

# Must Have

1. A Kanban board with full expansion on user stories.
2. A record of issues experienced throughout the project.
3. A relational database used to store data persistently for the project. 3 tables to demonstrate understanding.
4. A functional application
5. Unit tests and integration tests for validation of the application. (Strive to reach coverage of 80%)
6. Code fully integrated into a Version Control System (GitHub) using the feature/branch/model.
7. A risk assessment which outlines the issues and risks faced during the project time frame.
7. Fully designated test suites for the application, as well as automated validation of the application.
8. The project as a

# Should Have

1. 1 simple design provided meeting the agreed standards of patterns
2. Structured README.md with relevant content
3. Software is fully functional in all necessary areas (MVP) but still has small bugs and/or errors
4. Software has been tested in the most important areas, (judged by test coverage) with basic functions being tested. Best practises were adhered to for most areas of the project
5. Tools and workflows discussed in documentation are implemented throughout the project at an acceptable level, but more exploration could have occurred.
6. An acceptable level of

# Could Have

1. More extensive designs provided on both a system level and component level.
2. In depth README.md that explores the application of a high level.
3. Software has been tested in most relevant areas; best practises were consistently adhered to throughout the project.
4. Software is functional in all major areas but still has small bugs/errors.
5. Tools and workflows discussed in documentation are implemented throughout the project at a high level.
6. Analysis as to what was tested, what was out of scope and the reasons behind these decisions.
7. Summary of overall results is present.
8. A good level of test coverage met, utilising a broad range of tests.
9. A build server has been installed and used to produce a software

# Won't Have

1. Due to time constraints, the project may not be 100% completed
2. Fully functioning application in the highest specification criteria
3. All bugs and/or errors completely removed from the code
4. In depth details within some parts of documentation.
5. A vast range of colours and different shapes and size customisation of buttons and boxes.
6. Multiple tables interlink through the back end.

whole must meet the specification requirements stated within the Kanban board (Jira).	6. An acceptable level of test coverage met, utilising a good range of tests	after a push to the code repository.
9. . Fully functional application created in the OOP language, following the best practises and design principles.	7. Some analysis as to what was tested and why. Summary of overall results is present but isn't in much detail	10. Instructions on how to run integration test OR reports containing integration tests
10. . Required to model a relationship with the ERD.	8. A build serve has been installed and used basically to produce a software build	11. A more formal risk assessment process has been followed using the risk assessment matrix.
	9. Tests have been written that test that the application can access other systems that is connected to, for instance a database.	12.A dropdown menu
	10. Add a instance to table	13.More implementation with JavaScript to develop a more appealing interface.
	11. List all users/items.	
	12. Changes to users/items and orders need to be tied to a user	14. Basics tests written for the back-end. Results produced not acknowledged in any way. Some risks acknowledged but not fully followed up.
	13. Must be able to log in.	15.VCS implementation covering all requirements.
		16. Evolution of designs evident as the project progresses in all aspects.
		17.Deployment activities logged in a report. Certain advanced tools and techniques used have been explained thoroughly.

