deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

SpringCanteen

healthy

fast-food

Teste e Qualidade de Software

João Dourado (108636)

Miguel Figueiredo (108287)

José Mendes (107188)

Vítor Santos (107186)

Team

João Dourado

Team Manager

Responsible for distributing work among team members, meeting deadlines and guiding the team in the right direction.

Miguel Belchior

Product Owner

Represents stakeholders interests and has deep knowledge of the requirements of the product.

Vítor Santos

DevOps master

Responsible for the infrastructure and its configuration

José Mendes

QA Enginner

Ensures software meets quality standards by designing tests to identify and fix bugs

Product Concept

Restaurant Kiosk for ordering healthy food

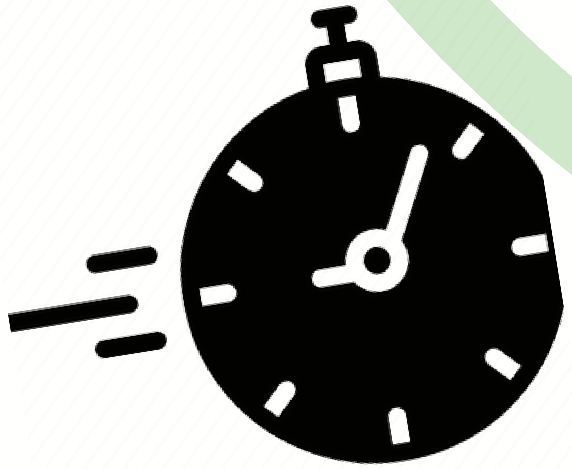
Expedite ordering food, using **Customer Kiosks** throughout the restaurant

Employee portal for easily managing orders

Simplify the process of delivering food, by having employees handle their own tasks - taking payments, cooking and delivering food!

Easy order tracking through Digital Signage

Our digital signage solution helps you keep track of the state of your order. From **being cooked** to **ready for pick up**!



Personas - Customer

Overview

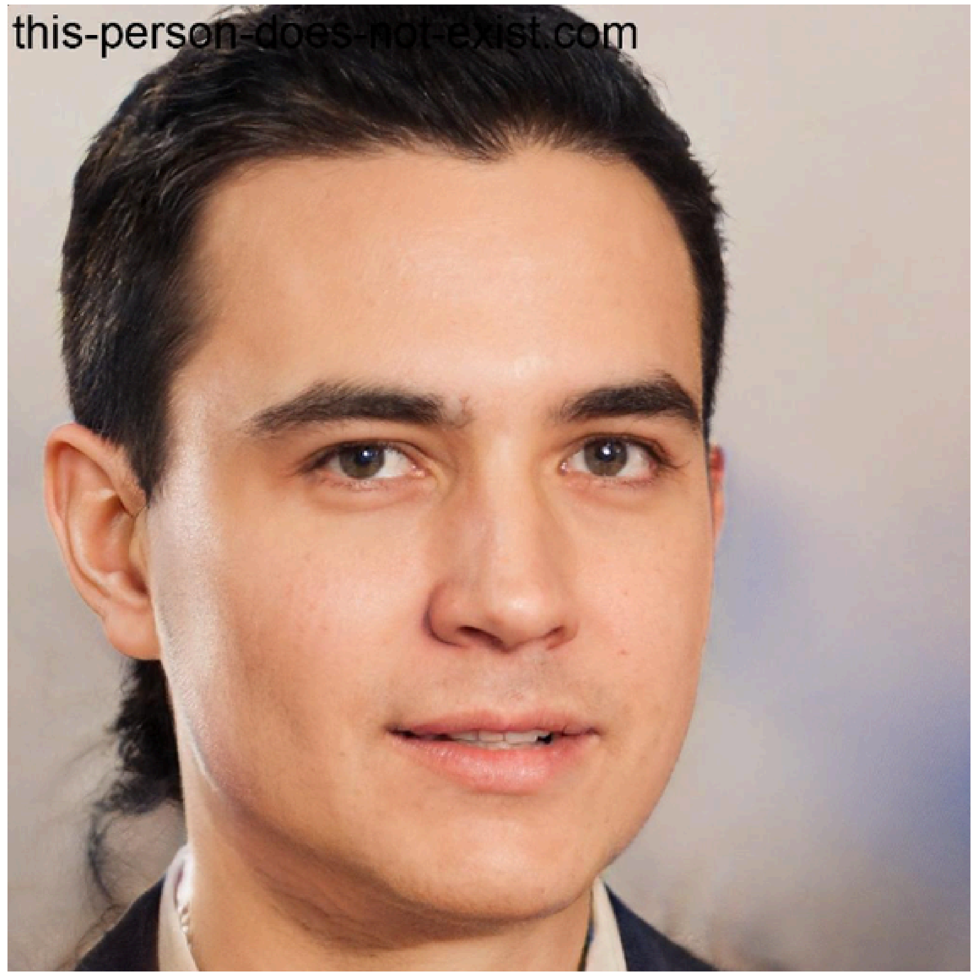
- Name: **Jorge**
- Age: 25 years
- Occupation: ER Nurse
- Experience: 4 years
- Work Schedule: Long night shifts

Experience

- Tech literacy: intermediate, recognizes and appreciates efficiency
- Fast-food/take-away restaurants with and without kiosk system integration

Challenges

- Limited time
- Unhealthy dietary options at most practical fast-food restaurant
- Inefficient and/or limited digital transformation at healthier alternatives (no kiosk, digital signage)



Personas - Restaurant Staff

Overview

- Name: **Carla**
- Age: 31 years
- Occupation: Fast-Food Restaurant Employee
- Experience: 5 years
- Work Load: Hundreds of orders over variable length several-hour-long shifts

Experience

- Tech Literacy: Some tech knowledge
- Order Handling: Physical counter orders, high customer volume

Challenges

- Order Management: Inefficient physical order handling at the counter
- Customer Turnaway: Long waiting lines at peak hours
- Work Pressure: High stress due to fast-paced environment
- Career Aspiration: Aiming for a promotion through improved order handling efficiency

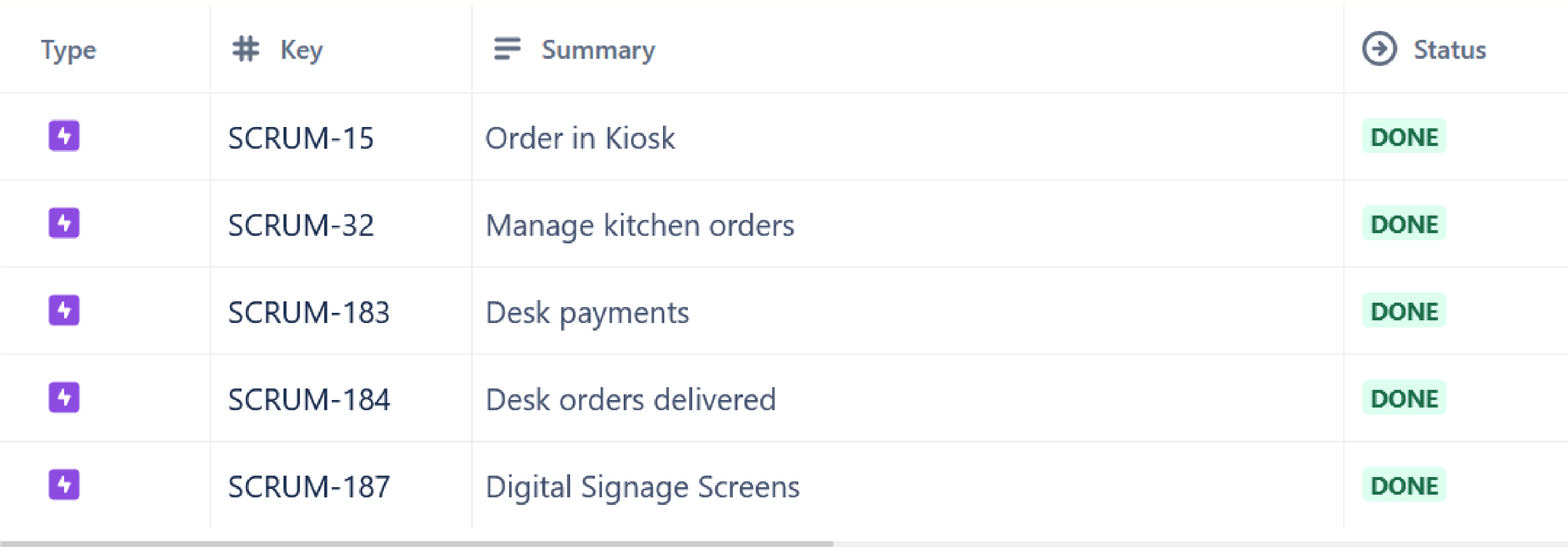
Scenarios

- **Scenario 1: Place an order**
  - Jorge places an order using the kiosk
- **Scenario 2: Customize an order**
  - Jorge places an order and customizes it
- **Scenario 3: Pay directly on the Kiosk**
  - Jorge pays directly using the kiosk
- **Scenario 4: Update the order status**
  - Carla updates the order from “Preparing” to “Ready”
- **Scenario 5: Accept payments**
  - Carla accepts payments from the desk

Epics

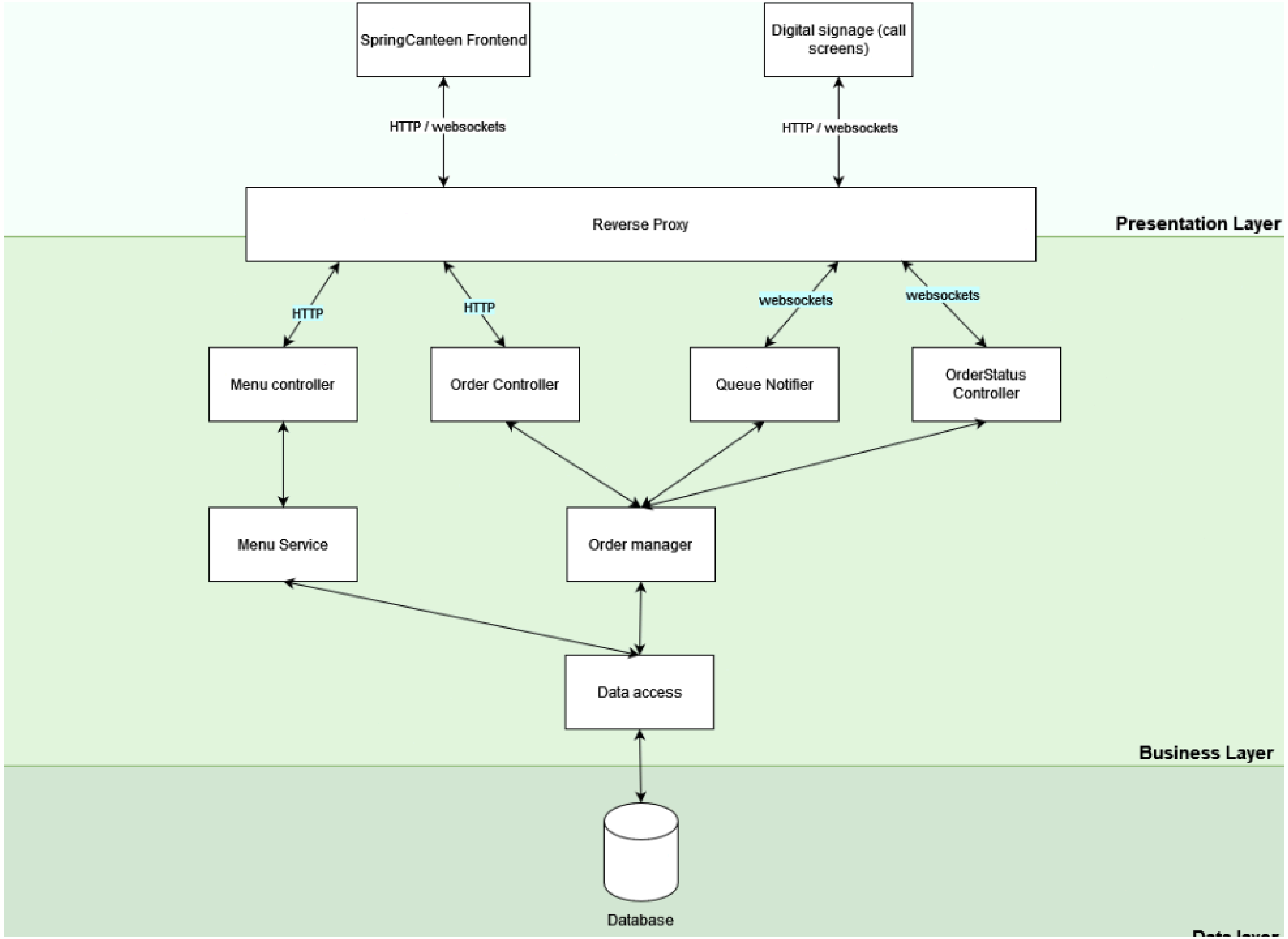
PRIORITY

- **Order in Kiosk** ✓
  - Customer placing an order, including all customization and priority setting
- **Manage Kitchen Orders** ✓
  - Kitchen cooks receiving orders and able to set them to “preparing” and “ready”
- **Desk Payments** ✓
  - Payments employee see orders that need paying and can receive payments
- **Desk orders delivered** ✓
  - Employee handling orders in desk can see existing orders and confirm them as delivered
- **Digital Signage Screens** ✓

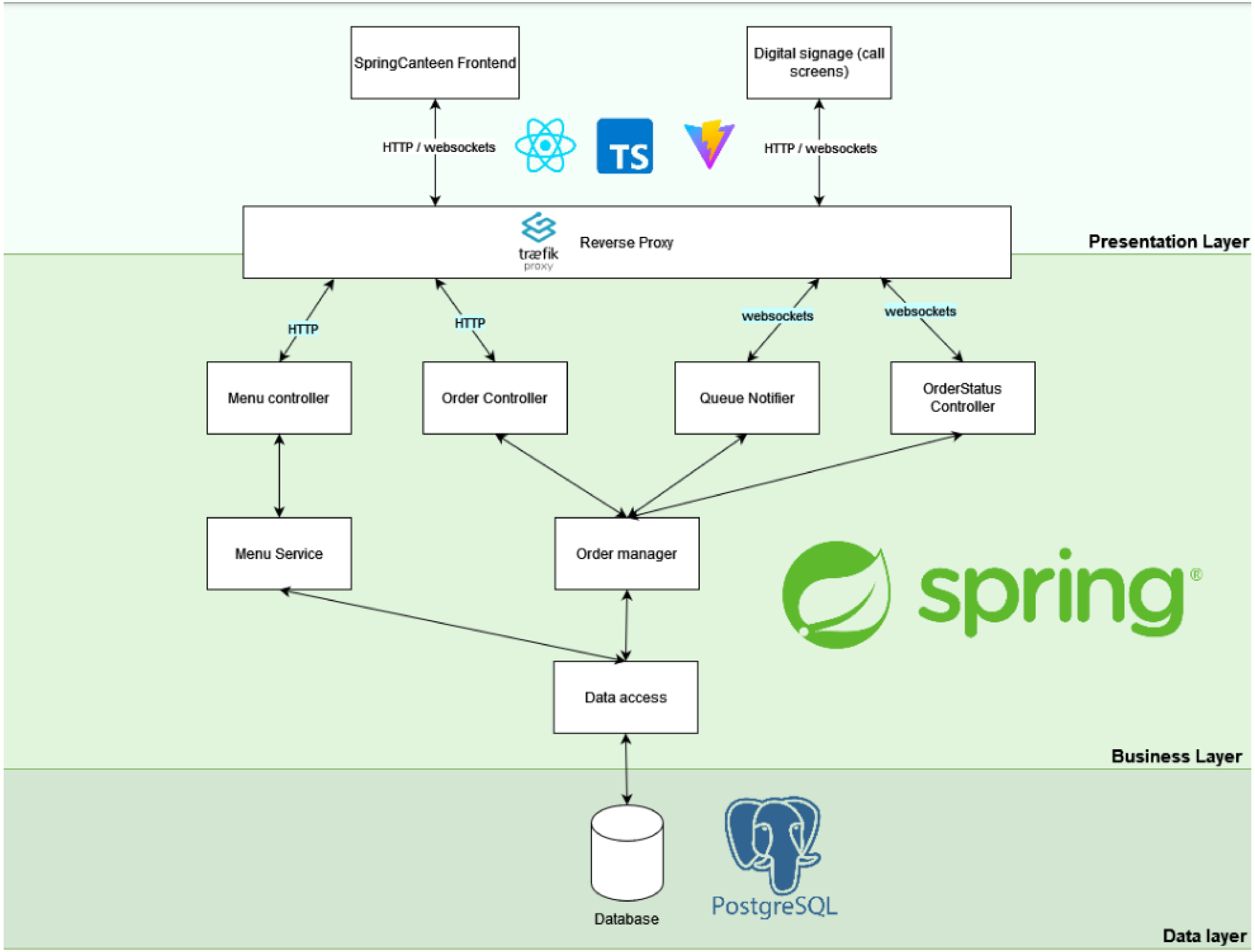


Type	# Key	Summary	Status
⚡	SCRUM-15	Order in Kiosk	DONE
⚡	SCRUM-32	Manage kitchen orders	DONE
⚡	SCRUM-183	Desk payments	DONE
⚡	SCRUM-184	Desk orders delivered	DONE
⚡	SCRUM-187	Digital Signage Screens	DONE

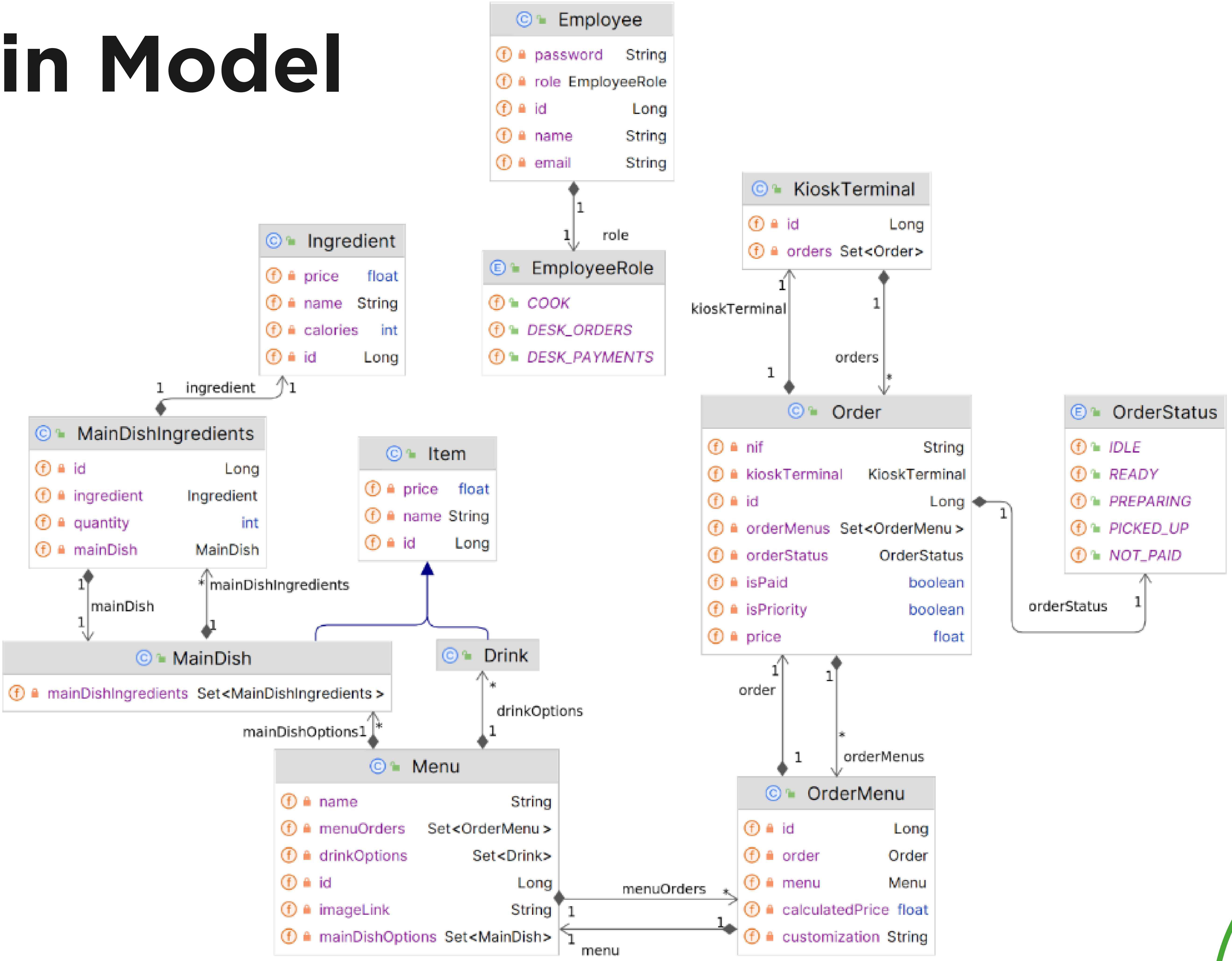
Architecture - Overview



Architecture - Deployment



Domain Model



Employee
- password String
- role EmployeeRole
- id Long
- name String
- email String

EmployeeRole
- COOK
- DESK_ORDERS
- DESK_PAYMENTS

KioskTerminal
- id Long
- orders Set<Order>

Ingredient
- price float
- name String
- calories int
- id Long

MainDishIngredients
- id Long
- ingredient Ingredient
- quantity int
- mainDish MainDish

Item
- price float
- name String
- id Long

Order
- nif String
- kioskTerminal KioskTerminal
- id Long
- orderMenus Set<OrderMenu >
- orderStatus OrderStatus
- isPaid boolean
- isPriority boolean
- price float

OrderStatus
- IDLE
- READY
- PREPARING
- PICKED_UP
- NOT_PAID

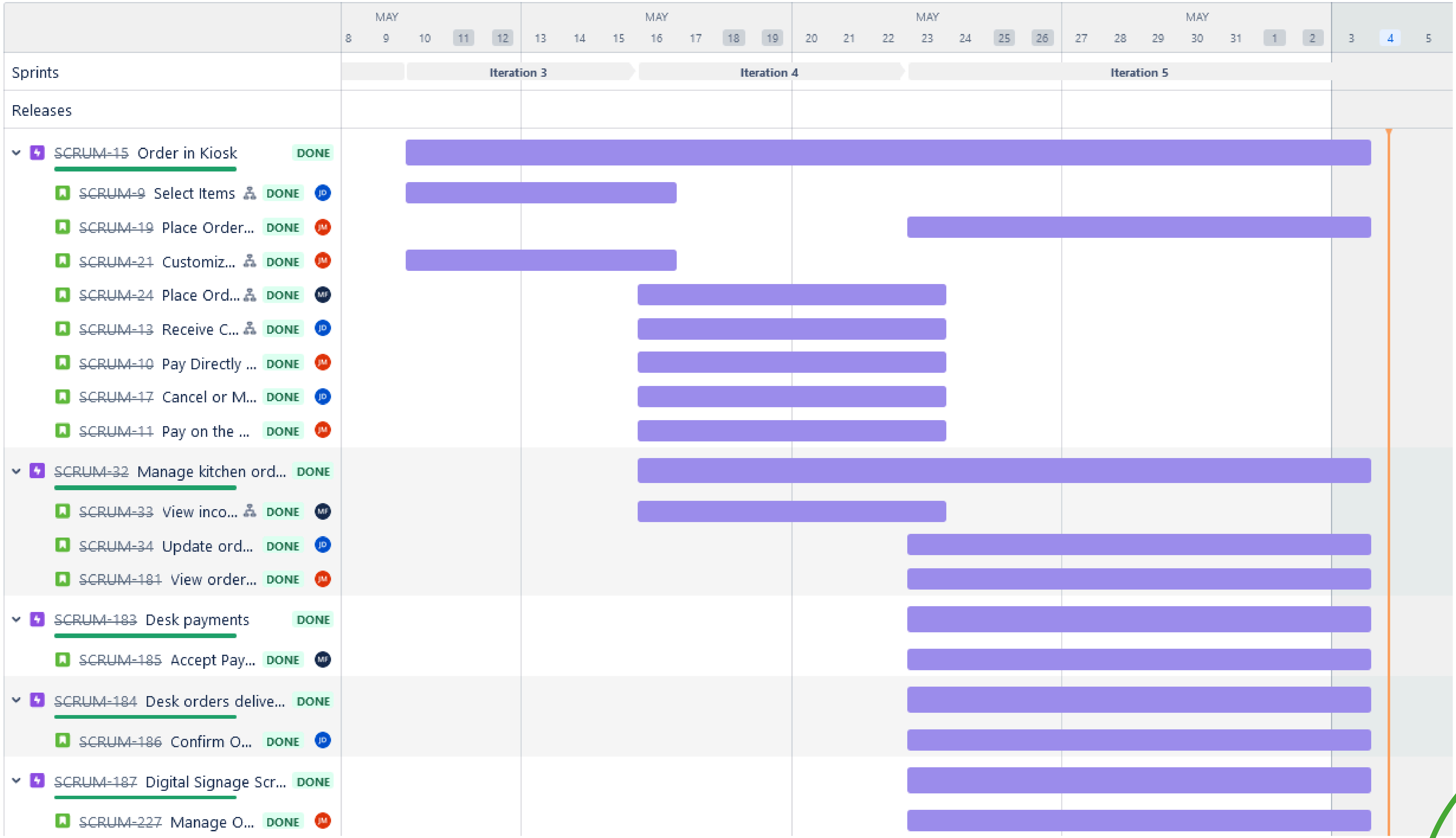
MainDish
- mainDishIngredients Set<MainDishIngredients >

Drink

Menu
- name String
- menuOrders Set<OrderMenu >
- drinkOptions Set<Drink>
- id Long
- imageLink String
- mainDishOptions Set<MainDish>

OrderMenu
- id Long
- order Order
- menu Menu
- calculatedPrice float
- customization String

Agile Backlog (1)



MAY 8 9 10 11 12 | MAY 13 14 15 16 17 18 19 | MAY 20 21 22 23 24 25 26 | MAY 27 28 29 30 31 1 2 | 3 4 5

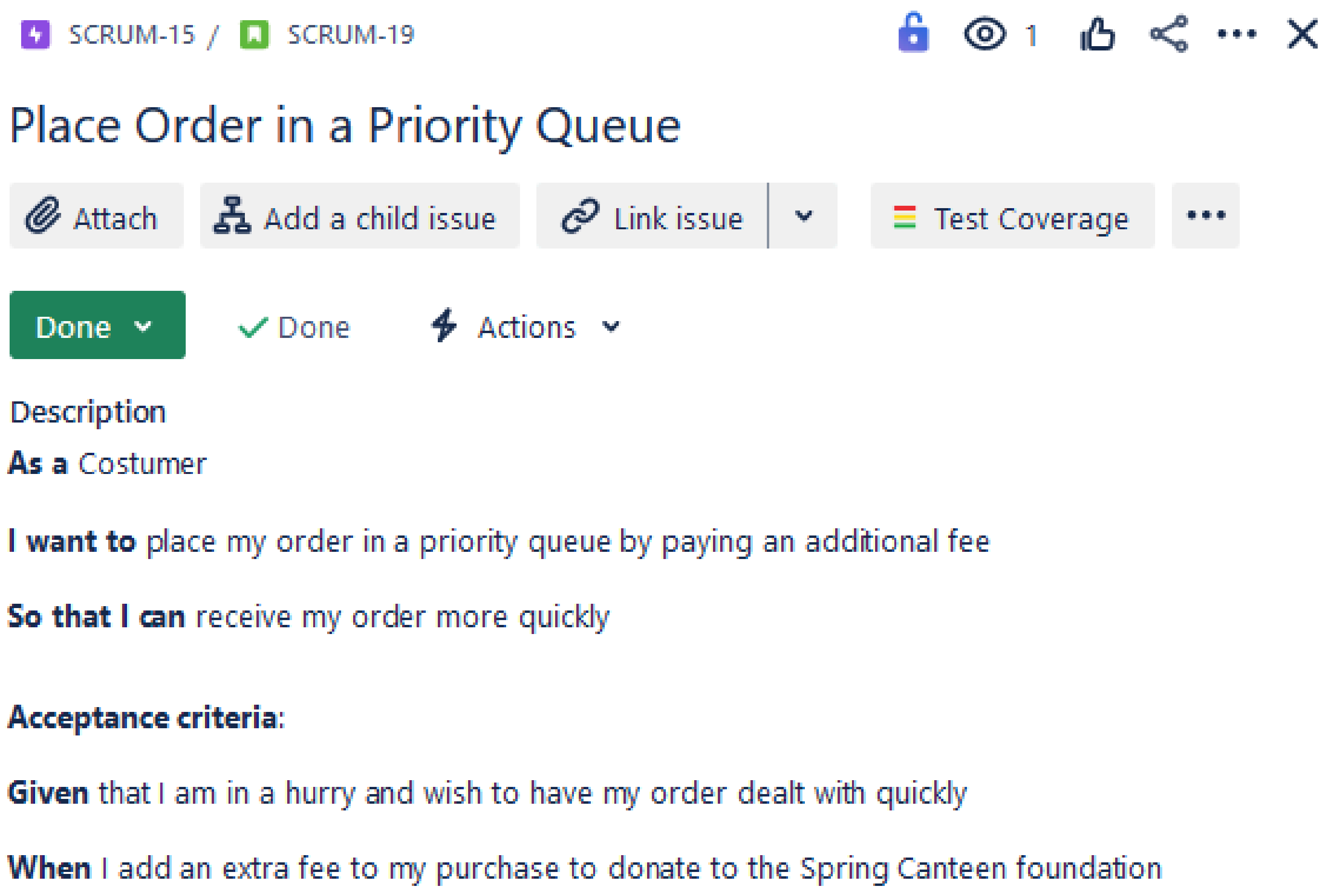
Sprints: Iteration 3, Iteration 4, Iteration 5

Releases

~~SCRUM-15~~ Order in Kiosk	DONE	
~~SCRUM-9~~ Select Items	DONE	JD
~~SCRUM-19~~ Place Order...	DONE	JM
~~SCRUM-21~~ Customiz...	DONE	JM
~~SCRUM-24~~ Place Ord...	DONE	MF
~~SCRUM-13~~ Receive C...	DONE	JD
~~SCRUM-10~~ Pay Directly ...	DONE	JM
~~SCRUM-17~~ Cancel or M...	DONE	JD
~~SCRUM-11~~ Pay on the ...	DONE	JM
~~SCRUM-32~~ Manage kitchen ord...	DONE	
~~SCRUM-33~~ View inco...	DONE	MF
~~SCRUM-34~~ Update ord...	DONE	JD
~~SCRUM-181~~ View order...	DONE	JM
~~SCRUM-183~~ Desk payments	DONE	
~~SCRUM-185~~ Accept Pay...	DONE	MF
~~SCRUM-184~~ Desk orders delive...	DONE	
~~SCRUM-186~~ Confirm O...	DONE	JD
~~SCRUM-187~~ Digital Signage Scr...	DONE	
~~SCRUM-227~~ Manage O...	DONE	JM

Agile Backlog (2)

- Work divided by **user stories** →
- **Story points**: 1 (very short) to 5 (very long)
- Iterative development: **user stories selected based on priority**



Place Order in a Priority Queue

Description

As a Costumer

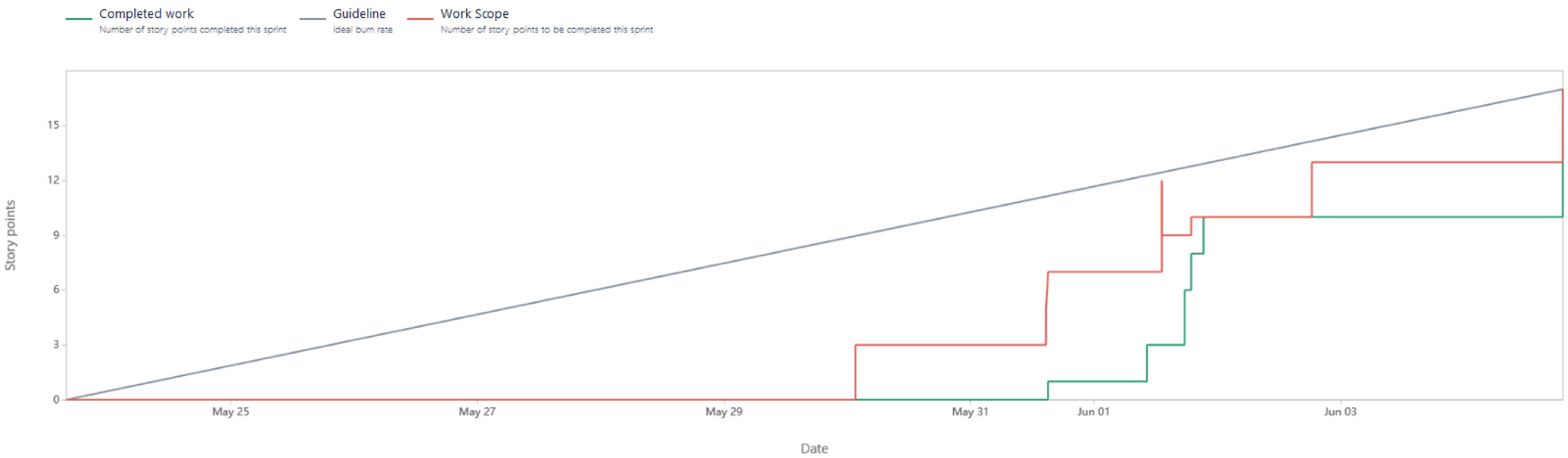
I want to place my order in a priority queue by paying an additional fee

So that I can receive my order more quickly

Acceptance criteria:

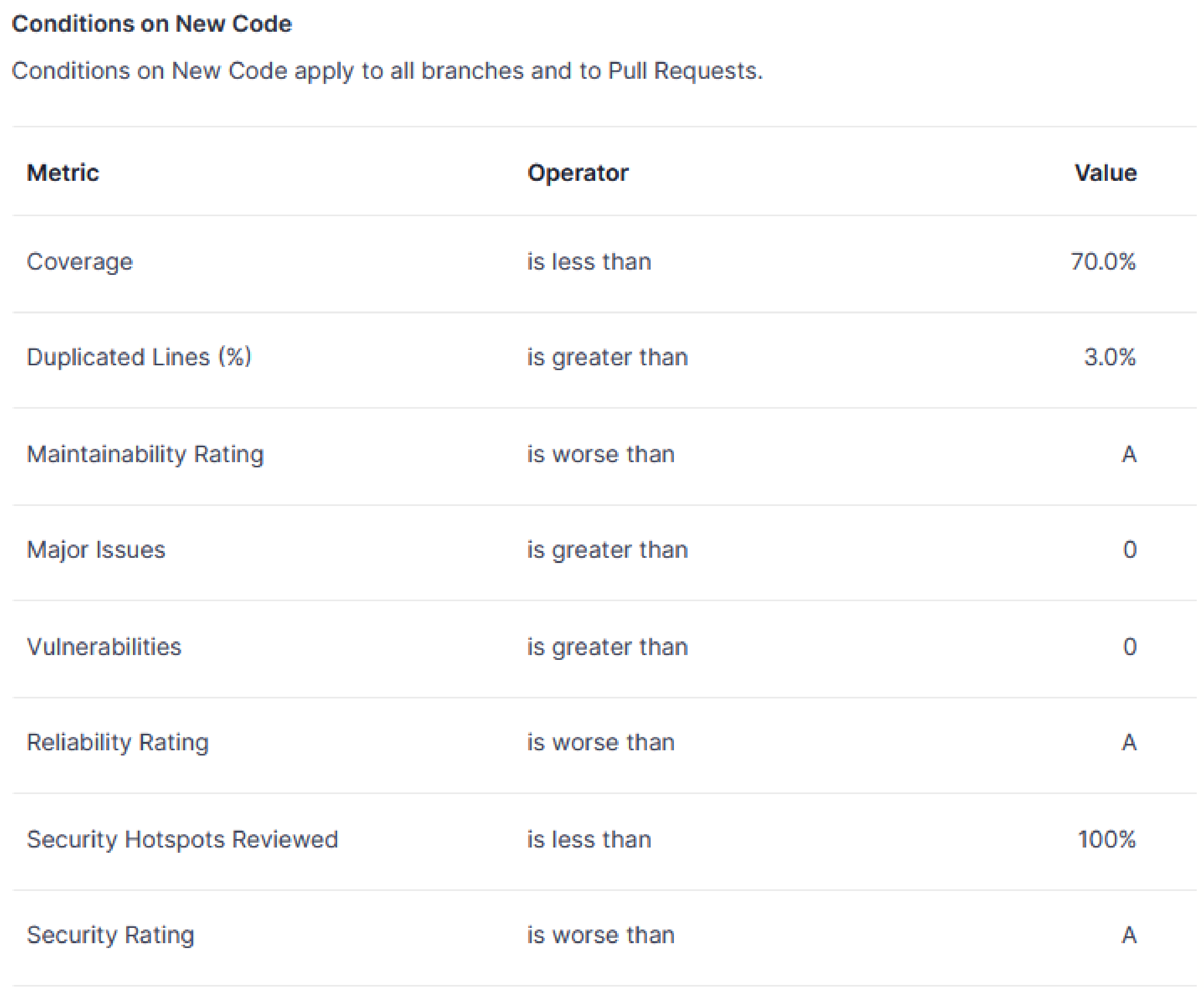
Given that I am in a hurry and wish to have my order dealt with quickly

When I add an extra fee to my purchase to donate to the Spring Canteen foundation



Code Quality Assurance

Strict SonarQube quality gate + SonarLint



Conditions on New Code

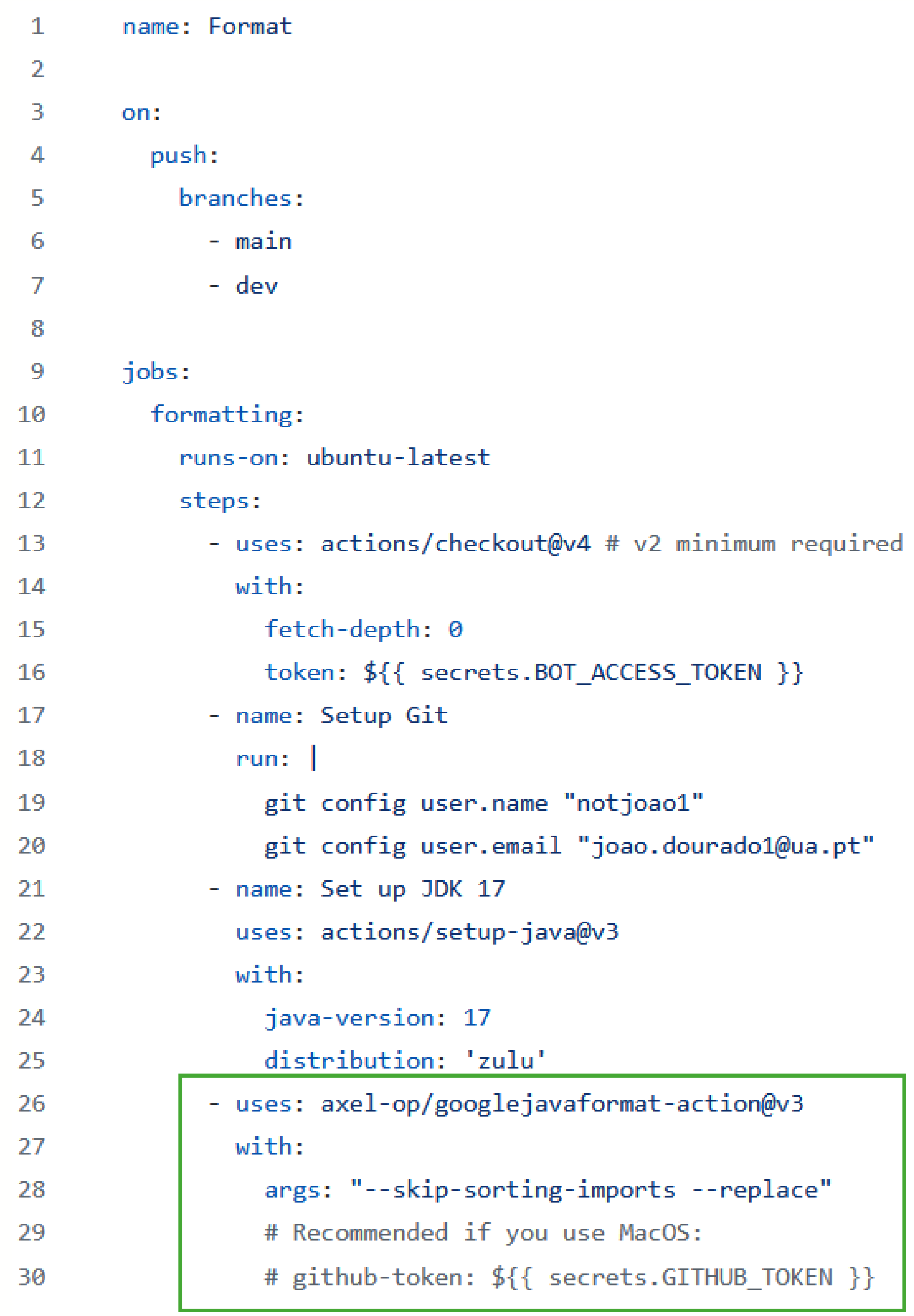
Conditions on New Code apply to all branches and to Pull Requests.

Metric	Operator	Value
Coverage	is less than	70.0%
Duplicated Lines (%)	is greater than	3.0%
Maintainability Rating	is worse than	A
Major Issues	is greater than	0
Vulnerabilities	is greater than	0
Reliability Rating	is worse than	A
Security Hotspots Reviewed	is less than	100%
Security Rating	is worse than	A

sonarlint

Google Java Code Style

(enforced through CI workflow)



```yaml
name: Format

on:
  push:
    branches:
      - main
      - dev

jobs:
  formatting:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4 # v2 minimum required
        with:
          fetch-depth: 0
          token: ${{ secrets.BOT_ACCESS_TOKEN }}
      - name: Setup Git
        run: |
          git config user.name "notjoao1"
          git config user.email "joao.dourado1@ua.pt"
      - name: Set up JDK 17
        uses: actions/setup-java@v3
        with:
          java-version: 17
          distribution: 'zulu'
      - uses: axel-op/googlejavaformat-action@v3
        with:
          args: "--skip-sorting-imports --replace"
          # Recommended if you use MacOS:
          # github-token: ${{ secrets.GITHUB_TOKEN }}
```

Development Workflow

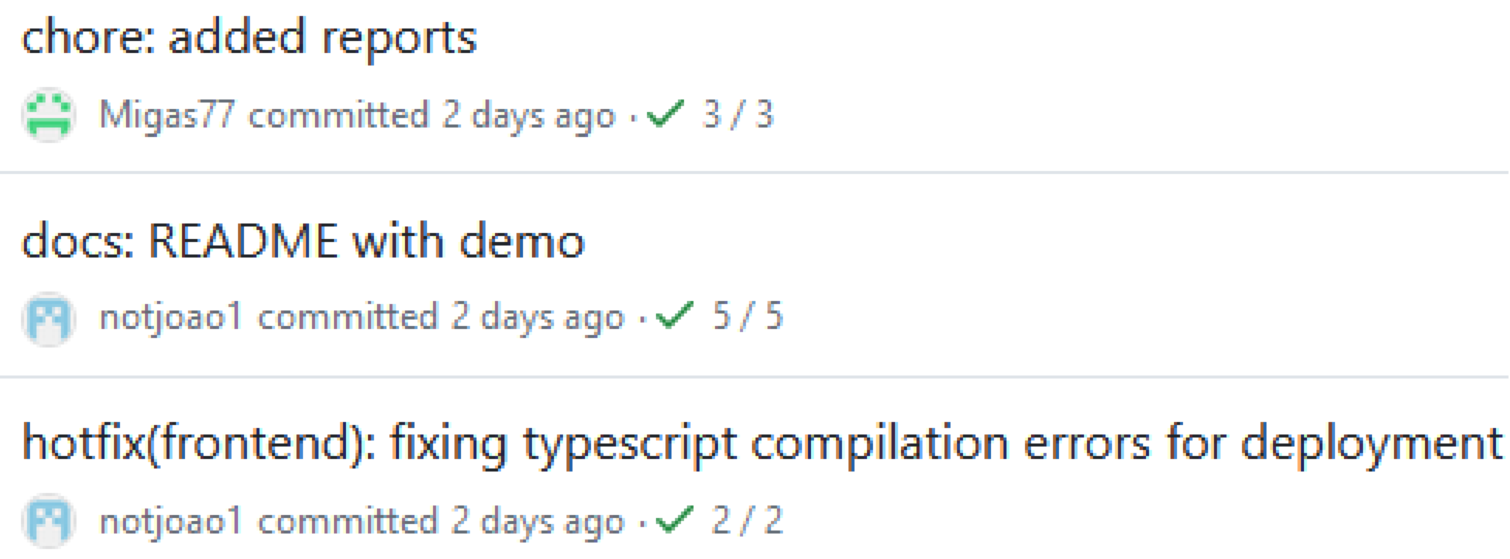
Definition of Done

- The code is written and with appropriate tests for the code developed, such as functional tests, controller tests, service tests, repository tests and integration tests;
- All tests are passing;
- The code's quality is checked by a static analysis tool and passes the quality gate;
- The user story implementation meets the acceptance criteria;
- The submitted code has been reviewed by at least 1 person.

Development Workflow

Conventions

- Conventional Commits 1.0.0



chore: added reports

Migas77 committed 2 days ago · ✓ 3 / 3

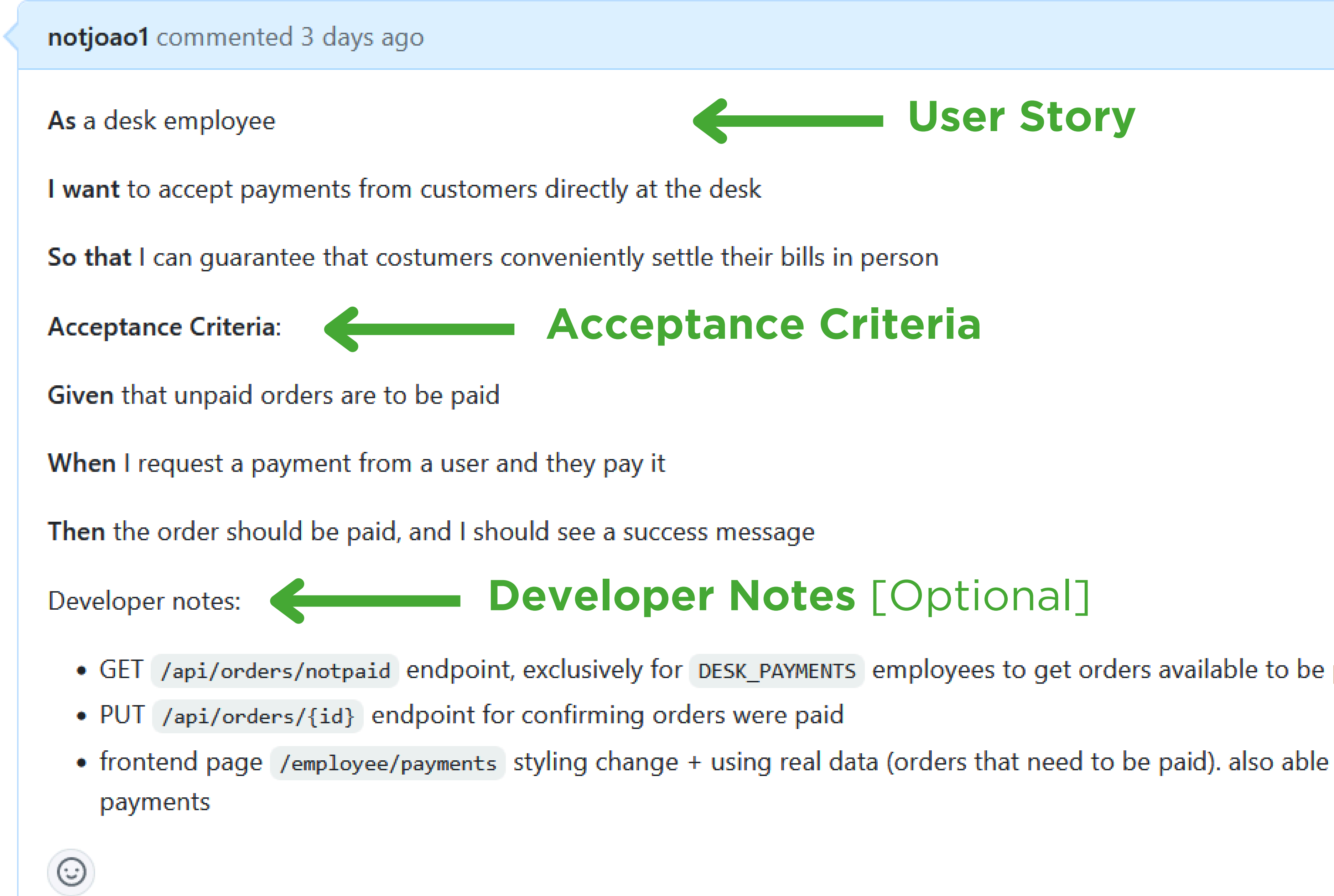
docs: README with demo

notjoao1 committed 2 days ago · ✓ 5 / 5

hotfix(frontend): fixing typescript compilation errors for deployment

notjoao1 committed 2 days ago · ✓ 2 / 2

- Conventional Pull Request Description



notjoao1 commented 3 days ago

As a desk employee

I want to accept payments from customers directly at the desk

So that I can guarantee that costumers conveniently settle their bills in person

Acceptance Criteria:

Given that unpaid orders are to be paid

When I request a payment from a user and they pay it

Then the order should be paid, and I should see a success message

Developer notes:

- GET `/api/orders/notpaid` endpoint, exclusively for `DESK_PAYMENTS` employees to get orders available to be
- PUT `/api/orders/{id}` endpoint for confirming orders were paid
- frontend page `/employee/payments` styling change + using real data (orders that need to be paid). also able payments

User Story

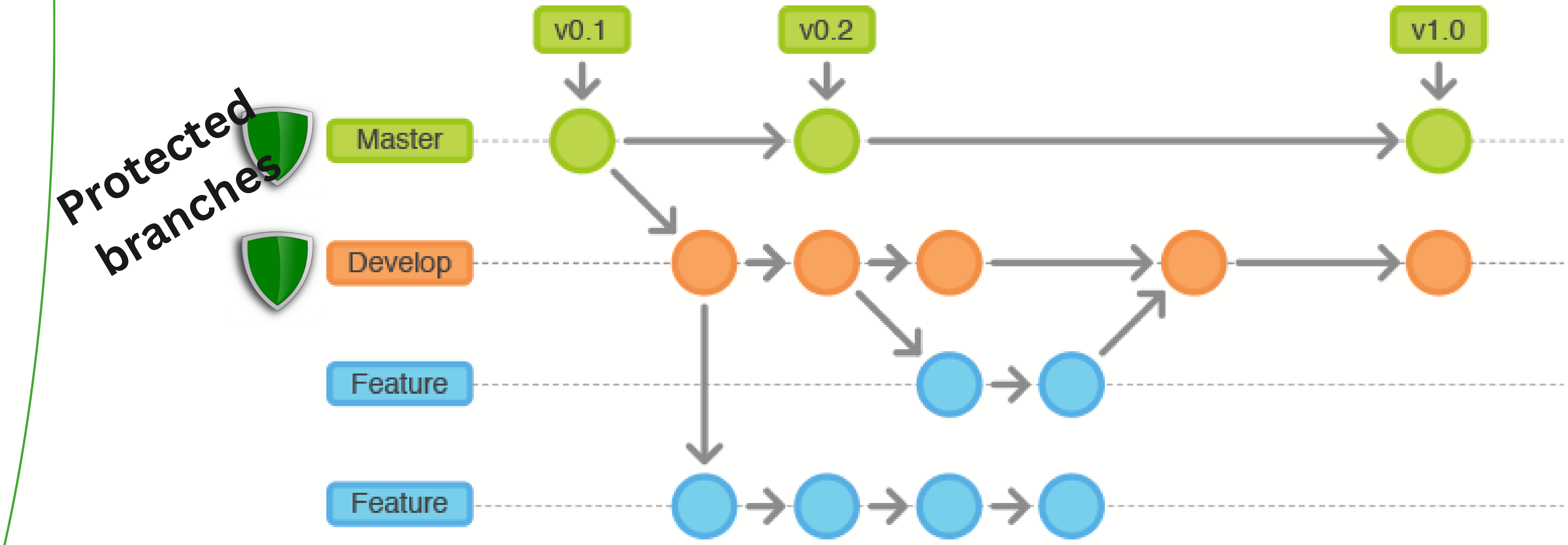
Acceptance Criteria

Developer Notes [Optional]

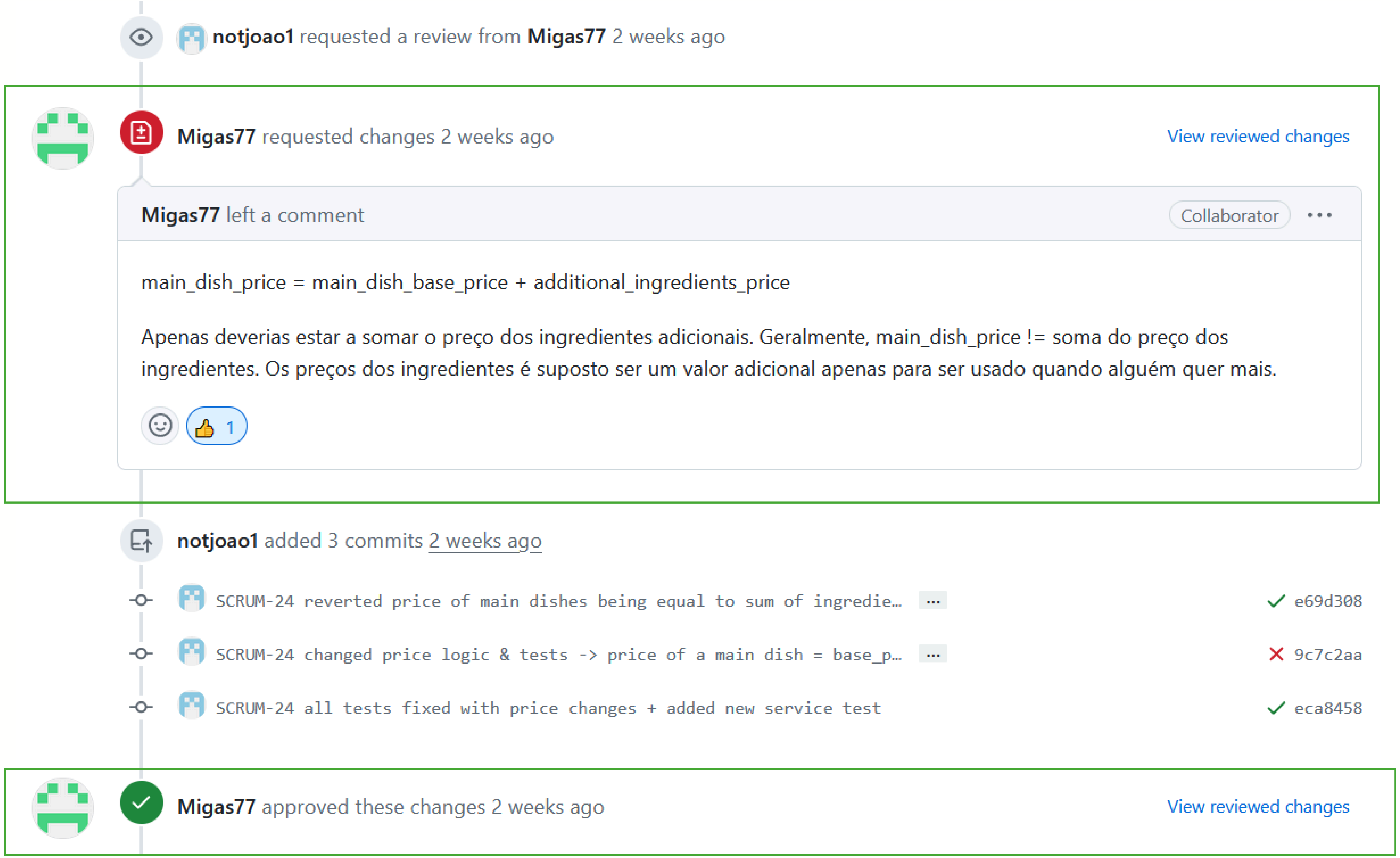
Development Workflow

Gitflow Workflow

- **main branch** - released, stable version of code
- **dev branch** - development branch - branch out from **dev** for adding features
- **feature branches** - short-lived branches for individual features



Code Review



notjoao1 requested a review from **Migas77** 2 weeks ago

Migas77 requested changes 2 weeks ago

View reviewed changes

Migas77 left a comment

Collaborator

main_dish_price = main_dish_base_price + additional_ingredients_price

Apenas deverias estar a somar o preço dos ingredientes adicionais. Geralmente, main_dish_price != soma do preço dos ingredientes. Os preços dos ingredientes é suposto ser um valor adicional apenas para ser usado quando alguém quer mais.

👍 1

notjoao1 added 3 commits 2 weeks ago

```
SCRUM-24 reverted price of main dishes being equal to sum of ingredie…   ✓ e69d308
SCRUM-24 changed price logic & tests -> price of a main dish = base_p…   ✗ 9c7c2aa
SCRUM-24 all tests fixed with price changes + added new service test     ✓ eca8458
```

Migas77 approved these changes 2 weeks ago

View reviewed changes

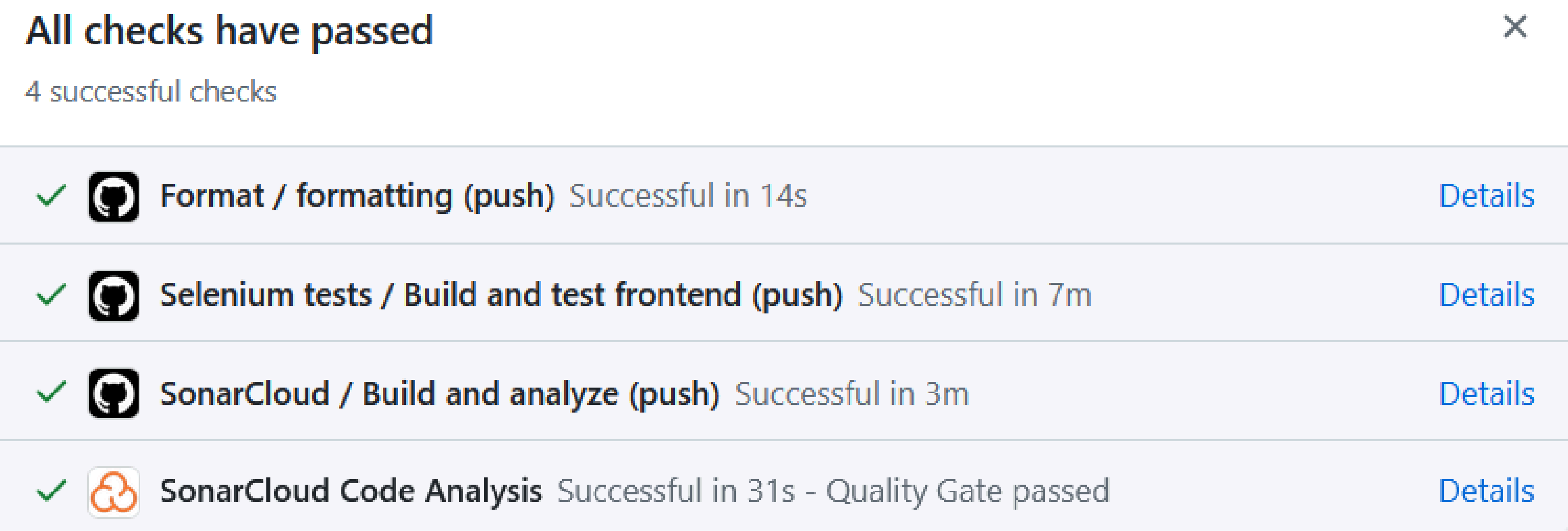
CI Pipeline

Automated workflows

Format to Google Java Code Style

Functional Tests

Static code analysis + post results to Xray



All checks have passed ×

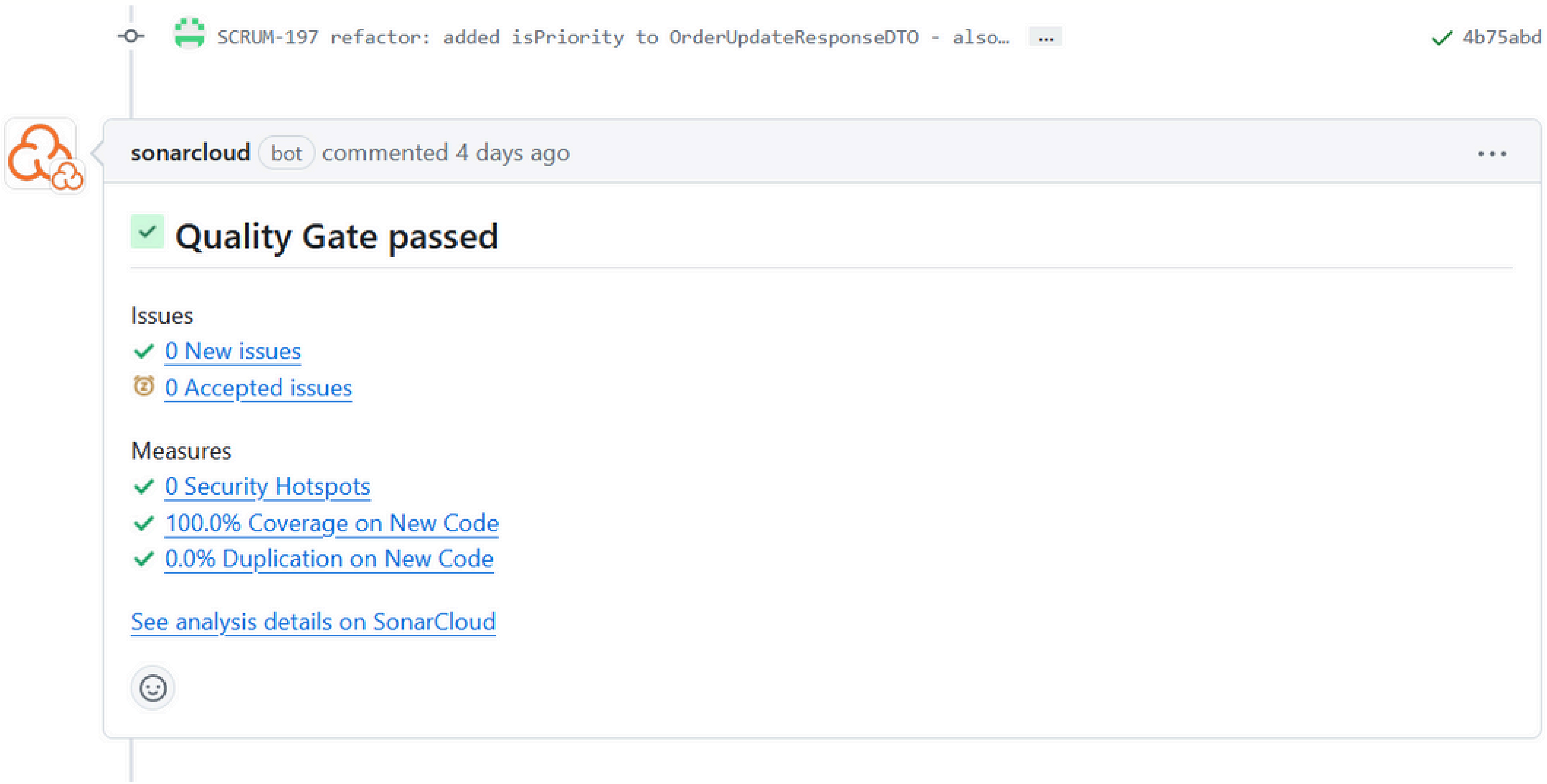
4 successful checks

✓ **Format / formatting (push)** Successful in 14s Details

✓ **Selenium tests / Build and test frontend (push)** Successful in 7m Details

✓ **SonarCloud / Build and analyze (push)** Successful in 3m Details

✓ **SonarCloud Code Analysis** Successful in 31s - Quality Gate passed Details



SCRUM-197 refactor: added isPriority to OrderUpdateResponseDTO - also… … ✓ 4b75abd

sonarcloud bot commented 4 days ago …

✓ Quality Gate passed

Issues

✓ 0 New issues

0 Accepted issues

Measures

✓ 0 Security Hotspots

✓ 100.0% Coverage on New Code

✓ 0.0% Duplication on New Code

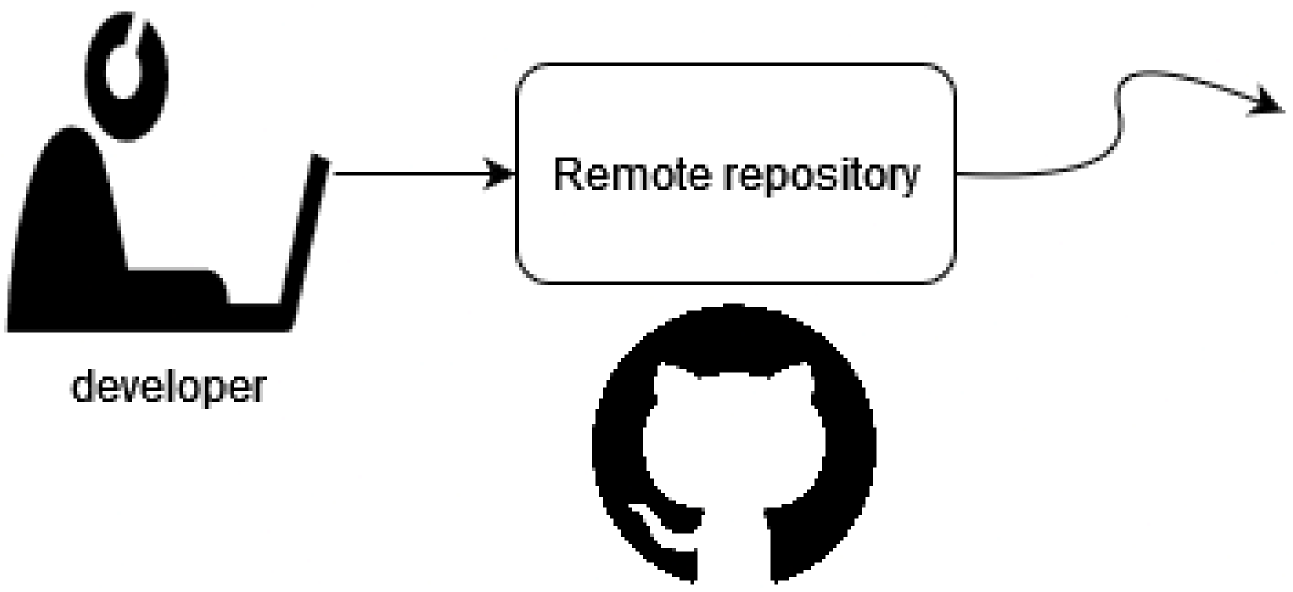
See analysis details on SonarCloud

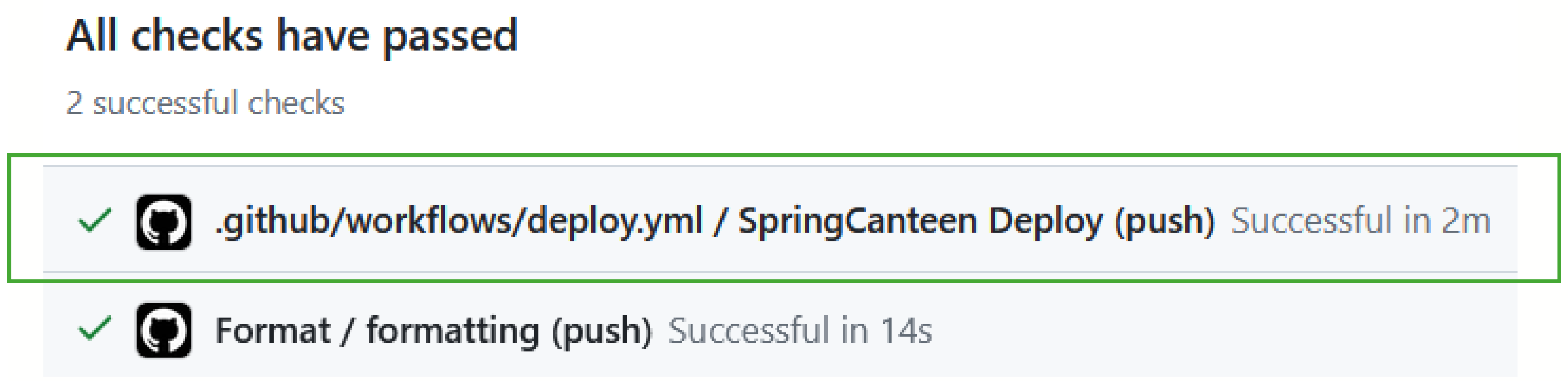
SonarCloud Integration

Continuous Deployment

- **Automatic deployment** through **workflow on push** to **main** branch
  - Commit to main
  - GitHub sends request to self-hosted runner on DETI VM
  - **docker compose -f docker-compose.prod.yml up --build -d**

production configuration





All checks have passed

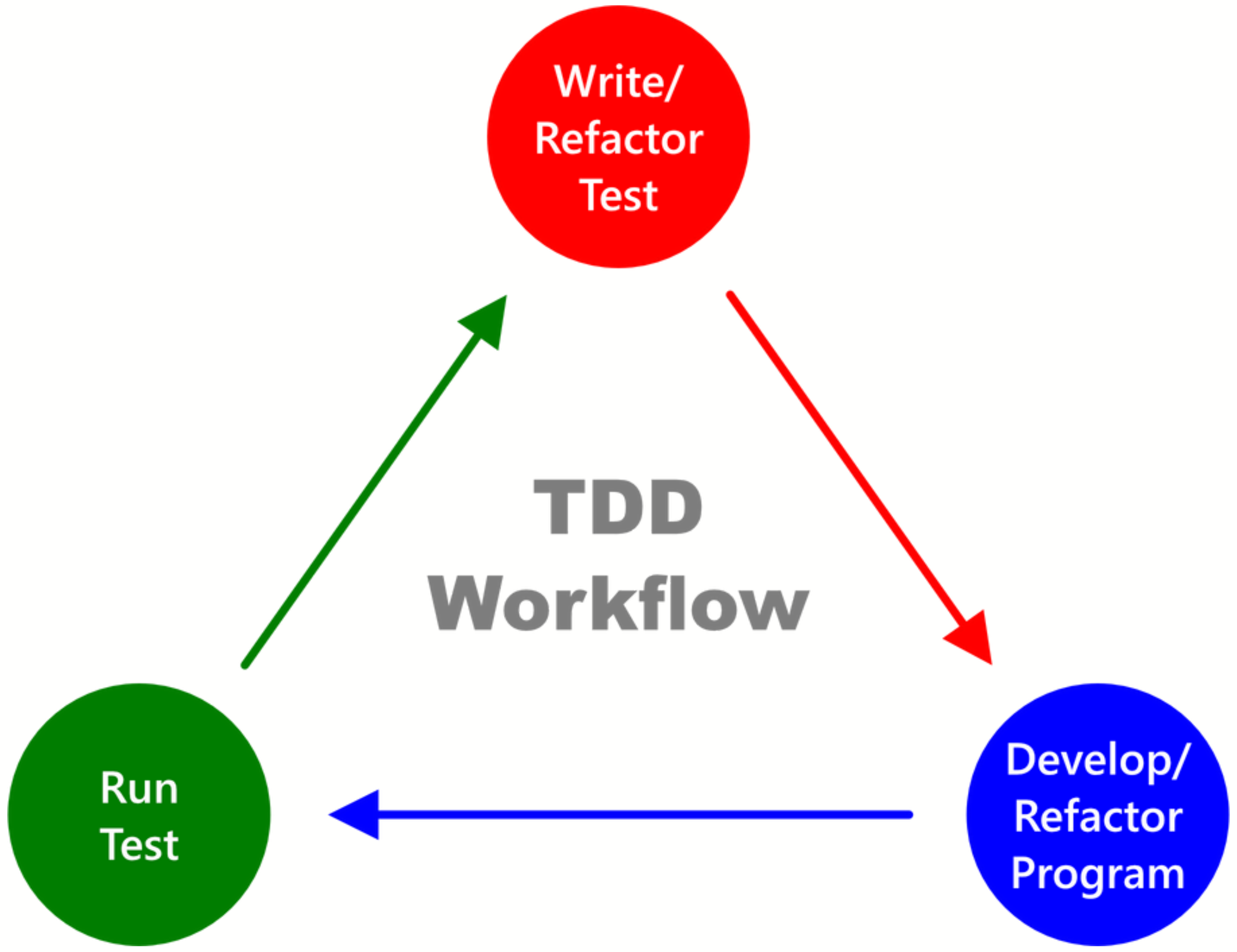
2 successful checks

✓ .github/workflows/deploy.yml / SpringCanteen Deploy (push) Successful in 2m

✓ Format / formatting (push) Successful in 14s

Testing Plan

- **Test Driven Development** with a **top-down approach**
- **Controller Testing With Mock Service** with **Mockito & Junit**
- **Service Testing With Mock Repository** with **Mockito & JUnit**
- **Data JPA Testing** - smoke testing as queries are implemented by Spring Data JPA
- **Integration testing** with **TestContainers**, **Flyway** and **RestAssured**
- **Functional tests** with **Selenium Webdriver + Cucumber** to validate user stories with a **black box** perspective



LIVE DEMO

Thank you for listening!