界面原型迭代计划

项目名称：交集——校园活动集成平台

组号：第9组

小组成员：胡彤 杨菡雪 全雨乐 张奕涵 杜心敏

迭代名称：界面原型迭代

制定日期：2023年2月22日

任务

PART1：需求

- [ ] 初步确定：同类产品分析，确定产品定位，分析优劣
- [ ] 调研需求：制定、发布问卷
- [ ] 分析需求：制定use-case模型
- [ ] 总结确定需求：制作Vision文档

PART2：技术学习

- [ ] 初选语言、工具和框架：JAVA、HTML、Prototype工具、Visio工具
- [ ] 建立GitHub仓库，管理文档、模型、代码
- [ ] 学习新技术

PART3：设计

- [ ] 界面初始设计：手绘草图、Visio工具
- [ ] 确定界面上细节：主要元素
- [ ] 用HTML实现界面原型

PART4：评审

- [ ] 小组内部评审、讨论，记录问题
- [ ] 改进需求文档
- [ ] 改进界面原型

- [ ] 编写《迭代评估报告》

- [ ] 开发总结，制作PPT汇报

预期成果

1. vision文档（2.27前）

根据问卷调查结果，确定项目定位、主要功能及其实现优先级，依照以下条目编写项目vision文档。

1. 简介 Introduction
2. 定位 Positioning
    - 2.1 商机 2.2 问题说明 2.3 产品定位
3. 项目涉众和用户描述 Stakeholder and User Descriptions
4. 产品概述 Product Overview
5. 产品特性 Product Features
6. 约束 Constraints
7. 质量范围 Quality Ranges
8. 优先级 Precedence and Priority
9. 其它产品需求 Other Product Requirements
10. 文档需求 Documentation Requirements

2. html界面原型（3.5前）

需包含以下界面的设计及页面间跳转逻辑：

- 活动大厅（主界面）
- 活动参与者个人中心
- 活动报名页面
- 活动发布者管理中心

3. 迭代评估报告（3.14前）

包含以下内容：

- 任务完成情况，开发进度
- 小组内部评审、讨论的问题记录
- 改进方案与实现情况
- 开发总结：经验与教训
- 制作PPT汇报展示

进度安排及人员分配

表格

	任务名	开始日期	截止日期	人员
	需求 3 条记录			
1	需求调研	2023/02/22	2023/02/25	杜心敏 全雨乐 杨菡
2	需求分析	2023/02/25	2023/02/26	杜心敏 杨菡雪 全雨
3	需求确定	2023/02/26	2023/02/26	杜心敏 杨菡雪 全雨
	技术 2 条记录			
1	基础功能技术学习	2023/02/25	2023/03/05	张奕涵 胡彤 杜心敏
2	进阶功能技术学习	2023/03/04	2023/03/10	张奕涵 全雨乐 杨菡
	界面 3 条记录			
1	界面草图设计	2023/02/25	2023/02/26	张奕涵 胡彤
2	界面修改确定	2023/02/27	2023/02/28	张奕涵 胡彤
3	界面html实现	2023/03/01	2023/03/04	张奕涵 胡彤
	开发 1 条记录			
1	基础功能开发	2023/03/04	2023/03/13	张奕涵 全雨乐 杨菡
	评审 4 条记录			
1	需求文档改进	2023/03/03	2023/03/04	张奕涵 杜心敏 杨菡
2	界面原型改进	2023/03/07	2023/03/08	张奕涵 杜心敏 杨菡
3	小组内部评审	2023/03/06	2023/03/07	杜心敏 张奕涵 胡彤
4	撰写评估报告、汇报PPT	2023/03/10	2023/03/13	张奕涵 杜心敏 胡彤
	13 条记录			

主要风险及应对方案

风险1：需求风险

软件项目在初期确定的需求往往都是模糊的、不确定的，而且随着项目的进展，需求还可能不断变化，这些问题如果没有得到及时的解决，就会对项目的成功造成巨大的潜在威胁。

需求风险主要表现：

1. 需求不够明确、不准确。
2. 需求变更风险， 即在项目的推进过程中，新的需求被挖掘出来，或当前需求需做出调整的风险。
3. 需求描述的多义性，不同开发人员对同一需求说明往往会产生不同的解读。
4. 需求开发时间不足，导致需求不完整、有遗漏。

应对方案：

1. 获取足够完整的、准确的需求。项目团队应与用户建立直接、快速的通信通道，如在项目初期向目标群体发布调查问卷，收集用户需求；对用户提出的模糊笼统的、不清楚的表述，开发者应在分析需求时发掘、揭示用户准确需求；从竞争产品入手，分析其优势与不足，丰富完善项目需求等。
2. 进行有效的需求变更管理。制定有效的变更控制流程，并形成文档，之后提出的所有变更都要根据该控制流程进行和控制；需求变更前向项目小组提出申请，成员评估后再决定需求的变更和项目计划的调整；每次需求变更都应妥善保管产生的相关文档，维护需求变更的历史纪录，记录变更日期、原因、负责人、版本号等内容。
3. 做好需求分析，测试及评审。建立需求模型能够帮助开发者理清数据、业务模式、工作流程及它们之间的关系，找出遗漏、冗余或不一致的需求；需求测试可以帮助开发者发现需求的错误、二义性、不可测性、遗漏等方面的问题，使得需求更清晰完善；在需求形成过程中分阶段评审，沟通好目标再落实到细节，保证需求质量。

风险2：技术风险

软件开发项目是一种技术密集型项目，重大的技术风险包括：

1. 软件结构体系存在问题，使完成的软件产品未能实现项目预定目标;
2. 项目实施过程中采用全新技术，由于技术本身存在缺陷造成开发出的产品性能以及质量低劣的问题
3. 软件开发过程中需要运用项目组缺乏经验的技术或专有技术

这些技术问题问题决定了项目的进度、质量，甚至成败，因此对软件项目中识别到的技术风险要有足够的重视，并采取积极的措施加以应对。

预防这种风险的办法是选用项目所必须的技术、其次，对新技术的使用要谨慎，要循序渐进，尽量采用成熟的技术方案完成软件开发工作。再次，在技术创新与技术风险之间进行平衡，并做好创新技术的研究和试验工作。需要对软件项目过程中使用的各种技术进行评估，软件项目管理在制定软件开发计划时必须考虑这些因素，并作出合理的权衡决策。

应对方案：

1. 实事求是：项目组一定要本着项目的实际要求，选用合适、成熟的技术，千万不要无视项目的实际情况而选用一些虽然先进但并非项目所必须且自己又不熟悉的技术。如果项目所要求的技术项目成员不具备或掌握不够，则需要重点关注该风险因素。在技术应用之前，需要针对相关人员开展好技术培训工作，确保团队的成员熟练掌握这些技术。

2. 慎重选择开发中运用到的技术：例如在我们的软件工程原理与实践这门课的大作业里，我们需要慎重选择编程语言，界面设计也需要我们选择合适的平台，选定的技术要成熟，不能有很大的漏洞，也不应太过于小众或晦涩，以免出现技术问题时不方便解决。
3. 平衡技术创新与技术风险：对新技术的使用要谨慎，团队应尽量采取成熟的技术方案完成软件开发工作。不能一味地追求技术上的创新，还应该考虑到技术上的风险。若团队希望在技术创新方面做的出色，就应该做好技术风险高的准备，因此要慎重对待创新与风险的平衡，以免造成项目推进到半程做不下去的情况，即费时又消耗人力，得不偿失。

风险3：进度风险

软件开发常常对项目工期有着严苛的要求，软件进度的延迟往往意味着违约或市场机会的缺失。课程中的软件开发也有相应的时间限制，项目进度控制出现问题则会导致课程任务无法及时完成，面临挂科风险。软件开发项目中影响进度的因素很多，如人为因素、技术因素、资金因素、环境因素等等。在软件开项目的实施中，人的因素是最重要的因素，技术的因素归根到底也是人的因素。

应对方案：

1. 制定项目计划。开始时的项目计划可以先制定得比较粗一些，随着项目的进展，特别是需求明确以后，项目的计划就可以进一步的明确，这时候应该对项目计划进行调整修订，通过变更手续取得开发人员的共识。
2. 项目开发过程中进行必要的测试。必要的测试是项目渐近明细的方式之一，随着项目的推进再进一步细化、调整、修正和完善。
3. 开发过程及时更新相关文档，记录项目进度、当前任务、版本管理等。软件开发组织在工期的压力下，往往放弃文档的编写与更新，结果在软件项目的晚期大量需要通过文档进行协调时，却拖累软件进度越来越慢。
4. 持续地监控，及时纠正实际进度与计划的偏离。项目进度控制是随着项目的进行而不断进行的，是一个动态过程，也是一个循环进行的过程。在计划制定时就要确定项目总进度目标与分进度目标；在项目进展的全过程中，进行计划进度与实际进度的比较，及时发现偏离，及时采取措施纠正或者预防，协调项目参与人员之间的进度关系。

风险4：安全风险

软件属于逻辑产品，很多情况下并非由于软件失效导致不安全情况出现，而是在软件正常工作时，在某种特殊条件下软硬件相互作用或由于人的使用问题导致异常情况发生。凡是与软件相关的接口、硬件状态、系统时序、人员操作、使用环境、软件自身的逻辑均属于应考虑的软件安全风险的范畴。

风险：

1. 信息泄露：随着网络化建设，用户和设备及应用程序和数据正在向私人控制区域之外迁移。这意味着，如果有人拥有正确的用户凭据，则他们将被允许进入他们请求的任何站点、应用程序或设备。这导致暴露的风险增加，从而瓦解了曾经值得信任的用户控制区域，并使许多用户面临数据泄露、恶意软件和勒索软件攻击的风险。

2. 恶意攻击：系统可用性既包括可靠性、软件复制、灾难恢复等，也包括可用性相关的安全性内容，保护系统使其免受降低可用性的恶意攻击。这样的恶意攻击叫做拒绝服务（Denial Of Service）。此类攻击会让软件某一环节无法正常运行，产生严重的后果。

应对方案：

1. 身份准入：基于严格身份验证过程，即只有经过身份验证和授权的用户和设备才能访问应用程序和数据。在本次项目中，以jaccount接口限制用户登入身份，保证用户可靠性，以此减少身份信息及其他个人信息泄露的风险。
2. 确保信息保密性：不在永久性 cookie 中存储敏感数据，不使用 HTTP-GET 协议传递敏感数据，记录详细的错误信息，及时捕捉异常现象。
3. 保障最弱环节的安全性：识别并改善系统安全性的最弱环节，直到安全性风险达到一种可接受的等级。
4. SQL注入防范：进行数据库操作的时候，对用户提交的数据必须过滤’ ; -- 等特殊字符。
5. 保护可用性：应能设置系统会话时间，防止会话劫持和重复攻击的风险。对于高度保护的应用系统，可将超时时间设置为5分钟，低风险的应用系统设置不能超过20分钟。