**“未来之星”一对一家教平台**

**软件开发计划**

**Version：3.0**

|  |  |  |
| --- | --- | --- |
| **项目组成员信息** | | |
| **小组名称** | **Dream of Future Star（Team 28）** | |
| **学号** | **姓名** | **本文档中主要承担的工作内容** |
| 14061055 | 朱天晨 | 编写文档并修改（60%） |
| 14061039 | 康嘉玮 | 审核文档并修改（20%） |
| 14061057 | 曹亚斌 | 修改文档（10%） |
| 14061042 | 卜令军 | 修改文档（10%） |
|  |  |  |
|  |  |  |
|  |  |  |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.0 | 2016.10.16 | 朱天晨 | 康嘉玮 | 最初版本 |
| V2.0 | 2016.10.23 | 朱天晨 | 康嘉玮 | 第二次提交版，修正软件开发过程和时间安排 |
| V3.0 | 2016.12.10 | 朱天晨 | 康嘉玮 | 第三次提交版，修改时间安排以及风险管理的部分内容 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 引言 7](#_Toc464384588)

[1.1 标识 7](#_Toc464384589)

[1.2 系统概述 7](#_Toc464384590)

[1.3 文档概述 8](#_Toc464384591)

[1.4 与其他计划之间的关系 9](#_Toc464384592)

[1.5 基线 9](#_Toc464384593)

[2 引用文件 9](#_Toc464384594)

[3 交付产品 9](#_Toc464384595)

[3.1 程序 9](#_Toc464384596)

[3.2 文档 10](#_Toc464384597)

[3.3 服务 11](#_Toc464384598)

[3.4 非移交产品 11](#_Toc464384599)

[3.5 验收标准 11](#_Toc464384600)

[3.6 最后交付期限 13](#_Toc464384601)

[4 所需工作概述 13](#_Toc464384602)

[4.1 对开发系统的需求与约束 13](#_Toc464384603)

[4.1.1 硬件环境 13](#_Toc464384604)

[4.1.2 软件环境 13](#_Toc464384605)

[4.2 对项目文档编制的需求与约束 13](#_Toc464384606)

[4.3 项目在系统生命周期中所处地位 13](#_Toc464384607)

[4.4 技术需求 14](#_Toc464384608)

[4.5 项目进度安排以及对资源的需求与约束 14](#_Toc464384609)

[4.6 其他需求与约束 14](#_Toc464384610)

[5 实施整个软件开发活动的计划 14](#_Toc464384611)

[5.1 软件开发过程 14](#_Toc464384612)

[5.2 软件开发总体计划 15](#_Toc464384613)

[5.2.1 软件开发方法 15](#_Toc464384614)

[5.2.2 软件产品标准 15](#_Toc464384615)

[5.2.3 可重用的软件产品 15](#_Toc464384616)

[5.2.4 处理关键性需求 16](#_Toc464384617)

[5.2.5 计算机硬件资源利用 16](#_Toc464384618)

[5.2.6 记录原理 17](#_Toc464384619)

[5.2.7 需方评审途径 17](#_Toc464384620)

[6 实施详细软件开发活动的计划 17](#_Toc464384621)

[6.1 项目计划和监督 17](#_Toc464384622)

[6.1.1 软件开发计划 17](#_Toc464384623)

[6.1.2 CSCI测试计划 18](#_Toc464384624)

[6.1.3 系统测试计划 18](#_Toc464384625)

[6.1.4 软件安装计划 19](#_Toc464384626)

[6.1.5 软件移交计划 19](#_Toc464384627)

[6.1.6 跟踪和更新计划 20](#_Toc464384628)

[6.2 建立软件开发环境 20](#_Toc464384629)

[6.2.1 软件工程环境 20](#_Toc464384630)

[6.2.2 软件测试环境 20](#_Toc464384631)

[6.2.3 软件开发库 21](#_Toc464384632)

[6.2.4 软件开发文档 21](#_Toc464384633)

[6.2.5 非交付软件 21](#_Toc464384634)

[6.3 系统需求分析 21](#_Toc464384635)

[6.3.1 用户输入分析 21](#_Toc464384636)

[6.3.2 运行概念 22](#_Toc464384637)

[6.3.3 系统需求 22](#_Toc464384638)

[6.4 系统设计 22](#_Toc464384639)

[6.4.1 系统级设计决策 22](#_Toc464384640)

[6.4.2 系统体系结构设计 23](#_Toc464384641)

[6.5 软件需求分析 23](#_Toc464384642)

[6.6 软件设计 24](#_Toc464384643)

[6.6.1 CSCI级设计决策 24](#_Toc464384644)

[6.6.2 CSCI体系结构设计 25](#_Toc464384645)

[6.6.3 CSCI详细设计 25](#_Toc464384646)

[6.7 软件实现和配置项测试 26](#_Toc464384647)

[6.7.1 软件实现 26](#_Toc464384648)

[6.7.2 配置项测试准备 26](#_Toc464384649)

[6.7.3 配置项测试执行 27](#_Toc464384650)

[6.7.4 测试和再测试 27](#_Toc464384651)

[6.7.5 配置项测试结果分析与记录 27](#_Toc464384652)

[6.8 配置项集成和测试 27](#_Toc464384653)

[6.8.1 配置项集成和测试准备 27](#_Toc464384654)

[6.8.2 配置项集成和测试执行 27](#_Toc464384655)

[6.8.3 修改和再测试 28](#_Toc464384656)

[6.8.4 配置项集成和测试结果分析与记录 28](#_Toc464384657)

[6.9 软件使用准备 28](#_Toc464384658)

[6.9.1 可执行软件的准备 28](#_Toc464384659)

[6.9.2 用户现场的版本说明的准备 28](#_Toc464384660)

[6.9.3 用户手册的准备 28](#_Toc464384661)

[6.9.4 在用户现场安装 28](#_Toc464384662)

[6.10 软件移交准备 29](#_Toc464384663)

[6.10.1 可执行软件的准备 29](#_Toc464384664)

[6.10.2 源文件准备 29](#_Toc464384665)

[6.10.3 支持现场的版本说明的准备 29](#_Toc464384666)

[6.10.4 “已完成”的CSCI设计和其他的软件支持信息的准备 29](#_Toc464384667)

[6.10.5 系统设计说明的更新 29](#_Toc464384668)

[6.10.6 支持手册的准备 29](#_Toc464384669)

[6.10.7 到指定现场的移交 29](#_Toc464384670)

[6.11 软件产品评估 30](#_Toc464384671)

[6.11.1 中间阶段的和最终的软件产品评估 30](#_Toc464384672)

[6.11.2 软件产品评估记录 30](#_Toc464384673)

[6.11.3 软件产品评估的独立性 30](#_Toc464384674)

[6.12 文档编制 30](#_Toc464384675)

[6.13 其他软件开发活动 32](#_Toc464384676)

[6.13.1 风险管理 32](#_Toc464384677)

[6.13.2 软件管理指标 32](#_Toc464384678)

[6.13.3 保密性和私密性 32](#_Toc464384679)

[6.13.4 分承包方管理 33](#_Toc464384680)

[6.13.5 与软件独立验证与确认机构的接口 33](#_Toc464384681)

[6.13.6 和有关开发方的协调 33](#_Toc464384682)

[6.13.7 项目过程的改进 33](#_Toc464384683)

[6.13.8 计划中未提及的其他活动 33](#_Toc464384684)

[7 进度表和活动网络图 34](#_Toc464384685)

[7.1 进度表 34](#_Toc464384686)

[7.1.1 人员分配表 34](#_Toc464384687)

[7.1.2 项目安排表 34](#_Toc464384688)

[7.2 活动网络图 38](#_Toc464384689)

[8 项目组织和资源 39](#_Toc464384690)

[8.1 项目组织 39](#_Toc464384691)

[8.2 项目资源 40](#_Toc464384692)

[8.2.1 人力资源 40](#_Toc464384693)

[8.2.2 开发人员使用的设施 40](#_Toc464384694)

[9 培训 40](#_Toc464384695)

[9.1 项目的技术要求 40](#_Toc464384696)

[9.1.1 管理技术 40](#_Toc464384697)

[9.1.2 开发技术 41](#_Toc464384698)

[9.2 培训计划 41](#_Toc464384699)

[9.2.1 技能需求 41](#_Toc464384700)

[9.2.2 成员现状 41](#_Toc464384701)

[9.2.3 培训方案 42](#_Toc464384702)

[10 项目估算 42](#_Toc464384703)

[10.1 规模估算 42](#_Toc464384704)

[10.2 工作量估算 44](#_Toc464384705)

[10.3 成本估算 46](#_Toc464384706)

[10.4 关键计算机资源估算 47](#_Toc464384707)

[10.5 管理预留 47](#_Toc464384708)

[11 风险管理 47](#_Toc464384709)

[11.1 人员变动 47](#_Toc464384710)

[11.2 资源损毁 47](#_Toc464384711)

[11.2.1 硬件资源损毁 47](#_Toc464384712)

[11.2.2 软件资源损毁 48](#_Toc464384713)

[12 支持条件 48](#_Toc464384714)

[12.1 计算机系统支持 48](#_Toc464384715)

[12.2 需要需方承担的工作和提供的条件 48](#_Toc464384716)

[12.3 需要分包商承担的工作和提供的条件 48](#_Toc464384717)

# 1 引言

## 标识

文档标识号： TD012016001DP01

文档标题：“未来之星”一对一家教平台系统软件开发计划

版本号：3.0

## 系统概述

本系统的名称为“未来之星”一对一家教系统，系统简称为“FSTS”。

本一对一家教系统用于为学习优秀，教学能力强的大学生提供在线的家教平台，同时为需要进行一对一辅导的中小学生提供上课的平台。有意愿做兼职带家教的大学生可以在本平台上进行注册、面试，面试成功后完善个人信息即可录入相应科目的教师库。与此同时，初高中以及部分小学生（包括其家长）可以在本平台注册账号成为学生，根据教师库中的教师信息选择老师，试听完成后如果满意，可以按照课时缴纳费用并成为该老师的正式学生。课程教学需要通过网站提供的视频通话系统以及课件放映系统来完成。本平台打破了传统上面对面教学的方式，使用网络教学，使得老师和学生在家就可以完成沟通和交流。同时在庞大的师生数据库中，平台可以通过标签特征的学习与筛选以及师生用户的行为特征，为每个老师推荐合适的学生，为每个学生推荐合适的老师。

本系统为初次开发，投资方为北京航空航天大学计算机学院，需求方为在校高校大学生以及全国各地的中小学生，当前计划系统在网站平台上运行。

## 文档概述

在软件的开发周期当中，需要开发组成员之间的合作交流与讨论，并且需要记录下整个开发的流程与遇见的问题以方便日后的查阅与解决，因而在整个开发周期中需要每位开发人员定期撰写开发文档，项目负责人负责每日例会以及例会报告的撰写。对于开发文档有以下几点要求：

1. 在实际开发阶段要求每位开发人员每天针对自己每日工作的完成情况撰写开发文档，内容包括完成的过程，需要注意的事项，已经解决的问题以及未解决的问题或可能出现的问题；
2. 项目负责人每日召开例会了解每个人的完成情况，汇总工程的完成进度，讨论当前的情况，集体解决尚未解决的问题并互相交流经验。会后项目负责人需要撰写会议的记录报告；
3. 我们将开发过程分为两个大题阶段以及若干个小阶段，每一阶段过后需要集体撰写每一阶段的总结报告。

撰写开发文档有以下几点目的：

1. 记录开发的过程，最重要的是记录做某项决定时的原因，例如使用某工具的原因，或是使用了某个框架结构，在某种情况下编写测试程序，某种情况下不需要测试，因而可以避免走已经被否定的道路，避免浪费时间精力；
2. 提供了自动化过程的实现可能，记录下一定的思考路线可以使得开发过程在一定程度上趋于同质化，节省了开发时间，缩短了开发周期，有利于更深层次的思考；
3. 更加有利于团队之间的协作与交流，并且避免重要的事情被忘记；
4. 有利于监督各成员的完成进度，督促项目组成员按时完成规定的任务。

## 与其他计划之间的关系

当本软件平台上线以后，可以为大学生提供良好的兼职工作平台，可以进一步开展发展大学生兼职平台的相关计划。

## 基线

由于该产品是初次开发，尚不存在前一个稳定版本，因而产品的设计不基于上一稳定版本的产品，而是基于客户的需求，客户要求能够实现教师或学生的注册，删除，查询以及信息录入等数据库的数据操作功能，实现学生在线对教师进行预约选择功能，实现上传PPT并进行在线视频授课功能，在完善阶段需要实现根据用户行为和用户标签将老师和学生进行配对，并根据结果进行推送的功能，项目将根据客户的这些要求来完成。

# 引用文件

[1] 北京航空航天大学 计算机学院SDP-开发计划文档模板 电子版 2016年10月

[2] 雨做的云 PHP业务逻辑层和数据访问层设计 电子版 2013年8月

# 交付产品

## 程序

1. 项目名称：“未来之星”一对一教学平台
2. 所用语言：PHP+MySQL
3. 源码发布位置：GitHub
4. 软件平台功能：

账号注册（包括教师和学生）

发布视频以及课件

录入教师和学生的详细信息，追踪用户行为

在线视频授课

根据一定条件推送合适的教师或学生

提供论坛功能，可以发布博客以及留言问题

提供交易平台（上线后实现）

## 文档

1. 《软件开发计划》：

描述开发者实施软件开发工作的计划，涵盖了新开发、修改、重用、再工程、维护和由软件产品引起的其他所有活动。向需求方提供了解和监督软件开发过程、所使用的方法、每项活动的途径、项目的安排、组织及资源的一种手段。

1. 《软件需求规格说明》：

描述对计算机软件配置项CSCI的需求，及确保每个要求得以满足的所使用的方法。

1. 《软件设计说明》：

描述了计算机软件配置项(CSCI的设计）。它描述了CSCI级设计决策、CSCI体系结构设计(概要设计)和实现该软件所需的详细设计。向需方提供了设计的可视性，为软件支持提供了所需要的信息。

1. 《软件测试说明》：

描述执行计算机软件配置项CSCI,系统或子系统合格性测试所用到的测试准备、测试用例及测试过程。需求方通过此能够评估所执行的合格性测试是否充分。

## 服务

本软件平台为客户提供如下服务：

1. 免费问题咨询：

在平台使用过程中遇到的任何问题都可以免费咨询开发人员；

1. 平台维护：

在平台上线期间对平台进行维护，解决期间可能出现的一系列问题，定期维护后台代码；

1. 平台升级：

必要时改进平台部分算法，进行升级，同时不断增添新的功能，完善平台早期出现的一些漏洞；

1. 信息保管：

平台对用户的信息进行加密处理并防止信息泄露，不断增强网站平台的抗攻击性。

## 非移交产品

1. 程序源代码：

软件开发过程中产生的所有源码，包括测试代码以及注释。

1. 后台数据库：

软件开发中用到的所有数据以及平台发布后收集的所有数据。

1. 《市场调研报告》：

软件开发计划阶段针对市场的定点调查报告以及市场分析报告。

1. 《需求分析报告》：

根据针对市场调研报告确立的需求分析所撰写的需求分析报告。

1. 开发人员开发过程中的各项开发文档。

## 验收标准

1. 由开发组与客户共同进行项目的验收工作，在软件需求评审阶段，仔细审阅软件需求规格说明书，指出不利于测试以及可能存在歧义的描述；在开发组完成开发后由第三方对项目进行测评，提供错误报告给客户方；
2. 验收过程需要进行以下几项测试：

功能项的测试：

对软件需求规格说明书中的所有功能项进行测试。

业务流程的测试：

对软件项目的典型业务流程进行测试。

平台的容错性测试：

包括平台对常见误操作是否进行提示，对重要数据的录入与删减是否有提示以及是否能判断数据的有效性，屏蔽用户的无效输入，识别非法输入并给出提示。

性能测试：

对软件需求规格说明书中明确的软件性能进行测试，保证软件平台满足规格说明书中的各项性能指标。

易用性测试：

测试网站页面是否对用户友好，提示信息是否清晰可见，避免中英文混杂的情况，保证各界面风格的一致性，以及各种查询的输出的直观性。

适应性测试：

参照使用环境以及需求规格说明书中的规定，在各种环境下对软件平台进行测试。

文档测试：

检查所有的用户文档是否齐全，内容及格式是否正确，表述是否清晰无误，保证用户文档便于使用者的理解。

1. 验收结果需保证测试用例的不通过数比例小于等于1.5%；
2. 验收结果需保证不存在不能执行正常功能或者危害安全的错误；
3. 验收结果需保证不存在严重影响系统功能的实现且无法解决的错误；
4. 验收结果要求严重影响系统功能但有合理解决办法的错误不超过5个；
5. 要求所有提交的错误在产品发布前都已经得到改正。

## 最后交付期限

2017年1月24日。

# 所需工作概述

## 对开发系统的需求与约束

### 硬件环境

服务器：工作站（租赁）

网络设备：网络交换机，网卡，网线

### 软件环境

服务器端环境：Windows 7 及以上

数据库软件：MySQL

开发工具：Eclipse

浏览器支持：IE（7.0及以上），Chrome

## 对项目文档编制的需求与约束

项目文档的编制至少包括但不限于软件开发计划、软件规格需求说明、软件结构设计说明以及软件测试说明，在开发过程中要求开发人员定期撰写开发文档以及开发会议记录，如有必要可以增加编写数据需求说明以及数据库顶层设计说明和软件测试报告。

## 项目在系统生命周期中所处地位

该项目是系统应用层的一个网站与数据库相结合的系统，是系统生命周期中重要的软件系统。

## 技术需求

面向对象编程技术

数据库开发技术

HTML、JavaScript、Java、PHP等前端和后端技术。

## 项目进度安排以及对资源的需求与约束

项目计划开展两个阶段，分为I阶段和II阶段，每个阶段又细分为三个小阶段，项目计划与十一月中旬完成I阶段，即开发阶段，在十一月下旬完成项目的第一次上线，并在项目结束之前完成II阶段的所有工作，包括上线后的各项测试以及版本的更新维护。

项目的上线需要租赁较大的服务器，暂定为租赁工作站。

## 其他需求与约束

需要学院的部分资金支持或是服务器支持；

需要课程理论支持；

需要大量的用户支持以及测试结果反馈支持。

# 实施整个软件开发活动的计划

## 软件开发过程

软件的开发过程基于敏捷开发模型。基于本项目的规模以及开发难度，可将项目分为两个迭代阶段I阶段和II阶段，具体的划分如下所示：

1. 开发阶段（I阶段）：

计划调研学习阶段

代码编写阶段

初步测试阶段

1. 初步上线阶段（II阶段）：

上线宣传阶段

用户测试阶段

版本更新阶段

在这里需要注意的是以下几点：

项目开发初期的需求分析不需要过于详细，只需要分析好基本的需求关系即可，因为基于敏捷开发模型的话，团队需要保证对于需求变化的响应是动态的并且及时的，这样可以保证最大程度的提高效率，避免由于需求变化导致I阶段的时间浪费。

建立系统化分析方法，对产品进行SWOT分析，随时适应市场的变化。

在I阶段这一迭代周期完成之时即需要交付可运行的产品，换言之可实现基本功能的网站系统此时即应上线。并且在II阶段初期开展大规模的用户反馈工作，将反馈结果作为第II阶段进一步开发的依据。

在开发过程的两个阶段内也要进行迭代，同时保证每次迭代时版本迁移都能比较平滑，杜绝大幅度的改动以及架构改动，考虑版本的兼容性。

在编码之前先编写测试代码，作为设计和验证，但是测试阶段还是放在I阶段最后。

撰写轻量级的文档，，轻文档不等于无文档。基于敏捷开发，我们不需要撰写过多文档，但是最基本的设计文档与开发文档还是必要的。

注重团队内部的沟通以及开发人员与客户之间的沟通，保证产品的质量。

## 软件开发总体计划

### 软件开发方法

软件开发采用敏捷开发方法，同时应用面向对象的开发原则，保证项目的顺利交付以及项目的模块化。

### 软件产品标准

软件开发阶段需要做到在实现全部要求的功能的同时保证界面对于用户的友好性，设计思维做到面向对象，代码要求可读性强，代码风格统一，每一个模块都要有代码注释，注释总量应占代码总量的30%以上。

软件测试阶段应保证测试用例覆盖所有的代码，尽可能考虑到每一种可能出现的情况，除了逻辑测试外还应该包括压力测试以及安全性测试，防止上线后带来服务器崩溃等问题。

### 可重用的软件产品

#### 吸纳可重用的软件产品

本软件系统不引用任何其他可重用软件产品的代码。

#### 开发可重用的软件产品

本软件系统在开发过程中应留下补充开发的接口，便于后期的增添功能，维护以及其他产品对本产品的借用。

### 处理关键性需求

#### 安全性保证

本平台后端采用MySQL数据库，保密性基于MySQL数据库的保密原理，同时对网站的前端进行加密，防止数据在传输时产生泄漏或被盗取。

#### 保密性保证

产品所获取到的用户信息仅作为产品内部的功能之用，不用于其他商业途径。

#### 私密性保证

用户的私人数据全部存储在服务器中，维护服务器的开发人员在没有权限的情况下不会随意获取到用户的个人信息。

#### 其他关键性需求保证

维护人员每天都会对服务器上的所有信息进行备份处理，即使意外断电也能使数据恢复，对用户不会造成任何损失。

### 计算机硬件资源利用

软件开发期间需要用到六台左右计算机进行代码的编写，测试阶段和上线阶段需要用到一个较大的服务器，资源分配的方式采取重要度优先的方式，设备使用时要避免机械性的破坏。

### 记录原理

在做出关键决策时需要优先考虑系统功能的完整性，其次考虑运行效率的最优化，最后考虑成本降为最低，在做出决策以后详细记录作出该决策的原因，并且在面对类似问题时有根据地做出相同的决定。

### 需方评审途径

评审时可以通过网站的访问量，注册数量以及在线人数峰值来进行评审，同时老师和学生的数量也可以作为评判依据，也可以使用测试用例对平台进行测试，但需要按照测试文档中规定的条件进行测试。

# 实施详细软件开发活动的计划

## 项目计划和监督

### 软件开发计划

软件开发从2016年10月3日开始，到2017年1月24日结束，共计82个工作日，时间安排如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 大阶段 | 小阶段 | 开始时间 | 结束时间 | 时长 | 所需资源 |
| 开发阶段（I阶段） | 计划调研学习阶段 | 2016.10.3 | 2016.10.11 | 7天 | 问卷及资金 |
| 代码编写阶段 | 2016.10.12 | 2016.11.14 | 25天 | 开发环境以及开发技术与资料 |
| 初步测试阶段 | 2016.11.16 | 2016.12.6 | 15天 | 数据库与服务器以及测试数据 |
| 初步上线阶段（II阶段） | 上线宣传阶段 | 2016.12.7 | 2016.12.14 | 6天 | 资金以及宣传单 |
| 用户测试阶段 | 2016.12.15 | 2016.12.27 | 9天 | 服务器 |
| 版本更新阶段 | 2016.12.28 | 2017.1.24 | 20天 | 开发环境与开发技术 |

表 6 – 1 软件开发计划表

### CSCI测试计划

软件配置项（CSCI）测试包含了功能测试、性能测试、外部接口测试、余量测试以及边界测试在内的多项测试，除此之外，如果有需要，还应进行人机交互界面测试、强度测试、可靠性测试、安全性测试、恢复性测试、安装性测试、互操作性测试、敏感性测试等多项测试。CSCI测试计划如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 测试类别 | 开始时间 | 结束时间 | 持续时间 |
| 功能测试 | 2016.11.16 | 2016.11.18 | 4天 |
| 性能测试 | 2016.11.19 | 2016.11.22 | 4天 |
| 外部接口测试 | 2016.11.23 | 2016.11.25 | 2天 |
| 余量测试 | 2016.11.26 | 2016.11.27 | 2天 |
| 边界测试 | 2016.11.28 | 2016.11.29 | 2天 |
| 人机交互界面测试 | 2016.11.30 | 2016.12.1 | 2天 |
| 强度测试 | 2016.12.15 | 2016.12.18 | 4天 |
| 可靠性测试 | 2016.12.19 | 2016.12.20 | 2天 |
| 安全性测试 | 2016.12.21 | 2016.12.22 | 2天 |
| 敏感性测试 | 2016.12.23 | 2016.12.24 | 2天 |

表 6 – 2 CSCI测试计划表

### 系统测试计划

系统测试计划安排表如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 测试时间 | 测试阶段 | 测试类型 | 测试目标 |
| 2016.11.22 | 单元测试 | 软件接口测试 | 保证接口调用的正确性以及参数的正确性 |
| 2016.11.23 | 单元测试 | 软件功能测试 | 保证软件的每个模块实现了规格中描述的功能并且运行的结果准确无误 |
| 2016.11.24 | 集成测试 | 功能测试 | 保证模块之间的相互调用不会带来异常结果，确保不同模块之间保持和谐 |
| 2016.11.25 | 系统测试 | 网站页面测试 | 保证页面内容与后台逻辑的和谐一致，确保不同页面风格一致且页面没有冗余和无效按钮 |
| 2016.11.26 | 系统测试 | 功能测试 | 保证整个网站的正常运作，确保各项功能能高效准确地进行 |

表 6 – 3 系统测试计划表

### 软件安装计划

本项目完成后为一个可发布的网站，发布在服务器平台上可以连接到广域网上，无需进行安装，只需进行注册即可。网站上有相应的注册教程以及各项功能的操作说明，从而确保每一位客户都能在极短的时间内掌握网站的使用流程及方法。

### 软件移交计划

项目将于2016年12月7日上线运行，同时也会在网站上发布以下内容：

1. 网站使用说明
2. 注册流程以及功能说明
3. 各项功能的操作流程
4. 网站维护说明

网站的源代码以及项目过程中的一系列说明文档不会公布在网站中，但是可以在GitHub中找到。

### 跟踪和更新计划

1. 网站发布后需要对网站的运行情况以及运行状态进行跟踪，我们计划采用对访问量的统计、注册量的统计以及获取数据库的更新频率，获取访问服务器的次数来跟踪网站的使用情况；
2. 我们项目组计划通过对用户行为特征的分析，来讨论网站对用户产生的影响；
3. 网站中设置有联系我们的入口用于用户向项目组进行反馈，通过用户的反馈我们可以跟踪整个网站系统的运行情况以及出现问题和错误的地方，便于进一步的修改和完善；
4. 我们计划每一周进行一次网站的版本更新，包括但不限于对界面风格的进一步完善以及增添部分附加功能（特别是基于大数据分析的一些功能）。

## 建立软件开发环境

### 软件工程环境

进行项目开发时需要配置以下软件环境：

1. Windows 7 及以上操作系统下的Apache Web服务器
2. MySQL数据库管理系统
3. PHP语言开发环境
4. Git版本控制
5. 其他常用语言开发环境
6. Linux操作系统环境（选用）

### 软件测试环境

软件测试时的环境与软件开发时的环境基本相同，除此之外还需要一些专用的测试工具，如：

1. 开源测试管理工具：Bugzilla、Bugfree等
2. 开源功能自动化测试工具：QualityCenter等

### 软件开发库

本网站系统的开发使用CodeIgniter Web Framework库

### 软件开发文档

本项目要求项目组成员每天需写一份开发报告用来记录每个人的工作进度以及开发过程中遇见的问题和重要的点，便于相互沟通。除此之外还有若干开发过程中的其他的重要的文档比如会议记录，开发计划，规格说明，测试文档等。本项目中所有的文档均使用Microsoft Office Word软件编写，并且通过Git软件控制版本。

### 非交付软件

开发过程中的所有源代码、测试用例、测试代码以及注释内容还有除交付文档以外的所有文档均为非交付软件。

## 系统需求分析

### 用户输入分析

1. 用户通过浏览器访问网站，浏览网页上的内容；
2. 用户在网站上在线注册，成为老师或者学生；
3. 用户在网站上登录，跳转自个人中心界面；
4. 用户在网站上进行面试；
5. 用户完善自己的资料，后台录入老师库或学生库；
6. 用户在网站上完成搜索课程，查询课程、查询老师、浏览老师等功能；
7. 用户在网站上观看课件；
8. 用户在网站上进行课程预约；
9. 用户在网站上进行在线授课或者在线听课；
10. 用户在网站上发表博客，提出问题或者回答问题，给他人留言；
11. 用户在网站上缴纳学费；

### 运行概念

本网站系统通过数据库的操作对用户从网页上发送的信息进行处理，包括录入数据库，修改数据库，查询数据信息，删除数据条目等等，后台的数据库逻辑反映到前端即为网站页面内容的修改。视频授课则是后台调用双方的摄像头并且将捕捉到的视频和音频内容传输到前端的用户页面中从而达到视频授课的效果。

### 系统需求

网站系统运行时需要的环境如下：

1. 服务器：Pentium 3以上或更高，内存4G以上； 硬盘至少80G以上；
2. 网络适配器：10M/100M自适应，一台UPS；
3. 工作站：Pentium 4以上微机；内存1G；硬盘至少80G以上；网络适配器10M/100M自适应；

## 系统设计

### 系统级设计决策

1. 系统的初始化：

系统在进行初始化的过程中，需要在服务器上开启Apache服务器，并且开启MySQL的数据库服务，系统初始化完成之后用户便可以通过网站地址对网站进行访问。

1. 系统运行时的自动控制：系统自身有一定的自动控制的能力，管理员在少数特殊情况下需要进行强制人工控制，例如管理员对于注册用户的权限的更改，以及对于网站部分页面的暂时关闭
2. 系统结束运行：当一定时间段内系统的访问量过大或者所开线程数过多时，此时服务器即将达到承受的限度，系统会及时做出反应并立即关闭服务器，从而保护服务器不受损坏并且有效防止数据的遗失。另外系统升级时，服务器维护人员需要手动关闭服务器并结束数据库等服务。

### 系统体系结构设计

浏览器交互

后端逻辑

操作

图 6 – 1 系统体系结构设计图

## 软件需求分析

1. 在特定环境下的用户交互：

用户通过网站页面上的操作实现注册、登录、查询、完善个人信息、预约课程、在线授课或在线上课、留言、发博客等操作，在一定条件下还可以进行支付操作。当用户通过浏览器向服务器发出请求后，服务器对请求做出相应的处理并向用户发送信息流，这些信息通过浏览器的解析呈现在用户面前的就是界面的转换或是界面内容的改动。

1. 系统处理对象：

系统处理的对象是用户通过浏览器发送的请求对象或者管理员直接向系统发出的请求对象。

1. 系统执行功能：

系统提供数据库的增添、删减、查找等数据库操作功能，同时也提供对用户行为的收集，对标签的分类功能从而实现不同用户间的匹配，用于推送的功能。另外系统还可以实现将不同的接口进行交互的功能，系统可以通过调用其他接口获取对象，并将处理后的对象作为另一个接口的输入，以实现实时交互的功能。

1. 定义接口：

用户接口，管理员接口，前端接口，数据库接口，系统摄像头接口，支付接口，文字处理接口，图像处理接口，用户匹配接口等

1. 约束条件：

每一个用户都有各自权限的限制，用户不能越级进行操作。

## 软件设计

### CSCI级设计决策

1. CSCI应当接受用户的注册、登录、查询、预约、以及上课请求，并相应地返回页面的变化以及修改数据库或修改相应的接口。
2. CSCI在响应用户或是管理员发出的请求时，应当尽可能的提高响应的效率，这就决定了数据的存储方式应当尽可能高效。因而我们需要优化相应的算法使得数据库中的元祖保持唯一性（使用哈希）。CSCI响应请求时也要检查请求的合法性以及可实现性。
3. 在网页前端对数据信息进行加密，同时在后台程序也使用加密算法对数据进行存取，从而保证了用户信息的安全性。服务器端会将用户信息定时备份，防止因为硬件原因使信息发生丢失。
4. 系统在开发过程中留下一些接口，目的是便于上线后对系统进行升级，添加功能。另一方面也为了便于与其他系统接口进行集成。

### CSCI体系结构设计

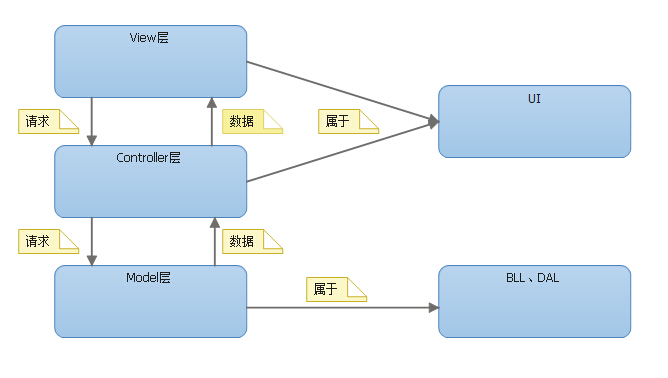


图 6 – 2 CSCI体系结构设计图

### CSCI详细设计

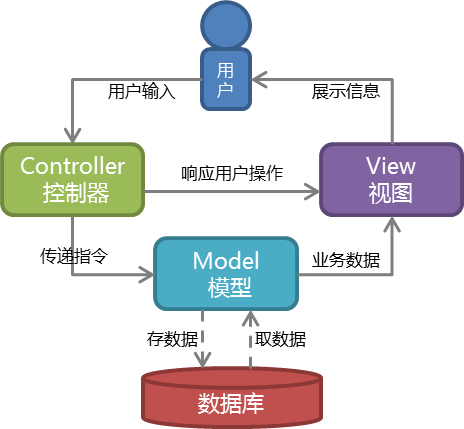


图 6 – 3 CSCI详细设计图

1. 模型

视图模型用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。“模型”有对数据直接访问的权力，例如对数据库的访问。“模型”不依赖“视图”和“控制器”，也就是说，模型不关心它会被如何显示或是如何被操作。

1. 视图

能够实现数据有目的的显示。在视图中一般没有程序上的逻辑。

1. 控制器

控制器起到不同层面间的组织作用，用于控制应用程序的流程。它处理事件并作出响应。“事件”包括用户的行为和数据模型上的改变。模式中的每个部分是分工明确、界定清晰的。操作数据的逻辑只存在于模型中，呈现数据的逻辑只存在于视图中，处理用户的请求和输入只存在于控制器中。

## 软件实现和配置项测试

### 软件实现

软件开发的最后阶段，将各个模块使用一定的逻辑连接起来，将网站前端界面与后端的逻辑运算相对应，并且将数据库操作嵌入到后端逻辑当中，即可完成项目系统的开发。

### 配置项测试准备

在测试阶段开始之前，需要对每个配置项进行规格分析，并且按照其规格说明设计测试数据和测试用例。与此同时编写测试说明，记录测试的要点，说明哪些模块应当重点测试，哪些模块不用测试。

### 配置项测试执行

配置项测试时按照预先设定的测试说明进行测试。

### 测试和再测试

配置项测试第一遍完成后需要反馈出错的信息，根据出错信息对代码进行修改，直到错误信息消除。直到所有的出错信息消除后再次进行测试（包括原测试数据与新测试数据）。

### 配置项测试结果分析与记录

配置项测试的结果需记录在软件测试报告当中，记录的内容包括测试数据，测试现象以及出错位置和出错的原因，并应当将改正前后进行对比说明。

## 配置项集成和测试

### 配置项集成和测试准备

配置项集成测试通过测试不同模块之间的协作或相互调用来确保配置项之间的集成是不影响各自的功能的。测试之前需要准备测试数据与测试要点，需要对各个配置项分组进行测试。

### 配置项集成和测试执行

配置项集成测试时需要观察不同配置项各自的行为以及交互时带来的影响，即便测试结果与理论上相同，也应当观察各个模块内部的值，确保集成后的每一部分都是完全正确的。

### 修改和再测试

配置项集成测试第一遍完成后需要反馈出错的信息，根据出错信息对代码进行修改，直到错误信息消除。直到所有的出错信息消除后再次进行测试（包括原测试数据与新测试数据）。

### 配置项集成和测试结果分析与记录

配置项集成测试的结果需记录在软件测试报告当中，记录的内容包括测试数据，测试现象以及出错位置和出错的原因，还有集成模块之间的关系，数据行为等，并应当将改正前后进行对比说明。

## 软件使用准备

### 可执行软件的准备

由于所开发项目是一个网站，因而不需要可执行文件的生成。

### 用户现场的版本说明的准备

将软件的代码复制到服务器上运行即可。项目源文件可在GitHub上查看。

### 用户手册的准备

用户手册在项目上线之前即已编写完毕，上线时将用户手册内容发布在网页中即可。

### 在用户现场安装

由于本项目是网站项目，不需要进行安装工作，只需要在服务器上运行即可。

## 软件移交准备

### 可执行软件的准备

由于所开发项目是一个网站，因而不需要可执行文件的生成。

### 源文件准备

将软件的代码复制到服务器上运行即可。项目源文件可在GitHub上查看。

### 支持现场的版本说明的准备

在软件开发过程中会记录软件的版本信息，每一个版本都会有相应的版本说明，内容包括项目的运行环境，功能以及使用方法等。

### “已完成”的CSCI设计和其他的软件支持信息的准备

相关内容会记录在用户手册当中。

### 系统设计说明的更新

项目系统在发布后会定期进行升级更新，目的是增添功能并且优化算法和界面，同时能够修补已发现的漏洞。

### 支持手册的准备

包含用户指南，维护手册在内的多项手册。

### 到指定现场的移交

由于所开发项目是网站，因而不需要到指定现场进行移交工作。

## 软件产品评估

### 中间阶段的和最终的软件产品评估

在项目开发的第一阶段完成后，会进行第一次产品评估，目的是评价项目的完成情况，同时给出下一阶段的开发路线，此次评估的结果将作为网站是否能上线的直接参考。产品开发第二阶段完成后将进行最终的产品评估工作，此次评估主要评估线上测试时产品的反应情况，目的是为产品的发展前景做出准确评估。

### 软件产品评估记录

在软件开发阶段，每一次的产品评估都需要记录产品评估的具体情况，并对评估的结果进行总结和分析。

### 软件产品评估的独立性

产品评估独立于开发和测试工作，项目最终的评估将不由项目组自己来完成，产品评估相对独立。

## 文档编制

1. 软件开发计划文档

在市场调研完成后，撰写软件可行性分析报告，同时撰写软件开发文档，文档中需要描述开发者实施软件开发工作的计划，并向需求方提供了解和监督软件开发过程以及所使用的方法、每项活动的途径、项目的安排、组织和资源等内容。该文档由项目组长撰写。

1. 软件需求规格说明

软件需求规格说明描述了对计算机软件配置项CSCI的要需求，并且确保在该架构下的项目能使每个要求都满足。软件需求规格说明在开发计划文档之后由项目组长确定项目的架构，并撰写软件需求规格说明。

1. 软件结构设计说明

该文档描述了计算机软件配置项的设计。文档由项目组长和开发人员在需求规格说明之后协作撰写。

1. 软件测试说明

软件测试说明文档主要用于软件开发阶段完成之后对软件测试要点的规定，文档中主要列举测试的项目、方法以及测试的目的。该文档由项目组全体在测试开始之前共同完成。

1. 数据需求说明

该文档主要用于向整个开发时期提供关于被处理数据的描述和数据采集要求的技术信息，该文档由项目组共同撰写。

1. 数据库顶层设计报告

该文档记录数据库的设计情况，该文档向需方提供了设计的可视性，为软件支持提供了所需要的信息，是实现数据库及相关软件配置项的基础。该文档由项目组长在设计完软件架构之后负责撰写。

1. 软件测试报告

软件测试报告主要用于对计算机软件配置项、软件系统或子系统或与软件相关项目执行合格性测试的记录，该记录可作为评估所执行的合格性测试以及其测试结果的参考。该报告由组内测试人员在第一次测试完毕后撰写。

1. 开发文档

项目组要求组内每一位开发人员每天需要做一份开发报告，用于组内研讨交流以及记录关键要点。

1. 会议记录

项目组每次开完会后由项目组长对会议内容进行总结记录，撰写会议记录报告。

以上文档只有前四个文档是最终移交文档，其余所有文档都是用于软件的开发过程中的记录分析以及互相总结交流之用，在项目结束时由项目组自行保留，不会进行移交工作。

## 其他软件开发活动

### 风险管理

在项目开发之前，项目组进行可行性分析的时候对项目开发期间可能出现的风险做出相应的预估，并准备对应的解决预案，内容记录在可行性分析中；如若在项目开发过程中出现不可预测的风险，则项目组召开紧急会议，尽量保证在不耽误项目开发的完成情况下规避风险。

在本项目中，我们会尽量做到消除根源，尽可能评估出所有可能发生的风险，然后判定每一个风险出现的概率、产生的影响及其重要性，并通过据此排出一个风险优先级。根据风险优先级，对风险制定不同的风险管理计划，确保各个单独的风险管理计划之间以及它们与相互计划之间的一致性。

在项目开发过程中，执行之前制定的风险管理计划，以缓解或消除可能会发生的风险。同时还要进行风险监控，在化解风险的过程中可能会出现新的风险，则制定相应的新的风险管理计划，规避或缓解新出现的风险。

### 软件管理指标

本项目采用由项目组长统一管理，项目组成员参与项目的开发，项目组长控制项目的进度与流程。项目组内定期开展组会讨论项目完成情况，共享经验，合作解决问题。

### 保密性和私密性

本项目的源代码公布在GitHub中，但是项目的版权归项目组自身所有。同时网站运营时的安全性也由项目组本身保证。用户在网站上的信息的安全性由加密算法以及数据库的加密保证，管理员在后台通过权限控制保证用户只能看到自身的重要信息。用户的信息一般由服务器保管，管理员不会泄露用户的信息。

### 分承包方管理

本项目不存在分承包方。

### 与软件独立验证与确认机构的接口

本项目需要提供相应的接口来保证项目本身处于可被验证状态。

### 和有关开发方的协调

本项目的开发方只有本项目组，因而项目组内的协调可以通过每次开会时的沟通交流，或者开发人员撰写的开发报告来进行。

### 项目过程的改进

在项目开发过程中，如果项目的进度流程出现问题，应当由项目负责人重新制订更加合适的流程，但需要保证项目能够按时交付。项目在第一阶段完成之后有一次固定的升级行为，在上线之后可以根据用户情况进行多次的升级与改进。

### 计划中未提及的其他活动

项目上线测试完成后，应当进一步增大推广力度，提高网站的知名度，增加网站访问量。

# 进度表和活动网络图

## 进度表

### 人员分配表

|  |  |  |
| --- | --- | --- |
| 任务分解 | 人员分工 | 具体人员 |
| 市场调研、需求分析 | 项目组全体成员 | 项目组全体成员 |
| 软件架构 | 项目组长 | 朱天晨 |
| 前端开发 | 前端开发人员 | 曹亚斌 |
| 后端逻辑 | 后端开发人员 | 康嘉玮 |
| 软件测试 | 软件测试人员 | 卜令军 |
| 文档撰写 | 项目组全体成员 | 项目组全体成员 |
| 上线宣传 | 项目组全体成员 | 项目组全体成员 |
| 收集反馈信息 | 项目组长 | 朱天晨 |
| 完善功能，修改漏洞，维护代码 | 后端开发人员 | 康嘉玮 |

表 7 – 1 人员分配表

### 项目安排表

本项目从2016年10月3日开始，到2017年1月24日结束，项目周期共82天，具体的时间安排如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 大阶段 | 小阶段 |  | 开始时间 | 结束时间 | 时长 |
| 开发阶段（I阶段） | 计划调研学习阶段 | 市场调研、预算估计以及  可行性分析 | 2016.10.3 | 2016.10.5 | 3天 |
| 撰写软件开发计划、可行性分析报告以及软件需求规格说明 | 2016.10.6 | 2016.10.13 | 6天 |
| 代码编写阶段 |
| UML建模、设计软件架构并撰写软件结构设计说明 | 2016.10.14 | 2016.10.20 | 5天 |
| 确定功能规范 | 2016.10.21 | 2016.10.21 | 1天 |
| 定义数据库模型 | 2016.10.24 | 2016.10.25 | 2天 |
| 编写实现的技术用例 | 2016.10.26 | 2016.10.27 | 2天 |
| 实现后端数据库数据录入 | 2016.10.28 | 2016.11.1 | 3天 |
| 实现注册登录功能 | 2016.11.2 | 2016.11.3 | 2天 |
| 初步设计网站各个界面 | 2016.10.28 | 2016.11.3 | 5天 |
| 实现在线视频教学逻辑功能 | 2016.11.4 | 2016.11.9 | 4天 |
| 实现论坛、博客、留言功能 | 2016.11.10 | 2016.11.15 | 4天 |
| UI及界面优化 | 2016.11.4 | 2016.11.14 | 7天 |
| 初步测试阶段 | 规定测试要点并撰写软件测试说明 | 2016.11.16 | 2016.11.17 | 2天 |
| 开发人员手动测试 | 2016.11.18 | 2016.11.21 | 2天 |
| 单元测试 | 2016.11.22 | 2016.11.23 | 2天 |
| 模块集成测试 | 2016.11.24 | 2016.11.24 | 1天 |
| 整体测试 | 2016.11.25 | 2016.11.25 | 1天 |
| 自动化测试 | 2016.11.28 | 2016.11.28 | 1天 |
| 测试人员覆盖性测试 | 2016.11.29 | 2016.11.30 | 2天 |
| 撰写测试报告 | 2016.12.1 | 2016.12.1 | 1天 |
| 修改代码以解决测试中出现的问题并再次测试修改后的代码 | 2016.12.2 | 2016.12.6 | 3天 |
| 初步上线阶段（II阶段） | 上线宣传阶段 | 网站上线运行 | 2016.12.7 | 2016.12.7 | 1天 |
| 宣传网站吸引用户使用 | 2016.12.8 | 2016.12.14 | 5天 |
| 用户测试阶段 | 压力测试 | 2016.12.15 | 2016.12.19 | 3天 |
| 安全性测试 | 2016.12.20 | 2016.12.21 | 2天 |
| 收集测试中遇到的问题、补充测试报告并修改代码解决遇到的问题 | 2016.12.15 | 2016.12.27 | 9天 |
| 版本更新阶段 | 收集用户行为和用户标签 | 2016.12.28 | 2017.1.24 | 20天 |
| 使用机器学习实现为用户推送合适的老师或学生功能 | 2016.12.28 | 2017.1.5 | 7天 |
| 优化前一版本的算法与代码 | 2017.1.6 | 2017.1.16 | 7天 |
| 制定用户手册与使用规范 | 2017.1.17 | 2017.1.20 | 4天 |
| 审阅所有文档 | 2017.1.23 | 2017.1.24 | 2天 |

表 7 – 2 项目安排表

## 活动网络图

详细的甘特图见附件中的mpp文件

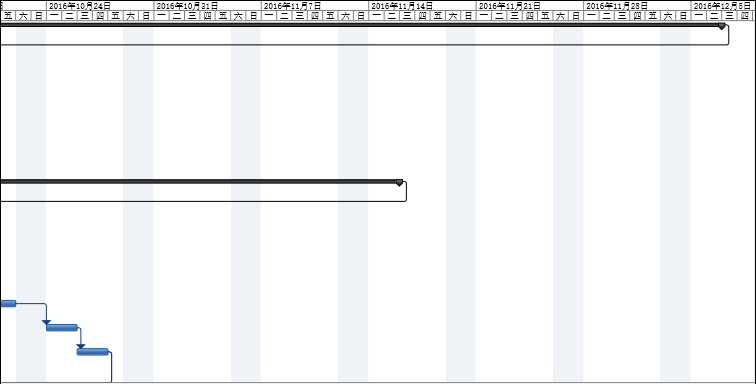
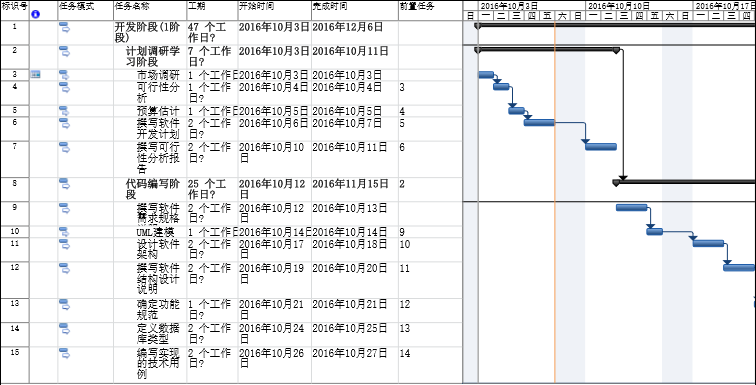


图 7 – 1 项目活动网络第一阶段图

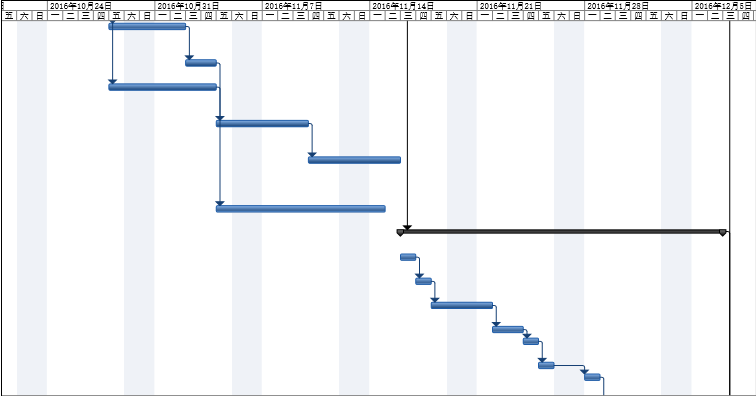
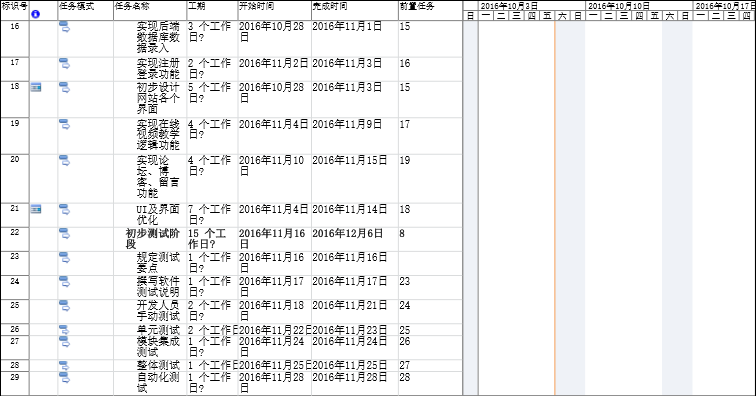


图 7 - 2 项目活动网络第一阶段续图

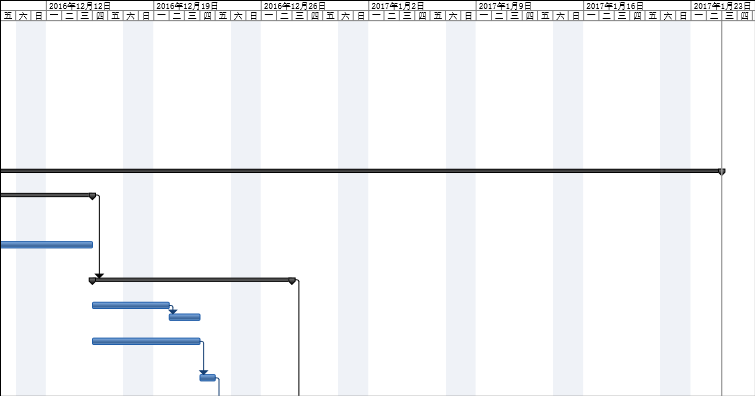
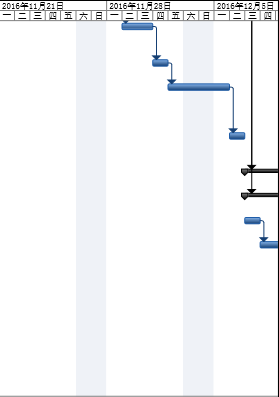
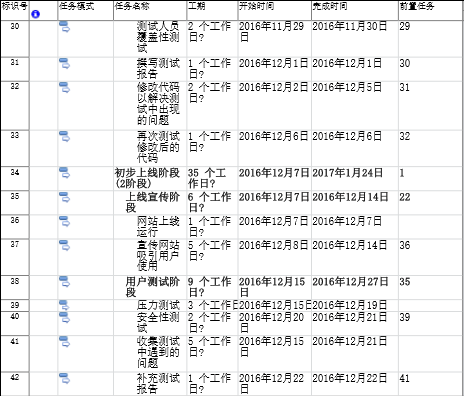


图 7 - 3 项目活动网络第二阶段图

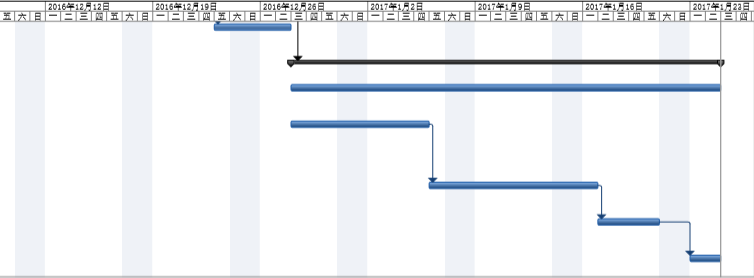


图 7 - 4 项目活动网络第二阶段续图



图 7 - 5 项目活动图例

# 项目组织和资源

## 项目组织

本项目采用类似软件项目小组的组织结构，分为产品经理，前端开发，后端开发以及测试人员，项目组的客户为北京航空航天大学计算机学院，学院能够实时了解项目的进展情况，但是不能干预项目的开发进度。开发完成的项目学院可以有条件的使用。

图 7 - 6 项目组织结构图

## 项目资源

### 人力资源

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 项目组角色 | 擅长使用语言 | 开发经历 |
| 朱天晨 | 产品经理 | C、C++、Python、Java、PHP、JavaScript | 参与开发过Android项目、Java项目及Web项目 |
| 康嘉玮 | 后端开发 | C、C#、Python、Java、PHP | 参与开发过Java项目和Web项目 |
| 曹亚斌 | 前端开发 | C、C++、Java、CSS、JavaScript | 参与开发过安卓项目和Java项目 |
| 卜令军 | 测试人员 | C、C++、Python、Java | 参与开发过Java项目、进行过多个项目的软件测试 |

表 8 - 1 人力资源分配表

### 开发人员使用的设施

笔记本电脑、服务器、打印机以及实验室

# 培训

## 项目的技术要求

### 管理技术

1. 燃尽图的使用
2. 数据自动化管理技术
3. 设备自动化管理技术
4. 软件测试管理技术
5. 安全管理自动化技术

### 开发技术

1. 敏捷开发技术
2. 面向对象思想
3. 面向服务架构思想
4. 用例驱动开发思想
5. 高级程序设计语言的应用以及算法设计
6. 自动化测试技术

## 培训计划

### 技能需求

本项目的开发需要PHP+MySQL辅以JavaScript语言，需要成员掌握面向对象的核心思想，能够独立抽象的建立模型，因而需要成员具有以下技能：

1. 至少会有三种以上的高级程序设计语言；
2. 有较强的面向对象思维模式，可以对具体事物进行抽象建模；
3. 能够使用PHP语言编写后台逻辑；
4. 能够熟练掌握MySQL和SQL语言；
5. 能够用HTML或者JavaScript语言编写网站界面；
6. 有较强的团队协作能力和沟通交流能力；
7. 有较强的撰写文档能力。

### 成员现状

大部分成员可以熟练掌握三种以上语言的用法，并且有一定的JAVA开发经验，面向对象思维能力较强，但是对于网站开发和数据库的开发这一部分较为陌生，其中PHP语言和JavaScript语言有部分人暂未掌握，也缺乏前端开发经验，同时对本项目的架构和开发流程不甚了解，因而有必要进行培训。

### 培训方案

|  |  |
| --- | --- |
| 时间 | 内容 |
| 2016.9.20-2016.9.22 | PHP语言学习 |
| 2016.9.23-2016.9.25 | JavaScript语言学习 |
| 2016.9.26-2016.9.28 | 网站开发学习 |
| 2016.9.29-2016.9.30 | MySQL和SQL语言学习 |
| 2016.10.7-2016.10.7 | 产品架构和开发流程讲解 |

表 9 - 1 培训时间安排表

# 项目估算

## 规模估算

经过预估计算，总共需要的模块数有七十多个，预计总代码量在三万八千行左右。

|  |  |  |
| --- | --- | --- |
| 类别 | 模块 | 预估代码量（行） |
| 前端界面 | 主页面 | 200 |
| 登录页面 | 300 |
| 注册页面 | 600 |
| 找回密码页面 | 200 |
| 帮助问题列表页面 | 600 |
| 帮助问题答案页面 | 400 |
| 联系我们页面 | 200 |
| 关于我们页面 | 200 |
| 教师面试页面 | 200 |
| 新闻讯息页面 | 500 |
| 学生用户信息页面 | 800 |
| 教师用户信息页面 | 800 |
| 修改个人信息页面 | 600 |
| 个人信息展示页面 | 600 |
| 用户等级页面 | 200 |
| 学生查询页面 | 300 |
| 学生查询结果页面 | 500 |
| 教师查询页面 | 300 |
| 教师查询结果页面 | 500 |
| 教学视频/课程搜索页面 | 600 |
| 教学视频/课程搜索结果页面 | 800 |
| 教学视频页面 | 1200 |
| 在线充值页面 | 800 |
| 所有教师列表页面 | 800 |
| 所有学生列表页面 | 800 |
| 预约选课页面 | 600 |
| 课件上传页面 | 400 |
| 课件放映页面 | 600 |
| 课程介绍页面 | 200 |
| 上课页面 | 1200 |
| 学生评价页面 | 600 |
| 论坛主页面 | 400 |
| 板块组页面 | 270 |
| 最新帖子页面 | 800 |
| 热帖页面 | 800 |
| 消息提醒页面 | 200 |
| 帖子页面 | 450 |
| 搜索页面 | 200 |
| 编辑帖子页面 | 700 |
| 留言板页面 | 2000 |
| 后台逻辑 | 主页逻辑 | 200 |
| 登录逻辑 | 200 |
| 注册逻辑 | 200 |
| 找回密码逻辑 | 100 |
| 帮助问题列表逻辑 | 200 |
| 帮助问题答案逻辑 | 200 |
| 顶层模块逻辑 | 300 |
| 底层模块逻辑 | 300 |
| 教师面试逻辑 | 200 |
| 新闻讯息逻辑 | 50 |
| 学生用户信息逻辑 | 200 |
| 教师用户信息逻辑 | 200 |
| 修改个人信息逻辑 | 800 |
| 个人信息展示逻辑 | 100 |
| 用户等级逻辑 | 100 |
| 学生查询逻辑 | 400 |
| 学生查询结果逻辑 | 200 |
| 教师查询逻辑 | 400 |
| 教师查询结果逻辑 | 200 |
| 教学视频/课程搜索逻辑 | 600 |
| 教学视频/课程搜索结果逻辑 | 300 |
| 教学视频逻辑 | 200 |
| 在线充值逻辑 | 400 |
| 所有教师列表逻辑 | 200 |
| 所有学生列表逻辑 | 200 |
| 预约选课逻辑 | 400 |
| 课件上传逻辑 | 400 |
| 课件放映逻辑 | 200 |
| 课程介绍逻辑 | 200 |
| 上课页逻辑 | 200 |
| 论坛主页逻辑 | 300 |
| 板块组逻辑 | 200 |
| 最新帖子逻辑 | 200 |
| 热帖逻辑 | 200 |
| 消息提醒逻辑 | 200 |
| 帖子页逻辑 | 400 |
| 搜索逻辑 | 400 |
| 编辑帖子逻辑 | 400 |
| 留言板逻辑 | 600 |
| 数据库操作逻辑 | 1000 |
| 数据收集分析逻辑 | 2000 |
| 师生匹配逻辑 | 2000 |
| 筛选推送 | 1000 |
| 数据库操作 | 新建数据库 | 50 |
| 数据添加 | 50 |
| 数据删除 | 50 |
| 数据修改 | 50 |
| 数据查询 | 50 |
| 数据属性添加 | 50 |
| 总计 |  | 38170 |

表 10 - 1 项目规模估算表

## 工作量估算

|  |  |  |
| --- | --- | --- |
| 序号 | WBS | 估计值（人天） |
| **1** | **开发阶段** | **102** |
| **1.1** | **计划调研学习阶段** | **11** |
| 1.1.1 | 市场调研 | 1 |
| 1.1.2 | 可行性分析 | 1 |
| 1.1.3 | 预算估计 | 1 |
| 1.1.4 | 撰写软件开发计划 | 4 |
| 1.1.5 | 撰写可行性分析报告 | 4 |
| **1.2** | **代码编写阶段** | **68** |
| 1.2.1 | 撰写软件需求规格说明 | 4 |
| 1.2.2 | UML建模 | 4 |
| 1.2.3 | 设计软件架构 | 4 |
| 1.2.4 | 撰写软件结构设计说明 | 4 |
| 1.2.5 | 确定功能规范 | 4 |
| 1.2.6 | 定义数据库模型 | 2 |
| 1.2.7 | 编写实现的技术用例 | 8 |
| 1.2.8 | 实现后端数据库录入 | 6 |
| 1.2.9 | 实现注册登录功能 | 4 |
| 1.2.10 | 初步设计网站各个界面 | 5 |
| 1.2.11 | 实现在线视频教学逻辑功能 | 8 |
| 1.2.12 | 实现论坛、博客、留言功能 | 8 |
| 1.2.13 | UI及界面优化 | 7 |
| **1.3** | **初步测试阶段** | **23** |
| 1.3.1 | 规定测试要点 | 2 |
| 1.3.2 | 撰写软件测试说明 | 1 |
| 1.3.3 | 开发人员手动测试 | 4 |
| 1.3.4 | 单元测试 | 4 |
| 1.3.5 | 模块集成测试 | 1 |
| 1.3.6 | 整体测试 | 1 |
| 1.3.7 | 自动化测试 | 2 |
| 1.3.8 | 测试人员覆盖性测试 | 2 |
| 1.3.9 | 撰写测试报告 | 1 |
| 1.3.10 | 修改代码以解决测试中出现的问题 | 4 |
| 1.3.11 | 再次测试修改后的代码 | 1 |
| **2** | **初步上线阶段** | **107** |
| **2.1** | **上线宣传阶段** | **21** |
| 2.1.1 | 网站上线运行 | 1 |
| 2.1.2 | 宣传网站吸引用户使用 | 20 |
| **2.2** | **用户测试阶段** | **32** |
| 2.2.1 | 压力测试 | 12 |
| 2.2.2 | 安全性测试 | 8 |
| 2.2.3 | 收集测试中遇到的问题 | 5 |
| 2.2.4 | 补充测试报告 | 1 |
| 2.2.5 | 修改代码解决遇到的问题 | 6 |
| **2.3** | **版本更新阶段** | **54** |
| 2.3.1 | 收集用户行为和用户标签 | 20 |
| 2.3.2 | 使用机器学习实现为用户推送合适的老师或学生功能 | 14 |
| 2.3.3 | 优化前一版本的算法与代码 | 14 |
| 2.3.4 | 制订用户手册与使用规范 | 4 |
| 2.3.5 | 审阅所有文档 | 2 |
|  | **总计** | **209** |

表 10 - 2 项目工作量估算表

## 成本估算

|  |  |
| --- | --- |
| 用途 | 预算（元） |
| 市场调查 | 200 |
| 设备维修 | 1000 |
| 服务器租借 | 6000 |
| 代码优化与维护 | 1000 |
| 上线宣传 | 1500 |
| 其他 | 1500 |
| 合计 | 11200 |

表 10 - 3 项目成本估算表

## 关键计算机资源估算

|  |  |  |  |
| --- | --- | --- | --- |
| 资源 | | | 配置 |
| 网络环境 | 网络类型 | | Ipv4广域网（校园网） |
| 网络带宽 | | 100M |
| 网卡 | | 100M网卡 |
| 数据传输速率 | | ADSL/512K |
| 网络硬件设备 | | 100M中心交换机 |
| 硬件设备 | 服务器\*1 | CPU | IntelCore 1G\*2 |
| 内存 | 1G |
| 硬盘 | SCSI 9G\*4 |
| 笔记本电脑\*4 | CPU | Intelcorei5 4200U 1.6G |
| 内存 | 4G |
| 硬盘 | 500G |

表 10 - 4 项目关键计算机资源估算表

## 管理预留

系统在容量以及功能上作了预留工作，以满足今后对系统升级的需要，是整个软件开发具有很强的应变能力，同时便于对软件系统进行优化与改进。

# 风险管理

## 人员变动

在项目的开发过程中有可能会出现组内成员由于特殊情况（如生病或其他原因）而不能按时完成工作的情况，此时应当及时吸纳另外的人进组递补，以保证项目能在规定的时间内完成。

## 资源损毁

### 硬件资源损毁

在项目的开发过程中由于使用Git进行版本管理，代码在编辑完成后会上传至GitHub中，因而即使硬件损坏，只要所写代码已及时备份到云端，那么很容易就可以恢复。另外在本地进行代码管理时也需要及时备份，以避免出现还未上传时硬件损坏导致代码丢失。

另外在项目发布后服务器上的内容每天都会进行人工备份，这样可以将硬件损毁的损失降低到最低。

### 软件资源损毁

本项目使用了Git进行版本控制，从而避免了文件的误删，损坏等情况带来的损失。

## 需求变化

当用户的需求产生变化时，首先对当前的项目完成程度进行评估，根据项目完成度以及需求变化程度开展组内讨论，讨论的内容是确定项目是否需要进行重构。如果确定要进行项目重构，则需要充分利用面向对象代码的可重用性，保证整个重构的工作可以在三天之内完成，并且修改调整后续的时间安排。如果不进行项目重构，则考虑在现有的基础上进行功能的删改，并进行相关的记录说明。

# 支持条件

## 计算机系统支持

本项目支持在Windows 7 及以上系统运行，支持IE 7.0 及以上版本的浏览器或Chrome浏览器。

## 需要需方承担的工作和提供的条件

1. 提供测试用数据；
2. 提供机器学习训练数据所用数据集；
3. 提供网络与服务器支持。

## 需要分包商承担的工作和提供的条件

暂未与分包商合作。