<P.E Grades Management System>

Software Requirements Specification

<1.0>

<2017.11.29>

<Deliang Zeng>

Requirements Analysis Engineer

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 2017.11.29 | Version 1.0 | Deliang Zeng | This is the first version. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

Revision History 2

Table of Contents 2

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. General Description 5

2.1 Product Perspective 5

2.2 Product Functions 5

2.3 User Characteristics 6

2.4 General Constraints 6

2.5 Assumptions and Dependencies 6

3. Functional Requirements 7

3.1 Context 7

3.1.1 Context Diagram 7

3.1.2 Context Analysis 7

3.2 Data Flow of Level 1 8

3.2.1 Data Flow Diagram of Level 1 8

3.2.2 Illustration 9

3.3 Details of Each Manipulation Block 10

3.3.1 Student Manipulation Block 10

3.3.2 Administrator Manipulation Block 10

3.3.3 Teacher Manipulation Block 11

3.3.4 Dean Manipulation Block 12

3.4 Summary: Data Flow of Level 2 12

3.5 Functional Structure of PEGMS 14

3.5 Entity Relationship Diagram 14

4. Non-Functional Requirements 15

4.1 Quality Requirements 15

4.1.1 Performance 15

4.1.2 Reliability 15

4.1.3 Availability 15

4.1.4 Security 16

4.1.5 Maintainability 16

4.1.6 Portability 16

4.2 Engineering Requirements 16

4.2.1 Design Constraints 16

4.2.3 Logical Database Requirements 16

5. Change Management Process 16

# Introduction

This section firstly introduces the main purpose of designing the P.E Grades Management System, the scope of this management system. And then some definitions, acronyms, and abbreviations will be listed beneath. At Last, we will give a common overview of the overall organization of this document.

## 1.1 Purpose

This document is intended for describing the requirements and architecture of the P.E Grades Management System. It is written for the development department of the software development company responsible for implementing this system.

## Scope

The P.E Grades Management System is a system designed to help user(s) querying, typing in, modifying P.E grades of students or granting authorities above to other users. More specifically, the PEGMS provides different functions and features for different types of users.

Users of dean type can publish the grades submitted by teacher users. They can also grant the modification authority on published grades to the teacher users. Users of teacher type can type in new grades, edit the existing grades, or submit the grades to the dean users. Meanwhile, they can put in personal algorithm to compute the “regular exercising score” of their own students.

Users of student type can log in to the PEGMS to query the grades of their own, and can set different filters to view grades of a specific term.

In addition, an administrator user is added to manage all the users in the system. For example, administrator can grant/revoke query authority to/from a student user.

## Definitions, Acronyms, and Abbreviations

dean: P.E dean

teacher: P.E teacher

grade: P.E grade

SRS: Software Requirements Specification

User Table: The table containing information of all the users in the system.

Published Grade Table: The table containing published grades.

Interim Grade Table: The table containing grades not confirmed by the dean.

PEGMS: The P.E Grades Management System.

UT: User Table

PGT: Published Grade Table

IGT: Interim Grade Table

## References

[1]IEEE Software Engineering Standards Committee. IEEE SA 830-1998, IEEE Recommended

Practice for Software Requirements Specifications. October 20, 1998.

[2]王安生. 软件工程化.北京: 清华大学出版社, 2014

## 1.5 Overview

The rest sections of this document are organized as following sequence:

General Description: This part includes product perspective, product functions, user characteristics and so on.

Functional Requirements: Use vivid diagrams with relevant annotations to elaborate

the functions the PEGMS will have.

Non-Functional Requirements: Some other requirements in performance, reliability,

security, etc.

Change Management Process: Describe what are to be done when an indispensable

change is needed in the system.

# General Description

## 2.1 Product Perspective

The system is expected to helps record the grades of the students. A student’s grades are consist of two parts: common class grades and regular exercising grades. The former are typed in by teachers while the latter are determined by his/her exercising circumstance and teacher’s rules.

## 2.2 Product Functions

(1)For a student user, the functions include: logging in/out, querying own grades.

(2)For a teacher user, the functions include: logging in/out, adding/deleting/querying/modifying grades of students in his/her class, submitting grades of the class to the dean.

(3)For a dean user, the functions include: publishing all the grades, allowing a certain teacher to manipulate the publisher grades once if some individual errors occur.

(4)For a administrator, the functions include: managing all users’ information, which may include: id, role, authorities, password, etc.

## 2.3 User Characteristics

Student: A basic role, whose grades are stored in database.

Teacher: A role who manages several students.

Dean: A role of higher position, managing the teachers.

Administrator: A virtual role that has the highest authority.

(Actually, the administrator can be a superior dean)

## 2.4 General Constraints

|  |  |
| --- | --- |
| Factor | constraint |
| Duration | Within 1 month |
| Expenditure | Within 1000 yuan |
| Products | Application, User Handbook |

## 2.5 Assumptions and Dependencies

We assume that:

1. Every student, teacher and dean have a personal account in the PEGMS.

2. These accounts are stored in the UT (User Table).

3. Only the administrator has the authority to modify the password of a specific user.

4. All grades of previous terms are published and read-only. We just take into account the grades of current term.

5. A student’s grade in a certain term is consist of two parts: class grade and grade of regular exercise.

# Functional Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation.

## 3.1 Context

### 3**.1.1 Context Diagram **Diagram\_1****

Border

Students

Deans

Administrator

PEGMS

Teachers

### **3.1.2 Context Analysis**

The Context Diagram shows the PEGMS, the external entities and the relationships among them.

The central rectangle represents the PEGMS.

Four outside rectangles respectively represent: administrator, deans, teachers, students.

The arrows show the directions of the data flows. Each type of entity contacts with the PEGMS through its unique interfaces.

## Data Flow of Level 1

### Data Flow Diagram of Level 1

Deans

Administrator

Teachers

Dean

Manipulation Block

Teacher

Manipulation Block

Students

IGT

UT

PGT

Student

Manipulation Block

Administrator

Manipulation Block

**Diagram\_2**

### ****Illustration****

The diagram\_2 amplifies the internal structure of PEGMS and shows the

flows going among the internal functional blocks. There are three tables used to store all the information: User Table, Published Grade Table, Interim Grade Table.

-UT：This table stores users’ information of all type. The information contains user’s id, password, which authorities the user has, etc. It can be accessed by administrator all the time and by deans under certain circumstance.

-IGT and PGT: The format of a record in both grade table is like this: (

Student’s Id, Teacher’s Id, Dean’s Id, Term, Last Modified Time, Score). When the grades haven’t been checked by a dean, they are in IGT and can’t be seen by the students. After being confirmed, grades will be transmitted from IGT to PGT. Consequently, the students can query them now.

In addition to storage blocks, four function blocks are set for different types of users: Student Manipulation Block for student users, Teacher Manipulation Block for teacher users……Inside the blocks, different kinds of transaction logic are implemented respectively. Later, we will elaborate them.

## Details of Each Manipulation Block

### ****3.3.1 Student Manipulation Block****

PGT

**Diagram\_3**

Students

Querying Module

This block is the simplest one, inside which is a Querying sub-block. It provides several interfaces for student users to query their own grades. When a user calls this block, it calls the query statements to query the database where the PGT is stored, get the grades and return them to the user.

### ****3.3.2 Administrator Manipulation Block. Diagram\_4****

Password Management Module

Authority Management Module

Selection Module

Administrator

UT

As the Diagram\_4 shows, administrator logs in to manage either users’ authorities or passwords. Respectively, two modules are set to access the UT. And before using one of the modules, administrator must go through a medium module named Selection Module.

### 3.3.3 Teacher Manipulation Block

IGT

PGT

Selection Module

Rules

Module

Compute

Module

Input Module

Teacher

**Diagram\_5**

Teacher Manipulation Module allows teacher users to modify, submit grades in IGT in normal time and modify grades in PGT under exceptional circumstances (This requires a dean user’s permission). Above all, every teacher shall access the Rules Module to set his own rules for grade of regular exercise, which can be permanently stored. Afterwards, the teacher only needs to input the class grade and records of regular exercise of a student to the Input Module. Then data will be delivered to the Compute Module. Soon, it automatically calls the Rules Module to get the rules and calculate the final grade. At last, the grade is written into IGT (or the PGT).

### 3.3.4 Dean Manipulation Block Diagram\_6

Dean

UT

PGT

IGT

Publication Module

Grant

Modification

Module

Selection Module

In Diagram\_6, a dean user has two kinds of operations. The first is to grant the modification authority to a teacher user so that the teacher user can access the PGT once (This operation is relevant to the UT). Another is to publish the interim grades in IGT, which will be automatically transmitted to PGT later.

## Summary: Data Flow of Level 2

Combine the previous five diagrams to form a global data flow diagram:

PEGMS

Diagram\_6: Data Flow Diagram of Level 2

Deans

Teachers

Input

Module

Selection Module

Rules

Module

Querying Module

Grant

Modification

Module

Publication

Module

Compute

Module

Students

Administrator

IGT

Password Management Module

Selection Module

UT

PGT

Authority Management Module

Selection Module

## 3.5 Functional Structure of PEGMS

Diagram\_7

I

II

IV

V

The diagram of functional structure elaborates every functional block and its sub-block(s) in a way of layering. From the first layer to the second, the blocks are classified according to rules. So this diagram can also be regarded as a **Use Case** **Diagram**.

## Entity Relationship Diagram

Entity Relationship Diagram provides an efficient approach to describe concept models through entity types, attributes and relationships. Before drawing the diagram, some relationships are listed beneath:

1. A student is managed by a unique teacher.
2. A teacher manage several classes, each of which includes many students. In other words, a teacher teaches lots of students. Meanwhile, he/she is supervised by a unique PE dean.
3. A dean supervises several teachers.
4. There is a unique superior dean owning the only administrator account and managing all other deans.
5. A record of grade table is also relevant to which term it is now, so a supplementary entity term is added in the diagram:

1..1

1..n

1..1

0..n

1..1

0..n

1..1

1..1

0..n

1..1

0..n

1..n

1..n

1..1

Student

Term

Grade

Administrator

Dean

Teacher

# Non-Functional Requirements

## 4.1 Quality Requirements

### 4.1.1 Performance

1. With a probability of higher than 97%, the PEGMS can support 8000 users on line or more.

2. With a probability of 100%, the system can support more than 1000 users’ simultaneous queries. And the delay shall not exceed 0.5 second.

3. With a probability of higher than 99%, the system can support more than 500 users’ simultaneous modifications. And the delay shall not exceed 3 seconds.

4. Less than every 10 minutes, the system shall refresh its data once.

5. The system shall have memories which supports ten-year storage of grades and three-year of log.

### 4.1.2 Reliability

1. At more than 98% of time, the system works regularly.

2. The system can check its work state once per day, and report the consequence to the administrator.

3. When faults occur, the system shall recover within 15 minutes.

### 4.1.3 Availability

1. Friendly user interface.

2. Explicit operation navigation.

### 4.1.4 Security

1. Backup Mechanism shall be supported.

2. Periodic check shall be supported with an interval of no more than one week.

### 4.1.5 Maintainability

1. Easy to test when errors occur.

2. Loose coupling. This means that a modification in one block will rarely affect another.

### 4.1.6 Portability

The system must be available in Windows series and Linux.

## 4.2 Engineering Requirements

### 4.2.1 Design Constraints

1. Use Java or C#.

2. Based on Web Service.

3. Match the MVC pattern.

### 4.2.3 Logical Database Requirements

MySQL DBMS.

# Change Management Process

If there are needs for change in requirement or project scope, a **Specification Change Request** should be submitted to be analyzed. Once SCR accepted, trace the process of modification in SRS. On the other hand, convert (or translate) the changes in SRS to those in development to form a **Requirement Change Request.** Then analyze the RCR. Not until the RCR is authorized can later works begin.