Database Management System

Project Introduction

Contents

- 1. Project background
- 2. Project objectives
- 3. Project functions
- 4. Development environment
- 5. Implement idea
- 6. Project iteration
- 7. Project preparation

Copyright Declaration

Project Background

Database Management System is large-scale software which can manipulate and manage the database. It is used to establish, use and maintain the database, and called DBMS for short. It manages and controls the database in a unified manner to ensure the security and integrity of the database. The user accesses the data in the database through DBMS. The database administrator also maintains the database through DBMS. It can make many applications and users establish, modify and query the database through different methods at the same time or different moments. Most DBMS provide DDL (Data Definition Language) and DML (Data Manipulation Language) for users to define the schema structure and permission constraint of the database, and implement the data operations: add, delete, etc.

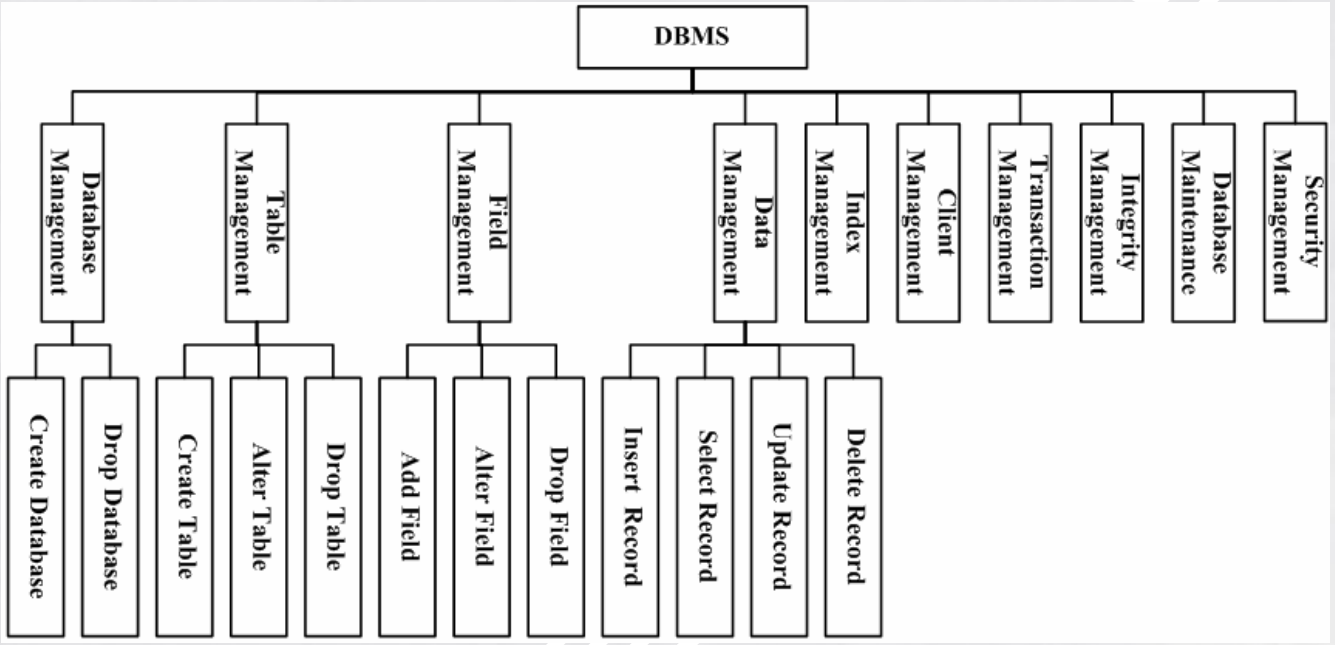
Project Objectives

Through the development of ”Database Management System (DBMS)” project to achieve the following goals:

1. To obtain knowledge of business background, and learn the function of “DBMS”.
2. To understand Microsoft Visual Studio 2010 IDE.
3. To master C++ Fundamental Programming, To master MFC framework, including MFC Dialog, MFC SDI, Tree View, List View and Basic Controls.
4. To master the DDL function, DML function and DCL function of the DBMS.
5. To understand OS conception, thread and process operation.
6. To perform project requirement research and analysis, read and fill the document of Project Requirement and System Design.
7. To understand the development process of a project and TSP, understand the software structure, iterative thinking and develop GUI application program.
8. And to get good coding habit, make improvements on personal self-management, elevate programming practice ability so as to develop enterprise application program.

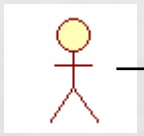
Project Functions

1. System Functional architecture



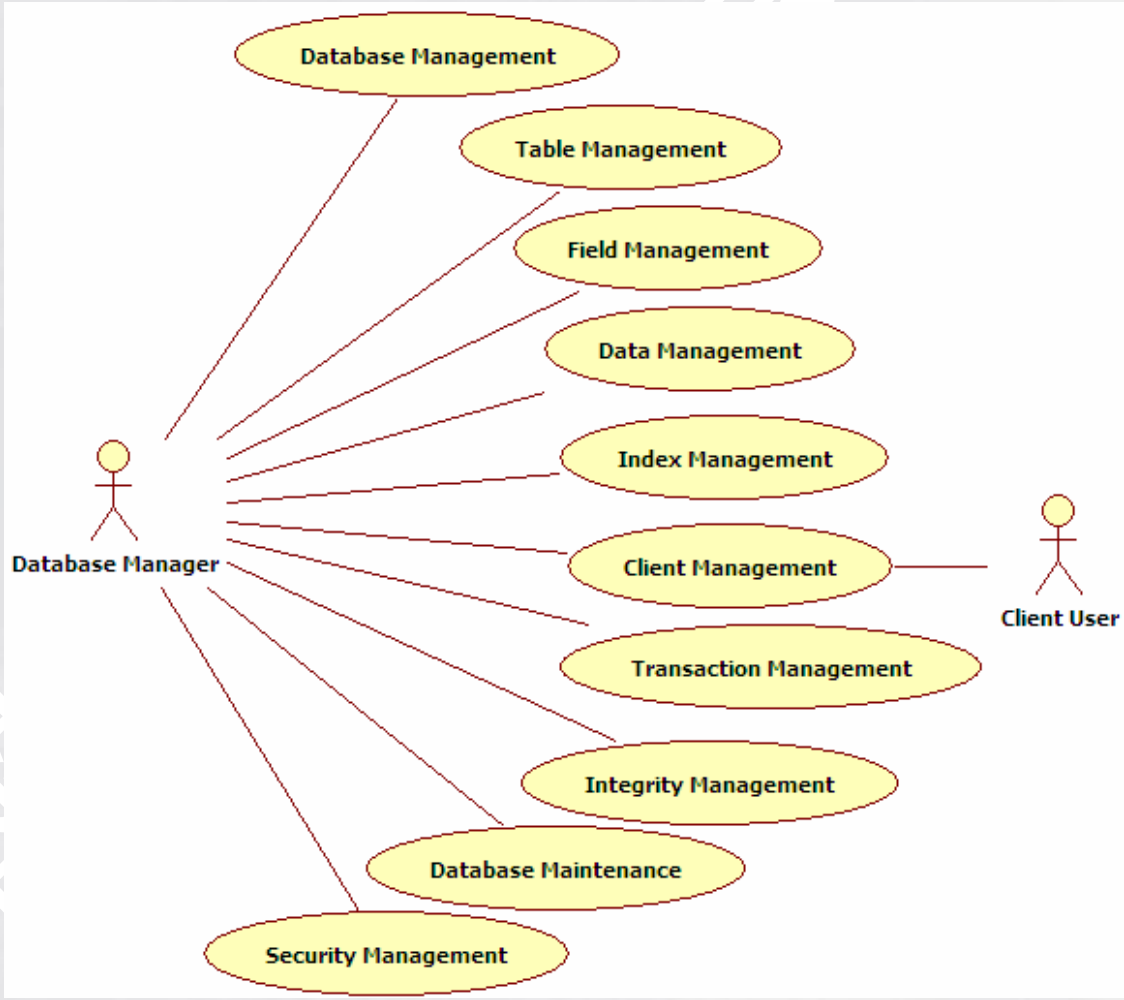
Project Functions

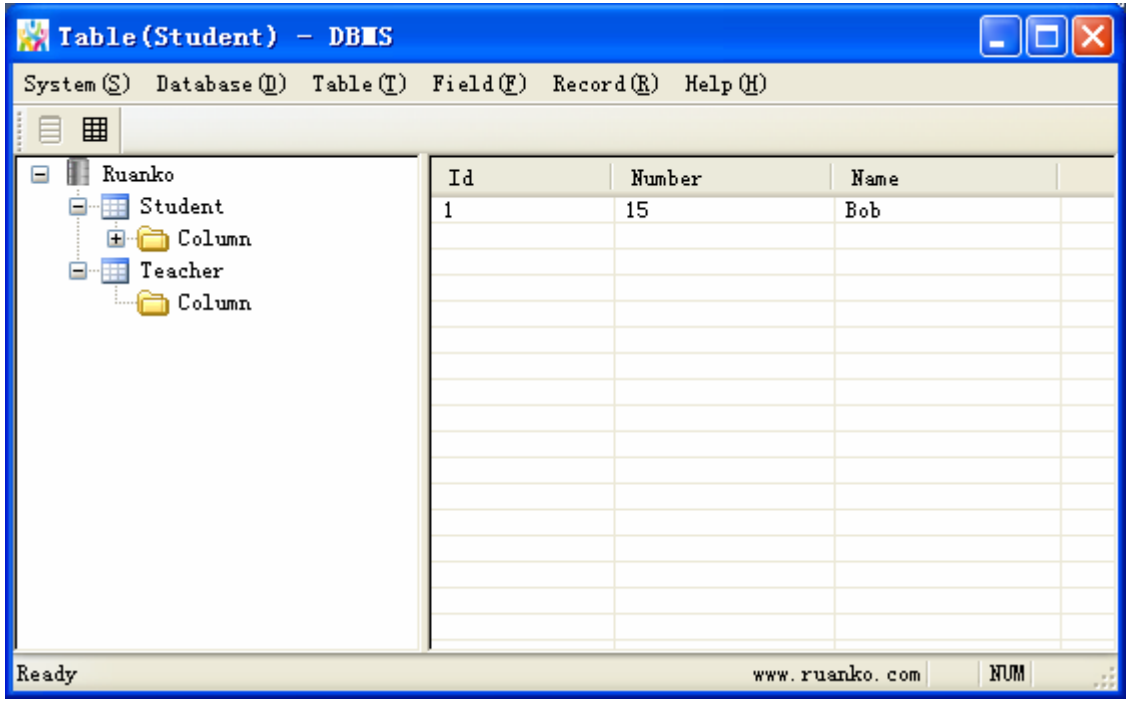
2. System use-case



actor

use case
(function)





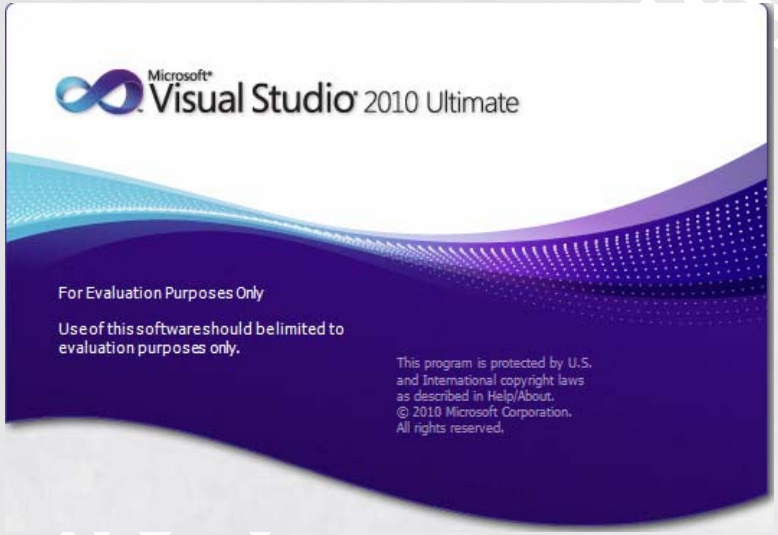
Implementation

Development Environment

1. Development platform: Windows.

2. Development tool: Microsoft Visual Studio (version 2008, 2010, 2010 etc)

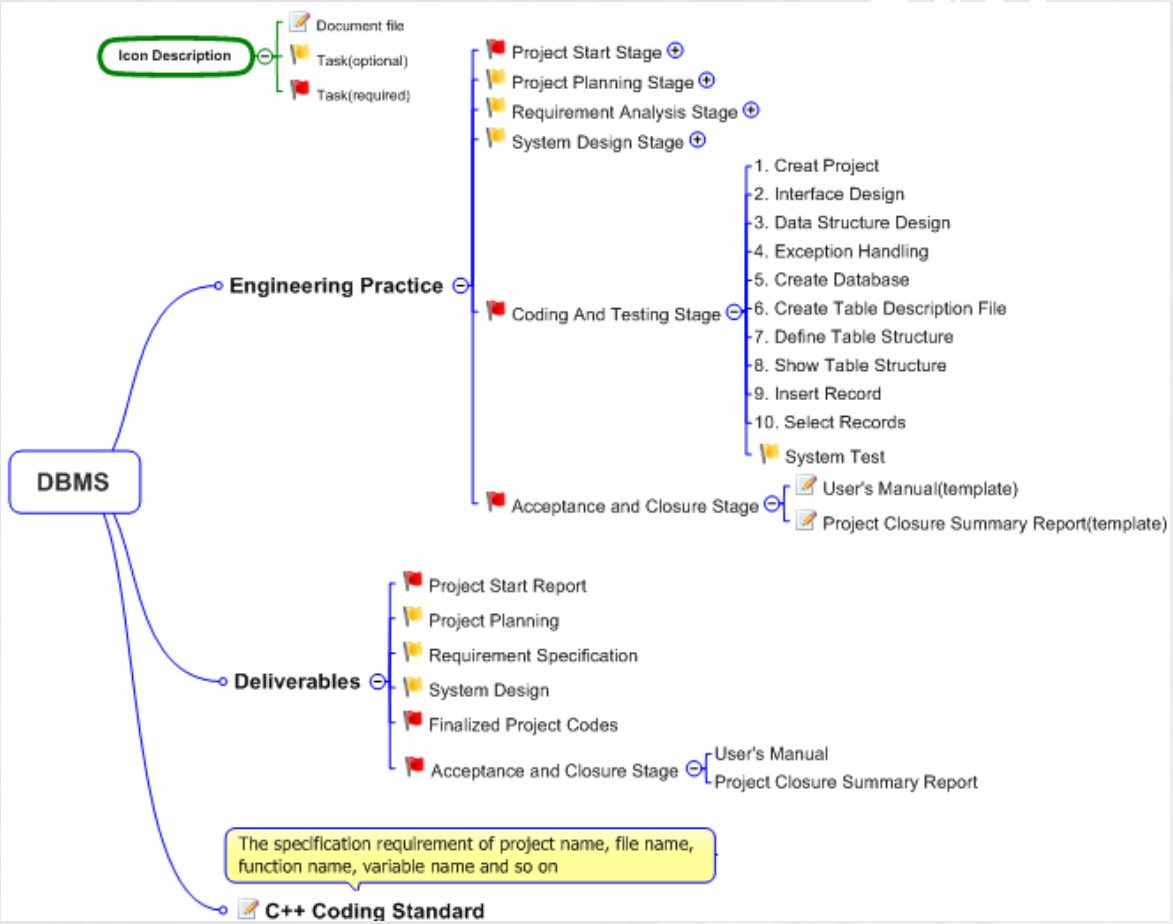
Example Visual Studio 2010



3. Data storage: binary file.

Implement idea

The Project is developed under an iterative thinking, following the path **Framework -> Interface -> Data design -> Function**, among which the Function development is separated into several child iterations, each completed as a child project with processes of requirement, design and implementation.



Project Iteration

软酷·卓越实验室 Centre Of Excellence

1. The function iteration idea of the first part:

Iteration	Function	Technology
create project	create MFC SDI project; build program hierarchy	VS2010, MFC SDI, 3 layer structure
interface design	create standard window; interface layout	standard window; resource view; CSpliterWnd
data structure	design DBMS data structure, file structure and entity class	data structure; entity class, binary file
exception handling	Customize exception class CAppException; uniformly handle exception	exception handling mechanism; customized exception
create database	create database description file "ruanko.db"	MFC SDI application start process; create tree view; read-write binary file
create table description file	create table description file "*.tb"	modal dialog box; menu event; view update
define table structure	create table definition file "*.tdf"	basic control; list control; list view; view switch
show table structure	read table definition file "*.tdf"	list view; pass by value among views
insert record	create record file "*.trd"	pass by value between view and dialog box; binary file location; Map data structure
select records	read record file "*.trd"	tree view; list view

Project Iteration

2. The function iteration idea of the second part:

Iteration	Function	Technology
index	create index description file *.tid; create index data file *.ix	hash table; hash function
secondary index	modify ix file structure	memory paging
data consistency check	update ix file content	modify binary file
user connection	.implement server and client	Socket communication
transaction management	.implement multi-user concurrent processing and transaction	multithreading; multi-thread synchronization
integrity check	create integrity description file *.tic	integrity constraint
database maintenance	database recovery and backup	file operation
security management	user management and authority management	authority management

www.ruanko.com

Thanks

Project Introduction