Report on Project

Submitted to

Hamidu Hasan Sonet

Submitted by

Team: Nonstop coders

Team Member 1:
Mahmudul Hasan Topu
ID: 0112330164

Team Member 2:
Sadika Binti Noor
ID: 0112330453

Section: M

Course: CSE 1112 (Structured Programming Language Laboratory)

Department of Computer Science and Engineering (CSE)

United International University (UIU)

Apartment Management System

Welcome to **Celestial High** apartment complex!
It is a beautiful high-rise building with 100 apartments! Now this extravagant apartment complex needs a well-organized Apartment Management System to store the data of its tenants.

This Apartment Management System needs to be able to let the landlord **ADD** information of the new tenants, **REMOVE** the information of a tenant who moved away, **SEARCH** information of any tenant, **UPDATE** the payment status of a tenant, **SORT** the tenants to see who has been living in the building for the longest time, and **COUNT** how many apartments are empty.

A couple of engineering students (Mahmudul Hasan Topu, Sadika Binti Noor) have built an Apartment Management System as such.
Now let us go through their source code and analyze how the system is built, and how it works:

After adding the essential ***stdio.h*** , ***string.h*** , ***stdlib.h*** functions, first a structure called tenant is created using the ***struct{}*** function. It is used to take data like – Apartment name, Tenant's Name, Tenant's Occupation, Number of family members, Renting Year, Contact Number, Payment status as input from the user and store it in a ***file*** called data.txt.

Then inside the ***main()*** function -

A menu is seen and the user is given 5 options –

a. Option 1 → Add Tenant.
b. Option 2 → Remove Tenant.
c. Option 3 → Tenant's Detail.
d. Option 4 → Update Payment.
e. Option 5 → All Tenant Sorted List.

The User will be given 3 chances to choose and option from the menu. If he fails to input a valid option from the menu 3 times in a row, he will automatically exit from the program.

But, after the user inputs correctly from the choice 1,2,3,4, or 5, the program will do the works accordingly:

Option 1 → Add Tenant:

To add the information of a tenant, the program completes the following steps-

1) First the complier reads all the data from data.txt file by using the ***fopen()*** , ***fclose() ,*** functions and ***"r"*** mode and stores it in an array.
2) Then it asks the user to input an apartment name in which the new tenant is moving in.

3) First the compiler checks If the input apartment name is invalid (if it starts with an integer instead of a letter). It uses ***atoi*** function to ensure the apartment name doesn't start with anything but a letter.
4) If the apartment name is valid, it checks if the apartment is empty or occupied. If the apartment is already occupied, using the ***goto*** and ***jump*** the program asks another input from the user.
5) Then if the input apartment is valid and empty, the user inputs the information of the new tenant (Name, Occupation, Number of family members, Renting Year, Contact Number, Payment status).
6) The data is then updated into the data.txt file by opening the file again and writing the new data into the text file with ***"w"*** mode and then closing the text file.
7) After that, the program counts the apartment names from the updated data file and displays the number of empty apartments in the building.
8) The program then will present 3 options to the user and ask for an integer as input. Then using the goto and jump function, for each option the user will be taken to a different part of the code.
    - ➔ If the input is 1, the program will return to the main menu.
    - ➔ If the input is 2, the user will be able to add another tenant to the data file.
    - ➔ If the input is 3, the program will exit.

Option 2 → Remove Tenant:

To remove the information of a tenant, the program completes the following steps-

1) First the complier again reads all the data from data.txt file by using the ***fopen()*** , ***fclose() ,*** functions and ***"r"*** mode and stores it in an array.
2) Then it asks the user to input an apartment name which is recently been emptied.
3) First the compiler checks If the input apartment name is valid and if it is occupied.
4) If the input apartment is valid and occupied, the program removes all the information of the previous tenant from that apartment, using the code how a value is erased from an array. Thus, the apartment is emptied.
5) The program then opens a new "temp.txt" file and writes the updated information and closes the file.
6) Then the "temp.txt" file is renamed to "data.txt" file, which is the core file of this program, using the ***rename()*** function.
    - ➔ Reason why the updated data is not directly written in the data.txt file is because – it would write the updated data after the existing data. This potential problem is dodged by using a temporary text file.
7) After that, the program counts the apartment names from the updated data file and displays the number of empty apartments in the building.
9) The program then will present 3 options to the user and ask for an integer as input. Then using the goto and jump function, for each option the user will be taken to a different part of the code.
    - ➔ If the input is 1, the program will return to the main menu.

- ➔ If the input is 2, the user will be able to remove another tenant from the data file.
- ➔ If the input is 3, the program will exit.

Option 3 → Tenant's Detail:

To see the information of tenants, the program completes the following steps -

First the program reads all the data from data.txt file by using the ***fopen()*** , ***f(close) ,*** functions and ***"r"*** mode and stores it in an array. Then it asks the user to choose an option between 1 and 2. Option 1 would show the user the information of a single tenant and option 2 would show him the information of all tenants. If the user chooses-

Option-1:

a. The program reads all the data from data.txt file by using the ***fopen()*** , ***f(close) ,*** functions and ***"r"*** mode and stores it in an array.
b. Then the program asks for an apartment name as input from the user, of which apartment the program will display the tenant's information of.
c. After checking if the apartment name is valid and if the apartment is occupied, the programs display all the information of that tenant living in that apartment.
d. The user then will be presented with 2 option, 1 and 2. If the user chooses –
   → Option 1: He will be able to search for the information of another tenant.
   → Option 2: The program will exit.

Option-2:

The program displays all the data of all the tenants and presents the user with 2 options (1 and 2), 1 for returning to main menu and 2 to exit the program.

Option 4 → Update Payment:

To update the payment status of the current month, the program completes the following steps –

1) After reading the data from text file, taking a valid and occupied apartment as input from the user, the program asks an input (Yes/No) from the user.
   Yes → If the payment is done
   No → If the payment is still due.
2) After taking the input from user, the program opens and writes the information in the temp.txt file and closes if.
3) Then it renames the temp.txt file to data.txt file.
4) Then the program presents the user with 3 options (1, 2, 3), 1 for returning to main menu, 2 for updating payment status again, and 3 to exit the program.

Option 5 → All Tenant Sorted List:

To sort the tenant's list in a descending order to see who has been living in the building for the longest time, the program completes the following steps –

1) The program first reads all the data from data.txt file.
2) Then it displays the apartment name and tenant's name one by one till the end of the file.
3) The displayed list is the list that shows who has been living in the apartment for the longest time in descending order.

   What is the logic behind the code?
   → The landlord updates the data file every time a new tenant starts living in an apartment. So naturally, the tenant living in the first apartment would be living in the building for the longest time, the next tenant in the data file would be living there for the second longest period of time and thus it goes on till the last tenant's name who must be the last person have moved in to the building.
4) Then the user is presented with 2 options (1 and 2), 1 for returning to main menu and 2 to exit the program.

In the source code there are many more essential functions used such as –

1) **fflush(stdin):** used to flush output buffers
2) **strcmp():** used to compare 2 strings
3) **strcspn():** used to find the first occurrence of any character from a set in a string, returning the length of the segment before any of those characters.
4) **rename():** used to change the name of a file.

----- Here the source code ends-----

Contribution:

Both members of the team ***Nonstop Coders*** sat together and wrote the source code after proper planning. Although the members wrote different segments of the code.

The segments mentioned below had the most contributions from ***Mahmudul Hasan Topu*** **(ID: 0112330164)**:

1. Add
2. Remove
3. Sort

The segments mentioned below had the most contributions from ***Sadika Binti Noor*** **(ID: 0112330453):**

1. Search
2. Payment Update
3. Count

→ All the segments of the code was merged by ***Mahmudul Hasan Topu*** **(ID: 0112330164).**

→The report explaining the source code was written by ***Sadika Binti Noor*** **(ID: 0112330453).**