KASPER – ID1 PROCESS DOCUMENTATION

SOFTWARE DEVELOPMENT TEAM:

Project Manager: Tushita Patel

Developing Lead: Dylan Prefontaine

Testing Lead: Jeremy Liau

Build Manager: Christopher Mykota-Reid

Developers: Gaurav Arora, Arianne Butler, Haotian Ma, Kristof Mercier, Melody Zhao

Test Team: Christopher May, Ryan Tetland

Contents

[1.0 Activity Log 2](#_Toc473837649)

[2.0 Peer Reviews 2](#_Toc473837650)

[3.0 Client Communication 2](#_Toc473837651)

[4.0 Risk Assessment 2](#_Toc473837652)

[Intro 2](#_Toc473837653)

[4.1 Technical Risks: 3](#_Toc473837654)

[4.2 Non-Technical Risks: 5](#_Toc473837655)

[4.3 Future Risks: 8](#_Toc473837656)

[5.0 Meeting Notes and Minutes 9](#_Toc473837657)

[6.0 Pair Programming Sessions 9](#_Toc473837658)

[7.0 Task Assignments 9](#_Toc473837659)

[8.0 Inspections Report??? 9](#_Toc473837660)

# 1.0 Activity Log

# 2.0 Peer Reviews

# 3.0 Client Communication

* Tushita

# 4.0 Risk Assessment

## Intro

The purpose of this Risk Assessment is to identify and plan for issues that may come up from the beginning of the project until the due date for ID-1. The risks are divided into three categories – technical, non-technical, and future. Technical risks are risks arising from the construction and design of our system. Non-technical risks focus on risks related to group coordination, management, client relations, and various other components associated with managing our team. For each of the risks identified, a description of the risk will be given including the severity of the risk, the likeliness of the risk arising, potential scenarios that would cause the risk to materialize, a plan to mitigate the risk, and a contingency plan to deal with the issue if it occurs. The risk and loss sections are on a scale from zero to one, with one being most severe. These values are informal approximations and should not be considered measured probabilities.

## 4.1 Technical Risks:

Problems with Integration

**Probability:** 0.8

**Loss:** 0.6

**Scenario:** For this project, most of the team will be using new and unfamiliar technologies. This may cause issues with different pieces functioning together. This might include problems with build tools, testing frameworks, development tools, or one of the many other tools being used by the team.

**Mitigation:** Team members will do extensive research in deciding which tools to use. This will (hopefully) reduce the number of integration issues that arise.

**Contingency Plan:** If an integration issue arises, further development will be halted until the issue is fixed. The exact manner of resolving the issue will depend on the problem. If the problem is not fixable, the integration of different tools may come under consideration.

Dev Team Does Not Provide Test Team with Code by the Agreed Upon Deadline

**Probability**: 0.4 Medium

**Loss**: 0.2

**Scenario**: Dev team attempts to finish all requirements without regard for the agreed upon deadline and/or

underestimates the time needed to finish something. This could result in the deliverable being improperly tested and underlying bugs may persist.

**Mitigation**: Set firm deadlines for Dev team to complete implementation.

**Contingency plan:** If the Dev team does not adhere to the deadlines, the outcome is that the code will not be thoroughly tested.

Code is Not Designed for Testing

**Probability**: 0.6

**Loss**: 0.3

**Scenario**: A segment of code has too large a scope to test individual functionality and therefore cannot be thoroughly tested.

**Mitigation:** Dev team must maintain awareness that their code will be tested and must make use of assertions; this will allow the Test team to catch defects earlier.

**Contingency Plan:** If the issue is caught early enough, the Dev team may be able to restructure the code to make testing easier. Otherwise the test team will have extra work.

Code is Not Well Tested by the Deliverable

**Probability**: 0.6

**Loss**: 0.4

**Scenario**: The Test team becomes busy and does not spend enough time on testing. Unknown bugs will exist in the project.

**Mitigation**: Set up schedules for each deliverable and communicate effectively to ensure enough time and effort is spent on testing.

**Contingency plan:** Hope that no major bugs appear during presentation. Inform client that more testing will be done in future. If this issue arises, it may have to be accepted.

Team Members do not Document Work Well Enough

**Probability**: 0.7

**Loss**: 0.5

**Scenario**: A team member is more concerned about finishing their work than documenting what they did. Could also arise from the assumption that everyone understands what needs done, but there is a lack of communication regarding who is doing what.

**Mitigation**: Schedule and individual work logs are set up on GitHub. Team members will encourage each other to keep it updated. Grades are associated with work logs, which will hopefully encourage group members to keep track of their work.

**Contingency Plan:** If there is a lack of documentation, past work sessions will be documented as accurately as can be remembered.

We Do Not Use the Most Effective Tools for the Job

**Probability**: 0.8

**Loss**: 0.2

**Scenario**: Assumptions that certain tools will work together might be false, or Travis could undergo an update and no longer support a required package.

**Mitigation**: Tool options will be thoroughly researched and risk scans will be performed to ensure the most accurate assessment possible.

**Contingency plan**: If the wrong tool is chosen, work that is not useable will be thrown away to make room for the correct tool.

Misunderstanding Requirements

**Probability**: 0.7

**Loss:** 0.6

**Scenario**: This risk is expected to occur during client meetings and would result from miscommunication. This could be due to the use of “jargon” that the client might misinterpret. The client might also not understand each component involved in producing the final product. Another similar risk is having conflicting priorities with the client. In order to display our progress, new work and increasing functionality will be provided for each ID deadline. The team must meet with the client often enough to set realistic goals for each ID.

**Mitigation:** Many of these issues can be limited if there is frequent and qualitative communication between our project manager and the client. It is also important to note that communication with the client should be limited for the other team members. This will keep the project better organized and improve efficiency by allowing team members to focus on their individual tasks. User functionality desired by the client will require evaluation of the underlying components involved to achieve their goals.

**Contingency plan:** If there is a misunderstanding of requirements during the project, we may have to call a team meeting to discuss any misconceptions and ensure that everyone is in agreement on the requirements. We will also reduce the requirements planned for the present ID in order to fix the issues that exist and get the project back on track.

Travis Servers are Busy

**Probability**: 0.8

**Loss**: 0.4

**Scenario**: Travis servers are busy and our build is more time consuming than anticipated. This will cause delay in debugging.

**Mitigation**: Build manager will set up a local Travis container so that testing can occur without the Travis server.

**Contingency Plan**: If Travis is busy right before and ID deadline, contact Travis support. Otherwise, be sure to finish deployment well in advance of the deadline.

## 4.2 Non-Technical Risks:

Busy Schedules

**Probability:** 0.9

**Loss:** 0.3

**Scenario:** During the course of the term, group members may have other commitments that will reduce the amount of time they can dedicate to the project. The majority of our team is participating in the CSPIP program and are expected to have interviews over the last two weeks of January (ID-1). It is also anticipated that many team members will be less available during midterm season (ID-3), as well as towards the end of term (ID-5). This risk is expected to occur with all team members at some point throughout the term.

**Mitigation:** In order to reduce the effects of unavailable team members, knowledge of the entire project must be shared amongst the group. This will allow the project to move forward regardless of varying availabilities. Clear communication is required for scheduling meetings and deadlines that work for everyone. Communication will be done at stand-up meetings and via Slack. Additionally, daytime meetings and peer reviews will not be scheduled from January 23rd – 30th, when group members have scheduled interviews. All meetings, peer reviews, and stand-ups are scheduled for a minimum amount of time, leaving the majority of communication to occur via Slack.

**Contingency Plan:** When certain group members are unavailable, others must step in to make up for their absence. Vice leads for each team will be assigned and will take over when the leads are absent.

Team Member Unexpected Absence

**Probability:** 0.8

**Loss:** 0.3

**Scenario:** An unexpected absence for a team member may include sickness or other emergencies that cannot be planned for in advance.

**Mitigation:** Knowledge of the entire project must be shared amongst team members. This will allow the team to continue production even if someone is unavailable. To gain a thorough team understanding, each member will perform pair programming sessions at least once per ID. Regular code reviews with as many members as possible will also be a strong preventative measure.

**Contingency Plan:** Communication between team members when someone is unavailable and fair distribution of the missing member’s assigned work.

Team Member Informed Absence

**Probability:** 1

**Loss:** 0.2

**Scenario:** A team member might be absent and unable to contribute for a known amount of time. This might also occur over shorter periods for events such as job interviews.

**Mitigation:** Knowledge of the entire project must be shared amongst team members. This will allow the team to continue production even if someone is unavailable. Group members are also frequently advised to write well-commented code to aid in readability.

**Contingency Plan:** Time will be set aside in group meetings to communicate informed absences and to redistribute the work of the absent team member.

Team Member Drops Class

**Probability:** 0.2

**Loss:** 0.7

**Mitigation:** Keeping team morale high could help prevent students from dropping the class. This involves good communication, patience, positivity, and inclusion. All group members have been asked to inform the team as early as possible if they intend to drop the class. This could provide the team with extra time to adjust.

**Contingency Plan:** If somebody drops the class, that person’s work must be fairly distributed amongst the team.

Group Dynamic Problems

**Probability:** 0.8

**Loss:** 0.4

**Scenario:** If somebody feels that another group member is not contributing in a meaningful way, they may choose to deal with the issue in an inappropriate way. This may also arise from a disagreement over design details, what tools to use, functionality, etc. Our team expects to encounter disagreements throughout the project.

**Mitigation:** Keeping team morale high will help to ensure that team members approach conflict from a constructive place. Team members will be reminded that there are effective ways to confront an issue, and that all group members have valuable ideas, and therefore all opinions should be heard.

**Contingency Plan:** If a group member is upset, time should be taken to calm down before dealing with the issue. The group member should then try to handle the issue individually with the person they are in conflict with. Should this prove ineffective, they can take their issue to the project manager to help mediate the situation.

Inaccurate Time Estimations

**Probability:** 0.8

**Loss:** 0.4

**Scenario:** Time estimations may be inaccurate due to lack of experience.

**Mitigation:** Thorough research can be performed toimprove our knowledge and the accuracy of our estimations.

**Contingency Plan:** Where reasonable, reduction of requirements for the current ID can create extra time to implement the top priority features. This does not necessarily apply to situations where team members fail to allocate sufficient time for a prioritized task.

Client Is Not Satisfied with Product

**Probability**: 0.2 low (client is flexible and open minded)

**Loss**: 0.2

**Scenarios**: The work required to achieve the final product is underestimated and the team is unable to deliver what was promised.

**Mitigation**: Frequent communicate with the client will be required in order to keep everyone on the same page. Risks will be monitored in order to improve accuracy of time predictions.

**Contingency plan:** Let client know as soon as we know we cannot deliver what we promise. He may have to just accept it how it is.

Client Becomes Unavailable

**Probability**: 0.8

**Loss**: 0.2

**Scenario**: The CTO goes on vacation and does not reply to our emails.

**Mitigation**: Open and frequent communication with the client will help the team gain awareness of when the client might become unavailable.

**Contingency Plan**: The team will move forward with the project based on the agreed upon highest priority.

Build Master Becomes Busy

**Probability:** 0.8

**Loss:** 0.3

**Scenario:** Chris becomes busy with other classes and is temporarily unable to contribute.

**Mitigation**: Designate a member to fill in for Chris if he becomes absent. The designated person should shadow/pair program with Chris so that both of them understand how the build works.

**Contingency plan:** Working on designating a temporary Build Master.

## 4.3 Future Risks:

No Access to Database by ID-2

**Probability:** 0.2

**Loss:** 0.4

**Scenario:** In order to develop certain aspects of the app, the team will require access to the client’s database. This is currently unavailable.

**Mitigation:** Communication with the client must be clear so that we can plan our ID requirements based on what is available.

**Contingency Plan:** If access to the database is not provided in time, we will be forced to set up our own database in order to properly develop and test our app. Departmental help may be required for set-up.

Server issues

* To be determined in later deliverables

# 5.0 Meeting Notes and Minutes

# 6.0 Pair Programming Sessions

# 7.0 Task Assignments

# 8.0 Inspections Report???

* Tushita