Project 1 - IMS

Work and presentation by Henry Oliver-Edwards

Introduction

- Part of the SDET2020DEC cohort
- Broke the specification down into smaller steps, made these steps into smaller actions I could take
- This broke the specification down into manageable little steps
- Work efficiently and have direction

Plan of action

Plan Of Action

Part 1 - Project setup

- Project management setup (Jira)
  - Create Jira Board
  - Create product backlog
  - Populate the board/backlog with user stories
  - Assess user stories and prioritise them
- Create risk assessment setup (?)
  - Must utilise matrix setup
  - .pdf format

Part 2 - Codebase and database setup

- Database (MySQL)
  - Create ERD Model of database
  - ERD copy in .png format
  - Create documents of fields in SQL database
  - Create local instance of MySQL database
- Version Control System (Git/Github)
  - Fork repository
  - Clone local version of that repository
  - Create Development branch

Part 3 - Development and testing

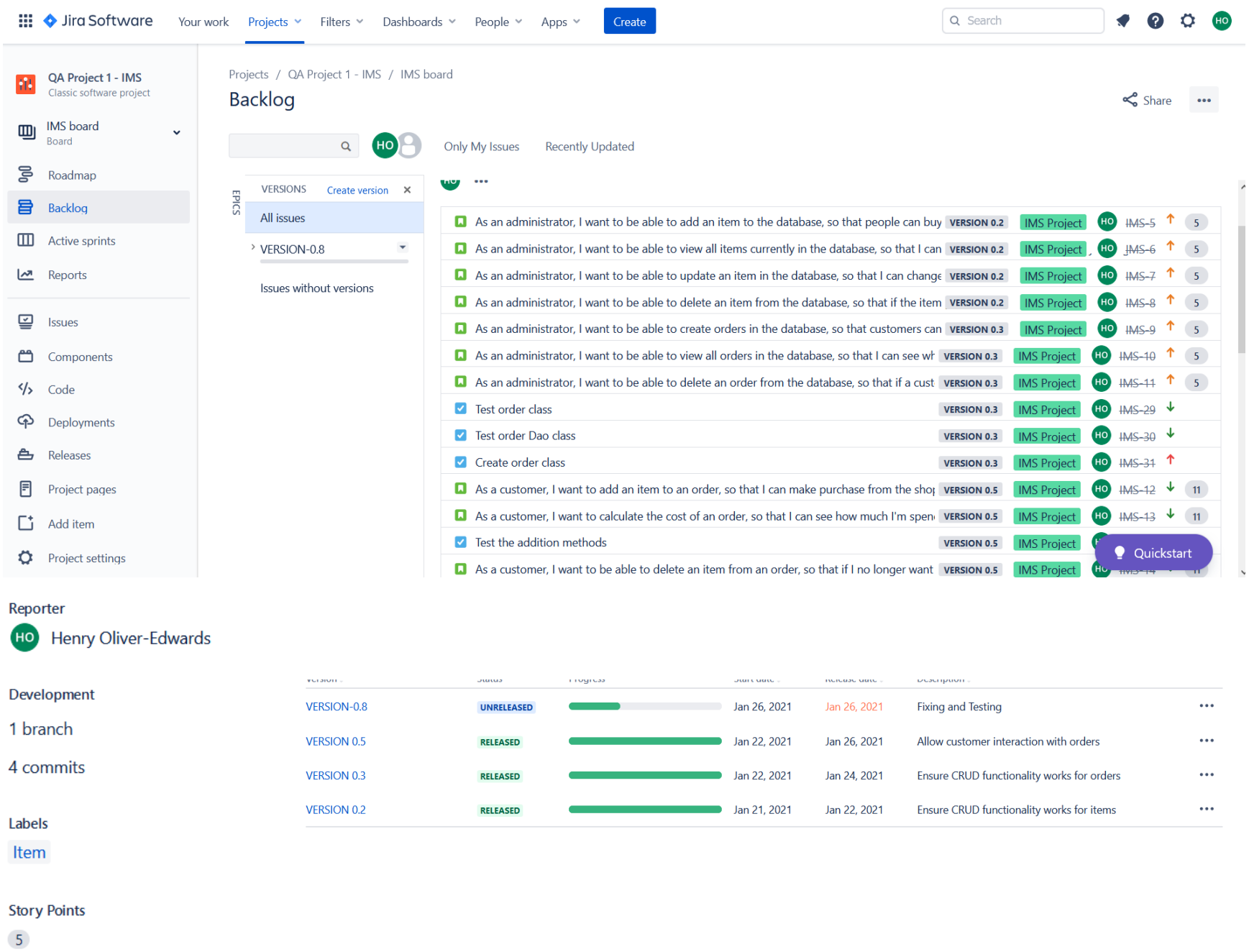
- Until Development has finished follow this cycle
  - VCS Management (Git)
    - Create feature branch on git
    - Switch to this branch
  - Development (Java)
    - Program the code for the desired feature, inspired by user stories

Management technologies and their usefulness

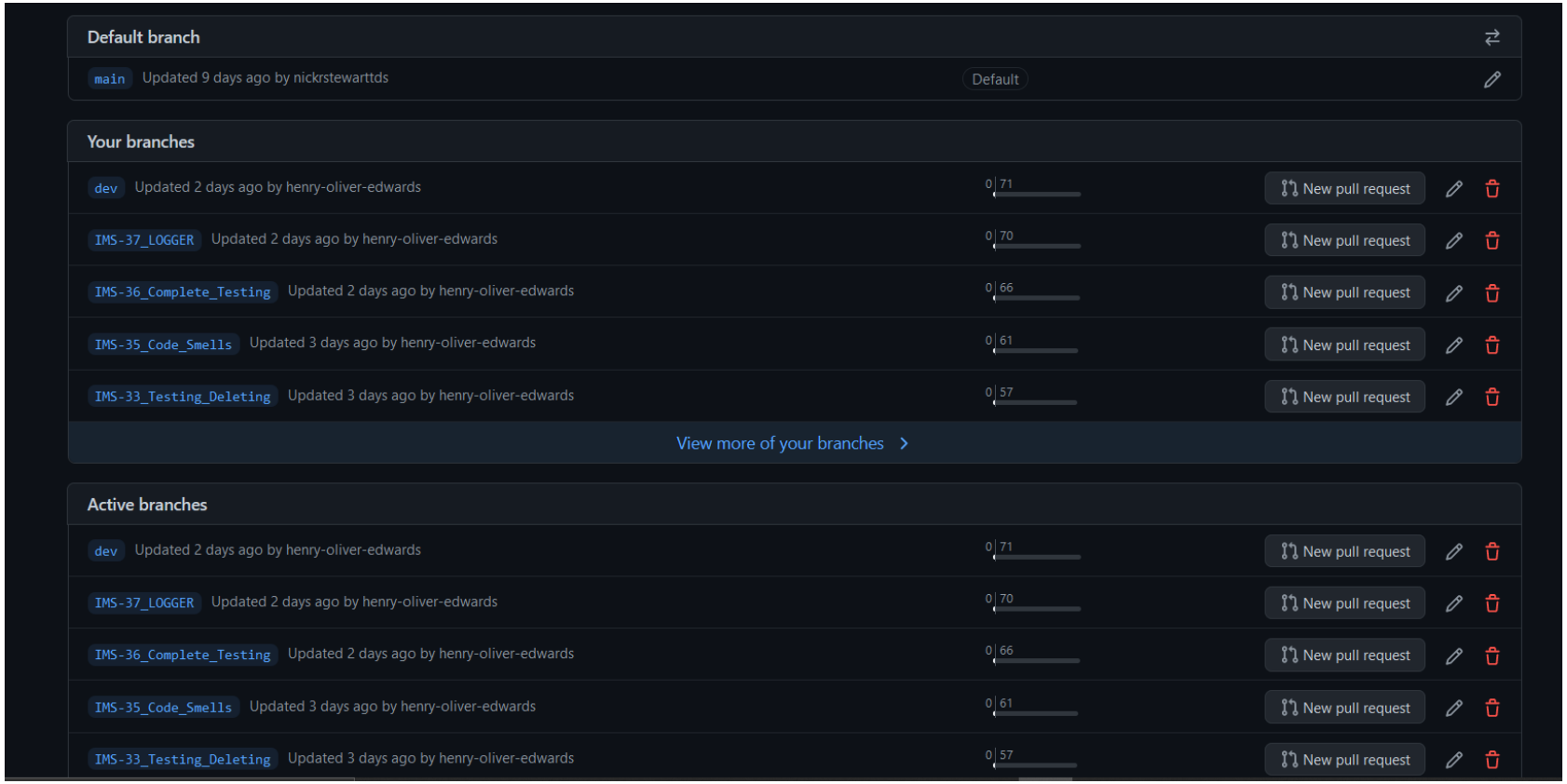
- Agile and scrum methodologies
- Jira
- Git/GitHub
- MySQL databases (and H2 databases) and the JDBC

- Agile and scrum provided me with a solid direction on how to work on this project, focus on what needs to get done
- Jira provided me with a way to manage my workflow and organize my user stories and tasks
- GitHub is a SCM tool that allows me to manage my code online and act as a backup
- MySQL (and H2) databases provided a way to store relational data and JDBC provided a way to interface with these databases

Jira Board (product backlog)



GitHub branches example



Project technologies and their usefulness

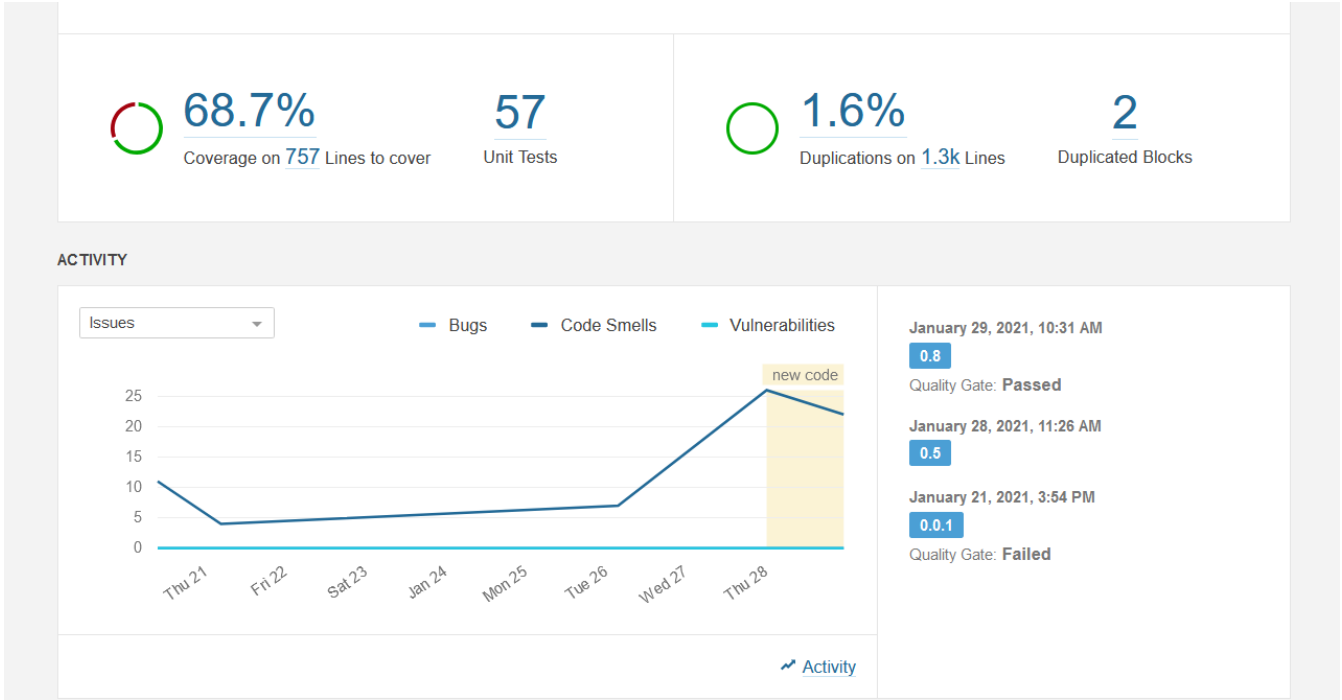
- Java and the JVM
- Maven
- JUnit
- Mockito
- SonarQube

- The project is entirely programmed in java and runs in the JVM
- Maven is a build tool which allows for the packing of java files into a fat-jar
- JUnit is a unit testing tool which individually tests the components of your code
- Mockito is a behavioral testing framework that allows us to test our codes output.
- SonarQube analyses code for bugs, security flaws and linting

Testing

- 75% overlal coverage
- Coverage on base classes, DAO classes and controller classes
- No coverage on menuing classes or runner class
- A mix of behavioural and and unit testing (using JUnit and Mockito)
- HTML coverage report at : Coverage Report :: Summary

SonarQube report



Demonstration time!

Sprint review

Completed

- All core CRUD functionality (for orders, customers and items)
- An aesthetically pleasing command line interface (CLI)
- Over 60% test coverage for the project

Not completed

- The ability to add lists to an order in the CLI, currently you can only add one item at a time
- 100% coverage would have been great to achieve
- Multiple UML diagrams showing a time lapse of the project
- A javadocs generated based on comments above the code

Sprint retrospective

What went well

- Code is clear and consise, hopefully it's self-documenting code, all funtions and variables are well named.
- All code and testing were completed within the given time frame.
- Management technologies utilised to some effect.

What could be improved

- Tasks kept being added to the Jira even after I had started the programming for the project.
- More time spend on making UMLs, too much time spent programming.

Conclusion

What I liked

- Interesting project utilising all the technologies we have learnt so far.
- The independence of the project was nice, programming at my own pace.
- Real life applications and implications.
- The management side made me feel like I was working for a real company.

What I didn't like

- I felt a little pressed for time.
- Some of the technologies I shoehorned in (such as mockito) because I only learnt them 2 days before the hand in date.
- Still not that confident in using Jira, especially with using it in a team.