**功能描述文档**

系统中根据不同的权限分为三类用户：管理员、教师、学生。通过主界面选择用户类型，输入账号（即为工号或学号）、密码登录，密码输入正确则可进入对应用户界面，进行相关操作。

**管理员：**

* 创建、修改、删除、显示教师帐号

教师帐户包括包括教师账号（工号）、姓名、密码（初始密码均为123456）、院系。用户以教师工号登录。

创建、删除教师账号可以选择创建或删除单个账号，直接输入相应信息；也可以选择通过文件批量创建或删除账号，文件中每行信息为一新建账号的账号（教师工号）、姓名、院系或删除账号的教师工号，信息间用“#”分隔，“#”间作为一条信息属性，允许出现空格。

修改教师账号，可以选择修改工号、姓名、院系或重置密码。其中重置密码只能将密码重置为初始密码123456，不能设置指定新密码，若要修改密码，需要用户登录后自行修改，这样也使得账号更加安全。

显示教师帐号可以显示所有已经创建的教师信息，包括账号（工号）、姓名、院系，但为了安全，不会显示其密码。

* 创建、修改、删除、显示课程

课程信息包括课程号、课程名称和课程描述，以及负责老师的工号。课程名称以简单的中文或英文命名。同一课程允许有多名负责老师。

创建、删除课程时可以选择创建或删除单门课程，直接输入对应课程信息；也可以选择通过文件批量创建或删除课程，文件中每行信息为一新建课程的课程号、名称、课程描述、负责教师数目、负责教师工号或删除课程的课程号，一行内的信息间用“#”分隔，“#”间作为一条信息属性，允许出现空格。

修改已经创建的课程，可以选择修改课程号、课程名称、课程描述以及负责教师。其中修改负责教师时，会将原来的一个或多个负责教师与该课程解绑，再将新的一个或多个负责教师与该课程绑定，故若当有多个负责教师，想要在原来基础上增加或删除负责教师而保留其他负责教师时，需要将所有负责教师工号输入，包括原来就已经绑定的负责教师。

显示课程可以显示所有已经创建的课程，包括课程号、课程名称、课程描述、负责教师工号。

* 修改管理员账号密码

修改该账号的密码。为确保安全，要修改密码需要先输入正确的原密码，再输入新的密码（长度不不少于6位）并需要再次输入以确认新密码，若两次输入一致，则修改密码成功，否则修改密码失败。

**教师：**

在管理员添加教师账户后，教师就可以登录系统，进行以下操作。

* 添加、修改、删除、显示、查找学生账户

对某门课程，添加、修改、删除、显示、查找学生账户。学生帐号的基本信息包括学号、姓名、专业、年级、班级，学生使用学号登录。

添加学生账户时，若库中不存在该学生，则创建该学生账户（初始密码123456）；若已存在，则直接添加该课程的学习关系。删除学生账户时，可以只在某课程中删除该账号，不影响该账号的其他课程的学习；也可以彻底删除该账号，即在所有课程中以及student库中彻底删除该账号。创建、删除学生账户时可以选择创建或删除单个学生账户，直接输入对应学生账户信息；也可以选择通过文件批量创建或删除学生账户，文件中每行信息为一新建学生账户的学号、姓名、专业、年级、班级或删除学生账户的学号，一行内的信息间用“#”分隔，“#”间作为一条信息属性，允许出现空格。

修改学生账户，包括修改学号、姓名、专业、年级、班级以及重置密码。其中重置密码只能将密码重置为初始密码123456，不能设置指定新密码，若要修改密码，需要用户登录后自行修改，这样也使得账号更加安全。

显示学生帐户，可以在某课程中，显示所有学生信息；或根据学号查找学生帐号，显示学生信息（学号、姓名、专业、年级、班级）。

* 发布、修改、删除、显示课程信息

课程信息（notice）是想告知学生的关于该课程的信息，如课程通知、参考资料、课堂纪律等。发布、修改、删除、显示某课程的课程信息，需要先进行权限检查，只有教授该课程的老师才可以对该课程的课程信息进行各项操作。

* 查看教授的课程

显示该老师教授的所有课程。

* 发布、修改、删除、显示作业/实验

布置作业或实验。作业信息由布置作业号、课程编号、布置老师工号、作业标题、作业要求组成。发布、修改、删除、显示作业/实验，需要先进行权限检查，只有教授该课程的老师才可以对该课程的作业、实验（包括教授该课程的其他老师布置的作业、实验）进行各项操作。

发布、删除作业/实验时可以选择发布或删除单个学生账户，直接输入对应作业/实验信息；也可以选择通过文件批量发布或删除作业/实验，文件中每行信息为一新建作业/实验的布置作业号、课程编号、布置老师工号、作业标题、作业要求或删除作业/实验的布置作业号，一行内的信息间用“#”分隔，“#”间作为一条信息属性，允许出现空格。

修改已布置的作业、实验信息，包括作业标题、作业内容。

显示作业/实验，可以显示该课程所有已发布的作业。

* 查看学生作业/实验完成情况

对于某一课程，可以查看某作业/实验的完成情况，显示信息包括作业应交人数、完成人数、未完成人数、完成率，选择作业编号又可查看已/未完成学生信息；也可以查看某学生的所有作业/实验（该课程）的完成情况，显示信息包括布置作业的编号、标题、要求和上交作业的编号、标题、内容。

需要权限检查，老师只可查看所教授课程的作业、实验（包括教授该课程的其他老师布置的作业、实验）的完成情况。

* 修改教师账号密码

修改该账号的密码。为确保安全，要修改密码需要先输入正确的原密码，再输入新的密码（长度不不少于6位）并需要再次输入以确认新密码，若两次输入一致，则修改密码成功，否则修改密码失败。

**学生：**

在教师添加学生账户后，学生就可以登录系统，进行以下操作。

* 查看所学课程

查看该学生所上的所有课程。

* 新建、修改、删除作业/实验

上交作业信息包括上交作业号、布置作业号、课程号、学生学号、标题、内容。新建、修改、删除作业/实验，需要先进行权限检查，只有学习该课程的学生才可以对该课程布置的作业、实验进行作答，只有该上交作业的上交学生可以修改、删除、查看该作业、实验。

新建上交作业时，先输入课程号，权限检查通过后，显示该课程布置的所有作业/实验，以便于选择作业布置编号。根据布置作业号，新建该布置作业的上交作业（包括标题，内容）。修改、删除作业时，输入课程号，先显示该课程上交的所有作业/实验，以便于选择作业上交编号。根据上交作业号，修改、删除该上交作业。

新建、修改上交作业/实验时，可以选择直接输入标题和内容；也可以选择通过文件上传，文件名作为（新）上交作业标题，文件内容作为（新）上交作业内容。

* 查询作业完成情况

分别显示该学生已完成的作业/实验以及未完成的作业/实验。

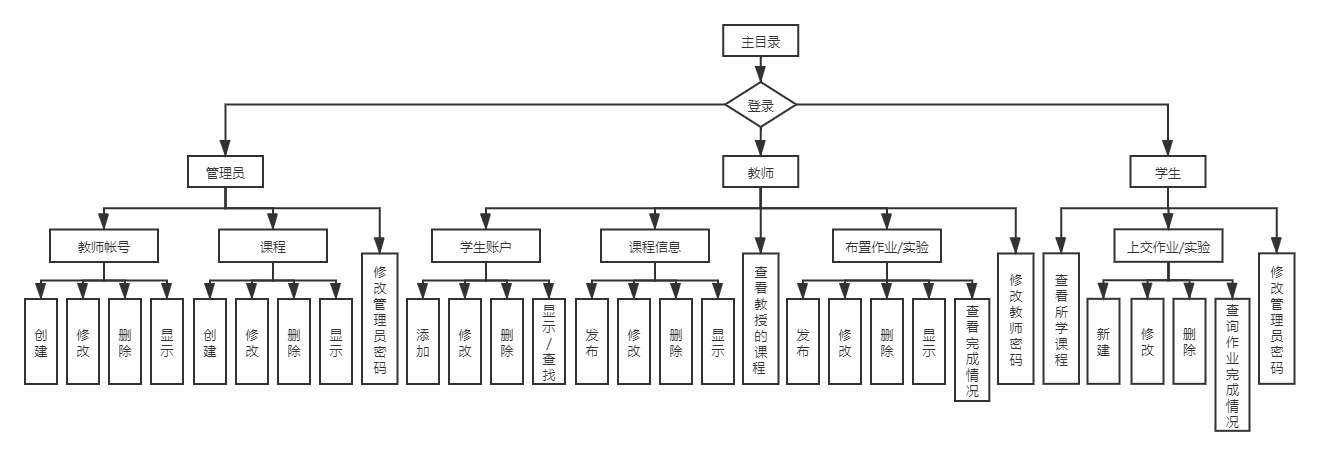
* 修改学生账号密码

修改该账号的密码。为确保安全，要修改密码需要先输入正确的原密码，再输入新的密码（长度不不少于6位）并需要再次输入以确认新密码，若两次输入一致，则修改密码成功，否则修改密码失败。

**设计文档**

* 设计思想

利用数据库来实现数据的存储，这比起用文件来存储，操作更加方便快捷，数据存储也更有条理。系统中根据不同的权限分为三类用户：管理员、教师、学生，有着不同的权限，可以进行不同的操作，保证安全。

* 功能模块
* 数据结构

为了数据存储，设计了management数据库，其中包括admin，teacher，student，course，teach\_course，take\_course，notice，hw\_assign，hw\_submit九个表。定义如下：

CREATE TABLE admin (

admin\_id varchar(10) not null,

name varchar(30),

password varchar(10),

primary key (admin\_id));

CREATE TABLE teacher (

teacher\_id varchar(10) not null,

name varchar(30),

password varchar(10),

department varchar(30),

primary key (teacher\_id));

CREATE TABLE student (

student\_id varchar(10) not null,

name varchar(30),

password varchar(10),

major varchar(30),

grade varchar(10),

class varchar(15),

primary key (student\_id));

CREATE TABLE course (

course\_id varchar(10) not null,

title varchar(30),

description varchar(50),

primary key(course\_id));

CREATE TABLE teach\_course (

teacher\_id varchar(10) not null,

course\_id varchar(10) not null,

primary key(teacher\_id,course\_id),

foreign key(teacher\_id) references teacher(teacher\_id) on delete cascade on update cascade,

foreign key(course\_id) references course(course\_id) on delete cascade on update cascade);

CREATE TABLE take\_course (

student\_id varchar(10) not null,

course\_id varchar(10) not null,

primary key(student\_id,course\_id),

foreign key(student\_id) references student(student\_id) on delete cascade on update cascade,

foreign key(course\_id) references course(course\_id) on delete cascade on update cascade);

CREATE TABLE notice (

notice\_no integer not null auto\_increment,

course\_id varchar(10) not null,

title varchar(50),

content varchar(500),

primary key(notice\_no),

foreign key(course\_id) references course(course\_id) on delete cascade on update cascade);

CREATE TABLE hw\_assign (

assign\_no integer not null auto\_increment,

course\_id varchar(10),

teacher\_id varchar(10),

title varchar(50),

requirement varchar(200),

primary key(assign\_no),

foreign key(teacher\_id) references teacher(teacher\_id) on delete cascade on update cascade,

foreign key(course\_id) references course(course\_id) on delete cascade on update cascade);

CREATE TABLE hw\_submit (

submit\_no integer not null auto\_increment,

assign\_no integer,

course\_id varchar(10),

student\_id varchar(10),

title varchar(50),

content varchar(500),

primary key(submit\_no),

foreign key(assign\_no) references hw\_assign(assign\_no) on delete cascade on update cascade,

foreign key(student\_id) references student(student\_id) on delete cascade on update cascade,

foreign key(course\_id) references course(course\_id) on delete cascade on update cascade);

* 算法

利用MySQL进行数据处理与操作。所有操作最终都通过sql语句在MySQL中完成。

新建、发布操作即向数据库对应表中插入新纪录（insert）；修改操作即修改数据库对应表中记录（update）；删除操作即删除数据库对应表中对应记录（delete）；查找、显示等操作即在数据库中进行查找操作（select）。其中，表与其他表之间的foreign key均为“on delete cascade on update cascade”，便于删除操作。