ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

The Design and Realization of an Integrated Platform for Intelligent Property

Haishan Wang[1,a,*], Chuansheng Wu[1,b]

[1]*University of Science and Technology Liaoning, Anshan, China*
[a]*2650321241@qq.com,* [b]*1034182681@qq.com*
[*]*Corresponding author*

Abstract: *With the rapid development of information technology and the advancement of intelligentization trend, the integrated platform of intelligent property, as an innovative solution to solve the traditional property management problems, has gradually received widespread attention. This paper firstly explains the concept of intelligent property, and then defines the integration platform. Subsequently, it analyzes in detail the functional modules that the platform may contain, including resource management, income and expenditure management and customer service. Finally, it discusses the technical realization of the platform, aiming to provide a comprehensive and in-depth interpretation and guidance for the construction of the integrated platform of intelligent property.*

Keywords: *property, integration, database, file storage*

1. Introduction

With the acceleration of urbanization and the improvement of people's living standard, property management, as an important part of urban management, is facing unprecedented challenges and opportunities. The traditional property management mode lacks the support of informatization and intelligent means, and it is difficult to adapt to the needs of modern urban development. In this context, intelligent property integration platform has emerged as an important way to solve the traditional property management problems[1].

By integrating modern information technology, Internet of Things (IoT) technology, big data technology and other advanced means, the Smart Property Integration Platform aims to break the limitations of traditional property management and realize the intelligence, efficiency and personalization of property management. The platform not only enhances the efficiency of property management and service quality, but also strengthens the harmony and safety of the community, providing owners with a more convenient, comfortable and safe living environment[2].

In the current property management, there are a lot of problems, such as high cost, single service, insufficient communication, safety hazards and so on. These problems not only affect the effect of property management, but also restrict the harmonious development of the community. Therefore, the construction of intelligent property integration platform is particularly important and urgent. Through the construction of this platform, it can realize the comprehensive management of property resources, accurate control of income and expenditure, and timely response to customer service needs, thus bringing revolutionary changes to property management.

This paper will provide a comprehensive and in-depth interpretation of the intelligent property integration platform, including the concept of intelligent property, the definition of the integration platform, functional modules, and technical implementation. Through the elaboration of this paper, we aim to provide readers with comprehensive knowledge and understanding of the integrated platform of intelligent property, and provide useful reference and guidance for promoting the development and application of intelligent property.

2. The concept of smart property

Intelligent property is a new type of management mode that integrates property management with modern technology. It makes use of information technology, Internet of Things (IoT) technology, big data technology and other means to transform and upgrade the traditional property management in order

ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

to achieve more efficient, intelligent and personalized services. This management mode aims to improve the quality of property services, enhance the harmony of the community, and improve the overall efficiency and level of property management[3-4].

The word "smart" in smart property is mainly reflected in the following aspects.

Here's a refined version of the text you provided:

2.1 Data-driven Decision-making

Intelligent property management offers a scientific foundation for decision-making in property management. By gathering and analyzing vast amounts of data, such as equipment operation records, owners' demands, and energy consumption statistics, it enables more precise and targeted management strategies that better cater to owners' needs.

2.2 Intelligent Services

Leveraging artificial intelligence technology, intelligent property systems deliver automated and intelligent services. For instance, an intelligent access control system facilitates face recognition for entry and exit, an intelligent security system ensures comprehensive safety monitoring, and an intelligent home system enables remote control of household appliances. These services significantly enhance the convenience and comfort of daily life[5].

2.3 Streamlined Communication

Intelligent properties foster instant communication between management and owners, as well as among owners themselves, through an established information platform. This facilitates smoother information exchange and more timely feedback on issues, contributing to a stronger sense of community cohesion and harmony[6].

2.4 Sustainable Development Focus

Intelligent property management also prioritizes green, environmentally friendly, and energy-efficient practices. By implementing intelligent energy management and environmental monitoring systems, it minimizes the environmental impact of property operations, thereby promoting sustainable development in property management. Intelligent property is an innovative property management model, which uses technology as a means to improve the quality and efficiency of services, providing a more convenient, comfortable and safe environment for modern urban life. With the continuous development and progress of technology, intelligent property will play a more important role in the future.

3. Definition of an integrated platform

Integration platform, in the context of intelligent property, refers to an integrated and unified management system, which integrates the originally scattered and independent property management links and functional modules organically through information technology to form an efficient, collaborative and intelligent work platform. This platform not only realizes data sharing and information circulation, but also enhances the overall performance of property management through intelligent processing and analysis[7].

Specifically, the "integration" of the integrated platform is reflected in the following aspects.

3.1 Functional integration

The integrated platform integrates multiple functional modules of property management, such as resource management, income and expenditure management, customer service, etc. These modules are interrelated and work together on the platform, forming a complete and closed-loop property management process. These modules are interrelated and work together on the platform, forming a complete and closed-loop property management process. This integrated management approach avoids information silos and duplication of efforts and improves work efficiency[8].

ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

3.2 Data sharing

The integrated platform realizes centralized data storage and sharing. Each functional module can exchange data in real time to ensure the accuracy and consistency of information. At the same time, the platform also provides data analysis and mining functions to help property managers extract valuable information from massive data to provide support for decision-making.

3.3 Process synergy

The integrated platform through the workflow engine and other technical means, to achieve the automation and synergization of property management processes. Each work link can be seamlessly connected, reducing the possibility of human intervention and error. This process synergy not only improves work efficiency, but also improves service quality[9].

3.4 Intelligent decision-making

The integrated platform utilizes artificial intelligence, big data and other technologies to realize the function of intelligent decision-making. The platform can predict and analyze the indicators of property management according to historical data and real-time data, providing scientific decision-making basis for managers. This intelligent decision-making approach makes property management more accurate and efficient.

Integration platform is the core component of intelligent property, which realizes the intelligence, efficiency and personalization of property management by means of function integration, data sharing, process synergy and intelligent decision-making. With the continuous development and progress of technology, the integrated platform will play a more important role in the field of intelligent property.

4. Functional modules

Functional modules of the intelligent property integration platform are the core components of the whole system, each of which undertakes different management responsibilities and works together to realize intelligent and efficient property management. The following is a detailed description of these functional modules.

4.1 The resource management module

Resource management module is the foundation of intelligent property integration platform, which covers the management of real estate resources, parking space resources, customer resources and special resources. This module through digital means, all kinds of resources for detailed archiving, status tracking and change management. For example, it can record the detailed information of real estate, including area, house type, utilization status, etc., and support quick query and related search. For parking space resources, it can realize the functions of rent and sale control, status statistics and expense management. Customer resource management includes information management of residents, family members, tenants and various cooperative units, which is convenient for property owners to carry out personalized services. Special resource management mainly focuses on non-traditional property resources such as advertisement space and warehouse, and realizes the functions of leasing and cost calculation.

4.2 The income and expenditure management module

Income and Expenditure Management Module is the financial core of Smart Property Integration Platform, which is responsible for charging, expenditure and financial supervision of the property. Through this module, the property can realize the automated processing of regular charges, advance receipts, operating charges, etc., which greatly improves the efficiency and accuracy of financial processing. At the same time, it can also be a variety of income and expenditure items for detailed management, to generate a variety of financial statements, financial decision-making for the property to provide data support. In addition, the module also supports the fee collector's receipt and use of management, as well as fee data audit, to ensure financial transparency and standardization.

ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

4.3 Customer service module

The customer service module is the window of direct interaction between the intelligent property integration platform and owners, which provides a series of service functions, including decoration management, reporting repairs, parking space leasing and complaints and suggestions, etc. Through this module, owners can conveniently submit decoration applications, repair requests and complaints and suggestions, and the property owners can quickly respond to and handle these requests. Through this module, owners can conveniently submit decoration applications, repair requests and complaints, etc., and the property owners can quickly respond to and handle these requests. For example, when the owner submits a repair request, the property owner can quickly assign maintenance personnel through the platform, and track and monitor the maintenance process. This digital management greatly improves the efficiency and satisfaction of customer service.

In addition to the above three main modules, the intelligent property integration platform can also be added according to the actual needs of other functional modules, such as security monitoring, energy management, smart home control, etc. These modules can be seamlessly integrated with the main modules to form a comprehensive, intelligent property management system. These modules can be seamlessly connected with the main module to form a comprehensive and intelligent property management system.

The functional modules of intelligent property integration platform are the key to realize intelligent property management. They manage all kinds of resources, monitor finances, and respond to and handle customer services through digital means, greatly improving the efficiency and satisfaction of property management.

5. Technical realization

When talking about the technical implementation, we need to consider how to implement the functionality of the customer service module described above. Below I will describe in detail how to implement the functions of adding a customer, deleting a customer, updating a customer's information, searching for a customer and printing a list of customers.

First, we need to define the 'Customer' structure and related functions, and ensure that they are declared and defined in the same file or through an appropriate header file.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// customer structure definition
typedef struct Customer {
    int id;
    char name[100];
    char email[100];
    char phone[20];
} Customer;

// declare the function prototype
void addCustomer(Customer *customers, int *customerCount, int id, const char *name, const char *email, const char *phone);
void removeCustomer(Customer *customers, int *customerCount, int id);
void updateCustomerInfo(Customer *customers, int *customerCount, int id, const char *name, const char *email, const char *phone);
Customer* findCustomer(Customer *customers, int customerCount, int id);
void printCustomers(const Customer *customers, int customerCount);

// ...  Other codes (such as main function)

// add clients
void addCustomer(Customer *customers, int *customerCount, int id, const char *name, const char *email, const char *phone) {
```

ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

```
    if (*customerCount >= 100) {
        fprintf(stderr, "Error: Customer list is full. Cannot add more customers.\n");
        return;
    }

    Customer newCustomer;
    newCustomer.id = id;
    strncpy(newCustomer.name, name, sizeof(newCustomer.name) - 1);
    newCustomer.name[sizeof(newCustomer.name) - 1] = '\0';
    strncpy(newCustomer.email, email, sizeof(newCustomer.email) - 1);
    newCustomer.email[sizeof(newCustomer.email) - 1] = '\0';
    strncpy(newCustomer.phone, phone, sizeof(newCustomer.phone) - 1);
    newCustomer.phone[sizeof(newCustomer.phone) - 1] = '\0';

    customers[(*customerCount)++] = newCustomer;
}

// Delete the client
void removeCustomer(Customer *customers, int *customerCount, int id) {
    int i, index = -1;
    for (i = 0; i < *customerCount; i++) {
        if (customers[i].id == id) {
            index = i;
            break;
        }
    }

    if (index == -1) {
        fprintf(stderr, "Error: Customer with ID %d not found.\n", id);
        return;
    }

    for (; index < *customerCount - 1; index++) {
        customers[index] = customers[index + 1];
    }

    (*customerCount)--;
}

// updating customer information
void updateCustomerInfo(Customer *customers, int *customerCount, int id, const char *name, const
char *email, const char *phone) {
    for (int i = 0; i < *customerCount; i++) {
        if (customers[i].id == id) {
            if (name != NULL) {
                strncpy(customers[i].name, name, sizeof(customers[i].name) - 1);
                customers[i].name[sizeof(customers[i].name) - 1] = '\0';
            }
            if (email != NULL) {
                strncpy(customers[i].email, email, sizeof(customers[i].email) - 1);
                customers[i].email[sizeof(customers[i].email) - 1] = '\0';
            }
            if (phone != NULL) {
                strncpy(customers[i].phone, phone, sizeof(customers[i].phone) - 1);
                customers[i].phone[sizeof(customers[i].phone) - 1] = '\0';
            }
            return;
        }
    }
```

ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

```
        fprintf(stderr, "Error: Customer with ID %d not found.\n", id);
}

// find customers
Customer* findCustomer(Customer *customers, int customerCount, int id) {
        for (int i = 0; i < customerCount; i++) {
                if (customers[i].id == id) {
                        return &customers[i];
                }
        }
        return NULL;
}

// print the list of clients
void printCustomers(const Customer *customers, int customerCount) {
        for (int i = 0; i < customerCount; i++) {
                printf("ID: %d\nName: %s\nEmail: %s\nPhone: %s\n\n",
                        customers[i].id,
                        customers[i].name,
                        customers[i].email,
                        customers[i].phone);
        }
}
```

Please note that the technical implementation is simplified and does not include all error handling, dynamic memory allocation, or persistent storage. In practice, you may want to consider the following.

Here's a refined version of the text you provided:

Utilize dynamic arrays, such as those managed with 'malloc' and 'realloc', to accommodate a varying number of customers.

If using C++, incorporate constructors and destructors into the 'Customer' structure to properly initialize memory and release resources.

Implement robust error handling mechanisms, including handling memory allocation failures, to enhance the program's stability.

When dealing with concurrent access, ensure data consistency by implementing appropriate locking mechanisms.

Preserve customer data persistently in databases or files to prevent data loss when restarting the program.

In the code mentioned, the 'addCustomer' function checks if the array has reached its capacity before appending a new customer to the end. The 'removeCustomer' function searches the array to locate and delete specific customers, subsequently rearranging the array elements to eliminate any gaps. The 'updateCustomerInfo' function finds the intended customer and updates their information accordingly. The 'findCustomer' function retrieves the pointer to a customer based on their unique ID. Lastly, the 'printCustomers' function iterates through the array, printing the details of each customer.Please note that the 'removeCustomer' and 'updateCustomerInfo' functions do not check the null pointer of the input parameter, which is very important in practical applications. In addition, when a customer is deleted, the above implementation will move all subsequent elements in the array to fill the blank, which may not be optimal in performance. In more complex applications, you may consider using linked lists, binary search trees, or other data structures to optimize these operations.

6. Conclusion

In this project, we have successfully implemented a basic customer service module, which provides the functions of adding, deleting, updating, searching and printing customer lists. By defining the 'Customer' structure and related operation functions, we can effectively manage customer data.

However, this implementation is simplified and has some limitations. For example, it uses static

ISSN 2522-3488 Vol. 8, Issue 2: 51-57, DOI: 10.25236/IJNDES.2024.080208

arrays to store customer data, which limits the number of customers that can be managed. In practice, we may need to use dynamic data structures, such as linked lists or dynamic arrays, to support a larger number of customers and allow customers to be added and removed dynamically at runtime.

In addition, current implementations do not consider concurrent access and thread safety. In a multi-threaded environment, we need to implement appropriate synchronization mechanisms, such as mutual exclusion locks or read/write locks, to ensure proper access to shared client data.

Finally, the current implementation does not persist customer data to a database or file. In practice, we may need to save customer data to persistent storage so that the data can be restored after a program restart.

In summary, although we have implemented a basic customer service module, there are some key issues and challenges that we need to consider and address before applying it to a real production environment. By improving the data structure, adding concurrent access control and implementing persistent data storage, we can build a more robust, scalable and reliable customer service system.

Acknowledgements

Liaoning University of Science and Technology Undergraduate Innovation Training Project (X202310146212)

References

[1] Wang Zhaojin. Integrated platform to build a smart city [J]. Shanghai Informatization, 2012.
[2] Unlisted author. Integrated platform changes "information island" - Shanxi road and bridge mode of digital integrated platform construction of large state-owned infrastructure enterprises [J]. China Highway, 2022.
[3] Zhou Hong. Optimization exploration of Technical Specification for Ultra-short Wave Monitoring and Management Integrated Platform [J]. Shanghai Informatization, 2023.
[4] Zhong Zhihong, Wu Yanming. Exploration and Practice of Digital Rural Integration Platform Construction - Taking Jiangxi Province as an example [J]. Agricultural Engineering Technology, 2023.
[5] Wang Kun, Li Jiwei. Beijing Tianjin Hebei Establish an Integrated Platform for Consumer Rights Protection [J]. Guangxi Quality Supervision Guide, 2017.
[6] Ding Wenjun. Innovation leads to building an industry-leading intelligent construction integration platform [J]. Construction Enterprise Management, 2023.
[7] Yang Junyi. Bond quantification practice based on investment research integration platform [J]. Bonds, 2021.
[8] Jin Hui, Zhou Hong, Guo Qing, Chen Lin. Necessity analysis and verification method analysis of large-scale information system pilot verification - take tobacco integration platform as an example [J]. Economist, 2023.
[9] Gao Ning. Research and practice of provincial data center operation and maintenance management integration platform in the context of three centralization [J]. Financial Technology Era, 2022.