    }
    while (choice != 2);
}

```

```

    while (choice != 2);
    return;
}

void readFile(nodePtr &root)
{
    int numPhones;
    string firstname, number, hnumber, complaint;
    ifstream infile ("ComplaintDB.txt", ios::in);
    infile >> numPhones;
    infile.ignore(500, '\n');
    for (int count = 0; count < numPhones; count++)
    {
        getline(infile, firstname, '\n');
        getline(infile, number, '\n');
        getline(infile, hnumber, '\n');
        getline(infile, complaint, '\n');

        root = new node (firstname, number, hnumber, complaint, root);
    }
}

```



```

void password(void){
system("cls");
char pass[10],ch,password[10]="1234";
int i=0;
cout<<"\n\t\t=====";
cout<<"\n\t\t|   House Inventory of Residential College Management System   |"<<endl;
cout<<"\t\t=====";
printf("\nMasukkan Kata Laluan:");
//scanf("%s",pass);
while(ch!=13)
{
ch=getch();

if(ch!=13 && ch!=8){
putch('*');
pass[i] = ch;
i++;
}
}
pass[i] = '\0';

if(strcmp(pass,password)==0){
printf("Password Matched!!");
printf("Loading ");
for(i=0;i<6;i++)
{
printf(".");
}
system("cls");
mainmenu();
}
else
printf("\nPassword is invalid!!\a");
printf("\nENTER correct password");
getch();
main();
}
}

```

```
void printList(nodePtr temp)
{
    while (temp != NULL)
    {
        cout << "Nama: " << temp->firstname << "\n";
        cout << "Nombor Matrik: " << temp->number << "\n";
        cout << "Nombor Rumah: " << temp->hnumber << "\n";
        cout << "Aduan anda: " << temp->complaint << "\n";
        cout << "\n";
        temp = temp->next;
    }
    cout << "\n\n";
}

int countNodes(nodePtr temp)
{
    int countB = 0;
    while (temp != NULL)
    {
        countB++;
        temp = temp->next;
    }
    return countB;
}
```


Conclusion

Advantages and Limitation of the project

The advantages of this project is to make registration and complaint from the students more efficient and easy. This system also reduce the use of paper as before these each house will be given a piece of paper for them to register their electronic stuffs. Other than that, this project also secure the information of the students as the privacy of the user will be stored safely. Last but not least, this project improve IT governances because the inventory is managed online and the system of management will smoothly being filled and the data can be retrieve easily.

There are some limitations for this project. Firstly, if this project cannot be done, it can cause inefficient for management of utility form as the staff need to print all the form and distribute it thus it will take so much time. Secondly, loose of data might be happened. If the data is lost, student need to refill the form. Lastly, there might be sometimes that the system will down as the number of students using the system is big. So it will take a bit time to complete the registration or complaint process.








Discussion and Recommendation

There are several aspects that need to take into a look in order to make this project successful. Firstly, we need to observe that this system is really need for user. An observation should be done in any suitable ways. The survey that we took among UPSI students is really helpful as it can give data and we can design our system according to the problem.

Secondly, every team member should know how to use any tools for them to develop the system. It is very important to master the skills on how to use the tools. It is better for one to study and learn from any sources to have the skills. We did learnt how to use software to create data flow diagram, to create coding and also to set up user interface for the system. It is good for us as we already have experience on how to use the tools.

Last but not least, once the system is developed, it should be tested before it is being used by the user. This step is very important because there might be any problem occurred when the system is being used for the first time. Hence, the system should be tested by any members of the team to make sure the system is ready to use.

BIBLIOGRAPHY

-  <https://mymanagementguide.com/feasibility-study-reporting-steps-to-writing-a-feasibility-study-report-fsr/>
-  Computer Hope, *Excel*, April 2020.
-  Design shack, *What is Adobe XD? A 101 Intro*, Compact Creative, 2020.
-  FinancesOnline, *Lucidchart Review*, Grojecka 70/13 Warsaw, 02-359 Poland.
-  Priyali Sharma, *Top 3 Techniques of System Development/MIS*, May 2014.
-  Tutorialspoint, *software case tools overview*, Nov 2019.
-  Wikipedia, *DevC++*.