

# 软测单元测试报告

## 1 引言

在医疗健康领域，糖尿病管理尤其需要依赖精确和实时的数据来指导患者的饮食和生活方式调整。本项目旨在开发一款集成化的糖尿病饮食管理与营养评估系统，该系统能够为糖尿病患者提供准确的食物营养信息和个性化的饮食建议。系统的核心功能包括利用先进的图像处理技术自动识别食物成分，评估营养价值，并结合糖尿病管理指南为患者提供实用的饮食指导。为确保该系统的准确性和实用性，进行全面而细致的软件测试成为确保项目成功的关键步骤。

### 1.1 目的

软件测试任务的重要性在于验证和确保系统功能的正确性、可靠性和用户友好性。

本文档将详细说明软件测试的目标、策略、方法及期望结果，以确保糖尿病饮食管理与营养评估系统在实际部署前达到设计和功能要求，为糖尿病患者提供一个可靠和有效的健康管理工具。通过严格的测试流程，我们致力于推动项目的成功实施，从而为糖尿病患者带来更优质的生活质量和健康状况。

### 1.2 测试策略

在本项目中，单元测试的执行策略基于类和方法的选择。首先，我们对每个类中的方法根据非空非注释代码行数（LOC）超过20或复杂度超过3的标准来选择需要进行单元测试的方法，确保关键且复杂的功能得到充分的验证。对于这些方法，我们采用独立的单元测试策略，通过设计特定的驱动程序和桩来实现，这样可以模拟依赖的模块，专注于测试目标类的功能。对于继承关系中的子类，我们采用分层增量测试策略，仅为子类中新增或变更的部分设计新的测试用例，而与父类相同的功能则重用已有的父类测试用例。测试的执行顺序按照《软件详细设计说明书》中的用例实现交互图来确定，从最少依赖关系的类开始，逐步扩展到依赖关系较强的类，这样可以系统地覆盖所有关键组件，并确保每个组件在集成前独立运行正常，保障了整个系统的稳定性和可靠性。通过这种综合的方法论，我们能够有效地发现并修正潜在的错误，从而提高软件的质量和性能。

### 1.3 范围

本单元测试计划作为整个软件开发项目的一部分，包含计划阶段、设计阶段、实现阶段和执行阶段四个阶段。该计划主要处理与糖尿病饮食管理与营养评估系统单元测试有关的任务安排、资源需求、人力需求、风险管理、进度安排等内容。

### 1.4 参考文献

- 《软件需求规格说明书（Software Requirement Specification）》
- 《软件详细设计说明书（Software Design Descriptions）》

- 《用户界面规格说明书（User Interface Specification）》

## 2 测试内容

根据《软件详细设计说明书》中的详细设计内容，单元测试的测试项目如以下所示：

- 用户管理模块
- 饮食记录模块
- 血糖记录模块
- 食谱模块

## 3 测试方法

总体上根据类规约和操作规约构建测试用例；

根据实际情况合理选择使用边界值分析法、等价类测试法、决策表测试法、正交实验法等黑盒测试方法以及语句覆盖、路径覆盖等白盒测试方法来构建基本的测试用例。

对于具有特殊需求的某些类可以辅以下面两种方法构造测试用例以做补充：

1. 状态转移测试方法：对于状态转换的类可以使用状态转移测试的方法，即通过类的状态图建立状态转移树，进而构造出补充的测试用例；
2. 基于实现的测试方法：借助传统逻辑覆盖测试法、数据流分析法等白盒测试方法构造补充的测试用例，以对程序的逻辑结构或数据流进行测试，来达到一定的代码覆盖率。

## 4 测试通过/失败标准

测试通过的标准表述如下：

1. 所有单元测试的用例都被执行并通过；
2. 所有发现的缺陷都被修正并通过回归测试；
3. 所有被测对象的前置条件和后置条件组合覆盖率达到100%，或能明确给出不需要达到的理由；
4. 单元测试报告被授权人批准。

测试失败标准表述如下：

1. 严重缺陷密度大于15个/KLOC；
2. 发现软件结构有重大设计问题，其修改会导致20%以上的接口、功能、数量的变化，进一步测试相关特性已经无意义；
3. 发现关键功能未被设计，该功能的设计会导致20%以上的接口、功能、数量的变化，进一步测试相关特性已经无意义。

测试结果审批过程：开发人员提交单元测试报告—开发或测试经理签字并提交 SQA→SQA对报告进行

评审并签字（测试经理参与）→产品经理签字。

## 5 测试挂起/恢复的条件

测试挂起的条件有：

1. 当某个类在单元测试执行过程中发现有阻塞用例的时候，该类的单元测试被挂起。
2. 当有20%以上的被测类都遇到有阻塞用例时，所有类的单元测试都被挂起。
3. 当出现有新增需求的时候，与该需求相关的所有类的单元测试都被挂起。
4. 当开发人员提出要进行设计变更的时候，相关类的单元测试将被挂起。

测试恢复的条件有：

1. 测试被挂起的条件已经被解决。
2. 需要恢复测试的对象达到单元测试入口条件，在这里要求这些被测对象已经通过代码走读（要提交走读报告）和语法检查（要提交检查结果）。

## 6 单元测试交付物

- 单元测试计划（Unit Test Plan）；
- 单元测试设计规格（Unit Test Design Specification）；
- 单元测试用例规格（Unit Test Case Specification）；
- 单元测试用例脚本；
- 单元测试驱动和桩代码；
- 单元测试执行日志（Unit Test Log）；
- 单元测试报告（Unit Test Report）。

## 7 角色和职责

角色	职责
测试经理	负责规划、协调、监控整个测试过程，确保按时交付高质量的软件产品
SQA	负责制定质量标准、监控开发流程、管理风险与缺陷，并通过持续的审查和改进确保软件产品的质量
开发及测试人员	完成单元测试需要的输入，完成具体的单元测试设计规格、单元测试用例规格的制定，执行单元测试，记录发现的问题，并负责问题的回归测试，最后解决问题

# 8 单元测试任务及规划

任务标识	任务描述	责任人	优先级	依赖关系	周期/天
UT_TASK_001	单元测试计划制定	测试经理	高		2
UT_TASK_002	单元测试计划评审	SQA	中	UT_TASK_001	2
UT_TASK_003	单元测试计划修改	测试经理	中	UT_TASK_001	2
UT_TASK_004	单元测试设计规格制定	开发及测试人员	中	UT_TASK_001	2
UT_TASK_005	单元测试设计规格评审	SQA	中	UT_TASK_004	5
UT_TASK_006	单元测试设计规格修改	开发及测试人员	中	UT_TASK_005	2
UT_TASK_007	单元测试用例规格设计	开发及测试人员	中	UT_TASK_006	2
UT_TASK_008	单元测试用例规格评审	SQA	中	UT_TASK_007	3
UT_TASK_009	单元测试用例规格修改	开发及测试人员	高	UT_TASK_008	2
UT_TASK_010	驱动、桩、用例脚本代码实现	开发及测试人员	低	UT_TASK_009	6
UT_TASK_011	驱动、桩、脚本代码检查	SQA	低	UT_TASK_010	2
UT_TASK_012	驱动、桩、脚本代码修改	开发及测试人员	低	UT_TASK_011	3
UT_TASK_013	单元测试执行及回归	开发及测试人员	中	UT_TASK_012	2
UT_TASK_014	单元测试报告	测试经理	中	UT_TASK_013	2
UT_TASK_015	单元测试报告审批	开发及测试人员	高	UT_TASK_014	1

# 9 环境需求

## 9.1 测试工具

- 使用CheckStyle检查代码格式规范
- 使用JUnit进行测试
- 使用Allure生成可视化单元测试报告

## 9.2 运行环境

操作系统	Windows
JDK版本	17
JUnit版本	5.11.0-M2

## 9.3 数据库环境

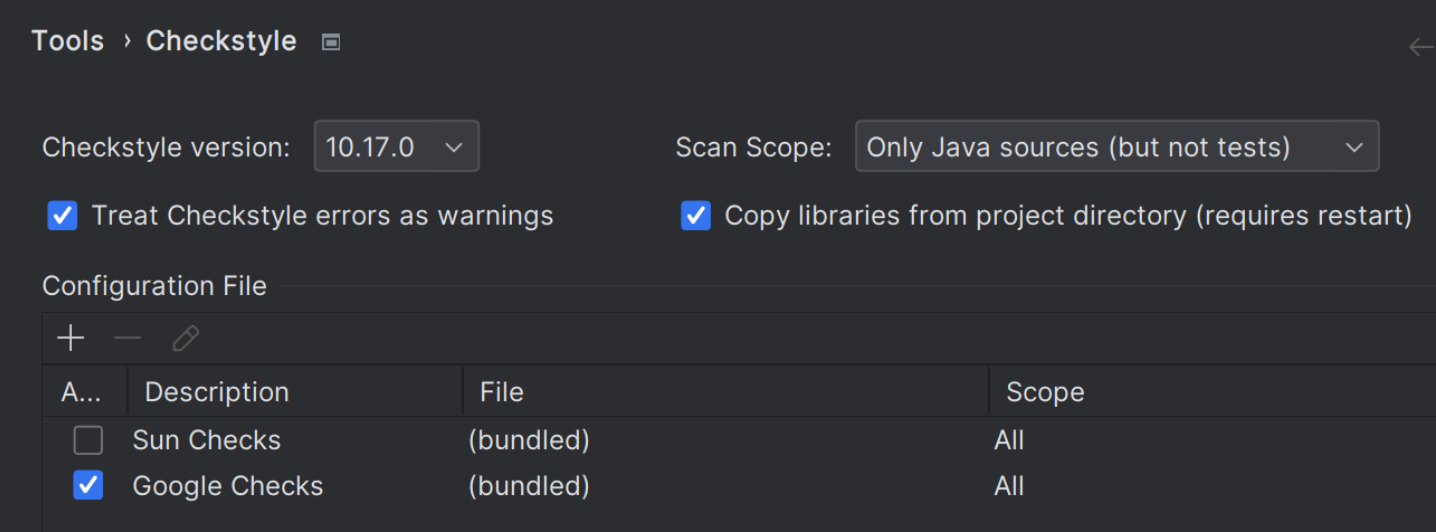
为了保证测试时数据库环境的一致性，保存2024年6月10日13时59分04秒时的数据库备份作为测试的数据库环境。

# 10 代码静态测试

我们的项目团队采用了Checkstyle静态分析工具对代码进行了检测。这两种工具虽然都是用于代码的静态测试，但它们的侧重点和实现方式存在一定的差异。

Checkstyle 主要关注于代码风格的规范化和一致性。它通过特定的配置文件，定义了一系列的代码风格规则，例如Java编码惯例、命名规范、代码注释的规则等，并将这些规则应用于代码库中以检查代码是否符合这些规定。使用Checkstyle可以显著提高代码的可读性和可维护性，因为它确保了项目中所有代码的风格一致性。此外，Checkstyle 还支持自定义规则，允许团队根据自己的需要调整检查标准，包括但不限于括号的使用、行长度、方法数量等方面。Checkstyle 的集成也相对容易，可以直接集成到常见的IDE和构建工具中，如Eclipse、IntelliJ IDEA和Maven，从而在开发过程中实时提醒开发者注意风格和规范问题。

在我们的项目中，我们选用了 `Google` 的代码规范来进行审视。



如下抽选了几个对我们的代码的静态检测结果：

ProblemsFile 59Project ErrorsServer-Side AnalysisVulnerable Dependencies

GlucoseServiceImpl.java

D:\temporary files\课程\大三下\2022-2023第二学期软件测试课程资料\project\期末项目\backend\softwaretesting\service\food-service\src\main\ja

Checkstyle: 导入语句'com.baomidou.mybatisplus.extension.service.impl.ServiceImpl' 字典顺序错误。应在'com.tongji.service.service.IGlucoseService'之前。 :12

Checkstyle: 导入组之前的额外空行 'java.util.List'。 :15

Checkstyle: 导入语句'java.util.List' 字典顺序错误。应在'org.springframework.stereotype.Service'之前。 :15

Checkstyle: Javadoc的第一句缺少一个结束时期。 :17

Checkstyle: 本行字符数 104个，最多： 100个。 :26

Checkstyle: 'method def modifier' 缩进了4个缩进符，应为2个。 :29

Checkstyle: 名称 'timeRangeDTO' 中不能出现超过 '1' 个连续大写字母。 :30

Checkstyle: 'if' 缩进了8个缩进符，应为4个。 :31

Checkstyle: 本行字符数 111个，最多： 100个。 :31

Checkstyle: 'if' 子元素缩进了12个缩进符，应为6个。 :32

ProblemsFile 202Project ErrorsServer-Side AnalysisVulnerable Dependencies

RecordServiceImpl.java

D:\temporary files\课程\大三下\2022-2023第二学期软件测试课程资料\project\期末项目\backend\softwaretesting\service\food-service\src

Checkstyle: 不应使用 '.\*' 形式的导入 - com.tongji.model.dto.\* 。 :7

Checkstyle: 导入语句'com.baomidou.mybatisplus.extension.service.impl.ServiceImpl' 字典顺序错误。应在'com.tongji.service.service.IRecordService'之前。 :19

Checkstyle: 导入组之前的额外空行 'java.math.BigDecimal'。 :28

Checkstyle: 导入语句'java.math.BigDecimal' 字典顺序错误。应在'org.springframework.web.client.RestTemplate'之前。 :28

Checkstyle: 导入语句'java.util.ArrayList' 字典顺序错误。应在'org.springframework.web.client.RestTemplate'之前。 :29

Checkstyle: 导入语句'java.util.Comparator' 字典顺序错误。应在'org.springframework.web.client.RestTemplate'之前。 :30

Checkstyle: 导入语句'java.util.List' 字典顺序错误。应在'org.springframework.web.client.RestTemplate'之前。 :31

Checkstyle: 导入语句'java.util.Objects' 字典顺序错误。应在'org.springframework.web.client.RestTemplate'之前。 :32

Checkstyle: Javadoc的第一句缺少一个结束时期。 :34

Checkstyle: 'member def modifier' 缩进了4个缩进符，应为2个。 :46

Checkstyle: 'member def modifier' 缩进了4个缩进符，应为2个。 :48

Checkstyle: 'member def modifier' 缩进了4个缩进符，应为2个。 :50

Checkstyle: 'method def modifier' 缩进了4个缩进符，应为2个。 :54

Checkstyle: 名称 'timeRangeDTO' 中不能出现超过 '1' 个连续大写字母。 :55

Checkstyle: 'if' 缩进了8个缩进符，应为4个。 :56

Checkstyle: 'if' 子元素缩进了12个缩进符，应为6个。 :57

CheckStyleScan

Rules: Google Checks

Checkstyle found 61 item(s) in 1 file(s)

GlucoseServiceImpl.java : 61 item(s)

导入语句'com.baomidou.mybatisplus.extension.service.impl.ServiceImpl' 字典顺序错误。应在'com.tongji.service.service.IGlucoseService'之前。 (12:1) [CustomImportOrder]

导入组之前的额外空行 'java.util.List'。 (15:1) [CustomImportOrder]

导入语句'java.util.List' 字典顺序错误。应在'org.springframework.stereotype.Service'之前。 (15:1) [CustomImportOrder]

Javadoc的第一句缺少一个结束时期。 (17:0) [SummaryJavadoc]

本行字符数 104个，最多： 100个。 (26:0) [LineLength]

'method def modifier' 缩进了4个缩进符，应为2个。 (29:5) [Indentation]

名称 'timeRangeDTO' 中不能出现超过 '1' 个连续大写字母。 (30:51) [AbbreviationAsWordInName]

'if' 缩进了8个缩进符，应为4个。 (31:9) [Indentation]

'if' 子元素缩进了12个缩进符，应为6个。 (32:13) [Indentation]

'if rcurlly' 缩进了8个缩进符，应为4个。 (33:9) [Indentation]

'if' 缩进了8个缩进符，应为4个。 (34:9) [Indentation]

'if' 子元素缩进了12个缩进符，应为6个。 (35:13) [Indentation]

'if rcurlly' 缩进了8个缩进符，应为4个。 (36:9) [Indentation]

'method def' 子元素缩进了8个缩进符，应为4个。 (37:9) [Indentation]

'method def' 子元素缩进了8个缩进符，应为4个。 (38:9) [Indentation]

'.' 应另起一行。 (39:75) [SeparatorWrap]

'method def' 子元素缩进了8个缩进符，应为4个。 (42:9) [Indentation]

'method def rcurlly' 缩进了4个缩进符，应为2个。 (43:5) [Indentation]

'method def modifier' 缩进了4个缩进符，应为2个。 (45:5) [Indentation]

'if' 缩进了8个缩进符，应为4个。 (47:9) [Indentation]

'if' 子元素缩进了12个缩进符，应为6个。 (48:13) [Indentation]

'if rcurlly' 缩进了8个缩进符，应为4个。 (49:9) [Indentation]

'method def' 子元素缩进了8个缩进符，应为4个。 (50:9) [Indentation]

# 11 测试用例分析与设计

.....(正文 主要部分).....

## 11.1 用户管理模块

## 11.2 饮食记录模块

被测函数详见下表

标识符	名称	代码行（LOC）
UT_TD_002_001	ResponseResult getRecord(TimeRangeDTO timeRangeDTO)	19
UT_TD_002_002	ResponseResult deleteRecord(Long id)	15
UT_TD_002_003	ResponseResult updateRecord(RecordDTO recordDTO)	16
UT_TD_002_004	ResponseResult addRecord(RecordAddDTO recordAddDTO)	15

### 11.2.1 getRecord测试分析与设计

#### 11.2.1.1 标识符定义

UT\_TD\_002\_001

#### 11.2.1.2 被测特性

- 输入时间范围为空时，返回时间范围不能为空的提示消息；
- 输入的时间范围合法时，返回用户在该时间范围内的所有记录，并按创建时间排序。

#### 11.2.1.3 测试方法

时间范围timeRangeDTO参数的等价类划分考虑为空、时间范围内无记录、时间范围内有记录三种情况。

#### 11.2.1.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_002_001_001	输入时间范围为空的情况	低
	输入时间范围内无饮食记录的情况	高



UT_TD_002_001_002		
UT_TD_002_001_003	输入时间范围内有饮食记录的情况	高

11.2.1.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

11.2.1.6 对应用例

测试项编号	UT_TD_002_001_001	
优先级	低	
测试项描述	测试输入时间范围为空的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	timeRangeDTO=(null,null)	返回时间范围不能为空的提示消息

测试项编号	UT_TD_002_001_002	
优先级	高	
测试项描述	测试输入时间范围内无记录的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	timeRangeDTO=(2019-12-30 01:00:00, 2019-12-30 02:00:00)	返回空列表

测试项编号	UT_TD_002_001_003	
优先级	高	
测试项描述	测试输入时间范围内有记录的情况	
前置条件	用户登录系统并进入“饮食记录”页面	



用例序号	输入	期望结果
001	timeRangeDTO=(2021-08-05 16:00:00, 2021-08-05 17:00:00)	返回按创建时间排序的记录列表 ResponseResult.okResult(sorted RecordDTOList)

## 11.2.2 deleteRecord测试分析与设计

### 11.2.2.1 标识符定义

UT\_TD\_002\_002

### 11.2.2.2 被测特性

- 输入ID为空时，返回ID不能为空的提示消息；
- 输入的ID在当前用户的记录中不存在时，返回记录不存在的提示信息；
- 输入的ID在当前用户的记录中存在时，成功删除该记录并返回删除成功的提示信息。

### 11.2.2.3 测试方法

ID参数的等价类划分为空、记录不存在、记录存在三种情况。

### 11.2.2.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_002_002_001	输入ID为空的情况	低
UT_TD_002_002_002	输入ID在当前饮食数据不存在的情况	高
UT_TD_002_002_003	输入ID在当前饮食数据存在的情况	高

### 11.2.2.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

### 11.2.2.6 对应用例

测试项编号	UT_TD_002_002_001
优先级	低
测试项描述	测试输入ID为空的情况

前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	id = null	ResponseResult.errorResult(400, "id不能为空")

测试项编号	UT_TD_002_002_002	
优先级	高	
测试项描述	输入ID在当前饮食数据不存在的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	id = 500L	ResponseResult.errorResult(400, "该饮食数据不存在")

测试项编号	UT_TD_002_002_003	
优先级	高	
测试项描述	输入ID在当前饮食数据存在的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	id=1L	返回 ResponseResult.okResult("删除成功")

### 11.2.3 updateRecord测试分析与设计

#### 11.2.3.1 标识符定义

UT\_TD\_002\_003

#### 11.2.3.2 被测特性

- 输入的 RecordDTO 中 ID、创建时间或类型为空时，返回相应的提示消息；
- 输入的 ID 在当前用户的记录中不存在时，返回记录不存在的提示信息；
- 输入的 ID 在当前用户的记录中存在时，成功更新该记录并返回修改成功的提示信息。

11.2.3.3 测试方法

RecordDTO 参数的等价类划分考虑为 ID、创建时间或类型为空，记录不存在，记录存在三种情况。

11.2.3.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_002_003_001	输入的recordDTO中ID\创建时间或类型为空的情况	低
UT_TD_002_003_002	输入ID在当前饮食数据不存在的情况	高
UT_TD_002_003_003	输入ID在当前饮食数据存在的情况	高

11.2.3.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

11.2.3.6 对应用例

测试项编号	UT_TD_002_003_001	
优先级	低	
测试项描述	测试recordDTO中ID、创建时间或类型为空的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	recordDTO = new RecordDTO(null, null, null)	返回 ResponseResult.errorResult(400, "id、时间、类型不能为空")

测试项编号	UT_TD_002_003_002	
优先级	高	
测试项描述	输入ID在当前饮食数据不存在的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果

001	recordDTO = new RecordDTO(500L, 2021-8-6 12:30:00, "早餐")	返回 ResponseResult.errorResult(400, "该饮食数据不存在")
-----	--	--

测试项编号	UT_TD_002_003_003	
优先级	高	
测试项描述	输入ID在当前饮食数据存在的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	recordDTO = new RecordDTO(1L, 2021-8-3 12:00:00, "午餐")	返回 ResponseResult.okResult("修改成功")

11.2.4 addRecord测试分析与设计

11.2.4.1 标识符定义

UT\_TD\_002\_004

11.2.4.2 被测特性

- 输入的recordAddDTO中创建时间或类型为空时，返回相应的提示消息；
- 输入的创建时间和类型不为空时，成功添加记录并返回记录ID。

11.2.4.3 测试方法

recordAddDTO 参数的等价类划分考虑为创建时间或类型为空，记录添加成功两种情况。

11.2.4.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_002_004_001	输入的 recordAddDTO 中创建时间或类型为空的情况	低
UT_TD_002_004_002	输入的创建时间和类型不为空的情况	高

11.2.4.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

11.2.4.6 对应用例

测试项编号	UT_TD_002_004_001	
优先级	低	
测试项描述	测试 recordAddDTO 中创建时间或类型为空的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	recordAddDTO = new RecordAddDTO(null, null)	ResponseResult.errorResult(400, "时间和类型不能为空")

测试项编号	UT_TD_002_004_002	
优先级	高	
测试项描述	测试输入的创建时间和类型不为空的情况	
前置条件	用户登录系统并进入“饮食记录”页面	
用例序号	输入	期望结果
001	recordAddDTO = new RecordAddDTO(2024-6-17 12:00:00, "午餐")	返回操作成功的信息

11.3 血糖记录模块

被测函数详见下表

标识符	名称	代码行（LOC）
UT_TD_003_001	ResponseResult getGlucose(TimeRangeDTO timeRangeDTO)	12
UT_TD_003_002	ResponseResult deleteGlucose(Long id)	15
UT_TD_003_003	ResponseResult updateGlucose(GlucoseDTO glucoseDTO)	18

UT_TD_003_004	ResponseResult addGlucose(GlucoseAddDTO glucoseAddDTO)	15
---------------	--	----

### 11.3.1 getGlucose测试分析与设计

#### 11.3.1.1 标识符定义

UT\_TD\_003\_001

#### 11.3.1.2 被测特性

- 输入时间范围为空时，返回时间范围不能为空的提示消息；
- 输入的时间范围不合法，即终止时间早于起始时间时，返回终止时间不能早于起始时间的提示信息；
- 输入时间范围合法时，返回用户在该时间范围内的所有血糖记录。

#### 11.3.1.3 测试方法

时间范围timeRangeDTO参数的等价类划分考虑为空、时间范围内无记录、时间范围内有记录三种情况。

#### 11.3.1.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_003_001_001	输入时间范围为空的情况	低
UT_TD_003_001_002	输入时间范围不合法的情况	低
UT_TD_003_001_003	输入时间范围内无血糖记录的情况	高
UT_TD_003_001_004	输入时间范围内有血糖记录的情况	高

#### 11.3.1.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

#### 11.3.1.6 对应用例

测试项编号	UT_TD_003_001_001

优先级	低	
测试项描述	测试输入时间范围为空的情况	
前置条件	用户登录系统并进入“血糖数据”页面	
用例序号	输入	期望结果
001	timeRangeDTO=(null,null)	返回时间范围为空的提示消息

测试项编号	UT_TD_003_001_002	
优先级	低	
测试项描述	输入时间范围不合法的情况	
前置条件	用户登录系统并进入“血糖数据”页面	
用例序号	输入	期望结果
001	timeRangeDTO=(2021-07-30 17:00:00,2021-07-30 16:00:00)	返回时间范围不合法的提示消息

测试项编号	UT_TD_003_001_003	
优先级	高	
测试项描述	测试输入时间范围内无血糖记录的情况	
前置条件	用户登录系统并进入“血糖数据”页面	
用例序号	输入	期望结果
001	timeRangeDTO=(2019-12-20 00:00:00,2019-12-30 00:00:00)	返回空列表

测试项编号	UT_TD_003_001_004	
优先级	高	
测试项描述	测试输入时间范围内有血糖记录的情况	
前置条件	用户登录系统并进入“血糖数据”页面	



用例序号	输入	期望结果
001	timeRangeDTO=(2021-07-30 16:00:00,2021-07-30 17:00:00)	返回列表[Glucose{id=2, userId=1, gluValue=6.300, time=2021-07-30T16:43}, Glucose{id=3, userId=1, gluValue=6.900, time=2021-07-30T16:58}]

### 11.3.2 deleteGlucose测试分析与设计

#### 11.3.2.1 标识符定义

UT\_TD\_003\_002

#### 11.3.2.2 被测特性

- 输入的id参数为空时，返回id不能为空的提示消息；
- 输入的id不存在时，返回血糖数据不存在的提示消息；
- 输入的id存在时，从数据库中删除该id对应的血糖记录。

#### 11.3.2.3 测试方法

id参数的等价类划分考虑为空、不存在记录、存在记录三种情况。

#### 11.3.2.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_003_002_001	输入的id参数为空的情况	低
UT_TD_003_002_002	输入的id参数无对应的血糖记录的情况	低
UT_TD_003_002_003	输入的id参数有对应的血糖记录的情况	高

#### 11.3.2.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

#### 11.3.2.6 对应用例

测试项编号	UT_TD_003_002_001
-------	-------------------

优先级	低	
测试项描述	测试输入的id参数为空的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“删除”	
用例序号	输入	期望结果
001	id=null	返回id不能为空的提示消息

测试项编号	UT_TD_003_002_002	
优先级	低	
测试项描述	测试输入的id无对应的血糖记录的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“删除”	
用例序号	输入	期望结果
001	id=1	返回血糖数据不存在的提示消息

测试项编号	UT_TD_003_002_003	
优先级	高	
测试项描述	测试输入的id有对应的血糖记录的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“删除”	
用例序号	输入	期望结果
001	id=659	id对应的血糖记录被从数据库中删除，返回删除成功的提示消息

### 11.3.3 updateGlucose测试分析与设计

#### 11.3.3.1 标识符定义

UT\_TD\_003\_003

#### 11.3.3.2 被测特性

- 输入的glucoseDTO参数中id为空时，返回id不能为空的提示消息；

- 输入的glucoseDTO参数中id无对应的血糖记录时，返回该血糖数据不存在的提示消息；
- 输入的glucoseDTO参数中id有对应的血糖记录，且gluValue和time中至少一个为空时，使用非空的值来对id对应的血糖记录进行更新，返回更新成功的提示消息；
- 输入的glucoseDTO参数中id有对应的血糖记录，且gluValue和time均不为空时，使用新的值来对id对应的血糖记录进行更新，返回更新成功的提示消息。

11.3.3.3 测试方法

id参数的等价类划分考虑为空、不存在记录、存在记录三种情况。

血糖值gluValue和时间time参数的等价类划分考虑至少有一个为空和均不为空两种情况。

11.3.3.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_003_003_001	输入的id参数为空的情况	低
UT_TD_003_003_002	输入的id参数无对应的血糖记录的情况	低
UT_TD_003_003_003	输入的id参数有对应的血糖记录且gluValue和time中至少一个为空的情况	高
UT_TD_003_003_004	输入的id参数有对应的血糖记录且gluValue和time中均不为空的情况	高

11.3.3.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

11.3.3.6 对应用例

测试项编号	UT_TD_003_003_001	
优先级	低	
测试项描述	测试输入的id参数为空的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“修改”	
用例序号	输入	期望结果
001	glucoseDTO=(null,null,null)	返回id不能为空的提示消息

测试项编号	UT_TD_003_003_002	
优先级	低	
测试项描述	测试输入 id 参数无对应的血糖记录的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“修改”	
用例序号	输入	期望结果
001	glucoseDTO=(1,null,null)	返回血糖数据不存在的提示消息

测试项编号	UT_TD_003_003_003	
优先级	高	
测试项描述	测试输入 id 参数有对应的血糖记录且 gluValue 和 time 中至少一个为空的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“修改”	
用例序号	输入	期望结果
001	glucoseDTO=(658,14.000,null)	id 对应的血糖记录的血糖值被修改为 14.00，返回修改成功的提示消息

测试项编号	UT_TD_003_003_004	
优先级	高	
测试项描述	测试输入 id 参数有对应的血糖记录且 gluValue 和 time 均不为空的情况	
前置条件	用户在“血糖数据”页面中选中血糖记录并点击“修改”	
用例序号	输入	期望结果
001	glucoseDTO=(657,13.800,2021-08-06 12:30:00)	id 对应的血糖记录的血糖值和时间被修改，返回修改成功的提示消息

### 11.3.4 addGlucose 测试分析与设计

11.3.4.1 标识符定义

UT\_TD\_003\_004

11.3.4.2 被测特性

- 输入的gluValue或time为空时，返回血糖值和时间不能为空的提示消息；
- 输入的gluValue或time均合法时，在数据库中创建新的血糖记录，并返回添加成功的提示消息。

11.3.4.3 测试方法

血糖值gluValue和时间time参数的等价类划分考虑至少有一个为空和均不为空两种情况。

11.3.4.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_003_004_001	输入的gluValue或time为空的情况	低
UT_TD_003_004_002	输入的gluValue和time均不为空的情况	高

11.3.4.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

11.3.4.6 对应用例

测试项编号	UT_TD_003_004_001	
优先级	低	
测试项描述	测试输入的gluValue或time为空的情况	
前置条件	用户在“血糖数据”页面中点击“添加记录”	
用例序号	输入	期望结果
001	glucoseAddDTO=(12.000,null)	返回血糖值和时间不能为空的提示消息

测试项编号	UT_TD_003_004_002
优先级	高

测试项描述	测试输入的gluValue和time均不为空的情况	
前置条件	用户在“血糖数据”页面中点击“添加记录”	
用例序号	输入	期望结果
001	glucoseAddDTO= (12.000,2024-06-10 12:30:00)	在数据库中创建新的血糖记录，并返回添加成功的提示消息

11.4 食谱模块

11.4.1 getCookbook测试分析与设计

11.4.1.1 标识符定义

UT\_TD\_004\_001

11.4.1.2 被测特性

- 输入的请求的食谱不存在时，返回查询无信息的提示消息；
- 输入的请求的食谱存在时，返回对应所需的食谱信息；

11.4.1.3 测试方法

将食谱信息按照等价类划分考虑为存在和不存在记录两种情况。

11.4.1.4 测试项标识

标识符标识符	测试项描述	优先级
UT_TD_004_001_001	请求的食谱不存在的情况	低
UT_TD_004_001_002	请求的食谱存在的情况	低

11.4.1.5 测试通过\失败标准

所有的测试用例都必须被执行，且没有发现错误。

11.4.2.6 对应用例

测试项编号	UT_TD_004_001_001
优先级	低

测试项描述	请求的食谱不存在的情况	
前置条件	用户在点击"食谱"跳转到对应页面	
用例序号	输入	期望结果
001	location="广西" season="春天" number=-1	返回查询无信息的提示消息

测试项编号	UT_TD_004_001_002	
优先级	低	
测试项描述	请求的食谱存在的情况	
前置条件	用户在点击"食谱"跳转到对应页面	
用例序号	输入	期望结果
001	location="东北" season="春季" number=1	返回对应所需的食谱信息

## 12 图形化测试结果报告

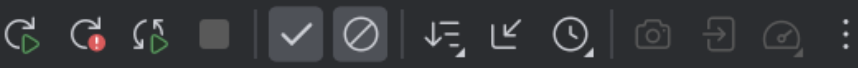
### 12.1 用户管理模块

### 12.2 饮食记录模块

### 12.3 血糖记录模块

使用JUnit对GlucoseServiceImpl类中的四个函数进行测试，共设计了13个测试用例，测试的结果如下，有一个测试用例未通过，其余测试用例均通过，



运行		GlucoseServiceImplTest	×
			
✓	✗	GlucoseServiceImplTest (com.tongji.service.service.impl)	2秒 251毫秒
✓		测试输入的gluValue或time为空的情况	867毫秒
✓		测试输入的gluValue和time均不为空的情况	408毫秒
✓		测试输入的空id参数为空的情况	43毫秒
✓		测试输入的空id参数无对应的血糖记录的情况	304毫秒
✓		测试输入的空id参数有对应的血糖记录且gluValue和time中至少一个为空的情况	89毫秒
✓		测试输入的空id参数有对应的血糖记录且gluValue和time均不为空的情况	82毫秒
✓		测试时间范围为空的情况	46毫秒
✗		测试时间范围不合法的情况	73毫秒
✓		测试输入时间范围内无血糖记录的情况	60毫秒
✓		测试输入时间范围内有血糖记录的情况	66毫秒
✓		测试输入的空id参数为空的情况	44毫秒
✓		测试输入的空id无对应的血糖记录的情况	77毫秒
✓		测试输入的空id有对应的血糖记录的情况	92毫秒

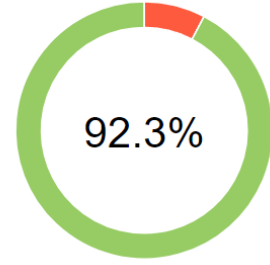
以下是使用Allure生成的单元测试报告，

ALLURE REPORT 2024/6/11

15:04:53 - 15:04:56 (2s 284ms)

13

测试用例



测试套 总共1项

com.tongji.service.service.impl.GlucoseServiceImple

1

12

显示所有

环境

没有环境变量

特性场景 总共1项

血糖记录模块单元测试

1

3

显示所有

总览

类别

测试套

图表

时间刻度

功能

包

功能

order

名称

用时

状态

通过用例状态过滤: 1 0 12 0 0

Marks:

血糖记录模块单元测试

1 12

血糖信息删除

3

✓ #1 测试输入的id参数为空的情况

42ms

✓ #2 测试输入的id无对应的血糖记录的情况

75ms

✓ #3 测试输入的id有对应的血糖记录的情况

92ms

血糖信息更新

4

✓ #1 测试输入的id参数为空的情况

41ms

✓ #2 测试输入的id参数无对应的血糖记录的情况

302ms

✓ #3 测试输入的id参数有对应的血糖记录且gluValue和time中至少一个为空的情况

88ms

✓ #4 测试输入的id参数有对应的血糖记录且gluValue和time均不为空的情况

82ms

血糖信息查找

1 3

✗ #2 测试时间范围不合法的情况

70ms

✓ #1 测试时间范围为空的情况

44ms

✓ #3 测试输入时间范围内无血糖记录的情况

59ms

✓ #4 测试输入时间范围内有血糖记录的情况

65ms

血糖信息添加

2


✓ #2 测试输入的gluValue和time均不为空的情况

406ms

✓ #1 测试输入的gluValue或time为空的情况

857ms

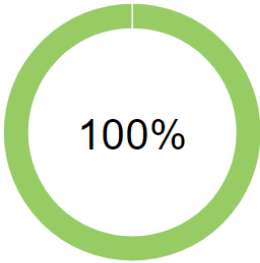
bug出现在getGlucose函数中，经检查发现bug原因是未对终止时间早于起始时间的异常情况做处理，解决后重新进行测试，结果如下，可见所有的13个测试用例均通过，

运行		GlucoseServiceImplTest	×
			
✓	GlucoseServiceImplTest (com.tongji.service.service.impl)	1秒 840毫秒	
✓	测试输入的gluValue或time为空的情况	731毫秒	
✓	测试输入的gluValue和time均不为空的情况	394毫秒	
✓	测试输入的空id参数为空的情况	45毫秒	
✓	测试输入的空id参数无对应的血糖记录的情况	81毫秒	
✓	测试输入的空id参数有对应的血糖记录且gluValue和time中至少一个为空的情况	87毫秒	
✓	测试输入的空id参数有对应的血糖记录且gluValue和time均不为空的情况	94毫秒	
✓	测试时间范围为空的情况	42毫秒	
✓	测试时间范围不合法的情况	43毫秒	
✓	测试输入时间范围内无血糖记录的情况	71毫秒	
✓	测试输入时间范围内有血糖记录的情况	65毫秒	
✓	测试输入的空id参数为空的情况	44毫秒	
✓	测试输入的空id无对应的血糖记录的情况	59毫秒	
✓	测试输入的空id有对应的血糖记录的情况	84毫秒	

以下是解决bug后生成的测试报告。

ALLURE REPORT 2024/6/11  
15:13:55 - 15:13:57 (1s 833ms)

13  
测试用例



测试套 总共1项

com.tongji.service.service.impl.GlucoseServiceImplTest	13
--	----

显示所有

环境

没有环境变量

特性场景 总共1项

血糖记录模块单元测试	4
------------	---

显示所有

总览

类别

测试套

图表

时间刻度

功能

包

功能

order

名称

用时

状态

通过用例状态过滤: 0 0 13 0 0

Marks:

▼ 血糖记录模块单元测试

13

▼ 血糖信息删除

3

✓ #1 测试输入的id参数为空的情况

43ms

✓ #2 测试输入的id无对应的血糖记录的情况

58ms

✓ #3 测试输入的id有对应的血糖记录的情况

101ms

▼ 血糖信息更新

4

✓ #1 测试输入的id参数为空的情况

41ms

✓ #2 测试输入的id参数无对应的血糖记录的情况

78ms

✓ #3 测试输入的id参数有对应的血糖记录且gluValue和time中至少一个为空的情况

84ms

✓ #4 测试输入的id参数有对应的血糖记录且gluValue和time均不为空的情况

93ms

▼ 血糖信息查找

4

✓ #2 测试时间范围不合法的情况

41ms

✓ #1 测试时间范围为空的情况

38ms

✓ #3 测试输入时间范围内无血糖记录的情况

81ms

✓ #4 测试输入时间范围内有血糖记录的情况

61ms

▼ 血糖信息添加

2

✓ #2 测试输入的gluValue和time均不为空的情况

379ms

✓ #1 测试输入的gluValue或time为空的情况

680ms

## 12.4 食谱模块

# 13 缺陷跟踪

软件测试中的缺陷跟踪是一项关键的质量控制过程，涉及到识别、记录、管理以及通报软件测试过程中发现的缺陷的活动。缺陷跟踪有助于确保所有发现的问题能够得到适当的处理、修复并进行验证，从而提升软件产品的质量。特别是在复杂的软件项目中，缺陷跟踪是一种提高透明度、促进团队合作以及优化资源管理的重要手段。

在我们的项目中，我们使用 `PingCode` 作为我们的缺陷跟踪工具。

在开发过程中对如下举例的信息进行了缺陷跟踪，下图列举了一些我们对缺陷项的跟踪，选择 `PingCode` 作为工具提升了我们的跟踪效率、促进团队协作、并且它提供了强大的分析和报告功能。

血糖管理项目

概览

规划

需求

缺陷

工作项

迭代

发布

基线

测试

页面

资源

报表

搜索 (Ctrl+G)

+ 新建缺陷

全部缺陷

处理中的缺陷

我负责的缺陷

未解决的缺陷

更多

Q 搜索 (Ctrl+G)

筛选

排序

更多

11 个缺陷

	编号	标题	状态	优先级	负责人	严重程度	复现概率
1	MAIN-89	路由拦截那里，放开获取验证码的接口	已修复	最高	王震岳	致命	必现
2	MAIN-88	后端算法的端口不应该在服务器开放	已修复	最高	刘冲	致命	必现
3	MAIN-87	查询某段时间血糖的时候，时间参数没有规范化	处理中	普通	王震岳	严重	必现
4	MAIN-86	分页查询食物的接口，参数缺少一个类型	处理中	较高	刘冲	严重	必现
5	MAIN-85	插入Record和RecordDeatail的时候，两条SQL可能有事务回滚的需求，添加上，否则造成数据不一致	已修复	最高	刘金硕	致命	小概率复现
6	MAIN-84	去掉下方多余的编号	已拒绝	最低	刘冲	建议	必现
7	MAIN-83	用户注册登录后，没有删除Redis里面缓存的验证码信息	已修复	较高	王震岳	一般	大概率复现
8	MAIN-82	血糖查询的SQL语句做下优化，索引做好	已修复	较高	刘冲	一般	大概率复现
9	MAIN-81	提交图片识别时，选择的食物数量恒定为3，没有根据选择改变	已修复	最高	刘金硕	严重	大概率复现
10	MAIN-80	血糖曲线添加优化	处理中	普通	刘冲	一般	必现
11	MAIN-79	UI主界面密码没有做隐私处理	已修复	普通	刘金硕	严重	必现