****

**软件项目管理**

**本科学生设计性实验报告**

**项目组长 汤政达 学号 0173951**

**成 员 汤政达**

**专 业 软件工程 班级 软件171**

**实验项目名称 高校体育场馆管理系统-项目计划书**

**指导教师 黄轩**

**开课学期 2019至 2020 学年 1 学期**

**上课时间 九至十六周**

目录

[高校体育场馆管理系统-项目计划书 2](#_Toc28704704)

[1. 引言 2](#_Toc28704705)

[1.1 目的 2](#_Toc28704706)

[1.2 范围 3](#_Toc28704707)

[2. 任务分解（自顶向下方法） 4](#_Toc28704708)

[3. WBS字典 5](#_Toc28704709)

[4. 风险评估及对策 12](#_Toc28704710)

[4.1 风险评估及对策表 12](#_Toc28704711)

[4.2风险评估分析 18](#_Toc28704712)

[5. 总体进度计划 19](#_Toc28704713)

[5.1 总体进度计划表 19](#_Toc28704714)

[5.2开发资源 20](#_Toc28704715)

[5.3项目进度设置 21](#_Toc28704716)

[6. 项目活动时间表（甘特图）（详见附件） 21](#_Toc28704717)

[7. 关键路径图（CPM图） 22](#_Toc28704718)

[8. 工期估算 22](#_Toc28704719)

高校体育场馆管理系统-项目计划书

# 1. 引言

## 1.1 目的

以开发者所在的高校——江西财经大学为例，所有的学生、教职工要想进入江西财经大学体育馆，需要事先携带好现金或者校园卡，在体育馆门口售票处购买票后，一人一票凭票入场。整个流程中有许多不便之处：只能用现金或者校园卡支付，而不能用手机支付，与现在人们的主流支付方式有些脱节；无法提前预知场馆内场地租用情况，可能会出现场地人满而白走一趟的情况；也无法进行预约等等。总而言之，目前高校采用的旧式场馆管理方法已经跟不上学生日益增长的需求；我们开发的高校体育场馆管理系统便由此诞生。

限定预期读者：

学生、教职工：身为用户，阅读此软件需求规格说明，有助于问题反馈及功能开发。

线上系统管理员、线下场地管理员：身为软件后台管理员，阅读此软件需求规格说明，有助于问题反馈。

程序员：了解系统功能，编写《用户手册》。

测试员：根据本文档编写测试用例，并对软件产品进行功能性测试和非功能性测试。

## 1.2 范围

通过我们开发的高校体育场馆管理系统，学生与教职工可以在线上进行场地租借的支付，无需带着现金前往场馆运动，省时省力。其次，用户可以通过此场馆管理系统，在线查看所有体育场馆场地的租借情况，并决定是否要前往体育馆。同时，用户可以进行预约操作，不必担心想去运动的时候没有空闲场地。最后，本系统还会向用户推送体育馆实时新闻，如某某馆因场地积水，暂时无法租借等等。而上面所说的所有功能，皆是目前高校采用的旧式管理系统无法做到的。有了我们开发的高校体育场馆管理系统，用户可以享受更便捷、更全面的场地租借服务。

# 2. 任务分解（自顶向下方法）

体育场馆管理系统

用户登录

用户注册

查询场

地状态

个人信息修改

预约信息管理

暂无

暂无

用户信息管理

场地状态管理

场地预约管理

更改场地状态

场地预约

预约支付

# 3. WBS字典

3.1用户注册WBS:

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | Login\_f\_1 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 注册 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 在用户进行操作之前需要进行登录，而登陆必须是用户才能登录，如果要成为用户就要先进行注册 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 注册是让该使用者成为一个用户，能够进行合理的操作，并且该用户的行为将会被实时的记录下来 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 在点击注册选项之后，会弹出注册页面，要求自己创建一个不同于其他人的用户名，自己设定的密码，以及验证的邮箱 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 用户在点击注册之后，将要输入用户名以及两遍密码，输入两遍密码之后还要输入邮箱，并通过点击邮箱进行验证 在此界面用户可以自主选择用户名，设定密码，填入邮箱。 用户名的条件在于：必须和数据库中的任何一个用户名不一样。 密码的条件在于：必须两遍密码都足够的长而且面膜长度必须大于6 邮箱将会接收到一条信息，只有点击了信息所给出的链接才能算验证成功。 最后验证成功之后会生成一个账号给用户使用，这个账号是唯一的，供用户使用。 | | | | | | | | | | | |

3.2用户登录WBS:

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | Login\_f\_2 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 登录 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 在用户进行操作之前都必须登录，只有登录成功了才能进行后续的操作 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 登录其实是让用户在操作是进行实名，也就是在用户进行操作之后能记录下来，从而实现更好的管理，以及信息的处理。 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 在点击登录选项之后，会弹出登录的界面，需要输入的是账号和密码，以及验证码，账号密码都是注册之后能给出的信息，而验证码是动态的4个字母 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 用户在点击登录之后，将要输入账号以及密码，输入密码之后还要输入验证码，所有信息核对正确之后才能登录成功 账号的条件在于：必须和数据库中的账号核对一样，并且存在。 密码的条件在于：必须和对应的账号一一对应。 最后验证成功之后会来到用户个人中心的一个界面，这个界面是首个用户登录的界面，供给用户查看信息或者修改信息的操作。 | | | | | | | | | | | |

3.3个人信息修改WBS：

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | Message\_change\_3 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 个人信息修改 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 在用户登录成功之后的界面上，会有一个个人信息管理的按钮，这个个人信息管理按钮是口语修改用户信息的一个按钮，像修改密码，用户名等等都是可以的 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 个人信息管理，主要是为了让用户方便对自己信息进行修改，因为有时候这些信息可能遭到泄漏，或者忘记，这些情况最好就是进行修改密码，当然同时还有一些时候是根据用户自己本身的需要进行的一些个性化的东西而需要做出的改动。 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 在登录成功之后的页面，点击个人信息管理，则会进入到另外一个具体的信息修改界面 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 用户在点击个人信息管理之后，会出现信息界面的处理，这个按钮的功能主要是为了更美观，并且作为一个信息管理界面的入口，在点击这个按钮才能进入到信息修改查询界面。  条件：登录成功。 | | | | | | | | | | | |

3.4查询场地状态WBS：

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | State\_search\_1 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 查询场地状态 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 用户可以对想要预约的体育场馆进行查询，看看要预约的场地状态是否可用。 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 查询场地状态，是指用户对该体育馆所有场地的信息（包括其状态信息）进行查询，以便后续的预约操作。 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 用户输入要查询的体育馆名称  系统返回该体育馆所有场地的信息（包括其状态信息） | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 用户在点击查询之后，输入要查询的体育场馆名称，便可以跳转到另外一个用户信息的界面，  系统返回该体育馆所有场地的信息进行查看。  条件：登录成功 | | | | | | | | | | | |

3.5更改场地状态WBS：

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | state\_change\_2 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 更改场地状态 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 系统管理员在点击进入场地状态的界面之后，可以点击部分可以修改的信息，进行修改。。 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 此操作仅限拥有管理员权限的用户，才可对场地的状态信息进行更改。 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 管理员在进入场地状态管理页面之后，显示出场地信息，管理员可对场地信息进行增删改操作。 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 管理员在点击信息更改之后，便可以跳转到另外场地状态信息的界面，进行查看，查看过程中点击任何可修改的内容，都可以直接在上面做修改，修改之后的信息点击保存之后会反馈到数据库之中。  条件：登录成功，且为管理员权限。 | | | | | | | | | | | | | |

3.6场地预约WBS：

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | space\_manage\_1 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 场地预约 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 用户可以场地状态为可预约的场地进行预约操作。 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 场地预约是使用本高校体育场馆管理系统最主要的目的，所以场地预约功能是本系统最重要的一个功能模块。 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 用户输入要预约的体育馆名称  系统读取并返回该体育馆所有可预约的场地名称  用户选择要预约的场地  系统读取并返回该场地可预约的时间段  用户选择相应的时间段进行预约  系统完成预约操作，生成订单并提醒用户对订单进行支付 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 同上，主要是数据库的读取到存取。  条件：登录成功。  预约成功条件：场地状态为可预约。 | | | | | | | | | | | |

3.7预约支付WBS：

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | order\_change\_2 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 预约支付 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 在点击了场地预约之后会出现两个类别，一个是对场地进行预约，另一个是对预约生成的订单进行支付。 | | |
| |  | | --- | | 详细描述 | | | |  |  | | --- | --- | | |  | | --- | | 一个完整而且成功的场地预约流程，是要先进行场地预约操作，然后进行订单支付操作。订单支付成功后，数据才会真正写入数据库中，表示预约成功。 | | | | |
| |  | | --- | | 处理流程 | | | |  | | --- | | 在点击订单支付之后，会出现一个支付方式选择界面，用户选择采用何种方式进行支付。选择完成后，将跳转到相应的支付界面进行支付。 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 在点击订单支付之后，会出现一个支付方式选择界面，用户选择采用何种方式进行支付。选择完成后，将跳转到相应的支付界面进行支付。订单支付成功后，数据才会真正写入数据库中，表示预约成功。  条件：登录成功。  支付成功条件：支付页面返回正确的支付成功信息。 | | | | | | | | | | | |

3.8预约信息管理WBS：

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 功能标识号 | | | |  | | --- | | Message\_manage\_1 | | |
| |  | | --- | | 功能名称 | | | |  | | --- | | 预约信息管理 | | |
| |  | | --- | | 说明 | | | |  | | --- | | 管理员在点击了用户信息管理之后，可以对所有用户的预约信息进行查看以及管理。 | | |
| |  | | --- | | 详细描述 | | | 场地工作人员线下对进入场馆人员的预约信息进行查询，来核实用户是否预约此场地。 |
| |  | | --- | | 处理流程 | | | |  | | --- | | 管理员在点击了用户信息管理之后，出现所有用户的预约信息，管理员可以通过精确查找的方式进行用户预约信息的管理。 | | |
| |  | | --- | | 算法流程 | | | | | | | | | | | | |  | | --- | | 管理员在点击了用户信息管理之后，出现所有用户的预约信息，管理员可以通过精确查找的方式进行用户预约信息的管理。  条件：登录成功，且为管理员用户。 | | | | | | | | | | | |

# 4. 风险评估及对策

## 4.1 风险评估及对策表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 排序 | 类别 | 概率 | 影响程度 | 风险描述 | 风险值 | 对策 |
| 1 | 需求风险 | 80% | 80% | 1：客户需求描述不清楚。  2：客户在描述了需求，但后面更改需求。客户需求在变化。  3：在沟通过程中，客户参与度不够，我们未了解客户的需求。  4：客户的专业与我们的专业相差甚远，我们错误理解客户的需求，客户也错误理解我们的想法。  5：需求调研不足，导致系统无法正确反映客户对系统的重要文档。  6：对客户的业务了解不清楚。  7：没有与客户进行良好的沟通。 | 67% | 1：在确定需求前与客户签署一份需求合同。  2：排专人实地调研考察高校的体育场运行方法。并且深入了解客户的业务。  3：在系统开发的每一个阶段，请客户确认是否是他们想要的。  4：以书面形成《用户需求》文档，避免疏漏造成的损失在后续阶段被逐步放大。  5：设立监督策略，项目开发中任何重大的决定都必须有客户参与进行。  6：需求变更需要经过统一负责人提出，并且要求用户领导认可，需求变更应该是定期而不是随时提出。  7：做好详细的记录，让客户了解需求变更的实际情况。  8：项目组与客户高层，中层，核心用户建立沟通渠道和机制。 |
| 2 | 管理风险 | 50% | 90% | 1：对计划和任务定义不充分。制定计划没有制定完善，导致后面出现问题  2：对高校体育馆信息管理系统不了解。  3：夸下海口，太过于自信。  4:项目所有者和决策者分不清。  5：不能与员工进行充分的沟通。  6:对该项目的估计和控制有可能有比较大的偏差，以至于最后超期，或者超支。 | 59% | 1：选择一个熟悉高校体育馆信息管理系统有能力的项目经理。  2：采用两个及以上项目经理策略，当第一个经理确定方案的时候也要通过另外一个经理的同意  3：调整计划，如有必要可以组织加班。  4：控制开支，尽量节省。 |
| 3 | 产品规模 | 20% | 100  % | 1:低估产品规模。对该系统不熟悉，或者未完全了解客户的需求，导致低估了系统的规模。  2：系统规模比以前产品的偏差值大。 |  | 1:对系统做好详细的了解，不轻易下决定。2:在产品进行阶段，要客户来复查是否是他们的要求。 |
| 4 | 技术风险 | 38% | 54% | 1：数据存储的数据太多，导致在删除和插入等各种操作的时候无法快速便捷的达到要求。  2：当数据用户突然增多的时候，系统会崩。  3：在系统服务器端，数据的处理效率较低，因此要考虑到是否需要扩展性，完善性。  4：待开发的系统的硬件接口，使用新的未经证实的接口。  5：在技术的应用领域不足。  6：为了实现良好的软件构架和稳定的组件，使用最新的面向对象的构件和基于微软的COM组件技术，与传统开发方法比较，有相当的多的额外工作需要做，这会给项目工期带来较大的风险。 | 50% | 1：放弃采用新的技术。  2：增加算法数据工程师（或者外包），修改数据存储，运行的算法。  3：增强数据库的存储能力。  4：聘请顾问。  5：如果不能放弃新技术，采用"范例"方式提高开发人员的构件组件的分析识别能力，适时调整构件组件的数量和粒度。 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 技术风险（人员风险） | 30% | 40% | 1：工作人员对方法技术掌握不足。 2：没有对熟悉高校体育馆信息管理系统有能力的员工。 3：缺少经验丰富的员工  4：员工对新的技术和应用不熟悉 | 36% | 1：培训员工。  2：聘请顾问，指导项目完成工程。  3：为专业技术团队聘请有能力的员工。 |
| 6 | 技术风险 | 20% | 50% | 系统体系结构设计不合理  1：过于简单的系统结构，对用户来使用效果不好  2：系统结构过于复杂和灵活会难以实现，并且也会给测试带来风险。 | 36% | 1：在设计系统体系结构的时候，反复检查确认系统系统结构。  2：必要时，在不影响进度与预算的情况下，重新设计系统体系。 |
| 7 | 时间风险 | 30% | 40% | 1：项目经理没有很好的制定计划。  2：技术人员工作经验不足，导致速度慢。  3：工期短，于是放弃编写文档，然后在工程晚期的时候需要文档协调工作，导致工程越来越慢。  4：在不同的工程阶段，人员配合不好。导致工作效率不高，配合不了。  5：该系统的数据处理的开发阶段工作量大，且时间有限，带来较大的时间风险。 | 36% | 1：项目经理制定的计划需要多人同意才能实施。  2:让对高校体育馆信息管理系统有经验的项目经理来制定计划。  3：协调好人员，让以前一起配合做过项目的员工一起配合工作。  4：为保证高校信息管理系统能在最短时间内提交，从生存期上采用敏捷式快速成型。和增量技术进行集成，逐步实现系统的功能和服务。使系统逐步完善起来，为了使平台能够尽快投入使用，除了使用以上策略，还与用户进行协调，实现服务和功能的优先级，按照优先级的顺序由高到底的进行开发，逐步完成全部服务和功能  5：设定应急计划。 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8 | 适应风险 | 17  % | 40% | 适应性：  系统在不进行修改的情况下适应不同的工作环境的能力  风险：  由于硬件的飞速发展和软件开发周期较长的矛盾，软件升级的需要显得非常迫切。如果软件的升级和移植非常困 难，软件的生命期必定很短，使得花费巨大人力物力开发出的软件系统只能在低性能的硬件或网络上运行，甚至被废弃不用，造成巨大的浪费。 | 32% | 1：尽可能地将系统的变化放在系统的配置上，即软件代码无需修改，仅仅是在系统提供的配置文件中进行适当的修改，然后软件重新加载进入运行状态，就完成了系统部分功能和性能要求的变化。  2：对于重大改动，需要打开源代码进行修改的，也仅仅是先继承原先的代码，然后用新的功能接替原先的调用接口，这样将把软件改动量减小到最低。  3：对于软件维护，根据业务需求的变化对系统进行修改。第一批的系统可以没有那么高的适应性，在维护阶段，进行逐步的升级，与提高。 |
| 9 | 人员风险 | 30  % | 60% | 由于近期公司进行了人员调整，并且项目比较多，人力比较紧缺。无法安排多人进行这个项目，并且很有可能会从该项目中抽取部分骨干技术人员去做别的项目，必然会引发计划的变更 | 40% | 1：与上层人员沟通看能否阻止这种调整。  2：公司聘请其他员工，补充成员。  3：重新进行任务分配。 |
| 10 | 质量风险 | 10% | 20% | 1：开发的进展以及软件的质量是否符合要求难于度量，从而使软件的管理难于把握。  2：不同的软件开发项目应当采用不同的或者说是有针对性的软件开发过程，而真正合适的软件开发过程是在软件项目的开发完成才能明了的。 | 25% | 1：根据项目的特点和开发经验进行选择，并在开发过程中不断的调整。  2：设立监督制度，项目开发中任何较大的决定都必须有主要技术环节甚至是由用 户参与进行的。 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11 | 设计风险 | 10% | 30% | 1：分析人员在设计系统结构时过于定制，系统的可扩展性较弱，会给后期维护带来巨大的负担，和维护成本的激增。  2：软件结构的过于灵活和通用，必然引起软件实现的难度增加，系统的复杂度会上升，这又会在实现和测试阶段带来风险，系统的稳定性也会受到影响。  3：文档的不健全造成实现阶段的困难，在后期的测试和维护造成灾难性的后果。 | 20% | 1：根据实际情况和产品进行折中。  2：源代码书写的规范性。把属于程序员自身个性风格的成分引入代码的比例降到最低限度，从而减小了系统整合的风险。  3：注重文档的书写，与文档的可读性。 |
| 12 | 测试风险 | 10% | 20% | 1：没有时间进行产品测试  2：由于时间不够，测试不够好，很多大的问题都没有测试出来。 | 14% | 1：找专业的测试公司完成测试。  2：调整计划，有必要的时间加班。 |
| 13 | 人员流动 | 4% | 40% | 1：项目成员辞职。  2：项目成员对这个项目的团队不满，去了别的项目的团队。  3：项目成员离世 | 30% | 1：搞好团队建设，营造一个好的工作环境和氛围，尽量减少人员流失的可能。  2：做好人员顶替的准备，留有一部分的余地。 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 14 | 资金风险 | 2% | 50% | 客户不能及时履行合同义务而使项目资金紧缺以至延误。 | 29% | 与客户进行沟通，必要时通过法律等必要手段使其履行义务。 |
| 15 | 安全风险 | 10% | 30% | 1：一直注重产品开发，忽略了专利保护。在IT行业，人员流失，变更非常严重，很可能导致产品和新技术泄密。导致产品被其他公司窃取。  2:在软件方面关于知识产权的认定目前还没有一个明确的行业规范。 | 29% | 1：和员工签署项目保密合同，该项目具有商业保密性，在必要时可以采取强制手段保护技术。  2：注重产权认定。 |
| 16 | 安全风险（系统） | 2% | 40% | 系统采用的数据库是MySql进行数据存储，并且服务器的版本比较老，没有自己公司的独立的数据库。因此不能保证有很高的安全性。如果有黑客攻击该系统，很有可能导致数据的丢失。 | 26% | 1：利用防火墙，安全补丁等措施尽可能避免类似的情况发生。  2：建立自己的数据库。  3：采用更加具有安全性的数据库来存储 列如MongoDB  4：及时进行修复和弥补，保留法律解决问题的权利。 |
| 17 | 竞争对手 | 5% | 30% | 1：在商务方面，竞争对手对客户影响。  2：竞争对手有更好的产品解决方案。 | 24% | 1：提出我们现有的匹配的解决方案，并且成功的案例。  2：在项目开始前，与客户签署合同，防止客户中途跑路，找了别家公司。 |
| 18 | 客户方面 | 1% | 20% | 客户财务状况不好，经营难以为继，有破产倒闭的风险 | 15% | 我们这次的系统是高校体院馆信息管理系统，是与高校合作，这方面基本没有风险，但还是要提前调查客户的资金状况。 |
| 19 | 社会风险 | 1% | 20% | 不可抗拒的因素  1：长时间断电。  2:传染病。 | 15% | 在家办公 |
| 20 | 数据记录风险 | 1% | 10% | 1：运维人员没有很好的考察高校运动场场地的大小，个数，类型等，导致在设计系统模式的时候设计不完善。  2：客户提供的数据不完善，存入数据库的数据不完善。 | 14% | 1：添加多个运维人员，实地考察场地。  2：与客户进行沟通，要求客户尽可能提供完善的数据。  3：系统的数据库存储采用灵活的方式，接口更加便捷，使得可以方便地更改变换数据。 |
| 21 | 价格 | 3% | 10% | 1:系统价格折扣非常低。  2：实施价格为闭口合同，折算人工日不到800元  3：高级业务蓝图价格为闭口合同，折算人工日不到3000元 | 14% | 1：与客户进行沟通  2：在不得已的情况下，放弃这个项目。 |

## 4.2风险评估分析

分析风险发生的概率（P），以及对项目的影响（I），从而根据R=F（P，I）计算出风险值。

根据风险概率矩阵图得出风险值

|  |  |  |  |
| --- | --- | --- | --- |
| PRI | LOW | Medium | High |
| Hight | L | H | H |
| Medium | L | H | H |
| High | L | M | M |

# 5. 总体进度计划

## 5.1 总体进度计划表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 子工作 | 完成时间 | 负责人 | 最终交付物 | 描述 |  |
| 团队组建和环境搭建 | 确定小组 | 第三周（3天） | 项目总管 | 团队名单 | 专注于开发高校体育场馆信息管理系统的开发人员 |  |
| 搭建环境 | 第三周（2天） | 项目组长 | java+resful+MySQL | 确定开发环境并搭建环境（多10%的备用机） |  |
| 制定项目管理计划书 | 第四周（5天） | 项目组长 | 项目管理计划书第一版 | 制定高校体育场馆信息管理系统软件过程管理 |  |
|  |  |  |  |  |  |  |
| 完成需求规格说明的出搞 | 采集用户的需求 | 第五周（3天） | 需求调研人员 | 需求的全面确定 | 针对的用户为（学生，教师，校外运动，外校学生，校园领导建议）通过与对象交流，查找了解和采集用户的需求多次。对需求进行汇总，制定需求规格说明。 |  |
| 分析用户的需求及制定需求规格说明 | 第五周（2天） | 需求规格说明书的初稿 |  |
| 需求规格的进一步完善和修改 | 第六周（3天） |  |  |
| 和学校领导做需求规格的最后一次需求共识 | 第六周（2天） |  |  |
| 制定需求规格说明的最后版本 | 第七周（2天） |  |  |
|  |  |  |  |  |  |  |
| 高校体育场馆信息管理系统设计 | 系统总体设计 | 第八周（4天） | 项目团队 | 软件系统设计报告初稿 | 制定系统总体方案，并根据需求说明联系实际进行相应的修改，开发模式为螺线周期开发 |  |
| 系统详细设计 | 第九周（1+2天） |  |
| 系统模式，架构，开发模式，数据库的选择 | 第九周（3天）+第十周（2天） |  |
|  |  |  |  |  |  |  |
| 开发系统源代码和测试 | 系统源码开发 | 第十周（3天）到十三周结束（共计18天） | 项目团队 | 实现需求功能的产品 | 工程的代码开发，测试和包装。 |  |
| 系统源码测试 | 在第十一周（第2天）到第十四周（3天）共计15天 |  |
| 系统功能块集成，检查运行情况，调整。 | 第十四周（2天）+第十五周（3天） |  |
|  |  |  |  |  |  |  |
| 系统交付 | 系统操作说明书 | 第十五周（2天） | 项目团队 | 产品操作技术文档和产品 | 产品和技术文档的提交 |  |

## 5.2开发资源

人员：小组（5名学生+1名教师）

操作系统：window 7 px开发兼容window10

开发地点和设备：机房，实验机环境配置机（8台），内存4G以上，可用外存128G以上。

软件：idea,MySQL,Mockplus原型开发软件，vision绘图软件。

## 5.3项目进度设置

第一周到第二周是可行性分析和风险估计。

第三周到第四周是环境，人员，项目计划书的规划。

第五周到第七周是进行需求分析制定需求规格说明文档。

第八周到第九周是系统设计：系统结构设计及子系统划分，系统功能模块详细设计，系统界面详细设计，外部界面的设计，内部界面设计，用户界面设计。

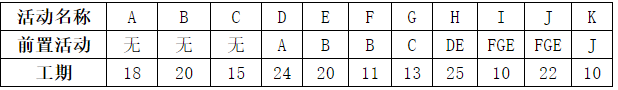
第十周到第十四周是：开发系统源码，测试，并且集成。

第十五周进行系统操作说明书和产品的交付。

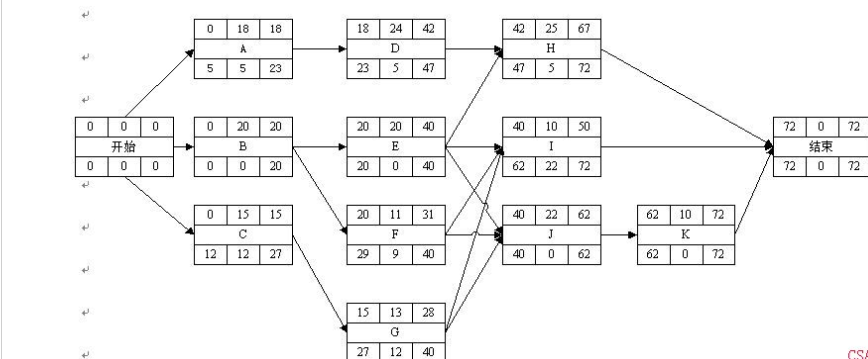
# 6. 项目活动时间表（甘特图）

# 7. 关键路径图（CPM图）

将项目活动时间表（甘特图）简化成一张数表，如下图所示：



将其转换成网络图，如下图所示：



# 8. 工期估算

关键路径即为B-E-J-K，其估算对应的工期为119天。