QA Project 2

QA

Introduction and Concept

I am Ben Phillips. Outside of my experience with QA I have little structured coding experience but I have been self teaching for up to a year.

I approached the specification with a specific logical order. First I wanted to order and finish any user stories that would be relevant and get them onto a Kanban board so it would be easy to find and see what the project needed.

Then I wanted to focus on the Spring api with a basic structure including JavaScript, CSS and a basic HTML page for a first commit.

Then I could focus on the backend to get the database working and the basic functionalities working before focusing on the next stage which would be testing.

Once everything was working, I wanted to focus on the front end to create the app interface that a user would be using.

Sprint Plan

With such a short time to deliver this project I had to think about the best way to organise the project.

Day 1 - User Stories and Kanban board and start with basic HTML features

Day 2 - HTML and JS utilisation

Day 3 - Focus on finishing backend and testing

Day 4 - Front end

Day 5 - Tidy up and documentation

Consultant Journey

The technologies that I have used for this project includes:

Maven - Dependency and Build Management

Git - Gitbash functionality for command line interface

Github - Repository management and integration with Jira for automatic Jira update

Confluence - Additional pages for risk matrices and further documentation

Mockito - Mocking dependencies for unit testing

Junit - Testing

MySQLWorkbench and Server

Jira - Kanban board and epic/stories visualisation

PlantUML - UML creation

CI and Testing

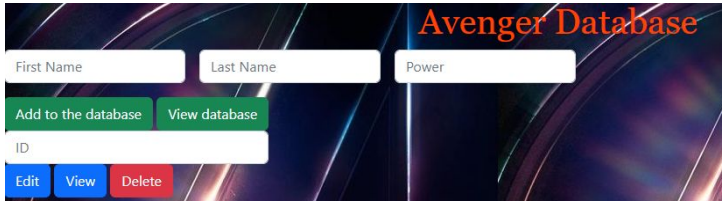
Version Control was managed with different branches using GitBash and GitHub. With this method I could push functioning code to the main branch for a working application interface and continue working on the code in the dev-feature1 branch. This also ensured that I had multiple states of code saved in a repository on github in case of a catastrophic data failure.

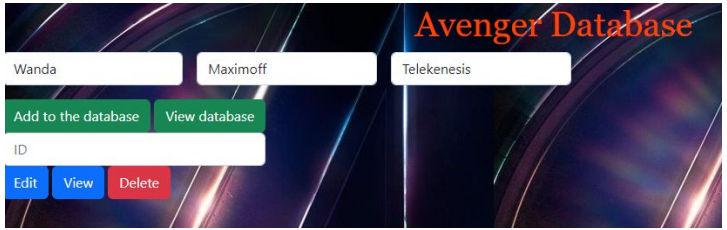
The testing was done with JUnit and Mockito and it covers all aspects of the application from the controllers to the services.

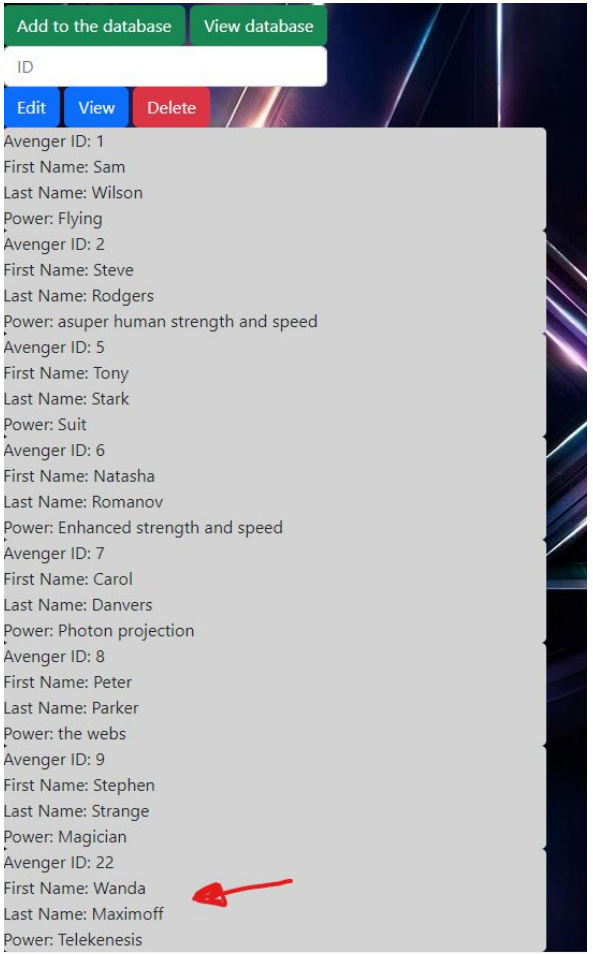
Demonstration

Enter the details you wish to add and click add to the database

To view the entry in the database, click view database

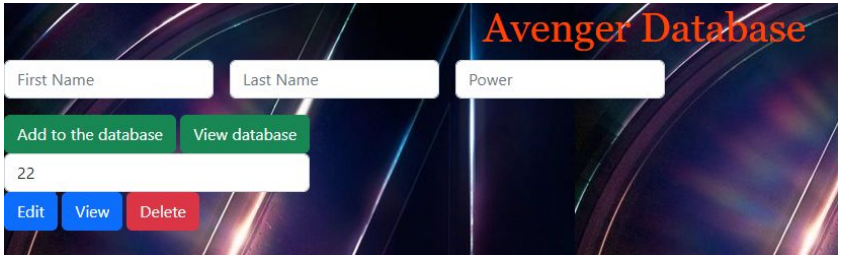


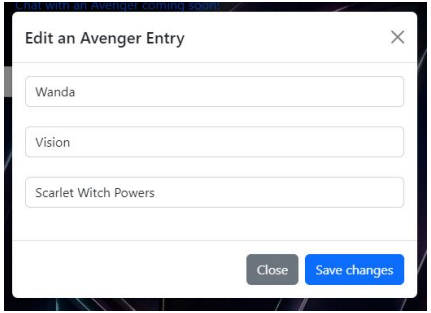




Demonstration

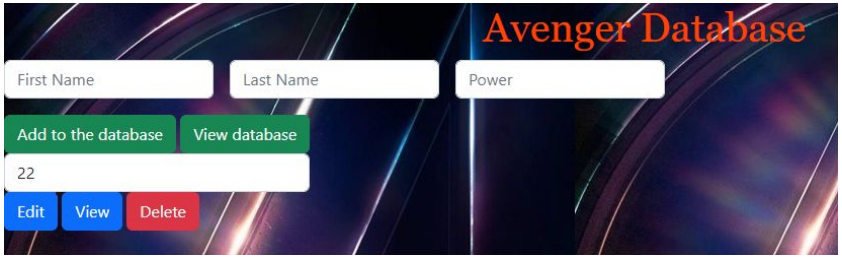
To edit an entry, enter the Id of the entry you wish to edit and then click edit
When the tab opens, enter the new information you wish to enter and click submit
The entry will then be changed

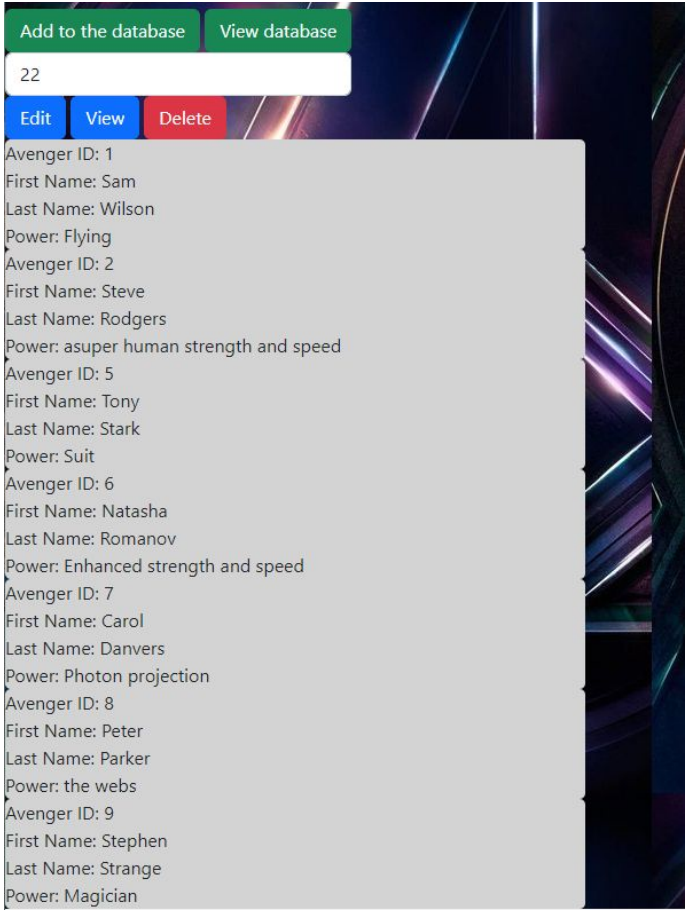




Demonstration

To delete an entry, enter the Id of the entry you wish to delete and then click delete
The entry will then be deleted





Sprint Review

I believe the sprint went well, however I did get stuck on a few basic syntax errors such as brackets where they shouldn't have been. Unfortunately on one of the days I was without wifi for half of the day which slowed me down.

I am pleased with the output that was achieved and with only one of the targeted user stories being unachieved, I believed a lot was accomplished.

Sprint Retrospective

I think more could have been done with the CSS aspect of the project as the design and layout of the application is not up the standard that I wanted. I wanted the application to flow and have a user friendly design which I believe it currently does not have.

I think this is down to the lost time and how I got stuck with certain aspects of the project but in future sprints/projects I will be able to organise my time better.

Conclusion

In conclusion I am pleased that the basic functionality for the project was achieved (CRUD) but I am not happy with the design or layout. It looks unprofessional and not user friendly at all. However with the amount that was achieved in the 5 days allotted with the setbacks that were presented, I believe a lot was achieved.

I believe the testing could have gone better as I believed that with the short time given, my time was better spent in getting the front end off of the ground rather than spending it on the testing when the application worked on the front end.