**Risk Assessment**

**Intro**

The purpose of this document is to identify and plan for issues that may come up from the beginning of the project until the due date for ID 1. 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, 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 does come up.

* **Misunderstand Requirements**

**Risk**: 0.7

**Loss:**

**Scenario**: We expect this risk to come up within the meetings with our client because of miscommunication. This could be because of the use of “jargon” that the client is used to using but is not properly understood. The client may also not entirely understand all components involved in producing the final product that they desire. In order to mitigate this risk,

Another potential risk involved is having different priorities than the client. In order to keep the client happy with our progress, we must provide them with something new for each deadline for the ID's. We 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 good and often communication between our project manager and the client. It is also important to note that communication should be limited for the other team members that are not the project manager or leads. This is important because it will keep the project better organized and improve efficiency by allowing team members to focus on their individual tasks. We also must evaluate the desired user functionality that the client wants, and also what underlying components must be involved to achieve this.

**Contingency plan:** If there is a misunderstanding of requirements during the project, we may have to reduce the requirements planned for the specific ID that we are on in order to fix the issues that exist and get the project back on the right track.

* **Busy Schedules**

**Risk:** 0.9

**Loss:**

**Scenario:** During the course of this term many of our group members may have other commitments that would reduce the amount of time that can be dedicated to our project. We expect that this will come up with all of our team members at some point in the term.

**Mitigation:** In order to reduce the effects of having any team member unavailable, we must share the knowledge of all aspects of the project with many different people. This will allow the project to move the project forward even if a team member is busy with other things. We also must communicate with each other to schedule meetings and deadlines that work with our schedules. This communication will be done at standup meetings before class and also on Slack.

**Contingency Plan:** When certain group members are unavailable, other people must step in to make up for their absence. We will assign vice leads for each team that will have to take over when the leads are unavailable.

* **Team Member Unexpected Absence**

**Risk:** 0.8

**Loss:**

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

**Mitigation:** Sharing of knowledge of the project must be shared throughout each team. This will allow the team to still be productive even if someone is unavailable. To do this we will perform pair programming sessions at least 1 time per person for each ID, and also have regular code reviews with as many people present as possible.

**Contingency Plan:** Communication between team members will have to take place to let each other know when someone is unavailable, and whether they have any assigned work that will need to be taken care of.

* **Team Member Informed Absence**

**Risk:** 1

**Loss:**

**Scenario:** This could come up if somebody has a trip planned and will not be able to contribute for a known amount of time. This could also be for shorter periods of time for things such as job interviews.

**Mitigation:** Sharing of knowledge of the project must be shared throughout each team. This will allow the team to still be productive even if someone is unavailable.

**Contingency Plan:** We will spend time in group meetings to let everyone know when someone will be unavailable and assign any work that they need done to somebody else who can do it.

* **Team Member Drops Class**

**Risk:** 0.2

**Loss:**

**Mitigation:** To try to prevent people from dropping we will try to keep morale high. This means trying to get along with each other and not get angry at each other. This will

**Contingency Plan:** If somebody drops the class, we will have to assign that persons workload to other group members.

* **Problems With Integration**

**Risk:** 0.8

**Loss:**

**Scenario:** For this project, many of us will be using new technologies that we are not familiar with, and this may cause issues with different parts working together.

**Mitigation:** We will have to do lots of research for the options that we consider and then choose what things we will use. This will hopefully reduce the number of integration issues that will come up.

**Contingency Plan:** If there is an integration issue we will have to stop any further development until we fix the issue. Fixing the issue will depend on what the problem is with, and if it is not fixable we may have to consider switching programs.

* **Group dynamic problems**

**Risk:** 0.8

**Loss:**

**Scenario:** If somebody feels as though another group member is not pulling their weight for the project they may get upset and deal with the issue in an inappropriate way.

**Mitigation:** We will try to keep morale high, and remember that there are proper ways to go about letting somebody know that you are upset with them.

**Contingency Plan:** If somebody is feeling upset they should consider taking some time to calm down before dealing with the issue. First they should approach the person that they have issue with, if this does not work then they should go to the project manager to help mediate the situation.

* **Inaccurate Time Estimations**

**Risk:** 0.8

**Loss:**

**Scenario:** We do not expect all of our time estimations to be accurate because we are not very well experienced with many of the things that we are using.

**Mitigation:** spending some time researching the new things that we will be using may improve the accuracy of our estimations.

**Contingency Plan:** To deal with this risk when it comes up, we will have to reduce the requirements for the particular ID that we are on. This will give us more time to get some things completely finished.

* **No Access to Database By ID 2**

**Risk:**

**Loss:**

**Scenario:** In order to develop some parts of the app, we will need access to the client’s database, which we currently cannot access.

**Mitigation:** We must communicate with the client so we can plan our tasks and requirements for the ID based on what we have access to.

**Contingency Plan:** If we are not given access to the database in time, we will have to set up our own so that we can properly develop and test our app. We may need help from people in the department to set this up.

* **Communication issues with client**
* **Security Requirements for Different Platforms**

**Risk**: 0.4

**Loss**:

**Scenario**:

* **Client Is Not Satisfied With Our Product**

**Risk**: 0.2 low (he seems open to test out almost anything)

**Loss**:

**Scenarios**: We underestimate the work required to achieve what we say we will do and the final product for the deliverable is not what we promised.

**Mitigation**: Communicate with client to stay on the same page. Monitor risks to improve accuracy of time needed.

**Contingency plan:** Let client know as soon as we know we cannot deliver what we promise.

* **Dev Team Does Not Provide Test Team With Code Early Enough**

**Risk**: 0.4 Medium

**Loss**: deliverable is not tested well enough and may have bugs that we do not know about.

**Scenario**: Dev team wants to finish all the requirements with disregard for the deadline.

Dev team underestimates the time needed to get their work done.

**Mitigation**: set firm deadlines for dev team to give test team the code.

**Contingency plan:**

* **Code is Not Designed Well For Testing**

**Risk**: 0.6

**Loss**: more work for testers, not as well tested.

* **Code is Not Well Tested by The Deliverable**

**Risk**: 0.6

**Loss**: unknown bugs will exist in the project

**Scenario**: testing team becomes busy and does not spend enough time on testing.

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

**Contingency plan:** Hope that no major bugs appear in presentation. Let client know that more testing will be done in future.

If this issue comes up it may have to be accepted.

* **Client Becomes Unavailable**

**Risk**: 0.8

**Loss**:

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

**Mitigation**: Set up schedule for meetings with stakeholder. Communicate with stakeholder to know when they will be unavailable.

**Contingency Plan**: move forward with project based on what we think is highest priority.

* **Build Master Becomes Too Busy**

**Risk:** 0.8

**Loss:**

**Scenario:** Chris has a busy week with other classes and cannot keep up with this class.

**Mitigation**: Have a designated member to take over for Chris (Dylan). Have Dylan shadow/ pair program with Chris so that both of them understand how it is working and also what needs to be done.

**Contingency plan:** Dylan takes on role as the Build Master.

* **Team Members do not Document Work Well Enough**

**Risk**: 0.7

**Loss**: based on marks (no record of work)

**Scenario**: team member is more concerned about finishing their work and forget to document what they did. We assume that everyone understands what needs done but there is a lack of communication.

**Mitigation**: set up schedule, and individual work logs on GitHub. Encourage each other to keep it updated.

Grades are associated to work logs which will encourage each group member to keep track of their work.

**Contingency Plan:** Document past work sessions as accurate as can be remembered.

* **We Do Not Use The Proper Tools For The Job**

**Risk**: 0.8

**Loss**:

**Scenario**: We assume that some tools will work together but turns out there exists a better option.

Using ionic v1 or v2??

Travis has an update and no longer supports a package we need (unlikely)

**Mitigation**: research the options for all the tools we could use early and perform risk scans to ensure we know that it will work well.

**Contingency plan**: throw out work that cannot be used from previous tool and change to new one.

* **Server issues (more of a risk later on deliverables)**