UFCFTR-30-3
Distribute & Enterprise Software Development

## Sprint Review Form

Group:	19
Sprint:	1
Members:	Rohaan Aslam (21017718) Arthur Milner (21035478) William Barnes (21031340) Seif Mansour (23012749) Chaya Moore (21030599)

Burn-down Chart

Below is the burn-down chart for sprint 1, unfortunately, we realised during the sprint that additional tasks were required, hence the fact it is going up and down. Going into sprint 2 we want to be sure our sprint backlog is comprehensive enough for the requirements to avoid this problem in the future.

The reason for the discrepancy in the burndown chart was because of a mid-sprint decision to change our authentication method to use JWTs instead of built-in django auth. This was after attending the lecture and seeing and promoted the use of JWT's hence we decided to implement it into our solution. However, this meant that rather than adding tasks to the backlog we instead just added them straight into sprint 1 prioritizing immediate implementation whilst this did allow us to set up authentication earlier it also massively increased the workload mid-sprint rather than deferring it into sprint 2.

Corrected Burndown chart
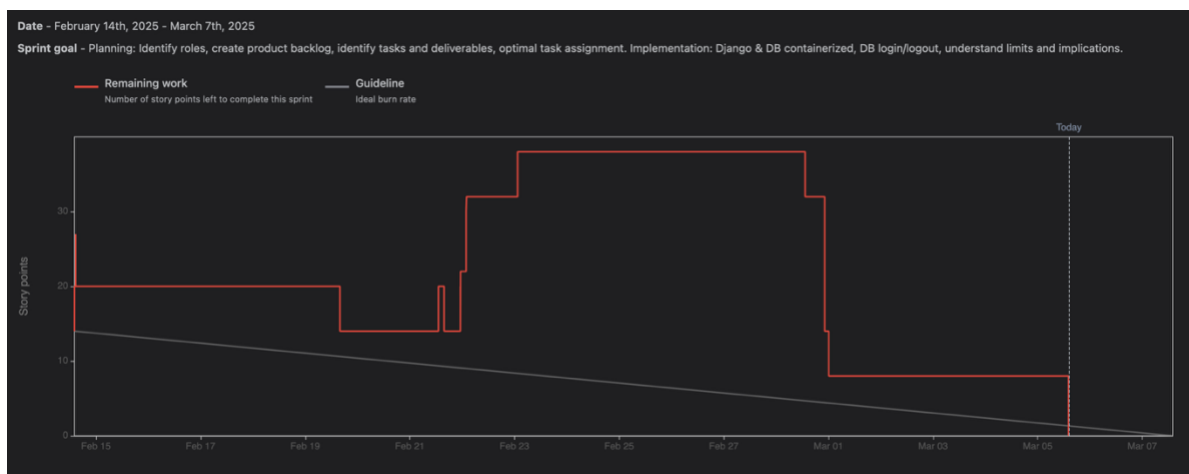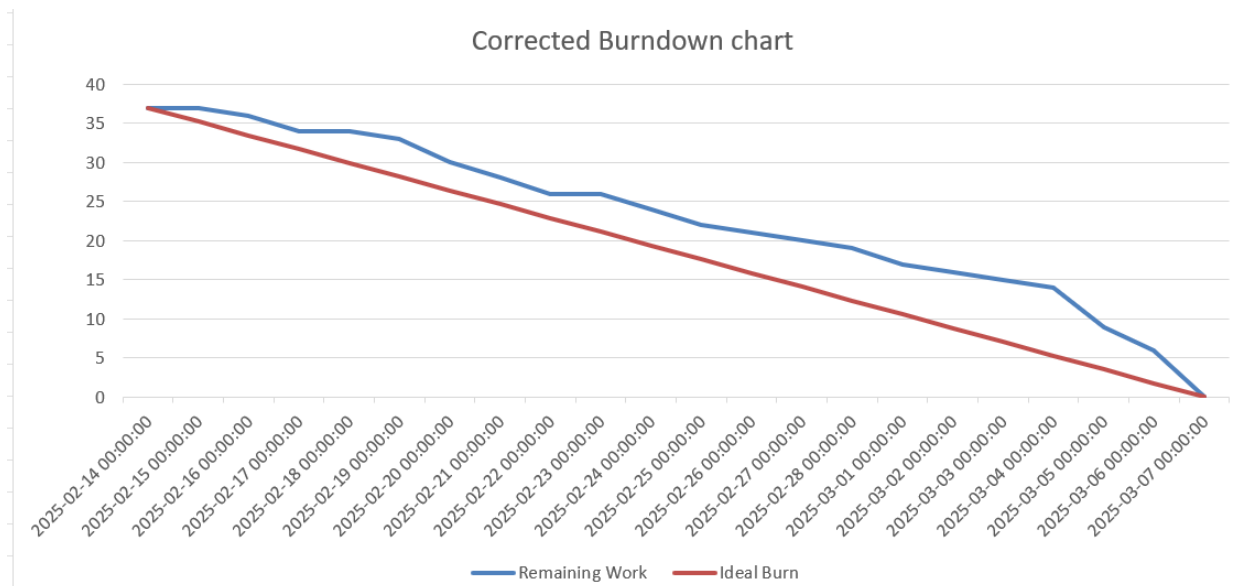
The corrected burndown chart above is what our burndown would have looked like if the extra work was pushed to the backlog instead of being added directly into the sprint this approach would have resulted in a more predictable and stable burndown.

Project Schedule and Backlog

Our backlog consists of epics, which contain user stories, which then contain multiple sub-tasks which are assigned to a single group member.

Sprint 1 Backlog:

Below is the backlog for Sprint 1, an expanded view of the backlog, including the user stories with their sub-tasks, can be seen in the snapshots for the Sprint 1 kanban board.

Sprint 1 Kanban Board:

Below is the kanban board for sprint 1, the assignee for each subtask can be seen in the avatar, with the letters representing the group member's initials. The purple text represents the epic, the icon represents a user story, whilst the icon represents a subtask.

TO DO	IN PROGRESS	DONE 47 ✓

˅ 🔖 S̶C̶R̶U̶M̶-̶4̶4̶ As a developer, I want to have a database designed and i...   (3 subtasks) **DATABASE IMPLEMENTATION** **DONE**

+ Create issue

> As a developer, I want to have a database designed and implemented for data persistence on Uni Hub.

Create the database schema.

🔗 S̶C̶R̶U̶M̶-̶1̶3̶3̶   ✓   WB

Implement database indexing and optimization.   •••

🔗 S̶C̶R̶U̶M̶-̶1̶3̶4̶   ✓   WB

Insert sample data into the tables.

🔗 S̶C̶R̶U̶M̶-̶1̶3̶5̶   ✓   WB

TO DO	IN PROGRESS	DONE 47 ✓

⌄ 🔖 ~~SCRUM-24~~ As a developer, I want to containerize Django, SMTP server and M...   (6 subtasks)   **CONTAINERIZATION**   `DONE`

+ Create issue

> As a developer, I want to containerize Django, SMTP server and MySQL, to create a consistent environment across the entire development of Uni Hub.

Create Django Project

🔗 ~~SCRUM-99~~   ✓   **AM**

Create a DockerFile for Django

🔗 ~~SCRUM-69~~   ✓ `5` **AM**

Configure MySQL in docker-compose file   •••

🔗 ~~SCRUM-70~~   ✓ `5` **AM**

Configure Django in docker-compose file

🔗 ~~SCRUM-71~~   ✓ `5` **R**

Ensure Django and MySQL containers are communicating

🔗 ~~SCRUM-72~~   ✓ `5` **R**

Configure mailhog SMTP in docker-compose file

🔗 ~~SCRUM-113~~   ✓ **AM**

TO DO	IN PROGRESS	DONE 47 ✓

∨ 🔖 SCRUM-68 As a developer, I want react to be containerized, to create a consis... (6 subtasks) CONTAINERIZATION DONE

+ Create issue

As a developer, I want react to be containerized, to create a consistent environment and promote separation between frontend and backend.

Create a React + Vite project

🔲 SCRUM-86 ✓ R

Install necessary dependencies

🔲 SCRUM-87 ✓ R

Create a Dockerfile for react application

🔲 SCRUM-88 ✓ R

Write docker-compose yml file for react setup

🔲 SCRUM-89 ✓ R

Configure polling in Vite for hot reload for Docker

🔲 SCRUM-90 ✓ R

Test communication between containerised React and Django

🔲 SCRUM-91 ✓ R

TO DO	IN PROGRESS	DONE 47 ✓

˅ 🔖 ~~SCRUM-92~~ As a user, I want to be able to log in seamlessly and a... (12 subtasks)   **STUDENT ACCOUNT PROFILE A...** `DONE`

＋ Create issue

> As a user, I want to be able to log in
> seamlessly and access my role-specific
> features, so that I can efficiently
> interact with the platform based on my
> permissions

DONE column cards:

Set up AuthContext to manage
authentication state   •••

⬚ ~~SCRUM-93~~   ✓   Ⓡ

Attach JWT to all API requests
for authentication

⬚ ~~SCRUM-94~~   ✓   Ⓡ

Create LoginPage with a form
to handle user login

⬚ ~~SCRUM-95~~   ✓   Ⓡ


Implement RBAC - public,
private and role-based routing
wrapper components

⬚ ~~SCRUM-96~~   ✓   Ⓡ

Ensure user session
persistance across page hard
reloads

⬚ ~~SCRUM-97~~   ✓   Ⓡ

Ensure unauthorised requests
are retried with fresh access
token if possible

⬚ ~~SCRUM-103~~   ✓   Ⓡ

Style LoginPage

🔲 SCRUM-98   ✓ (R)

Create custom permission
classes and views to test them

🔲 SCRUM-119   ✓ (AM)

Create custom Djoser-JWT
serializers to appropriately
handle JWT token

🔲 SCRUM-120   ✓ (AM)

Create custom Djoser-JWT
views to appropriately handle
JWT/auth

🔲 SCRUM-121   ✓ (AM)

Create & Style NavMenu
Sidebar Component

🔲 SCRUM-207   ✓ (R)

Test login page

🔲 SCRUM-209   ✓ (R)

✓ 🔖 SCRUM-75 As a user, I want to register for an account using my p... (13 subtasks) STUDENT ACCOUNT PROFILE A... DONE

+ Create issue

As a user, I want to register for an account using my personal and academic details and verify my email with all expected registration features, giving me access to the Uni Hub community.

Configure Mailhog SMTP email server with Django

SCRUM-76   ✓ AM

Configure Djoser/Django email activation settings

SCRUM-77   ✓ AM

Test Django Mailhog functionality

SCRUM-106   ✓ AM

Create & Style RegisterPage with a validated form for user registration

SCRUM-102   ✓ R

Add custom email activation template for React to handle

SCRUM-104   ✓ R

Create AccountActivation page and implement functionality

SCRUM-105   ✓ R

Test Account Activation works

⌗ ~~SCRUM-125~~ ✓ (R)

Configure Djoser/Django password reset email

⌗ ~~SCRUM-107~~ ✓ (AM)

Add custom password reset template for React to handle

⌗ ~~SCRUM-108~~ ✓ (AM)

Create PasswordReset page from which users can request a reset password link

⌗ ~~SCRUM-110~~ ✓ (AM)

Configure reset password link with React to allow for users to input their new password   ...

⌗ ~~SCRUM-111~~ ✓ (AM)

Test password reset is working

⌗ ~~SCRUM-112~~ ✓ (AM)

Test Registration Page

⌗ ~~SCRUM-210~~ ✓ (R)

Product Backlog:

Here is the entire product backlog which will be split between Sprints 2 and 3. All stories in the backlog have 1-5 subtasks attached.

Entire Project Schedule:

Defining the project schedule involved adding start and end dates to all user stories and epics. Below is a simplified Gantt chart, followed by the schedule as it appears in Jira.

Final Gantt Chart with Exact Task Dates

	February	March	April	May
Sprints	Sprint 1	Sprint 2	Sprint 3	

- ⚡ SCRUM-52 Initial Planning
  - 🔖 SCRUM-53 As a developer, I want to have a compr... `DONE`
- ⚡ SCRUM-28 Containerization `DONE`
  - 🔖 SCRUM-24 As a developer, I want to containerize ... `DONE`
  - 🔖 SCRUM-68 As a developer, I want react to be cont... `DONE`
- ⚡ SCRUM-49 Database Implementation `DONE`
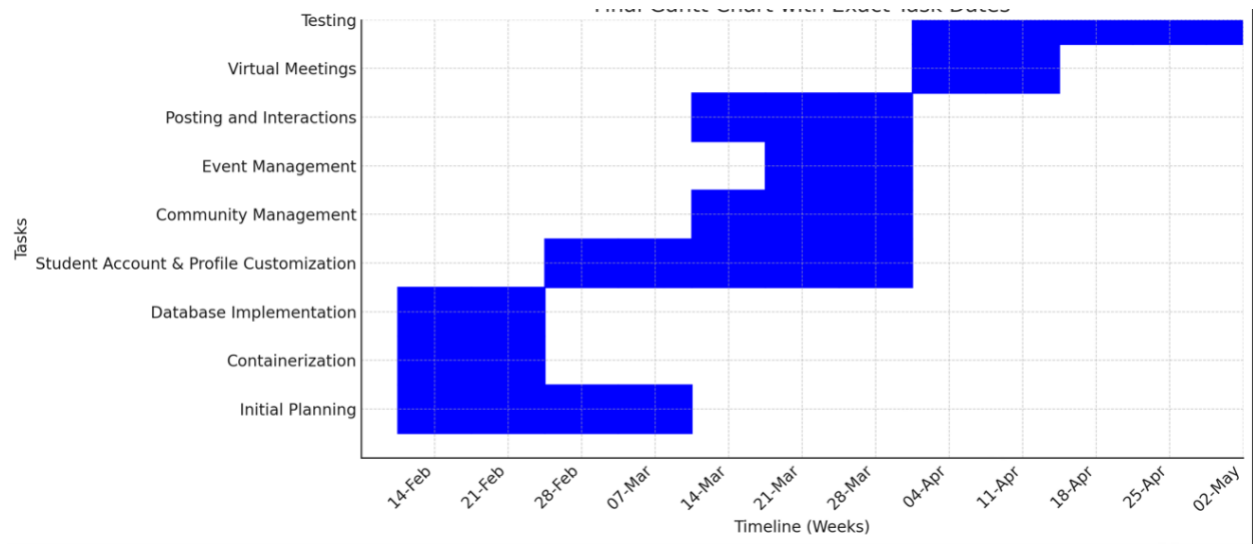 - 🔖 SCRUM-44 As a developer, I want to have a datab... `DONE`
- ⚡ SCRUM-6 Student Account Profile and Customisation
  - 🔖 SCRUM-92 As a user, I want to be able to log in se... `DONE`
  - 🔖 SCRUM-75 As a user, I want to register for an acc... `DONE`
  - 🔖 SCRUM-10 As a student, I want to be able to upda... `DONE`
  - 🔖 SCRUM-67 As a student, I want to be recommend... `TO DO`
  - 🔖 SCRUM-127 As a student, I want to add media (ph... `TO DO`
  - 🔖 SCRUM-128 As a student, I want to customize my... `TO DO`
  - 🔖 SCRUM-198 As a student, I want to filter commun... `TO DO`
  - 🔖 SCRUM-199 As a student, I want to showcase my ... `TO DO`
  - 🔖 SCRUM-200 As a student, I want to receive event ... `TO DO`

	February	March	April	May
Sprints	Sprint 1	Sprint 2	Sprint 3	

- ⚡ SCRUM-14 Community Management
  - 🔖 SCRUM-62 As a student, I want to be able to crea... `TO DO`
  - 🔖 SCRUM-63 As a community leader, I want to be a... `TO DO`
  - 🔖 SCRUM-129 As a community leader, I want to sen... `TO DO`
  - 🔖 SCRUM-195 As a community leader, I want moder... `TO DO`
  - 🔖 SCRUM-216 As a community leader, I do not want... `TO DO`
- ⚡ SCRUM-15 Event Management
  - 🔖 SCRUM-64 As an event manager, I want to be abl... `TO DO`
  - 🔖 SCRUM-130 As an event manager, I want to see a... `TO DO`
- ⚡ SCRUM-16 Posting and Interactions
  - 🔖 SCRUM-131 As a student, I want to receive notific... `TO DO`
  - 🔖 SCRUM-132 As a community leader, I want to pin i... `TO DO`
  - 🔖 SCRUM-189 As a student, I want to control who c... `TO DO`
  - 🔖 SCRUM-192 As a student, I want to search and filt... `TO DO`
  - 🔖 SCRUM-217 As a student, I want to be able to cre... `TO DO`
  - 🔖 SCRUM-222 As a student, I want to be able to foll... `TO DO`
- ⚡ SCRUM-66 Virtual Meetings
  - 🔖 SCRUM-65 As a community leader, I want to be a... `TO DO`
- ⚡ SCRUM-226 Testing
  - 🔖 SCRUM-227 As a developer, I want a list of test c... `TO DO`
  - 🔖 SCRUM-230 As a developer, I want Uni Hub to be ... `TO DO`

Today Weeks **Months**

Communication Issues

During sprint 2 we encountered a couple of communication challenges this mostly stemmed from difficulties in some members attending practical sessions. With some team members arriving late or did not attend at all, which had an impact on our collaboration. To mitigate this, we relied on Discord to help maintain communication throughout the sprint. While this helped to some extent, we as a group have recognised the need to improve this and to implement a better strategy in the next sprint to ensure we have more effective communication and participation within the group.

Reflections

As a group we effectively setup and utilised Jira for project management, using features such as, a generated kanban board from the current sprint, story points for task estimation, being able to assign tasks to members, and cloud collaboration. Member roles were split between focusing solely on the backlog, and focusing on developing the project alongside the backlog. This resulted in consistent development and a comprehensive backlog.

Pair programming was used for the bigger tasks, such as containerization and registration/user authentication, this helped tackle the new concepts in Django Rest Framework (DRF) and using JSON Web Tokens (JWTs).

We have containerised React, Django, MySQL, and MailHog for a complete login/sign-up authentication with email activation and email validated password reset. Account details can also be edited, and user profiles can be viewed. We have also setup the initial permissions for standard students/users, event managers, and community leaders as well as designed the Database.

The project's initial setup is fully complete, appropriate authorization is in place, and profile features are comprehensive. We have also created the entire project schedule and backlog to a good standard. The project is in a good place to begin SPRINT 2 tasks, such as functionalities regarding posts, communities, and events.

For sprint two we will look to spread out the work more evenly amongst all of the different group members to avoid over-reliance on just a few individuals in the group. During sprint 1 most of our front-end and back-end development was done by only two team members this was primarily down to poor communication as mentioned in the communication section and a group member experiencing technical difficulties with a laptop.

Steps taken to improve for the next sprint:
1. **Better Task Delegation –** We will ensure that team members have roles that are more clearly defined and responsibilities at the start of the sprint by making better use of Jira to assign user tasks more clearly.

2. **Improved Communication –** While we used discord for communication in sprint 1 we recognise the need for more structured updates, we will do this by scheduling regularly check ins throughout the week to discuss progress address any challenges and to

provide support to team members that are struggling with assigned work before it before a problem at the near end of the sprint.

3. **Contingency planning** – To ensure we prevent disruptions like the one we faced with hardware failure we will make better use of university computers and cloud-based development environments to ensure that technical difficulties do not hinder any progress.