

# MMA Fighter database

Zayne Jamil

# CONCEPT

- Wanted to make an application in which fighters could be logged alongside their MMA records written in the standard format (W-L-D (NCs)).
- Wanted to be able to search for a fighter by their name.
- Wanted to to be able to compare two fighters side by side.

# SPRINT PLAN

- My sprints were mainly split up into singular sections as this helped with version control.
- Sprints were split into the following general sections.
  - Front end functionality, mainly JavaScript as HTML design implemented BootStrap which made the workload much more manageable.
  - Front end testing done through Selenium.
  - Back end functionality, using Java to perform CRUD functions on each fighter.
  - Back end testing using JUnit and Mockito.
  - User stories, required documentation and acceptance criteria were all tracked using sprints too.

# CONSULTANT JOURNEY

- Initially wanted to make multiple pages with different functions such as a comparison page and a page solely for adding a fighter into the database.

Why did I decide to make a single page application?

- Didn't take long to leave this until after the MVP had been reached since there was constraints on time.
- Decided that the same functionality to add a fighter could be done with much less effort and would consume less time.
- Comparison feature was not required in the MVP so in the end to save time it was omitted. This could be added in the future.

# CI

Continuous integration with?

# TESTING

- Testing for the back end Java application was done using JUnit and Mockito, integration tests were run for the controllers and unit tests were run for the services.
- Front end testing was done using Selenium written in Java, this tested all the features of the front end. That being the add, search and update functionalities alongside the delete and read all features of the webpage.

# DEMONSTRATION

- Read All - Add - Update - Delete

# SPRINT REVIEW

- Most tasks were completed within the allotted time. Some issues became apparent near the end of the project.
- Some tasks could have been made more specific as to delegate time for tasks more accurately.
- Bugs with Selenium testing lead to delays in the project.



# SPRINT RETROSPECTIVE

- Tasks could be made to be more specific.
- The readability of the results on the front end could be improved.
- Cleaner organisation of files so when using version control there was less confusion when changing branches and pushing files to GitHub.
- Time management could be improved to leave time for some minor fixes and documentation.

# CONCLUSION

- This project has improved my confidence with working with various different code languages, IDEs and general best practices.
- All required MVPs were met and within the time limit.
- More functionalities can be seen that would be useful to add.
- Improvements with readability on the front end are apparent.

# QUESTIONS