

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.

Name ▾



James Smith



John Higgins



Sophie Fidan



Toby Meredith



James Lymbery

TO DO	IN PROGRESS	DONE 18 ✓
✓ UNIHUB-9 User Registration and Profile Creation (7 subtasks) USERS DONE + Create issue		Login GUI ✓ UNIHUB-12 ✓ USERS Registration GUI ✓ UNIHUB-13 ✓ USERS Base GUI ✓ UNIHUB-14 ✓ USERS Registration Functionality ✓ UNIHUB-15 ✓ USERS Login Functionality ✓ UNIHUB-16 ✓ USERS After login forwarding to homepage ✓ UNIHUB-101 ✓ USERS Registration GUI password show & hide ✓ UNIHUB-102 ✓ USERS
✓ UNIHUB-10 Profile View (2 subtasks) USERS DONE + Create issue		Profile Page GUI ✓ UNIHUB-22 ✓ USERS Profile Info View ✓ UNIHUB-33 ✓ USERS
✓ UNIHUB-22 User Data Validation (2 subtasks) USERS DONE + Create issue		Form Validation Rules ✓ UNIHUB-50 ✓ USERS Invalid Input Handling ✓ UNIHUB-51 ✓ USERS
TO DO	IN PROGRESS	DONE 18 ✓
✓ UNIHUB-9 User Registration and Profile Creation (7 subtasks) USERS DONE ✓ UNIHUB-10 Profile View (2 subtasks) USERS DONE ✓ UNIHUB-22 User Data Validation (2 subtasks) USERS DONE ✓ Everything else (7 issues) + Create issue		Hot Reloading (Docker Watch Issue) PLANNING ✓ UNIHUB-71 ✓ USERS Sprint 1 Report PLANNING ✓ UNIHUB-8 ✓ USERS Basic Admin Panel & Super User Setup USERS ✓ UNIHUB-74 ✓ USERS DaisyUI SetUp PLANNING ✓ UNIHUB-72 ✓ USERS Implement DRF, JWT & basic user creation/protected route PLANNING ✓ UNIHUB-73 ✓ USERS Split Front & Back end, fully finalise login API and get it working with front end PLANNING ✓ UNIHUB-75 ✓ USERS Database - Update User Table PLANNING ✓ UNIHUB-99 ✓ USERS

Access can be given to the Jira board with a request. The link:
<https://sudeff382.atlassian.net/jira/software/projects/UNIHUB/boards/5/timeline>

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 burn-up in the burndown chart during the first week of the cycle. This was because blocking issues arose during development which stopped group members from being able to work. For example, team members encountered problems with Docker due to formatting issues, such as the `entrypoint.sh` file which would incorrectly format to Windows line-endings, causing a crash on container startup. There were a few other minor issues that occurred and were added as tasks due to their blocking nature, such as ensuring hot reloads worked as expected inside the docker container to allow efficient development (no server restart required for changes).
- Some backlog items were initially unclear, leading to confusion about task details. To address this, the task descriptions in future sprints were updated.
- Apart from the aforementioned issues, another cause of burn-up was Daisy UI and Tailwind frameworks' updates that were released. The front-end implementation was postponed for this reason which also caused a slower burndown in the first week of the cycle. The release, which was luckily a week before the sprint ended. This decision proved beneficial, as the new update introduced more template components, leading to faster task completion afterwards.
- Progress was affected by switching to the Django Rest Framework, which was shown in lectures. It required the implementation to be changed but resulted in effective token handling, which was enabled by the Django Rest Framework.

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, with a very successful last week of the sprint that caused our tasks to burn very quickly. 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