ID5: Risk Analysis + Mitigation

New Risks

Technical Debt

For the stakeholder/developer that inherits this project they will also be inheriting our technical debt which could translate into hours of extra work.

- **Probability:** High
- **Severity:** High
- **Solution**: In an effort to mitigate this risk for the consumer we are going through the code to improve testability, making it more easily expandable and eliminating as many bugs as possible.

Directed triage:

Because of the number and size of bugs certain items had to be abandoned. There is the risk that we chose un-wisely and abandoned an easily implementable feature or bug fix. Also a risk of abandoning an item that would vastly improve the project.

- **Probability:** Medium
- **Severity:** High
- **Solution**: All issues were associated with a time estimation and the stakeholder was queried for a prioritized list of bugs to fix.

Old Risks

Critical path task blocking

This manifests when tasks that are necessary for the progression of the project or others work are not completed on time. This creates a blockage.

- **Probability:** Medium
- **Severity:** High
- **Solution**: Tasks are expected to be completed in a timely manner with the dev/test lead checking in to make sure the progress in on track.

Fatigue

In early deliverables, the severity/probability of fatigue was low. However, pushing towards the end of the term, it is not uncommon for students to experience ‘burnout’. Physical and emotional exhaustion would risk a drop in overall productivity, sloppy coding, and overlooking important details that could potentially derail the development process.

- **Probability:** High
- **Severity:** High
- **Solution**: Team members were encouraged to take regular breaks and ask for help when caught in a slump. A crackdown on regular code review as been established. Sonar Cloud has been used to check code coverage and code smells. And lastly, team members were encouraged to look after themselves - drink water, get 8 hours of sleep, etc.

Ensuring all Areas of the Deliverable are Being Worked On

When a software development team is faced with a large-scale project with fixed requirements and rigid deadlines, it was important to make sure that all areas of the upcoming deliverable are being worked on/completed. Anything that slips through the cracks has the potential to be costly.

- **Probability:** Medium
- **Severity:** Medium
- **Solution**: Since Trello ended up causing more confusion and frustration than it was worth, there was a switch back to Git issues for issue tracking. Regular stand ups, group meetings, and work sessions helped to ensure that all the deliverable requirements were up to date.

Time Constraints/Availability

As always, there is the risk of external factors (i.e. classes, jobs, personal commitments, etc.) influencing availability. With finals approaching, the possibility of time conflicts has only increased.

- **Probability:** High
- **Severity:** Medium
- **Solution**: There was only so much that could be done to mitigate this particular risk. All meetings and work sessions were scheduled in advance, with the intention of accommodating as many team members as possible. All foreseeable time constraints were brought up in advance. As per group policy, team members were expected to provide notice for last minute conflicts.

Unequal Team Contribution

It was anticipated from the beginning that different roles would come with different responsibilities, and that some of those roles would be more demanding. However, issues arise when individuals take on the bulk of the work, especially when compared to the efforts of those in similar roles. This can lead to animosity in the group as well as can contribute to burn out.

- **Probability:** High
- **Severity:** High
- **Solution**: A pie chart based on the totality of respective hours recorded in the time log, provided team members with a visual which displays the division of labour. Furthermore, the team members were expected to stay in touch with team leads, who could allocate tasks

Time Management

When approaching the later stages of software development, it was crucial that every single team member is able to put their time to good use. It was expected that every team member would be able to allocate time and effort towards each deliverable, but not at the expense of the workload from other courses. This is distinct from time constraints as poor time management can affect even those without other commitments.

- **Probability:** Low
- **Severity:** High
- **Solution**: Team members were encouraged to outline a set of achievable goals in advance, and allocate an appropriate amount of time to meeting them. With Trello out of the picture, team members were expected to take initiative in keeping up with Git, Git Issues, and Slack, for a clearer picture of expectations/deadlines. In this deliverable, the importance of attending all (possible) meetings/work sessions was emphasized.

Low Bus Number

From the very beginning of the semester, there has been the issue of whether there are enough team members – especially leads – to effectively and efficiently make progress with project development. Losing a team member only further complicated the situation.

- **Probability:** Low
- **Severity:** High
- **Solution**:There was an increase in communication between team members, and increased distribution of labour - particularly within the testing team, which was deemed to be particularly at risk.

Emphasis on Testing/Test Coverage

In earlier deliverables, testing was placed on the backburner, as there was very little to test. But in this final stage, testing is of utmost importance. Unknown side effects could have a negative impact on the quality of the project.

- **Probability:** High
- **Severity:** High
- **Solution**: The testing procedure has been revised in accordance to feedback from the previous deliverable. Assertions and logging were added to previous and new code. An early feature freeze took place to allow the testing team to cover existing features. Defects and test coverage continue to be tracked through git issues and the test matrix.

Definitions:

x	Low	Medium	High
Severity	The problem can be mitigated with little team coordination and effort.	The problem must be discussed, and a solution must be planned as a team.	The problem must be discussed, and a solution must be planned as a team. The problem may not be solvable within the allotted deadlines. This has a disastrous impact on the final outcome of the project.
Probability	The problem is not likely to occur.	The problem may occur, or external factors beyond control may contribute to the problem coming to fruition.	The problem is very likely to occur and should be immediately mitigated.