CIS 4250 – Software Design V
Instructor: Prof. S. Scott
Individual Accountability Report (IAR) Template

Note. Refer to the Project Manual for detailed instructions for IAR submissions.

Individual Accountability Report (IAR) Template
The following questions MUST be included and answered completely for each submitted IAR.

IAR must be submitted one of the following file formats: text or PDF.

Q1. Student Name: Emily Kozatchiner

Q2. Student ID: 1149665

Q3. Associated Team Deliverable: Interim Project Milestone 3

Q4. Team #: Group 5, Section 2

Q5. What were the main technical or methodological knowledge, skills and/or abilities (KSAs) that were required to complete this team deliverable? What prior courses or experiences (e.g. co-op, group project, etc.) from your Software Engineering degree did you draw on for these KSAs? (bulleted list is preferred):

- This sprint featured a lot of deliverables on my end that needed to be completed efficiently and correctly. The skills reminded me of my co-op placement when many tickets were assigned to me as an intern, and I had to tackle them not only quickly, but correctly. PR reviews and code pushes were also a huge part of the workforce and is used throughout the sprint.
- I believe I also completed a lot of testing for incorporated features. Testing and QA is an important skill to possess that I learned from previous internship and CIS*4150, a testing course I took to understand how to approach testing.
- Working in a sprint-like manner with a myriad of ticket assignments also reminded me of CIS*3760, where we also had a group of 5 that collaborated to complete tickets that we were interested in. Overall, this was a basic sprint that reused the skills that I learned in environments that used the Agile methodology

Q6. What was your existing level of experience with these topics/skills before your team began working on this deliverable? (1-2 sentences):

Definitely very experienced with completing and pushing simple code items. A lot of the skills I used will be present throughout my occupation, meaning that I have to be proficient in the basics.

Q7. Comment on your individual KSAs learning during this deliverable, and what additional learning may be needed to understand or be more competent with these topics / tasks in the future?

One KSA learning that I am actually very interested in exploring is conflict about a feature item. I had a disagreement on a design implementation between my group, and I felt that I conceded very quickly despite wanting to talk out why each option would be better. I am constantly learning how to best collaborate and share ideas, and I feel like this sprint showed me that I have more learning to do on how to handle disagreements and how to present ideas/thoughts.

Another individual learning was trying to be more vigilant in testing my code before merging, as I feel like I introduced bugs more often this sprint. I would like to take a longer time to test my code and extensively find use cases that could cause frequent problems because of my changes. Even though we caught many of these bugs, I encourage myself to test better on the following sprint and identify the edge cases.

Q8. What specific contributions did you make to this team deliverable? This should include technical or project management contributions.

- I implemented a safeguard against applying a late date on to-do tasks within the application. A warning message displayed to the user when adding/editing a past date/time. If the to-do item is added, it displays an icon or a red hue when the date is past.
- I ensured that applied filters by the user would remain applied if the user decided to reload the page. If some users want to keep a custom filter on their to-do items even on reload, they can with the new feature.
- Continuing on recurring task implementation, I added code for a user to edit a current recurring status on a task through the edit menu. Recurring status can be completely removed, added, or changed.
- Contribution on PR review and testing merged code in the main branch for any bugs that may have been missed during the review
- Seeked out collaboration and opinions on issue design and implementation and created child items for tasks based on inputs.

Q9. With whom did you collaborate for any of the above contributions (be specific – saying "all team members" is not sufficient. State which parts you worked on with whom)?

The sprint tickets assigned to me were solely individual work, so unfortunately not a huge amount of collaboration occurred for me. However, I collaborated heavily on the discussion of design for features and bug locating/fixing this sprint.

Many of my personal tickets lacked a certain direction in design implementation, so I made sure to be very vocal about my design choices and whether they fit the application. Jen and Jeremy really contributed to helping me design the due date icons and warning messages, working together to come to an agreed upon UI.

Towards the end of the sprint, Ben and I collaborated on reviewing each other's code/PRs to ensure that they were acceptable to demo. We found bugs within both of our PRs and Ben took the lead with fixing the issues to ensure that they would be merged properly, and our main branch would be correct.

Although Jen took the lead with organizing and splitting tickets throughout the sprint, I'd like to say that I helped create child tasks on some of the tickets and we all contributed to the discussion of ticket weights and how they should be handled. Overall, I believe that everyone had great collaboration this sprint, allowing for work to be completed efficiently.

Q10. Comment on how well you managed your time over the time period allocated in the Course timetable to this team deliverable (i.e. the time between the prior team deliverable to this team deliverable).

Time management was allocated very well this sprint, with the majority of us completing our tickets within the first week of the sprint according to the burndown chart. Board view was implemented immediately, the tech stories that would have caused merge conflicts if delayed were done ahead of time, and many others jumped on to their respective tasks.

Personally, I had a lot of tickets assigned to me at the beginning of the sprint, and managed to proportion my work equally between them throughout the two weeks of the sprint. As soon as a ticket was complete, there would be a time period allocated to testing main for bugs in order to catch and fix them before continuing. The cyclic process was efficient and consistent throughout the sprint.

Overall, everyone managed to complete their weight goal right before the deadline, with bug free code to demo on the following demo day. Considering our amount and heft of features, I would say we did a really good job.