# 珞樱—开发计划书

//编制项目开发计划的目的是用文件的形式，把对于在开发过程中各项工作的负责人员、开发进度、 所需经费预算、所需软、硬件条件等问题作出的安排记载下来，以便根据本计划开展和检查本项目的开发工作。编制内容要求如下：

## **1.** **引言**

### **1.1.** **编写目的**

//说明编写这份项目开发计划的目的，并指出预期的读者。

该文档旨在明确开发目标、人员等宏观调配，估算大致的开发经济需求，并对齐开发过程中所使用的软硬件工具，使开发过程结构化、标准化以提高开发效率。

预期的读者：组内的其他成员以及后期供老师审阅用。

### **1.2.** **背景**

//说明开发该软件系统的目的和现状

目的：开发“珞樱”的目的在于提供一个集成的武大学子自己的校园生活社区。其中包括：

* 论坛：在其中可以记录课余生活、讨论校园相关事件、沟通学习难题，在话题性、沟通性、范围性上贴合武大学子的日常生活；
* 课程评价：手机对课程老师的评价，向同学们客观展示数据统计，方便同学们在选课阶段参考
* 课程表：获取智慧珞珈的课程表api，导入课表并进行上课提醒
* 校园卡管理：通过应用查看校园卡明细。
* AI日程管理：利用大语言模型进行日常的日程安排。

现状：

* 论坛(Forum/BBS)是基于互联网的网上交流场所，其作用与“公告板”类似，是人们分享知识，交流观点的重要场所。随着如Facebook(现X),微博等社交媒体的兴起，人们倾向于在这些平台上进行实时的沟通交流，但是交流内容偏碎片化，又缺乏了论坛所拥有的话题性和讨论性。同时，许多一流高校都有自己的学生论坛，比如北京大学的“北大树洞”；上海交通大学的“水源论坛”，满足了学生的互联交流需求，给学生学习、生活等方面带来支持。
* 课程表：传统课程表仅有显示基本信息的功能，缺乏了日程提醒功能。
* 课程评价：为了满足学生在选课时对于老师上课时间、上课风格、教授水平高低以及给分好坏的参考，课程评价乃必须之选。而现有的数据集时效性差，最新版本的更新停留在3年前，并且交互行为不够新颖、方便。
* 校园卡管理：智慧珞珈虽然具备校园卡管理的功能，但每次使用时，步骤多、启动时长慢，不方便同学们在一些情况下的使用。需要快捷、集成的启动方式。
* AI日程管理(暂定)：大多数大学生缺乏对自己的生活的安排能力，虽然时间宽裕但是并无法合理、严谨地按照自己所想去安排日程。AI日程管理可以将学习、生活、爱好等自由、合理地安排，同时支持学生自主调整节奏，过好大学生活。

**该系统同其他系统或用户之间的基本的相互来往关系及其对软件系统的约束和要求。（系统流程图）**

### **1.3.** **定义（待）**

列出本文件中用到的专门术语的定义

### **1.4.** **参考资料 （待）**

列出本文件用到的参考文献，如：

a) 本项目的经核准的计划任务书或合同、上级机关的批文；

b) 属于本项目的其他已发表的文件；

c) 本文件中各处引用的文件、资料，包括所要用到的软件开发标准。 列出这些文件资料的标题、文件编号、发表日期和出版单位，说明能够得到这些文件资料的来源。

## **2.** **项目概述**

为了给同学创造良好的校园生活环境，结合信息化飞速发展的时代特色，我们推出的“珞樱”下应用主要聚焦武大学生生活。我们参考了小红书式的交互方式，主要在移动端上以”卡片话题“的展示方式将话题清晰地推出给用户，并结合了论坛的交流性和话题性，

本平台主要实现：

* 生活小助手：论坛、课程表、课程评价、校园卡管理、AI日程管理

### **2.1.** **工作内容**

简要地说明在本项目的目标和主要开发工作。

### **2.2.** **项目主要成员**

组长：杨嘉逸：管理能力强

组员（按音序）：

* 李嘉伦：效率奇高
* 刘玮祺：组织力极强
* 孙东方：项目经验多的要命
* 孙志成：创造力爆表

### **2.3.** **产品及成果**

#### **2.3.1.** **程序**

//列出需移交给用户的程序的名称、所用的编程语言及存储程序的媒体形式，并通逐项说明其功能和能力。

名称：珞樱

编程框架：

* Vue.js作为前端框架
* Spring Boot作为后端框架
* redis/ MongoDB作为数据库
* 接入ChatGPT接口进行制定日程安排

程序应用形式：

* 移动端为主：为了保证程序的可用性，我们以移动端和小程序开发为主。

#### **2.3.2.** **文档**

* 用户操作手册：
  1. 简介和概述：
     + 软件的名称和版本信息。
     + 软件的简要介绍，包括其主要功能和用途。
  2. 安装和配置：
     + 安装软件的步骤和要求。
     + 配置软件的详细说明，包括设置选项和参数。
  3. 界面导览：
     + 软件的主要界面和布局的介绍，包括各个部分的功能和用途。
     + 各种操作界面的截图和标注，帮助用户更好地理解界面的组成和操作方式。
  4. 基本操作：
     + 软件的基本操作指南，包括如何进行登录、退出、创建、保存、编辑等常见操作。
     + 操作的步骤和详细说明，以及相关的截图和示例。
  5. 高级功能：
     + 软件的高级功能和特性的介绍，包括如何使用高级功能、设置参数和选项等。
     + 高级操作的步骤和详细说明，以及相关的示例和截图。
  6. 故障排除：
     + 常见问题和故障的解决方法，包括错误消息的含义和解决方案。
     + 如何联系技术支持团队或获取帮助的信息。
  7. 附录：
     + 相关参考资料和链接，如用户手册更新信息、常见问题解答、技术支持联系方式等。
     + 术语表和缩略语表，帮助用户理解文档中的专业术语和缩略语的含义。
  8. 索引：
  + 索引可以按关键词列出手册中的内容，方便用户快速查找和定位需要的信息。
* 软件维护手册:主要包括软件系统说明、程序模块说明、操作环境、支持软件的说明、维护过程的说明，便于软件的维护。

//列出需提交的文档的名称及内容要点。

#### **2.3.3.** **服务**

//列出需向用户提供的各项服务，如培训安装、维护和运行支持等，应逐项规定开始日期、所提供支持的级别和服务的期限。

计划提供以下服务:

* 课堂培训:以讲课形式对客户进行培训，使其能够正确使用软件免费咨询:客户可以在工作时间向技术人员提出问题并获得解答。技术支持:对于某些客户，采取上门指导的方式。
* 软件维护:获取软件使用中的问题，提供补丁程序。
* 升级通知:在软件新版本发布的时候，及时通知(Email或邮寄信件)注册的用户，并提供试用版本。
* 软件升级:对于注册用户，只需较少的费用即可升级到新的版本。

#### **2.3.4.** **非移交产品**

//说明开发团队该开发但不必向用户移交的产品（文件甚至某些程序）。

* 可行性分析报告:说明该软件开发项目的实现在技术上、经济上和社会因素上的可行性，评述为了合理地达到开发目标可供选择的各种可能实施方案，说明并论证所选定实施方案的理由。
* 项目开发计划:为软件项目实施方案制订出具体计划，应该包括各部分工作的负责人员、开发的进度、开发经费的预算、所需的硬件及软件资源等。
* 软件需求说明书(软件规格说明书):对所开发软件的功能、性能、用户界面及运行环境等做出详细的说明。它是在用户与开发人员双方对软件需求取得共同理解并达成协议的条件下编写的，也是实施开发工作的基础。该说明书应给出数据逻辑和数据采集的各项要求，为生成和维护系统数据文件做好准备。
* 概要设计说明书:该说明书是概要实际阶段的工作成果，它应说明功能分配、模块划分、程序的总体结构、输入输出以及接口设计、运行设计、数据结构设计和出错处理设计等，为详细设计提供基础。
* 详细设计说明书:着重描述每一模块是怎样实现的，包括实现算法、逻辑流程等  
  测试计划:为做好集成测试和验收测试，需为如何组织测试制订实施计划。计划应包括测试的内容、进度、条件、人员、测试用例的选取原则测试结果允许的偏差范围等
* 测试分析报告:测试工作完成以后，应提交测试计划执行情况的说明对测试结果加以分析，并提出测试的结论意见。
* 开发进度月报:该月报系软件人员按月向管理部门提交的项目进展情况报告，报告应包括进度计划与实际执行情况的比较、阶段成果、遇到的问题和解决的办法以及下个月的打算等。
* 项目开发总结报告:软件项目开发完成以后，应与项目实施计划对照总结实际执行的情况，如进度、成果、资源利用、成本和投入的人力此外，还需对开发工作做出评价，总结出经验和教训。
* 软件问题报告:指出软件问题的登记情况，如日期、发现人、状态、问题所属模块等，为软件修改提供准备文档。
* 软件修改报告:软件产品投入运行以后，发现了需对其进行修正、更改等问题，应将存在的问题、修改的考虑以及修改的影响做出详细的描述提交审批。
* 源程序:软件开发过程中的全部代码以及注释。

### **2.4.** **验收标准**

对于上述这些应交出的产品和服务，逐项说明或引用资料说明验收标准。

1. 代码验收：在交付提交之前进行小组内评审，代码编写符合HB6465标准，与文档说明保持一致，代码风格统一，采用标准规范，没有以下错误：由于软件缺陷造成数据缺失，不符合设计要求，响应时间太长无法接受等问题。
2. 文档验收：在交付提交之前进行小组内评审，代码编写符合HB6465标准，功能与文档说明保持一致，清晰易读，没有语病与歧义。
3. 服务验收：服务硬件达到文档说明的要求，人员技术考核合格，在老师考核之后，对项目进行维修维护。

### **2.5.** **完成项目的最迟期限**

第十五周。

## **3.** **实施总计划**

### **3.1.** **工作任务的分解**

//对于项目开发中需完成的各项工作，从计划、需求分析、设计、实现、测试，包括文件的编制、评审等工作，软件安装工作等，进行分解，指明任务之间的依赖关系和每项任务的负责人和参加人员。（任务关系图/表）

所有组员都要进行全栈开发，但有所偏重：

前端：孙东方；杨嘉逸；

后端：李嘉伦；孙志成；刘玮祺

测试：一起！

### **3.2.** **接口人员**

//说明负责接口工作（与用户或其他系统的接口）的人员及他们的职责

### **3.3.** **进度**

//根据4.1任务之间的关系和用户时间约束，给出任务的进度安排（甘特图）和人员的分工，以及结束的标准

1-8周：项目策略案与文档的初步敲定，确定方向

8-13周：开发完成

13-15周：

* 13-14周完成测试；
* 13-15周同步进行文档最后阶段的撰写。

### **3.4.** **预算**

主要是时间预算！

### **3.5.** **关键问题**

//列出能够影响整个项目成败的关键问题、技术难点和风险，指出这些问题对项目的影响以及解决措施

| **问题描述** | **策略** |
| --- | --- |
| 课程表接口难以获得 | 向武汉大学智慧珞珈系统请求api接口 |
| 效率问题 | 坚决在每个阶段性ddl之前完成工作，包括本身任务以及接口任务。 |
| 专业基础知识不定 | 只有通过不断的实践 |
| 软件性能的影响 | 选择合适的软件，搭建良好的配置开发环境 |

## **4.** **支持条件**

说明为支持本项目的开发、测试、运行所需要的各种支持和条件

硬件环境

| **设备名称** | **用途** | **配置信息** | **备注** |
| --- | --- | --- | --- |
| Windows11 | c测试用 | windows11家庭版 | 无 |

软件环境

| **软件名称** | **软件类别** | **版本号** | **补丁程序** | **备注** |
| --- | --- | --- | --- | --- |
| VSCode | 编译器 | 1.87.2 | 无 | 编译测试用 |

## **5.** **专题计划要点**

//说明本项目开发中需制订的各个专题计划，包括沟通计划、质量保证计划等

### **5.1.** **沟通计划**

//沟通安排表：沟通时间、沟通内容、沟通方式、沟通对象、负责人

* 沟通时间：每周三晚上，周末；
* 沟通内容：周三晚对齐进度，讨论计划，做总结；周末集中进行开发
* 沟通方式：线下沟通，视情况线上讨论；
* 沟通对象：全体队员
* 负责人：组长杨嘉逸

### **5.2.** **质量保证计划**

介绍项目质量的目标和质量活动的安排

质量活动安排表：质量活动时间、质量活动内容、检查方式、检查依据、负责人、参与人