Table of Contents

Risk	Statement	Response	Objective	Likelihood	Impact	Risk Level
SQL Injection	Inputs could modify the database, removing or modifying entries	Using prepared statements for user input	Properly escaping user input to stop injection attacks from succeeding	Possible	Severe	10
Dropped Database	Developers accidentally dropping the database, causing all entries to be deleted	Using a sepereate database for testing, keep regular backups	Preventing the most likely causes of the database dropping and add a contingency if it does happen	Unlikely	Major	4
Code Gets Removed	The code for running the program gets removed from the machine that hosts it	Keep tagged releases in a seperate repositoty (github)	Providing a means of reobtaining the program from somewhere that it can't be easily removed	Unlikely	Major	4
Incorrect input	The input on the command line doesn't fit the design of the database	Add constraints to database, allow for updating items	Constraints help prevent bad inputs from being stored, the update query allows bad inputs to be corrected	Expected	Minor	8
Host Computer/s fails	The computer/server responsible for hosting the code/database stop working	Have backups of project and database allowed elsewhere	If the computer/server needs replacing, the system can be set back up easily and is up-to-date	Unlikely	Severe	5