Project Name	Project Fundimentals (I
Prepared By	Lloyd Low
Date	11-12-20

Problem Area or Activity	Risks Identified
Overall Project	Risk of Covid-19
	Loss of Files/progress
	Internet outages
	Leaving GCP running
	Running out of time
Sprint 1	Failing to update sprint board
	sql instance password leak
Sprint 2	Making changes on the wrong branch
	Accidentally merging or committing
Sprint 3	Improper coding methodolgy
	Stealing others code
Sprint 4	SQL Syntax errors
	Overcomplicating code
Sprint 5	Testing throws many errors that slow testing
	Misunderstanding of testing methodology
	Fatal testing errors or defective files
	Not testing methods that have been added to deal with faults
Sprint 6	Overcomplicating additions
	Not updating Tests to accommodate changes
Additional Risks Identified in Hindsight	Not fully understanding java-eclipse or SQL
	DataBase build and teardown issues
	NullPointerExceptions
	PC dying

MS)

Probability Of Occurance	Impact Intensity	Mitigation Strategy
Low	High	Wear a mask when out in public
Low	Extreme	Ensure regular commits
Medium	Medium	Ensure a stable internet connection
Medium	Low/Medium	Always check the GCP server is spun down after use
Medium	Extreme	Properly utilise Jiras planning functionality to elect time to core issues
Medium/High	Low	Regular checks and detailed stories to ensure bits aren't missed
Low	High	Set it to a non-standard password aka not "root"
Medium	Medium	Git bash tied to eclipse opening
Medium	High	Only allow merges to developer to take place through githubs portal
Medium	Medium	Make sure to refer to the guidance on QA community.
Low	High	Always give credit if any work is taken from another.
Medium	Medium	Refer to cheat sheets and QA community before implementing any SQL
High	Medium	Aim for brevity and minimise using overly long variable and method nam
High	Medium/High	Ensure to allocate a large amount of time to testing
High	Low/Medium	Make sure to refer to the guidance on QA community.
Low	High	Ensure thorough testing and high coverage
Medium	High	Methodical checking of tests to ensure they all pair up
High	Low/Medium	If this can't be avoided then ensure that the new code is in a new metho
Medium/High	High	Run all tests in coverage and address all faults.
Medium	Extreme	Properly read through notes made during the course so far.
High	Low/Medium	Read through all files before commencing to understand that the system
Medium	Medium	Understand the interactions between interfaces and that Controllers can
Low	Extreme	Ensure regular pushes to save as much progress from being lost as pos

Additional Measures
Observe social distancing
Save often
Connect via ethernet cable where possible
Set up expenditure warnings
Take the approach that barebone yet functional code is better than complex and broken code.
Set dependancies and blockers to ensure things are done in the correct order
Use local host of the sql instance for better security
Have added commented out text to all files to remind myself
Have set developer as the default branch and familirised myself with the steps needed to roleback commits
Avoid 'copy-paste without understanding' mentality
Avoid 'copy-paste without understanding' mentality
. methods
nes
keep methods in DAOs and Controllers concise
Read through the pre-created customer tests to better understand implementations
Ask for help if really stuck on certain aspects.
Run Junit tests in coverage mode to see where the tests don’t reach so that tests can be created to address that
od so that it can be easily unit tested and doesn’t alter other tests too much.
Work through code line by line to ensure all method calls are mocked appropriately.
Complete examples on QA community and don’t be afraid to ask for help.
n calls and builds a database from a file within the repository.
n't call eachother.
ssible.