

# Hobby Web Application

(HWA)

**Shameer Dar**  
QA Trainee

# Project Scope

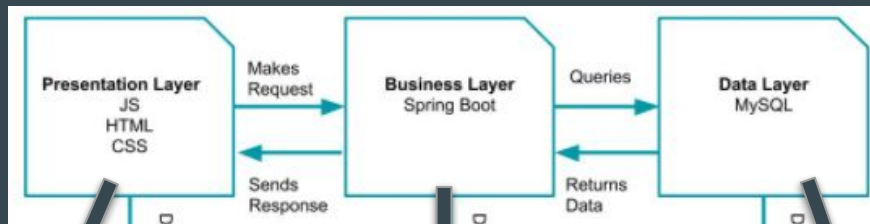
As outlined in the document the main goal was to build a web application that a end-user can interact with via a graphical user interface (GUI), In this scenario I chose to build a web application for libraries utilising the CRUD functionalities.

- **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. Read. Update. Delete**  
**Functionality**

# Multi-Tier Architecture



```
addbooks.html > ...
22 <section class="container1">
23
24 <div id="row1" class="row">
25
26 <div class="col-1">
27 <a href="index.html" class="nav-link">
28 </a>
29 </div>
30
31 <div class="col-2 nav-link-container">
32 <a href="index.html" class="nav-link"><b>Home</b></a>
33 <a href="index.html" class="nav-link"><b>About Us</b></a>
34 </div>
35 </div>
36
37 </section>
38
39 <section>
40 
41 </section>
42
43 </header>
44
45
46 <main>
47 <section class="container2">
48 <div>
49 <form id="libform" method="POST" action="">
50 <fieldset>
51 <legend id="legend"><b>Add Books</b></legend>
52 <div>
53 <label id="name1" for="name">Book Name: <input type="text" />
```

```
Book.java
1 package com.qa.library.domain;
2
3 import javax.persistence.Column;
4
5 @Entity
6 public class Book {
7
8     @Id
9     @GeneratedValue (strategy = GenerationType.IDENTITY) //AUTO INCREMENT
10    private Integer bookID;
11
12    @Column
13    private String name;
14
15    @Column
16    private String author;
17
18    @ManyToMany
19    private Library library;
20
21    // CONSTRUCTOR CRUD
22    public Book() {
23        super();
24    }
25
26    //CONSTRUCTOR
27    public Book(String name, String author) {
28        super();
29        this.name = name;
30        this.author = author;
31    }
32
33
34
35
36
37
38
39
40
41
42
```

The screenshot shows a database management tool interface. On the left, a schema diagram displays a table named 'book' within a 'library' database. The 'book' table has columns 'id' and 'name'. On the right, a query window shows the SQL statement `SELECT * FROM hwa.library;`. Below the query, a result grid displays the following data:

libd	name
1	Birch Library
5	QA Library

# Consultant Journey

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

- Project Management Tool: Jira

- Cloud Platform: Google Cloud Platform

- Version Control: Git, GitBash, GitHUB

- Build Tool: Maven

- Database Management System: H2, MySQL, Workbench

- API Development: Spring,

Postman

- Back-End Programming Language: Java

- Testing : JUnit + Mockito, Selenium,

SonarQube

- Front-End Web Technologies: HTML, CSS, Java Script

# Continuous Integration Jira

## User Stories with Epics

The screenshot displays the Jira interface for a project named 'Library Functionalit...'. On the left, a sidebar shows a hierarchy of epics and their associated user stories:

- HWA-1 Library Functionality**
  - HWA-10 AS A user I WANT to be able to remove Libraries in the system (IN PROGRESS)
  - HWA-12 AS A user I WANT to be able to update Libraries' information in the system (IN PROGRESS)
  - HWA-11 AS A user I WANT to be able to add Libraries in the system (IN PROGRESS)
  - HWA-9 AS A user I WANT to be able to view all Libraries in the system (TO DO)
- HWA-2 Books Functionality**
  - HWA-13 AS A user I WANT to be able to view all Books in the system (TO DO)
  - HWA-14 AS A user I WANT to be able to remove Books in the system (TO DO)
  - HWA-15 AS A user I WANT to be able to add Books in the system (TO DO)
  - HWA-16 AS A user I WANT to be able to update Books' information in the system (TO DO)
- HWA-3 Design Phase**
  - HWA-7 Implement HTML layout for INDEX.html (DONE)
  - HWA-8 Implement HTML layout for WELCOME PAGE.html (DONE)
- HWA-4 Testing**
  - HWA-5 Integration Test for LIBRARY functionality (TO DO)
  - HWA-6 Integration Test for BOOKS functionality (TO DO)
  - HWA-17 User Acceptance Testing for Library (TO DO)
  - HWA-18 User Acceptance Testing for Books (TO DO)

The right pane shows the detailed view of user story HWA-10:

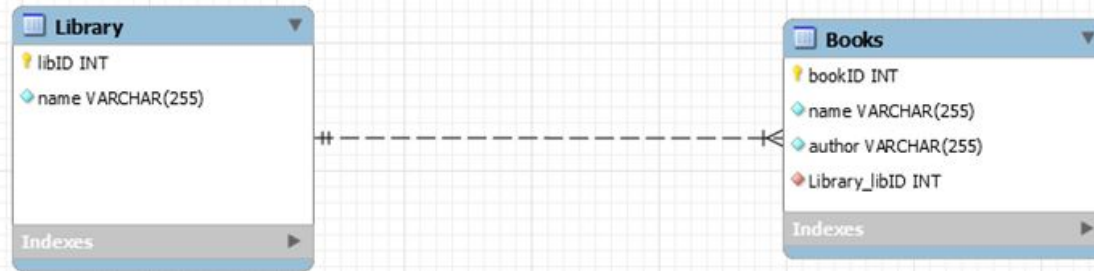
- Title:** AS A user I WANT to be able to remove Libraries in the system
- Status:** In Progress
- Description:** Add a description...
- Acceptance Criteria:**
  - GIVEN THAT I have libraries in the system
  - WHEN I click the remove button
  - THEN I should be able remove a library from the system
- Assignee:** Shameer Dar
- Reporter:** Shameer Dar
- Priority:** Highest
- Story Points:** 1
- Priority MoSCoW:** MUST HAVE

Acceptance Criteria

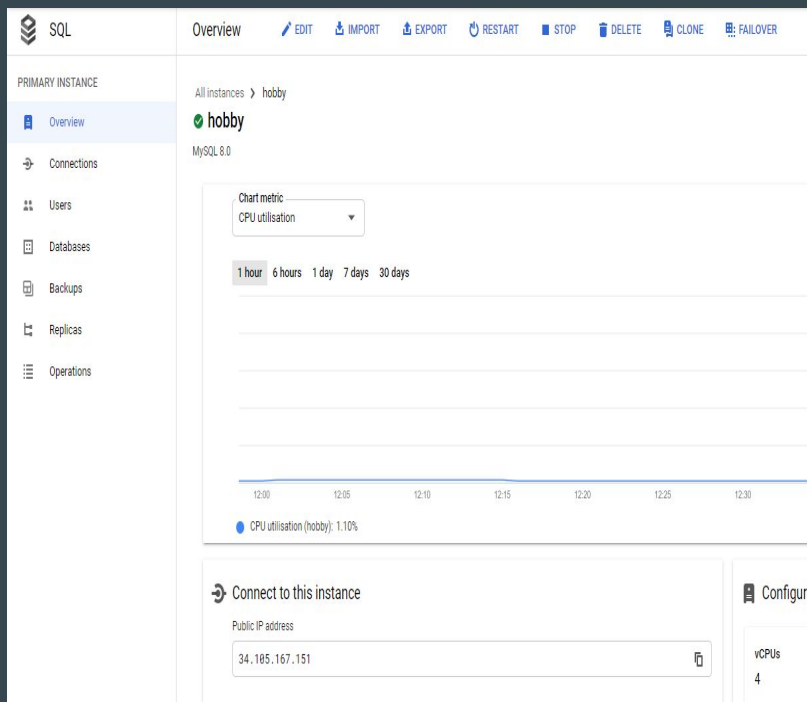
Story Points

Priority MoSCoW

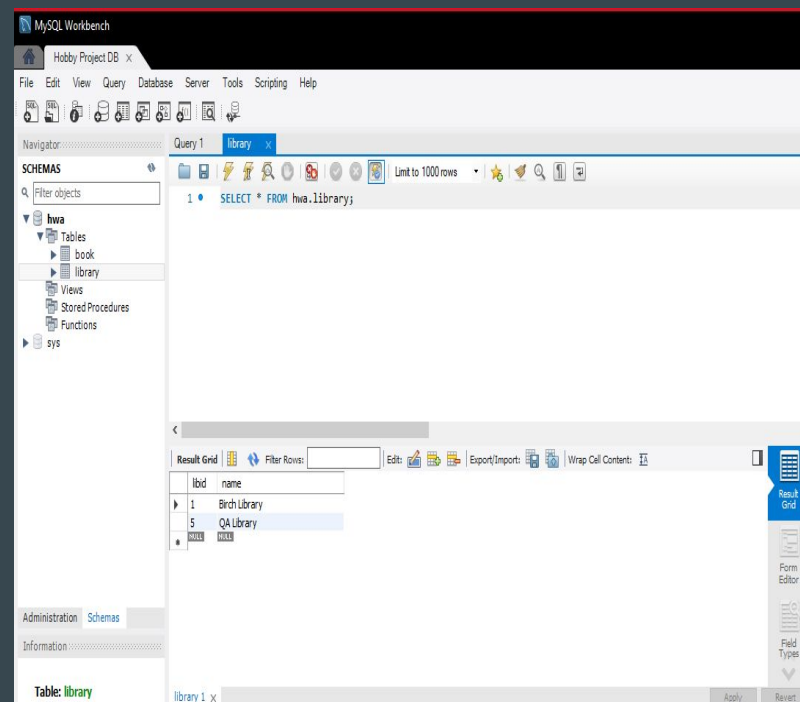
# ERD Diagram



# MySQL via Google Cloud Platform(GCP)



The screenshot shows the Google Cloud SQL console interface. On the left, there's a sidebar with navigation links: Overview (selected), Connections, Users, Databases, Backups, Replicas, and Operations. The main area displays the 'Overview' for a MySQL instance named 'hobby'. It shows the instance is running on MySQL 8.0. Below this, there's a 'Chart metric' dropdown set to 'CPU utilisation' and a time range selector with options: 1 hour, 6 hours, 1 day, 7 days, and 30 days. A line chart shows the CPU utilisation over time, with a current value of 1.10% at 12:00. At the bottom, there's a 'Connect to this instance' section showing the public IP address as 34.185.167.151 and the number of vCPUs as 4.



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The 'Navigator' pane on the left shows the 'hwa' database with tables 'book' and 'library', views, stored procedures, and functions. The 'Query' pane shows a query: `SELECT * FROM hwa.library;`. The 'Result Grid' pane shows the results of the query, which are two rows from the 'library' table:

libid	name
1	Birch Library
5	QA Library

The bottom status bar indicates the current table is 'library'.


# Continuous Integration GitHub

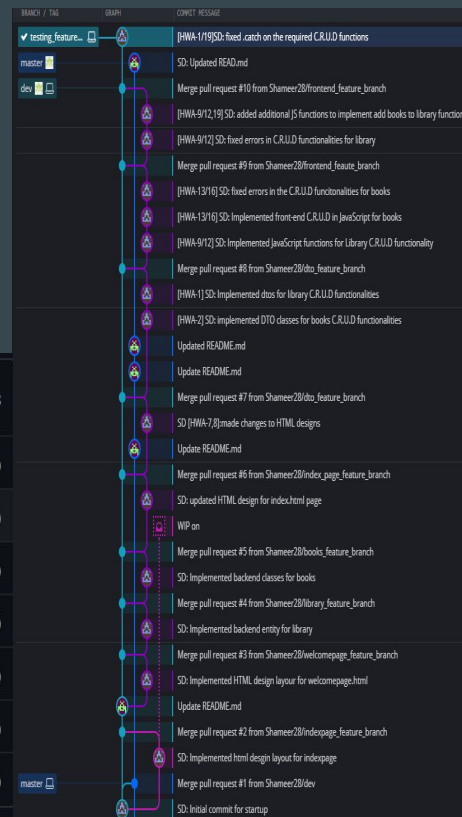
- Made regular commits.

```
shame@DESKTOP-01EQUEC MINGW64 ~/Desktop/HWA-Lib/library-project (testing_feature_branch)
$ git log --oneline
15ffcf6 (HEAD -> testing_feature_branch) [HWA-1/19]SD: fixed .catch on the required C.R.U.D functions
82c727a (origin/dev, dev) Merge pull request #10 from Shameer28/frontend_feature_branch
f3b8c00 [HWA-9/12,19] SD: added additional JS functions to implement add books to library function
475a1f9 [HWA-9/12] SD: fixed errors in C.R.U.D functionalities for library
373f0f2 Merge pull request #9 from Shameer28/frontend_feaute_branch
15a90c0 [HWA-13/16] SD: fixed errors in the C.R.U.D functionalities for books
b6d750f [HWA-13/16] SD: Implemented front-end C.R.U.D in JavaScript for books
d25fe90 [HWA-9/12] SD: Implemented JavaScript functions for Library C.R.U.D functionality
126a15f Merge pull request #8 from Shameer28/dto_feature_branch
b722490 [HWA-1] SD: Implemented dtos for library C.R.U.D functionalities
4bc943d [HWA-2] SD: implemented DTO classes for books C.R.U.D functionalities
c5620ce Merge pull request #7 from Shameer28/dto_feature_branch
fd97906 SD [HWA-7,8]:made changes to HTML designs
a1a8440 Merge pull request #6 from Shameer28/index_page_feature_branch
d8b8810 SD: updated HTML design for index.html page
a75c3b3 Merge pull request #5 from Shameer28/books_feature_branch
58a57d9 SD: Implemented backend classes for books
3f79d5e Merge pull request #4 from Shameer28/library_feature_branch
9ff9ba7 SD: Implemented backend entity for library
f5013c2 Merge pull request #3 from Shameer28/welcomepage_feature_branch
3bffa13 SD: Implemented HTML design layout for welcomepage.html
3e02a40 Update README.md
f868597 Merge pull request #2 from Shameer28/indexpage_feature_branch
0581714 SD: Implemented html desgin layout for indexpage
a0dc5ee SD: Initial commit for startup
e141ac1 Initial commit
```

- Utilised feature\_branch model

- Push to GitHub repository

Shameer28 Merge pull request #10 from Shameer28/frontend_feature_branch 82c727a 29 minutes ago 25 commits		
	.mvnw/wrapper	SD: Initial commit for startup 6 days ago
	src	[HWA-9/12,19] SD: added additional JS functions to implement add book... 30 minutes ago
	.gitignore	SD: Initial commit for startup 6 days ago
	README.md	Update README.md 6 days ago
	mvnw	SD: Initial commit for startup 6 days ago
	mvnw.cmd	SD: Initial commit for startup 6 days ago
	pom.xml	SD: Initial commit for startup 6 days ago





# 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

## Test Coverage

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ HWA-Library				
▼ src/main/java	92.9 %	2,154	165	2,319
> com.qa.library.domain	81.3 %	675	155	830
> com.qa.library.dto	75.4 %	196	64	260
> com.qa.library	75.3 %	165	54	219
> com.qa.library.service	15.0 %	3	17	20
> com.qa.library.utils	92.1 %	187	16	203
> com.qa.library.rest	94.4 %	67	4	71
> com.qa.library	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

## SonarQube

The screenshot displays the SonarQube web interface. On the left, a 'Filters' sidebar allows filtering by Type (CODE SMELL, Bug, Vulnerability, Code Smell), Severity (Blocker, Critical, Major, Minor, Info), Scope, Resolution, Status, Security Category, Creation Date, Language, Rule, Tag, Directory, File, Assignee, and Author. The main panel shows a list of issues for the project 'src/\_java/com/qa/library/domain/Book.java'. The issues include '2 duplicated blocks of code must be removed', 'Refactor this method to reduce its Cognitive Complexity from 19 to the 15 allowed', '1 duplicated blocks of code must be removed', 'Remove this unused method parameter "library"', 'Remove this unused method parameter "library"', 'Declare this local variable with "var" instead', and 'Declare this local variable with "var" instead'. Each issue entry shows its severity, effort, and a comment.

# 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?**