Fundamental Project Specification

Date of Issue: January 2020

CONTENTS

[Introduction 2](#_Toc35299140)

[Objective 3](#_Toc35299141)

[Domain 4](#_Toc35299142)

[Scope 5](#_Toc35299143)

[Constraints 6](#_Toc35299144)

[Deliverable 6](#_Toc35299145)

[Milestones 7](#_Toc35299146)

# Introduction

The purpose of this document is to outline the individual project specification that you will be working on during the training. This project will involve concepts from all core training modules; more specifically, this will involve:

* Agile & Project Management
* Databases
* Cloud Fundamentals
* Programming Fundamentals
* Continuous Integration
* Automated Testing

The individual project must encapsulate all aspects of the aforementioned modules. You will be provided with a domain to work with however creativity is encouraged as long as you meet the minimum requirement.

# Objective

The overall objective of the project is the following:

* To create an application with utilisation of supporting tools, methodologies and technologies that encapsulate all fundamental modules covered during training.

Specifically, you are required to create an application using the language from your Programming Fundamentals Module which interacts with a Managed Database. You must plan the approach you will take to complete this project using the design techniques learnt, and also create a CI Pipeline that can automate the building and deployment of your artifact.

# Domain

You are required to build an application that an end user can interact with via a CLI (Command Line Interface). The application needs to be an inventory management system that needs to be able to:

* Add a **customer** to the system
* View all **customers** in the system
* Update a **customer** in the system
* Delete a **customer** in the system.
* Add an **item** to the system
* View all **items** in the system
* Update an **item** in the system
* Delete an **item** in the system
* Create an **order** in the system.
* View all **orders** in the system.
* Delete an **order** in the system
* Add an **item** to an **order**.
* Calculate a cost for an **order**.
* Delete an **item** in an **order**

When considering the entities in this domain:

* A **customer** needs to have a name.
* An **item** needs to have a name and a value.
* An **order** needs to have a **customer** and contains **items**.

**Extension (unmarked)**:

* Add a **user** to the system
* List all **users**
* Changes to **customers**, **items** and **orders** need to be tied to a **user**.
* A **user** should have a **username** and **password**
* You must be able to log in as a **user** within the system to make any changes.

# Scope

The requirements set for the project are below. Note that these are a minimum set of requirements and can be added onto during the duration of the project.

The requirements of the project are as follows:

* A Kanban board with full expansion on user stories and tasks needed to complete the project. It should also provide a record of any issues or risks that you faced creating your project.
* A relational database used to store data persistently for the project, this database needs to have at least 3 tables in it, to demonstrate your understanding, you are also required to model a relationship.
* A functional application, following best practices and design principles, in a language that you have covered during training that meets the requirements set on your Kanban Board.
* Unit tests and integration tests for validation of the application. You must strive to provide high test coverage.
* Code fully integrated into a Version Control System which will subsequently be built through a CI server and deployed to an artifact repository manager.

You should consider the concept of MVP (Minimum Viable Product) as you plan your project, complete all the requirements above before you add extra functionality that is not specified above.

# Constraints

The time constraint of this application will be discussed when the specification is given out, as this can fluctuate based on several factors.

The other constraint for this is certain technology that needs to be used. The application needs to utilise the technology discussed during the training modules. The tech stack required would be the following:

* Kanban Board: Trello or GitHub project
* Database: GCP MySQL server or other Cloud Hosted managed Database.
* Programming language: Language covered in Programming Fundamentals (Java)
* Testing framework: JUnit
* Version Control system: Git
* Host of version control system repository: GitHub
* CI Server: Jenkins
* Cloud server: GCP virtual machine or other equivalent Cloud hosting Option.
* Artifact Repository Manager: Nexus or Artifactory

# Deliverable

The final deliverable for this project is the completed application with full documentation around utilisation of supporting tools, this will require a fully functional application.

A presentation of work will be required towards the end of the deadline. However, you will be required to track designs and work throughout the duration of the project and be able to show how they changed overtime.

You will be required to push your code to the Master branch once a week – this is so that the Trainer can keep track of your progress.

# Milestones

|  |  |
| --- | --- |
| **Milestone** | **Completion date:** |
| Remote database for the project is set up | End of Week 2 |
| ERD for application completed | End of Week 2 |
| Finish risk assessment | End of Week 2 |
| Hard coded SQL CRUD statements should be created for customers, items and orders and work as expected. | End of Week 3 |
| The hard-coded SQL statements now take in variables to create dynamically generated SQL statements.  The Java application should contain methods to take in these variables and use JDBC to connect and send these queries to a remote database. | Start of Week 4 |
| Take in user input and send the information to the correct method, which will send the query to the database | End of Week 4 |
| Continuous integration server automatically pushes package to nexus | Wednesday of Week 5 |
| Have a running, working and tested application which takes in user input and can create, read, update and delete customers, items and orders. | Wednesday of Week 5 |
| Complete a PowerPoint of the built application | Thursday of Week 5 |