## Course Matrix/term-group-project-c01w25-project-course-matrix

### User Stories and Their Dependencies

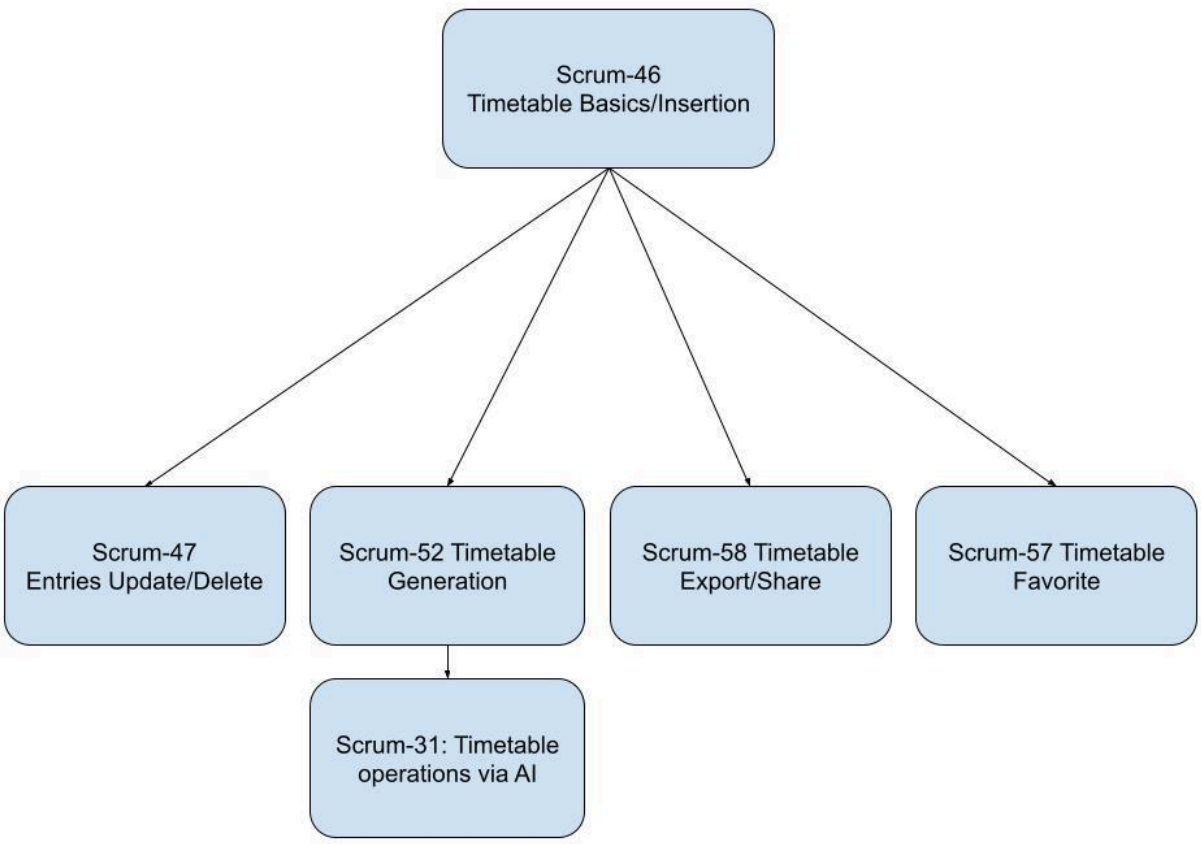
In Sprint 3 many of our user stories had dependencies that needed to be completed first before being completed themselves. The following is a list of all user stories that have dependencies that first need to be completed.

- Scrum-52 Timetable Generation (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-52)
  Dependency: (Start to Start) Scrum-46 Timetable Basics/Insertion (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-46)
- Scrum-47 Entries Update/Delete (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-47)
  Dependency: (Start to Start) Scrum-46 Timetable Basics/Insertion (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-46)
- Scrum-58 Timetable Export/Share (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-58)
  Dependency: (Start to Start) Scrum-46 Timetable Basics/Insertion (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-46)
- Scrum-57 Timetable Favourite (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-57)
  Dependency: (Start to Start) Scrum-46 Timetable Basics/Insertion (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-46)
- Scrum-31 Timetable Operations via AI (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-31)
  Dependency: (Start to Start) Scrum-52 Timetable Generation (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-52)

From this, we can gather that the critical points in our Sprint were:

1. Scrum-46 Timetable Basics/Insertion (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-46)
2. Scrum-52 Timetable Generation (https://cscc01-course-matrix.atlassian.net/browse/SCRUM-52)

### Network Diagram




## Keeping Schedule

The project remained on schedule due to the limited number of critical points during this sprint. The only major critical point was Scrum-46, a user story that had already been partially completed by the end of sprint 2. As a result, nearly all planned features were developed and completed on time.

## What Went Wrong

Sprint 3 presented a greater time constraint compared to Sprint 2 as it was limited to only 2 weeks. Within this period, the team had to develop most of the remaining features, address previous bugs and conduct testing for the final product. While most features were completed within the timeframe, the team faced challenges in testing and bug-fixing. Several bug fixes required more effort than anticipated, with some consuming nearly as much time as feature development.

Additionally, midterm scheduling extended into the first week of Sprint 3, further reducing the time available for project work. Merging features from different branches introduces unforeseen complexities, resulting in new bugs that demand extensive time and effort to resolve. Consequently, despite best efforts, the team experiences minor testing schedule deviations.

**Learnings**

The primary takeaway from Sprint 3 is the importance of accurately estimating available time and effort. The team initially set expectations higher than feasible, assuming Sprint 3 would yield similar results to Sprint 2 despite having nearly half the available time. Although all major features were completed, testing and quality assurance suffered due to time limitations.

Moving forward, Sprint 4 will focus on refining the codebase, thoroughly testing features, and resolving existing bugs rather than implementing new functionalities. This approach will ensure a polished final product with enhanced stability and performance.