Ahmad Naeem

QA  - Ahmadzeya.Naeem@acadmeytrainee.com

Fundemental Project Report

Table of Contents

[1. Introduction 3](#_Toc44517298)

[2. Requirements 3](#_Toc44517299)

[2.1 Functional Requirements 3](#_Toc44517300)

[2.2 Non-functional Requirements 4](#_Toc44517301)

[3. Risk Analysis 4](#_Toc44517302)

[3.1 Risk Matrix 4](#_Toc44517303)

[4. Design 5](#_Toc44517304)

[4.1 Database Design (Entity Relationship Diagram) 5](#_Toc44517305)

[4.2 Class Diagram 6](#_Toc44517306)

# Introduction

This is the first project I am working on during my training at QA. The goal for the project is to create a Java application which uses a command line interface to perform CRUD operations on a database. In order to achieve this, I will be using the following tools:

* **Version Control System**: Git
* **Source Code Management:** GitHub
* **Kanban Board**: Jira
* **Database Management System**: GCP instance of MySQL Server
* **Back-End Programming Language**: Java
* **Build Tool:** Maven
* **Unit Testing:** Junit

While completing this project I hope to improve my knowledge and skill set when working with these tools to better prep me for working in industry in the near future.

# Requirements

The requirements for this project are pretty straight forward. Here I have summarised them in Functional and non-functional requirements where the aim is to reach all requirements in order for the project to be a success.

## Functional Requirements

1. Program must be written in Java
2. Must have a database set up in GCP
3. Uses SQL to perform CRUD operations
4. Uses a Command Line interface
5. Add a customer to the database
6. View all customers in the system
7. Update a customer in the system
8. Delete a customer in the system
9. Add an item to the system
10. View all items in the system
11. Update an item in the system
12. Delete an item in the system
13. Create an order in the system.
14. View all orders in the system.
15. Delete an order in the system
16. Add an item to an order.
17. Calculate a cost for an order.
18. Delete an item in an order

## Non-functional Requirements

1. Should be runnable on any device which has Java installed once it’s been pulled down from GitHub
2. Should have regular versions accessible on GitHub
3. Should have database stored on Google Cloud Platform

# Risk Analysis

## Risk Matrix

Here I have created a table to analyse risks and suggested ways of preventing them. The likelihood is the possibility it will occur where 1 is low and 10 is highly possible. Impact is how badly the project will be effected if the risk did occur where 1 is not effected much and 10 is effected massively. The overall risk can be either low, medium and high. It takes into account likelihood and impact:

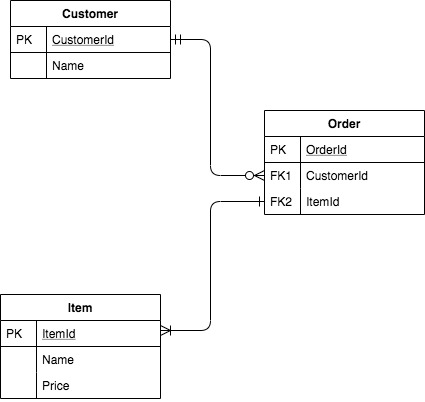
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Description | Prevention | Likelihood (1-10) | Impact (1-10) | Overall Risk |
| 1 | The whole project or parts of the project gets deleted from your computer accidently. | Use GitHub to regularly backup all files so if any data is lost it can be cloned from the repository. | 3 | 9 | MEDIUM |
| 2 | You make a major mistake and need to go back to a previous stage of working to undo the mistake. | Use GitHub and regularly make commits so if you need to revert to a previous version you can do that. | 7 | 5 | HIGH |
| 3 | The requirements are not understood properly and so the wrong program is created | Read the requirements carefully and fully understand it before designing. Ask the client prior to starting to fully clarify. | 4 | 8 | HIGH |
| 4 | You have technical difficulties and the laptop you are using to develop the program stops working | Have a backup device that has all the appropriate application to develop the program installed. Use GitHub to make transitioning between devices quick. | 3 | 4 | LOW |
| 5 | GitHub servers go down so you don’t have access to your repository temporarily or permanently loose data. | Create a hard back up regularly on a non-cloud service in case servers go down. | 1 | 8 | LOW |
| 6 | Program does not follow laws regarding data protection resulting in company being sued. | Ensure understanding of all laws which concern the company. Get a lawyer to review program before releasing. | 2 | 8 | LOW |
| 7 | You get ill and as a result fall behind the deadline | Create a plan to ensure you are working at the right past and include extra contingency time in case an event occurs. | 5 | 6 | MEDIUM |

Risks need to me monitored throughout the project to ensure nothing goes wrong allowing the project to be completed to the right high quality as well as being completed before the deadline and within the budget.

# Design

## Database Design (Entity Relationship Diagram)

I have created a design for the database that I will be using UML on a website called diagrams.net which provides all the tools to create a diagram simply. The diagram created in an entity relationship diagram which shows the database system with all the tables that are a part of the system. It also shows each table within the database system and the attributes that each table has highlighting the primary and any foreign keys on each table. Furthermore, the relationship between tables are expressed with lines along with cardinality depicting the type of relation between tables. Here is my ERD:



It’s a very simple ERD where I have 3 tables, one for customers, one for orders and one for items. Primary keys are shown with a PK next to them and Foreign keys with a FK. From the cardinality we can see that one and only one customer can have at least one or many orders. Each order can consist of one or many items.

## Class Diagram