**“学生毕业管理系统”**

**需求规格说明书**

**版 本 号： \_\_ V2.0 \_\_\_\_\_\_**

**编 写 者： 肖梦杰**

**审 核 者： 赵鹏程**

**批 准 者： 赵鹏程**

## 1.引言

### 1.1目的

（1）编写目的：该需求文档的撰写目的是为对“学生毕业管理系统”做出完整的需求分析，文档面向小组内部成员，用以按照需求明确软件设计、安排项目规划，以及日后对系统进行改进，为组内开发成员、维护人员提供共同的协议以保证开发任务能够顺利进行。这是项目开发的基础，对小组日后工作具有指导的意义。

（2）本系统是专为大学教师和学生设计的毕业管理系统，主要用于任课教师录入评价值，专业负责人格式化培养方案，课程负责人审核课程，导员预警学生和学生查询。

### 1.2文档范围

本文档是项目的需求文档，即需求规格说明书，是技术文档。

本文档使用对象为：

（1）项目负责人；

（2）组内开发成员。

未经过项目负责人书面许可，该文档不得提供给上述规定对象以外的人员阅读或使用。

### 1.3定义、简写和缩略语

学生毕业管理系统：本文档中简称为“系统”。

### 1.4引用文件

《“学生毕业管理系统”设计说明书》

### 1.5综述

## 2.总体描述

### 2.1产品描述

本项目的目标是：

（1）决策支持：在老师的支持和帮助下，全面理解需求；根据项目的需求及时提供所需信息，并在一定阶段对各模块的进度进行追踪及提示，实现工作的协同化、提高了工作效率。

（2）提高效率：“学生毕业管理系统”（以下简称“本系统”）是为计算和管理学生毕业要求达成度的管理信息系统。

需方：学生、任课教师、专业负责人、课程负责人和辅导员；

用户：学生、任课教师、专业负责人、课程负责人和辅导员；

开发方：“学生毕业管理系统”项目开发小组，成员：赵鹏程、严轶轩和肖梦杰；

支持机构：西北大学；

当前运行现场：自建测试网络环境；

计划运行现场：西北大学校园网环境。

### 2.2产品功能

本系统有针对每个角色有不同的功能，每个角色在系统中完成自己的任务，共同推动系统对每个学生的毕业要求指标点的评价值的计算，全面地了解每个学生的个人最终评价结果，辅导员预警和学生查询等功能；

（1）系统DFD模型：



（2）系统业务流程图：



（3）基于活动的业务流程图：

图中共5个角色，图中序号表示业务顺序，每个业务及其对应的数据载体都附在右下角；



### 2.3用户特点

本系统的用户很多元，有学生、任课教师、专业负责人、课程负责人和辅导员共5种主要角色，还包括系统维护成员。

虽然用户是多元的，但我们小组在开发的过程中泛化为两种角色，就是学生和老师，后期迭代过程中细化和增加老师的功能，以满足用户的所有需求；维护成员也就是项目的开发团队的学生成员。

### 2.4约束

### 2.5假设和依赖

### 2.6需求分配

## 3.具体需求

### 3.1：外部接口需求

1.用户界面

2.硬件接口

3.软件接口

4.通信接口

### 3.2：功能需求

**一、学生模块**

1.学生登录系统查询学业信息，包括：个人评价达成度、毕业要求评价结果、指标点达成度和课程达成度；

2.学生会收到辅导员发来的预警通知；

3.学生修改个人信息。

**二、任课教师模块**

1.任课教师在系统查看自己所带的课程；

2.任课教师可以在系统下载用来导入学生课程评价值的模板excel文件；

3.任课教师也可以查看和手动修改每个学生的课程评价值；

4.任课教师可以使用excel文件一键导入学生的课程评价值；

5.任课教师可以查看课程负责人对学生课程评价值的反馈信息；

6.任课教师管理个人信息。

**三、课程负责人模块**

1.课程负责人在系统查看自己所负责的课程；

2.课程负责人对任课教师录入的学生课程评价值进行审核；

3.课程负责人反馈给任课教师审核结果；

4.课程负责人管理个人信息。

**四、专业负责人模块**

1.专业负责人在系统中格式化培养方案；

2.专业负责人第一步一键导入excel格式化毕业要求；

3.专业负责人第二步格式化指标点；

4.专业负责人第三步一键导入excel分配课程指标点；

5.专业负责人将数据导出；

6.专业负责人对成绩数据进行分析，查看系统计算得到的课程平均分、班级平均课程评价值和年级个人评价结果；

7.专业负责人管理个人信息。

**五、辅导员模块**

1.辅导员在系统查看学生成绩数据，包括课程平均分、班级平均课程评价值和年级个人评价结果；

2.辅导员查看学生预警信息；

2.辅导员管理个人信息。

## 4：性能需求

（1）硬件平台：对所要开发系统、软件的需求和约束；

开发系统：需要windows 7或者更高版本的windows系统；

从功能指标上看，必须具备基本的文件管理功能，强大的控制能力和良好的扩充性。满足系统控制，输出和功能扩展。

（2）软件平台：

  根据系统功能的要求，操作系统应提供良好的图形显示支持和强大的多任务管理能力；

从应用软件开发的角度考虑，应选择开发平台功能强，共享软件资源丰富，硬件驱动支持多的操作系统；

支持python、nodejs、vue等前端开发与数据库存储的开发环境；

足够的并发用户数量（总用户量及并发用户量）。

## 5：设计约束

从应用软件开发的角度考虑，应选择开发平台功能强，共享软件资源丰富，硬件驱动支持多的操作系统；

**系统的设计约束：**

（1）尽量减少IO以及网络的访问，将多次的调用整合在一次操作中完成，尽量减少IO资源的浪费；

（2）对于系统的配置文件，数据库字段修改，或者其他显示复杂逻辑修改；尽量采用增加的操作；而少采用update的操作；update永远比insert成本大的很多；

（3）系统之间交互，pull的效果往往比push来的稳定性高；选择只读API，而不是读写API，“写”部分尽量采用事件驱动或者消息驱动；

（5）往往内存中的复杂数据结构组装要优先于数据库的链接。

**数据库设计约束：**

（1）尽可能在数据模型上控制业务对象的约束关系，如果通过程序逻辑去保证完整性与一致性，会存在一定的风险；

（2）数据模型总的唯一性约束，比如学号，一定要在数据库层面得到控制；

（3）尽量少用存储过程，将复杂的业务逻辑抽离到上层应用中，也就是时候尽量使用程序中的数据结构完成复杂的关系运算，避免用存储过程或者复杂的sql语句，因为应用服务器的扩展以及优化的成本往往比DB服务器的成本小的多；

（4）sql语句尽量不要依据业务逻辑以及动态拼接的sql字符串，而是采用预编译的方式，否则有sql注入的风险；

（5）如果主表与子表是一对一的关系，主键尽量相同。

**外部交互设计规范：**

（1）最好是pull对方的数据比较好，比对方push过来稳定性好；

（2）异步消息处理的时候，最好先落地到本地库再进行处理；这样避免消息的丢失，以及消息队列的堆积，导致消息系统挂掉；

（3）系统中只能有一种异常：处理中状态等待超时或者重试次数达到最大值。

## 6：其他需求