

1. 引言

- 1.1 编写目的
- 1.2 背景
- 1.3 术语表
- 1.4 参考资料

2. 项目概述

- 2.1 工作内容
 - 2.1.1 软件规模估计
 - 2.1.2 总体工作量估算
 - 2.1.3 项目软件过程与周期模型定义
 - 2.1.4 项目预算
 - 2.1.5 风险估计
 - 2.1.6 资源分析
- 2.2 主要参加人员
- 2.3 产品
 - 2.3.1 程序
 - 2.3.2 文件
 - 2.3.3 非移交的产品
 - 2.3.4 验收标准
 - 2.3.5 度量目标

3 实施计划

- 3.1 人员分组与小组职能安排
- 3.2 进度安排

4. 过程计划

- 4.1 主要实施环节
- 4.2 实施环节时间点
 - 4.2.1 WBS
 - 4.2.2 网络计划和关键路径

1. 引言

1.1 编写目的

该文档旨在制定BoYing娱乐票务平台的开发计划，合理分配项目开发阶段的人员和资源，确保按时有质量的完成BoYing项目。

1.2 背景

开发的软件系统的名称:BoYing娱乐票务平台

项目开发人员:组长: 1851632 石稼晟 组员:1854081 付诚 1850955 陈晓雯 1751022 李翠琪
1851486 方浩

目标用户:喜欢看如音乐剧、话剧、演唱会演出的用户和热衷于追星的粉丝等需要购票的群体

项目背景介绍:新一代互联网主力人群由于缺乏群体记忆，加之互联网快速分化，个性化群体快速崛起，文化需求多元化加重。现场娱乐市场作为一个多元化市场，分类众多，且各具特色。但由于行业观演环境还有待改善和培育，因此整个市场尚待焕发生机。在市场及需求的驱动下，现场娱乐在线票务平台发展迅速，比如大麦网、摩天轮票务、体育易等等。正是基于这些软件作为参考，制作一款具有特色的娱乐票务网站——博影。该项目为有现场观看音乐剧、话剧等演出需求的用户提供演出信息与购票服务，实现该软件的计算中心或计算机网络为互联网。

1.3 术语表

Boying:

博影，本项目名称，一个线上娱乐票务平台。

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。

1.4 参考资料

1. 《软件工程导论》，朱少民，清华大学出版社；
2. 《软件项目管理》，4thEdition.CHN. 2007；
3. 《Software Testing Foundations - A Study Guide for the Certified Tester Exam (4th Edition)》；
4. Vue官方文档。

2. 项目概述

2.1 工作内容

2.1.1 软件规模估计

本项目是一个娱乐票务网站平台，用户可以查看最近的音乐演出、舞台剧、展出等的门票销售情况并进行线上购买预定。该平台主要包括前台演出门票购买系统及后台管理系统。

前台系统包括：用户模块，演出模块，订单模块。

后台系统包括：权限管理，报表管理，用户管理，演出管理，目录管理，订单管理。

项目分为前台用户端系统与后台管理员端系统，工作量较大，然而团队成员有类似的项目开发经验，并且可以复用一部分代码，同时，前端与后端、用户端与管理端各一名成员，前后端、用户与管理端开发过程存在一定的并行性，以加快项目的进度。

2.1.2 总体工作量估算

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（12）：

- 登录页面
- 注册页面
- 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

2.1.3 项目软件过程与周期模型定义

过程	定义
需求分析	对用户的需求进行调查了解，定义用户的需求，撰写软件需求规格说明书；并根据软件需求规格说明书，制定测试计划，并对软件需求规格说明书和测试计划进行评审。
概要设计	根据需求分析文档，进行软件系统的总体结构、接口和相关的数据库设计，撰写软件概要设计规格说明书、制定软件开发计划，并根据软件概要设计规格说明书、软件开发计划，确定软件的规模，制定集成测试计划/系统测试计划。
详细设计	根据软件需求规格和软件概要设计进行软件详细设计。根据用户软件需求情况定义编码规范，并根据详细设计，制定单元测试计划。
编码和单元测试	编写程序代码，并根据每个单元编写情况进行单元测试；撰写单元测试报告。
集成测试	按功能分类对模块进行测试，对存在的问题进行跟踪修改，形成集成测试报告。
编制文档	根据软件需求规格说明书、软件概要设计规格说明书和软件系统撰写用户手册。
软件维护	软件使用运行期间的故障维护。

本项目采用增量模型。

增量模型融合了瀑布模型的基本成分（重复应用）和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。

与原型实现模型和其他演化方法一样，本质上是迭代的，但与原型实现不一样的是其强调每一个增量均发布一个可操作产品。早期的增量是最终产品的“可拆卸”版本，但提供了为用户服务的功能，并且为用户提供了评估的平台。

2.1.4 项目预算

本项目为课程项目，因此无预算需求。

2.1.5 风险估计

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

1. 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。

2.1.6 资源分析

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

2.2 主要参加人员

任务	姓名	学号
组长	石稼晟	1851632
编程	付诚	1854081
编程	陈晓雯	1850955
编程	李翠琪	1751022
编程	方浩	1851486

2.3 产品

2.3.1 程序

博影的前端由VUE框架制作而成，后端使用Spring Boot编写，配合使用MySQL与Redis数据库。

2.3.2 文件

- a. 软件开发项目方案书
- b. 软件需求规划书
- c. 软件设计说明书
- d. 测试报告
- e. 用户使用说明书

- f. 会议纪要
- g. 源代码
- h. 可执行程序

2.3.3 非移交的产品

软件的测试版本、中间版本、未完善版本等不必交付，开发人员的内部交流细节也不必告知用户。

2.3.4 验收标准

前端功能制作完成且界面美观，后端服务完善且稳定，程序各部分部署完成。产品部署后通过各项测试。

2.3.5 度量目标

度量目标	度量点
项目性能指标度量：通过对项目实际与计划的成本、工作量的比较，计算偏差，来评价本项目的估算能力、控制能力和预测能力	工作量估算偏差、人月成本、团队规模
过程质量度量：通过度量本项目不符合项的密度与历史基线相比较，来评价本项目的执行能力	过程质量
工作产品质量度量：通过度量评审及测试过程中产生的缺陷数，计算本项目的产品质量	提交的产品质量、质量成本
需求度量：通过需求变化率和需求稳定度，来评价需求管理和控制的能力	需求稳定率、需求变化率
风险度量：1、通过已识别的风险数占风险总数的百分比，来评价风险的识别能力；2、通过规避和缓解的风险数占已识别的风险数的百分比，来评价风险的控制能力。	风险识别率、风险控制力
生产率度量：1、通过度量规模和工作量，计算本项目的生产率；2、与组织当前的生产率基线比较，来评价本项目的生产率。	生产率
资产累计度量	文档累积率、数据累积率

3 实施计划

3.1 人员分组与小组职能安排

组长：石稼晟

任务：

项目方案书；

组织小组讨论，记录讨论内容；

协调组员、管理项目进度

设计：方浩、李翠琪

任务：

进行项目的需求分析与系统设计；

完成系统架构设计；

完成系统的数据库设计与用户界面UI设计；

完成项目的需求分析文档与系统设计文档。

前端：方浩、付诚

任务：

根据设计文档负责项目前端代码的编写；

进行相应的单元测试；

修复测试人员反馈的BUG；

向组长反馈开发进度，根据进度调整开发计划；

负责前端的部署工作。

后端：李翠琪、陈晓雯

任务：

根据设计文档负责项目后端代码的编写；

进行相应的单元测试；

修复测试人员反馈的BUG；

向组长反馈开发进度，根据进度调整开发计划；

负责后端的部署工作。

测试：石稼晟、付诚、陈晓雯

任务：

根据需求文档与设计文档设计测试用例，编写项目测试文档；

进行系统测试；

向开发人员反馈BUG；

向组长反馈项目进度，根据进度调整开发计划；

编写项目测试报告与用户手册。

3.2 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

4. 过程计划

4.1 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。

4.2 实施环节时间点

4.2.1 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。

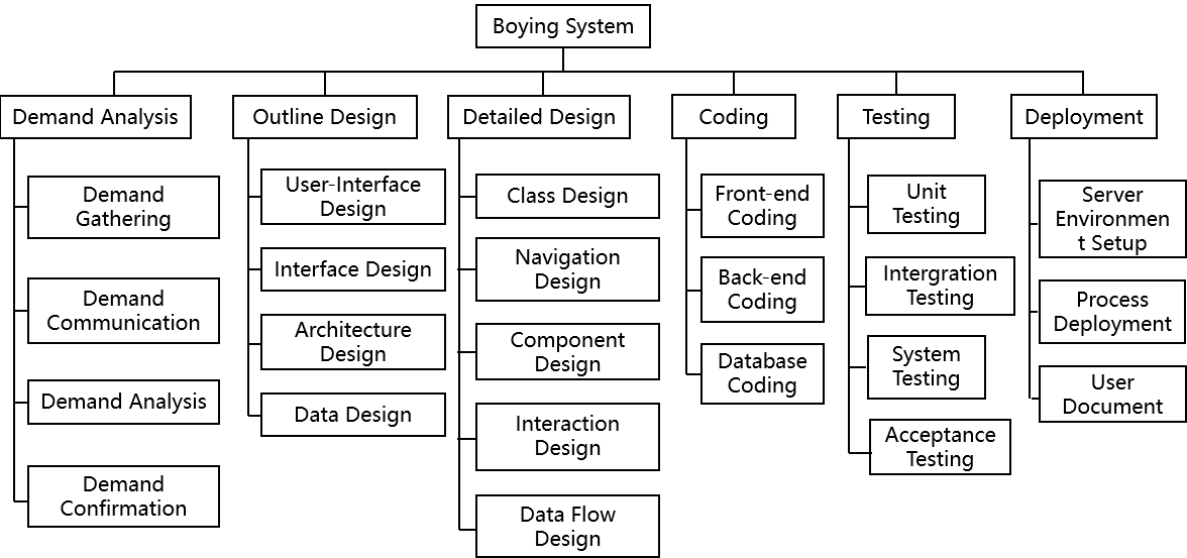
在大纲设计阶段，我们主要完成了界面设计、界面设计、架构设计、数据库设计等内容。

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。



4.2.2 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**:

