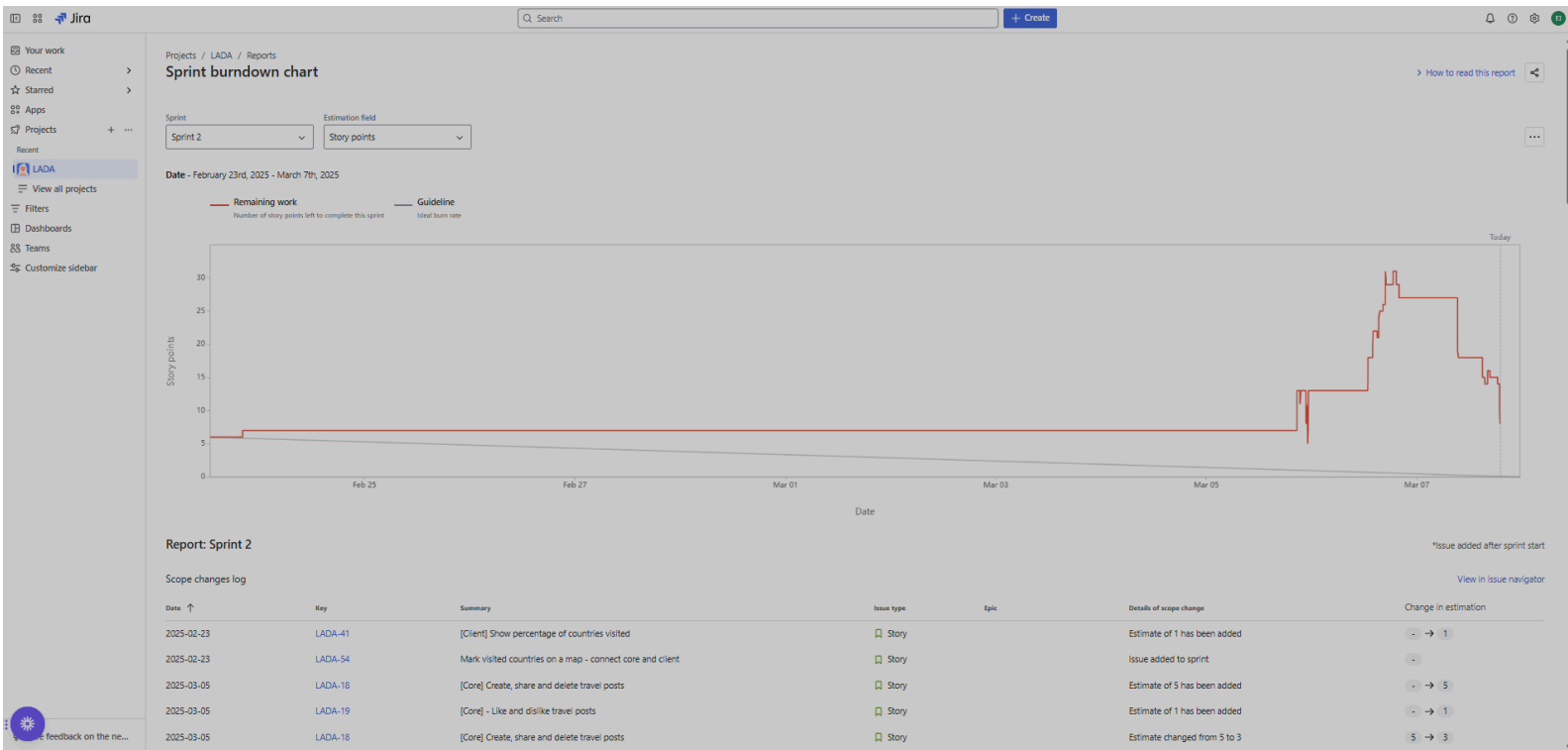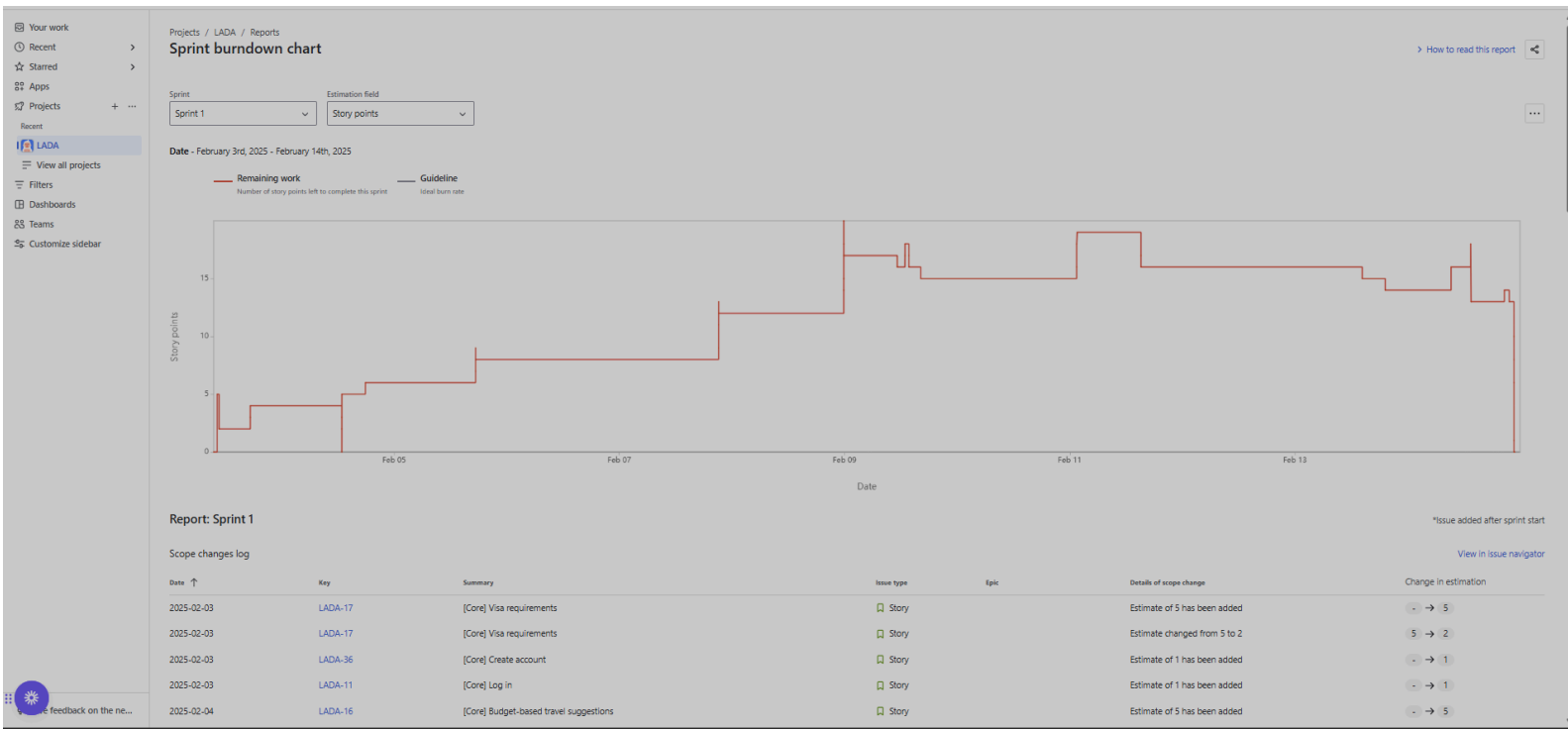Sprint 2 chart:

Sprint 1 chart:

Comments/Analysis:

Looking at the burndown charts for the two sprints, we can see that both sprints experienced tasks being added throughout the duration of the sprint. This likely came from the fact that we originally created stories without splitting them into Core and Client components since we just wanted to see what work needs to be done in general. However, while developers were working through the sprint and taking on tickets, they would split up the tickets into the frontend and backend components for clarity and to accurately divide responsibilities. Also, in Sprint 1, we can see that our work was spread out more

evenly throughout the sprint. In contrast, Sprint 2 saw most of the work being completed toward the end since all team members were pretty busy with other courses and midterms at the beginning of the sprint. Despite this, we managed to complete more work later in the sprint, which led to an overall higher velocity for Sprint 2. In terms of velocity, during Sprint 1 our team completed 25 story points, while in Sprint 2 we managed to complete 37 story points. The improved velocity in Sprint 2 reflects several positive changes in our approach. First, we became more accurate in estimating how long tasks would take, which allowed us to plan more effectively and complete a greater number of story points. Additionally, in Sprint 2, we made it a standard practice to create Jira tasks before working on something, unlike in Sprint 1 where tasks were sometimes done without creating corresponding stories. This extra step helped us keep better track of the work and align our efforts more clearly with our sprint goals. Another factor contributing to this difference in velocities between sprints is that we deliberately aimed to increase our velocity in Sprint 2 to see how much we could accomplish under a more focused effort. This helped us push our limits and understand our capacity better for future sprints. The increased velocity not only reflects our ability to work faster but also ensures that more work is completed, leaving fewer outstanding tasks for future sprints and giving us more time to do various bug fixes and enhancements.