**《软件工程》课程考核试卷**

**（2022-2023学年第二学期）**

**学 号： 632007090116 姓 名： 罗鑫**

**专 业 班 级： 计科2001班**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **分 项** | **内容50%** | **质量30%** | **文献10%** | **格式10%** | **总分100%** |
| **得 分** |  |  |  |  |  |

**完成时间：2023 年 06月28日**

**任 课 教 师： 梁 宗 保**

**高速公路养护管理系统**

**一、项目概述**

高速公路养护管理系统是一个旨在提高高速公路运营和维护效率的项目。该系统利用先进的信息技术和管理方法，帮助高速公路管理部门监控、评估和管理路段的状态，并实施必要的养护和维修工作。

**1. 目标：**

高速公路养护管理系统的主要目标是提高高速公路的安全性、可靠性和可持续性。通过及时发现和解决潜在问题，系统可以减少事故风险，延长路面使用寿命，并提供更好的行车体验。

**2. 数据采集和监测：**

系统利用各种传感器、监控设备和无人机等技术手段，实时采集高速公路的各项数据，包括路面状况、交通流量、天气条件等。这些数据被传输到中央服务器进行处理和分析。

**3.数据分析和决策支持：**

通过对采集到的数据进行分析和建模，系统可以评估路段的健康状况、预测潜在的问题，并生成相应的报告和建议。这些信息为决策者提供支持，帮助他们制定合理的养护计划和优化资源配置。

**4.养护计划和工作管理：**

系统根据分析结果生成养护计划和工作指南，包括路面维修、标线刷新、绿化养护等。计划可以根据紧急程度和资源可用性进行优先级排序，并分配给相关的养护单位或承包商。

**5.远程监控和维修支持：**

系统允许远程监控和控制养护工作。通过传感器和视频监控设备，养护人员可以实时观察路段情况，并进行远程指导和调整。此外，系统还提供故障报警和维修支持功能，以便快速响应和解决问题。

**6.数据记录和绩效评估：**

系统记录养护过程中的各项数据，包括维修记录、材料使用情况等。这些数据可用于绩效评估和质量控制，帮助监督机构监督和评估养护单位的工作质量。

高速公路养护管理系统的实施可以提高养护工作的效率和准确性，降低事故风险，延长路面寿命，并优化资源利用。这将有助于提升高速公路的运营水平和用户体验，为社会经济发展提供可靠的交通基础设施。

**二、可行性分析**

**1. 技术可行性**：

XYZ国家拥有先进的信息技术基础设施和通信网络，使得数据采集、传输和处理成为可能。市场上也存在成熟的高速公路养护管理系统解决方案。因此，技术可行性是存在的，系统的实施和运行是可行的。

**2. 经济可行性：**

系统实施的成本包括硬件设备、软件开发、系统集成、培训和维护等方面。同时，预计系统将带来一系列经济效益，如减少事故风险、降低维修成本和延长路面寿命等。通过对成本和效益进行综合评估，可以得出系统在经济上的可行性。

**3. 法律和政策可行性：**

在XYZ国家，相关的法律法规和政策支持高速公路养护工作的改善和现代化。养护管理系统需要遵守数据隐私保护、网络安全等方面的法规，确保数据的安全和合规。同时，政府部门对高速公路养护的发展和改进具有积极的态度和支持。

**4. 运营可行性：**

系统的运营需要合理的组织架构和充足的人力资源。XYZ国家已经建立了相应的高速公路管理部门和养护单位，这为系统的运营提供了基础。同时，需要进行系统培训和运维支持，确保系统能够顺利运行和持续发展。

**5. 社会和环境可行性：**

养护管理系统的实施将提高高速公路的安全性和可靠性，减少事故发生率，改善行车体验，促进经济发展。同时，系统的智能化管理和优化资源配置有助于减少能源消耗和环境污染，符合可持续发展的要求。

**三、需求分析**

**1.道路需求：**

a. 系统应能够管理和记录高速公路网络的拓扑结构和路段信息，包括路段长度、车道数、入口和出口等。

b. 系统需要能够标识和记录路段的维护周期、历史维修记录和维护要求，以便制定养护计划。

**2.用户管理需求：**

a. 系统应具备用户管理功能，支持不同权限级别的用户登录和访问控制。

b. 系统管理员应能够管理用户账户、权限设置和角色分配。

c. 用户登录后应能够根据其权限级别访问相应的功能和数据。

**3.养护任务需求：**

a. 系统应能够接收和记录养护任务的派发和分配情况。

b. 养护任务应包括维修工作、清洁工作、标线刷新等不同类型的任务。

c. 系统需要记录任务执行的时间、人员和材料使用情况，并生成任务完成报告。

**4.养护资源管理需求：**

a. 系统应能够管理养护资源，包括人力资源、设备和材料等。

b. 系统需要记录资源的可用性和分配情况，以便合理规划养护任务和资源调度。

c. 系统应提供库存管理功能，记录材料的库存量和使用情况，方便补充和采购。

**5.养护质量控制需求：**

a. 系统需要支持养护质量的评估和控制。

b. 养护质量标准应定义清晰，并记录在系统中供参考。

c. 系统应提供养护质量检查和验收功能，记录检查结果和问题反馈。

**6.统计和报表需求：**

a. 系统应能够生成各类统计数据和报表，包括养护任务完成情况、资源利用率、维修成本等。

b. 统计和报表应具备可定制化和导出功能，满足不同用户的需求。

**四、设计说明**

**1. 系统架构设计：**

采用分层架构，包括前端用户界面、应用层、数据层和基础设施层。前端用户界面提供用户交互界面，支持用户登录、数据展示和操作等功能。应用层负责业务逻辑处理，包括数据处理、分析、养护任务管理和报表生成等功能。数据层用于存储和管理系统的数据，包括路段信息、用户数据、任务数据和统计数据等。基础设施层提供系统所需的硬件和软件基础设施，包括服务器、数据库、网络等。

**2. 数据库设计：**

设计路段信息表，包括路段编号、长度、车道数等基本属性。设计用户表，包括用户ID、用户名、密码和权限等信息。设计任务表，包括任务ID、任务类型、执行时间、执行人员等信息。设计资源表，包括人力资源、设备和材料的信息，记录资源的可用性和分配情况。

**3. 用户界面设计：**

提供用户登录界面，支持用户身份验证和访问控制。设计可视化界面，用于展示路段状况、数据分析结果和养护任务状态等。提供任务派发和管理界面，支持任务的创建、分配和执行情况的跟踪。提供报表和统计界面，展示养护任务完成情况、资源利用率和维修成本等信息。

**4. 数据处理和分析模块设计：**

设计数据采集模块，负责实时采集路面状况、交通流量、天气数据等。设计数据处理模块，包括数据清洗、转换和存储，为后续的数据分析提供准备。设计数据分析模块，基于采集的数据进行路面健康状况评估、问题预测和报告生成等分析任务。

**5. 养护计划和工作管理模块设计：**

设计养护计划生成模块，基于数据分析结果生成养护计划和工作指南。设计任务分配模块，根据养护计划将任务分配给相应的养护单位或承包商。设计任务跟踪模块，记录任务执行情况，包括时间、人员和材料使用等。设计任务完成报告生成模块，生成任务完成报告和统计数据，供监督机构参考。

**6. 安全和隐私保护设计：**

采用安全加密协议，保护数据传输和存储的安全性。实施身份验证和访问控制，确保只有授权用户能够访问系统的敏感数据和功能。严格遵守相关隐私法规，保护用户和驾驶员的个人隐私信息。

**7. 扩展性和可维护性设计：**

设计系统模块化，使得各个模块可以独立开发、测试和维护。使用标准化的接口和协议，方便与其他系统进行集成。考虑系统的扩展性，能够适应未来业务需求的变化和系统规模的增长。

**五、测试用例说明**

**设计测试用例：**

根据高速公路养护系统的需求分析，可以设计以下4个测试用例模块，并实现其中的2个模块。

高速公路养护系统肯定是需要添加道路信息，我们才能进行管理，我们也要设计用户的管理，这样才能登录系统。我们还要设计查看所有的高速公路，我们才能更好的进行高速公路管理，最后我们还可以设计一个简单的前端首页进行可视化。

我们可以实现其中的添加道路信息和用户的增删改查和前端页面的简单设计。

**1. 添加道路信息**

@RestController

@RequestMapping("/roads")

public class RoadController {

@Autowired

private RoadRepository roadRepository;

@PostMapping

public ResponseEntity<Road> addRoad(@RequestBody Road road) {

Road savedRoad = roadRepository.save(road);

return new ResponseEntity<>(savedRoad, HttpStatus.CREATED);

}

}

首先使用@RestController注解标记了一个控制器类RoadController，然后使用@RequestMapping注解指定了该控制器类处理的请求路径为/roads。

接着在RoadController类中，我们使用@Autowired注解注入了一个RoadRepository对象，以便我们可以在添加道路信息的接口中使用该对象来保存道路信息到数据库中。

最后，在RoadController类中，定义了一个addRoad方法，该方法使用@PostMapping注解标记，表示该方法处理HTTP POST请求。在该方法中，我们使用@RequestBody注解获取HTTP请求中的道路信息，并将其保存到数据库中。最后，我们将保存成功的道路信息返回给客户端，并使用HttpStatus.CREATED表示HTTP响应状态码为201 CREATED。

测试过程：

使用postman进行接口测试，测试结果为

{

"success": true,

"code": 20000,

"message": "成功",

"data":null,

}

则表示添加成功。

**2. 用户管理接口**

@RestController

@RequestMapping("/users")

public class UserController {

private List<User> users = new ArrayList<>();

@PostMapping

public User createUser(@RequestBody User user) {

users.add(user);

return user;

}

@DeleteMapping("/{username}")

public void deleteUser(@PathVariable String username) {

User user = findUserByUsername(username);

if (user != null) {

users.remove(user);

}

}

@PutMapping("/{username}/promote")

public void promoteToAdmin(@PathVariable String username) {

User user = findUserByUsername(username);

if (user != null) {

user.setAdmin(true);

}

}

@PostMapping("/login")

public boolean authenticateUser(@RequestParam String username, @RequestParam String password) {

User user = findUserByUsername(username);

return user != null && user.getPassword().equals(password);

}

private User findUserByUsername(String username) {

for (User user : users) {

if (user.getUsername().equals(username)) {

return user;

}

}

return null;

}

}

**3.前端首页页面**

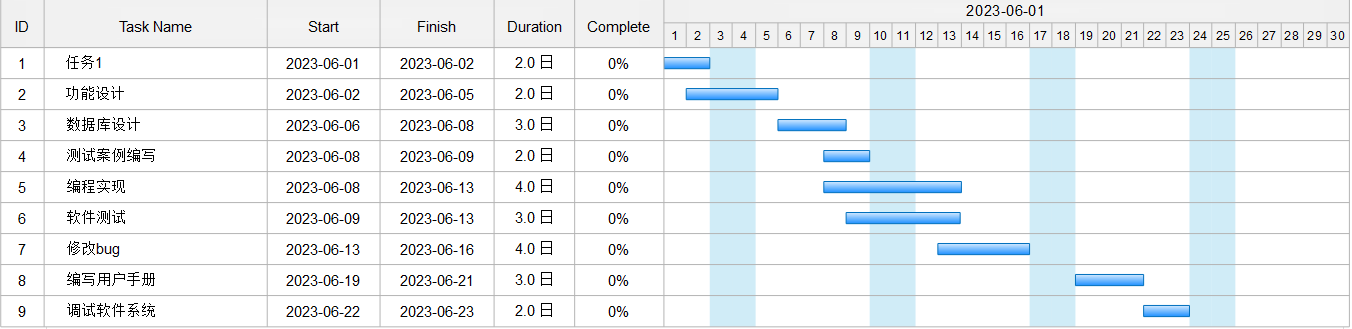


**六、项目管理**

**1. 项目进度管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 活动名称 | 持续周期 | 活动资源 | 前导活动 |
| A.需求分析 | 3天 | 将高速公路养护系统进行需求分析 |  |
| B.功能设计 | 2天 | 分析系统需要哪些功能 | A |
| C.数据库设计 | 2天 | 根据需求分析和功能进行数据库建表 | A,B |
| D.测试案例编写 | 2天 | 简单测试功能 | A,B,C |
| E.编程实现 | 5天 | 将功能详细的实现 | A,B,C |
| F.软件测试 | 3天 | 将系统的功能进行测试，提出bug | A,B,C |
| G.修改bug | 5天 | 将测试测出来的BUG进行修改完善 | A,B,C,D,E,D |
| H.编写用户手册 | 2天 | 编写系统使用说明 | D |
| I.调试软件系统 | 2天 | 调试软件系统 | A,B,C,D,E,F,G,H |

**甘特图：**



**2. 质量管理**

**a.需求管理：**

确保所有需求经过验证和审查，与相关利益相关者进行沟通和确认，以确保准确理解和明确需求。

使用需求跟踪工具跟踪和管理需求，确保需求的追踪性和一致性。

**b.设计评审：**

在设计阶段进行评审会议，邀请系统设计人员、开发人员和质量控制专家参与。检查系统设计文档、架构图和数据模型，确保设计满足性能、安全和可扩展性要求。

**c.编码规范：**

制定编码规范和标准，如命名约定、代码风格和注释规范等。进行代码审查，使用静态代码分析工具检查代码质量和潜在问题。

**d.单元测试：**

开发人员编写单元测试用例，并使用自动化测试工具进行测试。确保每个模块和函数的功能正确性和覆盖率。

**e.综合测试：**

进行系统级功能测试，确保系统的各个模块和功能能够协同工作。进行性能测试，验证系统的响应时间、吞吐量和并发能力。

**3. 风险管理**

**a.风险识别：**

通过与项目团队和利益相关者的沟通，识别潜在的风险事件。这可以包括技术风险、进度风险、需求变更风险、人力资源风险等。利用经验教训库、案例分析和专家意见，辅助风险识别工作。

**b.风险评估：**

对识别的风险进行定性和定量评估，评估其可能性、影响程度和优先级。使用**风险矩阵或其他评估工具，对风险进行分类和排序。**

**c.风险响应策略制定：**

针对每个风险事件，制定相应的风险响应策略。常见的风险响应策略包括避免、减轻、转移和接受。开发具体的应对措施和计划，以降低风险的可能性或影响程度。

**e.风险监控与控制：**

建立风险监控机制，定期跟踪和监测风险的发生和变化。根据风险情况，采取相应的控制措施，确保风险处于可接受的范围内。持续更新风险登记册和风险管理计划，记录风险状态和应对情况。

**七、参考文献：**

[1].Wang, Y., Huang, H., & Jiang, F. (2017). Development of an Intelligent Highway Maintenance Management System. Journal of Intelligent & Robotic Systems, 85(3), 475-490.

[2].Sivakumar, A. I., & Rajkumar, R. (2018). A Review on Highway Maintenance Management Systems. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 4(3), 19-26.

[3].胡光富, 陈晓玉, 王朝, 等. 基于无线传感器网络的高速公路养护系统研究[J]. 公路交通科技, 2018, 35(5): 59-62.

[4].孙瑞红, 唐继丹. 基于物联网的高速公路养护管理系统设计与实现[J]. 现代交通技术, 2018, 35(6): 93-97.

[5].王帅, 毛成林, 王晨. 基于BIM的高速公路养护管理系统设计与实现[J]. 中国公路学报, 2019, 32(6): 148-153.

[6].张凯, 王学江, 杨杰. 高速公路养护管理系统性能评估方法研究[J]. 交通科技与经济, 2020(3): 128-132.

[7].刘佳乐, 孙胜林, 孙波. 基于云计算的高速公路养护信息管理系统研究与设计[J]. 中国公路学报, 2020, 33(10): 133-139.

[8].高勇, 宋彩霞, 刘朝霞, 等. 基于互联网+的高速公路养护管理系统设计与实现[J]. 公路交通科技, 2021, 38(2): 117-121.