# UFCFTR-30-3
# Distribute & Enterprise Software Development

# Sprint Review Form

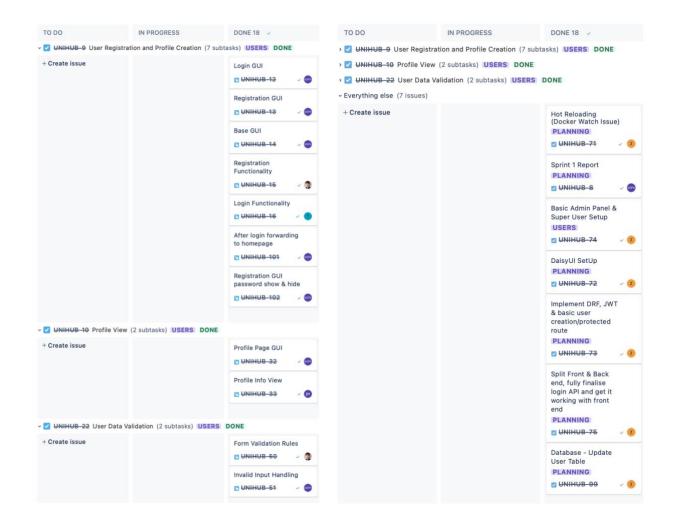| Group: | Group 21 |
|---|---|
| Sprint: | 1 |
| Members: | Sophie Fidan, James Lymbery, John Higgins, Toby Meredith, James Smith |

## Burn-down Chart



## Backlog list – Kanban Board:

**Note:** Some tasks have sub-tasks that were distributed among different group members. The 'Everything Else' section on the board reflects the tasks with no sub-tasks.

# Communication Issues

N/A

# Reflections

Group Dynamics: The team collaborated well together during the sprint, constantly communicating via a team chat. This ensured resolving problems quickly and keeping everyone in sync regarding project progress. Whenever any problem was faced, it was immediately discussed and solved, thus keeping the workflow intact.

Issues:

- Sprint 0 was created to set up Docker and Django before Sprint 1, but some issues arose during development, causing a slower burndown in the first week of the cycle. For

example, the database schema had to be optimised to make the project scalable. Some team members encountered problems with Docker Watch and the entrypoint.sh file format that had to be resolved before development could continue.

- Some backlog items were initially unclear, leading to confusion about task details. To address this, the task descriptions in future sprints were updated.
- One of the main causes of the initial slow burndown was the front-end development environment. Tailwind and Daisy UI frameworks were used in the GUI, but implementation was postponed due to an upcoming Daisy UI update. This decision proved beneficial, as the new update introduced more template components, leading to faster task completion afterwards.
- Progress was affected by the need to adjust token handling after a more effective approach was introduced in lectures. Although task completion was delayed, the adjustment ultimately resulted in a more maintainable implementation for future sprints.

Good Practices:  The task-branch workflow is used to allow parallel development. Only one team member reviewed and merged the branch into a main branch upon completion, for code quality and to avoid integration problems.

Despite the initial setback, all intended work was successfully done. The team responded effectively to changes, and our planned workflow helped enhance the efficiency of the sprints.

## Relevant Links

https://github.com/sudefidan/Docker_Thingy.git