《C++程序设计课程设计》答辩记录表

|  |  |  |  |
| --- | --- | --- | --- |
| **学年-学期** | 2024-2025-2 | **选课课号** | **62723** |
| **姓名** | **学号** | **班级** | **任课教师** |
| 李凯涛 | 2023327100056 | 23计科4班 | 霍成文、庄巧莉 |
| **题目** | 基于快递物件管理系统开发 | | |
| **答辩时间 (年/月/日)** | 2025/6/11 | **答 辩 地 点** | 10-605 |
| 答辩记录(可另附页)：  问题1：取件码生成规则不完善，纯随机生成，未考虑快件特性。  回答1：建议采用分类型生成规则，将快件特性（如尺寸、重量）编码至取件码中：结构设计上，取件码由3部分组成（2位类型码 + 3位货架区位码 + 3位随机校验码），例如：L2-05B-8X3。类型码：L1（小件）、L2（中件）、L3（大件）；区位码：对应仓库物理分区（如05B表示第5排B区）。校验机制上，通过哈希算法（如CRC8）校验区位码有效性，避免分配冲突。  **技术实现**： C++示例：基于快件类型生成区位码  std::string generatePickupCode(ParcelType type, int zone) {  std::string typePrefix = (type == SMALL) ? "L1" : (type == MEDIUM) ? "L2" : "L3";  char zoneCode = 'A' + (zone % 26);  return typePrefix + "-" + std::to\_string(zone/26) + zoneCode + "-" + generateRandomSuffix();  }  问题2：历史记录查询功能缺失，只能查询当天记录，不能查询具体某天  回答2：需在数据库设计中增加时间索引并优化查询接口：数据库改造上，为parcels表添加timestamp字段，建立复合索引(user\_id, timestamp)；使用日期分区表（如按周分区）提升查询效率。缓存策略上，对高频查询日期（如最近7天）使用Redis缓存；采用LRU策略管理缓存淘汰。  **查询接口**：SQL-- 支持时间范围的查询示例  SELECT \* FROM parcels  WHERE user\_id = ? AND timestamp BETWEEN ? AND ?  ORDER BY timestamp DESC LIMIT 100;  问题3：快件ID生成机制不合理，手动输入ID容易重复，且没有格式要求  回答3：推荐使用分布式ID生成算法，确保全局唯一性与可读性：ID结构上，<节点号><时间戳><序列号>（例：N1-20250615-00042）；节点号：区分不同代收点（2位）；时间戳：精确到秒（YYYYMMDDHHMMSS）；序列号：当日自增（5位，每日清零）。冲突检测上，在数据库层设置UNIQUE约束；插入前预校验（SELECT FOR UPDATE。  **技术实现**：C++示例：基于Snowflake变体算法  std::string generateParcelID(int nodeId) {  static std::atomic<uint32\_t> sequence(0);  time\_t now = time(nullptr);  return fmt::format("N{:02d}-{:%Y%m%d}-{:05d}",  nodeId, \*std::localtime(&now), ++sequence % 100000);  } | | | |

自评A