

Deliverable 1 - Dev						Deliverable 1 - Test					
Risk	Score	Severity (1/5)	Likelihood (1/5)	Mitigate	Contingency	Risk	Score	Severity (1/5)	Likelihood (1/5)	Mitigate	Contingency
1. Dropping class	10	5	2	- be respectful to group members - encourage group members	- have someone shadow high priority roles	1. Dropping class	10	5	2	- be respectful to group members - encourage group members	- have someone shadow high priority roles
2. Changing requirements	2	2	1	- meet with stakeholder frequently	- chosen hosting platform: Firebase allows easy database integrations if requirements change	2. Code changes close to deadline	8	2	4	- put in strict deadlines	- shadow dev team to prepare for what tests may be needed
3. Lag with datasets	12	4	3	- use small datasets - test early with dataset to observe need of optimizations	- create a server using node.js or other server language to offload	3. Unoptimized code	6	3	2	- testing to find the unoptimized code	- prioritize refactoring code
4. Inadequate documentation	4	2	2	- review code (methods and classes) before pushing to codebase - follow strict documentation rules	- refactor for self explanatory code	4. Inadequate documentation	4	2	2	- review code (methods and classes) before pushing to codebase - follow strict documentation rules	- refactor for self explanatory code
5. Poor communication	3	3	1	- discord always available - discussions frequently	- use discord for emergency meeting	5. Poor communication	3	3	1	- discord always available - discussions frequently	- use discord for emergency meeting
6. Learning new tech stack (gherkin, webxr, etc.) takes up time, higher chance of mistakes	9	3	3	- open communication between dev and test team to share what they know and help each other out - share where to learn this new tech stack	- prepared to help each other - walk group member through their problem - take on another task	6. Learning new tech stack (gherkin, webxr, etc.) takes up time, higher chance of mistakes	9	3	3	- open communication between dev and test team to share what they know and help each other out - share where to learn this new tech stack	- prepared to help each other - walk group member through their problem - take on another task
7. Scope creep	4	4	1	- become familiar with requirements - regular meetings with stakeholder	- communicate if you notice some scope creep and plan for redirection	7. Manual integration test plan	12	4	3	- research automatic testing - test different automatic testing tools	- well document the manual testing - clean test code - create proper test cases with manual testing