



软件项目管理报告

LOGO HERE

文件管理序列号: [K8UY-K9IO69-O6M243-OL889-F88688]

一 可行性研究报告

1. 可行性研究的前提

1.1 要求

通过调查, 要求系统需要有以下功能:

- (1) 要求有良好的人机界面;
- (2) 较好的权限管理;
- (3) 原始数据修改简单方便, 支持多条件修改
- (4) 方便的数据查询, 支持多条件查询;
- (5) 相应的权限下, 删除数据方便简单, 数据稳定性好;
- (6) 数据计算自动完成, 尽量减少人工干预;

1.2 目标

- a. 人力与设备费用的节省;
- b. 处理速度的提高;
- c. 控制精度或生产能力的提高;
- d. 管理信息服务的改进;
- e. 决策系统的改进;
- f. 人员工作效率的提高。

1.3 条件、假定和限制

- a. 开发软件运行的最短寿命为一年。
- b. 进行系统方案选择比较的期限: 2 周。
- c. 经费来源和使用限制: 自筹资金。
- d. 法律和政策方面的限制: 本软件公司版权所有, 未经作者允许, 非法传播、复制, 违者追究法律责任, 后果自负。
- e. 硬件 CPU p3、内存 256M. 。

f. 软件: access2003。

g. 运行环境：本软件应使用 Windows2003、Windows xp 操作系统。

h. 开发环境：本软件应使用 Windows2003、Windows xp 开发。

i. 开发软件投入使用的最迟时间为 2013 年 10 月 01 日。

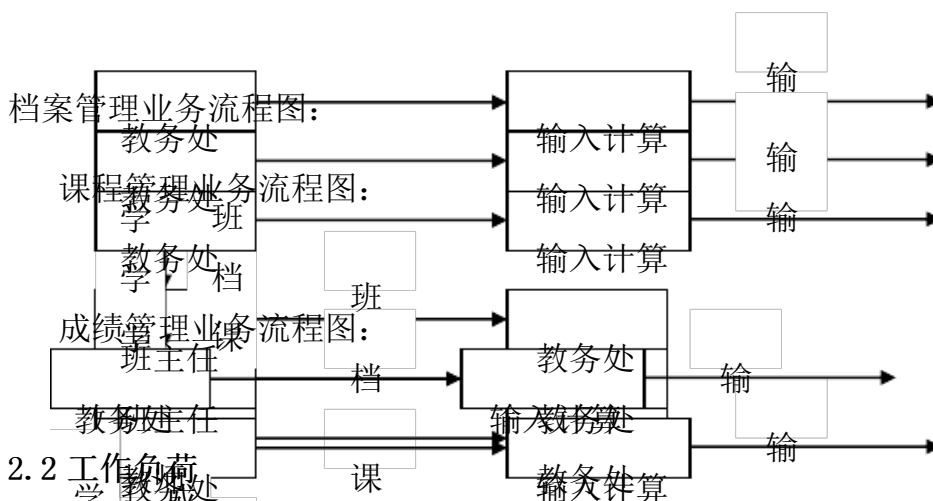
1.4 可行性研究方法

由于本系统管理的对象单一,都是在校学生,且每个数据内容具有较强的关联性,涉及的计算过程不是很复杂。因此,比较适合于采用数据库管理。且学校用于学生管理的微机都是 PIII 以上的机器,在存储量、速度方面都能满足数据库运行的要求。在技术难度方面,由于有指导老师的指导和相关参考文献,特别是网上资料,特别是参考其它程序的功能,因此完全可以实现。

2. 对现有系统的分析

2.1 处理流程和数据流程

班级管理业务流程图:



现有系统所承担的工作只能实现档案管理的简单功能,无法适应目

2.3 费用支出

2.3 费用拨出

开发这个项目总需三个人，4 台计算机，一个可容纳 6、7 个人的办公室，必须有充足的物质做精神动力，每台计算机上必须有所需要的软件，比如：办公软件、数据库软件、截图软件等，必须有 3000 万元的准备开支。

二 软件项目计划

1. 引言

1.1 编写目的

软件项目开发是一项系统而复杂的工作，它需要一个团队互相配合、分工协作。软件项目管理系统可以规范一个软件开发团队的日常工作，提高工作效率。

为了很好的管理整个开发过程，同时预算整个开发过程的费用及时间的安排，给开发人员，管理人员一个参照物，明白自己在每一个阶段所需要完成的任务，协助他们更好地完成开发工作。

预期的读者：开发人员，项目经理，测试人员

1.2 背景

- a. 学生信息管理系统
- b. 提出者：项目经理，开发者：XXX 开发团队。

2. 项目概述

2.1 工作内容

- 1 需求分析：1~3 个月
- 2 概要设计：2~3 个月
- 3 详细设计：2~3 个月
- 4 编码：2~3 个月
- 5 测试：1 个月
- 6 发布：1 个月

2.2 主要参加人员

参与者	个人情况
XX	软件工程专业学生，熟悉 java 语言，数据库编程
XX	软件工程专业学生，熟悉 C#语言
XX	软件工程专业学生，有很好的网页设计能力
XX	软件工程专业学生，有良好的界面设计的能力和测试经验
XX	专业为软件工程，从事开发工作一年，能过独立地完成小型项目的整个开发过程

2.3 产品

2.3.1 程序

名称	编程语言	媒体形式	功能及能力
系统功能	C#+SQL Server 2000	文本	管理学生的学籍信息，统计学生的相关信息。学生信息的增加、修改、删除、查询
数据信息管理	C#+SQL Server 2000	文本	学生学籍信息管理，学生选课信息管理
基本业务	C#+SQL Server 2000	文本	学生注册、学籍信息维护，学生选课，老师管理班级信息。
信息浏览与查询	C#+SQL Server 2000	文本	管理员学生学籍信息浏览、查询
数据库	SQL Server 2000	数据库文件	数据库文件可以直接附加到本地的 SQL Server 2000 中的数据库中
学生学籍管理系统	C#+SQL Server 2000	CD 光盘	程序的运行文件，运行之后只要发布之后就可以了

2.4 验收标准

验收标准参照表：

程序	CD 光盘	实现功能：在使用时，由管理员输入需要了解的关键字信息，然后通饭卡管理系统选择相应的管理事务。管理事务根据所提供的信息在数据库中查找相应的记录，返回相应的
----	-------	--

		记录给管理员。允许管理员在相应的权限下对数据进行修改。通过终端把得到的内容显示到相应的界面上。
安装指南	视频	有声的视频，分解安装的动作，并详细讲述过程及注意事项
用户使用手册	视频+文档	文档给出输入输出的约束条件，系统的运行环境，视频显示使用软件的整个流程，帮助用户学习使用该软件
故障解决说明书	文档	列出基本问题的详细解决办法，详细叙述整个流程

3. 实施计划

3.1 预算

1. 采购必要设备的投资：

网络平台的建设，包括了建设方式和联网建筑物数等等方面去计算，这一块需要 200 万左右；

服务器与存储系统，从发卡量和设备数量等估算，这一块需要 100 万左右；

射频卡终端，包括读写器与 POS 机，这一块需要 20 万左右。

2. 开发系统的投资：

按目前市场上一卡通管理系统的开发价格来看，开发所需的投大概在 50 万不等；

4. 总计： :350 万左右；

3.2 关键问题

本系统的操作过程简单，实现技术要求也不高，所以没有要特别列出的关键问题

4. 支持条件

4.1 运行环境

- a. 开发软件运行的最短寿命为一年。
- b. 进行系统方案选择比较的期限：2 周。
- c. 经费来源和使用限制：自筹资金。
- d. 法律和政策方面的限制：本软件公司版权所有，未经作者允许，非法传播、复制，违者追究法律责任，后果自负。
- e. 硬件 CPU p3、内存 256M. 。
- f. 软件：access2003。
- g. 运行环境：本软件应使用 Windows2003、Windows xp 操作系统。
- h. 开发环境：本软件应使用 Windows2003、Windows xp 开发。

4.2 需由用户承担的工作

数据库的初始化需要用户自己录入，这个应该在测试之前完成，所以编码之前，由开发人员做好数据库，然后由用户安排人录入初始数据库，且必须在 2013 年 6 月 1 日之前完成。

4.3 需由外单位提供的条件

本项目希望得到委托商的资金支持，人员支持，如取需求时，能够提供部分食堂为我们的测试的提供支持环境，还有技术支持。

三 风险条目跟踪表

序列号	1
确定日期	
撤消日期	<撤消风险确定日期
描述	未得到学校支持而导致的无法获得学生数据
可能性	0.2
影响	10
危害值	10.2
降低风险计划	项目开始前与校方沟通，否则项目无法进行
负责人	XXX
截止日期	2013、10、1

五 进度计划风险列表

1. 最常见的进度计划风险

- 1) 功能无限蔓延;
- 2) 质量不定
- 3) 计划过于乐观
- 4) 设计欠佳
- 5) 银弹综合症
- 6) 研发导向开发
- 7) 人员薄弱
- 8) 签约商失败;
- 10) 研发人员与客户的磨擦。

2. 进度计划风险完整列表

2.1 计划编制风险

- 1) 计划、资源和产品定义全凭客户或上层领导口头指令，并且不
完全一致;
- 2) 计划是优化的，是“最佳状态”;
- 3) 计划忽略了必要的任务;
- 4) 计划基于使用特定的小组成员，而那个小组成员其实指望不
上。
- 5) 在限定的时间内无法建成已定规模大小的产品;
- 6) 产品规模比估计的要大一些;

- 7) 工作量大于估算数;
- 8) 进度已经拖延的项目在重新评估时过于优化或忽视项目历史;
- 9) 过度的进度压力造成生产率下降;
- 10) 目标日期提前, 但没有相应地调整产品范围或可用资源;
- 11) 一个任务的延迟导致相关任务的连锁反应;
- 12) 涉足不熟悉的产品领域, 花费在设计 and 实现上的时间比预期的要多。

2.2 组织和管理

- 1) 项目缺乏一个有凝聚力的最高领导人;
- 2) 由于前期乏力, 项目长时间被搁置;
- 3) 解雇和削减开支导致项目小组能力下降;
- 4) 仅由管理层或市场人员进行技术决策, 导致计划进度延长;
- 5) 低效的项目组结构降低生产率;
- 6) 管理层审查/决策的周期比预期时间长;
- 7) 预算削减打乱项目计划;
- 8) 管理层做出了打击项目组织积极性的决定;
- 9) 非技术的第三方的工作比预期延长(如审批, 采购等);
- 10) 计划性太差, 无法适应期望的开发速度;
- 11) 项目计划由于压力而放弃, 导致开发混乱、低效;
- 12) 管理层强调英雄主义, 而忽视客观确切的状态报告, 这会降低发现和改正问题的能力。

2.3 开发环境

- 1) 设施没有及时到位;
- 2) 设施到位, 但不配套;
- 3) 设施拥挤、杂乱或者破损;
- 4) 开发工具未能及时到位;
- 5) 开发工具不如期望那样有效, 开发人员需要时间创建工作环境或切换新的工具;
- 6) 开发工具的选择不是基于技术需求, 不能提供计划要求的性能;
- 7) 新开发工具的学习期比预期的长, 内容繁多。

2.4 最终用户

- 1) 最终用户坚持新的需求;
- 2) 最终用户对于最后交付的产品不满意, 要求重新设计和重做;
- 3) 最终用户不买进项目产品, 无法提供后续支持;
- 4) 最终用户的意见未被采纳, 造成产品最终无法满足用户期望, 而必须重做。

2.5 客户

- 1) 客户坚持新的需求;
- 2) 客户对规划、原型和规格的审核/决策周期比预期长;
- 3) 客户没有或不能参与规划、原型和规格阶段的审核, 导致需求不稳定和耗时的重复;
- 4) 客户答复的时间比预期长(如回答需求中需澄清的问题);
- 5) 客户坚持技术决策而导致进度计划延长;

- 6) 客户对开发进度管理过细，导致实际进展变慢；
- 7) 客户提供的组件无法与开发的产品匹配，导致额外的设计和集成工作；
- 8) 客户提供的组件质量欠佳，导致额外的测试、设计和集成工作，以及额外的客户关系管理工作；
- 9) 客户要求的支持工具和环境不兼容、性能差或者功能不完善，导致生产率降低；
- 10) 客户不接受交付的软件，尽管它满足了所有的规格；
- 11) 客户期望的开发速度是开发人员无法达到的。

2.6 承包商

- 1) 承包商没有按承诺交付组件；
- 2) 承包商递交的组件质量低下无法接收，必须花时间改进质量；
- 3) 承包商没有买进项目开发需要的工具，进而无法提供需要的性能水平。

2.7 需求

- 1) 需求已经成为项目基准，但变化还在继续；
- 2) 需求定义欠佳，而进一步的定义会扩展项目范畴；
- 3) 添加额外的需求；
- 4) 产品定义含混的部分比预期需要更多的时间。

2.8 产品

- 1) 错误发生率高的模块需要比预期更多的测试、设计和实现工作；

- 2) 校正质量低下不可接受的产品，需要比预期更多的测试、设计和实现工作。
- 3) 在一个或多上新兴领域推广计算机技术使得计划进度的延长不可预期；
- 4) 由于软件功能的错误，需要重新设计和实现；
- 5) 开发额外不需要的功能（镀金）延长了计划进度；
- 6) 要满足产品规格与速度要求，需比预期更多时间，包括重新设计和实现的时间；
- 7) 严格要求与现有系统兼容，需要进行比预期更多的测试、设计和实现工作；
- 8) 要求与其他系统、复杂系统或不受本项目控制的系统相连，导致无法预料的设计、实现和测试工作。
- 9) 要求在不同操作系统下运行将花费比预期更长的时间；
- 10) 在不熟悉或未经检验的软（硬）件环境中运行产生未预料的问题；
- 11) 开发一种对组织全新的模块将比预期花费更长的时间；
- 12) 依赖正在开发中的技术将延长计划进度。

2.9 外部环境

- 1) 产品依赖政府规章，而规章的改变将是不可预期的；
- 2) 产品依赖草拟中的技术标准，而最后的标准将是不可预期的。

2.10 人员

- 1) 招聘人员所花时间比预期的长；

- 2) 作为先决条件的任务不能按时完成（如培训、其它项目）；
- 3) 开发人员和管理层之间关系不佳导致决策缓慢，影响全局；
- 4) 项目组成员没有全身心投入项目，进而无法达到需要的产品性能水平；
- 5) 缺乏激励措施，士气低下，降低了生产能力；
- 6) 缺乏必要的规范，增加了工作失误与重复工作；
- 7) 某些人需要更多时间适应不熟悉的软件工具和环境、硬件环境、编程语言；
- 8) 项目结束前，合同制人员离开团队，或雇员辞职；
- 9) 项目后期加入新的开发人员，额外的培训和沟通降低现有成员的效率；
- 10) 项目组成员不能有效地一起工作；
- 11) 由于项目组成员间的冲突，导致沟通不畅、设计欠佳、接口错误和额外的重复工作；
- 12) 有问题的成员没有调离项目组，损害了项目组其他成员的积极性；
- 13) 项目的最佳人选未加入项目组；
- 14) 项目的最佳人选已加入项目组，但因其他原因未能合理使用；
- 15) 没有找到项目急需的具有特定技能的人；
- 16) 关键人物只能兼职参与；
- 17) 项目人员不足；

- 18) 任务的分配与人员技能不匹配;
- 19) 人员工作的进展比预期的慢;
- 20) 项目管理人员怠工导致计划和进度失效;
- 21) 技术人员怠工导致工作遗漏或质量低下, 工作需要重做。

2.11 设计与实现

- 1) 设计过于简单, 无法确定主要事件, 并导致重新设计和实现;
- 2) 设计过于复杂, 导致一些不必要的工作, 影响实现效率;
- 3) 设计质量低下, 导致重复设计和实现
- 4) 使用不熟悉的方法, 导致额外的培训时间, 并重犯前期使用这种方法时导致的错误;
- 5) 产品采用低级语言来实施, 导致生产率比预期的低;
- 6) 一些必要的功能无法使用现有的代码和库实现, 开发人员必须使用新库或自选开发所要的功能;
- 7) 代码和库质量低下, 导致需要额外的测试、错误修正或重做;
- 8) 过高估计了增强型工具对计划进度的节省量;
- 9) 分别开发的模块无法有效集成, 需要重新设计或重做。

2.12 过程

- 1) 大量的纸面工作导致进程比预期的慢;
- 2) 进程跟踪不准确, 导致无法预知项目是否已落后于计划进度;
- 3) 前期的质量保证行为不真实, 导致后期的重复工作;
- 4) 质量跟踪不准确, 导致无法得知影响进度的质量问题;
- 5) 太不正规, 导致沟通不足, 质量问题和重复工作;

- 6) 过于正规，导致过多耗时无用的工作；
- 7) 向管理层撰写进度报告占用的开发人员的时间比预期的多；
- 8) 风险管理粗心，导致没有发现重大的项目风险；
- 9) 软件项目风险管理花费的时间比预期的多。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/558021005101006071>