MoSCoW Week 5 Project

Must have → create MVP

* Kanban board used on Jira to produce project management board. Needs to have full expansion of :(for the project)
  + user stories
  + Acceptance criteria
  + Tasks
* Code fully integrated into Version Control System (i.e. GitHub)
  + It should use the Feature-Branch Model - **main/dev/features**
* Produce a Risk Assessment for all issues and risks during project timeframe
* Use a relational database to store data continuously.
  + The database needs **at least 2 entities**
  + Also required to model a relationship with an **ERD** to show understanding
* Create a functional application in the OOP-Language (**JAVA**) -
  + Follow best principles coveredthat meet KANBAN board with a functioning front-end website and integrated APIs.
* Create a functional application front-end which connects to the back-end API
* Create a build of the application
  + Must include any dependencies it might need
  + Produced using an integrated build tool
* Use Static analysis tool (**SonarQube**) to run code
  + Show and use relevant refactoring of code according to the analysis smells, bugs, and vulnerabilities
* Create fully designed test suites for the application, as well as automated testing for validation of the application.
  + Must meet 80% of test coverage in backend and provide consistent reports and evidence to support a TDD approach
* Must use following tech stack:
  + Version control system - Git
  + Source Code management - GitHuib
  + Scrum board - Jira
  + Database - MySQL Server 5.7+ (local or cloud hosted)
  + Back-end Programming language - Java
  + Build Tool - Maven
  + Unit Testing - JUnit
  + Integration Testing - Mockito

Tools and workflows discussed in software documentation, implemented throughout project at high level

* Omit irrelevant stuff from VCS, and store code in a structured manner with branches. Use of configuration file to make build jobs portable.
* Relevant Tests written using relevant CRUD features within app.

Should have

* Fully functional software which has been tested in all areas
* Best practices followed and demonstrated, refactoring adhered to and implemented throughout project
* Software is in line with documentation with changes made where necessary. Designs fully implemented with justifications.
* Use git ignore files, and scripted hooks.

Could have

* Software implements concepts outside of brief at good level.
* Refactoring of code is rife throughout all relevant areas of software
* Build server installed and successfully built software after a push with artefact produced for successful builds

Won’t have