Software Projects with Agile Techniques

CRN: 34129

Assignment 1

George Wilkinson

@00677611

CHC119

5/11/24

Use Case Diagram

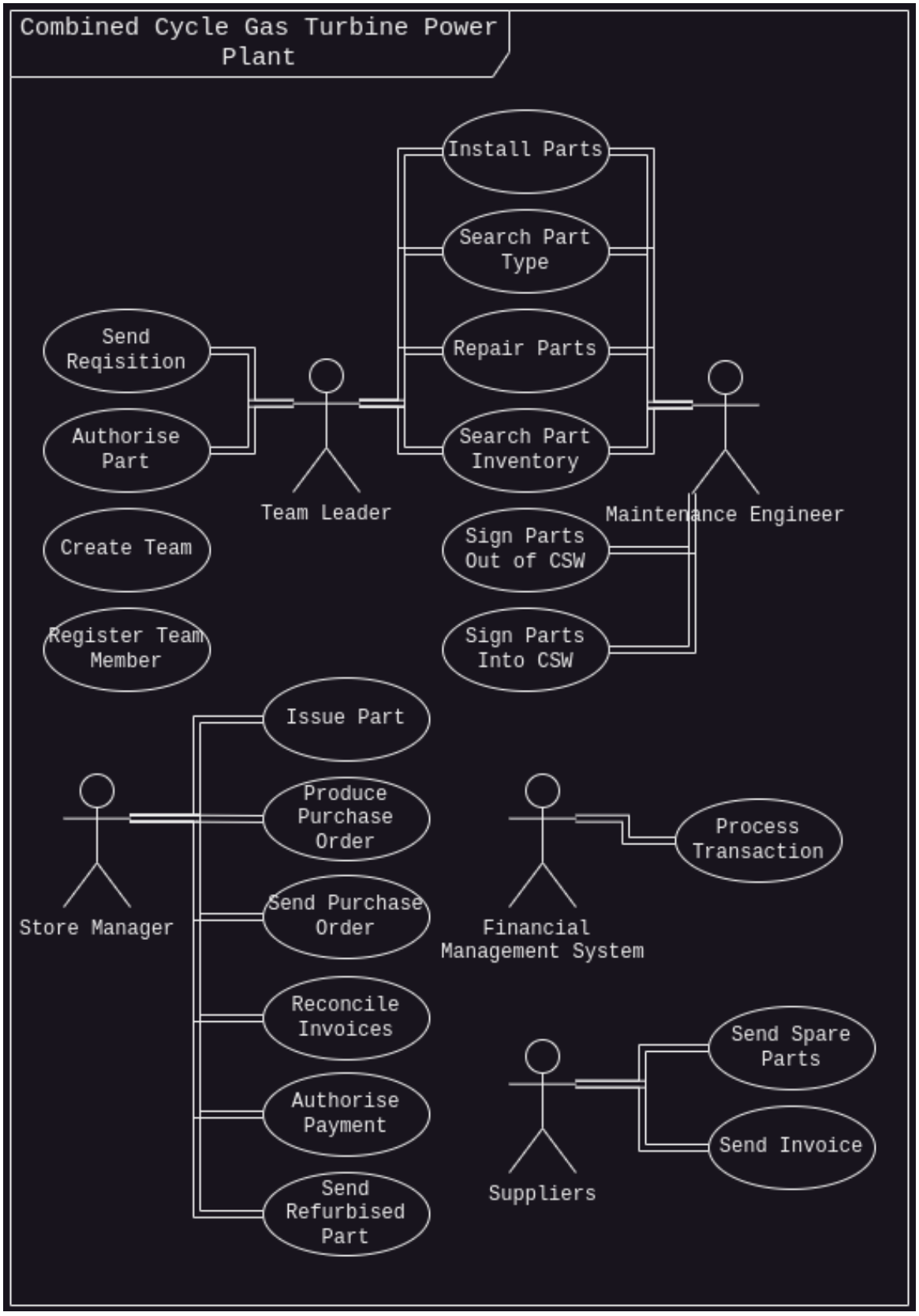

Figure 1: Created using Draw.io, a JS-based Graphical Diagramming Tool.

Textual Use Cases

For the following Textual Use Cases, I have chosen the two scenarios which show the most detail and complexity, and can be explained comprehensively.

Title	Produce Purchase Order
Scope	Requisitions and Finances
Goal	Create Purchase Order based upon a Requisition supplied by a Team Leader for a Part. Purchase Order is sent to Supplier to allow for requisition of parts. A handshake occurs using an Invoice sent from Supplier, and the Transaction sent from Store Manager.
Primary Actor(s)	Store Manager
Secondary Actor(s)	Supplier Financial Management System
Pre-conditions	Approved Purchase Order
Post-conditions	Purchase Order sent to Supplier
Main Success Scenario	1. Store Manager ensures requisition is valid. 2. Store Manager creates Purchase Order using details provided by requisition. 3. Store Manager submits Purchase Order to web-based system. 4. Store Manager sends Purchase Order to Supplier 5. Supplier reviews Purchase Order, and has the part. 6. Supplier creates Invoice containing relevant information for the Part, and the amount owed. 7. Supplier sends Invoice to Store Manager 8. Store Manager reconciles Invoices against Purchase Orders verifying validity. 9. Store Manager authorises payment. 10. FMS creates a Payment Transaction according to Invoice. 11. FMS processes Payment Transaction, crediting Supplier.

Extensions	1a. Requisition details Invalid
	  1a.1. Store Manager denies Requisition, allowing for resubmission
	3a. Web-based System Error
	  3a.1. Allow for review and resubmission of requisition by Store Manager
	4a. Supplier Details Incorrect
	  4a.1. Allow for review of Supplier details
	5a. Supplier does not have part in stock
	  5a.1. Notify Store Manager of lack of stock
	  5a.2. Notify Store Manager of delay
	8a. Invoice Invalid

	8a.1. Store Manager contacts Supplier of issues, allowing for resubmission of Invoice. 11a. Supplier details Invalid. 11a.1. Notify Store Manager of error and allow resubmission
Includes	Send Purchase Order (Included in Step 4) Send Invoice (Included in Step 7) Reconcile Invoice (Included in Step 8) Authorise Payment (Included in Step 9) Process Transaction (Included in Step 10/11)
Extends	Install Part (When Part has to be installed) Repair Part (If Part Requires Refurbishment)

Title	Installing Parts
Scope	Inventory Management and Maintenance
Goal	Maintenance Engineer Installs a part into the Power Station. Either by using a part from the CSW or via Requisition.
Primary Actor(s)	Maintenance Engineer
Secondary Actor(s)	Team Leader Store Manager
Pre-conditions	Part is needed, and maintenance required in the Power Station
Post-conditions	Part is installed into power station
Main Success Scenario	1. Maintenance Engineer determines installation of a new part is required 2. Maintenance Engineer searches Part Inventory for new Part 3. Maintenance Engineer signs the Part out of the CSW 4. Store Manager issues spare Part 5. Maintenance Engineer removes the old Part from Power Station, returning it to the CSW 6. Maintenance Engineer installs the new Part into the Power Station
Extensions	2a. Part is not in inventory 2a.1. Team Leader creates a Requisition for missing part, and sends it to the Store Manager 3a. Value of Part is >£50,000 3a.1. Team Leader authorises Part 3a.1.a. Team Leader denies authorisation 5a. Store Manager receives old part 5a.1. Store Manager disposes of part 5a.2. Store Manager processes part for refurbishment 6a. Part is faulty 6a.1 Part is returned to CSW to follow procedure of old part and Maintenance Engineer proceeds from 2.

	6b. Maintenance Engineer runs out of shift time 6b.1. Maintenance Engineer signs Part back into the CSW
Includes	Search Part Inventory (Included in Step 2) Search Part Type (Included in Step 2) Authorise Part (Included in Step 3a) Sign Part out of CSW (Included in Step 3) Issue Part (Included in Step 4) Sign Part into CSW (Included in Step 5 and 6a)
Extends	

Class Diagram

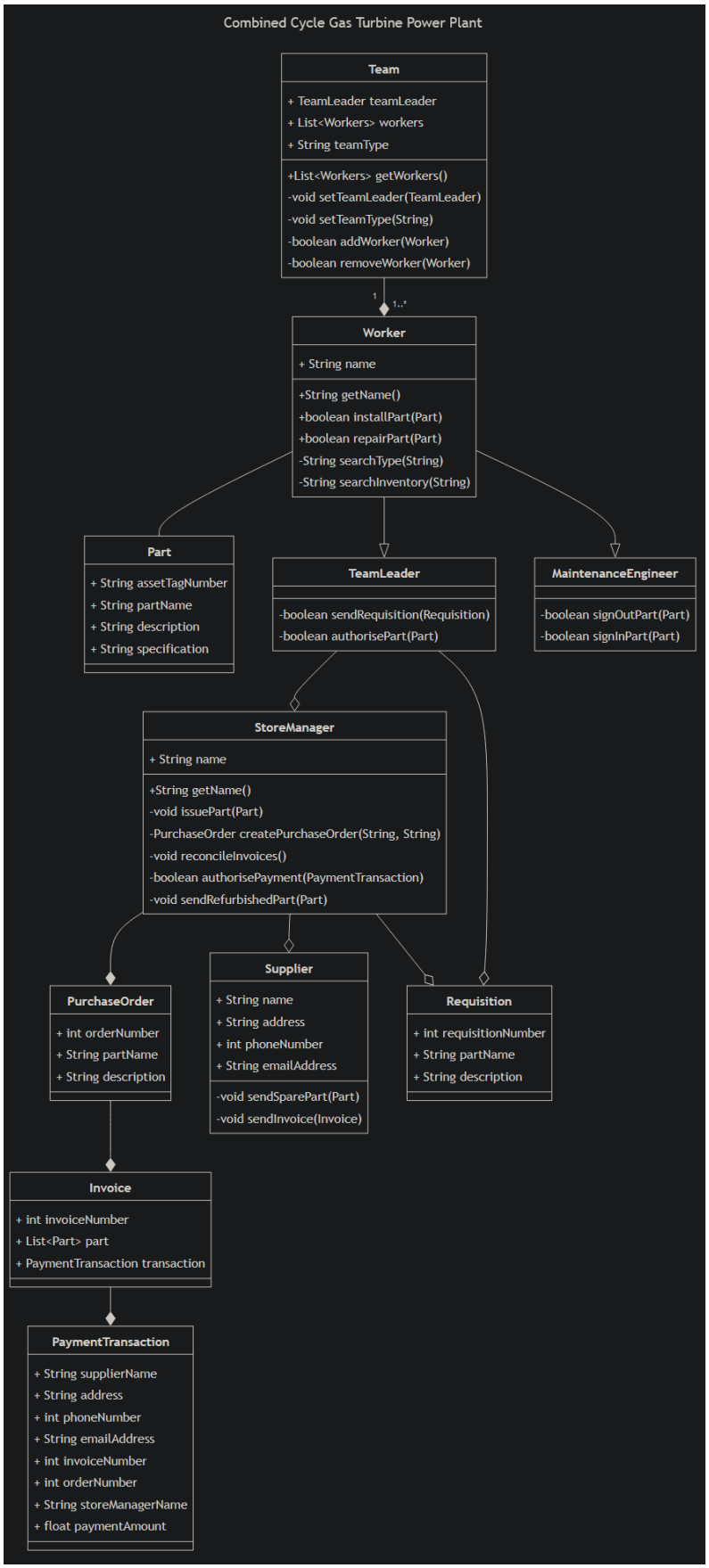

Figure 2: Created using Mermaid.js, a JS-based, Markdown Inspired Diagramming Tool.

References

Julian Michael Bass. (2023). Agile Software Engineering Skills. In *Springer eBooks*. Springer Nature. https://doi.org/10.1007/978-3-031-05469-3

Julian Bass' book on agile software engineering was of help, due to his in-detail explanation of terms, allowing me to further my understanding of how to structure my use cases. This provided valuable information on Primary and Secondary Actors and Textual Use Cases.

Visual Paradigm. (2024). *UML Class Diagram Tutorial*. Visual-Paradigm.com. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

This tutorial was helpful to learn the differing arrow types in UML Class Diagrams that were not covered in lectures and workshops previously. The standard UML arrow types were helpful to help me better explain the structure of my Class Diagram.

Visual Paradigm. (2023, October 11). *Demystifying Use Case Models: Bridging Textual Detail and Visual Insight - Visual Paradigm Guides*. Visual Paradigm Guides. https://guides.visual-paradigm.com/demystifying-use-case-models-bridging-textual-detail-and-visual-insight/

This article helped me properly structure my Textual Use Cases, specifically utilising Includes and Extends, which I did not know were a part of UML Textual Use Cases. This enabled me to have more comprehensive Textual Use Cases to better demonstrate my understanding of the assignment.

UML Class Diagrams | Thinking About Visual Basic.NET Programs | InformIT. (2020). Informit.com. https://www.informit.com/articles/article.aspx?p=98144&seqNum=6

This article provided me with information regarding Public and Private definitions in UML Class Diagrams, helping me further elaborate on my Class Diagram. This allows for methods to be private in the class definitions so that sensitive functions can't be used elsewhere.