		Impact				
		Negligible **1**	Minor **2**	Moderate **3**	Major **4**	Catastrophic **5**
Likelihood	Almost certain **5**	Moderate **5**	High **10**	Extreme **5**	Extreme **20**	Extreme **25**
	Likely **4**	Moderate **4**	High **8**	High **12**	Extreme **16**	Extreme **20**
	Possible **3**	Low **3**	Moderate **6**	High **9**	High **12**	Extreme **15**
	Unlikely **2**	Low **2**	Moderate **4**	Moderate **6**	High **8**	High **10**
	Rare **1**	Low **1**	Low **2**	Low **3**	Moderate **4**	Moderate **5**

Risk	Likelihood	Impact	Risk Rating	Preventative Measures	Response
Run out of time	4	3	12	Complete tasks with higher prioritisation first and start tasks with higher story points earlier. Refer to kanban board	Complete the tasks for all user stories that you can with the time you have left. If you ran out of time due to personal or unforeseen circumstances, talk to project trainer
Loss of files	2	4	8	Push all files and commits to your Github repository regularly as well as always saving files	Use last saved peice of work or most recent work from Github to build your code/databases again
Poor quality code	3	2	6	Test code frequently and use best pratises learnt throughout the Java teaching	Resolve bugs and logical errors when they're found
Low productivity	5	2	10	Set goals during the project using the SMART technique	Take time away from the project to clear your mind and come back to it feeling fresh
Inaccurate estimations	4	4	16	Focus on the work that has immediate priority. Include techspikes - allocated time to research an unfamiliar part of the project	Work on projeect out of training hours. Talk to the project trainer and explain the circumstances
Poor risk management	2	2	4	Identify potential risks and the likelihood of them. Create risk mitigation plans and carefully monitor the risks	Create a risk assessment upon the first encounter with a risk in order to prevent any other risks from occuring or taking too much time out of the project