CODE QUALITY REVIEW REPORT

This review checklist was necessary to follow as multiple team members were working on different sections of the code. As the project progressed, we worked on meeting these guidelines we personalized for our project from general well known principles. We used elements of the Facade design pattern and the Dependency Inversion Principle to ensure modules were independent and exhibited separation of concerns.

TOPIC	YES/NO	COMMENTS
Are appropriate unit tests carried out?	Yes	
Is the code easy to understand, with clear logic?	Mostly	
Are meaningful variable and function names used?	Sometimes	
Is the code well-documented?	Yes	Within the team
Are consistent formatting practices followed?	Yes	
Does the code adhere to established coding standards for languages used	Yes	
Is there any duplicated code that could be refactored into reusable functions or components?	Yes	We had to repeat a function that could have been refactored because we did not want to create errors as each case required to be more tailored
Are the code units easily testable and debuggable?	Yes	
Are functions and classes appropriately sized, with limited complexity?		
Is there high cohesion and low coupling	Yes	
Do functions and classes adhere to the single responsibility principle, focusing on a single task or responsibility?		
Is error handling implemented?	Yes	Where we could we definitely included error
Useful Comments	Yes	
Too many comments?	No	We used comments during

		development to guide each member, but we got rid of unnecessary comments as it came to a close
adherence to the defined architecture	Yes	We followed some Facade elements but were not too restrictive on ourselves
Avoid hardcoding	Yes	