Hobby Web Application

(HWA)

Shameer Dar
QA Trainee

Project Scope

As outlined in the document the main goal was to build a web application that a end-user can interact with via a graphical user interface (GUI), In this scenario I chose to build a web application for libraries utilising the CRUD functionalities.

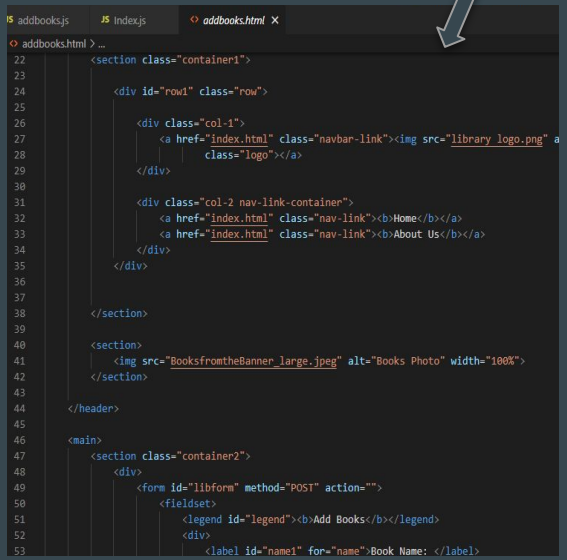
- **Jira Integration, creating user stories/issues to plan the project in the upcoming weeks.**
- **Entity diagram approach (ERD) to get a overview of how the tables will interact within the database.**
- **Created a SQL instance on Google Cloud Platform (Data Layer)**
- **Designing and Implementing HTML Pages (with CSS) for application (Presentation Layer)**
- **Started Coding; Backend Implementation – JAVA SpringBoot (Business Layer)**
- **Started Coding; Frontend Implementation – JAVA Script**
- **Testing & debugging**

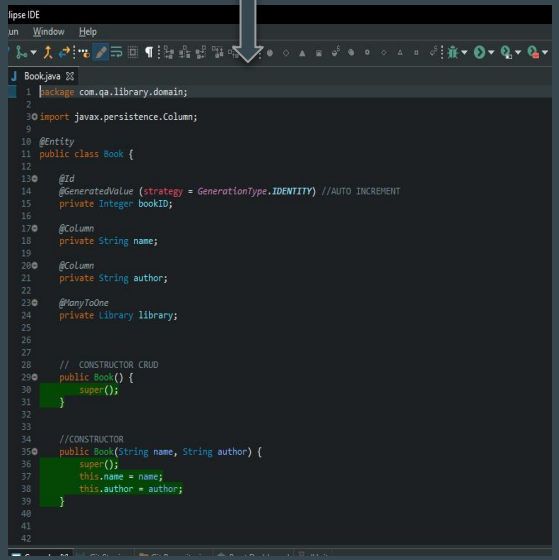
C.R.U.D

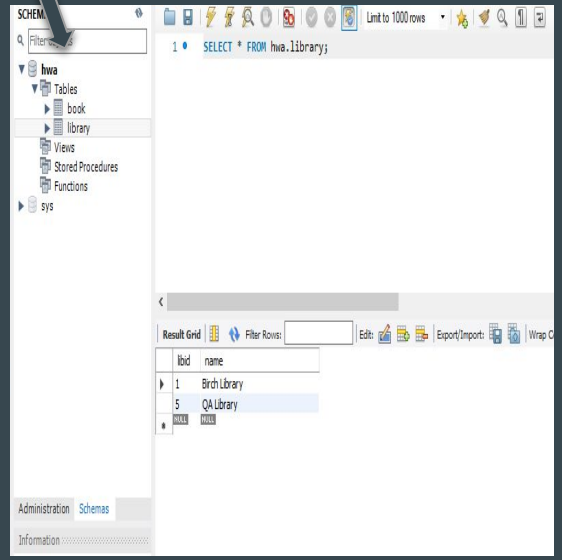
Create. **R**ead. **U**pdate. **D**elete

Functionality

Multi-Tier Architecture







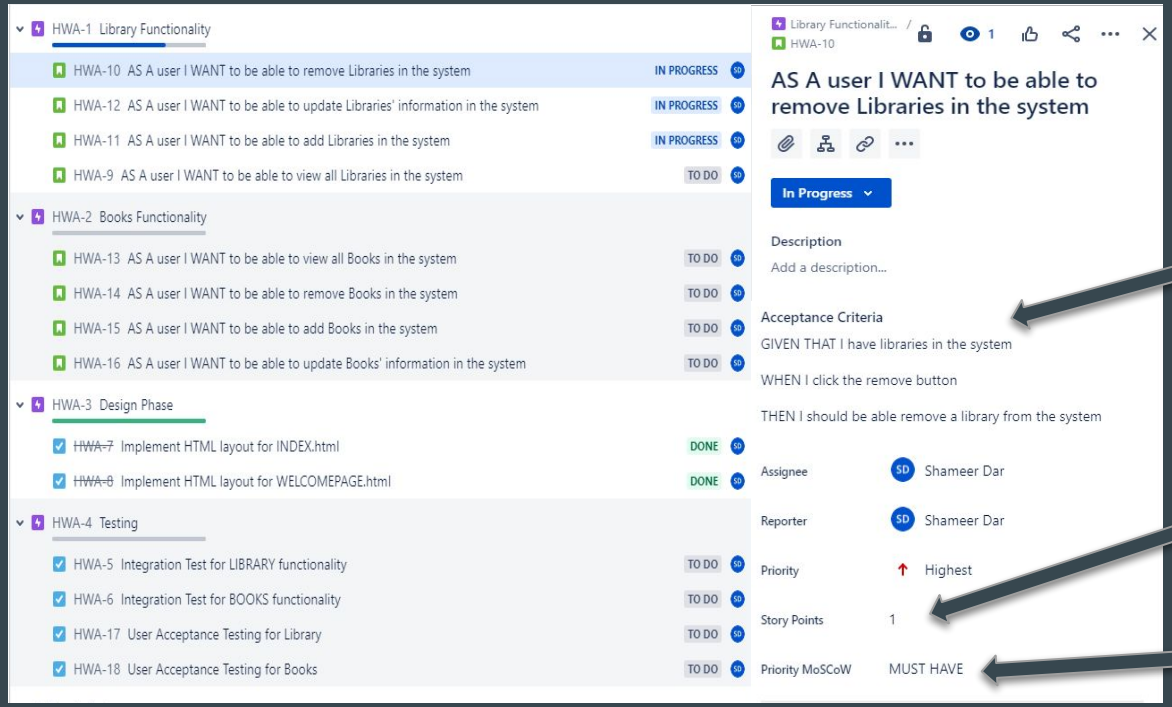
Consultant Journey

Here are the various technologies that were utilised for the completion of this project:

- **Project Management Tool: Jira**
- **Version Control: Git, GitBash, GitHUB**
- **Database Management System: H2, MySQL, Workbench**
- **Back-End Programing Language: Java**
- **Front-End Web Technologies: HTML, CSS, Java Script**
- **Cloud Platform: Google Cloud Platform**
- **Build Tool: Maven**
- **API Development: Spring, Postman**
- **Testing : JUnit + Mockito, Selenium, SonarQube**

Continuous Integration Jira

User Stories with Epics

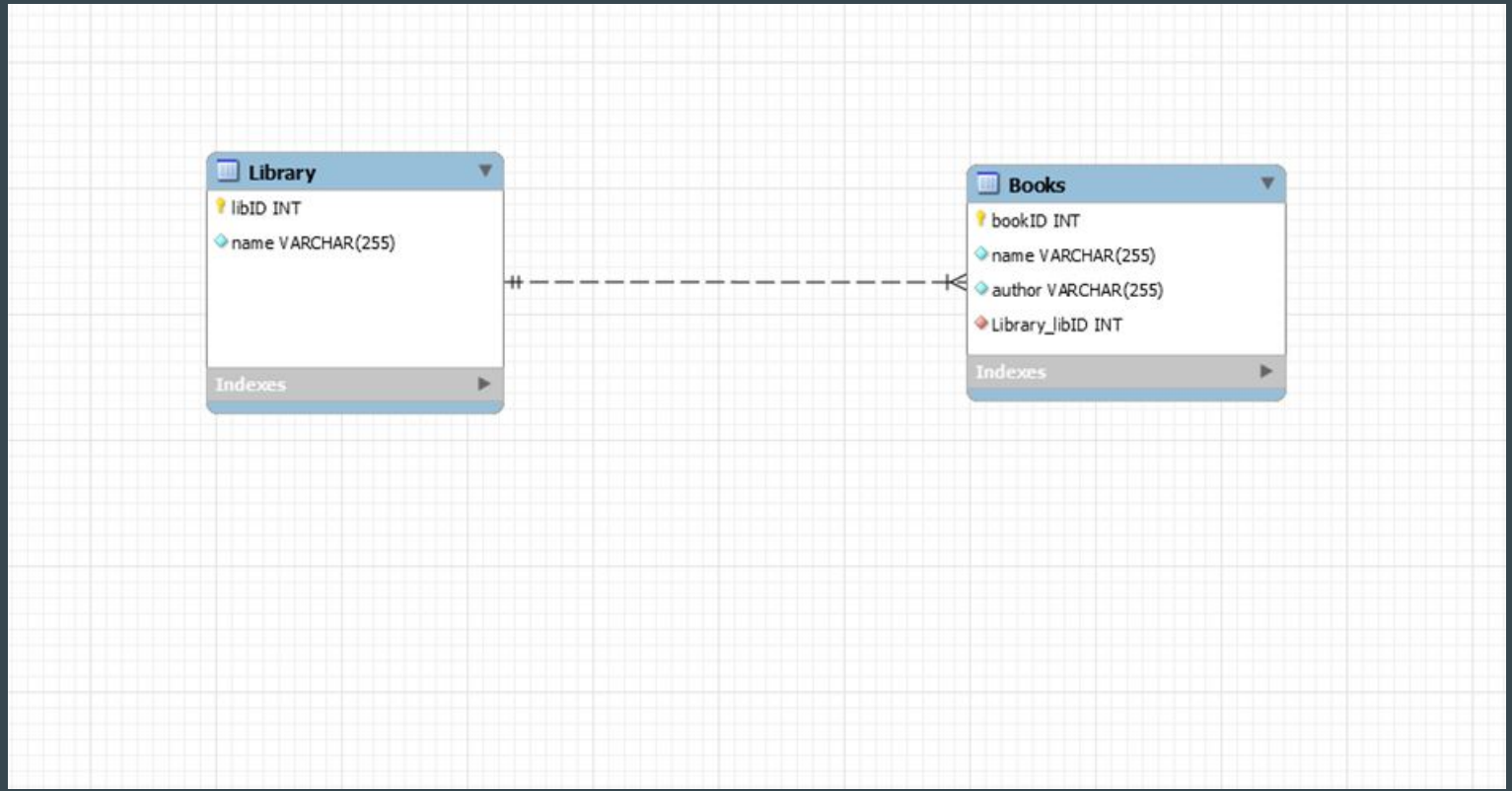


Acceptance Criteria

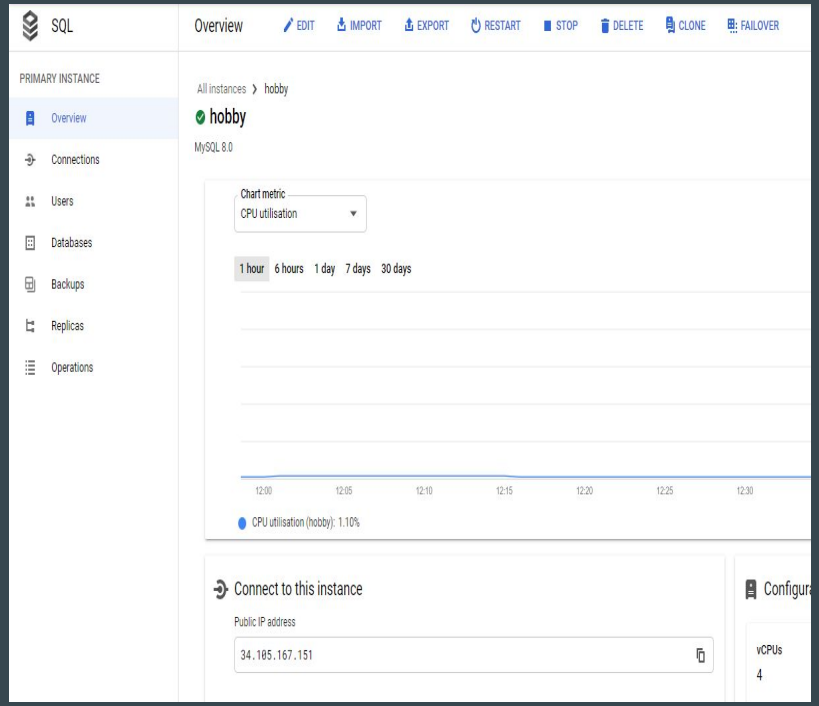
Story Points

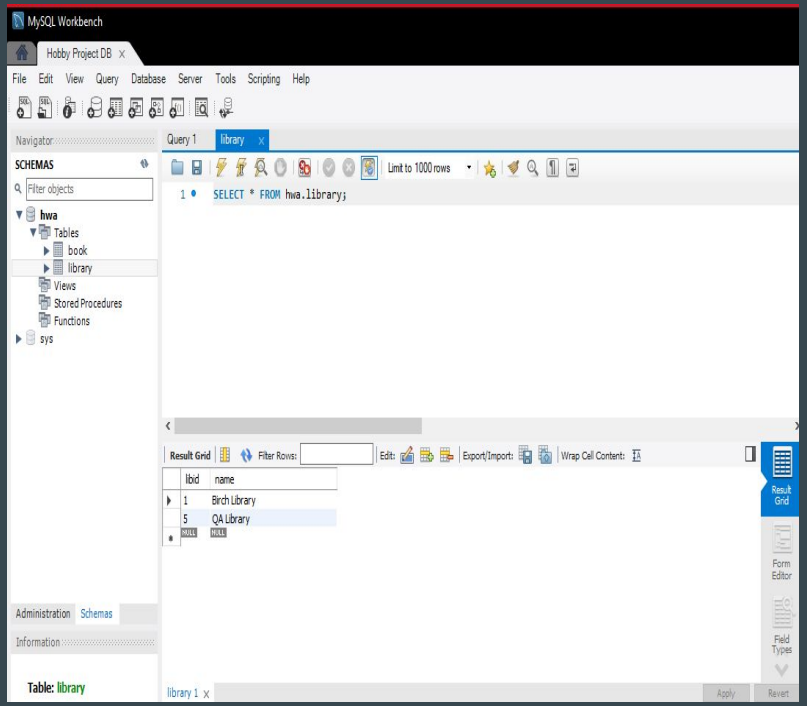
Priority MoSCoW

ERD Diagram


Library
libID INT
name VARCHAR(255)
Indexes
Books
bookID INT
name VARCHAR(255)
author VARCHAR(255)
Library_libID INT
Indexes

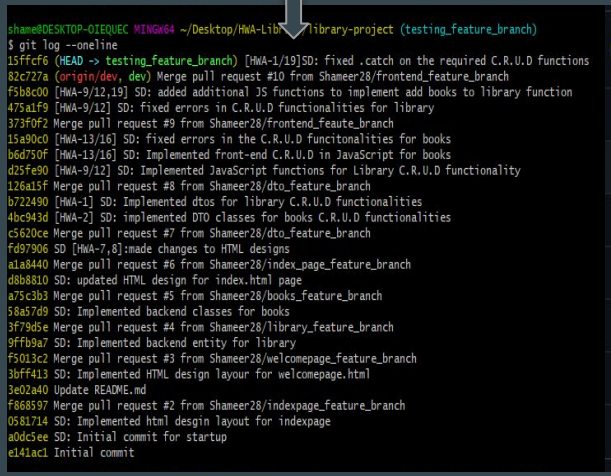
MySQL via Google Cloud Platform(GCP)

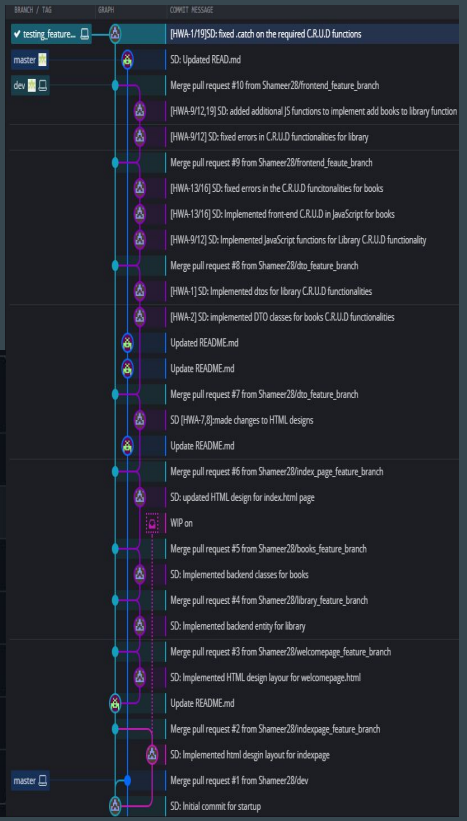




Continuous Integration GitHUB

- Utilised feature_branch model
- Made regular commits.
- Push to GitHUB repository

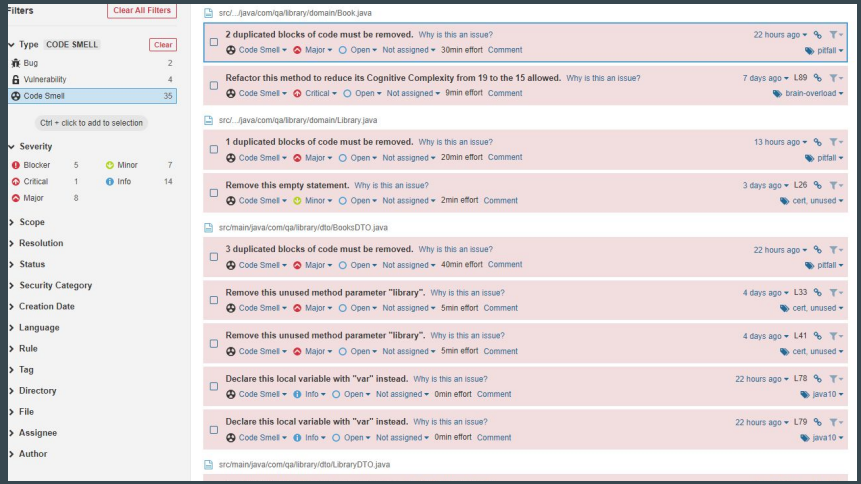




Testing

- For integration testing as previously mentioned JUnit+Mockito dependencies were used.
- For user-acceptance testing selenium dependencies was used
- In addition SonarQube was used as an extension to fix/refactor any bad codes

SonarQube



Test Coverage

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
HWA-Library	92.9 %	2,154	165	2,319
src/main/java	81.3 %	675	155	830
com.qa.library.domain	75.4 %	196	64	260
com.qa.library.dto	75.3 %	165	54	219
com.qa.library	15.0 %	3	17	20
com.qa.library.service	92.1 %	187	16	203
com.qa.library.utils	94.4 %	67	4	71
com.qa.library.rest	100.0 %	57	0	57
src/test/java	99.3 %	1,479	10	1,489
com.qa.library.integration	99.0 %	384	4	388
com.qa.library.unit	99.5 %	750	4	754
com.qa.library.acceptance	99.4 %	341	2	343
com.qa.library	100.0 %	4	0	4

Application Demonstration

Will now give a demo on my application carrying out the following user stories:

- **AS A** user **I WANT** to be able to add libraries in the system
- **AS A** user **I WANT** to be able to add books to a library
- **AS A** user **I WANT** to be able to view all books in the system
- **AS A** user **I WANT** to be able to update a book in the system
- **AS A** user **I WANT** to be able to delete a book in the system

Sprint Review

What did you complete?

- Full stack web application.
- CRUD functionality implemented.
- Testing <80%.

What got left behind?

- Everything completed in terms of MVPs.
- Website design not fully implemented, due to time constraints.

Sprint Retrospective

What went well?

- Constructed a full stack web application with the Minimum Viable Products.
- Good Initial understanding of automated testing (Selenium).
- Understanding for API development tools (Spring).
- Good initial understanding of front-end technologies; HTML, CSS, JavaScript.

What could be improved?

- Expand my knowledge on testing as a whole to be adapt a more TDD approach.
- Spend more time practicing with front-end technologies.
- Improve time management

Conclusion

Reflection on the project

- This project has allowed me to learn a lot of technologies mainly around how a full stack web application is implemented (front-end/back-end)
- Good understanding of testing code
- Very Confident in version control using Git, GitBash, GitHUB along with JIRA CI.

Future next steps

- Practice creating tests (unit, integration, user acceptance)
- Keep practicing full stack development

More project info can be found in the Documentation folder on the GitHUB Repository

Thank You for Listening

Questions?