

The Outline Design of Students' Schedule Management System Based on B / S Mode

Lan Mi¹

¹*Environmental Management College of China, No. 8 Golden Harbor Road, Beidaihe District, Qinhuangdao City, Hebei province, China*

Abstract: This study is based on a preliminary outline design involved in the development of B / S mode of student schedule management system. It can provide the foundation and basis for the Panel's later detailed design and coding, and also standardize the team development. Documents will be divided into various modules, and the functional requirements of corresponding module are raised; total class diagrams, class function definitions and the function of functions are obtained for reference of the later design

1 Purpose of compiling

This document is a comprehensive and systematic presentation of the students' agenda management software architecture, and it describes the main aspects of the student program management software from different angles by using a variety of views, in order to meet the different focus and demand of relevant stakeholders of the software of the system's. This article expresses the important decision-making software architecture design staff has made the software architecture. Software Group will develop structural frame member as defined in accordance with project development plans; accordingly programmer will take detailed design of each member; test designers will test framework in accordance with the general design of the system architecture; another architecture document is also used to guide the member implementation, integration and testing.

2 System Structure Analysis

2.1 The basic structure

The system will eventually face student users, who is not familiar with knowledge of related fields, but completely familiar with the operation of the computer; on the other hand, the difference between each user's computer configuration is also larger, so the system uses B / S structure .

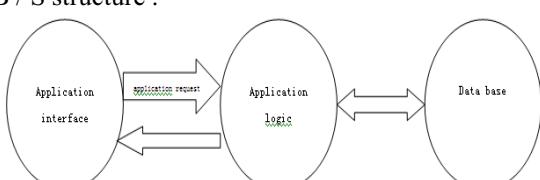


Figure 1 the three-layer B/Sstructure

2.2 Module and System Architecture

Based on the user needs analysis, it designs login module, UI module, the weather module, time module, log module, course module, etc. Thus the system structure can be designed.

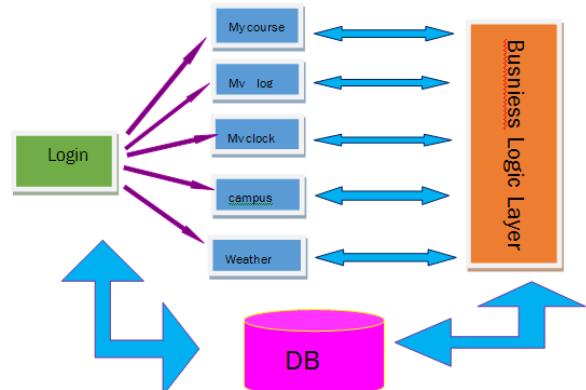


Figure 2 System Structure

3 Module Design

3.1 UI module Overview

The interface module is responsible for the preparation of the entire process, and it provides interactive interface. The general category of UI module is as follows:

^a Corresponding author: author@e-mail.org

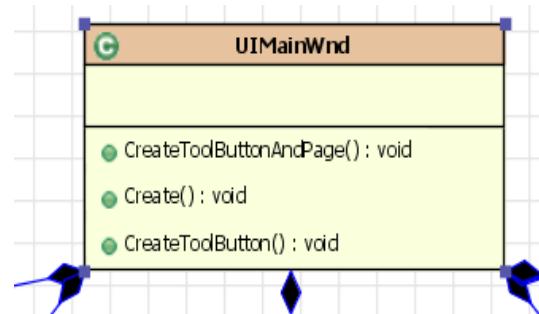


Figure 3 The general category of UI module

The definition of category is as figure 1

Number	Function	Detailed Contents of Function
1	Create()	Create and display windows and buttons
2	CreateToolButton()	Create a tool button on the main interface
3	OnEventBtnClick_C hangeSkin()	Change the skin

3.2 Login Module Overview

This module is the first event entering into the system. The module is divided into four events: 1, user login: users who have been properly registered can log in to this system; 2, user registration: users who haven't registered yet get an account; 3, users logs out: users exit the system; 4, users' offline use: allowing users to use the system without having to log in, but the user can not manipulate the data.

Functional point design of related classes:

Checkin, users log in through comparison between the information recorded and data. If the user's information exists, he can log in, or the account or password will be warned wrong.

SetUserAccount(), set current user account.

CompareSsid(), compare users account, if it appears FALSE, it points the wrong input.

CompareKey(), compare user passwords, if it returns FALSE, it means the user input is incorrect

IsUserExist(), User's existence is checked through a user account id (primary key in the database).

Checkout, User logs off. In the main interface, when the user wants to switch accounts he can use this function. The user logs out the current account and the interface will go to the login screen.

ExitCurrentUI(), end the current interface.

GetLoginUI(), jump login interface.

Register, User registration. Users can enter their registration information, including the account password. This account password will be entered into the database.

IsUserExist(), Check if the user exists, if it does not exist, it will return TRUE, if it exists, it will return FALSE.

SetUserAccount(), Enter the user's account into the database.

UseByOffline(), It enables users to work offline (ie, Registration is not required).
GetMainUI(), Jump to the main interface.
ExitCurrentUI(), exit.

3.3 Weather Module Design Overview

This module provides weather information to the user, the user can use this module to understand weather information on the current location, or to obtain weather information of other location by typing a different location (city). The total class diagram is shown in Figure 4:

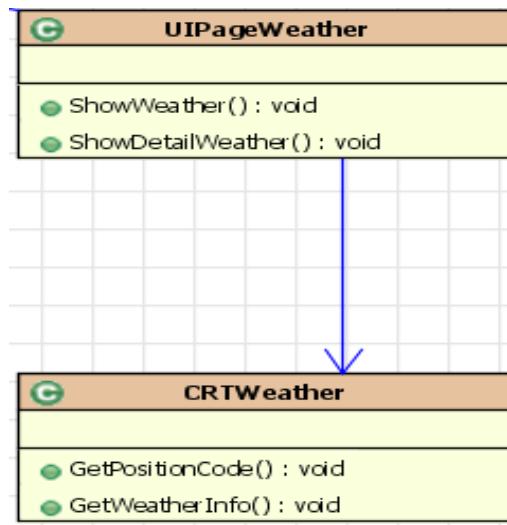


Figure 4 Weather module type diagram

3.4 Time Module Design Overview

This module displays the current time, a calendar function and date information. Total class diagram is shown in Figure 5:

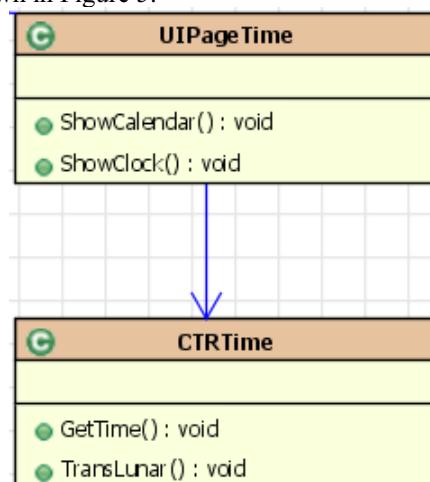


Figure 5 Time module type diagram

3.5 Log module Design Overview

The module provides log function for users. Users can make a record for a number of things worth recording during the day and store them in a database.

Users can add logs, change logs, delete the logs, view the log list, and set the log reminder.

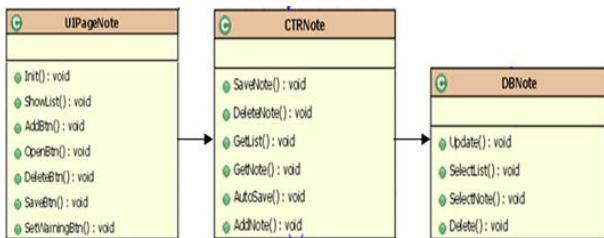


Figure 6 Log module type diagram

3.6 Course Module Overview

Course module provides users a maintainable curriculum. Users can fill in their own course information on the inside, and can choose to display real-time information about the current course of time on the desktop. By reading and writing database DBSchedule, one semester course information is stored and can be read according to user requirements. Its function processing logic is shown in Figure 7.

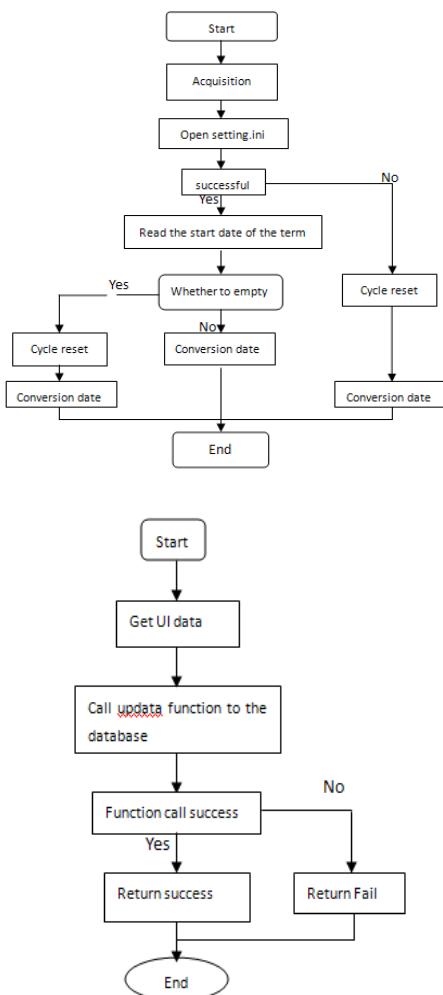


Figure 7 function logic processing diagram

4 Conclusion

This project is mainly designed for college students in general, with a higher use frequency. Besides meeting the needs of key software features, it also asks to consider security architecture design. There should be a more reliable security performance and fast response interactive features. In short, it requires stability, safety, convenience, and ease to operate and manage.

References

1. Barabba, V.P. (1991). "Through a Glass Less Darkly," Journal of the American Statistical Association, Vol. 86, No. 413, pp. 1-8.
2. Harsh, Stephen B., L. J. Connor, and G. D. Schwab. (1981). Managing The Farm Business. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
3. House, William C. (1983). Decision Support Systems – A Data-Based, Model-Oriented UserDeveloped Discipline. Petrocelli Books, Inc. New York, NY.
4. Paul T. Jaeger, Kim M.Thompson, E-government around the world: Lessons, challenges, and future directions, Government Information Quarterly, Vol. 20, 2003, pp.389-394.
5. B. Azad and S. Faraj, E-Government institutionalizing practices of a land registration mapping system, Government Information Quarterly, vol. 26, pp. 5-14, 2009.