QA - Inventory Management System Project (IMSP)

Adil Akbarali

Introduction

Adil Akbarali (https://github.com/adilakbarali)

- Broke down the scope into multiple bullet points
- Fork the project from GitHub and Import to Eclipse
- Base documentation
  - Designed an ERD
  - Set up a timescale for completion as well as a task list using a Kanban in Jira
- Validated results with the Deliverables Checklist for a MVP

Technologies Learned throughout the project

Below is a list of technologies that were both used and developed throughout:

- Git (Version Control and CI)
- Time management (Kanban using Jira)
- Database Management (MySQL)
- Programming (Java using Eclipse as an IDE)
- Building (Maven)
- Testing (JUnit using Mockito)

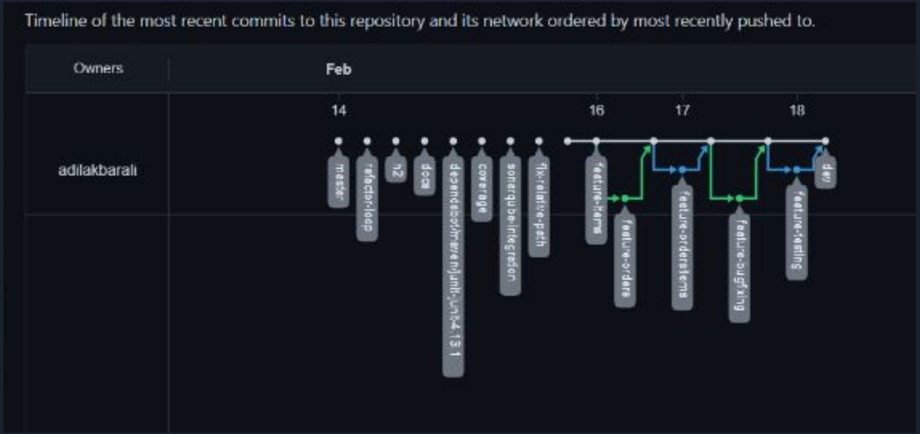
Already familiar with Java, Git and MySQL, however I was able to expand on my knowledge through the use of JDBC for database integration, Mockito for Unit testing, and Maven for package dependencies and building (other IDEs I have used previously had build tools packaged within)

Continuous Integration (CI) - Git

Using the feature-branch model, I implemented the addition of each Domain, Controller, and DAO as its’ own branch.

This gave room for reverting to previous state if accidental modifications were made elsewhere, allowing for identification of modified files outside of the scope of the branch.

GitHub Project



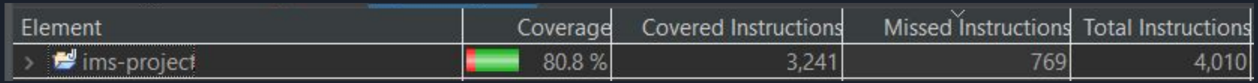
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

Testing (JUnit & Mockito)

Testing was carried out on any classes that were developed during the implementation of the project:

- Unit Testing
  - Controller Objects
  - Done using Mockito to “mock” objects
- Integration Testing
  - Data Access Objects (DAO)
  - Done using a test database, and running commands down the application stream.

Coverage Result: 80.8%



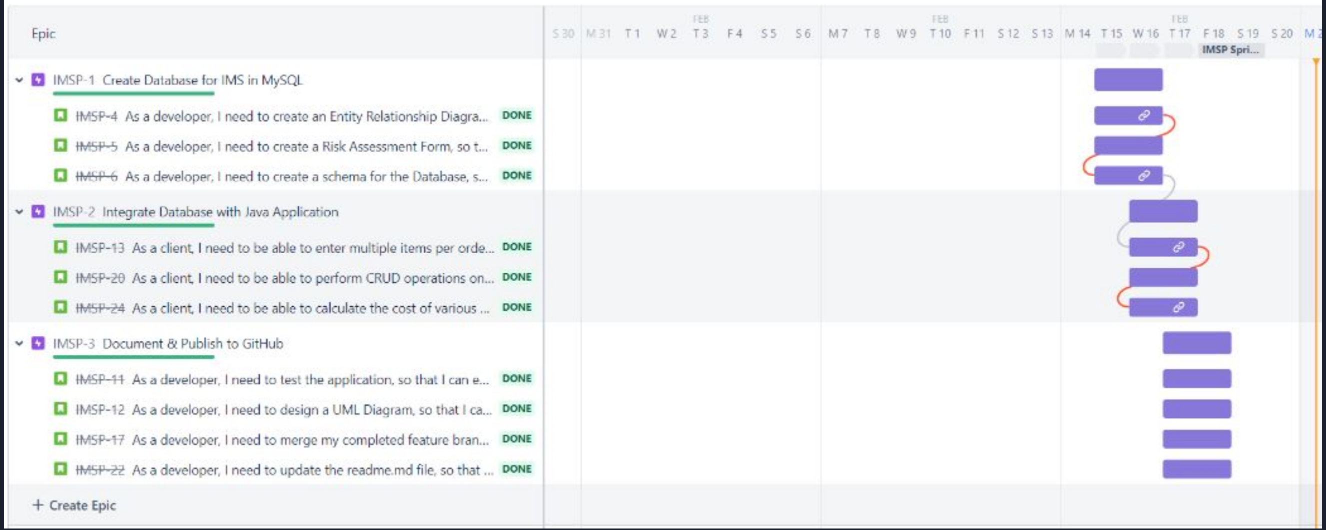
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
ims-project	80.8 %	3,241	769	4,010

Now time for a demonstration!

Sprint Review

All tasks labelled in all 3 sprints were completed successfully, with no tasks left behind, however ran slightly behind schedule versus the timespans set on Jira.

No extended functionality tasks were added to the Jira board, as the main focus was the MVP.



Epic

IMSP-1 Create Database for IMS in MySQL

- ~~IMSP-4~~ As a developer, I need to create an Entity Relationship Diagra... DONE
- ~~IMSP-5~~ As a developer, I need to create a Risk Assessment Form, so t... DONE
- ~~IMSP-6~~ As a developer, I need to create a schema for the Database, s... DONE

IMSP-2 Integrate Database with Java Application

- IMSP-13 As a client, I need to be able to enter multiple items per orde... DONE
- ~~IMSP-20~~ As a client, I need to be able to perform CRUD operations on... DONE
- ~~IMSP-24~~ As a client, I need to be able to calculate the cost of various ... DONE

IMSP-3 Document & Publish to GitHub

- ~~IMSP-11~~ As a developer, I need to test the application, so that I can e... DONE
- IMSP-12 As a developer, I need to design a UML Diagram, so that I ca... DONE
- ~~IMSP-17~~ As a developer, I need to merge my completed feature bran... DONE
- ~~IMSP-22~~ As a developer, I need to update the readme.md file, so that ... DONE

+ Create Epic

Sprint Retrospective

Things that went well:

- Breakdown of tasks allowed for tackling of larger tasks in smaller sections
- Assigning a time to tasks allowed me to understand the scale of the project
- Sprints allowed for me to tackle the project systematically

Things that didn’t go as planned:

- Keeping to the timescale
- Arrangement of particular tasks

General Issues and Problems Encountered

- First branch merge made it look like there was no additional branch merged into dev, instead commits made directly to the dev branch
- Eclipse did not want to use the specified Java version, therefor EqualsVerifier did not work (wrong Java Version)
- My particular implementation meant that I couldn’t use the h2 local instance database for testing

Conclusion

Overall, the project went very well despite a few bumps in the road along the way. Some things that I acknowledged through the completion of this:

- Allowed for development of skills across a range of technologies
- Enhanced time management skills
- I wish I had more time so that I could review extended functionality

Thank You!

Any questions?