

1. 项目定位

1.1 项目背景

1.1.1 客户现状

1.1.2 行业发展趋势分析

1.2 项目目标

1.3 项目功能

1.3.1 预期主要功能概要描述

1.3.2 项目扩展设想

2. 实施计划

2.1 主要实施环节

2.2 实施环节时间点

2.2.1 WBS

2.2.2 网络计划和关键路径

3. 总体方案

3.1 项目分析及目标定位

3.2 技术方案

3.3 主要功能

4. 资源要求

5. 项目预算

5.1 开发工作量

5.2 开发管理费用

6. 风险识别

7. 进度安排

1. 项目定位

1.1 项目背景

1.1.1 客户现状

随着经济的快速发展，越来越多的人开始关注精神文化生活，在业余时间会去选择欣赏演唱会、电影、话剧等各种形式的演出，同时人们对服务的快捷、便利性要求也越来越高，从而对售票的建设与管理提出了更高的要求。传统的线下售票方式存在很多的缺陷值得改善，比如实时差、交易时间过长、运营成本高、信息不共享等，已无法满足用户的需求，在这种背景下，我们希望搭建一个网络平台，让用户在到达演出地之前就能了解演出的相关内容，并实现在线订票，以免浪费时间，从而实现票务的信息化。

1.1.2 行业发展趋势分析

随着大麦网、淘票票、猫眼等娱乐票务平台的快速发展，在线票务平台似乎成为了当下电脑观众的首选。然而，演出和电影的观众仍然倾向于购买一流的演出，而且这个市场仍在蓬勃发展，观众越来越多。但是，目前还没有一个全面的平台可以让玩家购买各种演出的门票。所以，博影来了！

1.2 项目目标

Boying是一个综合类现场娱乐票务营销平台，业务覆盖演唱会、话剧、音乐剧、体育赛事等领域，目标是让用户安全、快速、便捷地购买各种演出票务，提高用户的体验感。

Boying——博影娱乐票务平台顺应互联网时代的发展潮流，同时为人们提供了一个优质的购票平台。

1.3 项目功能

1.3.1 预期主要功能概要描述

系统管理员在后台管理各种类型、不同场次的演出票务信息，并且对系统的用户、订单等各种信息进行管理。用户可以在网站上浏览演出列表，查看每一个演出的时间、地点、场次等详细信息，然后根据自己的喜好选择相应的场次进行购票。此外，Boying还提供了评价功能，用户可以对自己观看的演出进行评价。在个人信息界面，用户可以对账号基本信息进行修改，查看自己的历史订单与评价。

Boying是一个开放的平台，用户无需登录账号，便可以浏览网页上的票务信息，但是，用户必须注册账号并登陆后，才可以使用购票以及后续的功能。为了保护用户的隐私，其他用户在访问单个用户的个人主页时无法查看其基本信息和订单、评价信息，用户也可以根据自己的需要在个人主页中修改自己信息。

与传统的购票模式相比，Boying利用互联网的便利性，用户可以实时掌握最新的演出信息，并根据个人兴趣来决定是否买票，由此可避免传统售票模式给用户带来的不良感受，如浪费时间以及购不到票等；此外，Boying是传统售票模式的全流程模拟，用户可在任意有网络的地方完成整个购票流程，不再受到地域限制；对于票务的出品方，Boying实现的票务信息化可以降低实际的购票交易时间，提高公司工作人员的工作效率，降低公司的运营成本，Boying既可以实时共享票务信息给用户，也可以满足用户在线预订的需求，由此可缓解高峰期票务点的购票压力，既让用户享受了优质的购票体验，也为售票公司提供了便利。

为了方便网站的运营人员管理系统，我们为管理员设置了特殊的后台界面，管理员可以在后台管理系统的演出、用户、订单等信息。

1.3.2 项目扩展设想

Boying不仅可以作为购票平台，还可以作为商家宣传、推广其商品的平台，并且可以参与各种秒杀活动，为商家和消费者都提供更好的线上渠道。

2. 实施计划

2.1 主要实施环节

Step	Activity
0: 选择项目	Boying: 一个在线娱乐购票平台
1: 确定项目范围和目标 1.1 确定目标和效果衡量标准 1.2 确立权力 1.3 确定利益相关者 1.4 根据滑冰者分析修改目标 1.5 建立沟通方法	<ul style="list-style-type: none"> - 1.1: 代码主要基于java。编码行数约为10万行。项目总成本约为20000元。 - 1.2: 项目权限由项目指导委员会控制，特别负责目标的制定、监督和修改。目前，这项工作由我们小组的四个人负责。 - 1.3: 溜冰者。项目工作人员；同一组织内的项目外人员，如通讯人员；组织外人员游戏购买者、游戏制造商。 - 1.4: 审查和修改 - 1.5: 沟通方式
2: 确定项目基础设施 2.1 建立项目和战略规划之间的关系 2.2 确定安装标准和程序 2.3 确定项目团队组织	<ul style="list-style-type: none"> - 2.1: 需要决定以什么顺序来执行这些项目，需要建立一个框架来适应新的系统，如硬件和软件标准。 - 2.2: 应该有变更控制和配置管理的标准；在项目生命周期的每一个点上都可能质量检查的规定；还应该有一个测量程序来控制每个阶段必须收集的数据；项目经理应该了解任何相关的项目规划和控制标准。 - 2.3: 大型项目的负责人可能需要控制项目团队的组织结构。虽然我们的团队有非常简单的团队结构。
3: 分析项目特征 3.1 目标或产品驱动 3.2 分析其他项目特征 3.3 识别高层次的项目风险 3.4 考虑到用户关于实施的要求 3.5 选择一般的生命周期方法 3.6 审查总体资源估算	<ul style="list-style-type: none"> - 3.1: 主要是产品驱动。 - 3.2: 特点。 - 3.3: 评估所有项目的风险水平，进行风险事先检查，并关注高风险项目 - 3.4. 客户有时会有自己的监管要求。其中一些在Requirement. - 3.5 Scrum. - 3.6 确定所有项目的资源需求，并考虑项目的人员分配和其他问题。

Step	Activity
4: 识别项目产品和活动 4.1 识别和描述项目产品 4.2 记录通用产品流程 4.3 识别产品实例 4.4 产生理想的活动网络 4.5 修改理想以考虑到阶段和检查点的需要	- 4.1: 确定项目要创造的所有项目, 有助于确保所有需要进行的活动都得到考虑。包括可交付物、中间产品等, 既包括技术产品, 也包括与项目管理和质量有关的产品。这些产品都有自己的层次结构, 可以用产品分解结构来表示。 - 4.2: 通过产品流程图确定产品的创建或使用顺序 - 4.3. 当同一个共同的PFD片段与某类产品的多个实例相关时, 我们要设法确定每一个实例。 - 4.4. 理想的活动网络有足够的资源。 - 4.5: 引入检查点活动来修改活动网络。
5: 估算每项活动的工作量 5.1 进行自下而上的估算 5.2 修改计划以创建可控制的活动	- 5.1: 用网络计划估计每项活动所需的人员工作量、可能的时间消耗和所需的非人力资源 - 5.2: 需要长时间分割的活动, 需要短时间合并的活动。将活动的时间跨度设置为与用于监测和控制项目的报告期相同。
6: 识别出自下而上的估计 6.1 识别和量化基于活动的风险 6.2 适当时计划减少风险和应急措施 6.3 调整计划和估计以考虑到风险	- 6.1: - 6.2: 一些已确定的风险是可以避免的, 或者至少可以减少。如果存在风险, 应急计划就会规定要采取的行动。 - 6.3: 可能会改变计划, 或增加一些新的活动来减少风险。
7: 分配资源 7.1 识别和分配资源 7.2 考虑到资源限制, 修改计划和估算	- 7.1: 记录每项活动所需的员工类型, 确定可用于项目的员工, 并临时分配给这些项目。 - 7.2: 确定任务的优先次序, 以确保关键任务的完成; 确保可用人员的全部工作和高利用率, 用甘特图呈现。
8:计划评审/公布	- 8.1: 当每个任务完成后, 通过确定良好的质量标准来确定是否可以结束任务。 - 8.2: 认真记录计划, 使项目的各个部门了解计划, 并同意承诺执行计划。

Step	Activity
9/10:执行计划/较低层次的计划 &可能需要重复较低层次的计划	一旦项目开始，就有必要为即将开始的每个阶段制定更详细的计划，并放弃后续阶段的详细计划。

2.2 实施环节时间点

2.2.1 WBS

根据软件工程方法论，我们将整个系统开发分为需求分析、概要设计、详细设计、编码、测试和部署六个主要过程，然后分别分解这六个主要过程。

需求分析主要分为需求收集、需求沟通、需求分析和需求确认四个部分。

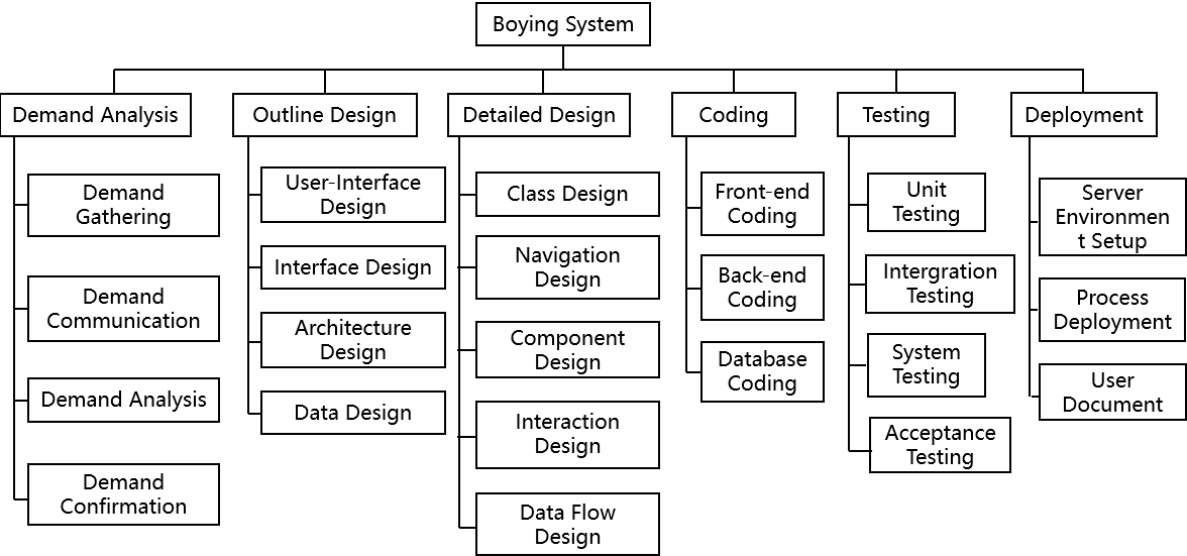
在大纲设计阶段，我们主要完成了界面设计、界面设计、架构设计、数据库设计等内容。

详细设计阶段分为模块设计、类设计、数据流设计、交互设计和导航设计。

编码分为前端编码、后端编码和数据库实现。

根据软件测试理论，将测试分为单元测试、集成测试、系统测试和验收测试。

最后，我们为部署阶段、项目部署和用户文档配置服务器环境。

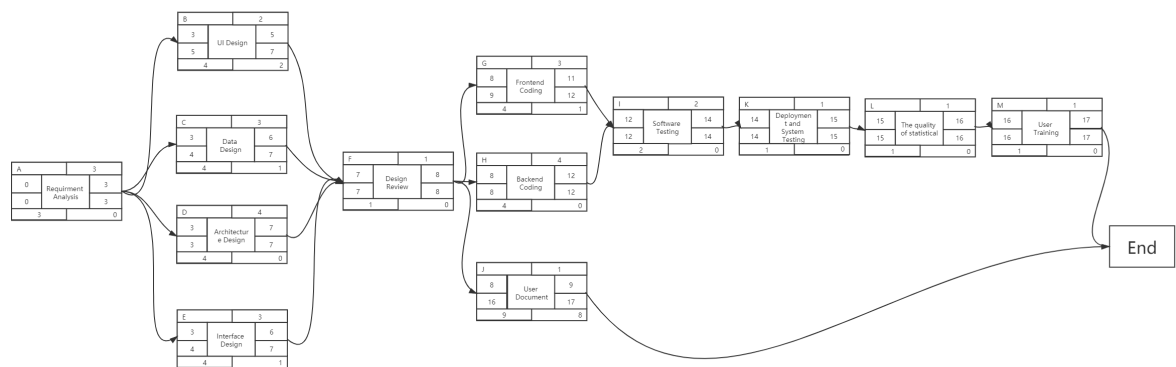


2.2.2 网络计划和关键路径

项目采用基于活动的方法来识别活动，将项目划分为主要生命周期阶段，分别考虑每个阶段的活动及其活动周期，并分析单个活动的前活动，得到以下活动列表：

No	Activity	Estimated Duration (Weeks)	Depends on
A	需求分析	3	None
B	用户界面设计	2	A
C	数据库设计	3	A
D	架构设计	4	A
E	接口设计	3	A
F	设计回顾	1	B,C,D,E
G	前端编程	3	F
H	后端编程	4	F
I	软件测试	2	G,H
J	编写用户文档	1	F
K	部署与系统测试	1	I
L	数据质量分析	1	K
M	用户培训	1	L

网络计划如下所示，黄色背景突出显示**关键路径**:



3. 总体方案

3.1 项目分析及目标定位

随着大麦、淘票票、猫眼等平台的发展，节目似乎成为了观众的首选。然而，其他游戏机的玩家仍然倾向于购买实物表演，以收集或其他目的，并且随着越来越多的玩家购买游戏机，这个市场仍在蓬勃发展。

在这样一个大的环境下，我们决定专门为市场打造一个销售平台，让各大展会厂家在这个平台上销售自己的产品，用户可以在这个平台上进行购买、评价、交流等活动，咨询和展示购买体验。

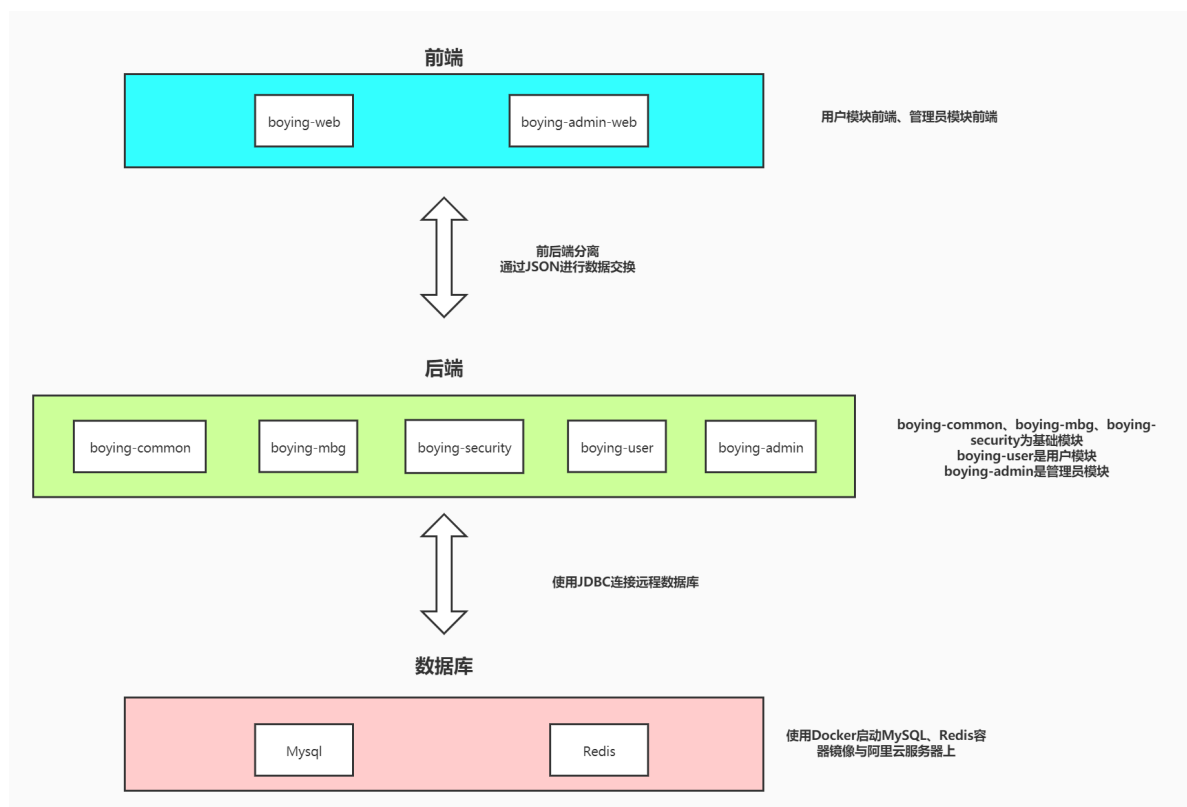
因此，客户必须通过网站访问秀店，管理公司。用户可以通过PC访问它。系统必须允许客户在我们的系统中搜索他或她最喜欢的节目，并将该节目添加到购物车或愿望列表中。该系统必须为出版商提供一个平台来添加新的节目，管理节目相关信息，以及方便地管理订单。

博盈的战略目标包括持续增长和盈利，以及提高认同度和建立平台氛围。本项目以实体展示市场为基础。希望以全面、精致的展示吸引用户的青睐，以简洁、精细化的操作方式优化管理人员的工作。以卓越的预警和反馈提升客户绩效，逐步打造用户在线购物实体展示是首选平台，为我们的买家和合作伙伴提供长期稳定的服务。为了实现这一目标，有必要确保足够的节目来源，并在前期允许一定的开销来增加平台的知名度和完整的节目购买体验。

具体和直接的目标是:

1. 满足业务需求。博盈必须能够完成提供给用户的采购，并为管理员提供收入和支出盘存报表等基本业务功能。
2. 提高工作效率。博英需要对一些预设的场景进行响应，优化管理员的管理，提高工作效率。
3. 提高盈利能力。博英需要完成销售报表、库存信息等的汇总，以便管理员及时调整经营策略，获得更大的利润。
3. 加强公司品牌。博盈致力于改善买家的购买体验，提升平台知名度，加强氛围建设，从侧面提升盈利能力。

3.2 技术方案



CentOS：一个免费的、开源的、可以重新分发的开源操作系统，是Linux发行版之一。

Docker：Docker是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的Linux或Windows机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。

MySQL：MySQL是一个关系型数据库管理系统，将数据保存在不同的表中，是一款开源的、安全、跨平台、高效的，并与PHP、Java等主流编程语言紧密结合的数据库系统。

有体积小、速度快、总体拥有成本低，尤其是开放源码这一特点。

Redis：Redis（Remote Dictionary Server）是一个高性能的key-value数据库，值（value）可以是字符串（String），哈希（Hash），列表（list），集合（sets）和有序集合（sorted sets）等类型。

Vue：一套用于构建用户界面的渐进式JavaScript框架。与其它大型框架不同的是，Vue被设计为可以自底向上逐层应用。Vue的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue也完全能够为复杂的单页应用（SPA）提供驱动。

Spring Boot：Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初

始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域（rapid application development）成为领导者。

RBAC：RBAC（Role-Based Access Control），是一种“用户-角色-权限”管理方式，即基于角色的权限访问控制。

HTML：HTML 是一种基于文本格式的页面描述语言，是网页通过的编辑语言。

3.3 主要功能

功能用户：

1. 用户在博英注册。
2. 用户登录Boying。
3. 用户视图在Boying中显示类别。
3. 用户视图显示了Boying的详细信息。
4. 用户在Boying中查看自我信息。
5. 用户在Boying中查看自我地址。
6. 用户在博英添加收货地址。
7. 用户在Boying购买节目。
8. 用户更新Boying中的订单。
9. 用户查看Boying中相关的订单。
10. 用户在Boying中查看订单详情。

功能为管理员

1. 管理员在博英注册。
2. 管理员登录Boying。
3. 管理员在博英查看客户列表。
4. 管理员在博英查看客户的个人信息。
5. 管理员在博英查看列表。
6. 管理员在Boying中查看显示的详细信息。
7. 管理员在博英查看订单列表。
8. 管理员在博英查看订单的详细信息。
9. 管理员查看博英客户报表。
10. 管理员查看博英演出的报表。
11. 管理员在博英查看订单报表。
12. 管理员在Boying中添加节目。
13. 管理员更新显示在博英。
14. 管理员删除Boying中的show。
15. 管理员冻结了博英的顾客。
16. 管理员在博英恢复客户。
17. 管理员在Boying中修改订单。
18. 管理员删除Boying中的订单。

功能分析：

1. 哪些电视剧第二季度比第一季度卖得好？
2. 哪类节目最赚钱？
3. 从下订单到发货的平均时间是多少？
3. 不同出版商发行的节目在盈利能力上有什么显著差异吗？

4. 资源要求

硬件资源：笔记本电脑、服务器

软件资源：MySQL等数据库、Idea等代码编写平台

人力资源：5名软件工程专业学生及一名指导老师

5. 项目预算

5.1 开发工作量

前台系统（7）：

- 登录界面
- 注册界面
- 首页
- 搜索演出页面
- 演出详情页面
- 个人信息页面
- 订单页面

后台系统（11）：

- 登录页面
- 演出管理页面
- 添加演出页面
- 目录管理页面
- 订单管理页面
- 活动管理页面
- 用户管理页面
- 角色管理
- 菜单管理
- 资源管理页面
- 报表管理页面

形成的文档：

- 项目需求文档
- 网站设计说明书
- 项目开发计划书
- 网站测试报告

5.2 开发管理费用

本项目为课程项目，因此无预算需求。

6. 风险识别

业务变化风险

由于本项目为课程项目，业务需求较为稳定，故业务变化风险较小。

管理风险

1. 团队成员均为学生，对人员的协调以及项目进度的把控缺乏管理经验。
2. 团队成员人数较少，在分工时需要每位团队成员承担多项职责，在项目过程中职责不明确以及职责产生冲突的风险较大。

技术风险

3. 由于本项目是以大麦网为原型的一个商业平台开发项目，而大麦网所采取的部分技术并没有公开，因此团队成员在开发的过程中需要自行解决一些商业上的技术需求。

7. 进度安排

标识号	任务名称	工期	开始时间	结束时间
1	项目启动	2	3.10	3.11
1.1	小组分工	2	3.10	3.11
2	需求分析	9	3.12	3.20
2.1	进行需求分析	2	3.12	3.13
2.2	确定需求	2	3.14	3.15
2.3	编写需求文档	5	3.16	3.20
3	系统设计	10	3.21	3.30
3.1	系统概要设计	3	3.21	3.23
3.2	系统详细设计	3	3.24	3.26
3.3	编写系统设计文档	4	3.27	3.30
4	开发、单元测试、集成测试	49	3.31	5.18
4.1	开发环境搭建	5	3.31	4.4
4.2	前台用户模块编码与测试	44	4.5	5.18
4.3	后台管理员模块编码与测试	44	4.5	5.18
5	系统测试与验收测试	14	5.19	6.1
5.1	系统测试	5	5.19	5.23
5.2	验收测试	5	5.24	5.28
5.3	编写测试文档	4	5.29	6.1
6	文档整理	7	6.2	6.8
6.1	编写用户手册	3	6.2	6.4
6.2	发布项目	2	6.5	6.6
6.3	整理文档	2	6.7	6.8