**迭代评估报告（简化版）**

　　　　　　　　　　　　　　　　　　评估日期：2020/10/9

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 1 | 项目名称 | 霍格沃茨地图（手机导游APP） |
| 迭代名称 | 界面原型迭代 | 实际起止日期 | 2020/9/14-2020/10/10 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）   1. 界面原型设计：使用Adobe XD完成了所有界面原型的设计，包含所有界面内容、界面样式和页面间的跳转以及相关事件的触发。界面原型设计历时整个迭代周期，经过多次讨论修改以及重新设计，最终得到第一次迭代最终版本界面原型，完成度100%，完成了所有页面的设计，质量优秀。 2. Vision文档：在小作业的基础上，结合老师的建议、助教的点评、以及小组成员多次讨论形成第一次迭代最终版本Vision文档。 3. 用例模型：在分工下部分组员完成用例模型初稿，经过小组所有成员讨论修改形成最终版本用例模型。 4. 软件需求规约：在分工下部分组员完成软件需求规约初稿，经过小组所有成员讨论修改形成最终版本软件需求规约。 5. 需求调研与分析：结合小组项目具体情况，决定自我开展需求的调研与分析，使用发放问卷的形式，收集有效问卷超过200份，并进行需求分析，形成需求调研与分析文档，并根据结果，进行了针对性的功能调整。 6. 项目架构设计：结合小组项目具体情况，额外提前开展项目的架构设计，经过多次讨论与尝试，否定了之前使用Flutter或React-Native开发前端的想法，经过慎重考虑并与老师讨论之后，决定使用从未接触过的Android原生开发本项目前端，虽然急剧增加了学习成本，但是增加了项目的可行性。后端仍旧使用之前确认的Spring Boot框架开发，数据库使用MongoDB数据库。 7. 相关技术小型项目实验：结合小组项目的具体情况，额外提前开展项目的相关技术小型项目实验。由于相比其他项目，本项目的相关技术风险更大，所以提前开展这些技术的小型demo实验，目前成功实现基于LBS的小型demo的定位实验，地图加载实验，并生成相关demo的apk进行不同型号真机测试。路线规划实验以及定位触发实验取得一些进展。 8. 数据库逻辑/概念/物理模型：结合小组项目的具体情况，额外提前开展项目的数据库逻辑/概念/物理模型设计，经过小组讨论与多次修改，设计完成一套兼顾项目需求与性能的数据库逻辑/概念/物理模型。 9. 开发总结：结合小组项目具体情况，每位组员将自己的开发总结记录成文，并汇总形成开发总结文档，以记录项目第一次迭代过程中的得失。 | | | |
| 评审/测试的结果：（执行了哪些评审和测试？评审和测试的结果如何？）   1. 进行了界面原型设计的第一次评审，发现景点详情中到此一游与景点评价形式相似，似有冗余，经过与老师沟通确认之后，明确了到此一游与评价各自的功能需求，并完成了重新设计与排版。 2. 进行了界面原型设计的第二次评审，由于组队出行可能需要拨打电话的需求，将登录界面的账号改为手机号以及使用手机号进行注册。 3. 进行了地图定位小型项目的测试，在经过几次测试后，发现了demo的权限获取、定位精度，定位信息显示等的问题，并进行修改继续测试，最终通过了不同机型上地图定位小型项目的测试。 4. 进行了地图加载小型项目的测试，发现并修改了显示比例，地图中心等的问题，最终通过了不同机型上地图加载小型项目的测试。 5. 进行了地图路线规划的小型项目的测试，发现初始定位未触发，陀螺仪无法调用等问题，仍在进行修改与调试。 6. 进行了Vision文档、用例模型、软件需求规约等文档的小组评审，并进行了修改，得到最终版本的相关文档。 | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）   1. 前端开发中，发现先前使用Flutter进行开发的方案不可行。百度地图开放平台只对Flutter提供最基本的定位服务，并不支持更多的服务请求。此后我们尝试React-native与Vue，利用其封装好的JS包进行开发，但其功能与性能仍旧无法满足要求。所以在综合考虑之后，决定放弃使用Flutter等前端框架，尝试全新的语言Android原生进行开发，因为该问题在项目开始便进行考虑，所以无需进行大量返工。 2. 页面设计时，发现到此一游与景点评价样式重复，在与老师交流之后，明确了到此一游与景点评价不同的功能要求，对这两个模块进行了重新设计，并修改了功能需求以及界面原型设计。 | | | |
| 经验和教训：   1. 将最大风险放在最高优先级考虑，这样能够减少很多不必要的返工，比如此次项目提前尝试前端的开发与相关技术的小型项目测试，由此及时改变开发框架，由Flutter更换为Android原生进行开发，避免了后期大量返工。 2. 经常进行小组评审与讨论，在工作进行中不断的与小组成员交流工作进展与方向，能够及时纠正错误，发现更多的想法，并且及时地进行修改与调整，这样能够避免一项任务进行到末尾才发现中间环节的问题而进行大量的返工。 3. 每周以及每天都应该细化迭代计划，制定每项任务的结束时间，这样能够使得所有任务按计划有序进行。 4. 在遵循计划的同时，也应该根据情况动态调整迭代计划的安排，有些任务超过预估难度可以适当增加时间以及负责人员，而有些任务与预期相比较为简单则可以适当减少时间并让负责人员参与到其他任务中。 | | | |