****

**软件测试计划**

**易拜 E-Bike——打造校园电动自行车生态社区**

|  |  |
| --- | --- |
| **撰写人：** | 黄楚丹 21312024  吴恺云21312537 |
| **学院：** | 智能工程学院 |
| **专业：** | 智能科学与技术专业 |

2024 年 4 月 21 日

**目录**

[1 项目概述 1](#_Toc166934601)

[1.1 项目背景 1](#_Toc166934602)

[1.2 项目描述 1](#_Toc166934603)

[1.3 项目目标 1](#_Toc166934604)

[1.4 项目范围 2](#_Toc166934605)

[1.5 项目通过标准 2](#_Toc166934606)

[1.6 文档目的 2](#_Toc166934607)

[2.测试目标与策略 3](#_Toc166934608)

[2.1测试目标 3](#_Toc166934609)

[2.1.1功能性目标 3](#_Toc166934610)

[2.1.2非功能性目标 3](#_Toc166934611)

[2.2测试范围 3](#_Toc166934612)

[2.3测试策略 3](#_Toc166934613)

[2.3.1整体策略 3](#_Toc166934614)

[2.3.2接口测试 4](#_Toc166934615)

[2.3.3功能测试 4](#_Toc166934616)

[2.3.4自动化测试 4](#_Toc166934617)

[2.3.5兼容性测试 5](#_Toc166934618)

[2.3.6性能测试 5](#_Toc166934619)

[2.3.7安全测试 6](#_Toc166934620)

[2.3.8回归测试 6](#_Toc166934621)

[2.3.9风险分析 7](#_Toc166934622)

[2.4测试环境 7](#_Toc166934623)

[2.4.1硬件环境 7](#_Toc166934624)

[2.4.2软件环境 7](#_Toc166934625)

[2.4.3网络环境 10](#_Toc166934626)

[2.5参考文档 10](#_Toc166934627)

[3 测试计划制定 11](#_Toc166934628)

[3.1 测试方案 11](#_Toc166934629)

[3.1.1 前端测试 11](#_Toc166934630)

[3.1.2 后端测试 11](#_Toc166934631)

[3.1.3 用户测试 11](#_Toc166934632)

[3.2 测试进度计划 11](#_Toc166934633)

[3.3 测试团队组织 13](#_Toc166934634)

[3.3.1成员角色与职责 13](#_Toc166934635)

[3.3.2沟通与协作机制 13](#_Toc166934636)

1 项目概述

## 项目背景

易拜 E-bike 项目诞生于当前大学校园内电动车使用管理的迫切需求之中。随着校园内电动车用户群体的日益扩大，充电桩与停车位的紧缺、购车渠道的可信度与性价比问题凸显，加之社会电动车安全事故频发，校内电动车管理面临诸多挑战。在此背景下，易拜 E-bike 应运而生，旨在有效协助解决上述问题，为大学校园内的电动车使用提供便捷、安全的解决方案。

本次测试计划专注于易拜 E-bike 1.0 版本的全面测试。

## 项目描述

易拜 E-bike 项目以解决当前大学生校内电动车交易、充电、维护等难题为出发点，同时响应大学生对电动车相关问题交流的强烈需求。项目致力于消除校内电动车使用中的不文明行为，构建更加和谐、有序的校园电动车生态。

本次测试的软件——易拜 E-bike，是一款针对大学校园内电动车使用管理需求而全新开发的软件。该软件所针对解决的问题在当前的软件市场中较为少见，专门为大学生提供电动车买卖、充电桩状态查询、车友交流以及不文明行为反馈等实用功能。通过全面的功能设计和精细化的用户体验优化，易拜 E-bike 旨在成为大学校园内电动车使用管理的首选工具。

## 项目目标

易拜 E-bike的核心目标在于通过所搭建的软件平台实现电动车买卖、充电桩查询、车友交流等功能的快速、便捷操作，从而减少用户在此类事务上花费的无效时间，大幅度提升校园内电动车相关事务的处理效率；不断完善和优化软件平台的功能模块，包括但不限于车辆管理、充电管理、用户反馈等，确保系统能够满足不同用户的多样化需求；针对校园内电动车使用中的充电桩不足、停车位紧张、安全问题频发等特定问题，提供切实有效的解决方案，保障用户使用的安全和便捷。

为确保易拜 E-bike软件成功实现，需确保所有规划中的功能模块均能够按时、按质完成，并经过严格的测试验证，保证功能的稳定性和可靠性，并通过用户调研和反馈收集，确保用户对软件平台的使用体验满意度达到或超过预期目标。

本项目的预期结果为完成并交付一套功能完善、性能稳定、用户友好的电动车管理软件平台。同时在平台内提供详细的用户操作手册和必要的技术支持与维护服务，帮助用户快速掌握软件平台的使用方法，确保软件平台在后续使用中能够持续稳定运行，并及时响应用户的需求和问题。

## 项目范围

项目所有功能、模块及接口。

用户群体为在校大学生。

## 项目通过标准

本项目的通过标准为系统无业务逻辑错误和二级的BUG。经确定的所有缺陷都已得到了商定的解决结果。所设计的测试用例已全部重新执行，已知的所有缺陷都已按照商定的方式进行了处理，而且没有发现新的缺陷。

注：缺陷的严重等级说明：

A：严重影响系统运行的错误；

B：功能方面一般缺陷，影响系统运行；

C：不影响运行但必须修改；

D：合理化建议

## 文档目的

1. 测试负责人根据该测试计划制定进一步的计划、安排（工作任务分配、时间进度安排）和控制测试过程；
2. 测试人员根据该测试计划中制定的范围、方法确定测试方案、设计测试用例、执行和记录测试过程并记录和报告缺陷。
3. 确定项目测试的策略、范围和方法；
4. 使项目测试工作的所有参与人员对本项目测试的目标、范围、策略、方法、组织、资源等有一个清晰的认识；
5. 使项目测试工作的所有参与人员理解测试控制过程；
6. 从策略角度说明本项目测试的组织和管理，指导测试进展，并作为项目测试工作实施的依据；
7. 本文档是本项目测试整个过程进行的依据、规范和标准。

本报告的主要读者是开发人员、测试人员、（需求）设计人员。在测试过程中严格按照本文档的制定的规范去执行。

2.测试目标与策略

## 2.1测试目标

### 2.1.1功能性目标

1. 确保软件功能符合用户需求
2. 发现各单元潜在的错误和问题
3. 验证接口的正确性
4. 验证数据的一致性和完整性
5. 验证各组件集成运行状态

### 2.1.2非功能性目标

1. 验证软件的性能
2. 验证软件可靠性、稳定性和安全性
3. 测试软件可用性

## 2.2测试范围

确定测试的覆盖范围，包括测试的模块、功能、用户场景等。

|  |  |
| --- | --- |
| 模块 | 功能 |
| 主页-广场 | 用于展示用户发布帖子 |
| 主页-热榜 | 用于展示当前热度较高帖子 |
| 主页-搜索栏 | 用于进行关键词搜索 |
| 充电 | 用于查看当前各充电桩充电状态 |
| 服务-易维修 | 用于进行电动车上门维修服务预约 |
| 服务-易反馈 | 用于向保卫处方反馈消防安全及违停问题 |
| 服务-易安行 | 用于发布电动车安全信息 |
| 服务-易转让 | 用于进行二手车交易 |
| 我的 | 用于设置个人信息及软件设置设定 |

## 2.3测试策略

### 2.3.1整体策略

1. 做到尽可能发现多的BUG；
2. 制定测试计划；
3. 完善测试用例；
4. 测试重点放在业务流程上。

### 2.3.2接口测试

|  |  |
| --- | --- |
| 测试范围 | 业务功能测试、参数验证、异常场景测试、性能测试、安全测试 |
| 测试目标 | 核实所有接口均已正常实现，即是否与接口文档结果一致 |
| 技术 | 采用黑盒测试、边界测试、等价类划分、接口安全等测试方法 |
| 工具与方法 | Postman，Fiddler、手工测试、接口自动化测试 |
| 开始标准 | 开发阶段对应的接口完成并且接口测试用例设计完成 |
| 完成标准 | 测试用例通过并且最高级缺陷全部解决 |
| 需考虑的特殊事项 | 暂无 |

### 2.3.3功能测试

|  |  |
| --- | --- |
| 测试范围 | 页面元素测试、用户交互测试、兼容性测试、性能测试、安全测试 |
| 测试目标 | 核实所有功能均已正常实现，即是否与需求一致 |
| 技术 | 采用黑盒测试、边界测试、等价类划分、接口安全等测试方法 |
| 工具与方法 | 手工测试、目测 |
| 开始标准 | 开发阶段对应的功能完成并且测试用例设计完成 |
| 完成标准 | 测试用例通过并且最高级缺陷全部解决 |
| 需考虑的特殊事项 | 暂无 |

### 2.3.4自动化测试

|  |  |
| --- | --- |
| 测试范围 | 主要业务功能，不改变业务功能 |
| 测试目标 | 使用编写脚本的方式，对业务进行回归测试 |
| 技术 | 自动化测试 |
| 工具与方法 | Python+Appium和自动化测试 |
| 开始标准 | 功能测试通过后进行自动化测试 |
| 完成标准 | 100%的测试用例执行通过并通过系统测试 |
| 测试重点与优先级 | 冒烟测试测试用例优先级高 |
| 需考虑的特殊事项 | 暂无 |

### 2.3.5兼容性测试

|  |  |
| --- | --- |
| 测试范围 | 1．使用不同机型进行测试。  2．不同操作系统和各种运行软件等各种条件的组合测试。 |
| 测试目标 | 核实系统在不同的软件和硬件配置中运行稳定 |
| 技术 | 黑盒测试 |
| 工具与方法 | 手工测试 |
| 开始标准 | 项目组移交系统测试 |
| 完成标准 | 在各种不同版本不同类项操作系统或者其组合下均能正常实现其功能（此测试根据开发提供依据决定测试范围） |
| 需考虑的特殊事项 | 暂无 |

### 2.3.6性能测试

|  |  |
| --- | --- |
| 测试范围 | 多用户长时间在线操作时性能方面的测试 |
| 测试目标 | 核实系统在大流量的数据与多用户操作时软件性能的稳定性，不造成系统崩溃或相关的异常现象 |
| 技术 | 性能测试 |
| 开始标准 | 自动化测试脚本设计并评审通过且项目组移交系统测试 |
| 完成标准 | 系统满足用户需求中所要求的性能要求 |
| 需考虑的特殊事项 | 暂无 |

### 2.3.7安全测试

|  |  |
| --- | --- |
| 测试范围 | 1.用户、管理员的密码安全  2.权限  3.非法攻击 |
| 测试目标 | 1.用户、管理员的密码管理  2. 应用程序级别的安全性：核实用户只能操作其所拥有权限能操作的功能。  3.系统级别的安全性：核实只有具备系统访问权限的用户才能访问系统。 |
| 技术 | 代码包或者非法攻击工具 |
| 工具与方法 | AppScan、Fiddler、探索性测试 |
| 开始标准 | 功能测试完成 |
| 完成标准 | 执行各种非法操作无安全漏洞且系统使用正常 |
| 测试重点与优先级 | 越权 |
| 需考虑的特殊事项 | 暂无 |

### 2.3.8回归测试

|  |  |
| --- | --- |
| 测试范围 | 所有功能、用户界面、兼容性、安全性等测试类型 |
| 测试目标 | 核实执行所有测试类型后功能、性能等均达到用户需求所要求的标准 |
| 技术 | 黑盒测试 |
| 工具与方法 | 手工测试和自动化测试 |
| 开始标准 | 每当被测试的软件或其环境改变时在每个合适的测试阶段上进行回归测试 |
| 完成标准 | 95%的测试用例执行通过并通过系统测试 |
| 测试重点与优先级 | 测试优先级以测试需求的优先级为参照 |
| 需考虑的特殊事项 | 软硬件设备问题 |

### 2.3.9风险分析

（1）测试人员对系统熟悉程度的风险

参与本项目的测试人员都是第一次接触该类型系统，在经过短期的系统培训后，仍然有可能没有完全掌握系统的业务细节，这将在后面的测试设计和测试执行工作造成一些测试逃逸现象（即一些要测试的方面没有测到）。

（2）时间方面的风险

本次项目时间只有一个月，却要完成测试规范的制定、整套测试用例的设计和执行一轮完整的测试，时间进度非常紧张，可能导致测试设计工作不够完善。

## 2.4测试环境

### 2.4.1硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 数量 | 配置 | 其他说明 |
| 手机 | 2 | HarmonyOS /iOS 512G内存 | 真机测试 |
| 电脑 | 2 | Windows 10 512G内存 | 真机测试 |
| 平板 | 2 | Android /iPadOS，512G内存 | 真机测试 |

### 2.4.2软件环境

|  |  |
| --- | --- |
| 类型 | 名称 |
| 操作系统 | HarmonyOS 4.0.0.116  iOS 13.7.14  Windows 10  Ubuntu 20.04 |
| 所需安装包 | aioredis==1.3.1  asgiref==3.8.1  async-timeout==4.0.3  attrs==23.2.0  autobahn==23.6.2  Automat==22.10.0  certifi==2024.2.2  cffi==1.16.0  channels==3.0.4  channels-redis==3.3.1  charset-normalizer==3.3.2  constantly==23.10.4  construct==2.5.3  coreapi==2.3.3  coreschema==0.0.4  cryptography==42.0.5  daphne==3.0.2  Django==4.2  django-filter==24.2  django-haystack==3.2.1  django-model-utils==4.5.0  django-utils-six==2.0  djangorestframework==3.15.0  djangorestframework-simplejwt==5.3.1  drf-yasg==1.21.7  drf-yasg2==1.19.4  elasticsearch==7.17.9  hiredis==2.3.2  hyperlink==21.0.0  idna==3.7  incremental==22.10.0  inflection==0.5.1  itypes==1.2.0  jieba==0.42.1  Jinja2==3.1.3  MarkupSafe==2.1.5  msgpack==1.0.8  packaging==24.0  pefile==2023.2.7  pillow==10.3.0  pyasn1==0.6.0  pyasn1\_modules==0.4.0  pycparser==2.22  PyJWT==2.8.0  pyOpenSSL==24.1.0  python-ptrace==0.9.9  pytz==2024.1  PyYAML==6.0.1  requests==2.31.0  ruamel.yaml==0.18.6  ruamel.yaml.clib==0.2.8  service-identity==24.1.0  six==1.16.0  sqlparse==0.5.0  Twisted==24.3.0  txaio==23.1.1  typing\_extensions==4.11.0  uritemplate==4.1.1  urllib3==1.26.18  whitenoise==6.6.0  Whoosh==2.7.4  zope.interface==6.3 |

### 2.4.3网络环境

1. 网络拓扑结构

我们的网络环境采用分层的星型拓扑结构，核心交换机位于网络的中心，连接着多个汇聚交换机。汇聚交换机再连接到各个楼层的接入交换机，为终端用户提供网络连接。此外，我们还配置了防火墙以保护内部网络免受外部威胁，并通过路由器实现与互联网的连接。

1. 网络设备

* 核心交换机：我们采用高性能的核心交换机，支持高速数据传输和多层交换，确保网络内部的高效通信。
* 汇聚交换机：汇聚交换机部署在各个关键节点，它们负责将接入交换机的数据流量进行汇聚，并转发到核心交换机。
* 接入交换机：接入交换机连接着各个用户的终端设备，如PC、打印机等，为用户提供稳定的网络连接。
* 防火墙：为保护内部网络安全，我们部署了企业级防火墙，对进出的数据流量进行过滤和监控，防止未经授权的访问和潜在的网络攻击。
* 路由器：路由器负责将内部网络与互联网连接起来，实现数据的转发和路由选择。

1. 网络配置

* IP地址分配：我们采用静态IP地址分配方式，确保每台设备都有一个固定的IP地址，便于网络管理和故障排除。
* 子网划分：根据业务需求，我们将网络划分为多个子网，以提高网络的可管理性和安全性。
* VLAN设置：通过VLAN技术，我们将不同的用户组隔离开来，减少广播风暴的发生，并提高网络的安全性。

1. 网络性能参数

* 带宽：我们的网络环境支持高达1Gbps的传输速率，确保大量数据传输的效率和稳定性。
* 延迟：网络延迟被控制在极低的水平，以满足实时性要求高的应用场景。
* 稳定性：通过冗余设计和故障切换机制，我们的网络环境具备高可用性，确保关键业务的连续运行。

## 2.5参考文档

接口文档。

3 测试计划制定

## 3.1 测试方案

### 3.1.1 前端测试

* 功能测试
* 性能测试
* 兼容性测试

### 3.1.2 后端测试

* 运行环境
* 电动车相关信息测试
* 交易相关信息测试
* 维修相关信息测试
* 安全相关信息测试
* 社交相关信息测试
* 用户相关信息测试

### 3.1.3 用户测试

* 业务流程测试
* 用户需求验证
* 用户文档和帮助信息测试

## 3.2 测试进度计划

在本项目中，我们将整个测试过程分为几个里程碑，并设定完成时间，作为整体测试进度计划，以控制整个过程。

|  |  |  |
| --- | --- | --- |
| 里程碑 | 完成标准 | 完成时间 |
| 人员培训 | 1. **测试团队培训**：测试团队已完成对所有被测系统的深入培训，对系统的业务逻辑、操作流程和潜在风险有清晰的认识；  2. **测试人员知识掌握：**测试人员通过实际操作和模拟测试，熟练掌握被测系统的具体功能和操作流程，为后续的测试工作打下坚实基础 | 2024.4.10 |
| 测试计划 | 1. **测试范围明确：**详细列出待测试的系统、功能点、性能指标等，确保测试范围全面、无遗漏；  2. **测试策略制定：**根据被测系统的特点和需求，制定合适的测试策略，包括测试方法、测试工具、测试环境等；  3. **测试时间安排：**制定详细的测试时间表，包括测试准备、测试执行、结果分析等各阶段的起止时间，确保测试进度可控；  4. **风险识别与应对：**对测试过程中可能遇到的风险进行识别，并制定相应的应对措施，降低测试风险 | 2024.4.21 |
| 测试设计 | 1. **测试用例设计完成：**根据测试需求，设计全面、有效的测试用例，确保所有功能点、性能要求和用户体验细节均被覆盖；  2. **性能测试方案确定：**制定性能测试方案，明确测试目标、测试指标、测试场景等，确保性能测试的针对性和有效性；  3. **自动化测试脚本编写：**根据测试需求，编写自动化测试脚本，提高测试效率和准确性；  4. **测试数据准备：**准备足够的测试数据，包括正常数据、异常数据和边界数据，确保测试的充分性和有效性；  5. **版本兼容设计完成：**明确不同系统版本、设备型号和应用版本的兼容性测试方案，确保应用在各种环境下均能稳定运行；  6. **安全场景设计完成：**针对应用可能面临的安全风险，设计全面的安全测试场景 | 2024.4.30 |
| 测试执行 | 1. **测试用例执行完毕：**按照测试计划，执行所有测试用例，确保测试覆盖率和测试充分性；  2. **缺陷跟踪管理：**对发现的缺陷进行详细记录、跟踪和管理，确保缺陷得到及时处理和修复；  3. **测试过程记录：**记录测试过程中的关键信息，如测试环境、测试数据、测试结果等，为后续分析和总结提供依据 | 2024.5.13 |
| 结果分析 | 1. **测试报告编写：**编写详细的测试报告，包括测试范围、测试过程、测试结果、缺陷统计等信息；  2. **测试结果评估：**对测试结果进行评估和分析，包括测试覆盖率、缺陷率、性能指标等，为产品发布和后续迭代提供有力支持；  3. **测试总结与建议：**根据测试结果，总结测试过程中的经验和教训，提出改进意见和建议，促进产品质量的持续提升。 | 2024.5.20 |

## 测试团队组织

### 3.3.1成员角色与职责

本项目的测试团队由两位核心测试人员构成：黄楚丹和吴恺云。他们各自的角色与职责如下：

* **黄楚丹：**作为测试计划制定的主要责任人，她将负责全面规划和管理测试活动，重点聚焦于后端测试，确保后端功能的稳定性和性能达标。同时，她还将负责关键时间节点的把控，确保测试进度与项目整体进度保持同步；
* **吴恺云：**作为测试过程的主要执行者，她将专注于前端测试和用户测试，确保用户界面的流畅性和用户体验的优质性。她还将细致把控测试细节，不放过任何可能影响产品质量的问题，为产品的顺利发布提供有力保障。

### 3.3.2沟通与协作机制

为确保测试团队的高效运作，我们建立了以下沟通与协作机制：

* **定期会议：**测试团队将每周进行一次例会，总结上周的测试进展，讨论所遇到的问题，并制定下周的测试计划。会议将由两位测试人员共同参与，确保信息的实时共享和问题的及时解决；
* **即时通讯：**测试团队将通过微信保持日常沟通，确保在测试过程中遇到的问题能够及时得到跟进和处理；
* **文档共享：**测试团队将使用保持定期更新测试计划、测试用例、测试报告等文档，确保团队成员能够随时查阅和更新相关信息，对齐团队内颗粒度；
* **跨部门协作：**测试团队将与开发团队、产品团队等保持紧密的合作，确保测试工作与开发进度和产品需求保持一致。在测试过程中，测试团队将及时与开发团队沟通测试进展和发现的问题，共同推动项目的顺利进行。