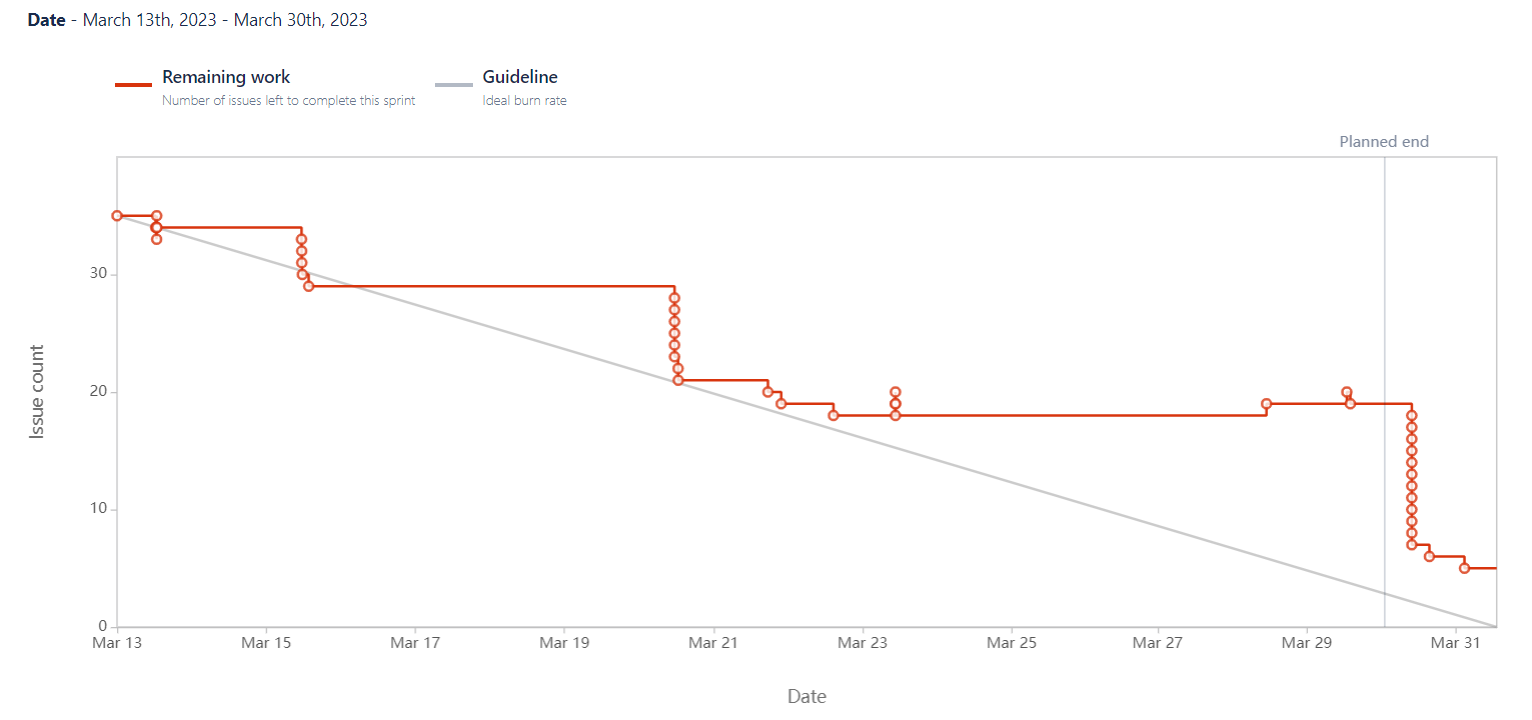
# **UFCFTR-30-3/UFCF85-30-3**

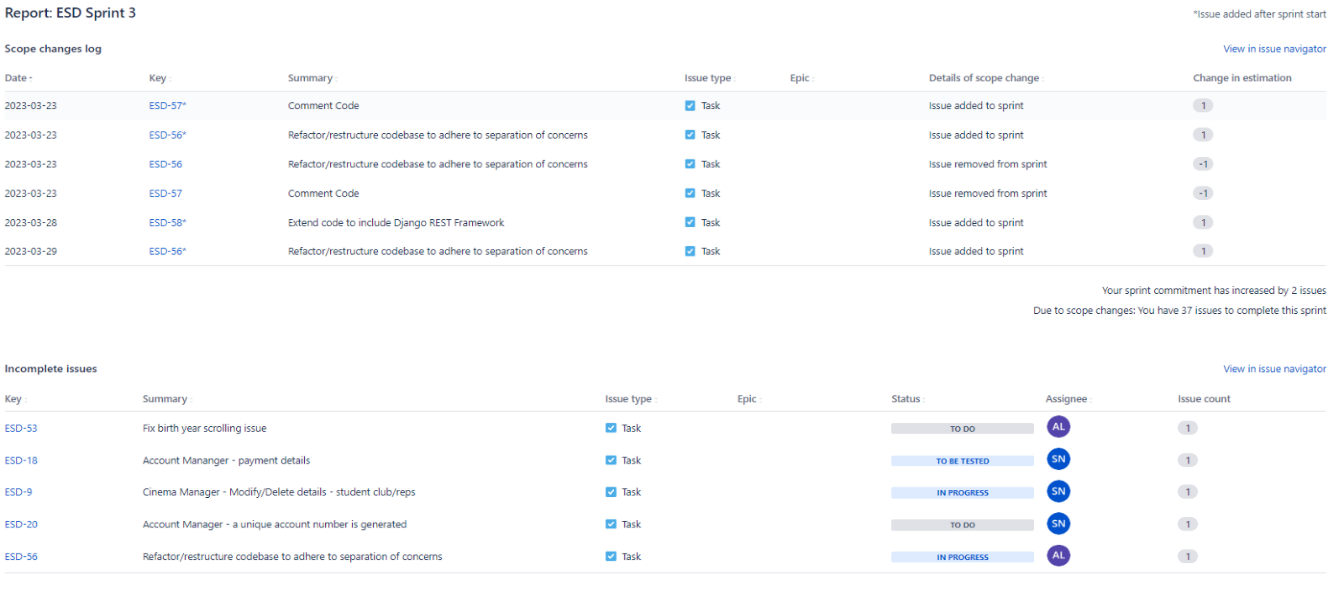
# **Distribute & Enterprise Software Development**

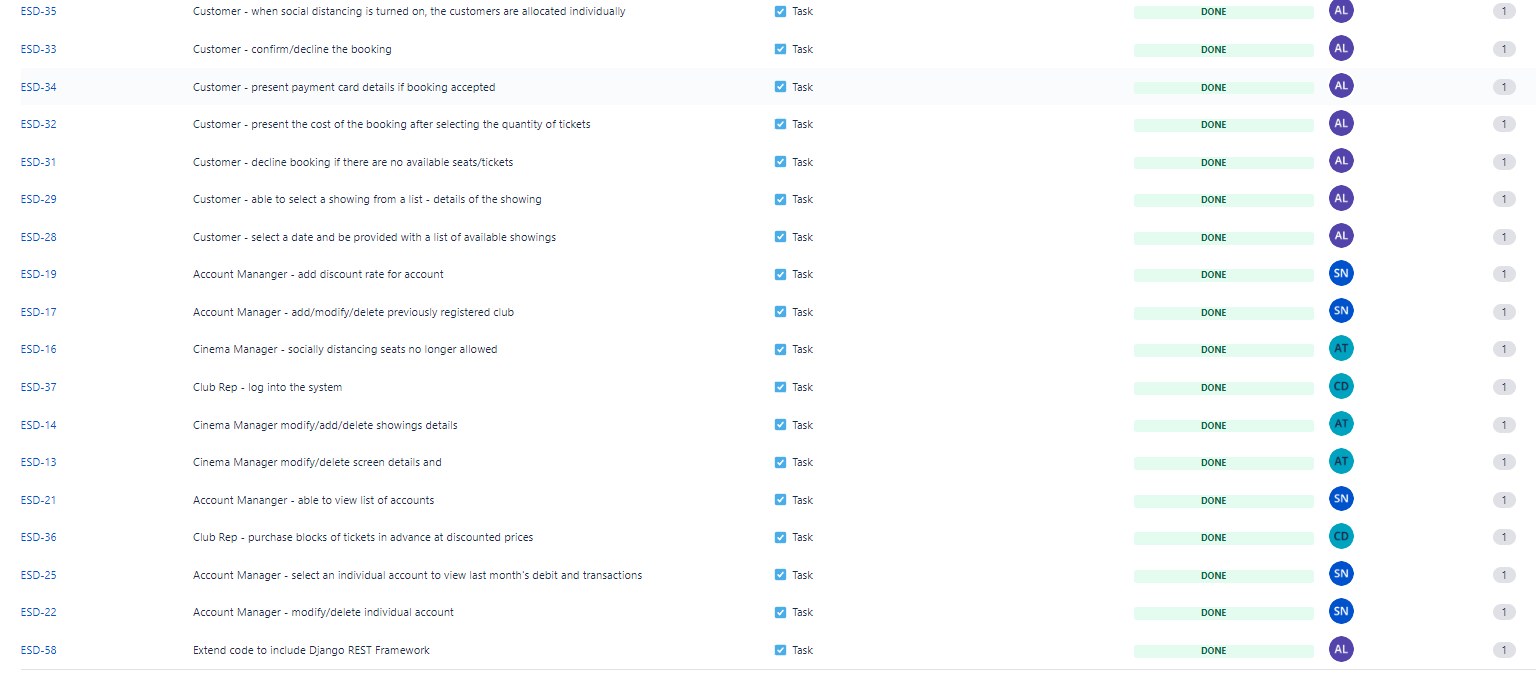
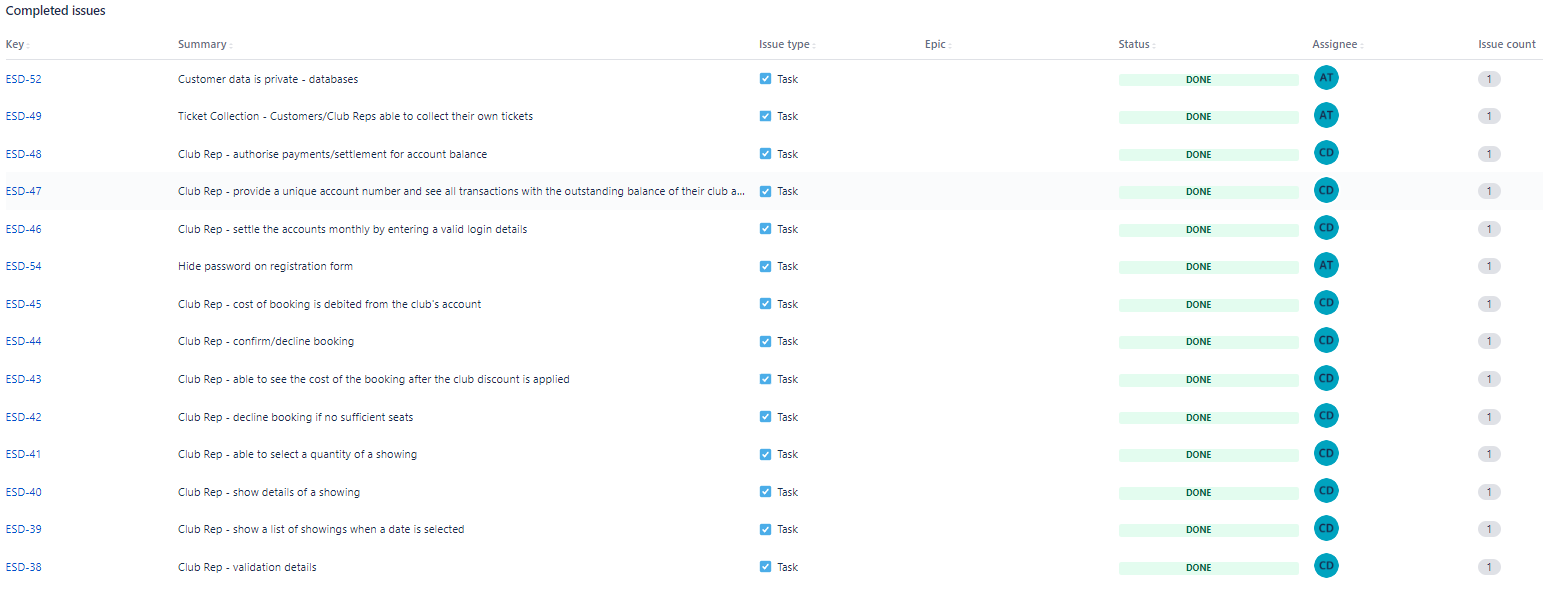
# **Sprint Review Form**

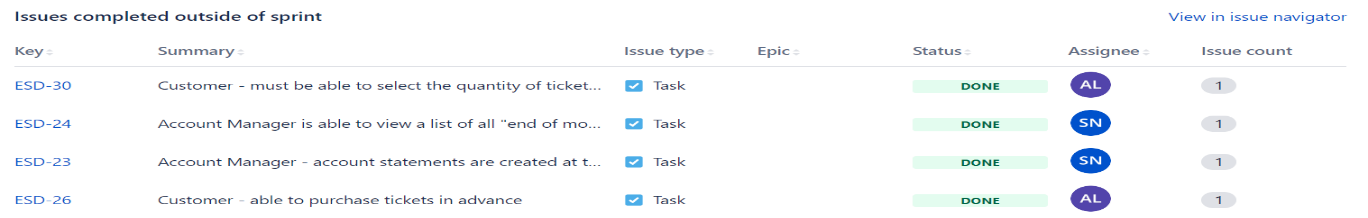
|  |  |
| --- | --- |
| **Group:** | 2 |
| **Sprint:** | **Sprint 3** – Weeks 7 – 10 (13.03 – 31.03.2023) |
| **Members:** | Andrei Timo, Alfie Lamerton, Shreevatsa Nayak, Christian Dawson, Sebastian Annett |

**<<Burn-down Chart>>** 

**<<Backlog list>>**







**<<Communication Issues>>**

We had no communication problems between the group members - Sebastian managed to keep in touch with us through MS Teams since the beginning of March (10-11/03/2023). Andrei did not manage to attend the tutorials due to health problems and due to the strikes with GWR, but all members of the group managed to communicate and do a weekly stand-up (each Wednesday), before the tutorial on Thursday. We communicated over our teams chat, over teams calls and through Github pull requests.

**<<Reflections>>**

All the functionalities that we planned to implement on **Sprint 3** have been implemented. We focused on making sure the core functionalities of the app work optimally during this sprint and will spend sprint 4 fine tuning those functionalities.

At the beginning of the sprint, we allocated our roles based on our user type. We believe this is the most optimal method of allocating the roles because it lets each team member work on the role that they have the most experience with. We have done this since the beginning of the development process; It has helped us define our tasks and understand how they fit into the app as a whole. Everyone has been clear on their role throughout the sprint and completed the necessary relevant tasks expected of them.

**Alfie** has been developing the customer user type throughout the development process. In **Sprint 3** he finished implementing all the tasks satisfying the customer user type and added Django REST Framework views and serializers for connecting to the external services that we have laid out for sprint 4.

The customer requirements Alfie implemented can be seen on the Jira board and closed Github pull requests. Alfie added unit testing reports for all the customer functionalities and tested the API endpoints using insomnia.

**Christian** has been diligently working on the development of the club representative user type since the initiation of the first Sprint. Throughout this process, he has been focusing on understanding and refining the various requirements necessary to provide a seamless experience for this user type. In Sprint 3, Christian conducted a thorough analysis to identify the specific requirements needed to cater to the club representative user type.

After identifying these requirements, Christian meticulously created individual tickets on a Jira kanban board to effectively manage the tasks involved in addressing each requirement. This approach allowed him to efficiently track progress and ensure that all necessary aspects were taken into consideration during the development process. As a result, Christian was able to successfully complete every ticket in a timely manner.

Christian also helped the account manager by creating the discount page used to apply discounts to selected club representatives. For sprint 4, Christian will be implementing the stripe API, writing unit tests for his user type, and documenting his test cases on the group confluence page.

**Shreevatsa** has been developing the account manager user type throughout the process. In **Sprint 3**, Shreevatsa has been responsible for implementing the functionalities for the CRUD system for both the accounts and the administrators, ensuring that all data can be easily created, read, updated, and deleted. Additionally, with the CRUD tables, he has implemented a robust exception handling which ensures that any invalid inputs are promptly flagged as errors, preventing potential errors that may occur during the registration of accounts.

Furthermore, during sprint 3, Shreevatsa was able to create the functionalities for the account statements using a monthly and annual transaction process. This has enabled Shreevatsa to fetch data and containerise the transaction receipt with user information, making it easier to generate transactions.

After the functionalities were implemented, Shreevatsa has created multiple test cases for the Account Manager, for him to make sure that all the functionalities are correctly working and is robust. The test cases were created in Jira which contains multiple user types and their test cases.

**Andrei** has been developing the **Cinema Manager** user type since the **1st Sprint**. In **Sprint 3** he finished implementing the tasks for this user type, and added the Django Framework views and models. After all functionalities were successfully implemented, Andrei created the necessary test cases for each of the requirements for the Cinema Manager user type. During the process of implementing, some bugs have been reported by **Sebastian –** editing an existing showing – when clicking on the “Edit” button, a random ID value was taken from the database and all information for that showing were displaying for the user. Andrei also created the report for Unit Testing using a Confluence template in which each member of the group can keep track of what was implemented, if it works properly or needs further reviews. Because Sebastian managed to establish communication with us, we all agreed to let him familiarize himself with the code and report and fix the bugs encountered.

During Sprint 3, our team conducted a comprehensive analysis of the project's progress and identified the following tasks to be addressed in the final sprint, Sprint 4:

* Refactoring models to use a unique identifier to improve security: By implementing unique identifiers for each model, we aim to enhance the overall security and data integrity of the system.
* Adding login authentication using Google OAuth2: Incorporating Google OAuth2 for user authentication will streamline the login process and provide a secure and convenient way for users to access the platform.
* Adding payment authentication using the Stripe API: Integrating the Stripe API for payment authentication will enable secure and efficient processing of transactions for users and provide a reliable payment infrastructure.
* Commenting code that is not self-explanatory: To improve code readability and maintainability, we will add comments to any sections of the codebase that may be difficult to understand or require additional context.
* Refactoring the application architecture into sub-apps in a way that adheres to the separation of concerns: By restructuring the application into modular sub-apps, we will promote a cleaner architecture that adheres to the separation of concerns principle, making it easier to manage and extend the project in the future.

With these tasks added to the backlog, our team is confident that there are no outstanding tasks left unaddressed. We strongly believe that we will successfully complete the project to the level we have outlined for Sprint 4. By prioritizing and efficiently tackling these tasks, we anticipate delivering a high-quality, robust, and user-friendly solution that meets the project's objectives and requirements.

**<<Relevant Links>>**

GitHub Link: <https://github.com/andrei2timo/ESD-UWEFlix-Cinema-Component-B>

Jira Board: <https://uweflix.atlassian.net/jira/software/projects/ESD/boards/1>

Confluence: <https://uweflix.atlassian.net/wiki/spaces/IS/pages/1146888/UWEFlix+Cinema+-+Unit+Testing+Report>