60DEGREES Systems Solutions

(PH) 000-000-000
www.60DegreesSolutions.com

60DEGREES
SYSTEMS SOLUTIONS

Upshift

IT System Proposal for Global Bike Group

Recipient[s]: Global Bike Incorporated
Board of Directors

John
Co-CEO

Peter
Co-CEO

Producer[s]: Burak Cem Boga
Project Planner

Jade Harris
Project/Documentation Manager

William Lawler
Systems Analyst

Vithyakaran Parameshwaran
Systems Analyst

Date: 27-05-2022

Executive Summary

When business workflows rely on manual procedures, efficiency is limited by the effort and repetition required to complete manual operations, human error, and the restricting ways processes must be acted out. Global Bike Incorporated is currently relying on such procedures. As such, operations have become error-prone and tedious. 60 DEGREES SYSTEMS SOLUTIONS proposes to solve this problem by designing and implementing a custom built iDempiere information system, which will streamline the procure-to-cash and order-to-cash business processes, and reduce friction between financial accounting, materials management, marketing and customer service, product development, and sales and distribution departments.

At a high level, the solution system will allow Global Bike Incorporated employees to, from anywhere with an internet connection:

- Streamline and automate the procurement process.
- Manage and maintain inventory.
- Streamline and automate the ordering process.
- Manage vendors, customers, and financial account information.
- Create and review inquiry, invoices, and order documents.
- Access the application interface from mobile, laptop, desktop, or tablet

Our solution aims to significantly reduce time, cost, and business resources by streamlining the process of procuring new products and then distributing these goods to customers. This will enable employees to achieve greater efficiency and accuracy, maximizing business profit and improving customer satisfaction. Our team of analysts will lean on industry experience and a fresh perspective to deliver an innovative and robust solution. We are confident that the proposed solution will deliver exceptional outcomes for Global Bike Incorporated.

Table of Contents

• Executive Summary 2
• Introduction 4

Chapter I – Project Planning Documents
• System Vision Document 6
• Project Iteration Schedule 9
• Work Breakdown Structure 13
• Gantt Chart 15

Chapter II – Systems Analysis Models
• Stakeholder Analysis 18
• Feasibility Analysis 23
• Use cases and UML Use Case Diagrams 26
• UML Activity Diagrams 40
• (Fully developed) Use Case Descriptions 45
• UML Domain Model Class Diagram 66

• Conclusion 77
• References 78
• Glossary 79
• Appendices 80

Introduction

Technology has irrevocably transformed the modern business world, allowing organisations to improve the efficiency, precision, and diversity of their workflows. The productivity constraints imposed by manual procedures, such as the effort required to organise and retrieve customer records, as well as the risk of human error, are two major issues. Information solutions that automate and optimise corporate operations gain a clear competitive advantage by saving time and resources, while boosting customer service and business decisions. Furthermore, the organisation and simplification of business data can improve employee satisfaction and productivity by improving synergy across firm divisions and procedures. The implementation of a robust, comprehensive system can support a company's continuous growth in terms of profit, scale, and market relevance, bolstering the prosperity of a business.

The lasting success of Global Bike Incorporated can be traced back to its humble origins as a supplier of high-performance bicycles to prosumers interested in touring and off-road racing. 60 DEGREES believes an innovative and robust information system solution that integrates the procure-to-cash and order-to-cash processes will greatly improve the efficiency of current manual procedures, ensuring the company's long-term viability in today's digital market, and opening up new sales and strategy options. The purpose of the following report is to propose the conceptualised system through a series of systems analysis models and diagrams in order to demonstrate the value of the solution system and acquire project approval from Global Bike Incorporated.

In the sections to follow, 60 DEGREES presents a simultaneous analysis of the present manual retail system and a proposal of the solution system. Information gathered about the system has been garnered from the supplied Information Slides about Global Bike Incorporated, as well as industry white papers, and information obtained during systems analysis. The report is divided into two chapters: project planning documents, and systems analysis documents, and aims to present the report using plain terminology. Project planning documents demonstrate the objectives, schedule, and thus feasibility of project UPSHIFT. Systems analysis documents present an evaluation of stakeholders and the feasibility of implementing the new system in the current workplace before presenting representations of the new system, considering a variety of details and perspectives. An explanation for the technical purpose precedes each diagram and document, and a summation explaining the figure in terms of the proposed system follows. Finally, the conclusion discusses the future design and implementation activities following the acceptance of the solution system.

Further, to encompass the proposed solution, the project title 'UPSHIFT' incorporates the cycling term *upshift* to indicate going up a gear to emphasise and 'play-on' the importance and potential of biking 'gear', and the power offered by the proposed system.

60DEGREES
SYSTEMS SOLUTIONS

Chapter I

Project Planning Documents

• System Vision Document 6
• Project Iteration Schedule 9
• Work Breakdown Structure 13
• Gantt Chart 15

System Vision Document

DETAILED DOCUMENT EXPLANATION

An information technology system is a system of interconnected components that collect, store, process and output data required to complete business tasks. The System Vision Document is developed to showcase the overall objective of the proposed Information System, demonstrating a thorough understanding of the company's initial problem, system capabilities that the proposed system will offer, and the resulting benefits of the system to Global Bike Incorporated. This document offers a general overview of the solution system in order to obtain approval from the client. As outlined in the company's mission statement, the translation of a pragmatic design philosophy to an IT system has led to a robust, streamlined solution, that will greatly enhance efficiency and competitiveness in a digital market.

KEYWORDS

System Vision Document
A summarised document that presents the overall objective of the new system.

Information Technology System (IT system or 'system'): A set of interconnected components that collect, store, process and output data required to complete business tasks.
Problem Description: The system problem and resulting ideas for the solution.
System Capabilities: The scope and abilities of the new system.
Business Benefits: The tangible (measurable in dollars) and intangible benefits for the company.

Most notably, the UPSHIFT system solution advises the use of a custom iDempiere Enterprise Resource Planning (ERP) system to develop the project. ERP refers to a type of software that an organisation can use to plan, centralise, and manage core business activities, such as accounting, procurement, supply chain and other processes. Accounting refers to the activities related to managing financial information, such as budgeting and accessing financial accounts. Procurement refers to the activities required by an enterprise to obtain goods from a vendor. Supply chain processes refer to the activities required by an organisation to deliver goods to a customer.

As mentioned, an ERP system refers to an IT system that a company can use to plan, centralise, and manage core business functions. iDempiere is an ERP development platform that supports the development and deployment of an ERP system. It is robust and mature, being founded in 2011, and is an open-source software with no licence costs. Open-source refers to the availability of a program's code as public. In this case, iDempiere is a beneficial ERP development platform as it is supported and driven by specialists originating in different fields. In project UPSHIFT, 60DEGREES proposes implementing the system as a customised ERP system developed using iDempiere to create a strong and robust system using the platform's tools to reduce development time and costs whilst improving quality.

Additional features iDempiere offers the solution system:

- iDempiere offers prewritten enterprise-driven modules which can be 'plugged-in' and customised. Alternatively, a developer can custom-write plug-in modules, offering additional detail to the system. This enables the system to be developed using a modular approach, meaning independent segments of the code can be swapped to alter the program. Significant advantages of this approach involve a reduction in time and cost, and increased robustness in the system as the modules have been thoroughly tested independently.

- The iDempiere user interface (UI) is fully navigable on PCs, tablets and smartphones offering flexibility and a robust UI that has been developed by specialists within the enterprise industry. UI refers to the point a user can interface with the system, such as screens, keyboards and microphones. The UI is critical to the success and efficiency of an information system.

KEYWORDS

iDempiere (ERP Development Platform)
An open-source system used for developing and deploying ERP systems.

Enterprise Resource Planning (ERP): A type of software or system that an organisation can use to plan, centralise, and manage core business activities.
Open-Source: Availability of a program's code as public, oftentimes facilitating community collaboration from industry specialists.
User Interface (UI): The points at which users can interface with the system, such as screens, keyboards and microphones.
Benefits of a custom iDempiere system:

- Reduced development time and cost, as pre-made and pre-tested modules (segments of code) can be plugged-in to customize the system. Further, custom plug-ins can also be developed.
- Versatile UI as the iDempiere system is fully navigable on PCs, tablets, and smartphones

SYSTEMS VISION DOCUMENT | DOCUMENTATION

Problem Description
In the changing modern age, high-performance cycling equipment has remained essential to the success of the most demanding competitors in both touring and off-road racing. It is critical for Global Bike Incorporated to have a streamlined system that increases the synergy between Materials Management, Financial Accounting and Sales and Distribution departments. Additionally, if the procure-to-cash and order-to-cash operations can be digitised and automated, then the system can obtain greater efficiency, accuracy and versatility, facilitating a more competitive position within today's digital market. It is also important to organise vendor and customer information electronically in order to send targeted communications to business stakeholders, improving corporate relationships.

It is recommended that a new system be developed and deployed so Global Bike Incorporated can more readily supply customers with distinctive and innovative biking equipment. This system should be deployed online, and developed using a custom iDempiere ERP system.

System Capabilities
The new system will be capable of:

- Collecting and storing information about the vendor
- Collecting and storing information about each customer
- Managing information about products and their stock status
- Automating invoices, payments, and receipts to the vendor
- Automating quotes, invoices, and receipts to customer
- Managing General Ledger (G/L) financial account balances and associated expenditures
- Automating vendor purchase orders and their Request for Proposals (RFP)
- Automating sales order referencing a quotation
- Connecting via Wi-Fi (Internet) and transmitting data

Business Benefits

It is anticipated that the deployment of the new system will provide the following benefits to the Global Bike Group:

- Increase connection with an international market (US and Germany), thereby enabling greater sales and company product outreach.
- Maintain correct and current information about vendors and their products, thereby facilitating rapid and deep communication, and catching trends more rapidly.
- Maintain correct and rapid information and images about new products and accessories, thereby facilitating an enhanced customer experience and thus improving the quality and speed of sales.
- Maintain correct and current information about customers and their personal preferences, thereby facilitating greater customer engagement and thus improving sales.
- Centralise procure-to-order process and business financial accounts, thereby improving accuracy and efficiency for customer and management personnel, and thus facilitating more rapid sales.
- Reduce menial tasks by automation through the procure to order process, thereby improving efficiency for customer and management personnel, and thus facilitating greater sales and enhancing competitiveness in a digital market.
- Automated insight and management of G/L balance, thereby improving the quality and speed of vendor expenditures and associated business decisions.

60DEGREES
SYSTEMS SOLUTIONS

Project Iteration Schedule

DETAILED DOCUMENT EXPLANATION

Iterative development is a methodology for system development that breaks the project into several iterations, each producing a deliverable that gradually builds the final system. An iteration is a small system development project. This approach allows feedback to be incorporated frequently and new requirements to be prioritised and scheduled, producing a refined and accurate solution. The UPSHIFT solution system has been further decomposed into subsystems. The process of dividing the system allows the complexity of the project to be handled logically (**APPENDIX C: Dividing the system into subsystems) .**

The Project Iteration Schedule is a list of iterations for each subsystem, with use cases assigned to each iteration. A use case is an activity that the system performs often in response to a user's request or an event that is triggered. A time estimate is also included, considering the total scope and configuration of the system, and the personnel available to work on the project.

KEYWORDS

Project Iteration Schedule
A list of iterations for each subsystem, with use cases assigned to each iteration and a time estimate.

Project: A planned undertaking of tasks with a beginning and an end to reach an objective
Use Case: An activity the system performs often in response to a user's request.
Subsystem: An identifiable and fully functional part of a complete system.
Iteration: A system development mini-project that often produces a deliverable at its completion, building a system through multiple iterations.

The Order Fulfilment Subsystem
This subsystem will perform the tasks of creating and fulfilling a sale from the point of sales inquiry to the delivery of a product. It is a complex system, and thus the use cases are grouped into customer inquiry (moderately sized iteration), sales order and quote (moderately sized iteration), delivery and inventory (significant iteration), and reviewing document flow (complex iteration) to ensure the order has correctly been processed. Finally, there is a week to refine and integrate the iteration (significant iteration).

Iteration	Time estimate	Use cases assigned to iteration
1	3 weeks	1. Create customer inquiry 2. View customer inquiry
2	3 weeks	3. Create customer quotation 4. Create sales order referencing a quotation 5. Display sales order
3	4 weeks	6. Start delivery process 7. Pick materials on delivery note 8. Post goods issue 9. Check stock status

The Order Fulfilment Subsystem (Continued)

Iteration	Time estimate	Use cases assigned to iteration
4	1. weeks	10. Review document flow
5	2 weeks	11. Clean up, final test, harden site, tune database and other final steps
Total	15 weeks	

The Customer Account Subsystem

This subsystem will provide services to manage and enhance the customer experience. The use cases are grouped into customer management functions (significant iteration) and the search function (moderately sized iteration). Finally, there is a week to refine and integrate the iteration (significant iteration).

Iteration	Time estimate	Use cases assigned to iteration
1	4 weeks	1. Create new customer 2. Create contact person for customer 3. Update customer account 4. Remove a customer
2	1 week	5. Search for a customer
3	2 weeks	6. Clean up, final test, harden site, tune database and other final steps
Total	7 weeks	

The Supply Chain Subsystem

This subsystem is responsible for managing inventory levels and adding products, as well as managing the vendors that order the products. It is a complex system, and thus the use cases are grouped into purchase order (significant iteration), vendor management (significant iteration), delivery and inventory (significant iteration), and reviewing document flow (complex iteration) to ensure the order has correctly been processed. Finally, there is a week to refine and integrate the iteration (significant iteration).

Iteration	Time estimate	Use cases assigned to iteration
1	4 weeks	1. Create purchase order referencing a RFQ 2. Display purchase order 3. Verify physical receipt of goods
2	4 weeks	4. Add a new vendor 5. View vendor details 6. Update a vendor 7. Remove a vendor 8. Search for a vendor

The Supply Chain Subsystem (Continued)

Iteration	Time estimate	Use cases assigned to iteration
3	2 weeks	12. Verify physical receipt of goods 13. Display vendor line items
4	2 week	14. Clean up, final test, harden site, tune database and other final steps
Total	12 weeks	

The Marketing Subsystem

This subsystem will provide services to manage and enhance the customer product experience. The use cases are grouped into product management functions (significant iteration) and the search function (moderately sized iteration). Finally, there is a week to refine and integrate the iteration (moderate iteration).

Iteration	Time estimate	Use cases assigned to iteration
1	4 weeks	1. Add new product 2. View product details 3. Update a product 4. Remove a product
2	1 week	5. Search for a product
3	1 week	6. Clean up, final test, harden site, tune database and other final steps
Total	7 weeks	

The Financial Accounting Subsystem

This subsystem is responsible for providing functions to manage and review financial information. It is a complex system, and thus the use cases are grouped into vendor payment (significant iteration), customer payment (significant iteration), financial account management (significant iteration), account balances (minor iteration). Finally, there is a week to refine and integrate the iteration (significant iteration).

Iteration	Time estimate	Use cases assigned to iteration
1	4 weeks	1. Post payments to vendor 2. Create invoice receipt from vendor 3. Create goods receipt for purchase order
2	4 weeks	4. Create invoice for customer 5. Display billing document and customer invoice 6. Post receipt of customer payment

The Financial Accounting Subsystem (Continued)

Iteration	Time estimate	Use cases assigned to iteration
3	4 weeks	7. Add a new financial account 8. View financial account details 9. Update a financial account 10. Remove a financial account 11. Search a financial account
4	1 weeks	12. Display account balances
5	2 week	13. Clean up, final test, harden site, tune database and other final steps
Total	15 weeks	

60DEGREES
SYSTEMS SOLUTIONS

Work Breakdown Structure

DETAILED DOCUMENT EXPLANATION

The Work Breakdown Structure (WBS) identifies and compiles an organised list of individual tasks that must be completed. The document also includes an estimated (**' ~ '**) time for completion of each task, considering project factors such as the internal members of the development team and the complexity of the project. The following diagram compiles a list for the first iteration of the project: planning and analysis.

KEYWORDS

Work Breakdown Structure
An organised list of individual tasks that must be complete for an iteration

' ~ time ': Estimated amount of time that the task will take

WORK BREAKDOWN STRUCTURE | DOCUMENTATION

1. **Initiate the project.**
   - 1.1 Form the project team~ 4 days
   - 1.2 Identify the stakeholders ~ 4 days
   - 1.3 Identify the problem and document the objective of the solution system (System Vision Document) ~ 9 days
   - 1.4 Determine project risks and feasibility (Feasibility analysis) ~ 7 days

2. **Plan and organise the project.**
   - 2.1 Decide roles for each member < 1 day.
   - 2.2 Create WBS ~ 4 days.
   - 2.3 Assign people to tasks < 1 day.
   - 2.4 Create Gantt chart ~ 1 day
   - 2.5 Determine the major components (functional areas) that are needed (divide the system into several subsystems) ~ 10 days.
   - 2.6 Define the iterations and assign each functional area to an iteration. ~ 12 days.

3. **Discover and understand project details.**
   - 3.1 Identify people who have an interest in the project (Stakeholder analysis) ~ 12 days.
   - 3.2 Do fact-finding tasks to understand the requirements ~ 7 days.
   - 3.3 Develop a list of use cases and a use case diagram ~ 5 days.

4. **Analyse in detail the use cases and domain classes for each iteration.**
   - 4.1 Perform in-depth fact finding to understand details of each use case (Fully developed use case descriptions) ~ 8 days.
   - 4.2 Understand and document the detailed workflow of each use case (use case diagrams and activity diagrams) ~ 5 days.
   - 4.3 Develop a list of classes and a class diagram (UML Domain Model Class Diagram) ~ 7 days.

5. **Refine the solution**
   - 5.1 Refine requirements and findings ~ 18 days.
   - 5.2 Refine models and documents ~ 18 days.

6. **Propose the solution.**
   - 6.1 Meet with key stakeholders and propose solution < 1 day.

WORK BREAKDOWN STRUCTURE | DETAILED DOCUMENT EXPLANATION

The first task that must be completed for the analysis iteration is to initate the project: forming the project team and beginning to understand the project requirements by identifying the stakeholders. Further insight can be developed through creating the System Vision Document and conducting feasibility analysis. After these preliminary investigation tasks have occurred and a grasp on the project problem is gained, the project team can begin to organise accordingly. The roles for each member and assigning members to tasks will be completed in less than a day, while planning activities like the WBS and Gantt chart take several days. Next, the proposed solution will be divided into subsystems to gain insight into our proposal. Finally, the project iteration schedule will be created, concluding the planning activities. Utilising the information gained from identifying the stakeholders, stakeholder analysis and interviews is estimated to take a significant amount of time. Other fact-finding tasks will be used to understand the requirements and a list of use cases and diagrams will be produced to gain further insight into the proposal. Next, in-depth details of each use case, work flow and domain classes will be produced for the solution, taking a 20 days. Finally a the solution will be refined completely, and lastly the solution will be proposed.

Gantt Chart

DETAILED DOCUMENT EXPLANATION

Following the Work Breakdown Schedule (WBS) completion, the schedule is graphically presented to expand on the tasks with further and more formal details. The Gantt Chart is a bar chart that graphically portrays the tasks as horizontal bars superimposed onto a calendar to represent the project schedule. Tasks from the WBS are listed in the Task Name column, and expanded on with their dependency tasks, and the resources assigned them, often development team members. Dependency tasks refer to tasks that must occur before a specific task can be complete. These tasks are represented graphically on the calendar. The Gantt Chart also illustrates a critical path using red-coloured horizontal bars to represent tasks that must remain on schedule to prevent the project being delayed. This offers insight into the tasks that must be monitored closely and accurately.

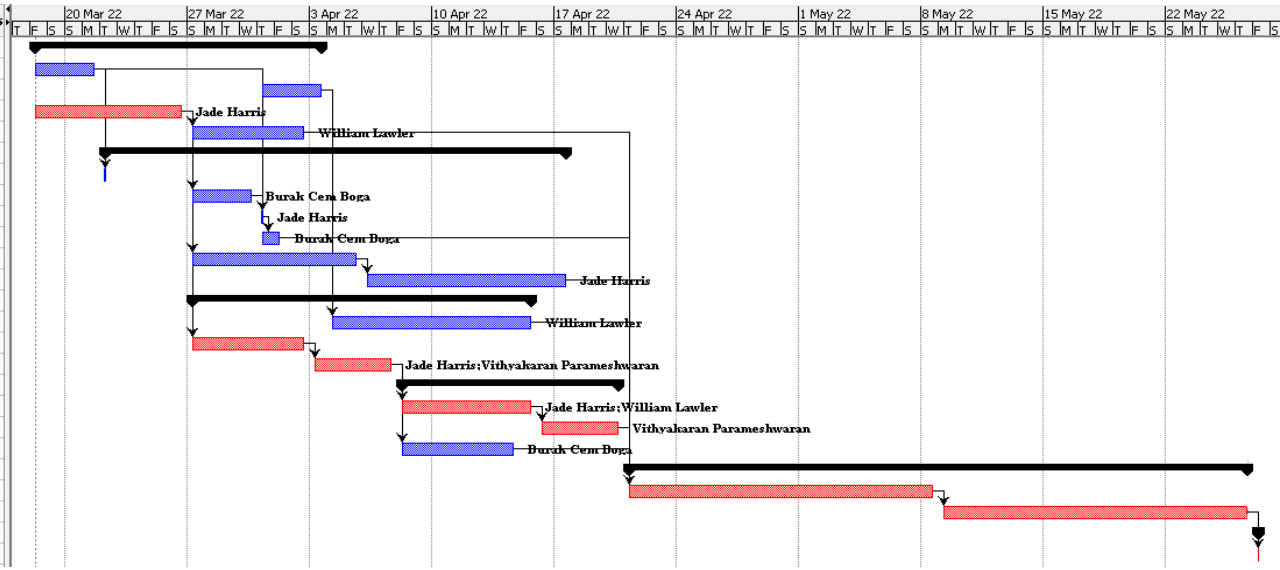
KEYWORDS

Gantt Chart
A bar chart that graphically portrays the tasks as horizontal bars superimposed onto a calendar to represent the project schedule.

Critical path: Red-coloured horizontal bars to represent the tasks that must remain on schedule to prevent the project being delayed
Predecessors/Dependency tasks: Tasks that must occur before a specific task can be complete
Resources: Resources, often team members, assigned to achieve the completion of the task.

GANTT CHART | DOCUMENTATION

	Name	Duration	Start	Finish	Predecessors	Resource Initials
1	**Initiate the project**	17 days	18/03/22 8:00 AM	3/04/22 5:00 PM		
2	Form the project team	4 days	18/03/22 8:00 AM	21/03/22 5:00 PM		
3	Stakeholder identification	4 days	31/03/22 8:00 AM	3/04/22 5:00 PM		
4	Create System Vision Document	9 days	18/03/22 8:00 AM	26/03/22 5:00 PM		Jade
5	Perform Feasibility analysis	7 days	27/03/22 8:00 AM	2/04/22 5:00 PM	4	Will
6	**Plan and organise**	27 days	22/03/22 8:00 AM	17/04/22 5:00 PM		
7	Decide the roles for each member	0.031 days	22/03/22 8:00 AM	22/03/22 8:15 AM	2	
8	Create WBS	4 days	27/03/22 8:00 AM	30/03/22 5:00 PM	4	Gem
9	Assign people to tasks	0.042 days	31/03/22 8:00 AM	31/03/22 8:20 AM	2;8	Jade
10	Create Gantt chart	1 day	31/03/22 8:20 AM	1/04/22 8:20 AM	9	Gem
11	Divide the system into several subsystems	10 days	27/03/22 8:00 AM	5/04/22 5:00 PM	4	
12	Create Project Iteration Schedule	12 days	6/04/22 8:00 AM	17/04/22 5:00 PM	11	Jade
13	**Discover and understand project details**	20 days	27/03/22 8:00 AM	15/04/22 5:00 PM		
14	Stakeholder analysis	12 days	4/04/22 8:00 AM	15/04/22 5:00 PM	3	Will
15	Do fact-finding tasks to understand the requirements	7 days	27/03/22 8:00 AM	2/04/22 5:00 PM	4	
16	Develop a list of use cases and use case diagrams	5 days	3/04/22 8:00 AM	7/04/22 5:00 PM	15	Jade;Vith
17	**Analyse in detail the use cases and domain classes**	13 days	8/04/22 8:00 AM	20/04/22 5:00 PM		
18	Fully developed use case descriptions	8 days	8/04/22 8:00 AM	15/04/22 5:00 PM	16	Jade;Will
19	Activity diagrams	5 days	16/04/22 8:00 AM	20/04/22 5:00 PM	18	Vith
20	UML Domain Model Class Diagram	7 days	8/04/22 8:00 AM	14/04/22 5:00 PM	16	Gem
21	**Refine the solution**	36 days	21/04/22 8:00 AM	26/05/22 5:00 PM		
22	Refine requirements and findings	18 days	21/04/22 8:00 AM	8/05/22 5:00 PM	5;10;12;14;19;20	
23	Refine the models and documents	18 days	9/05/22 8:00 AM	26/05/22 5:00 PM	22	
24	**Milestone: Propose the solution**	0.25 days	27/05/22 8:00 AM	27/05/22 10:00 ...		
25	Meet with the stakeholders and propose the solution	0.25 days	27/05/22 8:00 AM	27/05/22 10:00 AM	23	



GANTT CHART |DETAILED DOCUMENT EXPLANATION

Expanding on the work breakdown structure, the Gantt chart contains the tasks and their durations. Generally, one team member was sufficient to allocate to each task, allowing many tasks to be completed simultaneously and thus with greater efficiency. Due to the scope of *Developing a list of use cases and use case diagrams* for each subsystem, and creating 10 detailed *Fully developed use case descriptions*, the resources allocated were increased allowing them to be completed faster, and thus their predecessor tasks could be started sooner. Additionally, by planning the *Refine the requirements and findings*, and *Refine the models and documents* to be critical tasks that must be completed on time, our schedule can gain greater flexibility, margin for error and the opportunity to expand on additional features.

60DEGREES
SYSTEMS SOLUTIONS

Chapter II

Systems Analysis Documents

• Stakeholder Analysis 18
• Feasibility Analysis 23
• Use cases and UML Use Case Diagrams 26
• UML Activity Diagrams 40
• (Fully Developed) Use Case Descriptions 45
• UML Domain Model Class Diagram 66

Stakeholder Analysis

DETAILED DOCUMENT EXPLANATION

Stakeholders refers to the personnel who have an interest in the successful development of the new system, such as the clients who may fund the project, and users who will use the system regularly. A critical factor for the proposed system's success is to consider and structure the project in a way that facilitates accurate communication with Global Bike Incorporated's executives and other key stakeholders of the project. Stakeholder analysis aims to identify all the personnel with interest in the project, and by analysing their power and interest, gain insight into management techniques for communication and satisfaction. This classification is documented on a power-interest grid in the diagram **FIGURE 1: Stakeholder Analysis.** Further information regarding the power-interest grid framework can be found in **APPENDIX A: Power-interest Grid.**

To support the power-interest identification of stakeholders, a further table (**APPENDIX B: Alternative Stakeholder Analysis Table Format**) expands on the analysis and classification of stakeholders in the power-interest grid. This is presented in the subsequent diagram **FIGURE 2: Stakeholder Analysis Elaboration**, which also provides a detailed document explanation to the stakeholder analysis on the power-interest grid.

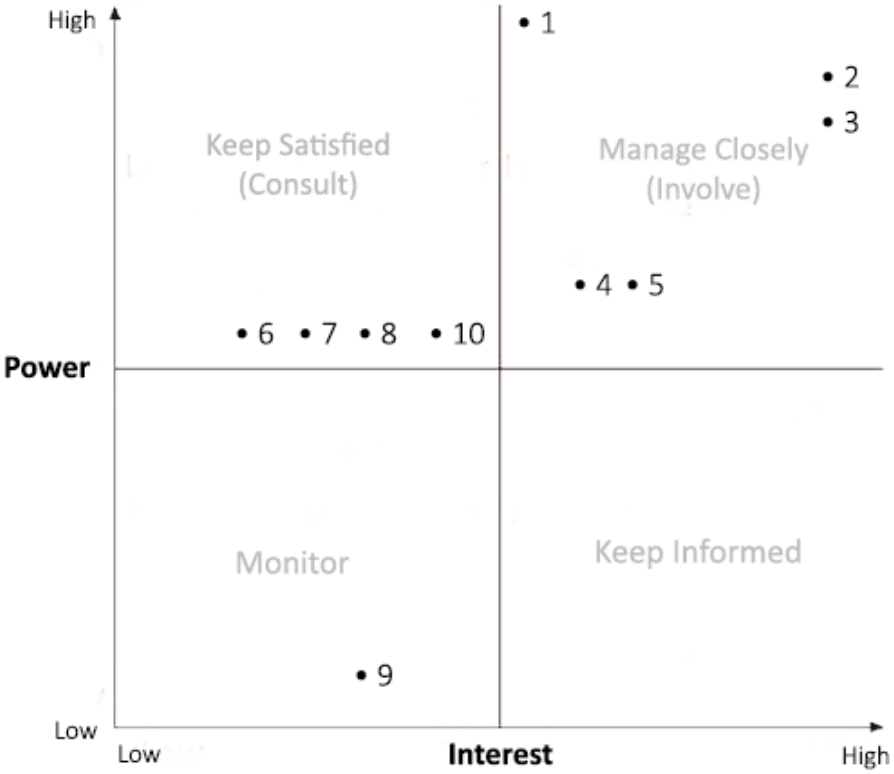
KEYWORDS

Stakeholder Analysis
A grid that aims to identify all the personnel with interest in the project, and by analysing their power and interest, gain insight into management techniques for communication and stakeholder satisfaction.

Stakeholders: Personnel who have an interest in the successful development of the new system.
Power: A metric for classifying stakeholders based on their influence and often financial investment in the project.
Interest: A metric for classifying stakeholders based on their investment into the success of the project.
Stakeholder type:

- **Internal:** Personnel within the organisation, such as senior managers.
- **External:** Personnel outside the organisation's control and influence, such as vendors.
- **Executive:** Personnel who do not interact directly with the system however use information output by the system (such as senior managers), or have significant financial interest such as the Board of Directors (governing body of an enterprise elected to oversee the activities of an organization).
- **Operational:** Personnel who regularly interact with the system to complete work tasks.

FIGURE 1: Stakeholder Analysis



1. Board of directors
2. John
3. Peter
4. VP Marketing, Chief Information Officer, Chief Financial Officer, VP Human Resources
5. VP Research and Development, VP Operations
6. Materials Management Department
7. Financial Accounting Department
8. Sales and Marketing
9. Vendors (US and Germany)
10. Customers (US and Germany)

FIGURE 2: Stakeholder Analysis Elaboration

Name	Power/Interest	Stakeholder Type	Potential Management Strategies
1 - Board of Directors (BoD)	High / moderate	Internal executive	As the governing committee that supervises the activities of the Global Bike Group, the BoD is critical to the success of the solution. Given this stakeholder's moderate interest as senior managers, and their power in funding the project, management strategies involve: **Manage closely** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Report project status, Regularly update stakeholder through periodic meetings and implement feedback
2 - John	High / high	Internal executive	In a senior management position, this stakeholder inherits Marketing, Information, Finance, and Human Resources stakeholders. Given the high involvement and interest of the CEO, and the power of this internal executive stakeholder to the project's success, management strategies involve **Manage closely** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Report project status, Regularly consult stakeholder through periodic meetings and implement feedback
3 - Peter	High / high	Internal executive	In a senior management position, this stakeholder inherits Development and Research, and Operations stakeholders. Given the high involvement and interest of the CEO, and the power of this internal executive stakeholder to the project's success, management strategies involve **Manage closely** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Report project status, regularly consult stakeholder through periodic meetings and implement feedback

Name	Power/Interest	Stakeholder Type	Potential Management Strategies
4 - VP Marketing, Chief Information Officer, Chief Financial Officer, VP Human Resources	Significant / significant	Leading / internal executive	Given the significant power of these senior managers, in particular the CFO who generally assigns funds for the project, and the significant interest of these senior managers whose departments are affected by the system, management strategies involve: **Manage closely** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Report project status, regularly consult stakeholder through periodic meetings and implement feedback
5 - VP Research and Development, VP Operations	Significant / significant	Leading / internal executive	Aligning with the above assessment, these senior managers are positioned with similar power and slightly more interest as the VP of Operations is involved in the implementation of effective systems, while the VP of Research and Development explores such future ventures. Given this interest, management strategies involve: **Manage closely** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Report project status, regularly consult stakeholder through periodic meetings and implement feedback
6 - Materials Management **7 -** Financial Accounting **8 -** Sales and Marketing	Significant / moderate	Reserved / internal operational	Given the operational level of these departments as stakeholders, they would have moderate interest in the success of this project, and moderate power as end users of the system. Management strategies involve: **Keep Satisfied (Consult)** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Regularly consult stakeholders for clarity in system requirements through interviews, questionnaires, stakeholder feedback and work process observation

Name	Power/Interest	Stakeholder	Potential Management Strategies
9 - Vendors (US and Germany)	Low / low	External operational	Given the external operational level of vendors, this stakeholder would have minimal interest in the operational success of this project as least affected actors, and due to this minimal involvement also retain minimal power. Management strategies involve: **Monitor** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Maintain open communications for stakeholder feedback, and ensure the new system retains efficiency
10 - Customers (US and Germany)	Significant / moderate	External operational	Given the new system's platform utilising the Internet, the scope for this stakeholder may be increased to external stakeholders beyond the organisation's US and Germany business partners. Regardless, the critical role of this end user in the system's success through sales indicates their significant power and moderate interest. Thus, management strategies involve: **Keep Satisfied (Consult)** (refer to FIGURE 1) In order to facilitate this approach, 60DEGREES will: Regularly consult stakeholder for clarity in system requirements through interviews, questionnaires, and user feedback.

60DEGREES
SYSTEMS SOLUTIONS

Feasibility Analysis

DETAILED DOCUMENT EXPLANATION

Project risk and feasibility analysis ascertains whether a project can be started and accomplished successfully by identifying and assessing the potential risks to the project's success. The objective of this activity is to address and eliminate these risks, presenting a comprehensive solution that Global Bike Incorporated can feel confident in. 60 DEGREES also establishes plans and procedures to ensure these risks do not interfere with the success of the project.

Feasibility analysis is often decomposed into four categories to evaluate risks and feasibility (Source: Satzinger, Jackson, R & Burd (2016)):

1. Organisational risks and feasibility: Evaluate and resolve organisational and cultural risks that may arise within Global Bike Incorporated to enhance employee enthusiasm and commitment.
2. Technological risks and feasibility: Evaluate and resolve proposed technological requirements and available expertise for implementation.
3. Resource risks and feasibility: Evaluate and resolve the availability of resources including required personnel, computer resources and physical facilities.
4. Schedule risks and feasibility: Evaluate and resolve the uncertainties regarding system scope, potential deadline, availability of team members and schedule slippage, and consider contingency plans.

KEYWORDS

Feasibility Analysis
Ascertains whether a project can be started and accomplished successfully by identifying, evaluating, and managing the potential risks to the project success.

Organisational risks and feasibility: Organizational and cultural risks.
Technological risks and feasibility: Technological requirements and available expertise.
Resource risks and feasibility: Availability of resources including personnel.
Schedule risks and feasibility: Project scope, deadlines, availability of team members and schedule slippage

FEASIBILITY ANALYSIS | DOCUMENTATION

Organisational Risks and Feasibility
Presently, the process of Global Bike Incorporated are done manually. Automation of these processes could employee fears could arise surrounding loss of employment through the automation of business tasks which could severely affect organizational culture. By incorporating employee perspective consistently throughout the development process and implementing thorough training programs, this risk can be greatly reduced. This will allow employees to embrace the enhanced response time and assistance with their regular work lives, shifting the organization structure positively. Additionally, the proposed system will primarily simplify monotonous tasks to facilitate agency and innovation. Therefore, providing that significant support and understanding is offered to system users, the organization feasibility of the project is very good.

The below risks and procedures have been considered to ameliorate organisational risks:

- Computer phobia
    - Employee training
- Perceived loss of control on the part of staff or management

Organisational Risks and Feasibility (continued):

- Perceived loss of control on the part of staff or management
    - Involve employees in implementation of new system, employee training
- Potential shifting of political and organizational power due to new system
    - Ensure the solution is targeting synergy between departments rather than replacing them with automated system, incorporate diversity in staff roles involved in system implementation
- Fear of change of job responsibilities
    - Involve employees in implementation of new system, employee training
- Reversal of long-standing work procedures
    - Involve employees in implementation of new system, employee training
- Fear of loss of employment due to increased automation
    - Involve employees in implementation of new system, employee training

Technological Risks and Feasibility

The technological requirements of the system are as follows:

- Laptop, tablet, phone, or device compatible with iDempiere ERP system
- Internet to process communicate with the online application and external database server in proper response times
- Sufficient hardware to process and display system output.
- Development equipment and expertise for the iDempiere platform
- Deployment equipment and expertise for the iDempiere platform.

The below risks and procedures have been considered to ameliorate technological risks:

- Available expertise for web-development and use of iDempiere
    - Hire consultants or outsource to experts experienced in the iDempiere code base to write custom plugins. Alternatively train the development team, especially given the open-source development of iDempiere, many online resources and research could be available.
- Portability of the web application between device interfaces
    - Given the ERP-development-driven purpose of iDempiere, the custom system is intended to be used online, facilitating portability between device interfaces. However, tablet-size screens are selected for optimal interface usability. Alternatively, development could be outsourced, or the team could be trained to utilise the iDempiere API to create more complex mobile-friendly systems
- Available technological requirements for team to shift from manual processes to digital procedures
    - Additional training, available resources, intuitive interface, involve employees in the design of the new system
- Equipment for running the application (computer, tablet, phone)
    - Computer systems are often a one-time initial purchase, and tablet-sized devices could offer warehouse staff flexibility in where they work. Additionally, personal computers with access to the internet could potentially be used to access to the system
    - Generally, internet access is available, otherwise enterprise packages are often offered

Given the robust, affordable application iDempiere offers the solution system, the technological feasibility of the project is greatly enhanced. As the user interface (UI) is fully navigable on PCs, tablets and smartphones, the equipment for employees to use can be more diverse. This could allow Materials Management workers to use specialised tablet devices for reduced weight. Therefore, following the initial outlay to source sufficient equipment, the technological feasibility of the project is viable to a significant extent.

Resource Risks and Feasibility

The below risks and procedures have been considered to ameliorate resource risks:

- Skilled members become unavailable to the team
    - Hire new consultants or outsource plugin development; train the development team in the iDempiere code base
    - Additionally, iDempiere offers general industry plugins
- Computer resources
    - Application's online browser-based platform will offer employees diversity in the devices they utilise to operate the system
    - Using an optimised online system may reduce processing resources
    - In terms of plugin development, a full development environment is required including Eclipse and Java Java Development Kit (JDK), however these can be used with an adequate system thus are not excessively costly. For general customisation, the iDempiere UI can be used.
- Physical facilities
    - The application's online browser-based platform would allow for easier use and allow employees to swap between different devices using only their login credentials.
    - Servers to run and allow for smooth traffic flow and to protect against packet drop.
- Support staff
    - Global Bike Incorporated may need to invest in a Technical Support department
    - However, the internet can greatly facilitate communication and as iDempiere is an open source program, numerous online resources can solve technical issues

In terms of financial resources, as iDempiere is a free open-source software, it significantly improves the resource feasibility of the project further. Therefore overall, the resource feasibility of the project is viable to a significant extent.

Schedule Risks and Feasibility

The below procedures have been considered to ameliorate schedule risks:

- Scope
    - Monitor use case list and maintain an open items list to remedy scope creep
    - Given the use of iDempiere to create the system in an ERP standardised industry way, the scope for this project should be manageable, allowing other components to be focused on
- Upper management Imposes deadline
    - Through the use of iDempiere, a robust program can be implemented then iteratively expanded.
- Schedule slippage
    - Through the emphasis on an iterative methodology in our project iteration schedule, schedule slippage can be managed by reducing the scope for particular iterations.

In terms of schedule, derived from the **Gantt Chart** analysis of the system takes approximately 8 weeks. Summing the total weeks estimated for the project derived from the **Project Iteration Schedule**, implementation of the system takes approximately 56 weeks. By estimating these time frames considering critical tasks, environmental conditions and realistically, the schedule feasibility of the project is significant.

Use Cases and UML Use Case Diagrams

DETAILED DOCUMENT EXPLANATION

As defined in the **Project Iteration Schedule** document in this report, a use case is an activity the system performs often in response to a user's request. The following documents identify the use cases for each subsystem as well as their actors and a brief use case description. An actor refers to personnel or an external entity that interacts with the use case by supplying or receiving data. A brief use case description is a text-based model that describes the processing details for the given use case, typically used for simple use cases or to represent complex use cases briefly.

Additionally, a technique called Create Read Update Delete (CRUD) is used to ensure that sufficient use cases exist to add, view, amend and remove particular pieces of information, such as the customer or a product. By ensuring a use case exists for each of these actions, the solution system can correctly maintain its data.

Unified Modelling Language (UML) is a set of standards and notation for graphical models produced in system development. The use of UML increases the accuracy and efficiency of system development and documentation, and reduces the likelihood of miscommunication.

The use case diagram is a UML model that is used to illustrate use cases and their relationships to actors. The diagram represents each use case in an oval. The diagram also contains an automation boundary, which represents the border between the computerised portion of the application with the actors who operate the application as a rectangle containing the use cases. A single line connects actors to use cases, to graphically represent which use cases they are involved in. The use case diagram can be categorised by actor, listing all the use cases a particular actor interacts with, or by the use cases involved in each subsystem. Given 60 DEGREES efficient approach of decomposing the system into subsystems, the latter is presented in the following diagrams.

KEYWORDS

Use cases
A list of activities each subsystem performs often in response to a user's request.
UML use case diagrams
Unified Modelling Language (UML) model that is used to illustrate use cases and their relationships to actors.

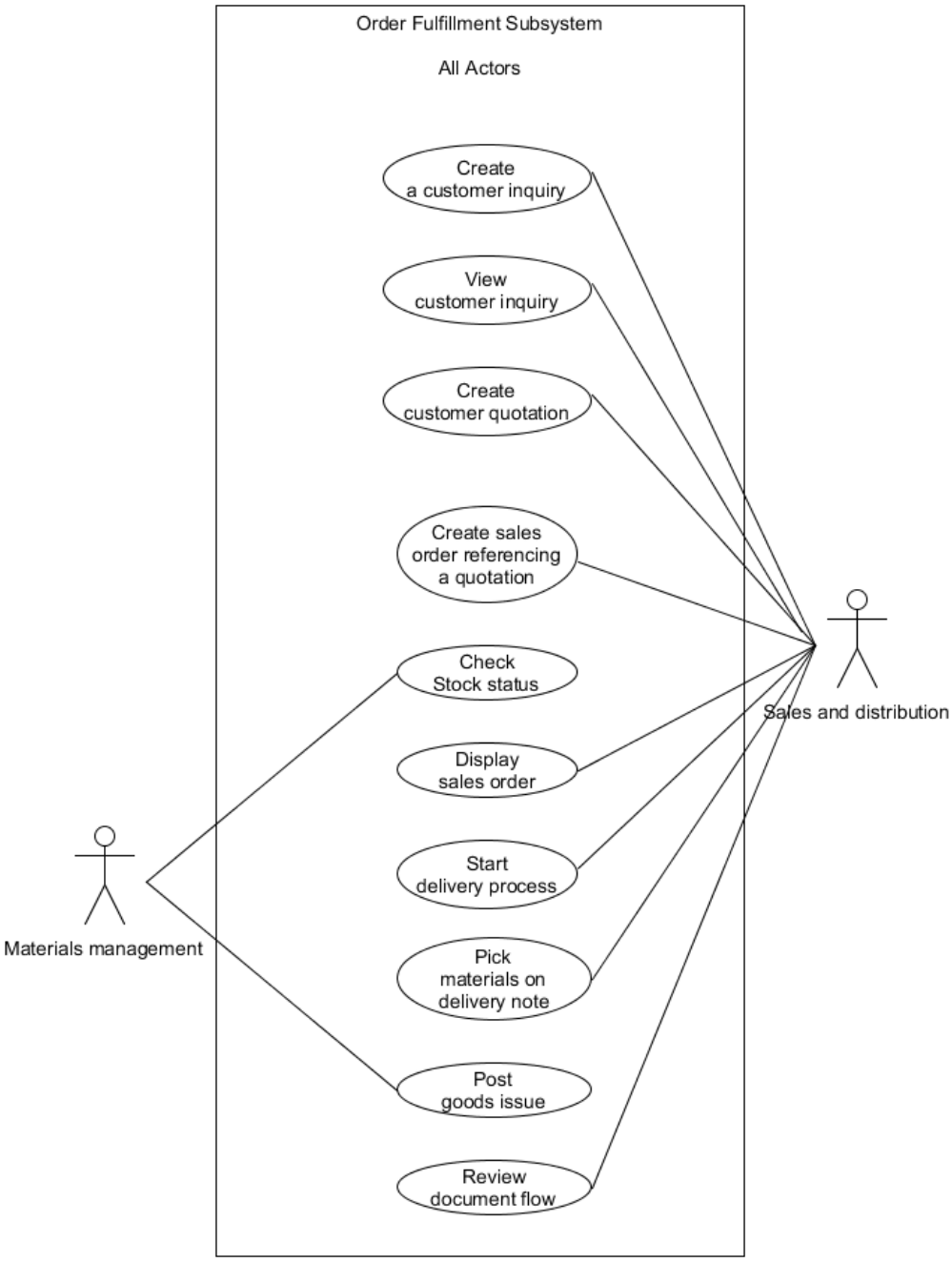
Unified Modelling Language (UML): Set of standards and notation for graphical models.
Use case: An activity the system performs often in response to a user's request (oval).
Automation boundary: Represents the border between the computerised portion of the application with the actors who operate the application (single line border around use cases).
Actor: Personnel or an external entity that interacts with the use case by supplying or receiving data.
Create Read Update Delete: Sufficient use cases exist to add, view, amend and remove particular pieces of information

The Order Fulfillment Subsystem

Use case	Actor	Brief use case description
Create sales order referencing a quotation	Sales and distribution	Sales and distribution employee enters a quotation number, and the system assigns a sales order number, generates a sales order document record from the quotation given, and creates a sales order record.
Check stock status	Materials management	Materials management employee enters product number, at the system retrieves the product record and returns the number of items in stock.
Display sales order	Sales and distribution	Sales and distribution employee enters a sales order number, and the system retrieves and displays the sales order record with the referenced quotation number.
Start delivery process	Sales and distribution	Sales and distribution employee enters a sales order number, and the system alters the status of the sale to delivered and sends a message to the external shipping application to deliver the product.
Pick materials on delivery note	Sales and distribution	Sales and distribution employee enters the sales order number, and the system retrieves the sales order product, employee selects materials by entering their product number and quantity, the system assigns a Materials delivery note number and generates a delivery note record listing the items and their quantities.
Post goods issue	Materials management	Materials management employee enters a material delivery note number, and the system retrieves the materials delivery note record and iteratively deducts the stock from the product inventory record.
Review document flow	Sales and distribution	Sales and distribution employee enters a customer inquiry number, and the system retrieves and displays the customer quotation, customer sales order, customer invoice and materials delivery note with the matching inquiry number.

The Order Fulfillment Subsystem (Continued)

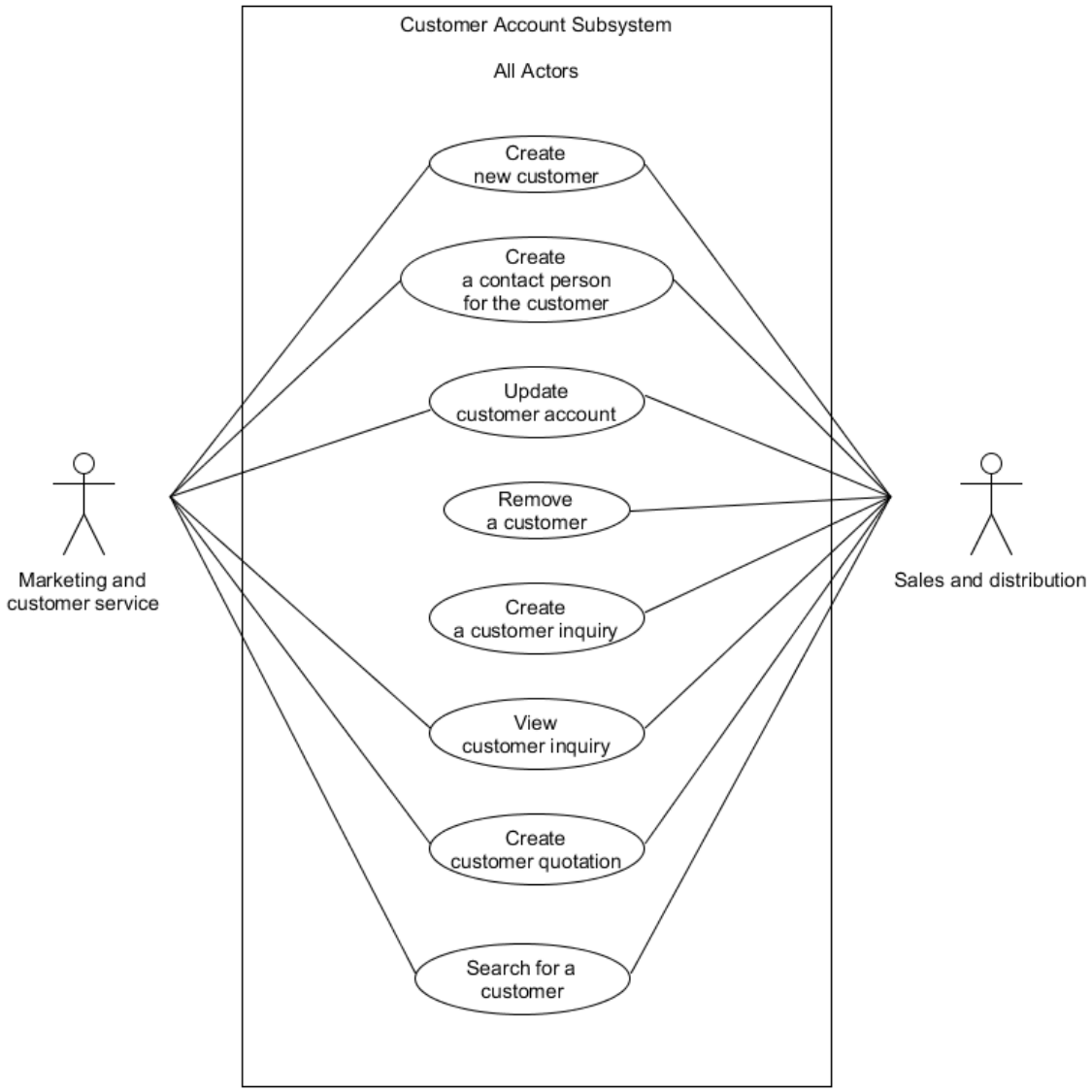
Use case	Actor	Brief use case description
Create customer inquiry	Sales and distribution	Sales and distribution employee enters customer inquiry data and the product the customer is requesting a quotation for, and the system assigns an inquiry number and creates an inquiry record.
View customer inquiry	Sales and distribution, Marketing and customer service	Employee enters an inquiry number, and the system retrieves and displays the inquiry data.
Create customer quotation	Sales and distribution	Sales and distribution employee enters a customer inquiry number and related quotation information such as cost projected, and the system generates a quotation, assigns a quotation number, and creates a quotation record referencing the inquiry number



The above use case diagram presents the use cases for the order fulfillment subsystem and their relationships with the sales and distribution and materials management department. Sales and distribution and materials management are allowed to access the order fulfillment subsystem directly, as indicated by the relationship lines. Sales and distribution can access most of the use cases such as *Create a sales order, Display sales order, Start delivery process*, *Pick materials on delivery note*, and *Review document flow* because most of these use cases are related to sales and distribution process. For example, use cases that related to sales and distribution often contain keywords sales and delivery. *Create a* customer *inquiry* and *View customer inquiry* is used in this system by the sales and distribution employee actor to begin the order to be fulfilled. *Create customer quotation* then accessed by sales and distribution as Global Bike Incorporated generates the figures for the quote. An employee from the materials management department is the other actor involved in this subsystem who is related to handling goods and stocks, so they interact with the subsystem to *Check stock status*, and execute *Post goods issues.*

The Customer Account Subsystem

Use case	Actor	Brief use case description
Create new customer	Sales and distribution, Marketing and customer service	Sales and distribution employee enters new customer account data and then follows up with an address for the business partner and optionally an email address, and the system assigns account number and creates a customer record.
Create a contact person for the customer	Sales and distribution, Marketing and customer service	Sales and distribution employee enters contact person data including phone number, address and email, and enters the customer number that the contact person is for, and the system creates a record for the contact person and associates it with a customer account number.
View customer details	Sales and distribution, Marketing and customer service	Department employee enters a customer number, and the system retrieves and displays customer data.
Update customer account	Sales and distribution, Marketing and customer service	Employee enters customer account number, and the system retrieves and displays current customer data, customer can select information to update and enter the new value, and the system updates the customer record.
Remove a customer	Sales and distribution, Marketing and customer service	Employee enters customer account number, and the system removes customer record.
Search for a customer	Sales and distribution, Marketing and customer service	Employee enters a query, and the system searches for a similar customer name or number, and displays the results, employee selects a customer, the system invokes the view customer details to display the account information.

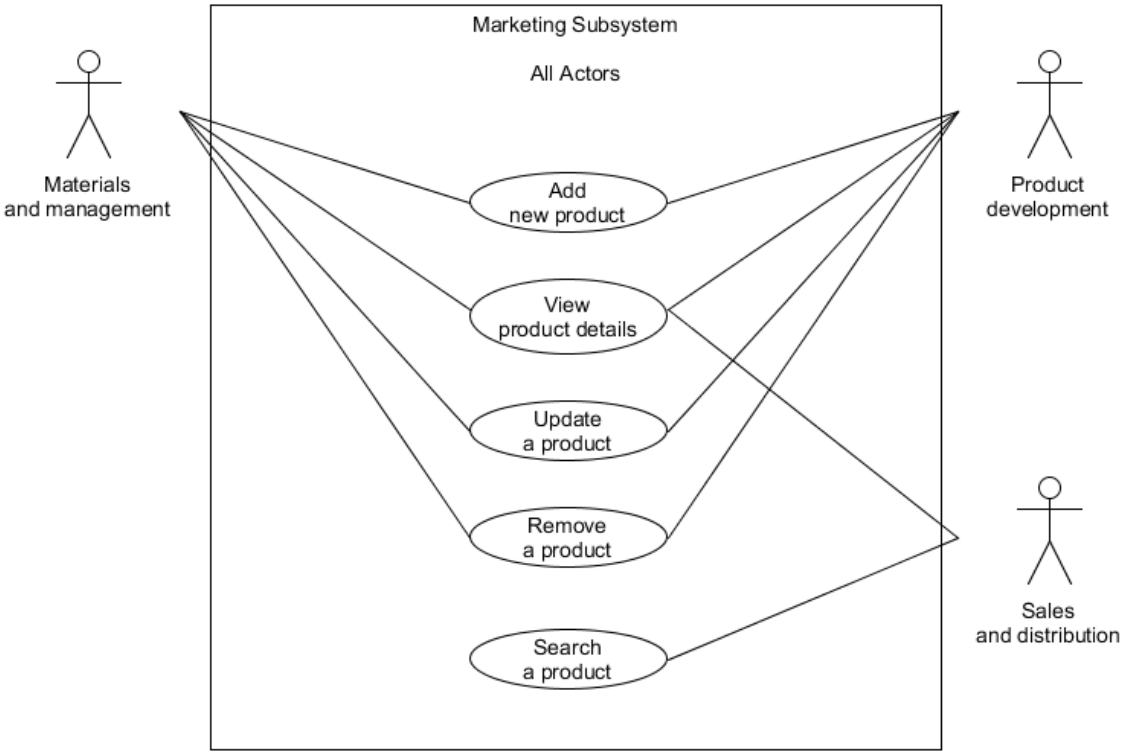


This use case diagram presents all the use cases relating to the Customer Account subsystem and their relationship with the sales and distribution department and marketing and customer service actors. As sales and distribution employees will be receiving the information to insert into the system via in-store interactions or phone call, this actor has access to use cases like *Create a new customer*, *Create a contract person for the customer, Update customer account*.

Create new customer is first use case which is also involved with customer, because the customer receives an email address output from the system containing confirmation and a summary of their details. The Customer is also involved in *create contact person for the customer*, *update customer account* and *Create customer* inquiry as they receive email confirmation with the information produced by these use cases.

The Marketing Subsystem

Use case	Actor	Brief use case description
Add new product	Materials management, Sales and distribution, Product development	Department employee enters data for a new product and product image(s), and the system creates a product record, and optionally the employee can enter the standard number for that product or the system can generate a new product number.
View product details	Materials management, Sales and distribution, Product development	Department employee enters a product number, and the system retrieves and displays product data.
Update a product	Materials management, Sales and distribution, Product development	Department employee enters product number, and the system retrieves and displays product data, employee selects the information to update and inserts the new value, and the system updates the product record.
Remove a product	Materials management, Sales and distribution, Product development	Department employee enters a product number, and the system removes the product record.
Search for a product	Materials management, Sales and distribution, Product development	Department employee enters a query, and the system searches for a similar product name, product description or product name, and displays the results, employee selects a product, the system invokes the view product details to display the product details.



The above figure presents the use cases from the Marketing subsystem and their relationship with the materials management, product development and sales and distribution departments. The materials management department has access to these CRUD functions for the system's products as they are directly involved in procuring the goods for vendors. Here, the difference between a line item and a product in the system is emphasised, in that each line item received from the vendor must be entered into the system as a product. As such, it is assumed that materials management manages the products and materials by interacting with the *View Product Details, Remove a product* and *Update a product*. Because materials management accesses *Update a product*, it is also assumed they would be able to *Search for a product* to update.

The product development department consider customer preferences and business strategy, so it is assumed they will also manage product details. This allows them to *View product details* to assist in customer inquiries, and *Update product details* to improve the product's marketing based on customer tastes and recommendations, and *Remove a product* that doesn't get sufficient attention from customers. They also have access to *Add a new product* in which case a product with no stock can be added.

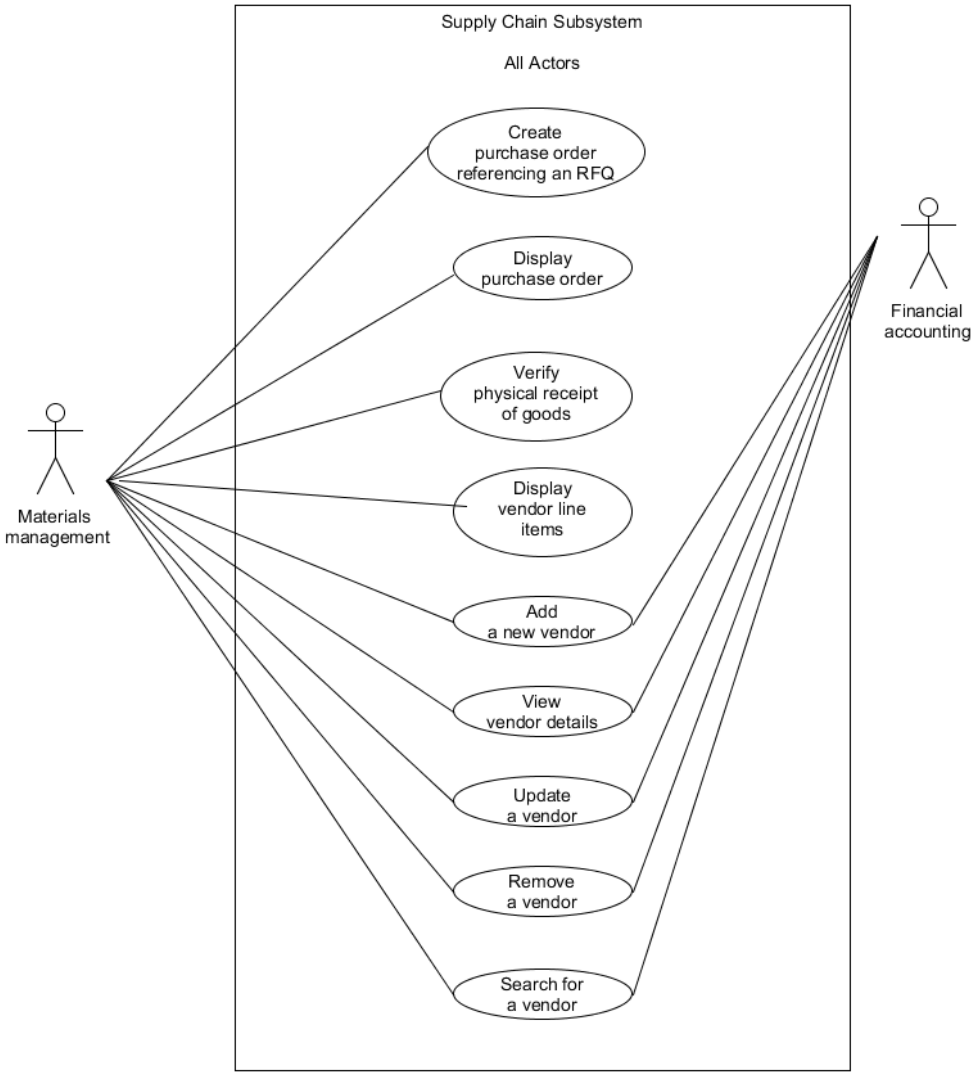
Sales and distribution directly deal with customers, so they can *View product details* to explain the customer about the product and *Search for a product* if necessary, to facilitate rapid customer service.

The Supply Chain Subsystem

Use case	Actor	Brief use case description
Create purchase order referencing an RFQ	Materials management	Materials management employee references the accepted Request for Quote (RFQ) from the vendor, and the system generates a purchase order record listing the items, quantities and agreed costs, assigns a purchase order number and creates a purchase order record.
Display purchase order	Materials management	Materials management employee enters a purchase order number, and the system retrieves and displays the purchase order information and its RFQ reference number.
Create goods receipt for purchase order	Materials management	Materials management employee enters a purchase order number that has been successfully received, and the system generates a goods receipt record, assigns a goods receipt number, and increases warehouse stock accordingly.
Verify physical receipt of goods	Materials management	Materials management employee enters a goods receipt number, system displays receipt, and Materials Management records correctness.
Display vendor line items	Materials management	Materials management employee enters a purchase order number, and the system retrieves and lists the items and their details.
Add a new vendor	Materials management, Financial Accounting	Materials Management or financial accounting enters new vendor data including a vendor address and optionally an email address, and the system assigns vendor number, creates a vendor record.
View vendor details	Materials Management, Financial Accounting	Materials Management or Financial Accounting enters vendor number, and the system retrieves and displays vendor data.

The Supply Chain Subsystem

Use case	Actor	Brief use case description
Update a vendor	Materials management, Financial accounting	Materials management or financial accounting employee enters a vendor number, and the system retrieves and displays vendor data, user selects information to update and inputs the new value, and the system updates the vendor record.
Remove a vendor	Materials management, Financial accounting	Materials management or financial accounting enters vendor number, and the system removes vendor record.
Search for a vendor	Materials management, Financial Accounting	Department employee enters a query, and the system searches for a similar vendor name, and displays the results, employee selects a vendor, the system invokes the view vendor details to display the vendor details.



The above figure presents the use cases from the Supply Chain subsystem and their relationship with the materials and management and financial accounting departments. Materials management have access to all use cases in the supply chain subsystem as they are the primary actor responsible for product procurement in the supply chain. They are involved with use cases such as *Create purchase order and Display purchase order* to maintain the stock level of materials and manage the ordering of new supplies, *Verify physical receipt of goods* ensure accuracy of warehouse stock. Additionally, *Display line the items* allows the items from each vendor to organise the inventory of the system's products in a readable way. *Add a new vendor, View vendor details, Update a vendor and Remove a vendor* allows the department to see and manage vendor information, enhancing the supply chain.

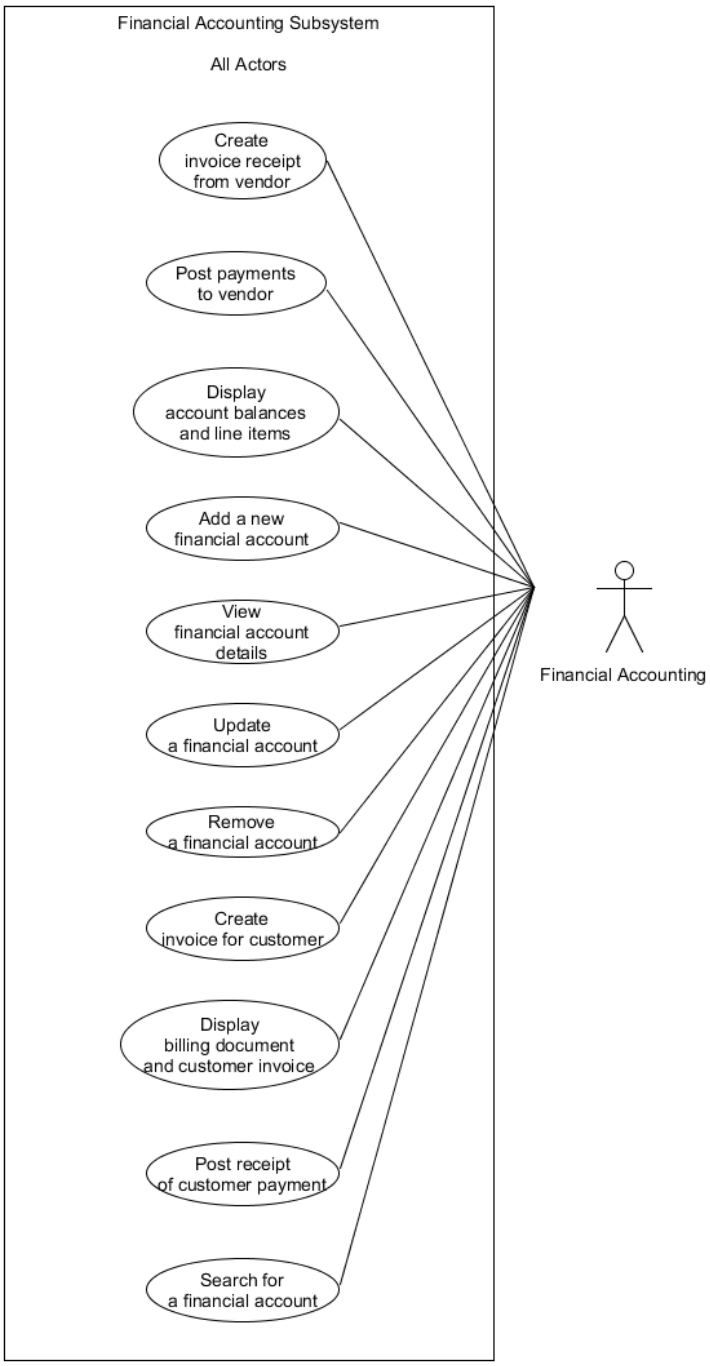
The other actor is the Financial accounting who is involved with use cases like, *Add a new vendor, and View vendor details* to associate vendor financial account information to send payments to in financial statements, and *Update a vendor and Remove a vendor* to ensure information is current in invoice receipts..

The Financial Accounting Subsystem

Use case	Actor	Brief use case descriptions
Create invoice receipt from vendor	Financial accounting	Financial accounting employee enters information from the invoice received from the vendor and information from the payment sent, and the system generates an invoice receipt, assigns an invoice receipt number, and creates an invoice receipt record.
Post payments to vendor	Financial accounting	Financial accounting employee enter purchase order number, and the system retrieves purchase order record and its associated vendor record, the financial accounting employee enters General Ledger account number, and the system retrieves the address from the associated vendor record.
Display account balances and line items	Financial accounting	Financial accounting employee enters a financial account existing in the system, and the system retrieves and displays the current balances for the account and lists the costs of each line item.
Add a new financial account	Financial accounting	Financial accounting employee enters data for a new General Ledger financial account namely financial account number, and the system verifies it is a valid account then creates a financial account record.
View financial account details	Financial accounting	Financial Accounting enters financial account number, and the system retrieves and displays the financial account record.
Update a financial account	Financial accounting	Financial Account enters financial account number, and the system retrieves and displays account data, user selects the information to update with the new values, and the system verifies and updates the financial account record.
Remove a financial account	Financial accounting	Financial Account enters the financial account number for an account is the system, and the system removes the account record.

The Financial Accounting Subsystem

Use case	Actor	Brief use case descriptions
Search for a financial account	Financial accounting	Financial accounting employee enters a query, and the system searches for a similar financial account name or number, and displays the results, employee selects a financial account, the system invokes the view financial account details to display the account information.
Create invoice for customer	Financial accounting	Financial accounting enters sales order number, and the system retrieves the invoice data, assigns an invoice number and generates an invoice record applying appropriate GST.
Display billing document and customer invoice	Financial Accounting	Financial accounting enters a sales order number, and the system retrieves and displays the billing document and the customer invoice record.
Post receipt of customer payment	Financial accounting	Financial accounting enters a billing document and customer invoice number and a financial account number, and the system verifies the corresponding payment amount has been received, generates a payment receipt, and retrieves the address from the associated customer record.



The above figure presents the use cases from the Financial Accounting subsystem and their relationship with the financial accounting actor. Financial accounting is the only actor allowed to access all the use cases in the subsystem, particularly given the sensitive and important nature of financial information. *Create invoice receipt from vendor* and *Create invoice for customer* are both accessible to create and record the appropriate documentation payments. *Post payments to vendor* and *Post receipt of customer payment* are both handled by the financial accounting department by using information received from *View financial account details*. In order to accurately manage the organisations General Ledger financial accounts, financial accounting can supply and interact with *View financial account details*, *Update a financial account*, *Remove a financial account* and *Search for financial account.* Financial accounting can also *Display billing document and customer invoice* for appropriate financial management and overview purposes. Further, *Display account balances and line items* allows the financial accounting employee to remain aware of financial performances and positions.

60DEGREES
SYSTEMS SOLUTIONS

UML Activity Diagrams

DETAILED DOCUMENT EXPLANATION

As defined in the **Use cases and UML Use Case Diagrams** document, Unified Modelling Language (UML) is a set of standards and notation for graphical models produced in system development. A UML Activity Diagram is another UML model that is used to describe the various actor and system activities, actor or component that completes each activity, and the workflow of these activities. A workflow is sequential flow of steps required to complete one business process. An activity diagram is often used to decompose the business processes into a progression of actions. The diagram can also be used to decompose use cases into a progression of actions as used in **Use Case Descriptions (Fully Developed)**. For each agent or system component that performs an activity in the workflow, a rectangular column is added to the diagram and labelled to represent a swimlane for that actor. This swimlane will contain all of the processes that that actor completes. It is called a swimlane because each agent performs the tasks alongside other agents in the workflow.

The activity diagram begins with a filled in black circle named pseudo to denote the start of a workflow, and at the end of the workflow a psuedo with an outer border represents the end of the workflow. Flattened ovals represent the individual activities in the workflow, which are labelled in a verb-noun format, and arrows referred to as transition arrows represent the flow of activities. A diamond is used to represent a decision where the workflow will select a path based on the result of the given condition. A workflow path is a consecutive set of activities connected by transition arrows. A heavy solid line named the synchronisation bar is an activity diagram component that either splits the path into multiple paths with separate processes (a split synchronisation bar), or recombines separated paths (a join synchronisation bar). An instance that the synchronisation bar could be used is to represent a loop. The split synchronisation bar is generally given a text label, for example 'for each item', and a join synchronisation bar is often given a label 'end foreach' .

Note, the system is not included as a swimlane as the following diagrams model the business process workflows for the current manual process. The system's activities are however included in the activity diagrams modelled for each **Use Case Descriptions (Fully Developed).** Additionally, use case activity diagrams also include a flattened oval with a pink-background to denote the activity of triggering another use case.

KEYWORDS

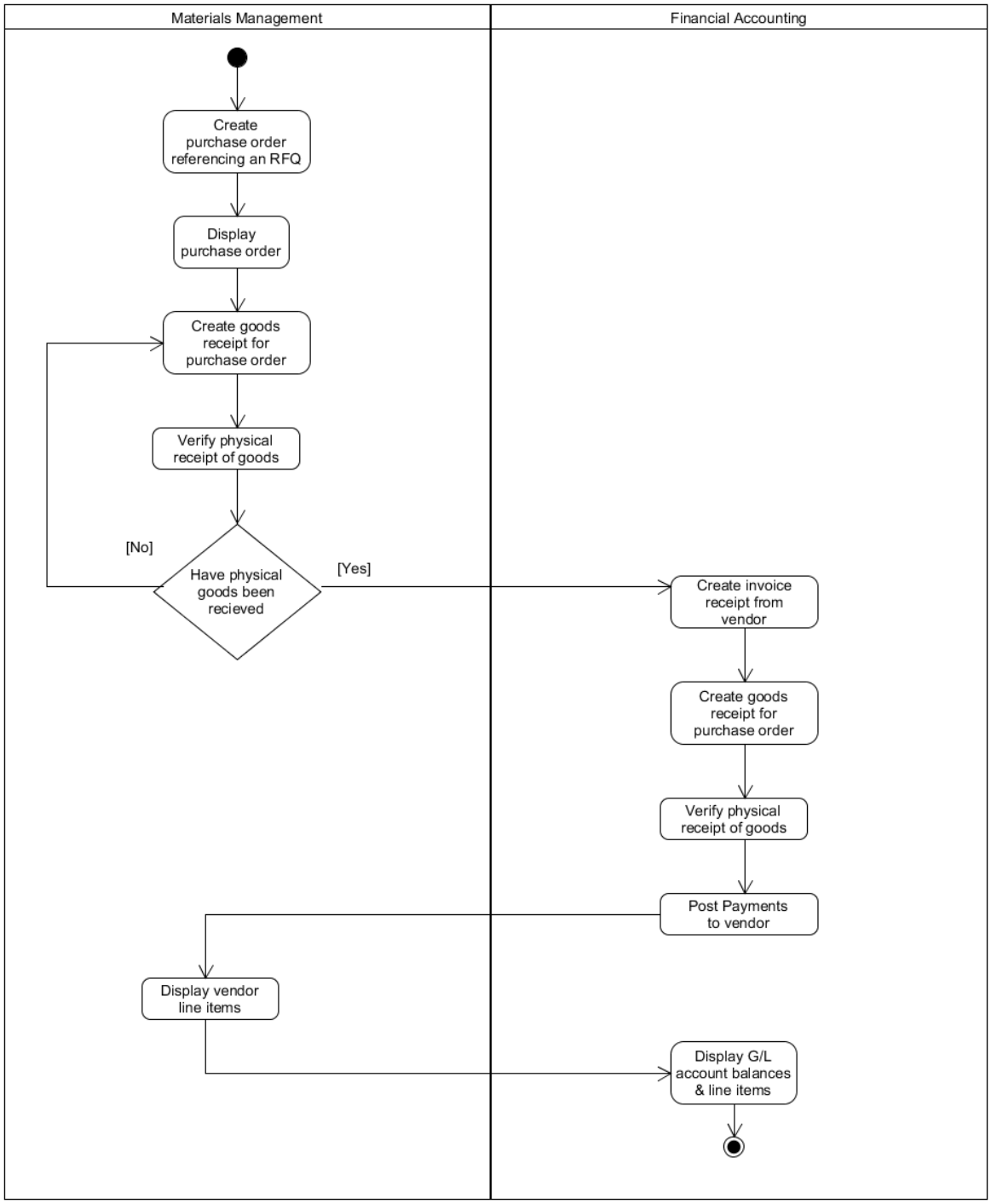
UML Activity Diagrams
A type of UML model to describe the various actor and system activities, actor or component that completes each activity, and the workflow of these activities.

- **UML Activity Diagrams section:** Can decompose the business processes into a progression of actions.
- **(Fully Developed) Use Case Descriptions:** Can also be used to decompose use cases into a progression of actions.

Workflow: A sequential flow of steps required to complete one business process
Swimlane: A rectangular column that encases all of the processes that actor completes
Psuedo: A filled in black circle to denote the start of a workflow, and a pseudo with an outer border to represent the end of the workflow.
Transition arrows: Arrows that represent the flow of activites.
Activity: A step in the business process workflow, denoted by a flattened oval
Decision: A choice where the workflow will select a path based on the result of the given condition.
Path: Consecutive activities with workflow arrows connecting them.
Synchronisation Bar: A heavy solid line that splits the workflow into multiple paths (a split syncrhonisation bar) or recombines separated paths (a join synchronisation bar).

Procure-to-cash

The following figure is an activity diagram that analyses and describes the various activities in the procure-to-cash business process workflow. It contains two swimlanes for each actor: the materials management department and the financial accounting department. Processing begins with a materials management employee who performs the activity *Create purchase order referencing a Request For Quote (RFQ)*, then will *Display purchase order* allowing them to review the created purchase order. Next, the employee will *Create goods receipt for purchase order*, which will acknowledge that the items have been successfully delivered from the vendor to Global Bike Incorporated's warehouse. The employee will then perform the activity *Verify physical receipt of goods*, which leads to a decision indicated by the diamond 'Have goods been physically received'. If the physical goods have not been received, then a new *Create Goods Receipt for purchase order* will be performed until the goods are correctly obtained. Otherwise, if they are physically present in the warehouse, the workflow will cross into the financial accounting actor swimlane who perform the activity *Create invoice receipt from vendor*, indicating the payment of the invoice for the goods received. This *Create goods receipt from purchase order* and *Create invoice receipt from vendor* process is repeated for the remaining items when they are delivered. Next, the financial accounting actor will *Post Payments to Vendor*. The workflow moves to the materials management department who can then *Display vendor line items* which will list the goods purchased from a particular vendor. Finally, the diagram returns to the financial accounting department swimlane who will *Display General Ledger account balances and line items* to review and confirm the correct payment.

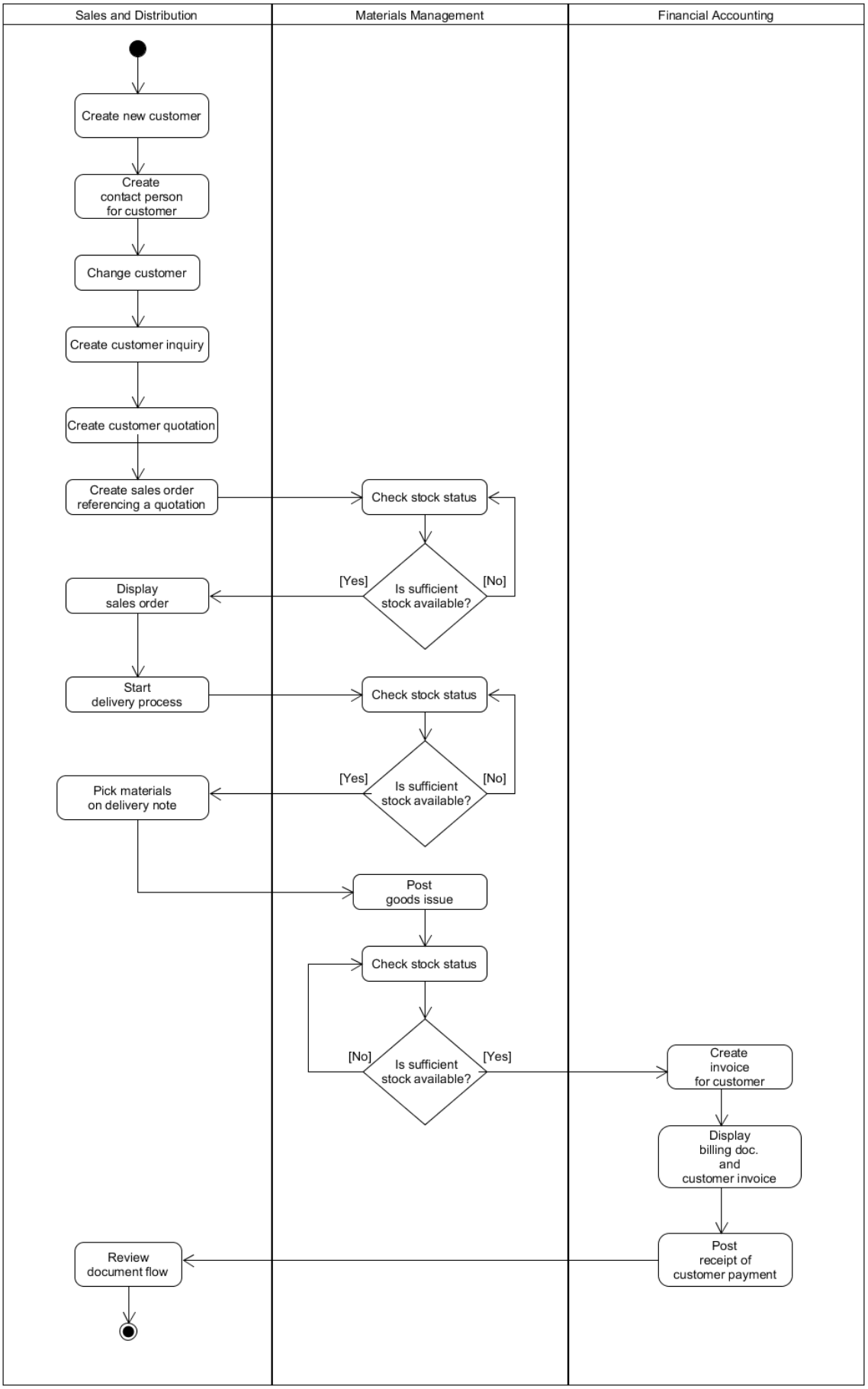

Materials Management
Financial Accounting
Create purchase order referencing an RFQ
Display purchase order
Create goods receipt for purchase order
Verify physical receipt of goods
Have physical goods been recieved
[No]
[Yes]
Create invoice receipt from vendor
Create goods receipt for purchase order
Verify physical receipt of goods
Post Payments to vendor
Display vendor line items
Display G/L account balances & line items

Order-to-cash

The following figure is an activity diagram that analyses and describes the various activities in the order-to-cash business process workflow. It contains three swimlanes for each actor: the sales and distribution, materials management, and financial accounting departments. Processing begins with the sales and distribution actor who performs the activity *Create new customer* as they input data regarding a new business partner, and *Create contact person for customer* to act as a company master who can also communicate with Global Bike Incorporated. Afterwards, the account is changed to interact with the contact person through the *Change customer* activity. Next, the sales and distribution employee can *Create customer inquiry* associated with the contact person representing the business partner by entering the customer inquiry data, before making an estimation of inquiry purchase costs in the activity *Create customer quotation*. The customer will then accept or reject the quotation, which is outside the scope of the businesses internal workflow. Assuming the customer accepts the quote, sales and distribution will *Generate a sales order referencing a quotation* to confirm the sales of goods and services.

After that, the workflow will shift to the materials management swimlane who checks whether there is sufficient stock of the sold goods at the given time using *Check stock status*. The decision diamond 'Is stock available?' represents the result of the check stock status activity. If stock is not available, then the workflow will loop to recheck the stock status for the given sales order until stock status indicates sufficient products exist. Otherwise, if stock does exist then the swimlane will return to sales and distribution who perform *Display sales order* to confirm the details and *Start delivery process*, which alters the status of the sales order to dispatched. Next, materials management will *Check stock status* once more with a decision diamond 'Is stock available?' as a data accuracy check that there is enough stock materials to be picked on the delivery note. If enough stock exists, then materials management will conduct *Pick materials on delivery note* to record the materials and quantity contained in the delivery. Note, the cost of the goods is contained in the invoice.

The workflow moves to the materials management department who then performs the activity *Post goods issue* to reduce stock of the goods according to the quantity of products moved out of the warehouse. The department will then *Check stock status* again to ensure that the stock has been reduced correctly. Now the goods have been successfully dispatched, the workflow moves to the financial accounting swimlane who will *Create invoice for customer*. This will display the costs for the goods they have ordered. After creating the invoice for the customer, Financial accounting perform *Display billing document and customer invoice* to review the information and ensure the data is correct. Once the payment has been received, the department will *Post receipt of customer payment*. Finally, the workflow will return to the sales and distribution actor to *Review document flow*, which will allow all of the documents to be verified and the order-to-cash business process complete for one order.


Sales and Distribution
Materials Management
Financial Accounting
Create new customer
Create contact person for customer
Change customer
Create customer inquiry
Create customer quotation
Create sales order referencing a quotation
Check stock status
[Yes]
Is sufficient stock available?
[No]
Display sales order
Start delivery process
Check stock status
[Yes]
Is sufficient stock available?
[No]
Pick materials on delivery note
Post goods issue
Check stock status
[No]
Is sufficient stock available?
[Yes]
Create invoice for customer
Display billing doc. and customer invoice
Post receipt of customer payment
Review document flow

(Fully Developed) Use Case Descriptions

DETAILED DOCUMENT EXPLANATION

As defined in the **Project Iteration Schedule** document and addressed in the **Use case and UML use case diagrams** document in this report, a use case is an activity the system performs often in response to a user's request. The brief use case description offered in the **Use case and UML use case diagrams** section of this report offers enough detail for simple use cases or a brief overview of complex cases. The fully developed use case description offers a formal template of 11 sections (detailed below) to comprehensively expand on the details of each use case. is the most formal and thorough method for documenting a use case. It consists of the following sections:

1. Use case name
2. Scenario: Unique instance of a use case that can trigger a unique set of internal activities
3. Triggering event: An event is something that occurs at a specific time and place, can be described, and should be remembered by the system such as customer makes a purchase. The triggering event is an event that occurs to cause the use case.
4. Brief description:
5. Actors: The personnel, group or an external entity that interacts with the use case by supplying or receiving data, also referred to as an external agent.
6. Related use cases: Identifies and describes how other use cases are related to the use case, such as invoking or requiring information. For instance, *Create a new customer* use case may be invoked by *Create a customer inquiry*, as a customer account must exist for an employee to create a customer inquiry.
7. Stakeholders: A stakeholder is the personnel who have an interest in the successful development of the new system, such as the clients who may fund the project, and users who will use the system regularly. This section generally identifies stakeholders who may be interested in the use case but are not necessarily directly interacting with the system like actors.
8. Preconditions: Conditions that must be true before the use case can start
9. Postconditions: Conditions that must be true upon successful completion of the use case.
10. Flow of activities: Describes the detailed workflow to use the use case. Can be presented as a two-column version with a text-based explanation, as an activity diagram, or both formats can be presented for maximum detail.
11. Exception conditions: Alternative activities that may be required when the conditions of the use case are other than normal.

To demonstrate thorough consideration of technical details in an understandable way, **two central use cases from each subsystem** have been elaborated using a fully developed use case description. Additionally, an **activity diagram** representing the flow of activities is included. This employs the notation listed in **UML Activity Diagrams**, as well as including an oval symbol with a pink-coloured background. This represents a workflow activity when the system calls (or invokes) a different use case.

KEYWORDS

(Fully Developed) Use Case Descriptions
Offers a formal template of 11 sections (detailed below) to comprehensively expand on the details of each use case.

Activity Diagram – Flow of activities
A type of UML model to describe the various actor and system activities, actor or component that completes each activity, and the workflow of these activities. decompose use cases into a progression of actions.

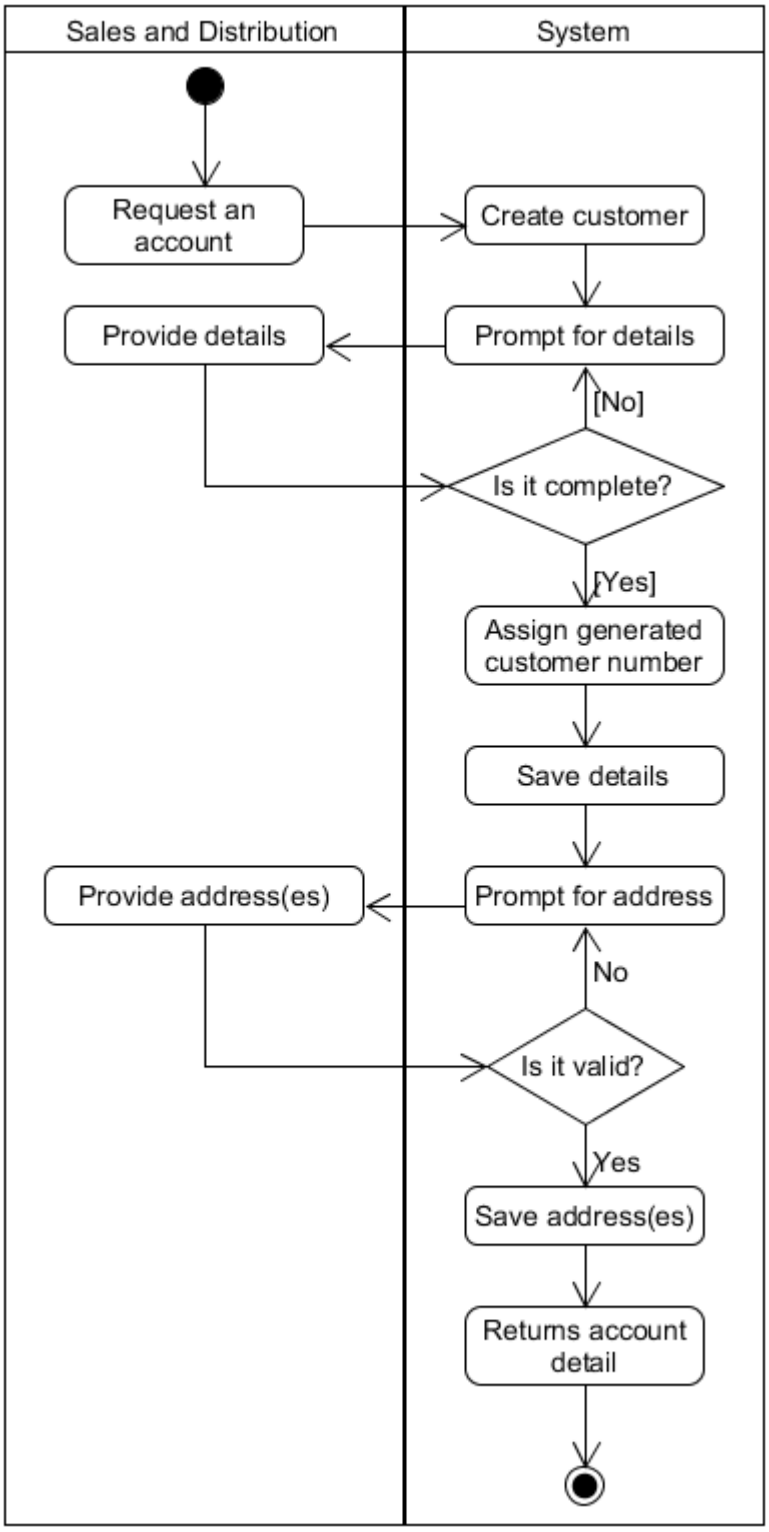
Subsystem: Customer Account subsystem

(Fully developed) Use case description for use case Create new customer

Use case name:	Create new customer	
Scenario:	Create new customer	
Triggering event:	New business partner wants to be a customer with Global Bike Incorporated	
Brief description:	Sales and distribution employee enters new customer account data and then follows up with an address for the business partner, and the system assigns account number and creates a customer record.	
Actors:	Sales and Distribution	
Related use cases:	Might be invoked by the *Create customer quotation* use case. Information associated with the customer *Create a contact person for customer* use case.	
Stakeholders:	Sales and Distribution (department)	
Preconditions:	Customer Account subsystem must be available.	
Post conditions:	Customer must be created and saved. One or more Addresses must be created and saved.	
Flow of activities:	**Actor**	**System**
	1. Sales and distribution employee indicates desire to create customer account and enters basic customer information. 2. Sales and distribution enters an address.	1.1 System creates a new customer record. 1.2 System generates and assigns a customer number 1.3 System prompts customer for address. 2.1 System creates and validates address. 2.2 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data is incomplete. 2.1 The address isn't valid.	

Subsystem: Customer Account subsystem

Activity Diagram for use case <u>Create new customer</u>



Subsystem: Customer Account subsystem

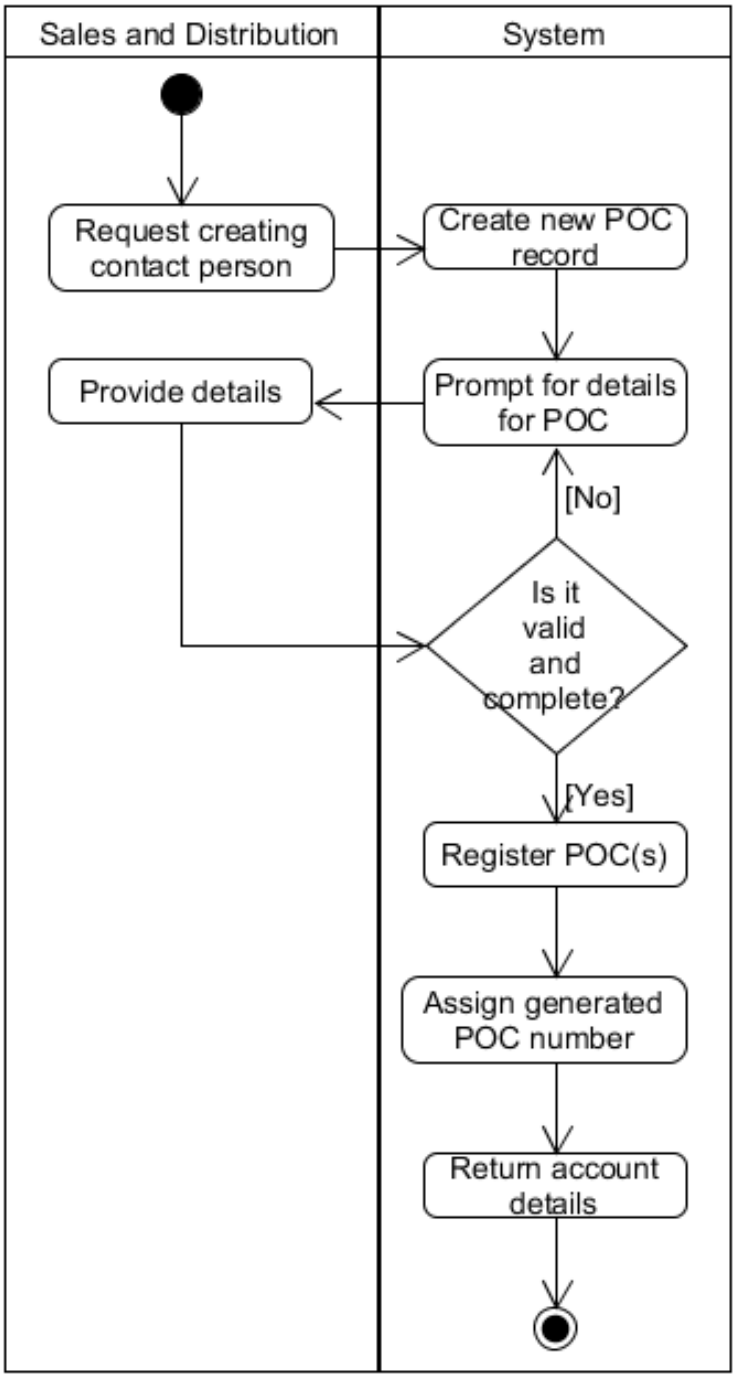
(Fully developed) Use case description for use case <u>Create contact person for customer</u>

Subsystem: The Customer Account subsystem. **Use Case Description (Fully Developed) 2.**

Use case name:	Create contact person for customer	
Scenario:	Create contact person fro customer	
Triggering event:	The creation of customer account *(it needs contact details)*	
Brief description:	Sales and distribution employee enters contact person data including phone number, address and email, and enters the customer number that the contact person is for, and the system creates a record for the contact person and associates it with a customer account number.	
Actors:	Sales and Distribution	
Related use cases:	Fulfils the process of *Create new customer use case*. Enables the CRUD customer use cases to manipulate data.	
Stakeholders:	Sales and Distribution (department)	
Preconditions:	Customer Account subsystem must be available.	
Post conditions:	Contact person must be created, and contact details must be saved.	
Flow of activities:	**Actor**	**System**
	1. Sales and distribution employee creates a contact person for the customer account. 2. Sales and distribution employee enters the contact details of the contact person.	1.1 System creates a new contact person record. 1.2 System generates and assigns a contact person number. 2.1 System prompts customer for contact person details. 2.2 System creates and validates contact details. 2.3 System returns valid contact person details.
Exception conditions:	1.1 Basic contact person data is incomplete. 2.1 The contact details aren't valid.	

Subsystem: Customer Account subsystem

Activity Diagram for use case <u>Create contact person for customer</u>



Subsystem: Order Fulfillment subsystem

(Fully Developed) Use Case Description for use case <u>Create sales order referencing a quotation</u>

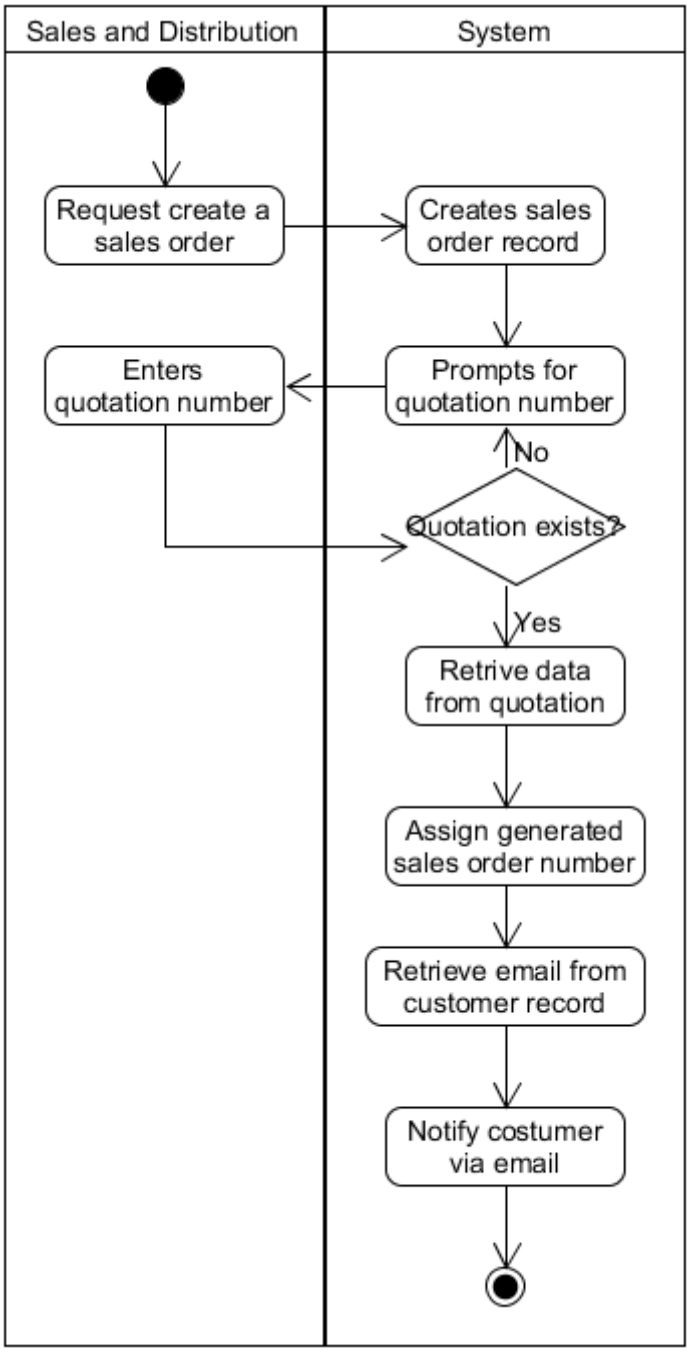
<table>
<tr><td>Use case name:</td><td colspan="2">Create sales order referencing a quotation</td></tr>
<tr><td>Scenario:</td><td colspan="2">Create sales order referencing a quotation</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Business partner successfully purchases product</td></tr>
<tr><td>Brief description:</td><td colspan="2">Sales and distribution employee enters sales order data and a quotation number, and the system assigns a sales order number and creates a sales order record and retrieves customer address to send confirmation.</td></tr>
<tr><td>Actors:</td><td colspan="2">Sales and Distribution</td></tr>
<tr><td>Related use cases:</td><td colspan="2">Invoked by the Create customer quotation use case. Requires information from the View customer account use case.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Sales and Distribution (department)</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Customer Account subsystem must be available.
Order Fulfillment subsystem must be available.
Marketing subsystem must be available</td></tr>
<tr><td>Post conditions:</td><td colspan="2">Sales order must be created and saved.
Quantity and price of goods must be associated with sales order.
Quotation number must be referenced.
Address for customer requesting order must be retrieved and saved.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Sales and distribution employee indicates desire to create a new sales order.

2. Sales and distribution employee enters a quotation number.</td><td>1.1 System creates a new sales order record.
1.2 System prompts employee for quotation number.

2.1 System retrieves quotation record using quotation number.
2.2 System creates sales order using product information (cost, quantity, productID) from the quotation, and references the quotation number.
2.3 System assigns a sales order number.
2.3 System retrieves customer record using customer number on the quotation.
2.4 System returns address</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1.2 Quotation being referenced does not exist.
2.1 Quotation does not exist for quotation number.
2.2 Information is missing in quotation for sales order document inputs.
2.3 Customer does not exist for customer number.
2.4 The address isn’t valid.</td></tr>
</table>

Subsystem: Order Fulfillment subsystem

Activity Diagram for use case <u>Create sales order referencing a quotation</u>



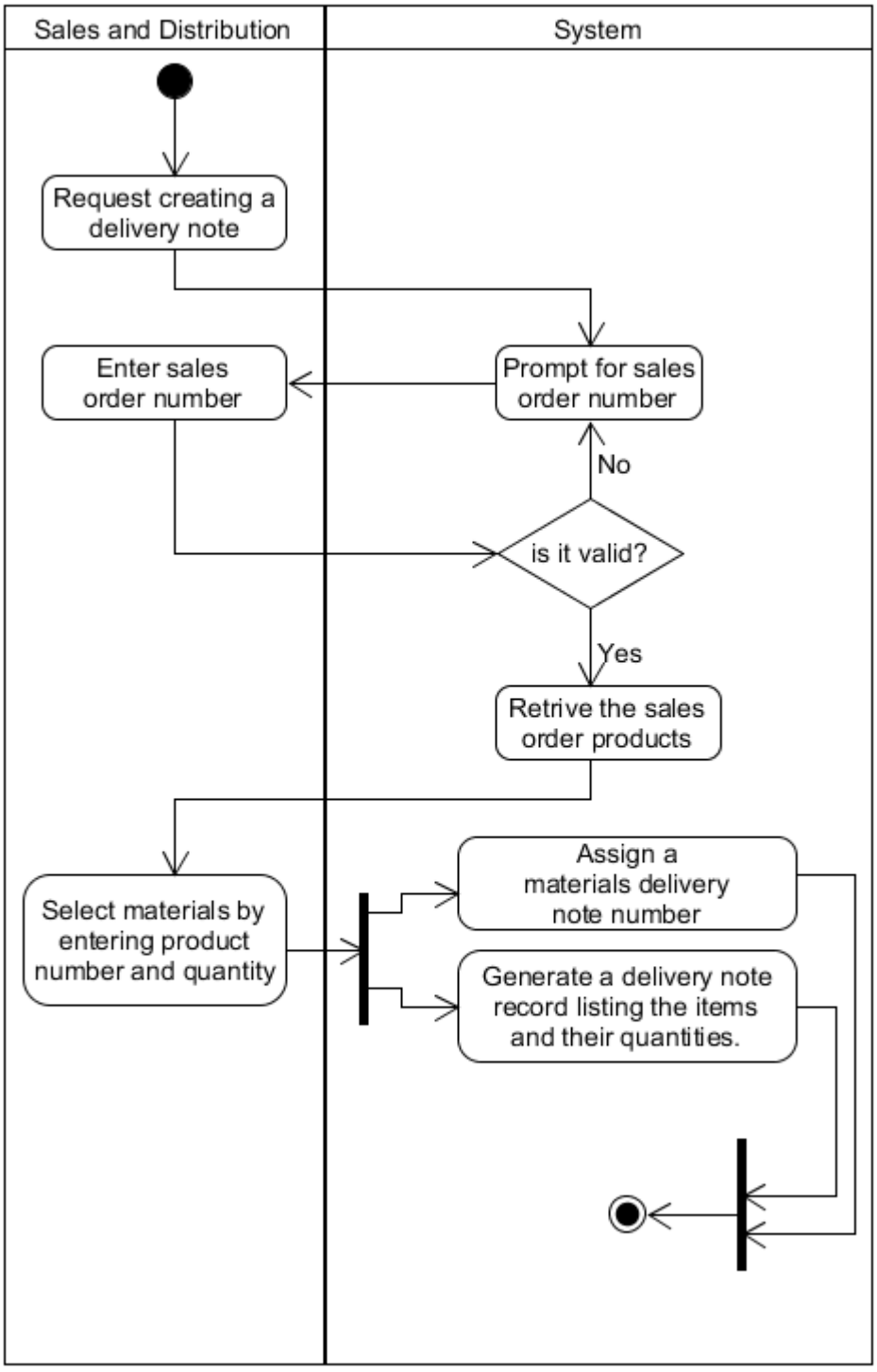
Subsystem: Order Fulfillment subsystem

(Fully Developed) Use Case Description for use case Pick materials on delivery note

Use case name:	Pick materials on delivery note	
Scenario:	Pick materials on delivery note	
Triggering event:	Business partner successfully orders the product(s)	
Brief description:	Sales and distribution employee enters the sales order number, and the system retrieves the sales order product, employee selects materials by entering their product number and quantity, the system assigns a Materials delivery note number and generates a delivery note record listing the items and their quantities.	
Actors:	Sales and Distribution (department)	
Related use cases:	Invoked by the *Create sales order referencing a quotation* use case. Requires information from the *Display sales order* use case.	
Stakeholders:	Sales and Distribution (department)	
Preconditions:	Customer Account subsystem must be available. Order Fulfillment subsystem must be available. Marketing subsystem must be available	
Post conditions:	Delivery note must be created and saved. Quantity and price of goods must be associated with sales order. Sale order number must be referenced. Address for customer requesting order must be retrieved and saved.	
Flow of activities:	**Actor**	**System**
	1. Sales and distribution employee indicates desire to create a delivery note for a new sales order. 2. Sales and distribution employee enters a sales order number.	1.1 System creates a delivery note for a new sales order. 1.2 System prompts employee for sales order number. 2.1 System retrieves sales order record using sales order number. 2.2 System creates delivery note using product information (cost, quantity, productID) from sales order, and references the sales order number. 2.3 System retrieves customer record using customer number on the sales order.
Exception conditions:	1.2 Sales order being referenced does not exist. 2.1 Sales order does not exist for sales order number. 2.2 Information is missing in sales order for delivery note inputs. 2.3 Customer does not exist for customer number. 2.3 The address isn't valid.	

Subsystem: Order Fulfillment subsystem

Activity Diagram for use case <u>Pick materials on delivery note</u>



Subsystem: Marketing subsystem

(Fully Developed) Use Case Description for use case <u>View Product Details</u>

<table>
<tr><td>Use case name:</td><td colspan="2">View product details</td></tr>
<tr><td>Scenario:</td><td colspan="2">View product details</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Employee wants to view the product details of an item</td></tr>
<tr><td>Brief description:</td><td colspan="2">User enters productNumber, and the system retrieves and displays the details for the product.</td></tr>
<tr><td>Actors:</td><td colspan="2">Marketing and Customer Service, Materials Management, Sales and Distribution</td></tr>
<tr><td>Related use cases:</td><td colspan="2">Might be invoked by the Search for product use case. Might be invoked by Create customer inquiry and Update a product. Might invoke Check stock status or Create a customer inquiry.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Sales and Distribution (department), Materials Management (department), Marketing and Customer Service (department)</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Marketing subsystem is available.
Image display features must be available.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">Product record must be retrieved and displayed.
Check for stock might need to be invoked and displayed.
Image for product must be displayed.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Actor indicates desire to view a product.

2. Actor enters a product number.

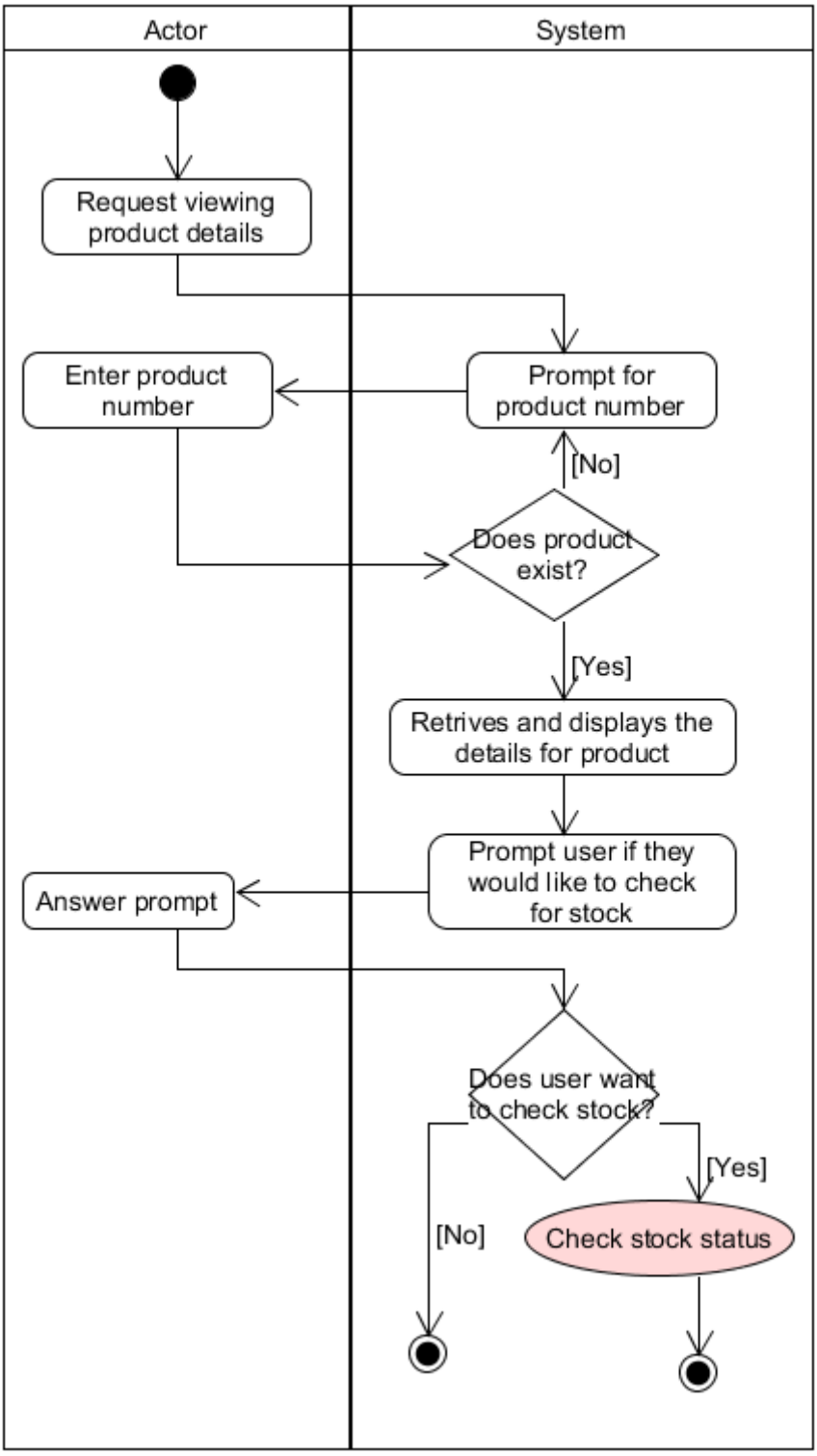
3. Actor enters choice if they want to check the product for stock.

4. Customer (Business Partner) enters choice if they want to create a customer inquiry.</td><td>1.1 System prompts user for product number.

2.1 System retrieves the product record using the product number and displays the information.
2.2 System displays the product image.
2.3 System prompts user if they would like to check for stock.
3.1 If true, system invokes Check stock status use case.
3.2 System prompts user if they are using a customer account if they would like to create a customer inquiry.
4.1 If true, system invokes Create a customer inquiry use case.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">2.1 Product does not exist for the product number.
2.2 Image cannot display for object or no image exists.
4.1 No customer has logged in.</td></tr>
</table>

Subsystem: Marking subsystem

Activity Diagram for use case <u>View product details</u>



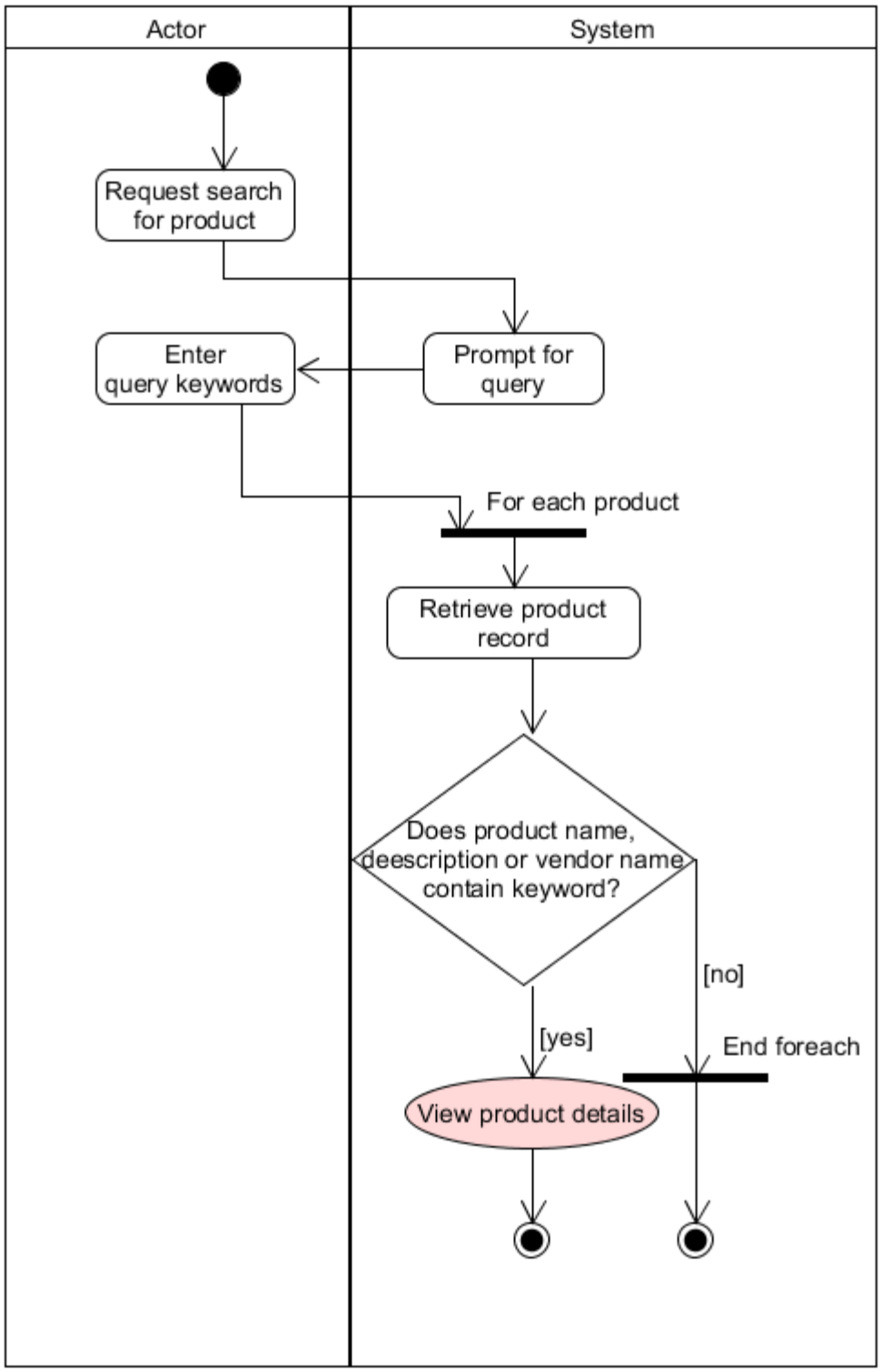
Subsystem: Marketing subsystem

(Fully Developed) Use Case Description for use case Search for a product

Use case name:	Search for a product	
Scenario:	Search for a product	
Triggering event:	Employee wants to search for a product.	
Brief description:	Department employee enters a query, and the system searches for a similar product name, product description or vendor name, and displays the results, employee selects a product, the system invokes the view product details to display the product details.	
Actors:	Marketing and Customer Service, Materials Management, Customer, Sales and Distribution	
Related use cases:	The *Add new product* use case should already created the product that is being searched.	
Stakeholders:	Sales and Distribution (department), Materials Management (department), Marketing and Customer Service (department)	
Preconditions:	Marketing subsystem is available. Image display features must be available.	
Post conditions:	Product records must be searched. Matching product(s) must be displayed.	
Flow of activities:	**Actor**	**System**
	1. Actor indicates desire to search a product.	1.1 System prompts user for product number or any other product information (name, vendor, category).
	2. Actor enters a product number, or any other product information (name, vendor, category) to narrow its search results.	2.1 System displays the matching product(s.)
Exception conditions:	2.1 Product does not exist for the product number.	

Subsystem: Marking subsystem

Activity Diagram for use case <u>Search for a product</u>



Subsystem: Supply Chain subsystem

(Fully Developed) Use Case Description for use case Display vendor line items

Vendor Line Item: A list of items ordered from a vendor. Each item and its details are a line item of the purchase order.

<table>
<tr><td>Use case name:</td><td colspan="2">Display vendor line items</td></tr>
<tr><td>Scenario:</td><td colspan="2">Display vendor line items</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Materials Management employee wants to view the vendor line items for a purchase order</td></tr>
<tr><td>Brief description:</td><td colspan="2">Materials Management enters a purchase order number, and the system retrieves and lists the items purchased and their quantity and cost.</td></tr>
<tr><td>Actors:</td><td colspan="2">Materials Management</td></tr>
<tr><td>Related use cases:</td><td colspan="2">Might be invoked by the Post payments to vendor use case. Might invoke the Add new product use case</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Materials Management (department)</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Supply Chain subsystem must be available.
Marketing subsystem must be available.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">Vendor items must be retrieved and listed.
Vendor items may be added as products to the storefront.</td></tr>
<tr><td>Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td></td><td>1. Materials management employee indicates desire to display vendor line items.

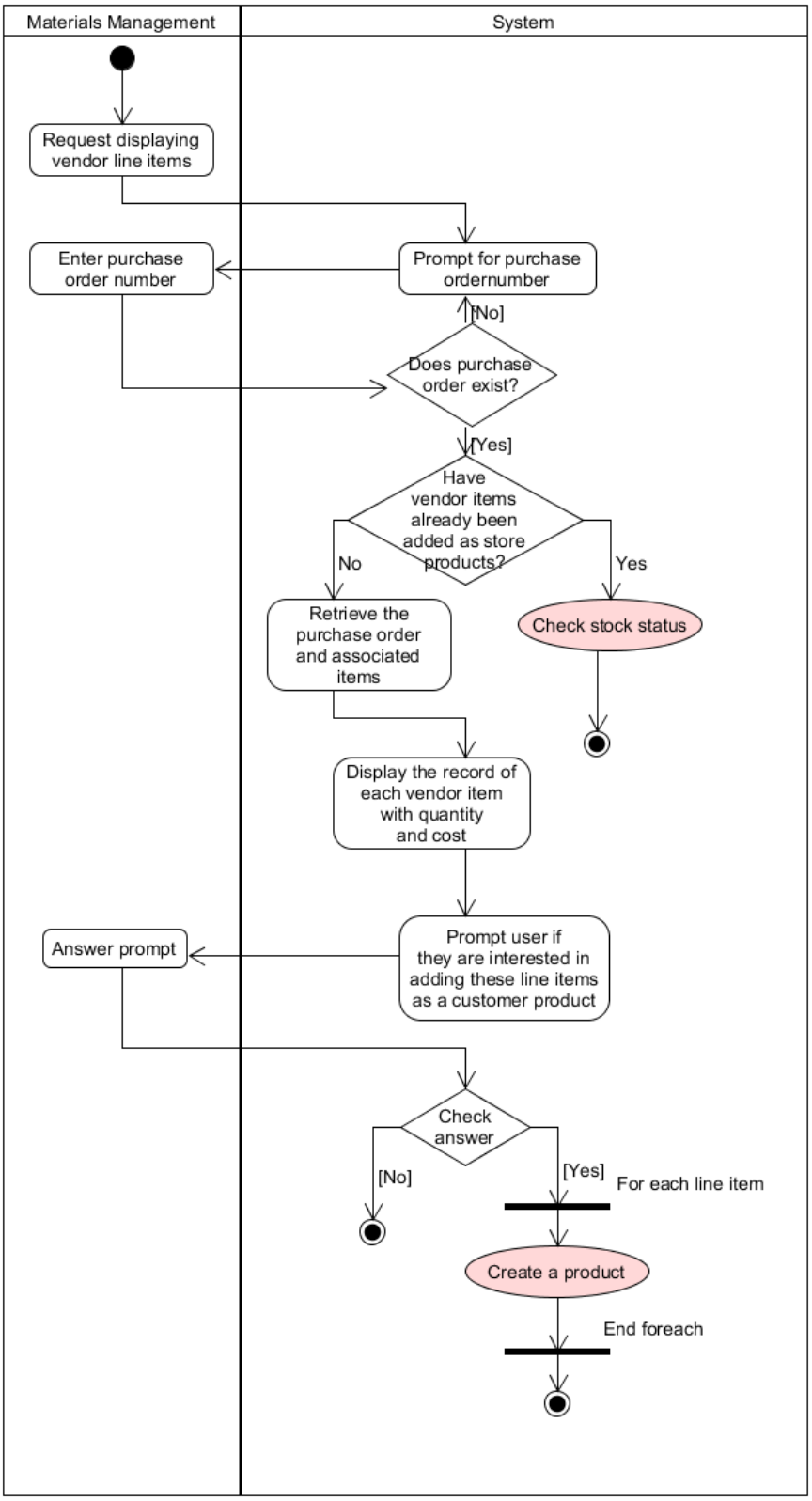
2. Materials management employee enters purchase order number.

3. Materials Management employee enters choice they want to add items as store products.</td><td>1.1 System prompts employee for purchase order number.

2.1 System retrieves the purchase order using the purchase order number.
2.2 System retrieves purchase order items associated with the purchase order.
2.3 System displays the record of each vendor item with quantity and cost.
2.4 System prompts user if they are interested in adding these line items as a customer product.
3.1 If true, system invokes Create a customer product use case for each of the line items.</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">2.1 Purchase order does not exist for the purchase order number.
2.2 No items exist for the purchase order .
2.4 Vendor items have already been added as store products. Thus, invoke Check stock status to increment running quantity.
3.1 No items exist for the purchase order.</td></tr>
</table>

Subsystem: Supply Chain subsystem

Activity Diagram for use case <u>Display vendor line items</u>



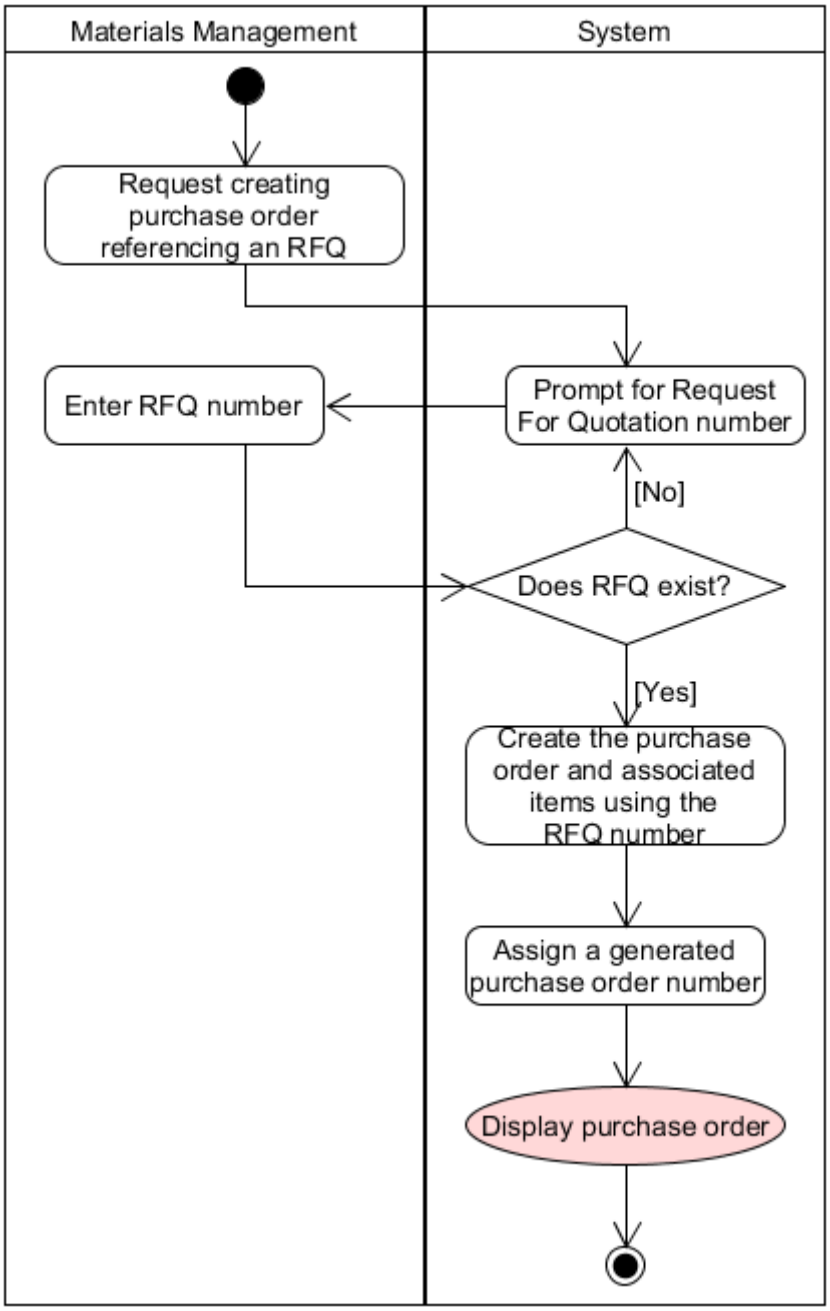
Subsystem: Supply Chain subsystem

(Fully Developed) Use Case Description for use case <u>Create purchase order referencing an RFQ</u>

Use case name:	Create purchase order referencing an RFQ	
Scenario:	Create purchase order referencing an RFQ	
Triggering event:	Materials Management employee needs to create a purchase order by referencing a quotation	
Brief description:	Materials management employee references the accepted Request for Quote (RFQ) from the vendor, and the system generates a purchase order record listing the items, quantities and agreed costs, assigns a purchase order number, and creates a purchase order record.	
Actors:	Materials Management	
Related use cases:	*Display purchase order* use case can be used after this use case.	
Stakeholders:	Materials Management (department)	
Preconditions:	Supply Chain subsystem must be available. Marketing subsystem must be available.	
Post conditions:	Purchase order must be available to display. Purchase order items must be available to display.	
Flow of activities:	**Actor**	**System**
	1. Materials management employee references the quotation to create a purchase order.	1.1 System prompts employee for quotation number. 1.2 System creates the purchase order using the quotation number. 1.3 System creates purchase order items associated with the quotation 1.4 Assigns a unique purchase order number. 1.5 Invoke *Display purchase order* use case.
Exception conditions:	1.1 Quotation does not exist for the quotation number. 1.3 No items exist for the quotation .	

Subsystem: Supply Chain subsystem

Activity Diagram for use case <u>Create purchase order referencing an RFQ</u>



Subsystem: Financial Accounting Subsystem

(Fully Developed) Use Case Description for use case <u>Display G/L account balances and line items</u>

Line Item: A specific item listed in a budget

General Leger (G/L): A set of numbered accounts a business uses to track its financial transactions and to prepare financial reports

<table>
<tr><td>Use case name:</td><td colspan="2">Display G/L account balances and line items.</td></tr>
<tr><td>Scenario:</td><td colspan="2">Display account balances.</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Financial Accounting employee wants to view the General Leger (G/L) account balances and items purchased.</td></tr>
<tr><td>Brief description:</td><td colspan="2">Financial Accounting employee enters an account number, and the system retrieves and displays the current balances and lists the line items purchased for the account.</td></tr>
<tr><td>Actors:</td><td colspan="2">Financial Accounting</td></tr>
<tr><td>Related use cases:</td><td colspan="2">Might be invoked by the Display vendor line items use case. May invoke Create a G/L account use case</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Financial Accounting (department)</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Financial Accounting subsystem must be available.
G/L account services must be available.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">G/L account balance retrieved and displayed.
Line items must be retrieved and listed.</td></tr>
<tr><td rowspan="2">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Financial Accounting employee indicates desire to view the G/L account balances and line items.

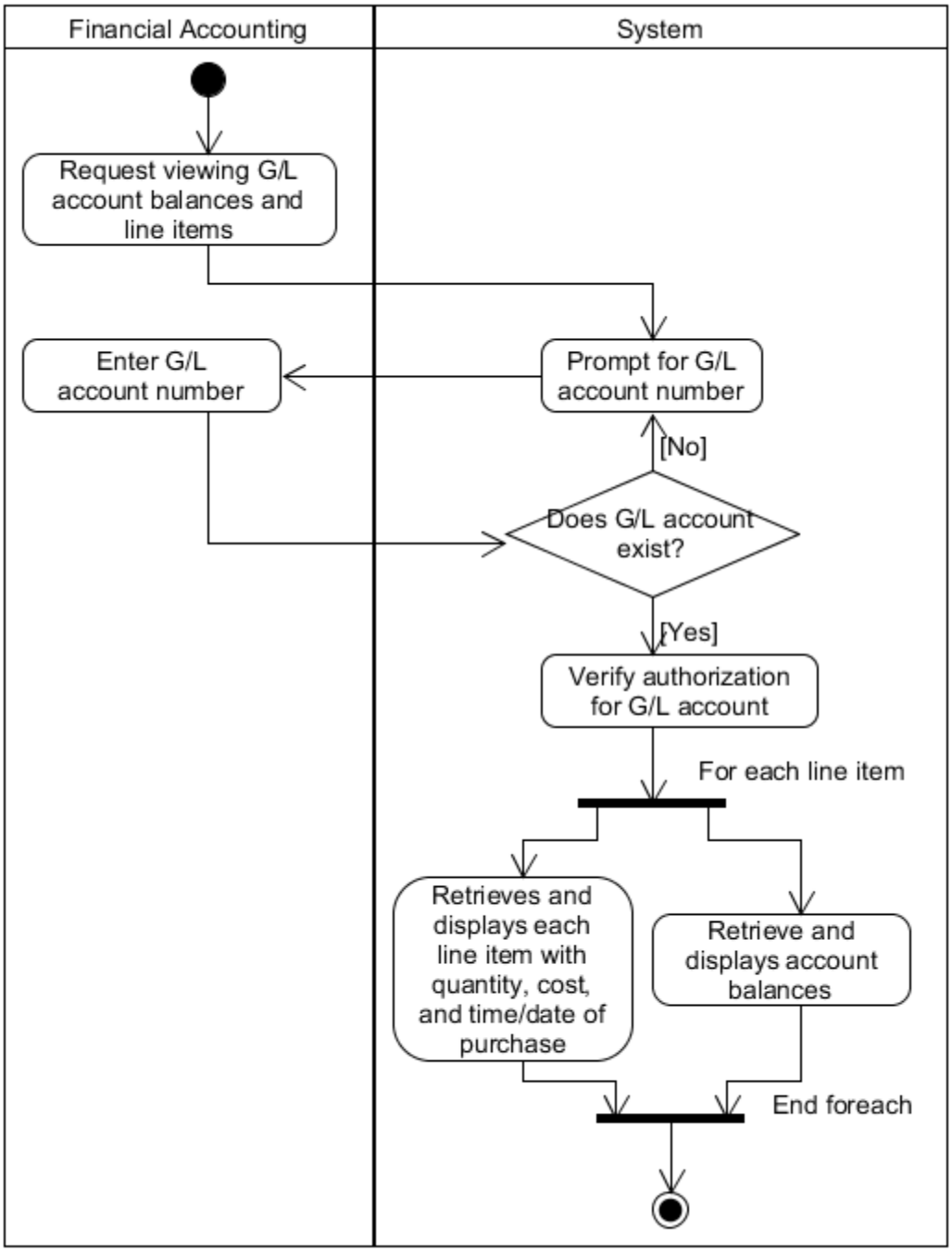
2. Financial Accounting employee enters account number.

.</td><td>1.1 System prompts user for G/L account number.

2.1 System verifies authorization for G/L account.
2.2 System retrieves and displays account balances.
2.3 System retrieves and displays each line item with quantity, cost, and time/date of purchase</td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1.1 G/L account number doesn’t exist
2.1 G/L account information isn’t valid or accessed denied.
2.3 No items exist</td></tr>
</table>

Subsystem: Financial Accounting Subsystem

Activity Diagram for use case <u>Display G/L account balances and line items</u>



Subsystem: Financial Accounting Subsystem

(Fully Developed) Use Case Description for use case <u>Post receipt of customer payment</u>

<table>
<tr><td>Use case name:</td><td colspan="2">Post receipt of customer payment</td></tr>
<tr><td>Scenario:</td><td colspan="2">Post receipt of customer payment</td></tr>
<tr><td>Triggering event:</td><td colspan="2">Financial Accounting employee wants to post the receipt to the customer</td></tr>
<tr><td>Brief description:</td><td colspan="2">Financial accounting enters a billing document and customer invoice number and a financial account number, and the system verifies the corresponding payment amount has been received, generates a payment receipt, and retrieves the address from the associated customer record.</td></tr>
<tr><td>Actors:</td><td colspan="2">Financial Accounting</td></tr>
<tr><td>Related use cases:</td><td colspan="2">Create invoice for customer use case must be satisfied before.</td></tr>
<tr><td>Stakeholders:</td><td colspan="2">Financial Accounting (department), Customer</td></tr>
<tr><td>Preconditions:</td><td colspan="2">Invoice must be sent to the customer
Financial Accounting subsystem must be available.
G/L account services must be available.</td></tr>
<tr><td>Post conditions:</td><td colspan="2">Customer must receive the receipt</td></tr>
<tr><td rowspan="3">Flow of activities:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Financial Accounting employee indicates desire to post a receipt to the customer.

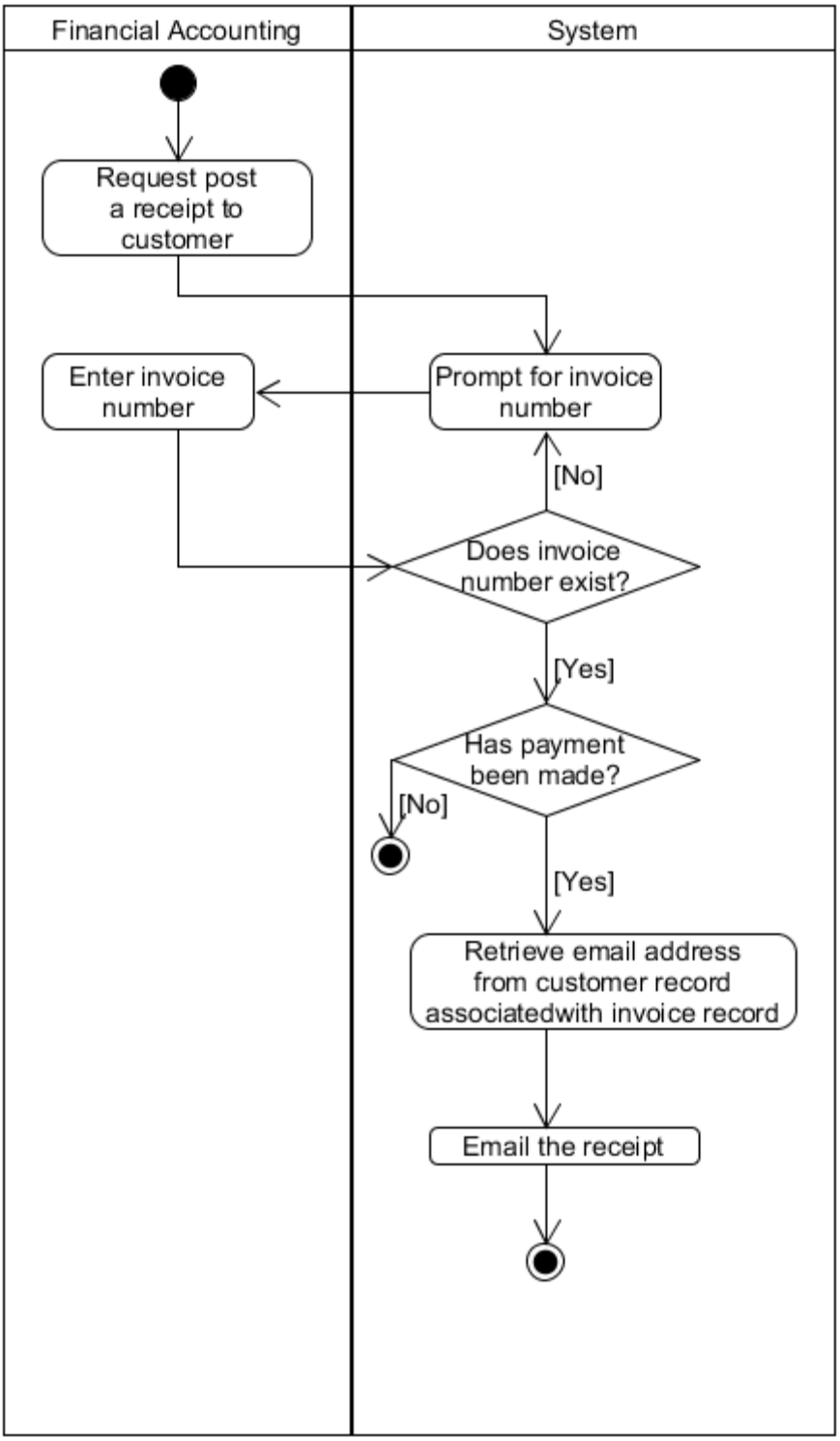
2. Financial Accounting employee sends the receipt.</td><td>1.1 System prompts user for the invoice number.

1.2 System checks whether the payment has been made by the customer

2.1 System sends the receipt if the invoice has been paid.</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>Exception conditions:</td><td colspan="2">1.1 Invoice number doesn’t exist
1.2 Invoice amount has not been by the customer</td></tr>
</table>

Subsystem: Financial Accounting Subsystem

Activity Diagram for use case <u>Post receipt of customer payment</u>



UML Domain Model Class Diagram

DETAILED DOCUMENT EXPLANATION

A problem domain is the specific area (domain) of the business that is being solved by the new system. For example, in a shipping system the problem domain may include a customer. 'Things' (domain classes) are data entities within the problem domain that end users interact with to accomplish their work, and that the solution system must work with and remember. In this way, a customer would be a considered a 'thing' in the shipping problem domain. A domain class is a classification used to describe a set of things (known as objects). An object is the UML term for an instance of a 'thing', such as a specific customer. Another perspective of defining these concepts is that a class can be thought of as a blueprint that each object is created from, for instance a car class could offer a template to creating a Toyota. A domain class are classes that describe objects (things) in a system's problem domain.

As defined in the **Use cases and UML Use Case Diagrams** document, Unified Modelling Language (UML) is a set of standards and notation for graphical models produced in system development. A UML class diagram is a diagram that represents classes (blueprints to create a set of objects) and the associations between classes. A UML domain model class diagram (DMCD) is a type of UML class diagram that only represents classes from the problem domain. It is used to demonstrate understanding of the project problem area (problem domain) by decomposing the system.

In a UML domain model class diagram, rectangles represent a class, and a line connecting two classes represents an association between classes. An association is the UML term for a relationship between these classes, for instance a UML association or relationship between a vendor and a package could be vendor 'sells' a package. An association can also have a multiplicity. Multiplicity refers to how many objects or instances of a class can exist in the relationship. For example, a vendor could sell many packages, however one specific package would one be sold by one vendor. This would be represented as 1..* if the diagram is read from right-to-left, indicating that many packages belong to one vendor as they can sell many items, and a multiplicity of 1 if the diagram is read from left-to-right, indicating that only one vendor sells or sends the specific package. Each class also can contain an attribute, which is a descriptive piece of information about an object. For instance, weight could be an attribute of the class Package. Attributes are listed in the second row of the rectangle that represent each class.

Additionally, UML DMCD utilises camelCase naming conventions, meaning that the words are concatenated to form a single word without spaces or an underscore and the first letter of each embedded word is capitalised, for example customerNumber. Class names are also distinguished by beginning the entire word with an uppercase letter such as CustomerAccount. This is known as upper camelCase. Constrastingly, an attribute of a class begins with a lowercase letter as demonstrated in customerNumber. This is known as lower camelCase.

As discussed, DMCD illustrates associations (also referred to as relationships) between classes. DMCD can also illustrate more complex associations, such as a generalisation/specialisation relationship. This is a hierarchical relationship where a more specific class, such as a Toyota, inherits the attributes from a more general class, a car. Here, the car is known as a superclass, and the Toyota class is known as a subclass. This relationship can be further illustrated by considering a Toyota IS A specific type of car. The generalisation/specialisation relationship is represented by an arrow that connects subclasses to the superclass. Subclasses are placed at the base of the arrow shaft, while superclasses are positioned at the end of the arrowhead.

As mentioned, classes are a blueprint that classify and can act as a template for each object to be created. This is known as a concrete class, where a class allows individual objects to be created. For instance a concrete class OnlineSale could create many objects that represent an Online Sale, giving a value to the timeSaleOccurred and the totalAmount of the sale. Generally, subclasses in a generalisation/specialisation relationship are a concrete class. However, a class can also act as a template for other classes, meaning that it does not create any objects. This is known as an abstract class and is represented by italicising the class name. Abstract class are generally used for superclasses (also known as parent classes), so that their subclasses (child classes) can inherit their attributes. As stated, contrastingly, subclasses are concrete classes. This creates their 'blueprint' from their own attributes, and those inherited from their parent class. Because they are concrete classes, objects can then be created using this more complex template. An example of these classes is an abstract class *Sale* which contains an attribute timeSaleoccurred, and two concrete classes *OnlineSale and InStoreSale* which inherit the timeSaleOccurred attribute from Sale, as well as having a timeOnSite and registerID attribute respectively. This use of abstact and concrete classes in a specialistaion/generalisation relationship can be viewed in the following DMCD for the superclasses *Production* and *BusinessPartner.*

The UML Domain Model Class Diagram includes additional notation and concepts **(APPENDIX E: UML Domain Model Class Diagram additional notation and concepts**) which is not included in the following DMCD.

KEYWORDS

UML Domain Model Class Diagram (DMCD)
A type of UML class model that represents classes (blueprints to create objects) only from the problem domain.

UML class diagram: A UML diagram that represents classes (blueprints to create a set of objects) and the associations between classes.
Problem domain: Specific area (domain) of the business that is being solved by the new system.
'Things': Data entities (domain classes) within the problem domain that end users interact with to accomplish their work, and that the new system must worth with and remember.
Domain class: A classification used to describe a set of 'things', that acts a blueprint or template during implementation which each object is created from. This is also referred to as a class in the DMCD as all classes in the DMCD are inside the problem domain. These are represented by rectangles.
Object: UML term for an *instance* of a thing.
Association: The UML term for a relationship between two classes. E.g a vendor could *sell* many packages. Represented as a line connecting two classes.
Multiplicity: Descriptor of an association to represent how many objects of the class can exist in the relationship.

- **1:** The class closest to this label can only have one object.
- **1..*:** The class closest to this label can have at least one object to unlimited objects created.
- **0..* or *:** The class closest to this label can have no objects, or unlimited objects created.

Attribute: Descriptive piece of information about an object, E.g weight could be an attribute of the class Package. They are listed in the second row of the rectangle that represent each class.
Camelcase: A naming convention that concatenates many words to form a single word without spaces, where the first letter of each embedded word is capitalized.

- Upper camelCase is used to name classes, where the first letter of the entire word is an uppercase letter.
- Lower camelCase is used to name attributes, where the first letter of the entire world is a lowercase letter.
- **1..*:** The class closest to this label can have at least one object to many objects created.

.

KEYWORDS

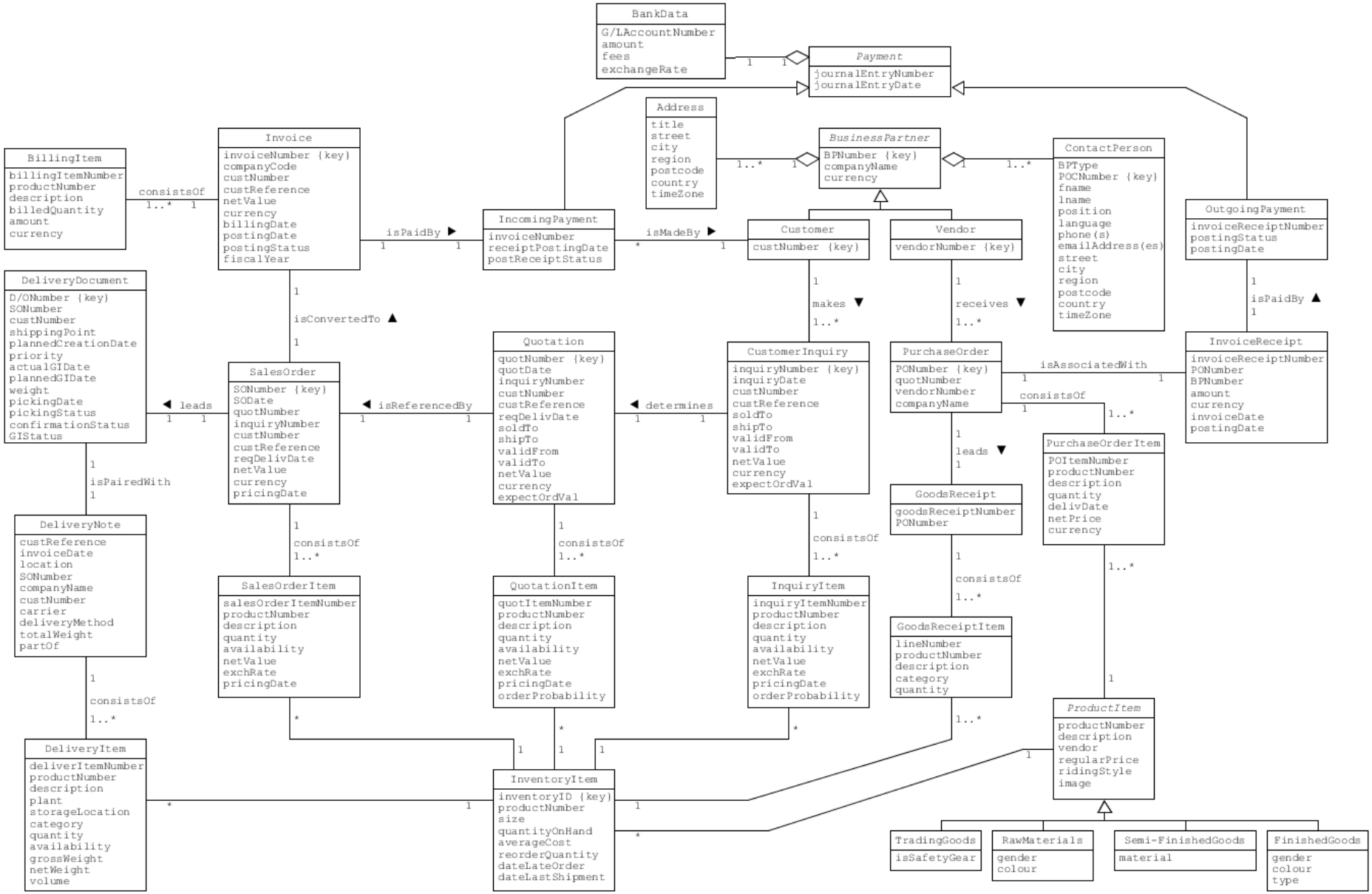
UML Domain Model Class Diagram (DMCD) (Continued):
A type of UML class model that represents classes (blueprints to create objects) only from the problem domain.

Generalisation/Specialisation relationship: Hierarchal relationship where a more specific class, such as the type of car Toyota, inherits attributes from a more general class, Car. It is represented by an arrow that connects the subclasses (placed as the base of the arrow shaft) to the superclass (end of the arrowhead).

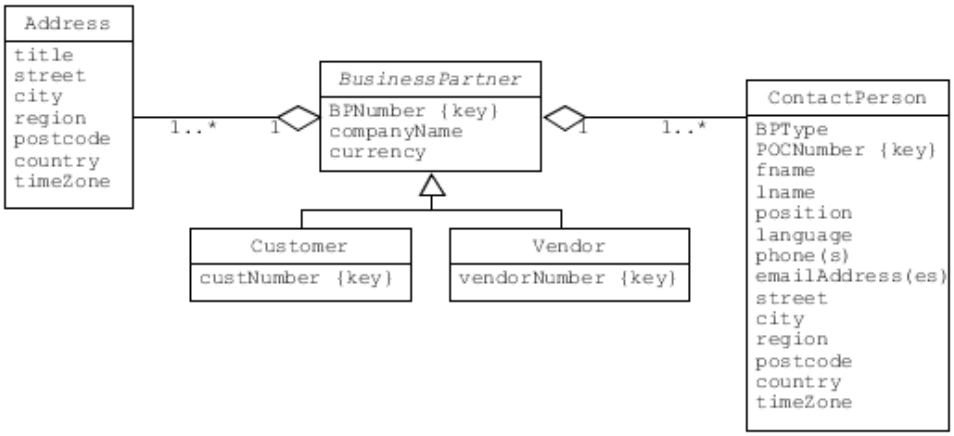
- **Superclass:** The parent class that is considered more general, E.g Car.
- **Subclass:** The child class that inherits attributes from the parent (superclass), E.g Toyota.

Abstract class: Does not create objects. Instead, acts as a template for other classes to inherit its attributes. Represented as an italicized class name. Generally a superclass, E.g *Sale* is an abstract superclass of OnlineSale and InStoreSale. This means that if *Sale* had an attribute timeSaleOccurred, both OnlineSale and InStoreSale would inerhit this attribute, as well as their own unique attributes such as timeOnSite for an OnlineSale and a registerID for an InStoreSale.

Concrete class: Can create objects. Can use the attributes inherited form an abstract superclass to create a more complex template for the objects created. E.g OnlineSale and InStoreSale are concrete subclasses of *Sale.*

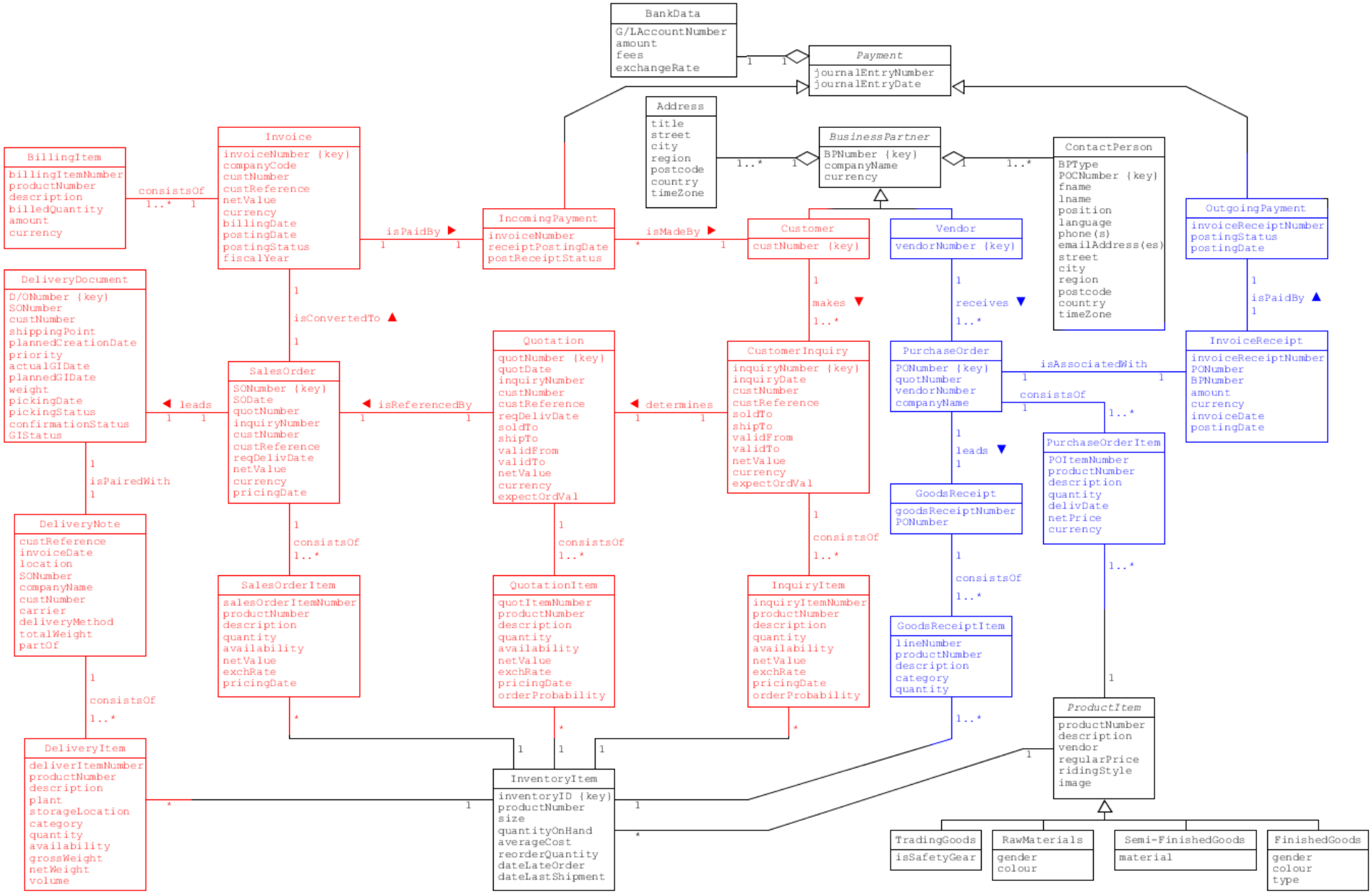

BankData
G/LAccountNumber
amount
fees
exchangeRate
Payment
journalEntryNumber
journalEntryDate
Address
title
street
city
region
postcode
country
timeZone
BusinessPartner
BPNumber {key}
companyName
currency
ContactPerson
BPType
POCNumber {key}
fname
lname
position
language
phone(s)
emailAddress(es)
street
city
region
postcode
country
timeZone
BillingItem
billingItemNumber
productNumber
description
billedQuantity
amount
currency
consistsOf
Invoice
invoiceNumber {key}
companyCode
custNumber
custReference
netValue
currency
billingDate
postingDate
postingStatus
fiscalYear
isPaidBy
IncomingPayment
invoiceNumber
receiptPostingDate
postReceiptStatus
isMadeBy
Customer
custNumber {key}
Vendor
vendorNumber {key}
OutgoingPayment
invoiceReceiptNumber
postingStatus
postingDate
isPaidBy
InvoiceReceipt
invoiceReceiptNumber
PONumber
BPNumber
amount
currency
invoiceDate
postingDate
makes
receives
DeliveryDocument
D/ONumber {key}
SONumber
custNumber
shippingPoint
plannedCreationDate
priority
actualGIDate
plannedGIDate
weight
pickingDate
pickingStatus
confirmationStatus
GIStatus
isConvertedTo
leads
SalesOrder
SONumber {key}
SODate
quotNumber
inquiryNumber
custNumber
custReference
reqDelivDate
netValue
currency
pricingDate
isReferencedBy
Quotation
quotNumber {key}
quotDate
inquiryNumber
custNumber
custReference
reqDelivDate
soldTo
shipTo
validFrom
validTo
netValue
currency
expectOrdVal
determines
CustomerInquiry
inquiryNumber {key}
inquiryDate
custNumber
custReference
soldTo
shipTo
validFrom
validTo
netValue
currency
expectOrdVal
PurchaseOrder
PONumber {key}
quotNumber
vendorNumber
companyName
isAssociatedWith
consistsOf
PurchaseOrderItem
POItemNumber
productNumber
description
quantity
delivDate
netPrice
currency
isPairedWith
DeliveryNote
custReference
invoiceDate
location
SONumber
companyName
custNumber
carrier
deliveryMethod
totalWeight
partOf
consistsOf
SalesOrderItem
salesOrderItemNumber
productNumber
description
quantity
availability
netValue
exchRate
pricingDate
QuotationItem
quotItemNumber
productNumber
description
quantity
availability
netValue
exchRate
pricingDate
orderProbability
InquiryItem
inquiryItemNumber
productNumber
description
quantity
availability
netValue
exchRate
pricingDate
orderProbability
GoodsReceipt
goodsReceiptNumber
PONumber
GoodsReceiptItem
lineNumber
productNumber
description
category
quantity
ProductItem
productNumber
description
vendor
regularPrice
ridingStyle
image
DeliveryItem
deliverItemNumber
productNumber
description
plant
storageLocation
category
quantity
availability
grossWeight
netWeight
volume
InventoryItem
inventoryID {key}
productNumber
size
quantityOnHand
averageCost
reorderQuantity
dateLateOrder
dateLastShipment
TradingGoods
isSafetyGear
RawMaterials
gender
colour
Semi-FinishedGoods
material
FinishedGoods
gender
colour
type

DOMAIN MODEL CLASS DIAGRAM | DETAILED DOCUMENT EXPLANATION



The system is organised around two categories of business partners (Customer and Vendor) and the processes (*order-to-cash* and *procure-to-pay*) in which they are involved.

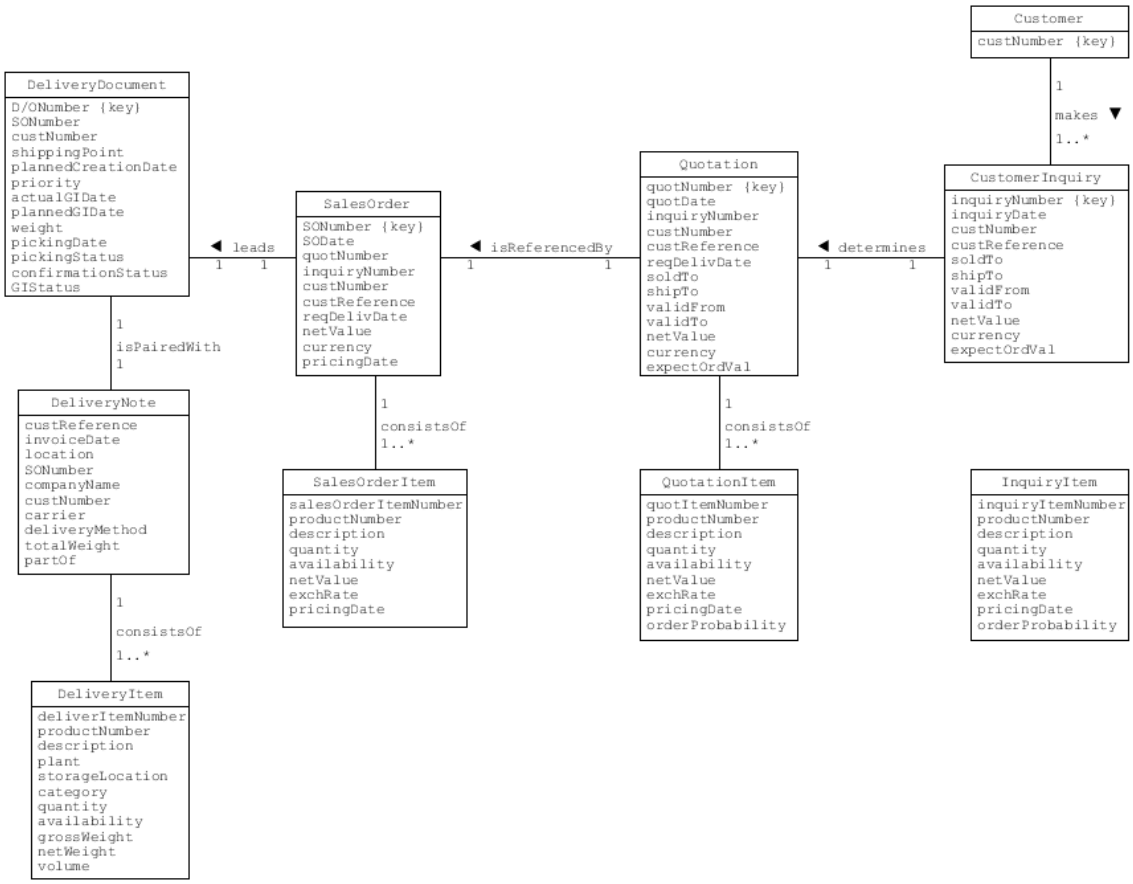
- As you will see in the following, to satisfy the requirements of various scenarios, a corporation has a unique number for itself (BPNumber) and another for its role (custNumber/vendorNumber).
- Even if the given data does not demonstrate the necessity for it, in the future, a corporation may play both roles (Customer and Vendor) at the same time, thus the system must keep more than one address and contact person.
- Address and Contact Person can identify the business type they belong to using the "title" and "BPType" fields.
- "POCNumber" is assigned to each contact person to help them be identified.
- Companies are from different countries, so defining the currency in use is critical in the following cases.
- Time zones have been specified for both the contact person and the address as it is useful knowledge for logistics and communication.



The red classes in the DMCD are tied to the order-to-cash business process, whereas the blue classes are related to the procure-to-pay business process

Customer

Order-To-Cash



This part of the DMCD focuses on only the order part of it for explanatory purposes.

After creating a customer account for a business, the customer can make an inquiry. It has a unique number to identify it easily. “pricingDate” field in its item list exists in case of the prices changes before the quotation is agreed upon.

- "custReference" is a number from the customer's system that aids in identifying the enquiry in both systems.
- “expectOrdVal” is the “expected order value” which is affected by the probability of items being ordered.
- It also includes a list of items.

Then, a quotation with values decided by the enquiry is generated and delivered to the customer for approval.

- It has “reqDelivDate” as the “ required delivery date”
- It has its own item list since it may be updated for any reason throughout the quote process. Such as a product is removed from sale after the inquiry.

If the customer accepts the quotation, a sales order with a quotation reference is produced.

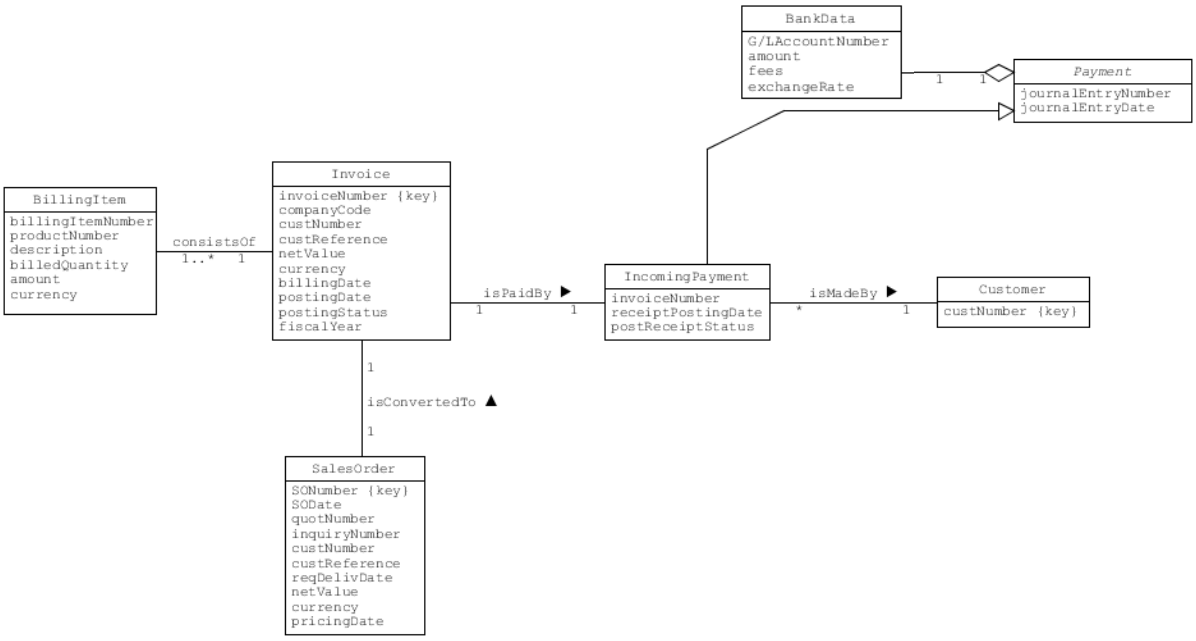
- It also has its own item list because the "order probability" field is no longer required because the deal is complete.

At the same time, the delivery process starts. Data from the sales order is used to create a delivery document, a delivery note and a list of items for it.

- DeliveryDocument
    - The fields that contain the abbreviation “GI” from the delivery document are related to the goods issue process.
        - plannedGIDate: Predicted goods issue date for the items.
        - *It is helpful for inventory management.*
        - actualGIDate: Actual date for the goods issue.
        - GIStatus: Whether the goods issue has been done.
    - plannedCreationDate: It is to plan the date for the creation of the delivery items.
    - confirmationStatus: Whether the delivery document is confirmed by the warehouse.
- DeliveryNote
    - The “partOf” field refers to a situation where the delivery has been separated into multiple parts.
    - custReference: It is needed for the customer to cross-check from their system during the delivery’s correctness.
- DeliveryItem
    - “plant”, “storageLocation” and “category” fields are to locate the item in (the) warehouse(s).

Customer

Order-**To-Cash**



The process starts by converting the sales order into an invoice.

Then, when the customer does the payment through their G/L account, the payment is recorded in the journal. At last, the data from the invoice and the payment is used to create a receipt as an output for the customer.

- Invoice
    - “custReference” is used again for the customer side cross-check from their system.
    - “billingDate” is the date customer is billed.
    - “postingDate” is when the invoice is posted to the customer.
    - “postingStatus” indicates whether the invoice is posted to the customer.
    - “fiscalYear” is the financial year of the invoice.
- IncomingPayment
    - “receiptPostingDate” is the date when the receipt is posted to the customer.
    - “postReceiptStatus” indicates whether the receipt is sent to the customer.
- Payment
    - “invoiceNumber” is for connecting the payment and the invoice.
    - “journalEntryNumber” is for the payment records in our system.
    - “journalEntryNumber” is the date when it is recorded in our system.
- “BankData” is a subsection for the payment. Because it has its own fields.

Vendor

Procure-To-Pay

The vendor receives our purchase order with the corresponding quotation number.

Our purchase order has a list of items that specify their quantity, price and delivery information.

Goods receipt is created according to purchase order.

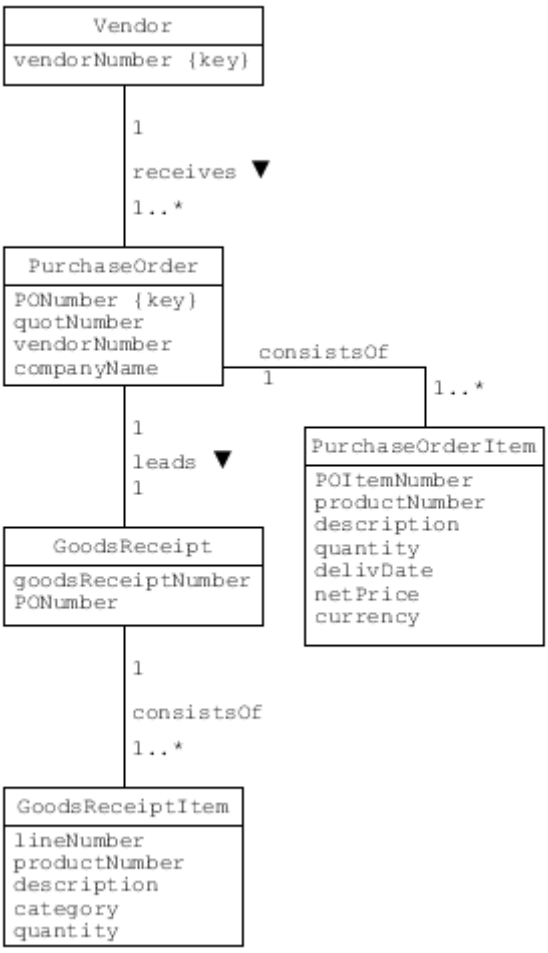
The goods receipt has its item list that is related to the warehouse.

- PurchaseOrder

  In reference to the successful vendor's quotation. The successful quotation is mentioned, and the details are imported into a new PO.

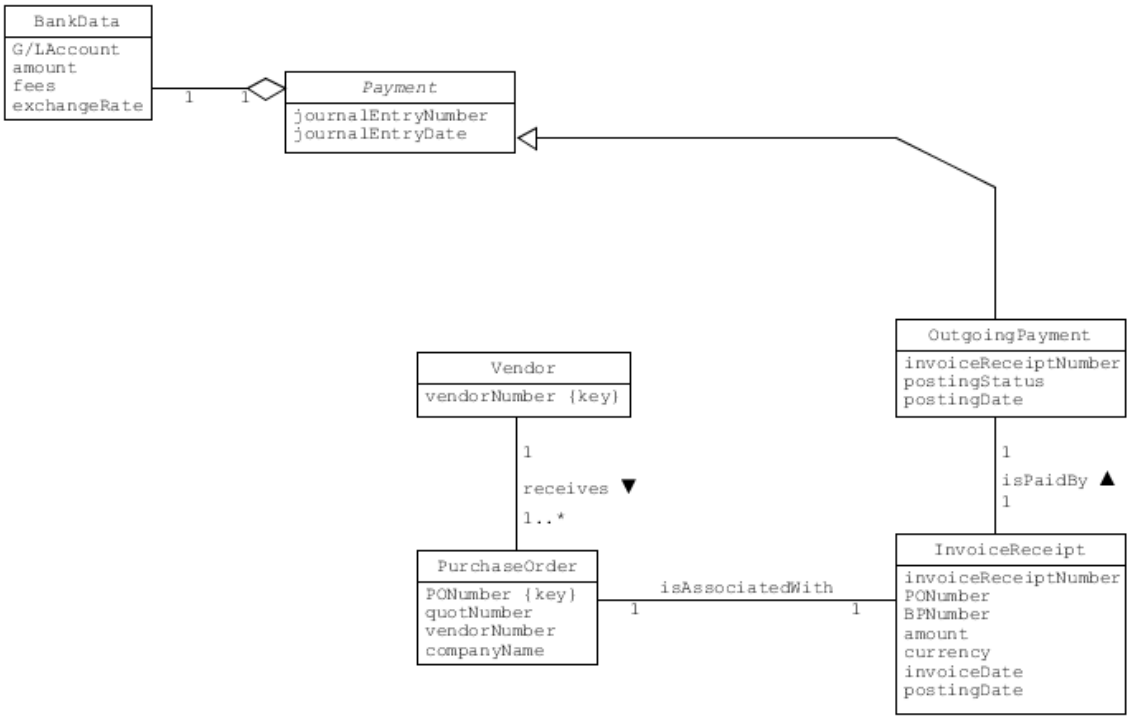
- GoodsReceipt:

  Take the remaining stock goods requested from the customer into inventory. A goods receipt document referencing our purchase order will be generated to ensure that we receive the products bought within the timeframes required and in excellent condition. Products on hand will be raised, and an accounting record acknowledging the worth of these goods will be created.



Vendor

Procure-**To-Pay**



After sending the purchase order, we receive an invoice from the vendor and we register the data into our system to process the payment.

- InvoiceReceipt:

  Customer invoice received in connection with the most recent PO and goods receipts. This invoice will be stored as an Accounts Payable to the client and sent to an existing G/L account in our Chart of Accounts. It will be resolved subsequently by issuing a cheque to the consumer. The invoice is submitted to back up the required journal entry.

- OutgoingPayment:

  A payment was made to the vendor to satisfy their total Accounts Payable debt. In the G/L, a journal entry is submitted to Accounts Payable for the vendor and the bank checking account.

  - postingStatus: Whether the payment has been placed.
  - postingDate: The date when the payment is placed.

60DEGREES
SYSTEMS SOLUTIONS

Conclusion

As a result of the findings presented, it is critical that Global Bike Incorporated takes advantage of the digital opportunities by implementing an information system to improve synergy between departments and business procedures, thereby increasing efficiency, accuracy, employee morale, and customer service. Project UPSHIFT recommends integrating and streamlining the procure-to-cash and order-to-cash processes in a custom ERP application that will enable departments to make modifications to a central database. This will boost productivity by allowing accurate business data to be stored and retrieved promptly, as well as improving customer and vendor relations with targeted communication via management subsystems and automated emailing processes. Strengthening ties with these business partners will allow Global Bike Incorporated to remain at the forefront of new vendor products and invite customers to interact with the company's rapid service.

The new system will particularly benefit corporate strategy and insight into business analytics by integrating the management of General Ledger financial accounts. Additional intuitive features to assist in business functions such as searching for products, vendors, financial accounts, and customers, as well as managing these domain entities by updating, viewing, and removing information emphasises the quality-of-life improvements offered by the solution.

Based on the presented systems analysis, if Project UPSHIFT is accepted, the team will move onto the design and implementation of the solution system. Systems design will use the current models and depthful understanding of the system as input into design diagrams. Subsequently, implementation will be considered from the produced systems design documents, using iDempiere to customise an effective final solution.

60DEGREES
SYSTEMS SOLUTIONS

References

- 'The Ultimate Guide to a Truly Effective Procure-to-Pay Process',(2022) <http://saphelp.ucc.ovgu.de/NW750/EN/83/a7c6535e601e4be10000000a174cb4/content.htm>
- 2012 'iDempiere', Community Powered Enterprise, <https://www.idempiere.org/>.
- 2012 "iDempiere" 8.2 Phong , viewed 30 May 2022.
- Bridges, J 2021," What Is a Feasibility Study? How to Conduct One for Your Project", PROJECT MANAGEMENT, viewed 14 April 2022, <https://www.projectmanager.com/training/how-to-conduct-a-feasibility-study>
- Chigona, W 2010, 'Investigating the impact of Stakeholder Management on the implementation of a of a Public Access Project: Case of Smart Cape - Scientific Figure, ResearchGate, viewed 10 April 2022, <https://www.researchgate.net/figure/Power-and-Interest-grid-Freeman-1984_fig1_266608284>
- Dauda, A 2021, " Types and Benefits of Information Systems for Organizations to Know",Bizinfong, viewed 22 April 2022, <https://www.bizinfong.com/business/benefits-of-information-systems/>
- Hayes, K 2017," Benefits of ERP: Advantages and Disadvantages of an Enterprise Resource Planning System ", SelectHub, viewed on 28 April 2022, <https://www.selecthub.com/enterprise-resource-planning/erp-advantages-and-disadvantages/>
- McAbee,J 2022, "How to Do a Feasibility Study", wrike, viewed on 30 April 2022, <https://www.wrike.com/blog/how-to-do-a-feasibility-study/>
- Price, M 2021, " The Order To Cash Process(O2C) - and Ways To Improve This", corcentric, viewed 18 April 2022, <https://www.corcentric.com/blog/the-order-to-cash-process-o2c-and-ways-to-improve-this/#:~:text=The%20order%2Dto%2Dcash%20process%20encompasses%20all%20steps%20from%20when,invoicing%20and%20ultimately%20payment%20collection.>
- Satzinger, J , Jackson, R & Burd, S. 2016, 'Systems Analysis And Design In A Changing World', 7 Edition, Cengage Learning, United States of America, viewed 20 March 2022.
- Schwalbe, K 2018, 'Information Technology Project Management', Cengage Learning, United States of America. "Document Flow", viewed 23 April 2022.

60DEGREES
SYSTEMS SOLUTIONS

Glossary

General Leger (G/L): A set of numbered accounts a business uses to track its financial transactions and to prepare financial reports

Goods Receipt: Acknowledge the receipt of goods from a vendor and into the warehouse, thus increasing the stock.

Request For Quote (RFQ): Term referring to the company requesting vendors to submit their price quotes for the opportunity to fulfill the purchase order

Purchase Order: Official offer issued by a buyer to the vendor, containing types, quantities and agreed costs for each product.

Software Development Life Cycle (SDLC)

Sales Order: Document by seller to confirm the sales of goods or services

Materials Delivery Note: Document produced by the shipping of goods that lists the products and quantity of products contained in the shipment, and that is sent out to the customer. The price of the goods is typically contained on the invoice.

Post Goods Issue: The delivery of goods ownership from the warehouse to the buyer resulting in a reduction of present stock.

Invoice: A list of goods issued by a seller to a buyer which indicates items, quantities, and their costs.

Billing Document: A sales invoice which indicates items, quantities, and their costs.

Customer invoice: An invoice which indicates items, quantities and costs that is received by the customer.

Document Flow: A list of consecutive documents show how far a sales document has been processed, including a quotation, sales order, delivery note and an invoice.

Systems Vision Document: Document developed to showcase understanding of the company's initial problem, system capabilities that the proposed solution will offer, and the resulting benefits of the system to Global Bike Incorporated.

Vendor Line Item: A list of items ordered from a vendor. Each item and its details are a line item of the purchase order.

Invoice Receipt: Whereas an invoice is issued by the vendor to request a payment, the invoice receipt acts as a proof of payment.

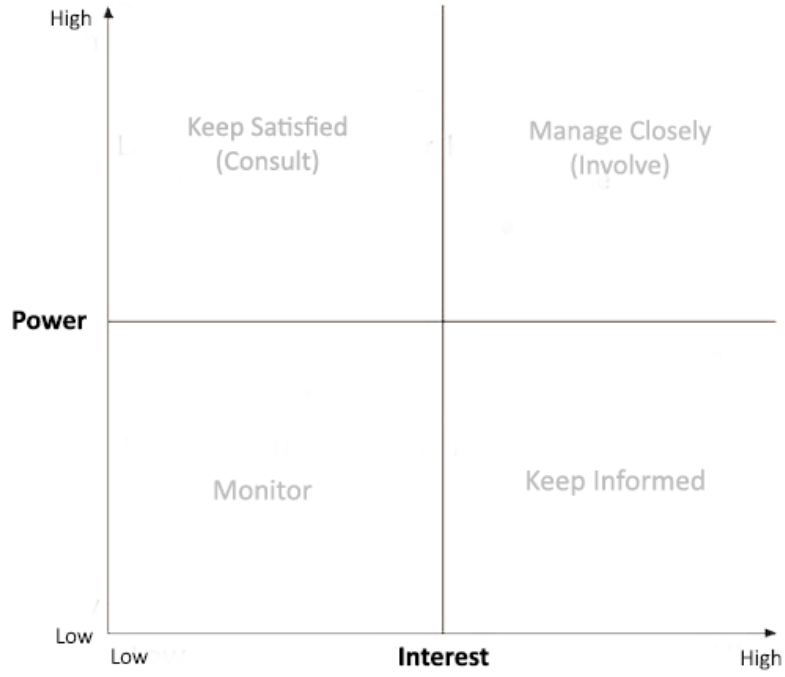
Appendices

The appendices contains lengthy materials that may compromise readability, or are unsuitable for the body of the report, but are helpful to understand the system proposal. Such figures involve diagrams, charts and other supporting documents.

Appendix A

Power-Interest grid

A matrix used to assess stakeholders based on their power or influence and interest in a project, established by Colin Eden and Fran Ackermann in their novel *Making Strategy*. This grid provides insight into management techniques for stakeholder communication and satisfaction.



Appendix B

Sample alternative stakeholder analysis table format

Alternatively, to the power-interest grid **(APPENDIX A)**, a table can be used to assess stakeholders based on their power or influence and interest in a project has been demonstrated by Kathy Schwalbe in their publication *Information Technology Project Management*. This table provides further insight into management techniques for stakeholder communication and satisfaction.

Table 13-2. Sample stakeholder analysis

Name	Power/Interest	Current Engagement	Potential Management Strategies
Stephen	High/high	Leading	Stephen can seem intimidating due to his physical stature and deep voice, but he has a great personality and sense of humor. He previously led a similar refinery upgrade program at another company and knows what he wants. Manage closely and ask for his advice as needed. He likes short, frequent updates in person.
Chien	High/medium	Resistant	Chien is a very organized yet hardheaded man. He has been pushing corporate IT standards, and the system the PM and sponsor (Debra and Stephen) like best goes against those standards, even though it's the best solution for this project and the company as a whole. Need to convince him that this is okay and that people still respect his work and position.
Ryan	Medium/high	Supportive	Ryan has been with the company for several years and is well respected, but he feels threatened by Debra. He also resents her getting paid more than he does. He wants to please his boss, Chien, first and foremost. Need to convince him that the suggested solution is in everyone's best interest.
Betsy	High/low	Neutral	Very professional, logical person. Gets along well with Chien. She has supported Debra in approving past projects with strong business cases. Provide detailed financial justification for the suggested solution to keep her satisfied. Also ask her to talk to Chien on Debra's behalf.

Source: Schwalbe (2018)

Appendix C: Sample Document Flow

A formal example created by Systems Application and Products (SAP) to illustrate the features of the document flow.

Document flow

Document	Date	Complete processing status
. Sales order 5362	June 10	Completed
Delivery 80003721	June 10	Completed
. Picking order 19980610	June 10	Completed
. Goods delivery 49007080	June 10	Completed
. Invoice 90005596	June 10	Completed
.. Accounting document 100007729	June 10	Not cleared

Source: SAP SE (2021)

APPENDIX D: Dividing the system into subsystems

Given all of the use cases for the proposed solution derived from Global Bike Incorporated's Procure-to-Cash and Order-to-cash processes, the following figure outlines the process for dividing the system.

The new system will have five subsystems:

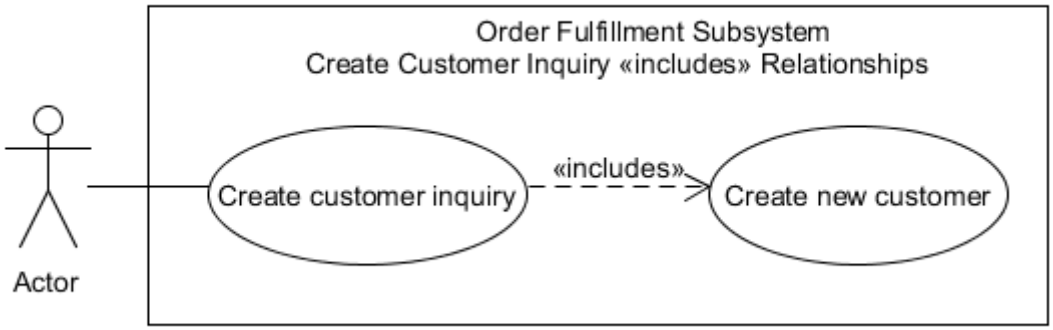
- **Order Fulfillment subsystem:** Will perform all the tasks of creating and fulfilling a sales order and its associated delivery process. From point of sales inquiry to delivery of a product to the customer.
- **Customer Account subsystem:** Provides services that manage and enhance the customer experience. Employees can quickly create, view, search and maintain customer account information.
- **Supply Chain subsystem:** Is responsible for managing inventory levels and adding products, as well as the vendors that offer
- **Financial Accounting subsystem:** Provides functions for employees to manage and review financial information.
- **Marketing subsystem:** Is primarily for employees to set up the information and services for customers to accurately purchase biking products.

This decomposition is based on **page 112** (Worked example of grouping use cases into subsystems) of referenced and **page 43** (Worked example of dividing system into subsystems) in 'Systems Analysis And Design In A Changing World'.

Satzinger, J , Jackson, R & Burd, S. 2016, 'Systems Analysis And Design In A Changing World', 7 Edition, Cengage Learning, United States of America, viewed 20 March 2022.

APPENDIX E: UML Use Case Diagrams additional notations

A dashed arrow can also be used with the text label '<<includes>> connecting a use case to another use case. This represents that the use case connected to the base of the arrow <<includes>> or calls the use case at the arrowhead end.



Concept source: Satzinger, Jackson, R & Burd (2016)

APPENDIX F: UML Domain Model Class Diagram additional notation and concepts:

These notations and concepts were not included in the project's DMCD, however are additional concepts:

An association is the UML term for a naturally occurring relationship between two 'things' or classes.

Whole-part relationships: Association that represents one class is a part or component of another class

- **Aggregation**: A whole-part relationship where the component parts also exist as individual objects that will not be destroyed if the 'whole' class is destroyed. Indicated by a hollow diamond symbol connecting the 'part' class to the 'whole'.
- **Composition**: A whole-part relationship where the component parts cannot exist as individual objects therefore they are also destroyed if the 'whole' class is destroyed. Indicated by a filled in diamond symbol connecting the 'part' class to the 'whole'.

Association class: When the association between two classes has its own class. For instance, an association EnrolmentEvent may exist between a Student class and a University class. It is represented by a dotted line branching from the association line.