**“XXX”一对一家教平台**

**软件开发计划**

**Version：1.1**

编 写 人：\_\_\_\_\_\_\_\_朱天晨\_\_\_\_\_\_\_\_

校 验 人：\_\_\_\_\_\_\_\_XXX\_\_\_\_\_\_\_\_

编写日期：\_\_\_\_\_\_\_2016.10.01\_\_\_\_\_\_\_

目录

[1 引言 7](#_Toc463784969)

[1.1 标识 7](#_Toc463784970)

[1.2 系统概述 8](#_Toc463784971)

[1.3 文档概述 8](#_Toc463784972)

[1.4 与其他计划之间的关系 9](#_Toc463784973)

[1.5 基线 9](#_Toc463784974)

[2 引用文件 9](#_Toc463784975)

[3 交付产品 10](#_Toc463784976)

[3.1 程序 10](#_Toc463784977)

[3.2 文档 10](#_Toc463784978)

[3.3 服务 11](#_Toc463784979)

[3.4 非移交产品 11](#_Toc463784980)

[3.5 验收标准 12](#_Toc463784981)

[3.6 最后交付期限 13](#_Toc463784982)

[4 所需工作概述 13](#_Toc463784983)

[4.1 对开发系统的需求与约束 13](#_Toc463784984)

[4.1.1 硬件环境 13](#_Toc463784985)

[4.1.2 软件环境 13](#_Toc463784986)

[4.2 对项目文档编制的需求与约束 14](#_Toc463784987)

[4.3 项目在系统生命周期中所处地位 14](#_Toc463784988)

[4.4 技术需求 14](#_Toc463784989)

[4.5 项目进度安排以及对资源的需求与约束 14](#_Toc463784990)

[4.6 其他需求与约束 14](#_Toc463784991)

[5 实施整个软件开发活动的计划 15](#_Toc463784992)

[5.1 软件开发过程 15](#_Toc463784993)

[5.2 软件开发总体计划 15](#_Toc463784994)

[5.2.1 软件开发方法 15](#_Toc463784995)

[5.2.2 软件产品标准 15](#_Toc463784996)

[5.2.3 可重用的软件产品 16](#_Toc463784997)

[5.2.4 处理关键性需求 16](#_Toc463784998)

[5.2.5 计算机硬件资源利用 17](#_Toc463784999)

[5.2.6 记录原理 17](#_Toc463785000)

[5.2.7 需方评审途径 17](#_Toc463785001)

[6 实施详细软件开发活动的计划 17](#_Toc463785002)

[6.1 项目计划和监督 17](#_Toc463785003)

[6.1.1 软件开发计划 17](#_Toc463785004)

[6.1.2 CSCI测试计划 18](#_Toc463785005)

[6.1.3 系统测试计划 19](#_Toc463785006)

[6.1.4 软件安装计划 19](#_Toc463785007)

[6.1.5 软件移交计划 19](#_Toc463785008)

[6.1.6 跟踪和更新计划 20](#_Toc463785009)

[6.2 建立软件开发环境 20](#_Toc463785010)

[6.2.1 软件工程环境 20](#_Toc463785011)

[6.2.2 软件测试环境 21](#_Toc463785012)

[6.2.3 软件开发库 21](#_Toc463785013)

[6.2.4 软件开发文档 21](#_Toc463785014)

[6.2.5 非交付软件 21](#_Toc463785015)

[6.3 系统需求分析 21](#_Toc463785016)

[6.3.1 用户输入分析 21](#_Toc463785017)

[6.3.2 运行概念 22](#_Toc463785018)

[6.3.3 系统需求 22](#_Toc463785019)

[6.4 系统设计 22](#_Toc463785020)

[6.4.1 系统级设计决策 22](#_Toc463785021)

[6.4.2 系统体系结构设计 23](#_Toc463785022)

[6.5 软件需求分析 23](#_Toc463785023)

[6.6 软件设计 24](#_Toc463785024)

[6.6.1 CSCI级设计决策 24](#_Toc463785025)

[6.6.2 CSCI体系结构设计 25](#_Toc463785026)

[6.6.3 CSCI详细设计 25](#_Toc463785027)

[6.7 软件实现和配置项测试 25](#_Toc463785028)

[6.7.1 软件实现 25](#_Toc463785029)

[6.7.2 配置项测试准备 25](#_Toc463785030)

[6.7.3 配置项测试执行 25](#_Toc463785031)

[6.7.4 测试和再测试 26](#_Toc463785032)

[6.7.5 配置项测试结果分析与记录 26](#_Toc463785033)

[6.8 配置项集成和测试 26](#_Toc463785034)

[6.8.1 配置项集成和测试准备 26](#_Toc463785035)

[6.8.2 配置项集成和测试执行 26](#_Toc463785036)

[6.8.3 修改和再测试 26](#_Toc463785037)

[6.8.4 配置项集成和测试结果分析与记录 26](#_Toc463785038)

[6.9 CSCI合格性测试 26](#_Toc463785039)

[6.9.1 CSCI合格性测试的独立性 26](#_Toc463785040)

[6.9.2 在目标计算机系统上测试 27](#_Toc463785041)

[6.9.3 CSCI合格性测试准备 27](#_Toc463785042)

[6.9.4 CSCI合格性测试演练 27](#_Toc463785043)

[6.9.5 CSCI合格性测试执行 27](#_Toc463785044)

[6.9.6 修改和再测试 27](#_Toc463785045)

[6.9.7 CSCI合格性测试结果分析与记录 27](#_Toc463785046)

[6.10 CSCI/HWCI集成和测试 27](#_Toc463785047)

[6.10.1 CSCI/HWCI集成和测试设备 27](#_Toc463785048)

[6.10.2 CSCI/HWCI集成和测试执行 27](#_Toc463785049)

[6.10.3 修改和再测试 28](#_Toc463785050)

[6.10.4 CSCI/HWCI集成和测试结果分析与记录 28](#_Toc463785051)

[6.11 系统合格性测试 28](#_Toc463785052)

[6.11.1 系统合格性测试的独立性 28](#_Toc463785053)

[6.11.2 在目标计算机系统上测试 28](#_Toc463785054)

[6.11.3 系统合格性测试准备 28](#_Toc463785055)

[6.11.4 系统合格性测试演练 28](#_Toc463785056)

[6.11.5 系统合格性测试执行 28](#_Toc463785057)

[6.11.6 修改和再测试 28](#_Toc463785058)

[6.11.7 系统合格性测试结果分析与记录 29](#_Toc463785059)

[6.12 软件使用准备 29](#_Toc463785060)

[6.12.1 可执行软件的准备 29](#_Toc463785061)

[6.12.2 用户现场的版本说明的准备 29](#_Toc463785062)

[6.12.3 用户手册的准备 29](#_Toc463785063)

[6.12.4 在用户现场安装 29](#_Toc463785064)

[6.13 软件移交准备 29](#_Toc463785065)

[6.13.1 可执行软件的准备 29](#_Toc463785066)

[6.13.2 源文件准备 29](#_Toc463785067)

[6.13.3 支持现场的版本说明的准备 30](#_Toc463785068)

[6.13.4 “已完成”的CSCI设计和其他的软件支持信息的准备 30](#_Toc463785069)

[6.13.5 系统设计说明的更新 30](#_Toc463785070)

[6.13.6 支持手册的准备 30](#_Toc463785071)

[6.13.7 到指定现场的移交 30](#_Toc463785072)

[6.14 软件配置管理 30](#_Toc463785073)

[6.14.1 配置标识 30](#_Toc463785074)

[6.14.2 配置控制 30](#_Toc463785075)

[6.14.3 配置状态统计 31](#_Toc463785076)

[6.14.4 配置审核 31](#_Toc463785077)

[6.14.5 发行管理和交付 31](#_Toc463785078)

[6.15 软件产品评估 31](#_Toc463785079)

[6.15.1 中间阶段的和最终的软件产品评估 31](#_Toc463785080)

[6.15.2 软件产品评估记录 31](#_Toc463785081)

[6.15.3 软件产品评估的独立性 31](#_Toc463785082)

[6.16 软件质量保证 31](#_Toc463785083)

[6.16.1 软件质量保证评估 31](#_Toc463785084)

[6.16.2 软件质量保证记录 32](#_Toc463785085)

[6.16.3 软件质量保证的独立性 32](#_Toc463785086)

[6.17 问题解决过程 32](#_Toc463785087)

[6.17.1 问题/变更报告 32](#_Toc463785088)

[6.17.2 更正活动系统 32](#_Toc463785089)

[6.18 联合评审 32](#_Toc463785090)

[6.18.1 联合技术评审 32](#_Toc463785091)

[6.18.2 联合管理评审 32](#_Toc463785092)

[6.19 文档编制 32](#_Toc463785093)

[6.20 其他软件开发活动 33](#_Toc463785094)

[6.20.1 风险管理 33](#_Toc463785095)

[6.20.2 软件管理指标 33](#_Toc463785096)

[6.20.3 保密性和私密性 34](#_Toc463785097)

[6.20.4 分承包方管理 34](#_Toc463785098)

[6.20.5 与软件独立验证与确认机构的接口 34](#_Toc463785099)

[6.20.6 和有关开发方的协调 34](#_Toc463785100)

[6.20.7 项目过程的改进 34](#_Toc463785101)

[6.20.8 计划中未提及的其他活动 34](#_Toc463785102)

[7 进度表和活动网络图 35](#_Toc463785103)

[7.1 进度表 35](#_Toc463785104)

[7.1.1 人员分配表 35](#_Toc463785105)

[7.1.2 35](#_Toc463785106)

[7.2 活动网络图 39](#_Toc463785107)

[8 项目组织和资源 41](#_Toc463785108)

[8.1 项目组织 41](#_Toc463785109)

[8.2 项目资源 42](#_Toc463785110)

[8.2.1 人力资源 42](#_Toc463785111)

[8.2.2 开发人员使用的设施 42](#_Toc463785112)

[9 培训 42](#_Toc463785113)

[9.1 项目的技术要求 42](#_Toc463785114)

[9.1.1 管理技术 42](#_Toc463785115)

[9.1.2 开发技术 43](#_Toc463785116)

[9.2 培训计划 43](#_Toc463785117)

[9.2.1 技能需求 43](#_Toc463785118)

[9.2.2 成员现状 43](#_Toc463785119)

[9.2.3 培训方案 44](#_Toc463785120)

[10 项目估算 44](#_Toc463785121)

[10.1 规模估算 44](#_Toc463785122)

[10.2 工作量估算 44](#_Toc463785123)

[10.3 成本估算 46](#_Toc463785124)

[10.4 关键计算机资源估算 46](#_Toc463785125)

[10.5 管理预留 47](#_Toc463785126)

[11 风险管理 47](#_Toc463785127)

[11.1 人员变动 47](#_Toc463785128)

[11.2 资源损毁 47](#_Toc463785129)

[11.2.1 硬件资源损毁 47](#_Toc463785130)

[11.2.2 软件资源损毁 47](#_Toc463785131)

[12 支持条件 48](#_Toc463785132)

[12.1 计算机系统支持 48](#_Toc463785133)

[12.2 需要需方承担的工作和提供的条件 48](#_Toc463785134)

[12.3 需要分包商承担的工作和提供的条件 48](#_Toc463785135)

# 1 引言

## 标识

软件工程的名称为“XXX”一对一家教平台，该软件工程的最终呈现形式是名称为“XXX”的网站。

## 系统概述

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

软件开发过程中需要对需求进行考察分析，编写相应的需求文档，并根据需求文档设计相应的类，对于每一个类以及每一个方法都要撰写相应的规格，规范其输入以及可能出现的结果，这样既方便测试，也可以有效的排查软件当中出现的问题。本软件的基线是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。

# 引用文件

[1] Roger S.Pressman。软件工程—实践者的研究方法。机械工业出版社，2010年10月

# 交付产品

## 程序

1. 项目名称：“XXX”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. 《用户使用手册》：

介绍软件平台的功能以及使用方法，为用户操作提供操作指南，使用户在最短的时间内完成平台的注册，了解平台的使用技巧；

1. 《平台维护手册》：

具体介绍平台的使用环境，适用系统，对各个模块的功能与作用进行说明，阐述维护时的注意事项；

1. 《开发人员手册》：

记录开发过程中的问题，以及开发过程的注意要点，阐释每一个模块的细节问题，为后续的开发提供模板以及注意事项。

## 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. 《软件开发计划》：

软件开发前期对于开发的设计与估算，软件工程计划的核心。

1. 《软件需求规格说明》：

设计阶段对于每一个模块，每一个类的规格设计文档。

1. 《软件结构设计说明》：

软件架构设计的说明文档，展示了软件平台整体的结构。

1. 《软件测试说明》：

软件测试阶段撰写的关于测试的相关记录。

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

该项目在系统生命周期中占据核心地位，该项目的完成是整个系统按照计划运行的前提。

## 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

软件开发的过程包含两个阶段，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

## 软件开发总体计划

### 软件开发方法

平台：Windows 7 及以上操作系统

浏览器支持：IE 7.0 及以上或Chrome浏览器

版本控制：Git

数据库支持：MySQL

开发语言：PHP

开发方式：敏捷开发，面向对象

### 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 |

### 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 |

### 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 |

### 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：Quality Center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

## 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

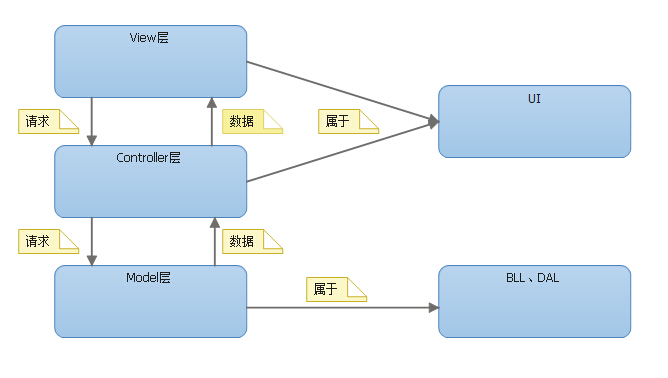
1. 约束条件：

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

### CSCI体系结构设计



### CSCI详细设计

## 软件实现和配置项测试

### 软件实现

### 配置项测试准备

### 配置项测试执行

### 测试和再测试

### 配置项测试结果分析与记录

## 配置项集成和测试

### 配置项集成和测试准备

### 配置项集成和测试执行

### 修改和再测试

### 配置项集成和测试结果分析与记录

## CSCI合格性测试

### CSCI合格性测试的独立性

### 在目标计算机系统上测试

### CSCI合格性测试准备

### CSCI合格性测试演练

### CSCI合格性测试执行

### 修改和再测试

### CSCI合格性测试结果分析与记录

## CSCI/HWCI集成和测试

### CSCI/HWCI集成和测试设备

### CSCI/HWCI集成和测试执行

### 修改和再测试

### CSCI/HWCI集成和测试结果分析与记录

## 系统合格性测试

### 系统合格性测试的独立性

### 在目标计算机系统上测试

### 系统合格性测试准备

### 系统合格性测试演练

### 系统合格性测试执行

### 修改和再测试

### 系统合格性测试结果分析与记录

## 软件使用准备

### 可执行软件的准备

### 用户现场的版本说明的准备

### 用户手册的准备

### 在用户现场安装

## 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

### 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。

## 软件配置管理

### 配置标识

### 配置控制

### 配置状态统计

### 配置审核

### 发行管理和交付

## 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。

## 软件质量保证

### 软件质量保证评估

### 软件质量保证记录

### 软件质量保证的独立性

## 问题解决过程

### 问题/变更报告

### 更正活动系统

## 联合评审

### 联合技术评审

### 联合管理评审

## 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。

## 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。

### 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

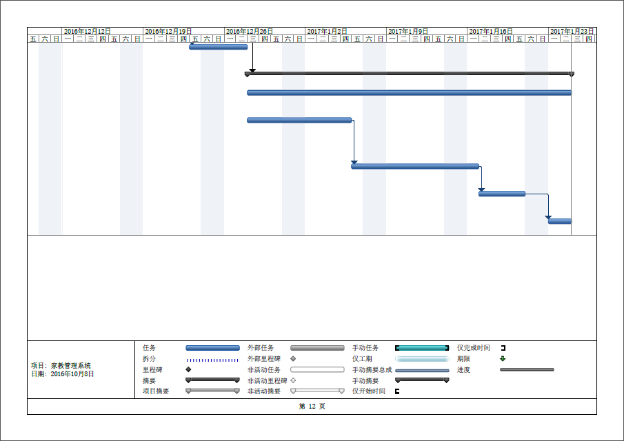
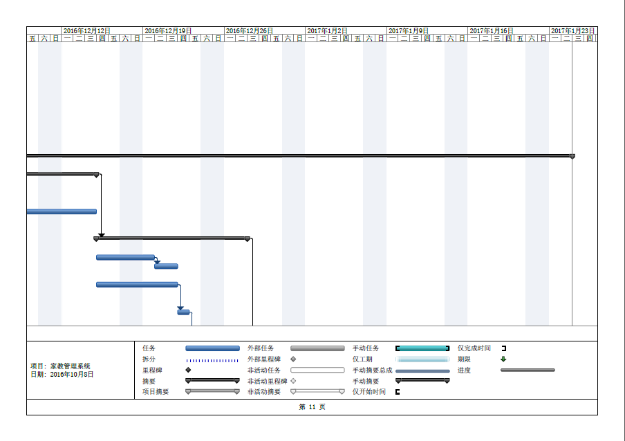
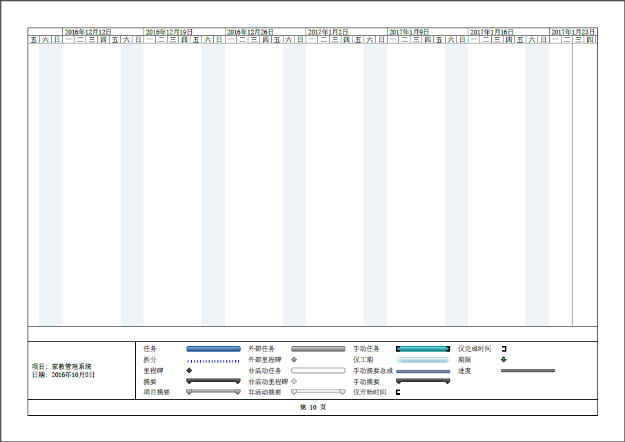
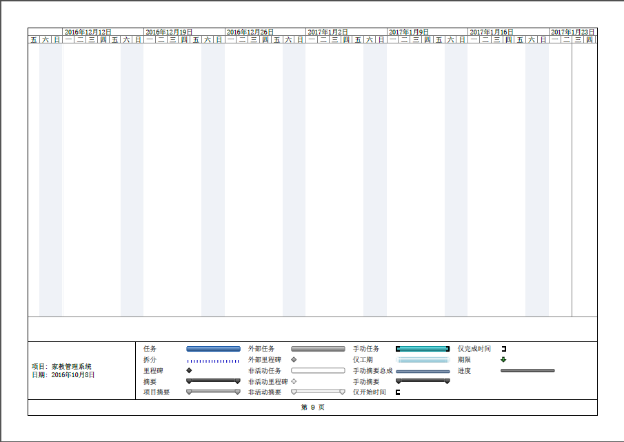
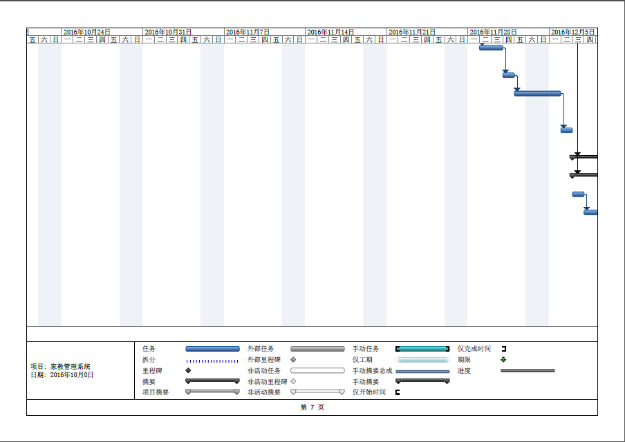
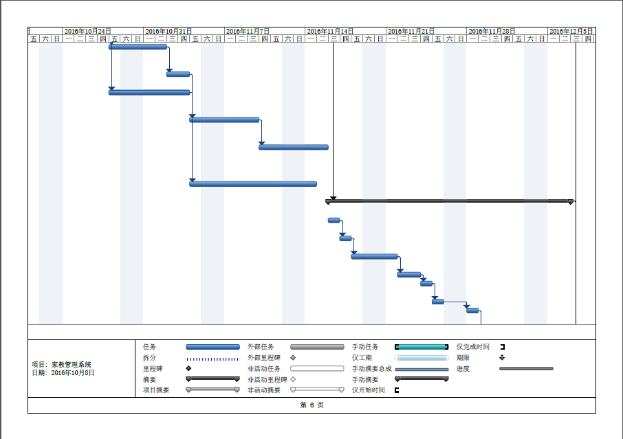
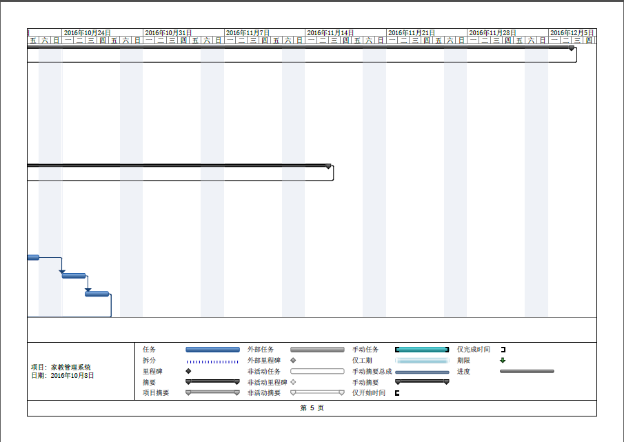
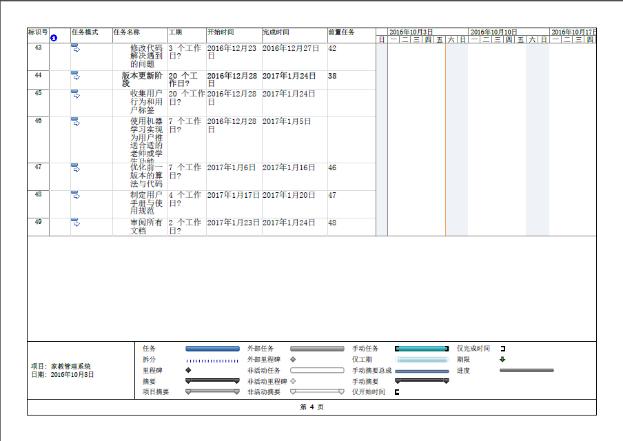
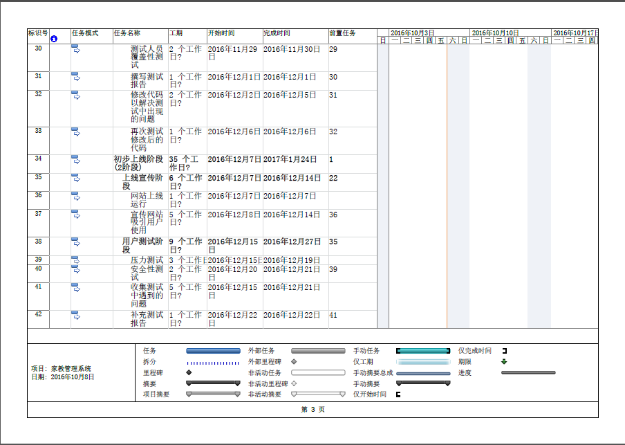
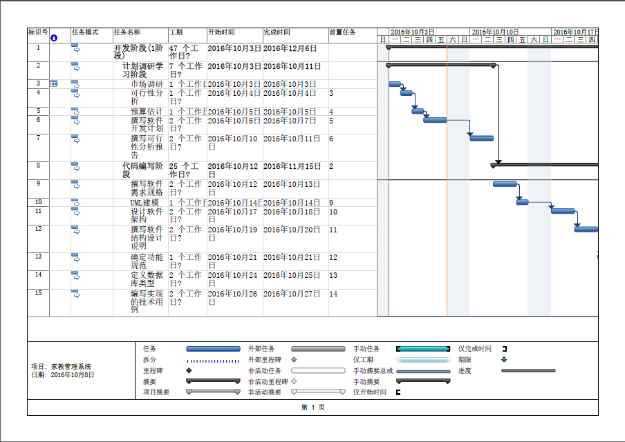
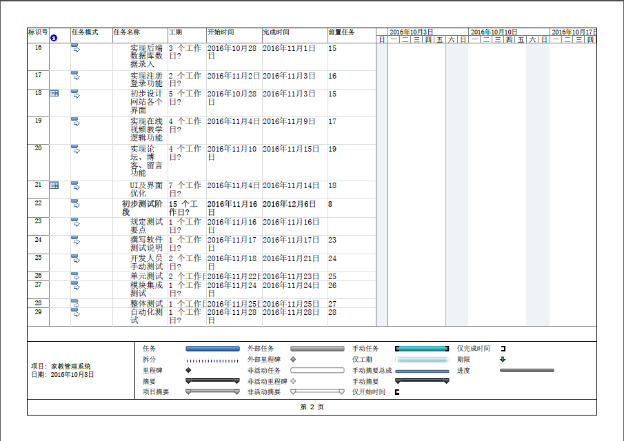
|  |  |
| --- | --- |
| 任务分解 | 人员分工 |
| 市场调研、需求分析 | 项目组全体人员 |
| 软件架构 | 项目组长 |
| 前端开发 | 前端开发人员 |
| 后端逻辑 | 后端开发人员 |
| 软件测试 | 软件测试人员 |
| 文档撰写 | 项目组全体成员 |
| 上线宣传 | 项目组全体成员 |
| 收集反馈信息 | 项目组长 |
| 完善功能，修改漏洞，维护代码 | 后端开发人员 |

### 

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 | 2016.10.3 | 2016.10.3 | 1天 |
| 可行性分析 | 2016.10.4 | 2016.10.4 | 1天 |
| 预算估计 | 2016.10.5 | 2016.10.5 | 1天 |
| 撰写软件开发计划 | 2016.10.6 | 2016.10.7 | 2天 |
| 撰写可行性分析报告 | 2016.10.10 | 2016.10.11 | 2天 |
| 代码编写阶段 | 撰写软件需求规格说明 | 2016.10.12 | 2016.10.13 | 2天 |
| UML建模 | 2016.10.14 | 2016.10.14 | 1天 |
| 设计软件架构 | 2016.10.17 | 2016.10.18 | 2天 |
| 撰写软件结构设计说明 | 2016.10.19 | 2016.10.20 | 2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 | 2016.11.16 | 2016.11.16 | 1天 |
| 撰写软件测试说明 | 2016.11.17 | 2016.11.17 | 1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 | 2016.12.2 | 2016.12.5 | 2天 |
| 再次测试修改后的代码 | 2016.12.6 | 2016.12.6 | 1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 | 2016.12.15 | 2016.12.21 | 5天 |
| 补充测试报告 | 2016.12.22 | 2016.12.22 | 1天 |
| 修改代码解决遇到的问题 | 2016.12.23 | 2016.12.27 | 3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 |

## 活动网络图



# 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。

## 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 |

### 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 |

# 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行左右。

## 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** |

## 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 |

## 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 | Intel Core 1G\*2 |
| 内存 | 1G |
| 硬盘 | SCSI 9G\*4 |
| 笔记本电脑\*4 | CPU | Intel core i5 4200U 1.6G |
| 内存 | 4G |
| 硬盘 | 500G |

## 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。

# 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。