

# Fundamental Project Specification: Inventory Management System (IMS)

Sam Goodland

# Introduction

- Hello!
- BSc Grad. Applied Computing
- Approached domain requirements head-on
- In retrospect, additional planning would have aided

# Consultant Journey

- Technologies I was already familiar with: Java, SQL
- Technologies I'd used but not mastered: Git, GitHub
- Technologies I'd never used before: Jira, Maven, JUnit

# Continuous Integration (CI)

- Version control was approached with industry standards in mind

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with 'main' branch selected, '5 branches', and '0 tags'. Action buttons include 'Go to file', 'Add file', and 'Code'. A status bar indicates 'This branch is even with dev.' with links for 'Pull request' and 'Compare'.

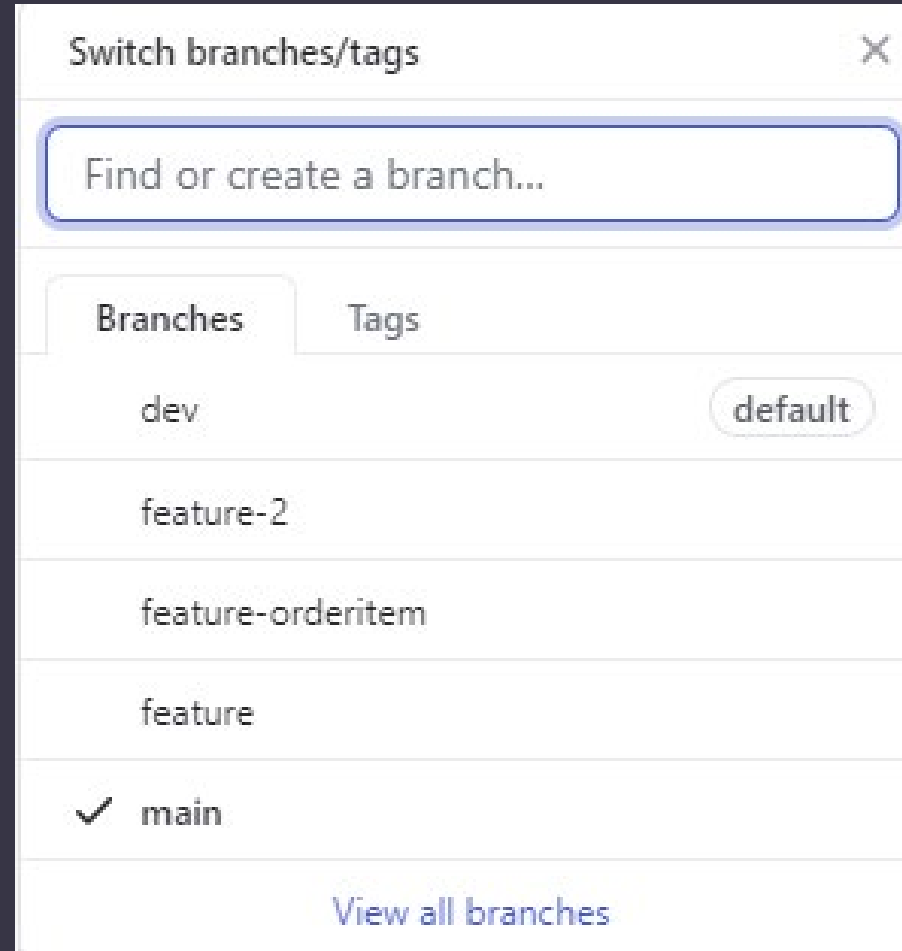
The main content area displays a commit history table:

Commit Hash	Commit Message	Time Ago
173d647	3 minutes ago	19 commits
src	SG: Finished Java coding	3 minutes ago
target/classes/com/qa/ims	SG: Fixed changes made to Utils that broke code	yesterday
.gitignore	SG: Added CRUD functionality for Orders	2 days ago
LICENSE.md	SG: Completed necessary CRUD functionality for Orders	3 days ago
QA-IMS.iml	SG: Added OrdersItem package	4 days ago
README.md	SG: Completed necessary CRUD functionality for Orders	3 days ago
pom.xml	SG: Completed necessary CRUD functionality for Orders	3 days ago

On the right sidebar, there are sections for 'Releases' (No releases published, Create a new release), 'Packages' (No packages published, Publish your first package), and 'Languages' (Java 100.0%).

# Continuous Integration (CI) – cont.

- Feature-dev-main branches utilised; new feature for every new part



# Testing

- Hardest part of the project
- Had the least amount of experience with

# Demonstration – the Codebase

Projects / ScrumProjectSam / SCRUM board

IMS Codebase

SG

Only My Issues

Recently Updated

0 days remaining

Complete sprint

TO DO

IN PROGRESS

DONE

As a developer, I need to make sure the package structure is sensible

IMS Codebase

SCRUM-25

As a developer, I need to achieve full CRUD functionality, with adherence to the Enterprise Architecture Model, for customers, items and orders

IMS Codebase

SCRUM-23

As a developer, I need to ensure my Java front-end connects via JDBC to a GCP-based MySQL instance

IMS Codebase

SCRUM-24

As a developer, I need to adhere to OOP, SOLID & refactoring

IMS Codebase

SCRUM-26

Quickstart



Completed Issues

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (-)
SCRUM-23 *	As a developer, I need to achieve full CRUD functionality, with adherence to the Enterprise Architecture Model, for customers, items and orders	<div></div> Story	<div></div> Medium	<div>DONE</div>	-
SCRUM-24 *	As a developer, I need to ensure my Java front-end connects via JDBC to a GCP-based MySQL instance	<div></div> Story	<div></div> Medium	<div>DONE</div>	-
SCRUM-25 *	As a developer, I need to make sure the package structure is sensible	<div></div> Story	<div></div> Medium	<div>DONE</div>	-
SCRUM-26 *	As a developer, I need to adhere to OOP, SOLID & refactoring	<div></div> Story	<div></div> Medium	<div>DO</div>	-

Quickstart



# Demonstration – the IMS repo

Projects /  ScrumProjectSam / SCRUM-6 /  
 SCRUM-12

As a developer, I need to ensure the application is built in the repo's root





Description

Projects /  ScrumProjectSam / SCRUM-6 /  
 SCRUM-11


As a developer, I need to make sure that the master branch can compile at the end of the project



Projects /  ScrumProjectSam / SCRUM-6 /  
 SCRUM-10

As a developer, I need to ensure that the repo is split into appropriate developer, feature and main/master branches











Projects /  ScrumProjectSam / SCRUM-6 /  
☒ SCRUM-7

As a developer, I need to initialise a new repo for the IMS project



## Completed Issues

[View in Issue Navigator](#)

Key	Summary	Issue Type	Priority	Status	Story Points (-)
SCRUM-7 *	As a developer, I need to initialise a new repo for the IMS project	 Task	 Medium	<span>DONE</span>	-
SCRUM-10 *	As a developer, I need to ensure that the repo is split into appropriate developer, feature and main/master branches	 Story	 Medium	<span>DONE</span>	-
SCRUM-11 *	As a developer, I need to make sure that the master branch can compile at the end of the project	 Story	 Medium	<span>DONE</span>	-
SCRUM-12 *	As a developer, I need to ensure the application is built in the repo's root	 Story	 Medium	<span>DO</span>	-

 Quickstart



# Demonstration – the Documentation

SG

Only My Issues

Recently Updated

TO DO

As a developer, I need to have a working .gitignore to ignore folders & build-generated files

IMS Documentation & Planning

↑

-

SCRUM-17

As a developer, I need to include a README.md that explains how the project application can be tested and ran

↑

-

SCRUM-18

As a developer, I need to keep all documentation in a folder, including a risk assessment, an ERD & UML, and a copy of my presentation in PDF format

↑

-

SCRUM-21

As a developer, I need to create a project management board with the following: user stories, acceptance criteria, story point estimation & MoSCoW prioritisation

↑

-

SCRUM-22

IN PROGRESS

DONE

Quickstart

Completed Issues

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (-)
SCRUM-7 *	As a developer, I need to initialise a new repo for the IMS project	<input checked="" type="checkbox"/> Task	↑ Medium	DONE	-
SCRUM-10 *	As a developer, I need to ensure that the repo is split into appropriate developer, feature and main/master branches	<div></div> Story	↑ Medium	DONE	-
SCRUM-11 *	As a developer, I need to make sure that the master branch can compile at the end of the project	<div></div> Story	↑ Medium	DONE	-
SCRUM-12 *	As a developer, I need to ensure the application is built in the repo's root	<div></div> Story	↑ Medium	DO	-

Quickstart

# Sprint Review

– what was completed

- All Codebase aspects, save the calculation of the total cost
- All Repository & Documentation aspects
- Continuous Integration aspects

## Sprint Review

– What was abandoned

```
public double calculateOrderCost(){  
    double cost = 0.0;  
    for (Item item : this.items){  
        cost += item.getItemValue();  
    }  
    return cost;  
}
```

- The total calculation of a cost
- The test coverage 80% threshold

# Sprint retrospective – what went well

- MySQL
- Java
- Git
- GitHub
- Maven

Sprint  
retrospective –  
what could  
have been  
done better

- Junit
- Jira

# Conclusions drawn

- 1. Time allocation is key
- 2. Knowledge of Java, GitHub, Git & SQL currently meets or is close to meeting expected standards
- 3. Additional JUnit practising will need to be conducted in spare time
- 4. Scope creep was biggest reason for project shortcomings

# Conclusion

- Thank you for listening!
- Are there any questions?