University of Kent

CO553 Sprint One Report

A report consisting of our sprint review and retrospective

Group C2-3

Agile Development and Software Security B (CO553)

CO553 Scrum Project Report

The aim of our project is to create a dynamic cinema website that can be used to book tickets for films. The website will retrieve and display data from a collection of relations we have in our database. Additionally, it uses user input to write to some of the relation (e.g. register and time/ticket selection page).

Instead of assigning the traditional roles of product owner, scrum master, team member 1 and team member 2, we decided to divide our team into a scrum division and a coding division. Two members would handle the creation of product and sprint backlogs after brainstorming with the team, while the other two members would work on the code for the website.

We believed that by working in pairs, members can bounce ideas of one another and assist each other with issues. Although we made sure to be flexible with our work and assist the other division if they found something challenging.

Creating the Product and Sprint Backlog

In our first scrum meeting we established that the product and backlog were a priority and so decided to start work on that immediately. In order to do this, we brainstormed several user stories, we set a target of at least 20 user stories. We made sure to break down our epic stories into multiple smaller user stories.

Collectively as a team we decided to go for the “1,2,3,5,8,10,13,20,40” point system as we felt that it would allow us to differentiate between the difficulty of the stories. Once we finished assigning points, we organised the product backlog taking into consideration how challenging it would be and how important it would be to the project.

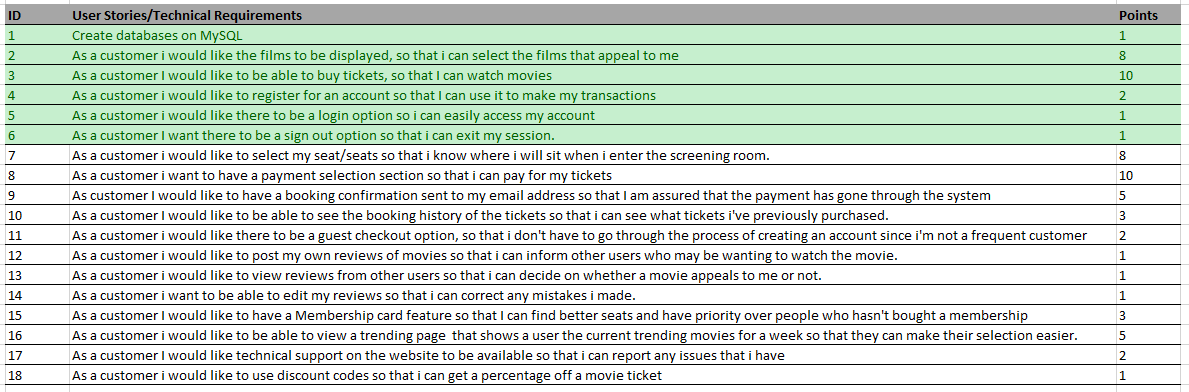


Figure 1: Product backlog

After completing our product backlog, we entered into a discussion about our team velocity. Taking into consideration our current workload and the possibility of it increasing due to other assignments, we decided on a range between 23-25 points per sprint. We then took a collection of user stories who’s point equated to 23, which happened be the first 6 user stories in our product backlog and moved them into our first sprint backlog.

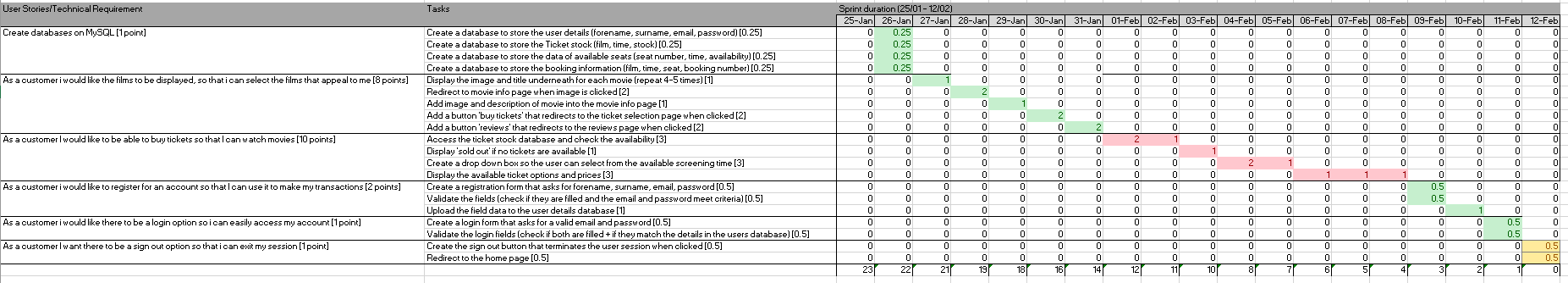


Figure 2: Sprint backlog 1

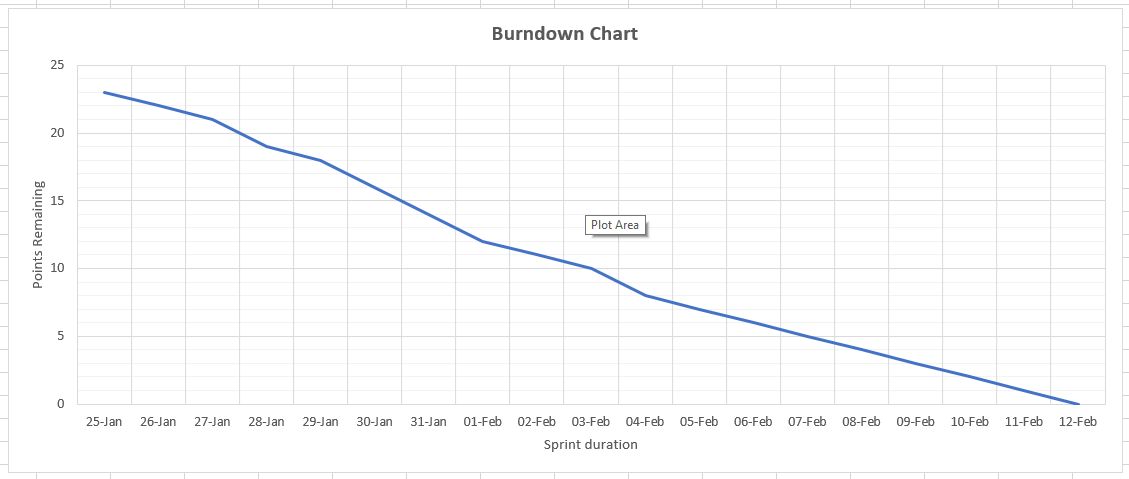


Figure 3: Burnout diagram

In the sprint backlog, we broke each user story into smaller manageable tasks and assigned points to each task that would total to the point score of its corresponding story. These tasks were spread throughout the 19-day sprint. This was document on a spreadsheet we made that was inspired by the template on Moodle. The table was designed to display the remaining points after each day. These values were then used to plot a burnout diagram.

Sprint Review

In terms of our first sprint, our group ensured that a plan was created to assign tasks to members within the group. Once everyone was clear on what tasks to complete, we worked in pairs and planned to arrange multiple meetings throughout the week to review what work had been completed and to allow each member to discuss any difficulties that they may have faced.

As a group, we have managed to achieve the creation of a MySQL database to store the movie booking website information. Once this was created, the following information was stored: user details, ticket stock, availability of seats within the cinema and booking information.

Following the creation of the database, the next stage of the plan was to begin creating our webpage. We started by creating a homepage which would be linked to each tab of the cinema page. Secondly, we created a page which showed the currently airing movies (with a picture). On click, each movie is accompanied by a brief description and a review which is currently in development. There is also a button for the user to add their reviews, but it’s yet to be given functionality as it’s a task that will be in our second sprint. Moreover, we added a ‘Buy Tickets’ button that redirects users to a ticket selection page which we will increment in our next sprint.

Furthermore, our next part of the sprint that we had planned was to create a registration page for users to input their first name, surname, email, and password. This form would also validate the fields and as a result, user information would be stored in the database.

Referring to our previous point regarding our homepage we decided collectively that a login option on the homepage would provide ease to users who already have an account. The login forms similarly to the registration page contained a form that asks users for their username and password which are validated and checked if they match the credentials stored in our database.

As a group we faced a few challenges during our first sprint. The first challenge we faced was setting up the database. Since we all needed to edit the database, we would need access to one that was not associated to any individual in our group. We overcame this challenge by contacting cs-syshelp@kent.ac.uk and requesting a database, which we were soon given access to.

Another challenge we faced was migrating the code over to the CodeIgniter framework. Due to the difference in structure and syntax between CodeIgniter and vanilla html, we had to make sufficient changes to our code to fit the MVC structure. To overcome this challenge, we had our coders take a few days to move the code over and adjust it to fit the MVC structure.

The next challenge we faced was configuring the base\_url for out CI website. Because we did not have a folder in a domain to store our work so that it could be accessed by everyone in our group, we had to adjust our approach. To resolve this issue, we communicated with the staff running the module and requested a location to save our work to and to set as our base\_url. Since it would take some time for them to grant such a request, we decided to temporarily set the base\_url to one of our private repositories on the domain and have that member run it off there to test everything.

Sprint Retrospective

In terms of group communication, we also set out to meet at least three times a week at a pre-arranged time that we were all comfortable with. In doing so we were able to discuss work in progress and plan when ticking off our user stories throughout the first sprint.

In our next sprint we will make sure to utilise our groups text chat to update the other group members on incremental changes we have made each, this would be treated as a daily stand-up. In our next sprint we will make use of git branching which we were taught in CO552, this way we can avoid any redundancies in our code.

Moreover, some things that we will continue to do as a group will be to meet more frequently as we have found this effective in keeping up to date with our sprint progression and being able to share ideas on our webpage. We also concluded that by continuing our weekly meetings it will ensure that all members of the group have sufficient work to show and discuss any difficulties occurred, allowing us to work together to fix potential issues.