Construction and design of dynamic community management system for the elderly

Rui Wang*, *Chenghang* Zhou, and *Zhuofu* Deng

Software College of Northeastern University, 195 Innovation Road, Hunnan District, Shenyang City, Liaoning Province, China

Abstract. In view of that the current information management and service efficiency of most pension institutions is low and can not meet the needs of the elderly for professional care, combined with the convenience of the current application of the Internet, a dynamic community management system for the elderly was designed. This system is designed for managers and members in the community, and it is composed of five modules: membership management, housing management, check-in management, entertainment venue management and user management. In the realization of multiple sub-functions, the combination of file operation and structure array makes the structure clear and complete. At the same time, the realization of sub-functions directly applies the array function, reducing the number of lines of code inside the function, making the function more prominent and easier to identify.

1 Introduction

At present, the problem of aging population in society is becoming increasingly serious. Most pension institutions have low informatization level, lack of modern service equipment, low management, and service efficiency, which cannot meet the needs of the elderly for professional care[1,2]. In order to improve the living standard of the elderly while improving the economic level, reduce the work intensity of nursing staff, and improve the management level of nursing homes, it requires the application of Internet & Pension combined with intelligent hardware and software, namely the information platform.

It will be of great benefit to improve the elderly pension service level using information technology to build the pension service platform and provide information network services. If the development of pension system is in harmony with the construction of smart city, it will also inject new vitality and vitality into the high-quality development of pension service[3,4].

The dynamic community management system for the elderly will bring convenience to community managers and residents through powerful computer technology. It constructs entertainment facilities and community ecology for the elderly in their daily life into the system, so that the elderly can enjoy community life conveniently and the management of the staff is more effective. This reduces the consumption of human resources in community

* Corresponding author: wangr1211@163.com

management and reduces administrative costs while improving the accuracy and reliability of information, improving community management and personnel services, and establishing an efficient information transmission and service platform.

2 Construction of the management system

The dynamic community management system is designed for community managers and the members of the community (elderly or perhaps the children of the elderly, i.e. the customers who actually use the system). The system consists of five main modules: membership management, housing management, check-in management, entertainment venue management and user management. The construction of the system is shown in Figure 1.

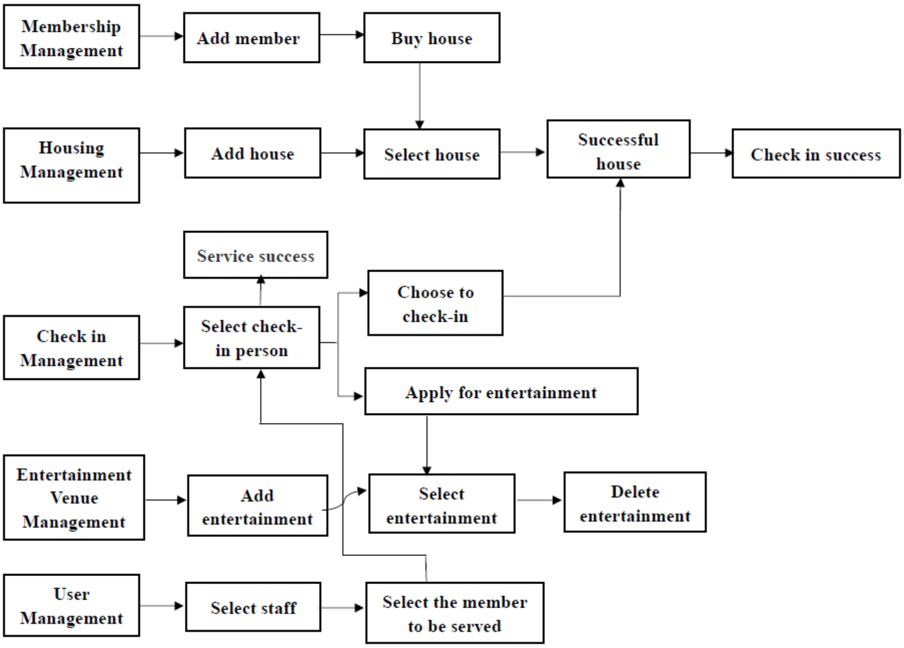


Fig. 1. Construction of the dynamic community management system for the elderly.

The membership management module, which mainly deals with all data and information related to members, consists of several sub-modules: add member, search member, revise member, delete member, buy house and other operations. This module realizes the flexible update and modification of member information. When members purchase houses, the module can also provide housing information query.

The housing management module is targeted at community managers. It mainly deals with the relationship between houses and whether members check in, such as recording the basic situation of the house, purchasing, or using of the houses, which can better facilitate the management of the houses. The housing management module can realize two major functions, that is, community managers can build optional houses, and members can query optional rooms. In the house inquiry, the system sets up several parameters such as house number, house price, house size, house type, and house state to achieve the management of house information.

The check-in management module is the core module of the software system, including four sub-modules of buying house, checking in, renting out and applying for entertainment facilities. These four sub-modules build entertainment facilities and community ecology in the daily life of the elderly into the system.

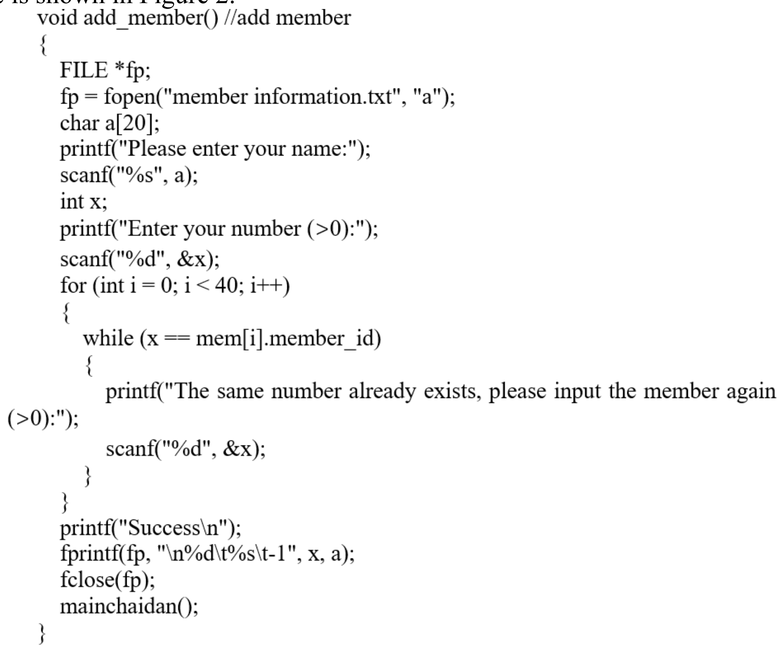
The entertainment venue management module is targeted at community managers, mainly dealing with the renting of entertainment venue facilities in the community, which can make it more efficient for residents to use community venue facilities. Module is divided into three sub-modules including add, select, and delete entertainment venue facilities.

The user management module is targeted at the community staff, which belongs to the most basic part of the system. It is related to all staff in the community and deals with the basic data of all staff in the whole community. The operations include adding, modifying, deleting, and viewing the member to be served. The function of this module makes the management of staff more effective.

3 Implementation and debugging

In the construction of the membership module, a while (1) loop is used to find the ID number of the member. Since we do not know whether the community members have houses or how many houses they have, we use loop statements to output the house information, and use fflush(stdin) to clear the cache.

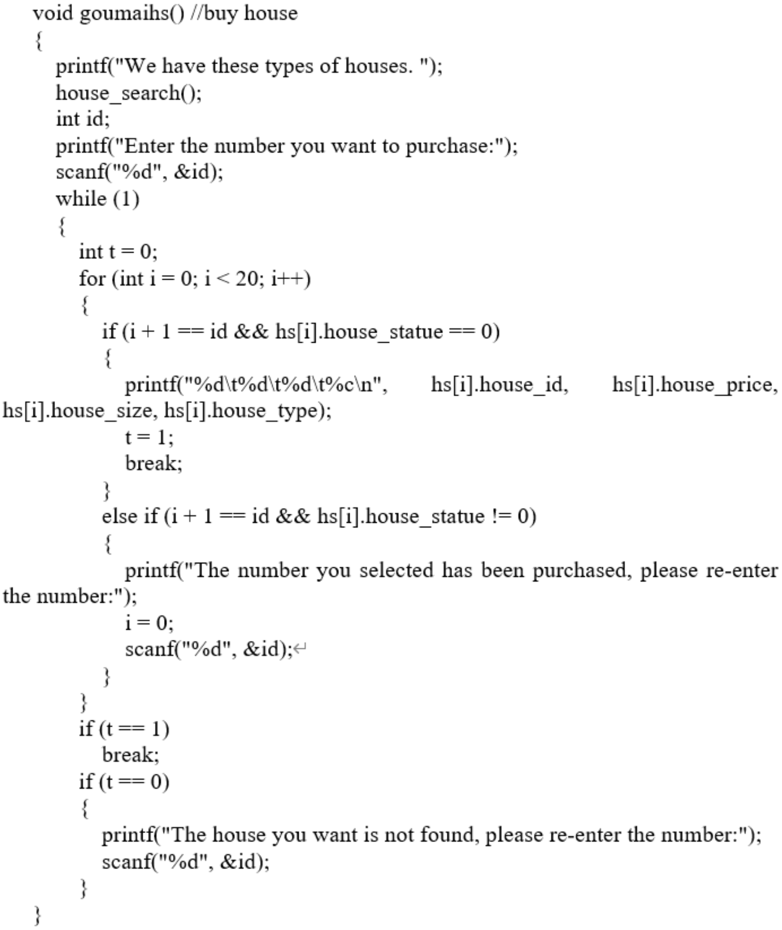
In addition, to ensure that each member's information is unique, a double loop is established in the membership management module to achieve error reporting when members enter repeated information and provide an operation interface for users to re-enter. The C code is shown in Figure 2.



```
    void add_member() //add member
    {
       FILE *fp;
       fp = fopen("member information.txt", "a");
       char a[20];
       printf("Please enter your name:");
       scanf("%s", a);
       int x;
       printf("Enter your number (>0):");
       scanf("%d", &x);
       for (int i = 0; i < 40; i++)
       {
          while (x == mem[i].member_id)
          {
             printf("The same number already exists, please input the member again
(>0):");
             scanf("%d", &x);
          }
       }
       printf("Success\n");
       fprintf(fp, "\n%d\t%s\t-1", x, a);
       fclose(fp);
       mainchaidan();
    }
```

Fig. 2. A double loop in the membership management module.

When a user buys a house, if only purchase data is added to the system, a house may be purchased repeatedly by multiple users. To avoid this case and make the code more realistic, “hs[i].house statue” was introduced in the housing management module, as shown in Figure 3,where “0” means available for purchase, “1” means check-in. With this improvement in the software system, houses can be purchased only when they are available.



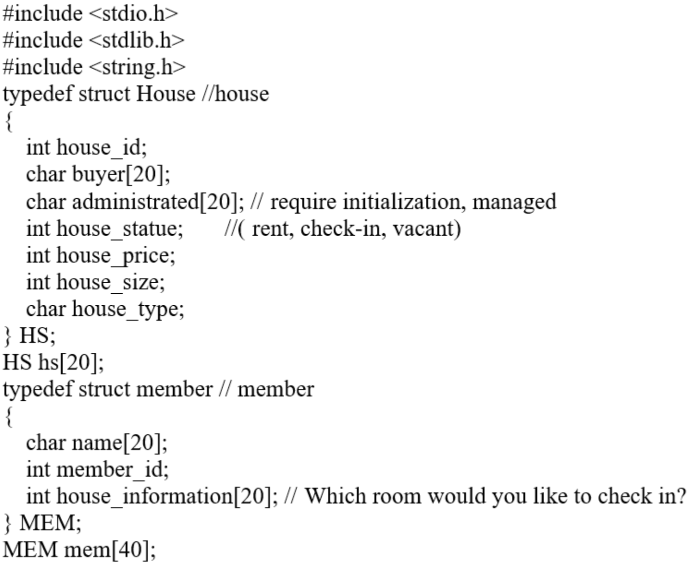
```
  void goumaihs() //buy house
  {
    printf("We have these types of houses. ");
    house_search();
    int id;
    printf("Enter the number you want to purchase:");
    scanf("%d", &id);
    while (1)
    {
      int t = 0;
      for (int i = 0; i < 20; i++)
      {
        if (i + 1 == id && hs[i].house_statue == 0)
        {
          printf("%d\t%d\t%d\t%c\n",       hs[i].house_id,       hs[i].house_price,
hs[i].house_size, hs[i].house_type);
          t = 1;
          break;
        }
        else if (i + 1 == id && hs[i].house_statue != 0)
        {
          printf("The number you selected has been purchased, please re-enter
the number:");
          i = 0;
          scanf("%d", &id);↵
        }
      }
      if (t == 1)
        break;
      if (t == 0)
      {
        printf("The house you want is not found, please re-enter the number:");
        scanf("%d", &id);
      }
  }
```

Fig. 3. Hs[i].house statue applied to the housing management module.

When building sub-functions in each module, it is usually to open and write files repeatedly, which is inconvenient for the simplicity of the program. Therefore, a large number of functions about file reading and output are built, which can effectively reduce the amount of code writing for each module. Structure arrays are also used in the programming of each module of the system instead of linked list structure. By calling the array one by one in the program, the functions of each module of the system can be realized and more prominent to identify. For example, as shown in Figure 4, the following codes contains the array functions in the housing management and member management.

4 Conclusions

The dynamic community management system for the elderly is very important for the construction of pension information platform. The software system serves the elderly and community managers, including five modules of member management, housing management, check-in management, venue facilities management and user management. The realization of the system can make it convenient for the elderly enjoy community life, and managers can manage more effectively. The system pays attention to the combination of file operation and structure array to make the software system complete. In particular, the array function is directly applied in the module construction of membership, entertainment facilities, add member, deletion, modification and so on, which will reduce the number of lines of code inside the function and make the system function more prominent and easier to identify.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct House //house
{
   int house_id;
   char buyer[20];
   char administrated[20]; // require initialization, managed
   int house_statue;       //( rent, check-in, vacant)
   int house_price;
   int house_size;
   char house_type;
} HS;
HS hs[20];
typedef struct member // member
{
   char name[20];
   int member_id;
   int house_information[20]; // Which room would you like to check in?
} MEM;
MEM mem[40];
```

Fig. 4. Array functions in software system.

This work was supported by the Undergraduate Innovation and Entrepreneurship Training Program of Northeastern University.

References

1. L.T. Zhao, J.W. Qian, W. Shan, China-An International J. **19** 112(2021)
2. H.K. Kim, S.H. Lee, J. Econ. Ageing **20**, 100340(2021)
3. T.T. Yu, R.P. Wang, EURASIP J. Wirel. Comm. **2021**,147(2021)
4. F. Arthur-Holmes, W. Agyemang-Duah, J. Gerontol. Soc. Work **63**,699(2020)