Risk List ID1

Group Risks:

Lack of experience with the technical stack.

- Probability: high
- Severity: medium
- Solutions:
    - Learn the technology stack gradually during the project
    - Team bootcamps and group learning efforts
    - Speak to Osgood about his ‘react guru’

Low bus number, combined with team member unreliability. May be the result of many factors such as winter weather, final schedules, etc. This could result in missed deadlines and unexpected pressure on the other team members.

- Probability: medium
- Severity: high
- Solutions:
    - Pair program to share responsibility
    - Regular meetings and stand-ups to share progress
    - Notify project manager and risk officer about schedule conflicts

Member might drop out of the course

- Probability: low
- Severity: high
- Solutions:
    - Increase bus number
    - Encourage open communication for group members feeling overwhelmed by other courses

Continuous stream of requirement changes

- Probability: low
- Severity: medium
- Solutions:
    - Meet with stakeholder(s) to discuss a feature freeze

Technical risks:

Once updated a major version of software, we might break our code since function will be deprecated.

- Probability: medium
- Severity: low

- Solutions:
    - Use the old version of software
    - If problems could be fixed immediately or source code have been tested beforehand, update to the new version

Shortcomings in external software which we depend on.

- Probability: medium
- Severity: medium (or high depending on when the problem is found)
- Solutions:
    - Ensure adequate amount of prototyping is done which stress the limits of the technology we are choosing
    - Prototypes should be created to serve a specific use case and not be arbitrary implementation.

Unrealistic expectations about performance of large datasets on low powered computers

- Probability: high
- Severity: high
- Solutions:
    - Frequently test our data processing on computers of similar specification that the clients will be using
    - Have an accurate description of the hardware our software is expected to run on
    - Have an accurate set of sample data to test with

Scope/Size of Effort underestimated

- Probability: High
- Severity: High
- Solutions:
    - Use scalable work breakdown technique and plan for possible addition of aid during the process of a deliverable
    - Advocate more resources than possibly needed for a deliverable; the added member would not necessarily work but will be kept updated incase their input is required/needed.

End Date of project predetermined without analysis of implication; project plan/schedule is not structured by the team rather by stakeholder

- Probability: High
- Severity: Medium
- Solutions:

    - Stakeholder(s) may be updated earlier about realistic possible phase at the intended time of project completion.
    - Project team should ideally prioritize functions relatively critical

Communications and Information flow might be limited

- Probability: Medium
- Severity: Severe
- Solutions:
    - Use separate slack channel post different kinds of events
    - Members need to check slack often

Imperfect prioritization

- Probability: Medium
- Severity: Medium
- Solutions:
    - Check the deadline of the issue posted on git, if impossible to finish before the deadline, project manager should discuss it later.

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 planned as a team.	The problem must be discussed and a solution 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 out of our control may contribute to the problem coming to fruition.	The problem is very likely to occur and should be immediately mitigated.