

Student Sports Performance Management System

Software Architecture Analysis

Version 1.0

2017.12.19

蔡仲谋
2015212073

Introduction to Software Engineering
Fall 2017

Revision History

日期	版本描述	作者	说明
2017.12.19	Version 1.0	蔡仲谋	first wrote the architecture analysis

Document Approval

The following Software architecture analysis has been accepted and approved by the following:

签名	姓名	标题	日期
蔡仲谋	蔡仲谋	student sports performance management system	2017.12.19

student sports performance management system

Table of Contents

REVISION HISTORY	2
DOCUMENT APPROVAL	2
1. INTRODUCTION	1
1.1 OVERVIEW	1
2. ARCHITECTURE REQUIREMENTS	1
2.1 SIGNIFICANT TARGET	1
2.2 ARCHITECTURE USE CASES	2
2.2.1 User Role	2
2.2.2 analysis of use case diagram	3
2.2.3 CLASSES / OBJECTS	3
2.2.3.1 teacher	3
2.2.3.2 administrator	3
2.2.3.3 student	3
2.3 BASIC CONSTRAINTS	3
2.4 ASSUMPTIONS AND DEPENDENCIES	4
2.5 ARCHITECTURE REQUIREMENTS OF DIFFERENT SIDES	4
2.6 ENGINEERING REQUIREMENTS	5
2.6.1 Hardware Requirements	5
2.6.2 Architecture Requirements	5
2.6.3 Business Constraints	5
2.7 NON-FUNCTIONAL REQUIREMENTS	5
2.7.1 Performance	5
2.7.2 Reliability	6
2.7.3 Availability	6
2.7.4 Security	6
2.7.5 Maintainability	6
2.7.6 Portability	6
2.8 RISKS	7
2.8.1 Demand Risks	7
2.8.2 Organization and Management Risk	7
2.8.3 Personnel Risks	7
2.8.4 Developing Environment Risks	8
2.8.5 Customer Risks	8
2.8.6 Product Risks	9
2.8.7 Design and Implementation Risks	9
2.8.8 Process Risks	9
3. SOLUTIONS	10
3.1 RELEVANT ARCHITECTURE MODE	10
3.1.1 Conceptual level architecture	11
3.1.2 Module level architecture	12
3.1.3 Running level architecture	14
3.2 ARCHITECTURE OVERVIEW	14
3.3 Architecture View	15

3.4 BEHAVIOR VIEW	16
3.4.1 login	16
3.4.2 change password	16
3.4.3 query of grade	16
3.4.4 authorization of grade	17
3.4.5 entering of grade	17
3.4.6 management of personal information	17
3.4.7 retrieve of data	18
3.5 Implementation problem	18
4. SYSTEM QUALITY ANALYSIS AND JUDGEMENT	19
4.1 CONVERSATION ANALYSIS	19
4.2 PROTOTYPE ANALYSIS	19
4.3 RISKS	20
4.3.1 Demand Risks	20
4.3.2 Organization and Management Risk	20
4.3.3 Personnel Risks	20
4.3.4 Developing Environment Risks	21
4.3.5 Customer Risks	21
4.3.6 Product Risks	22
4.3.7 Design and Implementation Risks	22
4.3.8 Process Risks	23
A. APPENDICES	23

1. Introduction

1.1 Overview

The purpose of this system is to facilitate students to conduct sports scores inquiry, teachers to make sports exam rules, and the input as well as analysis of test scores, at the same time facilitate the archives of the administrator to score, improve everyone's efficiency. This document made participated personnel able to generally understand the composition as well as architecture of the system through the functional analysis of the system.

2. Architecture Requirements

2.1 significant target

1. login function,

Any user needs to enter a user name and password, and log in to the corresponding permission interface when matching. If no match is found, an error is reported and abnormal information is recorded to ensure security.

2. Job placement and acceptance

Teachers can post custom rules for assignments and grading, and students can submit information in the appropriate format, scoring according to criteria or manually by the teacher.

3. Grade entry and revision

The teacher can manually enter the grades of a subject according to the format, and the academic staff can change the grades.

4.grade inquiry

Student login system can check their own grades

5.archive

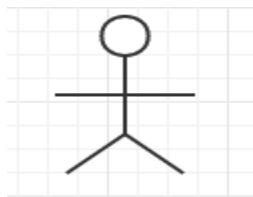
After the examiner has verified the result, the result can be filed and saved permanently.

6.information change

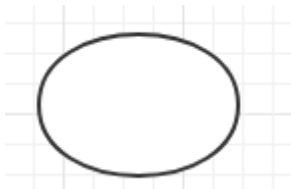
Administrator can modify the student information, students can change the password.

2.2 Architecture Use Cases

2.2.1 User Role



role



function

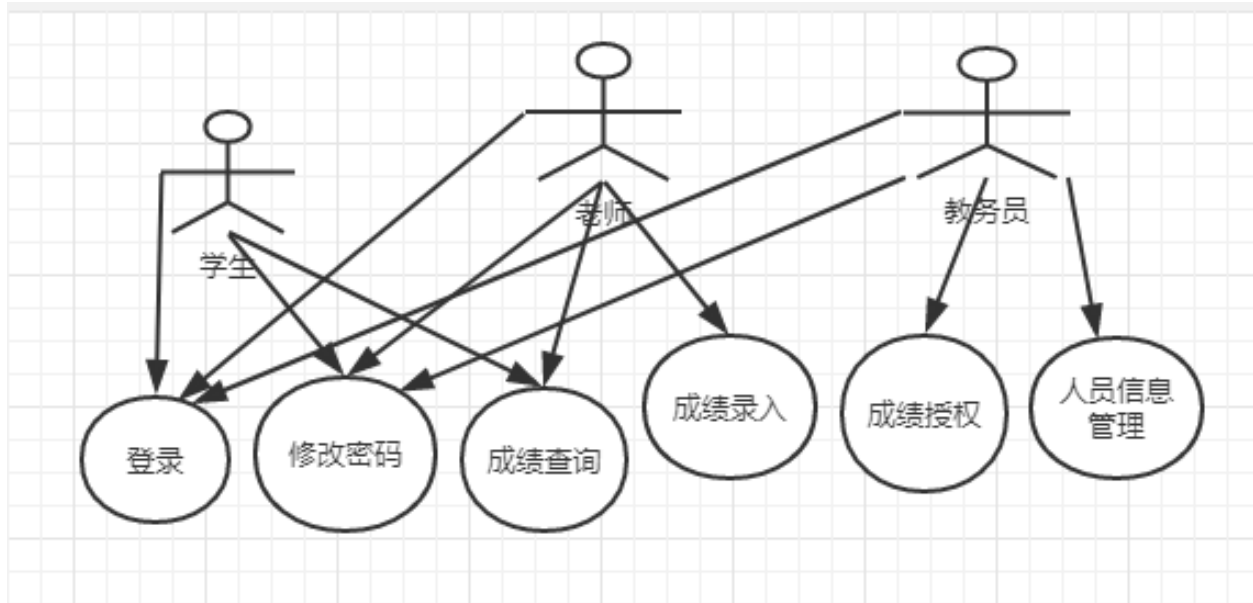


data flow



external data flow

2.2.2 analysis of use case diagram



2.2.3 Classes / Objects

2.2.3.1 teacher

login, change password, query and entry of grades

2.2.3.2 administrator

login, change password, authorization of grades, management of personal information

2.2.3.3 student

login, change password, query grades

2.3 Basic Constraints

Basic Constraints for the Project: The school that uses this project has 20,000 students, 500 physical education

student sports performance management system

teachers and 20 Administrators. The system requires all students to record daily exercise.

2.4 Assumptions and Dependencies

1. Assuming that this system user base is 20,000 students, 500 teachers and 20 academicians
2. Assuming the system's performance data not only to be stored in the system database but also for local media archiving.
3. Assuming the teacher entered the exam, the system can measure the results.

2.5 Architecture Requirements of different sides

读者	目的
PROJECT MANAGER	Based on this document to get an overview of the entire project, assess the project duration, cost, etc. so that the progress of the project can be reasonably arranged and managed
DESINER	Based on this document to project modeling and architecture design, database design, etc., to develop an iterative plan.
DEVELOPER	It is also important to understand the project as a whole when developing the parts that you are responsible for. At the same time, this document is also a reference for developers in developing their own responsible parts and as a reference for considering whether the required functionality has been achieved.
TESTER	The product is tested against the description in this document to ensure that the function of the final product matches the description in this demand document.

2.6 Engineering Requirements

2.6.1 Hardware Requirements

- Window series operating system,
- 512M memory
- Intel Core i5 or higher cpu
- 1T or larger hard-disk

2.6.2 Architecture Requirements

using MVC three layer architecture:

- M represents the core of the system
- V represents the system's representation
- C represents the input/output control

2.6.3 Business Constraints

This project should serve as a plug-in for the existing educational system in the school to facilitate the unified management of the school.

2.7 Non-Functional Requirements

2.7.1 Performance

- 1) The system should be able to support 90% of users to do smooth query operation at the same time .
- 2) The system should be able to support 50% of teachers to do smooth bulk physical record entry physical education operation at the same time .
- 3) the responding time of the system can not exceed 1.5s under 90% circumstances.
- 4) The system's database capacity is to meet the needs of all functional modules, while storing more than six months of daily information and more than ten years of file information

2.7.2 Reliability

1. The system should support multiple live disaster recovery, to avoid unexpected circumstances leading to the service to stop running. On this basis, try to avoid the shortcomings that may lead to service collapse.
2. Due to the large number of user groups, there is a need to consider all aspects of user error input.
3. The system should have a self-test function, being able to monitor the operation of the various functional modules, the system found its own problems.

2.7.3 Availability

1. Simple user interface, prompt information easy to understand, operating functions in the system-wide consistency.
2. Enough guide information, to ensure the smooth use of the user
- 3 Set shortcut keys, user-friendly quick switch operation.

2.7.4 Security

1. Strict control of authority to prevent the disclosure of information.
2. Encrypt sensitive information multiple times.

2.7.5 Maintainability

Code should follow the principle of object-oriented design, designed to be scalable, easy to maintain, in order to reduce the cost of maintenance and secondary development in the future, to ensure the quality of the code.

2.7.6 Portability

Requires that the system works well on most operating systems

2.8 Risks

2.8.1 Demand Risks

- ① poor demand definition, and further definition will expand the scope of the project;
- ② product-defined part of the ambiguity than expected to require more time;
- ③ In the demand do not participate enough customers;

2.8.2 Organization and Management Risk

- ① by the management or market personnel to make decisions, the plan is slow, longer schedule;
- ② project team structure low productivity
- ③ management review and decision-making cycle longer than expected;
- ④ budget cuts, disrupt the project plan;
- ⑤ management made a decision to combat the enthusiasm of the project organization;
- ⑥ lack of necessary norms, leading to errors and duplication of work;

2.8.3 Personnel Risks

- ① The prerequisites (such as training and other items) can not be completed on time;
- ② poor relations between developers and management, leading to slow decision-making, affecting the overall situation;
- ③ lack of incentives, morale is low, reducing the production capacity;

Some people need more time to adapt software tools and environments that are not yet familiar

- ⑤ After the project is joined by new developers, it needs to be trained and gradually communicate with existing members so as to reduce the working efficiency of existing members.
- ⑥ did not find the project urgently needed people with specific skills.

2.8.4 Developing Environment Risks

- ① new development tools longer than expected learning period, a wide range of content.
- ② development tools not in place in time;
- ③ development tools are not as effective as expected, developers need time to create a working environment or switch to new tools;

2.8.5 Customer Risks

- ① customers are not satisfied with the final delivery of the product, requiring redesign and redo;
- ② the customer's opinion has not been adopted, resulting in the final product can not meet user requirements, which must be redone;
- ③ customer planning, prototype and specifications of the audit decision-making cycle longer than expected;
- ④ Customer does not or can not participate in the planning, prototype and specification stage of the audit, resulting in demand instability and changes in product life cycle;
- ⑤ The time of customer's response (such as answering or clarifying the time of the question related to the demand) is longer than expected;
- ⑥ The quality of customer-supplied components is poor, resulting in additional testing, design and integration efforts, and additional customer relationship management.

2.8.6 Product Risks

Correcting poor quality unacceptable products requires more testing, design and implementation than expected;

- ② the development of additional functions do not need (gold-plated), to extend the schedule;
- ③ strict requirements compatible with the existing system, the need for more than expected test, design and implementation work;
- (4) It is required to connect with other systems or systems not controlled by the project team, leading to unpredictable design, implementation and testing;
- ⑤ unforeseen problems arising from the operation in unfamiliar or untested software and hardware environments;
- ⑥ developing a brand new module will take longer than expected;
- ⑦ relying on the technology under development will extend the schedule.

2.8.7 Design and Implementation Risks

- ① design quality is low, leading to repeated design;
- ② some of the necessary features can not be achieved using the existing code and libraries, developers must use the new library or develop new features;
The low quality of the code and library, resulting in the need for additional testing, error correction, or re-production;
- ④ overestimated the enhanced tools for the planned savings;
- ⑤ respectively developed modules can not be effectively integrated, you need to re-design or production.

2.8.8 Process Risks

- ① a lot of paper work makes process slower than expected;

- ② pre-quality assurance behavior is not true, leading to duplication of work later;
- ③ too informal (lack of compliance with software development strategies and standards), resulting in poor communication, poor quality, and even the need to re-develop; Too formal (doctrine adhere to software development strategies and standards), resulting in excessive useless jobs;

3. Solutions

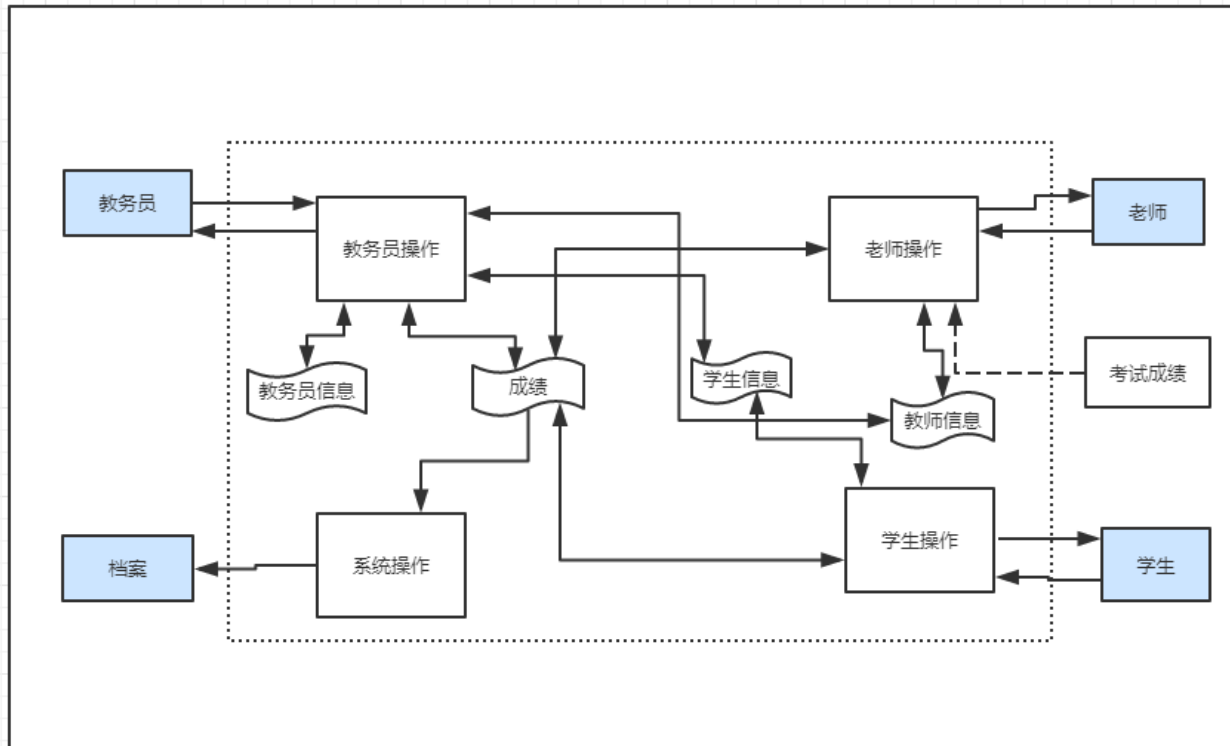
3.1 Relevant architecture mode

The system plans to use shared warehouse mode, the following reasons:

1. The system is a plug-in system of Academic Affairs Office, the Office of Academic Affairs system as a data warehouse, can be very good and Academic Affairs Department System
2. Reusability, as long as the interface to provide each other, this plug-in can work with different mother system
3. Simplify the design, the original system is data-centric, this model is in line with this system, it can be used directly, almost without design.

The following shows the architecture at each level:

3.1.1 Conceptual level architecture



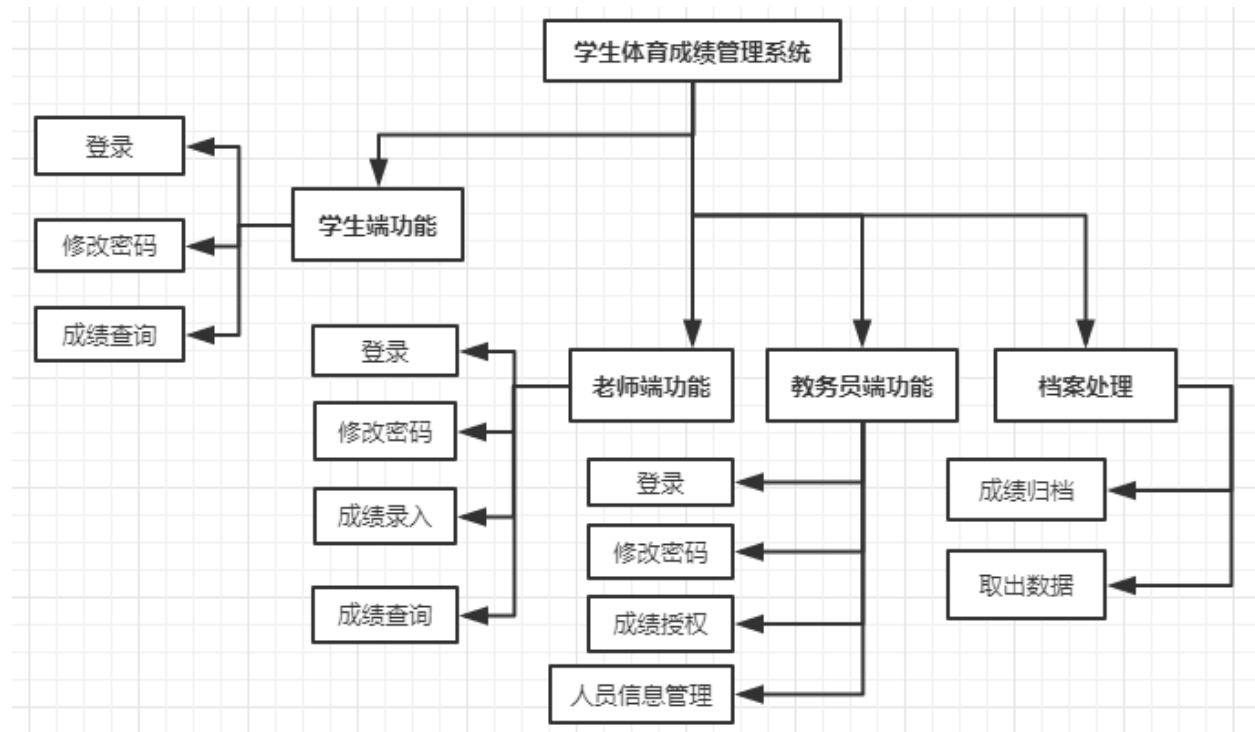
The system is divided into four parts, namely, the administrator side, teacher side, student side and file side, test scores as external data input.

1. The administrator end is operated by the administrator, and the direction of data flow is that the sachet operates the scholar information, the student information, the teacher information and the test scores. That is, the data flow can be inquired or modified, so the direction of the data flow is bidirectional.
2. The teacher end by the physical education teacher operation, the data flow direction is the teacher operation teacher information and the score, because both can inquire also may revise, therefore the data flow direction also is bidirectional

3. Student end operation by the students, the data flow direction is the students operating the students information and results, the results must submit both test and query results, so the data flow is bidirectional.

4. The case ends by the system operation, mainly for archive management, the direction of the data flow is the result injected into the file side, so the data flow direction is one-way.

3.1.2 Module level architecture



functional module analysis:

1. administrator end functional module:

A. login

administrator login and record information

B. password change

change and update the password

C. grade authentication

Authenticated unauthenticated, then authorized to enter the database archive.

D. doing addition, deletion, update, query on the personnel information, including teachers and students

2. teacher end functional module

A. login

teacher login and record information

B. password change

change and update the password

C. grade entry

Record and modify the grades of the published exams while generating unidentified grades.

D. grade inquiry

Query the final grade, check the statistics of the class the teacher is teaching.

3. student functional module

A. login

student login and record information

B. change password

change and update the password

C. grade inquiry

query all the information of the classes the student is attending, including the grades

4. case end functional module

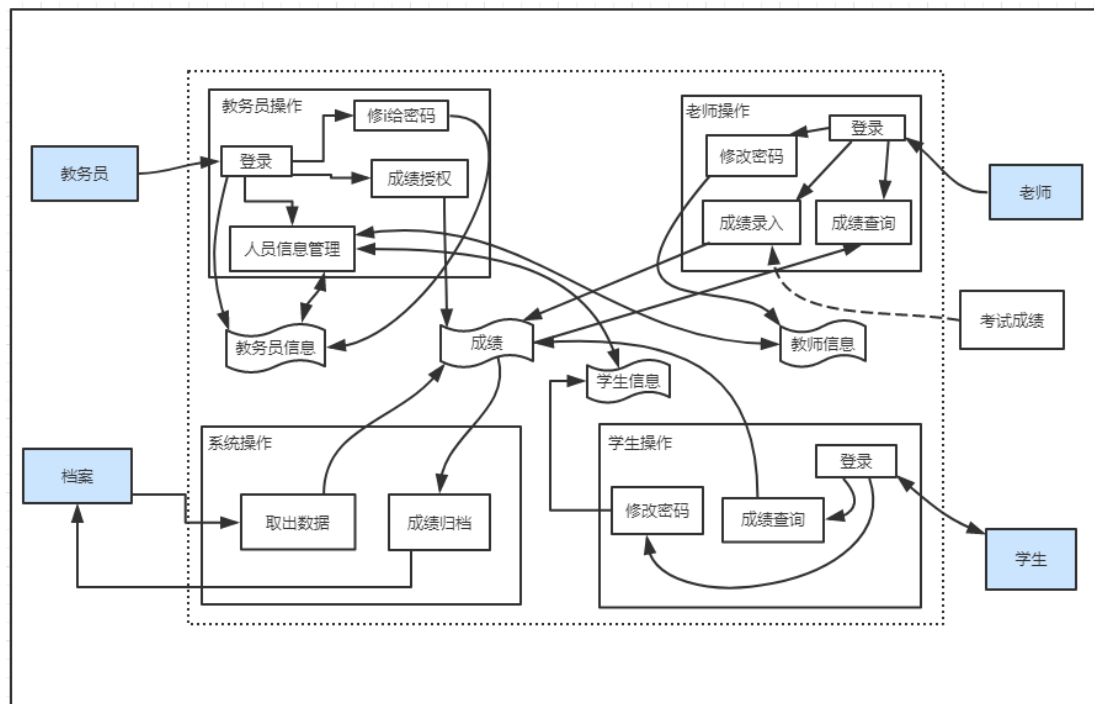
A. get the data

check the data inside the database when necessary

B. grade archiving

the grades are authorized into the file database, for permanent preservation.

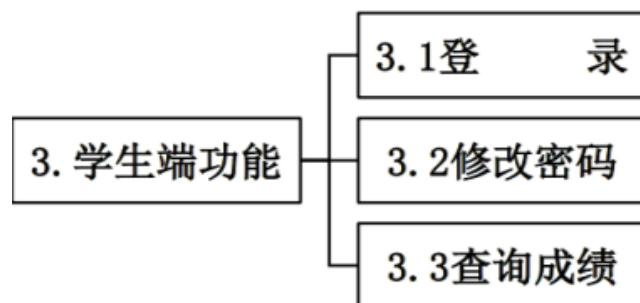
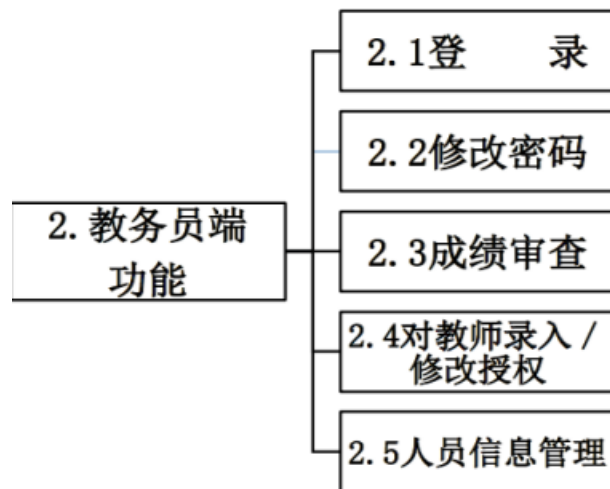
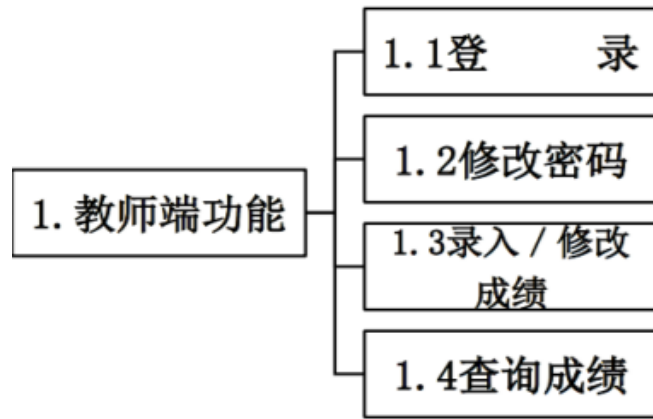
3.1.3 Running level architecture

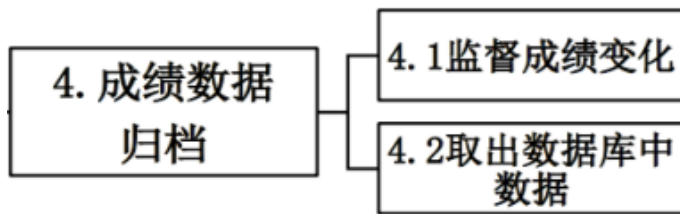


3.2 Architecture Overview

The system uses a shared warehouse architecture, making the database of the Academic Affairs Department as the central data warehouse, the other part as the independent components, making the data of the data warehouse shared under authorization. All components communicate and display data through the data warehouse.

3.3 Architecture View





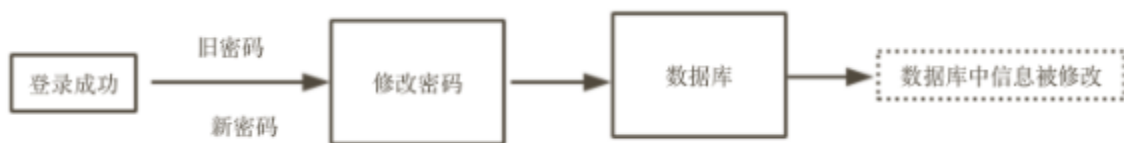
3.4 Behavior View

3.4.1 login



Users enter their own credentials and password, the system first determine the user does not exist and does not belong to the user group, and then determine the correct password,

3.4.2 change password



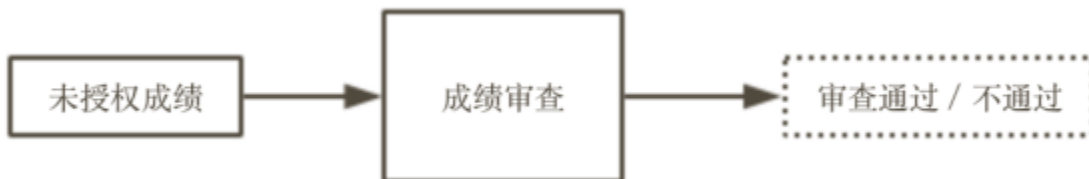
In the login state, it is allowed to enter the old password and the new password to modify the password.

3.4.3 query of grade



In the login state, it is allowed to query the data using corresponding authority.

3.4.4 authorization of grade



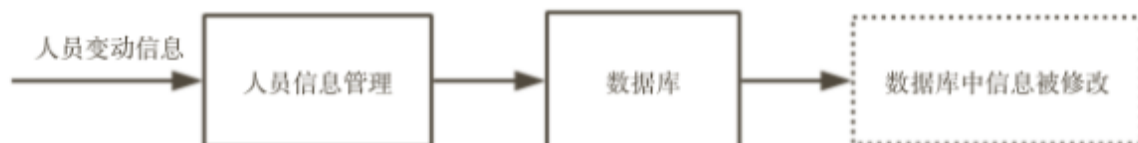
For unauthorized grades, archive management. can be carried after the audit

3.4.5 entering of grade



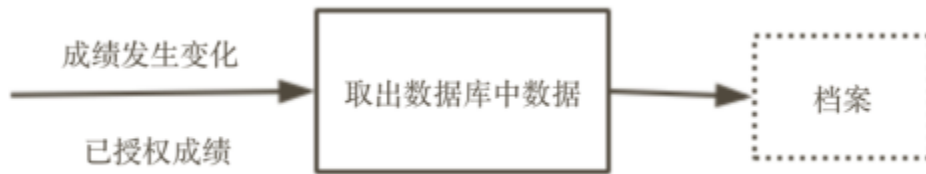
the teacher record the grades , and gets a copy of an unauthorized grades.

3.4.6 management of personal information



when personal information changed, the teacher and update the personal information.

3.4.7 retrieve of data



when grades need to be changed, the manipulation above would be conducted

3.5 Implementation problem

After adopting the shared warehouse mode, all the quality problems are the query and transmission speed problems of the central data warehouse, so as long as the two speeds are controlled, the quality problems can be guaranteed.

1. This mode guarantees the quality requirements such as scalability, portability and maintainability.
2. For performance, when the number of visitors increases greatly, the access speed may be limited by the query speed bottleneck of the data warehouse;
3. For reliability and security, this model loopholes, when the client masquerading the database request, it may cause data leakage.

solution:

4. For performance, se distributed database technology to improve the query speed;
5. For security, enhance authentication, data transmission multiple encryption.

4. System quality analysis and judgement

4.1 Conversation Analysis

- 1) Teachers release examinations, students get the news after the completion of the exam, the teacher after the examination of students to complete the examination results, apply for archiving after the error, the archivists file authorization, the system is responsible for archiving, data stored in the database
- 2) Administrators log on to manage student and teacher information, granting permission to change the contents of the database after determining their rights, and retrieving permissions after they do not operate for an interval
- 3) Students log in to the system, check their own scores and personal information, the center of the database to determine its authority to return data; when the students modify personal data, the database to determine the permissions, authorize changes in the modification is completed, save the new data, recover Modify permissions.

4.2 Prototype Analysis

System is essentially a layer of B / S structure of the system, after completion of a different permissions to share a central client data warehouse type. The central data warehouse can be used directly according to the needs of a database or link the school's educational system.

.

4.3 Risks

4.3.1 Demand Risks

- ① poor demand definition, and further definition will expand the scope of the project;
- ② product-defined part of the ambiguity than expected to require more time;
- ③ In the demand do not participate enough customers;

4.3.2 Organization and Management Risk

- ① by the management or market personnel to make decisions, the plan is slow, longer schedule;
- ② project team structure low productivity
- ③ management review and decision-making cycle longer than expected;
- ④ budget cuts, disrupt the project plan;
- ⑤ management made a decision to combat the enthusiasm of the project organization;
- ⑥ lack of necessary norms, leading to errors and duplication of work;

4.3.3 Personnel Risks

- ① The prerequisites (such as training and other items) can not be completed on time;
- ② poor relations between developers and management, leading to slow decision-making, affecting the overall situation;
- ③ lack of incentives, morale is low, reducing the production capacity;

Some people need more time to adapt software tools and environments that are not yet familiar

⑤ After the project is joined by new developers, it needs to be trained and gradually communicate with existing members so as to reduce the working efficiency of existing members.

⑥ did not find the project urgently needed people with specific skills.

4.3.4 Developing Environment Risks

① new development tools longer than expected learning period, a wide range of content.

② development tools not in place in time;

③ development tools are not as effective as expected, developers need time to create a working environment or switch to new tools;

4.3.5 Customer Risks

① customers are not satisfied with the final delivery of the product, requiring redesign and redo;

② the customer's opinion has not been adopted, resulting in the final product can not meet user requirements, which must be redone;

③ customer planning, prototype and specifications of the audit decision-making cycle longer than expected;

④ Customer does not or can not participate in the planning, prototype and specification stage of the audit, resulting in demand instability and changes in product life cycle;

⑤ The time of customer's response (such as answering or clarifying the time of the question related to the demand) is longer than expected;

⑥ The quality of customer-supplied components is poor, resulting in additional testing, design and integration efforts, and additional customer relationship management.

4.3.6 Product Risks

Correcting poor quality unacceptable products requires more testing, design and implementation than expected;

② the development of additional functions do not need (gold-plated), to extend the schedule;

③ strict requirements compatible with the existing system, the need for more than expected test, design and implementation work;

(4) It is required to connect with other systems or systems not controlled by the project team, leading to unpredictable design, implementation and testing;

⑤ unforeseen problems arising from the operation in unfamiliar or untested software and hardware environments;

⑥ developing a brand new module will take longer than expected;

⑦ relying on the technology under development will extend the schedule.

4.3.7 Design and Implementation Risks

① design quality is low, leading to repeated design;

② some of the necessary features can not be achieved using the existing code and libraries, developers must use the new library or develop new features;

The low quality of the code and library, resulting in the need for additional testing, error correction, or re-production;

④ overestimated the enhanced tools for the planned savings;

⑤ respectively developed modules can not be effectively integrated, you need to re-design or production.

4..8 Process Risks

- ① a lot of paper work makes process slower than expected;
- ② pre-quality assurance behavior is not true, leading to duplication of work later;
- ③ too informal (lack of compliance with software development strategies and standards), resulting in poor communication, poor quality, and even the need to re-develop; Too formal (doctrine adhere to software development strategies and standards), resulting in excessive useless jobs;

A. Appendices